Marian Bubak
Geert Dick van Albada
Peter M.A. Sloot
Jack J. Dongarra (Eds.)

LNCS 3037

Computational Science – ICCS 2004

4th International Conference
Kraków, Poland, June 2004
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 3037

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA
Josef Kittler
University of Surrey, Guildford, UK
Jon M. Kleinberg
Cornell University, Ithaca, NY, USA
Friedemann Mattern
ETH Zurich, Switzerland
John C. Mitchell
Stanford University, CA, USA
Moni Naor
Weizmann Institute of Science, Rehovot, Israel
Oscar Nierstrasz
University of Bern, Switzerland
C. Pandu Rangan
Indian Institute of Technology, Madras, India
Bernhard Steffen
University of Dortmund, Germany
Madhu Sudan
Massachusetts Institute of Technology, MA, USA
Demetri Terzopoulos
New York University, NY, USA
Doug Tygar
University of California, Berkeley, CA, USA
MosheY.Vardi
Rice University, Houston, TX, USA
Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Marian Bubak Geert Dick van Albada
Peter M.A. Sloot Jack J. Dongarra (Eds.)

Computational Science - ICCS 2004

4th International Conference
Kraków, Poland, June 6-9, 2004
Proceedings, Part II

Springer

eBook ISBN: 3-540-24687-8
Print ISBN: 3-540-22115-8

©2005 Springer Science + Business Media, Inc.

Print ©2004 Springer-Verlag
Berlin Heidelberg

All rights reserved

No part of this eBook may be reproduced or transmitted in any form or by any means, electronic, mechanical, recording, or otherwise, without written consent from the Publisher

Created in the United States of America

Visit Springer's eBookstore at: http://ebooks.springerlink.com
and the Springer Global Website Online at: http://www.springeronline.com

Preface

The International Conference on Computational Science (ICCS 2004) held in Kraków, Poland, June 6–9, 2004, was a follow-up to the highly successful ICCS 2003 held at two locations, in Melbourne, Australia and St. Petersburg, Russia; ICCS 2002 in Amsterdam, The Netherlands; and ICCS 2001 in San Francisco, USA.

As computational science is still evolving in its quest for subjects of investigation and efficient methods, ICCS 2004 was devised as a forum for scientists from mathematics and computer science, as the basic computing disciplines and application areas, interested in advanced computational methods for physics, chemistry, life sciences, engineering, arts and humanities, as well as computer system vendors and software developers. The main objective of this conference was to discuss problems and solutions in all areas, to identify new issues, to shape future directions of research, and to help users apply various advanced computational techniques. The event harvested recent developments in computational grids and next generation computing systems, tools, advanced numerical methods, data-driven systems, and novel application fields, such as complex systems, finance, econo-physics and population evolution.

Keynote lectures were delivered by David Abramson and Alexander V. Bogdanov, *From ICCS 2003 to ICCS 2004 – Personal Overview of Recent Advances in Computational Science;* Iain Duff, *Combining Direct and Iterative Methods for the Solution of Large Sparse Systems in Different Application Areas;* Chris Johnson, *Computational Multi-field Visualization;* John G. Michopoulos, *On the Pathology of High Performance Computing;* David De Roure, *Semantic Grid;* and Vaidy Sunderam, *True Grid*: *What Makes a Grid Special and Different?* In addition, three invited lectures were delivered by representatives of leading computer system vendors, namely: Frank Baetke from Hewlett Packard, Eng Lim Goh from SGI, and David Harper from the Intel Corporation.

Four tutorials extended the program of the conference: Paweł Płaszczak and Krzysztof Wilk, *Practical Introduction to Grid and Grid Services*; Grzegorz Młynarczyk, *Software Engineering Methods for Computational Science;* the *CrossGrid Tutorial* by the CYFRONET CG team; and the Intel tutorial.

We would like to thank all keynote, invited and tutorial speakers for their interesting and inspiring talks.

Aside of plenary lectures, the conference included 12 parallel oral sessions and 3 poster sessions. Ever since the first meeting in San Francisco, ICCS has attracted an increasing number of more researchers involved in the challenging field of computational science. For ICCS 2004, we received 489 contributions for the main track and 534 contributions for 41 originally-proposed workshops. Of these submissions, 117 were accepted for oral presentations and 117 for posters in the main track, while 328 papers were accepted for presentations at 30 workshops. This selection was possible thanks to the hard work of the Program

Committee members and 477 reviewers. The author index contains 1395 names, and almost 560 persons from 44 countries and all continents attended the conference: 337 participants from Europe, 129 from Asia, 62 from North America, 13 from South America, 11 from Australia, and 2 from Africa.

The ICCS 2004 proceedings consists of four volumes, the first two volumes, LNCS 3036 and 3037 contain the contributions presented in the main track, while volumes 3038 and 3039 contain the papers accepted for the workshops. Parts I and III are mostly related to pure computer science, while Parts II and IV are related to various computational research areas. For the first time, the ICCS proceedings are also available on CD. We would like to thank Springer-Verlag for their fruitful collaboration. During the conference the best papers from the main track and workshops as well as the best posters were nominated and presented on the ICCS 2004 Website. We hope that the ICCS 2004 proceedings will serve as a major intellectual resource for computational science researchers, pushing back the boundaries of this field. A number of papers will also be published as special issues of selected journals.

We owe thanks to all workshop organizers and members of the Program Committee for their diligent work, which ensured the very high quality of the event. We also wish to specifically acknowledge the collaboration of the following colleagues who organized their workshops for the third time: Nicoletta Del Buono (New Numerical Methods) Andres Iglesias (Computer Graphics), Dieter Kranzlmueller (Tools for Program Development and Analysis), Youngsong Mun (Modeling and Simulation in Supercomputing and Telecommunications).

We would like to express our gratitude to Prof. Ryszard Tadeusiewicz, Rector of the AGH University of Science and Technology, as well as to Prof. Marian Noga, Prof. Kazimierz Jeleń, Dr. Jan Kulka and Prof. Krzysztof Zieliński, for their personal involvement. We are indebted to all the members of the Local Organizing Committee for their enthusiastic work towards the success of ICCS 2004, and to numerous colleagues from ACC CYFRONET AGH and the Institute of Computer Science for their help in editing the proceedings and organizing the event. We very much appreciate the help of the Computer Science and Computational Physics students during the conference. We owe thanks to the ICCS 2004 sponsors: Hewlett-Packard, Intel, IBM, SGI and ATM, SUN Microsystems, Polish Airlines LOT, ACC CYFRONET AGH, the Institute of Computer Science AGH, the Polish Ministry for Scientific Research and Information Technology, and Springer-Verlag for their generous support.

We wholeheartedly invite you to once again visit the ICCS 2004 Website (http://www.cyfronet.krakow.pl/iccs2004/), to recall the atmosphere of those June days in Kraków.

June 2004

Marian Bubak, Scientific Chair 2004
on behalf of the co-editors:
G. Dick van Albada
Peter M.A. Sloot
Jack J. Dongarra

Organization

ICCS 2004 was organized by the Academic Computer Centre CYFRONET AGH University of Science and Technology (Kraków, Poland) in cooperation with the Institute of Computer Science AGH, the University of Amsterdam (The Netherlands) and the University of Tennessee (USA).

All the members of the Local Organizing Committee are the staff members of CYFRONET and/or ICS. The conference took place at the premises of the Faculty of Physics and Nuclear Techniques AGH and at the Institute of Computer Science AGH.

Conference Chairs

Scientific Chair – Marian Bubak (Institute of Computer Science and ACC CYFRONET AGH, Poland)
Workshop Chair – Dick van Albada (University of Amsterdam, The Netherlands)
Overall Chair – Peter M.A. Sloot (University of Amsterdam, The Netherlands)
Overall Co-chair – Jack Dongarra (University of Tennessee, USA)

Local Organizing Committee

Marian Noga
Marian Bubak
Zofia Mosurska
Maria Stawiarska
Milena Zając
Mietek Pilipczuk
Karol Frańczak
Aleksander Kusznir

Program Committee

Jemal Abawajy (Carleton University, Canada)
David Abramson (Monash University, Australia)
Dick van Albada (University of Amsterdam, The Netherlands)
Vassil Alexandrov (University of Reading, UK)
Srinivas Aluru (Iowa State University, USA)
David A. Bader (University of New Mexico, USA)

J.A. Rod Blais (University of Calgary, Canada)
Alexander Bogdanov (Institute for High Performance Computing and Information Systems, Russia)
Peter Brezany (University of Vienna, Austria)
Marian Bubak (Institute of Computer Science and CYFRONET AGH, Poland)
Rajkumar Buyya (University of Melbourne, Australia)
Bastien Chopard (University of Geneva, Switzerland)
Paul Coddington (University of Adelaide, Australia)
Toni Cortes (Universitat Politècnica de Catalunya, Spain)
Yiannis Cotronis (University of Athens, Greece)
Jose C. Cunha (New University of Lisbon, Portugal)
Brian D'Auriol (University of Texas at El Paso, USA)
Federic Desprez (INRIA, France)
Tom Dhaene (University of Antwerp, Belgium)
Hassan Diab (American University of Beirut, Lebanon)
Beniamino Di Martino (Second University of Naples, Italy)
Jack Dongarra (University of Tennessee, USA)
Robert A. Evarestov (SPbSU, Russia)
Marina Gavrilova (University of Calgary, Canada)
Michael Gerndt (Technical University of Munich, Germany)
Yuriy Gorbachev (Institute for High Performance Computing and Information Systems, Russia)
Andrzej Goscinski (Deakin University, Australia)
Ladislav Hluchy (Slovak Academy of Sciences, Slovakia)
Alfons Hoekstra (University of Amsterdam, The Netherlands)
Hai Jin (Huazhong University of Science and Technology, ROC)
Peter Kacsuk (MTA SZTAKI Research Institute, Hungary)
Jacek Kitowski (AGH University of Science and Technology, Poland)
Dieter Kranzlmüller (Johannes Kepler University Linz, Austria)
Domenico Laforenza (Italian National Research Council, Italy)
Antonio Lagana (Università di Perugia, Italy)
Francis Lau (University of Hong Kong, ROC)
Bogdan Lesyng (ICM Warszawa, Poland)
Thomas Ludwig (Ruprecht-Karls-Universität Heidelberg, Germany)
Emilio Luque (Universitat Autònoma de Barcelona, Spain)
Michael Mascagni (Florida State University, USA)
Edward Moreno (Euripides Foundation of Marilia, Brazil)
Jiri Nedoma (Institute of Computer Science AS CR, Czech Republic)
Genri Norman (Russian Academy of Sciences, Russia)
Stephan Olariu (Old Dominion University, USA)
Salvatore Orlando (University of Venice, Italy)
Marcin Paprzycki (Oklahoma State University, USA)
Ron Perrott (Queen's University of Belfast, UK)
Richard Ramaroson (ONERA, France)
Rosemary Renaut (Arizona State University, USA)

Alistair Rendell (Australian National University, Australia)
Paul Roe (Queensland University of Technology, Australia)
Hong Shen (Japan Advanced Institute of Science and Technology, Japan)
Dale Shires (U.S. Army Research Laboratory, USA)
Peter M.A. Sloot (University of Amsterdam, The Netherlands)
Gunther Stuer (University of Antwerp, Belgium)
Vaidy Sunderam (Emory University, USA)
Boleslaw Szymanski (Rensselaer Polytechnic Institute, USA)
Ryszard Tadeusiewicz (AGH University of Science and Technology, Poland)
Pavel Tvrdik (Czech Technical University, Czech Republic)
Putchong Uthayopas (Kasetsart University, Thailand)
Jesus Vigo-Aguiar (University of Salamanca, Spain)
Jens Volkert (University of Linz, Austria)
Koichi Wada (University of Tsukuba, Japan)
Jerzy Wasniewski (Technical University of Denmark, Denmark)
Greg Watson (Los Alamos National Laboratory, USA)
Jan Węglarz (Poznań University of Technology, Poland)
Roland Wismüller (LRR-TUM, Germany)
Roman Wyrzykowski (Technical University of Częstochowa, Poland)
Jinchao Xu (Pennsylvania State University, USA)
Yong Xue (Chinese Academy of Sciences, ROC)
Xiaodong Zhang (College of William and Mary, USA)
Alexander Zhmakin (Soft-Impact Ltd, Russia)
Krzysztof Zieliński (Institute of Computer Science and CYFRONET AGH, Poland)
Zahari Zlatev (National Environmental Research Institute, Denmark)
Albert Zomaya (University of Sydney, Australia)
Elena Zudilova (University of Amsterdam, The Netherlands)

Reviewers

Abawajy, J.H.
Abe, S.
Abramson, D.
Adali, S.
Adcock, M.
Adriaansen, T.
Ahn, G.
Ahn, S.J.
Albada, G.D. van
Albuquerque, P.
Alda, W.
Alexandrov, V.
Alt, M.
Aluru, S.
Anglano, C.
Archibald, R.
Arenas, A.
Astalos, J.
Ayani, R.
Ayyub, S.
Babik, M.
Bader, D.A.
Bajaj, C.
Baker, M.
Baliś, B.
Balk, I.
Balogh, Z.
Bang, Y.C.
Baraglia, R.
Barron, J.
Baumgartner, F.
Becakaert, P.
Belleman, R.G.
Bentes, C.
Bernardo Filho, O.
Beyls, K.
Blais, J.A.R.
Boada, I.
Bode, A.

Bogdanov, A.
Bollapragada, R.
Boukhanovsky, A.
Brandes, T.
Brezany, P.
Britanak, V.
Bronsvoort, W.
Brunst, H.
Bubak, M.
Budinska, I.
Buono, N. Del
Buyya, R.
Cai, W.
Cai, Y.
Cannataro, M.
Carbonell, N.
Carle, G.
Caron, E.
Carothers, C.
Castiello, C.
Chan, P.
Chassin-de-Kergommeaux, J.
Chaudet, C.
Chaves, J.C.
Chen, L.
Chen, Z.
Cheng, B.
Cheng, X.
Cheung, B.W.L.
Chin, S.
Cho, H.
Choi, Y.S.
Choo, H.S.
Chopard, B.
Chuang, J.H.
Chung, R.
Chung, S.T.
Coddington, P.
Coeurjolly, D.
Congiusta, A.
Coppola, M.
Corral, A.
Cortes, T.
Cotronis, Y.
Cramer, H.S.M.
Cunha, J.C.
Danilowicz, C.
D'Auriol, B.
Degtyarev, A.
Denazis, S.
Derntl, M.
Desprez, F.
Devendeville, L.
Dew, R.
Dhaene, T.
Dhoedt, B.
D'Hollander, E.
Diab, H.
Dokken, T.
Dongarra, J.
Donnelly, D.
Donnelly, W.
Dorogovtsev, S.
Duda, J.
Dudek-Dyduch, E.
Dufourd, J.F.
Dumitriu, L.
Duplaga, M.
Dupuis, A.
Dzwinel, W.
Embrechts, M.J.
Emiris, I.
Emrich, S.J.
Enticott, C.
Evangelos, F.
Evarestov, R.A.
Fagni, T.
Faik, J.
Fang, W.J.
Farin, G.
Fernandez, M.
Filho, B.O.
Fisher-Gewirtzman, D.
Floros, E.
Fogel, J.
Foukia, N.
Frankovic, B.
Fuehrlinger, K.
Funika, W.
Gabriel, E.
Gagliardi, F.
Galis, A.
Galvez, A.
Gao, X.S.
Garstecki, L.
Gatial, E.
Gava, F.
Gavidia, D.P.
Gavras, A.
Gavrilova, M.
Gelb, A.
Gerasimov, V.
Gerndt, M.
Getov, V.
Geusebroek, J.M.
Giang, T.
Gilbert, M.
Glasner, C.
Gobbert, M.K.
Gonzalez-Vega, L.
Gorbachev, Y.E.
Goscinski, A.M.
Goscinski, W.
Gourhant, Y.
Gualandris, A.
Guo, H.
Ha, R.
Habala, O.
Habib, A.
Halada, L.
Hawick, K.
He, K.
Heinzlreiter, P.
Heyfitch, V.
Hisley, D.M.
Hluchy, L.
Ho, R.S.C.
Ho, T.
Hobbs, M.
Hoekstra, A.
Hoffmann, C.
Holena, M.
Hong, C.S.
Hong, I.

Hong, S.
Horan, P.
Hu, S.M.
Huh, E.N.
Hutchins, M.
Huynh, J.
Hwang, I.S.
Hwang, J.
Iacono, M.
Iglesias, A.
Ingram, D.
Jakulin, A.
Janciak, I.
Janecek, J.
Janglova, D.
Janicki, A.
Jin, H.
Jost, G.
Juhola, A.
Kacsuk, P.
Kalousis, A.
Kalyanaraman, A.
Kang, M.G.
Karagiorgos, G.
Karaivanova, A.
Karl, W.
Karypis, G.
Katarzyniak, R.
Kelley, T.
Kelly, W.
Kennedy, E.
Kereku, E.
Kergommeaux, J.C. De
Kim, B.
Kim, C.H.
Kim, D.S.
Kim, D.Y.
Kim, M.
Kim, M.J.
Kim, T.W.
Kitowski, J.
Klein, C.
Ko, P.
Kokoszka, P.
Kolingerova, I.
Kommineni, J.
Korczak, J.J.
Korkhov, V.
Kou, G.
Kouniakis, C.
Kranzlmüller, D.
Krzhizhianovskaya, V.V.
Kuo, T.W.
Kurka, G.
Kurniawan, D.
Kurzyniec, D.
Laclavik, M.
Laforenza, D.
Lagan, A.
Lagana, A.
Lamehamedi, H.
Larrabeiti, D.
Latt, J.
Lau, F.
Lee, H.G.
Lee, M.
Lee, S.
Lee, S.S.
Lee, S.Y.
Lefevre, L.
Leone, P.
Lesyng, B.
Leszczynski, J.
Leymann, F.
Li, T.
Lindner, P.
Logan, B.
Lopes, G.P.
Lorencz, R.
Low, M.Y.H.
Ludwig, T.
Luethi, J.
Lukac, R.
Luksch, P.
Luque, E.
Mairandres, M.
Malawski, M.
Malony, A.
Malyshkin, V.E.
Maniatty, W.A.
Marconi, S.
Mareev, V.
Margalef, T.
Marrone, S.
Martino, B. Di
Marzolla, M.
Mascagni, M.
Mayer, M.
Medeiros, P.
Meer, H. De
Meyer, N.
Miller, B.
Miyaji, C.
Modave, F.
Mohr, B.
Monterde, J.
Moore, S.
Moreno, E.
Moscato, F.
Mourelle, L.M.
Mueller, M.S.
Mun, Y.
Na, W.S.
Nagel, W.E.
Nanni, M.
Narayanan, M.
Nasri, A.
Nau, B.
Nedjah, N.
Nedoma, J.
Negoita, C.
Neumann, L.
Nguyen, G.T.
Nguyen, N.T.
Norman, G.
Olariu, S.
Orlando, S.
Orley, S.
Otero, C.
Owen, J.
Palus, H.
Paprzycki, M.
Park, N.J.
Patten, C.
Peachey, T.C.

Peluso, R.
Peng, Y.
Perales, F.
Perrott, R.
Petit, F.
Petit, G.H.
Pfluger, P.
Philippe, L.
Platen, E.
Plemenos, D.
Pllana, S.
Polak, M.
Polak, N.
Politi, T.
Pooley, D.
Popov, E.V.
Puppin, D.
Qut, P.R.
Rachev, S.
Rajko, S.
Rak, M.
Ramaroson, R.
Ras, I.
Rathmayer, S.
Raz, D.
Recio, T.
Reichel, L.
Renaut, R.
Rendell, A.
Richta, K.
Robert, Y.
Rodgers, G.
Rodionov, A.S.
Roe, P.
Ronsse, M.
Ruder, K.S.
Ruede, U.
Rycerz, K.
Sanchez-Reyes, J.
Sarfraz, M.
Sbert, M.
Scarpa, M.
Schabanel, N.
Scharf, E.
Scharinger, J.
Schaubschlaeger, C.
Schmidt, A.
Scholz, S.B.
Schreiber, A.
Seal, S.K.
Seinstra, F.J.
Seron, F.
Serrat, J.
Shamonin, D.P.
Sheldon, F.
Shen, H.
Shende, S.
Shentu, Z.
Shi, Y.
Shin, H.Y.
Shires, D.
Shoshmina, I.
Shrikhande, N.
Silvestri, C.
Silvestri, F.
Simeoni, M.
Simo, B.
Simonov, N.
Siu, P.
Slizik, P.
Slominski, L.
Sloot, P.M.A.
Slota, R.
Smetek, M.
Smith, G.
Smolka, B.
Sneeuw, N.
Snoek, C.
Sobaniec, C.
Sobecki, J.
Sofroniou, M.
Sole, R.
Soofi, M.
Sosnov, A.
Sourin, A.
Spaletta, G.
Spiegl, E.
Stapor, K.
Stuer, G.
Suarez Rivero, J.P.
Sunderam, V.
Suzuki, H.
Szatzschneider, W.
Szczepanski, M.
Szirmay-Kalos, L.
Szymanski, B.
Tadeusiewicz, R.
Tadic, B.
Talia, D.
Tan, G.
Taylor, S.J.E.
Teixeira, J.C.
Telelis, O.A.
Teo, Y.M
Teresco, J.
Teyssiere, G.
Thalmann, D.
Theodoropoulos, G.
Theoharis, T.
Thurner, S.
Tirado-Ramos, A.
Tisserand, A.
Toda, K.
Tonellotto, N.
Torelli, L.
Torenvliet, L.
Tran, V.D.
Truong, H.L.
Tsang, K.
Tse, K.L.
Tvrdik, P.
Tzevelekas, L.
Uthayopas, P.
Valencia, P.
Vassilakis, C.
Vaughan, F.
Vazquez, P.P.
Venticinque, S.
Vigo-Aguiar, J.
Vivien, F.
Volkert, J.
Wada, K.
Walter, M.
Wasniewski, J.
Wasserbauer, A.

Watson, G.
Wawrzyniak, D.
Weglarz, J.
Weidendorfer, J.
Weispfenning, W.
Wendelborn, A.L.
Weron, R.
Wismüller, R.
Wojciechowski, K.
Wolf, F.
Worring, M.
Wyrzykowski, R.
Xiao, Y.
Xu, J.
Xue, Y.
Yahyapour, R.
Yan, N.
Yang, K.
Yener, B.
Yoo, S.M.
Yu, J.H.
Yu, Z.C.H.
Zara, J.
Zatevakhin, M.A.
Zhang, J.W.
Zhang, N.X.L.
Zhang, X.
Zhao, L.
Zhmakin, A.I.
Zhu, W.Z.
Zieliński, K.
Zlatev, Z.
Zomaya, A.
Zudilova, E.V.

Workshops Organizers

Programming Grids and Metasystems

V. Sunderam (Emory University, USA)
D. Kurzyniec (Emory University, USA)
V. Getov (University of Westminster, UK)
M. Malawski (Institute of Computer Science and CYFRONET AGH, Poland)

Active and Programmable Grids Architectures and Components

C. Anglano (Università del Piemonte Orientale, Italy)
F. Baumgartner (University of Bern, Switzerland)
G. Carle (Tubingen University, Germany)
X. Cheng (Institute of Computing Technology, Chinese Academy of Science, ROC)
K. Chen (Institut Galilée, Université Paris 13, France)
S. Denazis (Hitachi Europe, France)
B. Dhoedt (University of Gent, Belgium)
W. Donnelly (Waterford Institute of Technology, Ireland)
A. Galis (University College London, UK)
A. Gavras (Eurescom, Germany)
F. Gagliardi (CERN, Switzerland)
Y. Gourhant (France Telecom, France)
M. Gilbert (European Microsoft Innovation Center, Microsoft Corporation, Germany)
A. Juhola (VTT, Finland)
C. Klein (Siemens, Germany)
D. Larrabeiti (University Carlos III, Spain)
L. Lefevre (INRIA, France)
F. Leymann (IBM, Germany)
H. de Meer (University of Passau, Germany)
G. H. Petit (Alcatel, Belgium)

J. Serrat (Universitat Politècnica de Catalunya, Spain)
E. Scharf (QMUL, UK)
K. Skala (Ruder Boskoviç Institute, Croatia)
N. Shrikhande (European Microsoft Innovation Center, Microsoft Corporation, Germany)
M. Solarski (FhG FOKUS, Germany)
D. Raz (Technion Institute of Technology, Israel)
K. Zieliński (AGH University of Science and Technology, Poland)
R. Yahyapour (University Dortmund, Germany)
K. Yang (University of Essex, UK)

Next Generation Computing

E.-N. John Huh (Seoul Women's University, Korea)

Practical Aspects of High-Level Parallel Programming (PAPP 2004)

F. Loulergue (Laboratory of Algorithms, Complexity and Logic, University of Paris Val de Marne, France)

Parallel Input/Output Management Techniques (PIOMT 2004)

J. H. Abawajy (Carleton University, School of Computer Science, Canada)

OpenMP for Large Scale Applications

B. Chapman (University of Houston, USA)

Tools for Program Development and Analysis in Computational Science

D. Kranzlmüller (Johannes Kepler University Linz, Austria)
R. Wismüller (TU München, Germany)
A. Bode (Technische Universität München, Germany)
J. Volkert (Johannes Kepler University Linz, Austria)

Modern Technologies for Web-Based Adaptive Systems

N. Thanh Nguyen (Wrocław University of Technology, Poland)
J. Sobecki (Wrocław University of Technology, Poland)

Agent Day 2004 – Intelligent Agents in Computing Systems

E. Nawarecki (AGH University of Science and Technology, Poland)
K. Cetnarowicz (AGH University of Science and Technology, Poland)
G. Dobrowolski (AGH University of Science and Technology, Poland)
R. Schaefer (Jagiellonian University, Poland)
S. Ambroszkiewicz (Polish Academy of Sciences, Warsaw, Poland)
A. Koukam (Université de Belfort-Montbeliard, France)
V. Srovnal (VSB Technical University of Ostrava, Czech Republic)
C. Cotta (Universidad de Málaga, Spain)
S. Raczynski (Universidad Panamericana, Mexico)

Dynamic Data Driven Application Systems

F. Darema (NSF/CISE, USA)

HLA-Based Distributed Simulation on the Grid

S. J. Turner (Nanyang Technological University, Singapore)

Interactive Visualisation and Interaction Technologies

E. Zudilova (University of Amsterdam, The Netherlands)
T. Adriaansen (CSIRO, ICT Centre, Australia)

Computational Modeling of Transport on Networks

B. Tadic (Jozef Stefan Institute, Slovenia)
S. Thurner (Universität Wien, Austria)

Modeling and Simulation in Supercomputing and Telecommunications

Y. Mun (Soongsil University, Korea)

QoS Routing

H. Choo (Sungkyunkwan University, Korea)

Evolvable Hardware

N. Nedjah (State University of Rio de Janeiro, Brazil)
L. de Macedo Mourelle (State University of Rio de Janeiro, Brazil)

Advanced Methods of Digital Image Processing

B. Smolka (Silesian University of Technology, Laboratory of Multimedia Communication, Poland)

Computer Graphics and Geometric Modelling (CGGM 2004)

A. Iglesias Prieto (University of Cantabria, Spain)

Computer Algebra Systems and Applications (CASA 2004)

A. Iglesias Prieto (University of Cantabria, Spain)
A. Galvez (University of Cantabria, Spain)

New Numerical Methods for DEs: Applications to Linear Algebra, Control and Engineering

N. Del Buono (University of Bari, Italy)
L. Lopez (University of Bari, Italy)

Parallel Monte Carlo Algorithms for Diverse Applications in a Distributed Setting

V. N. Alexandrov (University of Reading, UK)
A. Karaivanova (Bulgarian Academy of Sciences, Bulgaria)
I. Dimov (Bulgarian Academy of Sciences, Bulgaria)

Modelling and Simulation of Multi-physics Multi-scale Systems

V. Krzhizhanovskaya (University of Amsterdam, The Netherlands)
B. Chopard (University of Geneva, CUI, Switzerland)
Y. Gorbachev (St. Petersburg State Polytechnical University, Russia)

Gene, Genome and Population Evolution

S. Cebrat (University of Wrocław, Poland)
D. Stauffer (Cologne University, Germany)
A. Maksymowicz (AGH University of Science and Technology, Poland)

Computational Methods in Finance and Insurance

A. Janicki (University of Wrocław, Poland)
J.J. Korczak (University Louis Pasteur, Strasbourg, France)

Computational Economics and Finance

X. Deng (City University of Hong Kong, Hong Kong)
S. Wang (Chinese Academy of Sciences, ROC)
Y. Shi (University of Nebraska at Omaha, USA)

GeoComputation

Y. Xue (Chinese Academy of Sciences, ROC)
C. Yarotsos (University of Athens, Greece)

Simulation and Modeling of 3D Integrated Circuits

I. Balk (R3Logic Inc., USA)

Computational Modeling and Simulation on Biomechanical Engineering

Y.H. Kim (Kyung Hee University, Korea)

Information Technologies Enhancing Health Care Delivery

M. Duplaga (Jagiellonian University Medical College, Poland)
D. Ingram (University College London, UK)
K. Zieliński (AGH University of Science and Technology, Poland)

Computing in Science and Engineering Academic Programs

D. Donnelly (Siena College, USA)

Sponsoring Institutions

Hewlett-Packard
Intel
SGI
ATM
SUN Microsystems
IBM
Polish Airlines LOT
ACC CYFRONET AGH
Institute of Computer Science AGH
Polish Ministry of Scientific Research and Information Technology
Springer-Verlag

This page intentionally left blank

Table of Contents – Part II

Track on Numerical Algorithms

Hierarchical Matrix-Matrix Multiplication Based on Multiprocessor Tasks 1
S. Hunold, T. Rauber, G. Rünger

Improving Geographical Locality of Data for Shared Memory Implementations of PDE Solvers 9
H. Löf, M. Nordén, S. Holmgren

Cache Oblivious Matrix Transposition: Simulation and Experiment 17
D. Tsifakis, A.P. Rendell, P.E. Strazdins

An Intelligent Hybrid Algorithm for Solving Non-linear Polynomial Systems 26
J. Xue, Y. Li, Y. Feng, L. Yang, Z. Liu

A Jacobi–Davidson Method for Nonlinear Eigenproblems 34
H. Voss

Numerical Continuation of Branch Points of Limit Cycles in MATCONT 42
A. Dhooge, W. Govaerts, Y.A. Kuznetsov

Online Algorithm for Time Series Prediction Based on Support Vector Machine Philosophy 50
J.M. Górriz, C.G. Puntonet, M. Salmerón

Improved A-P Iterative Algorithm in Spline Subspaces 58
J. Xian, S.P. Luo, W. Lin

Solving Differential Equations in Developmental Models of Multicellular Structures Expressed Using L-systems 65
P. Federl, P. Prusinkiewicz

On a Family of A-stable Collocation Methods with High Derivatives 73
G. Y. Kulikov, A.I. Merkulov, E.Y. Khrustaleva

Local Sampling Problems 81
S.-Y. Yang, W. Lin

Recent Advances in Semi-Lagrangian Modelling of Flow through the Strait of Gibraltar 89
M. Seaïd, M. El-Amrani, A. Machmoum

Efficiency Study of the "Black-Box" Component Decomposition Preconditioning for Discrete Stress Analysis Problems 97
M.D. Mihajlović, S. Mijalković

Direct Solver Based on FFT and SEL for Diffraction Problems with Distribution 105
H. Koshigoe

Non-negative Matrix Factorization for Filtering Chinese Document 113
J. Lu, B. Xu, J. Jiang, D. Kang

On Highly Secure and Available Data Storage Systems 121
S.J. Choi, H.Y. Youn, H.S. Lee

Track on Finite Element Method

A Numerical Adaptive Algorithm for the Obstacle Problem 130
F.A. Pérez, J.M. Cascón, L. Ferragut

Finite Element Model of Fracture Formation on Growing Surfaces 138
P. Federl, P. Prusinkiewicz

An Adaptive, 3-Dimensional, Hexahedral Finite Element Implementation for Distributed Memory 146
J. Hippold, A. Meyer, G. Rünger

A Modular Design for Parallel Adaptive Finite Element Computational Kernels 155
K. Banaś

Load Balancing Issues for a Multiple Front Method 163
C. Denis, J.P. Boufflet, P. Breitkopf, M. Vayssade, B. Glut

Multiresolutional Techniques in Finite Element Method Solution of Eigenvalue Problem 171
M. Kamiński

Track on Neural Networks

Self-Organizing Multi-layer Fuzzy Polynomial Neural Networks Based on Genetic Optimization 179
S.-K. Oh, W. Pedrycz, H.-K. Kim, J.-B. Lee

Information Granulation-Based Multi-layer Hybrid Fuzzy Neural Networks: Analysis and Design 188
B.-J. Park, S.-K. Oh, W. Pedrycz, T.-C. Ahn

Efficient Learning of Contextual Mappings by Context-Dependent Neural Nets 196
P. Ciskowski

An Unsupervised Neural Model to Analyse Thermal Properties of Construction Materials 204
E. Corchado, P. Burgos, M. Rodríguez, V. Tricio

Intrusion Detection Based on Feature Transform Using Neural Network 212
W. Kim, S.-C. Oh, K. Yoon

Track on Applications

Accelerating Wildland Fire Prediction on Cluster Systems 220
B. Abdalhaq, A. Cortés, T. Margalef, E. Luque

High Precision Simulation of Near Earth Satellite Orbits for SAR-Applications 228
M. Kalkuhl, K. Nöh, O. Loffeld, W. Wiechert

Hybrid Approach to Reliability and Functional Analysis of Discrete Transport System 236
T. Walkowiak, J. Mazurkiewicz

Mathematical Model of Gas Transport in Anisotropic Porous Electrode of the PEM Fuel Cell 244
E. Kurgan, P. Schmidt

Numerical Simulation of Anisotropic Shielding of Weak Magnetic Fields 252
E. Kurgan

Functionalization of Single-Wall Carbon Nanotubes: An Assessment of Computational Methods 260
B. Akdim, T. Kar, X. Duan, R. Pachter

Improved Sampling for Biological Molecules Using Shadow Hybrid Monte Carlo 268
S.S. Hampton, J.A. Izaguirre

A New Monte Carlo Approach for Conservation Laws and Relaxation Systems 276
L. Pareschi, M. Seaïd

A Parallel Implementation of Gillespie's Direct Method 284
A.M. Ridwan, A. Krishnan, P. Dhar

Simulation of Deformable Objects Using Sliding Mode Control with Application to Cloth Animation 292
F. Rum, B. W. Gordon

Constraint-Based Contact Analysis between Deformable Objects 300
M. Hong, M.-H. Choi, C. Lee

Prediction of Binding Sites in Protein-Nucleic Acid Complexes 309
N. Han, K. Han

Prediction of Protein Functions Using Protein Interaction Data 317
H. Jung, K. Han

Interactions of Magainin-2 Amide with Membrane Lipids 325
K. Murzyn, T. Róg, M. Pasenkiewicz-Gierula

Dynamics of Granular Heaplets: A Phenomenological Model 332
Y.K. Goh, R.L. Jacobs

Modelling of Shear Zones in Granular Materials within Hypoplasticity 340
J. Tejchman

Effective Algorithm for Detection of a Collision between Spherical Particles 348
J.S. Leszczynski, M. Ciesielski

Vorticity Particle Method for Simulation of 3D Flow 356
H. Kudela, P. Regucki

Crack Analysis in Single Plate Stressing of Particle Compounds 364
M. Khanal, W. Schubert, J. Tomas

A Uniform and Reduced Mathematical Model for Sucker Rod Pumping 372
L. Liu, C. Tong, J. Wang, R. Liu

Distributed Computation of Optical Flow 380
A.G. Dopico, M.V. Correia, J.A. Santos, L.M. Nunes

Analytical Test on Effectiveness of MCDF Operations 388
J. Kong, B. Zhang, W. Guo

An Efficient Perspective Projection Using VolumePro™ 396
S. Lim, B.-S. Shin

Reconstruction of 3D Curvilinear Wireframe Model from 2D Orthographic Views 404
A. Zhang, Y. Xue, X. Sun, Y. Hu, Y. Luo, Y. Wang, S. Zhong, J. Wang, J. Tang, G. Cai

Surface Curvature Estimation for Edge Spinning Algorithm 412
M. Cermak, V. Skala

Visualization of Very Large Oceanography Time-Varying Volume Datasets 419
S. Park, C. Bajaj, I. Ihm

Sphere-Spin-Image: A Viewpoint-Invariant Surface Representation for 3D Face Recognition 427
Y. Wang, G. Pan, Z. Wu, S. Han

Design and Implementation of Integrated Assembly Object Model for Intelligent Virtual Assembly Planning 435
J. Fan, Y. Ye, J.-M. Cai

Adaptive Model Based Parameter Estimation, Based on Sparse Data and Frequency Derivatives 443
D. Deschrijver, T. Dhaene, J. Broeckhove

Towards Efficient Parallel Image Processing on Cluster Grids Using GIMP 451
P. Czarnul, A. Ciereszko, M. Frączak

Benchmarking Parallel Three Dimensional FFT Kernels with ZENTURIO 459
R. Prodan, A. Bonelli, A. Adelmann, T. Fahringer, C. Überhuber

The Proof and Illustration of the Central Limit Theorem by Brownian Numerical Experiments in Real Time within the Java Applet 467
M. Gall, R. Kutner, W. Wesela

An Extended Coherence Protocol for Recoverable DSM Systems with Causal Consistency 475
J. Brzezinski, M. Szychowiak

2D and 3D Representations of Solution Spaces for CO Problems 483
E. Nowicki, C. Smutnicki

Effective Detector Set Generation and Evolution for Artificial Immune System 491
C. Kim, W. Kim, M. Hong

Artificial Immune System against Viral Attack 499
H. Lee, W. Kim, M. Hong

Proposal of the Programming Rules for VHDL Designs 507
J. Borgosz, B. Cyganek

A Weight Adaptation Method for Fuzzy Cognitive Maps
to a Process Control Problem 515
E. Papageorgiou, P. Groumpos

A Method Based on Fuzzy Logic Technique for Smoothing in 2D 523
A. Çinar

Proportional-Integral-Derivative Controllers Tuning for Unstable
and Integral Processes Using Genetic Algorithms 532
M.A. Paz-Ramos, J. Torres-Jimenez, E. Quintero-Marmol-Marquez

Enabling Systems Biology: A Scientific Problem-Solving Environment . . . 540
M. Singhal, E.G. Stephan, K.R. Klicker, L.L. Trease, G. Chin Jr., D.K. Gracio, D.A. Payne

Poster Papers

Depth Recovery with an Area Based Version of the Stereo Matching
Method with Scale-Space Tensor Representation
of Local Neighborhoods 548
B. Cyganek

Symbolic Calculation for Frölicher-Nijenhuis
$\mathbb{R}$-Algebra for Exploring in Electromagnetic Field Theory 552
J. de Cruz Guzmán, Z. Oziewicz

Spherical Orthogonal Polynomials and Symbolic-Numeric Gaussian
Cubature Formulas 557
A. Cuyt, B. Benouahmane, B. Verdonk

The Berlekamp-Massey Algorithm. A Sight from Theory
of Pade Approximants and Orthogonal Polynomials 561
S.B. Gashkov, I.B. Gashkov

An Advanced Version of the Local-Global Step Size Control
for Runge-Kutta Methods Applied to Index 1
Differential-Algebraic Systems 565
G. Y. Kulikov

INTEGRATOR: A Computational Tool to Solve Ordinary Differential
Equations with Global Error Control 570
G. Y. Kulikov, S.K. Shindin

Reconstruction of Signal from Samples of Its Integral in Spline Subspaces 574
J. Xian, Y. Li, W. Lin

The Vectorized and Parallelized Solving of Markovian Models for Optical Networks 578
B. Bylina, J. Bylina

A Parallel Splitting up Algorithm for the Determination of an Unknown Coefficient in Multi Dimensional Parabolic Problem 582
D.S. Daoud, D. Subasi

A-Posteriori Error Analysis of a Mixed Method for Linear Parabolic Problem 586
M.I. Asensio, J.M. Cascón, L. Ferragut

Analysis of Parallel Numerical Libraries to Solve the 3D Electron Continuity Equation 590
N. Seoane, A.J. García-Loureiro

Parallel Solution of Cascaded ODE Systems Applied to ^{13}C-Labeling Experiments 594
K. Nöh, W. Wiechert

A *k*-way Graph Partitioning Algorithm Based on Clustering by Eigenvector 598
T.-Y. Choe, C.-I. Park

Network of Networks 602
J. de Cruz Guzmán, Z. Oziewicz

MSL: An Efficient Adaptive In-Place Radix Sort Algorithm 606
F. El-Aker, A. Al-Badarneh

Parallel Chip Firing Game Associated with *n-cube* Edges Orientations 610
R. Ndoundam, C. Tadonki, M. Tchuente

A Fast Multifrontal Solver for Non-linear Multi-physics Problems 614
A. Bertoldo, M. Bianco, G. Pucci

Modelling of Interaction between Surface Waves and Mud Layer 618
L. Balas

Computational Modelling of Pulsating Biological Flow 622
X.S. Yang, R.W. Lewis, H. Zhang

Effect of Heterogeneity on Formation of Shear Zones in Granular Bodies 626
J. Tejchman

Effect of Structural Disorder on the Electronic Density of States in One-Dimensional Chain of Atoms 630
M. Wołoszyn, B.J. Spisak

The Estimation of the Mathematical Exactness of System Dynamics Method on the Base of Some Economic System 634
E. Kasperska, D. Słota

Size of the Stable Population in the Penna Bit-String Model of Biological Aging 638
K. Malarz, M. Sitarz, P. Gronek, A. Dydejczyk

Velocity Field Modelling for Pollutant Plume Using 3-D Adaptive Finite Element Method 642
G. Montero, R. Montenegro, J.M. Escobar, E. Rodríguez, J.M. González-Yuste

Organization of the Mesh Structure 646
T. Jurczyk, B. Głut

Kernel Maximum Likelihood Hebbian Learning 650
J. Koetsier, E. Corchado, D. MacDonald, J. Corchado, C. Fyfe

Discovery of Chemical Transformations with the Use of Machine Learning 654
G. Fic, G. Nowak

Extraction of Document Descriptive Terms with a Linguistic-Based Machine Learning Approach 658
J. Fernández, E. Montañés, I. Díaz, J. Ranilla, E.F. Combarro

Application of Brain Emotional Learning Based Intelligent Controller (BELBIC) to Active Queue Management 662
M. Jalili-Kharaajoo

A Hybrid Algorithm Based on PSO and SA and Its Application for Two-Dimensional Non-guillotine Cutting Stock Problem 666
J.Q. Jiang, Y.C. Liang, X.H. Shi, H.P. Lee

Evolving TSP Heuristics Using Multi Expression Programming 670
M. Oltean, D. Dumitrescu

Improving the Performance of Evolutionary Algorithms for the Multiobjective 0/1 Knapsack Problem Using ε-Dominance 674
C. Groşan, M. Oltean

Genetic Evolution Approach for Target Movement Prediction 678
S. Baik, J. Bala, A. Hadjarian, P. Pachowicz

Adaptive Transfer Functions
in Radial Basis Function (RBF) Networks 682
G.A. Hoffmann

Disturbance Rejection Control of Thermal Power Plant
Using Immune Algorithm 687
D.H. Kim, J.H. Cho

The Design Methodology of Fuzzy Controller Based on Information
Granulation (IG)-Based Optimization Approach 691
S.-K. Oh, S.-B. Roh, D.-Y. Lee

PID Controller Tuning of a Boiler Control System
Using Immune Algorithm Typed Neural Network 695
D.H. Kim

A Framework to Investigate and Evaluate Genetic Clustering
Algorithms for Automatic Modularization of Software Systems 699
S. Parsa, O. Bushehrian

An Artificial Immune Algorithms Apply to Pre-processing Signals 703
M. Święcicki, W. Wajs, P. Wais

Identification and Control Using Direction Basis
Function Neural Network 708
M. Jalili-Kharaajoo

A New Dynamic Structure Neural Network for Control
of Nonlinear Systems 713
M. Jalili-Kharaajoo

Proposing a New Learning Algorithm to Improve Fault Tolerance
of Neural Networks 717
M. Jalili-Kharaajoo

Nonlinear Parametric Model Identification and Model Based Control
of *S. cerevisiae* Production 722
B. Akay

The Notion of Community in United States
Computational Science Education Initiatives 726
M.E. Searcy, J.T. Richie

Author Index 731

This page intentionally left blank

Table of Contents – Part I

Track on Parallel and Distributed Computing

Optimization of Collective Reduction Operations 1
R. Rabenseifner

Predicting MPI Buffer Addresses 10
F. Freitag, M. Farreras, T. Cortes, J. Labarta

An Efficient Load-Sharing and Fault-Tolerance Algorithm in Internet-Based Clustering Systems 18
I.-B. Choi, J.-D. Lee

Dynamic Parallel Job Scheduling in Multi-cluster Computing Systems 27
J.H. Abawajy

Hunting for Bindings in Distributed Object-Oriented Systems 35
M. Sławińska

Design and Implementation of the Cooperative Cache for PVFS 43
I.-C. Hwang, H. Kim, H. Jung, D.-H. Kim, H. Ghim, S.-R. Maeng, J.-W. Cho

Track on Grid Computing

Towards OGSA Compatibility in Alternative Metacomputing Frameworks 51
G. Stuer, V. Sunderam, J. Broeckhove

DartGrid: Semantic-Based Database Grid 59
Z. Wu, H. Chen, Changhuang, G. Zheng, J. Xu

A 3-tier Grid Architecture and Interactive Applications Framework for Community Grids 67
O. Ardaiz, K. Sanjeevan, R. Sanguesa

Incorporation of Middleware and Grid Technologies to Enhance Usability in Computational Chemistry Applications 75
J.P. Greenberg, S. Mock, M. Katz, G. Bruno, F. Sacerdoti, P. Papadopoulos, K.K. Baldridge

An Open Grid Service Environment for Large-Scale Computational Finance Modeling Systems 83
C. Wiesinger, D. Giczi, R. Hochreiter

The Migrating Desktop as a GUI Framework for the "Applications on Demand" Concept 91
M. Kupczyk, R. Lichwała, N. Meyer, B. Palak, M. Płóciennik, M. Stroiński, P. Wolniewicz

Interactive Visualization for the UNICORE Grid Environment 99
P. Bała, K. Benedyczak, A. Nowiński, K.S. Nowiński, J. Wypychowski

Efficiency of the GSI Secured Network Transmission 107
B. Baliś, M. Bubak, W. Rząsa, T. Szepieniec

An Idle Compute Cycle Prediction Service for Computational Grids 116
S. Hwang, E.-J. Im, K. Jeong, H. Park

Infrastructure for Grid-Based Virtual Organizations 124
L. Hluchy, O. Habala, V.D. Tran, B. Simo, J. Astalos, M. Dobrucky

Air Pollution Modeling in the CrossGrid Project 132
J.C. Mouriño, M.J. Martín, P. González, R. Doallo

The Genetic Algorithms Population Pluglet for the H2O Metacomputing System 140
T. Ampuła, D. Kurzyniec, V. Sunderam, H. Witek

Applying Grid Computing to the Parameter Sweep of a Group Difference Pseudopotential 148
W. Sudholt, K.K. Baldridge, D. Abramson, C. Enticott, S. Garic

A Grid Enabled Parallel Hybrid Genetic Algorithm for SPN 156
G.L. Presti, G.L. Re, P. Storniolo, A. Urso

An Atmospheric Sciences Workflow and Its Implementation with Web Services 164
D. Abramson, J. Kommineni, J.L. McGregor, J. Katzfey

Twins: 2-hop Structured Overlay with High Scalability 174
J. Hu, H. Dong, W. Zheng, D. Wang, M. Li

Dispatching Mechanism of an Agent-Based Distributed Event System 184
O.K. Sahingoz, N. Erdogan

An Adaptive Communication Mechanism for Highly Mobile Agents 192
J. Ahn

Track on Models and Algorithms

Knapsack Model and Algorithm for HW/SW Partitioning Problem 200
A. Ray, W. Jigang, S. Thambipillai

A Simulated Annealing Algorithm for the Circles Packing Problem 206
D. Zhang, W. Huang

Parallel Genetic Algorithm for Graph Coloring Problem 215
Z. Kokosiński, M. Kołodziej, K. Kwarciany

Characterization of Efficiently Parallel Solvable Problems on a Class of Decomposable Graphs 223
S.-Y. Hsieh

The Computational Complexity of Orientation Search in Cryo-Electron Microscopy 231
T. Mielikäinen, J. Ravantti, E. Ukkonen

Track on Data Mining and Data Bases

Advanced High Performance Algorithms for Data Processing 239
A.V. Bogdanov, A.V. Boukhanovsky

Ontology-Based Partitioning of Data Steam for Web Mining: A Case Study of Web Logs 247
J.J. Jung

Single Trial Discrimination between Right and Left Hand Movement-Related EEG Activity 255
S. Cho, J.A. Kim, D.-U. Hwang, S.K. Han

WINGS: A Parallel Indexer for Web Contents 263
F. Silvestri, S. Orlando, R. Perego

A Database Server for Predicting Protein-Protein Interactions 271
K. Han, B. Park

PairAnalyzer: Extracting and Visualizing RNA Structure Elements Formed by Base Pairing 279
D. Lim, K. Han

A Parallel Crawling Schema Using Dynamic Partition 287
S. Dong, X. Lu, L. Zhang

Hybrid Collaborative Filtering and Content-Based Filtering for Improved Recommender System 295
K.-Y. Jung, D.-H. Park, J.-H. Lee

Object-Oriented Database Mining: Use of Object Oriented Concepts for Improving Data Classification Technique 303
K. Waiyamai, C. Songsiri, T. Rakthanmanon

Data-Mining Based Skin-Color Modeling Using the ECL Skin-Color Images Database 310
M. Hammami, D. Tsishkou, L. Chen

Maximum Likelihood Based Quantum Set Separation 318
S. Imre, F. Balázs

Chunking-Coordinated-Synthetic Approaches to Large-Scale Kernel Machines 326
F.J. González-Castaño, R.R. Meyer

Computational Identification of -1 Frameshift Signals 334
S. Moon, Y. Byun, K. Han

Track on Networking

Mobility Management Scheme for Reducing Location Traffic Cost in Mobile Networks 342
B.-M. Min, J.-G. Jee, H.S. Oh

Performance Analysis of Active Queue Management Schemes for IP Network 349
J. Koo, S. Ahn, J. Chung

A Real-Time Total Order Multicast Protocol 357
K. Erciyes, A. Şahan

A Rule-Based Intrusion Alert Correlation System for Integrated Security Management 365
S.-H. Lee, H.-H. Lee, B.-N. Noh

Stable Neighbor Based Adaptive Replica Allocation in Mobile Ad Hoc Networks 373
Z. Jing, S. Jinshu, Y. Kan, W. Yijie

Mobile-Based Synchronization Model for Presentation of Multimedia Objects 381
K.-W. Lee, H.-S. Cho, K.-H. Lee

Synchronization Scheme of Multimedia Streams
in Mobile Handoff Control 389
G.-S. Lee

Poster Papers

The Development of a Language for Specifying Structure
of a Distributed and Parallel Application 397
R. Dew, P. Horan, A. Goscinski

Communication Primitives for Minimally Synchronous Parallel ML 401
F. Loulergue

Dependence Analysis of Concurrent Programs
Based on Reachability Graph and Its Applications 405
X. Qi, B. Xu

Applying Loop Tiling and Unrolling to a Sparse Kernel Code 409
E. Herruzo, G. Bandera, O. Plata

A Combined Method for Texture Analysis and Its Application 413
Y. Zhang, R. Wang

Reliability of Cluster System with a Lot of Software Instances 417
M. Szymczyk, P. Szymczyk

A Structural Complexity Measure for UML Class Diagrams 421
B. Xu, D. Kang, J. Lu

Parallelizing Flood Models with MPI:
Approaches and Experiences 425
V.D. Tran, L. Hluchy

Using Parallelism in Experimenting and Fine Tuning of Parameters
for Metaheuristics 429
M. Blesa, F. Xhafa

DEVMA: Developing Virtual Environments with Awareness Models 433
P. Herrero, A. de Antonio

A Two-Leveled Mobile Agent System for E-commerce
with Constraint-Based Filtering 437
O.K. Sahingoz, N. Erdogan

ABSDM: Agent Based Service Discovery Mechanism
in Internet 441
S. Li, C. Xu, Z. Wu, Y. Pan, X. Li

Meta Scheduling Framework for Workflow Service on the Grids 445
S. Hwang, J. Choi, H. Park

Resources Virtualization in Fault-Tolerance and Migration Issues 449
G. Jankowski, R. Mikolajczak, R. Januszewski, N. Meyer, M. Stroiński

On the Availability of Information Dispersal Scheme for Distributed Storage Systems 453
S.K. Song, H.Y. Youn, G.-L. Park, K.S. Tae

Virtual Storage System for the Grid Environment 458
D. Nikolow, R. Słota, J. Kitowski, Ł. Skitał

Performance Measurement Model in the G-PM Tool 462
R. Wismüller, M. Bubak, W. Funika, T. Arodź, M. Kurdziel

Paramedir: A Tool for Programmable Performance Analysis 466
G. Jost, J. Labarta, J. Gimenez

Semantic Browser: an Intelligent Client for Dart-Grid 470
Y. Mao, Z. Wu, H. Chen

On Identity-Based Cryptography and GRID Computing 474
H.W. Lim, M.J.B. Robshaw

The Cambridge CFD Grid Portal for Large-Scale Distributed CFD Applications 478
X. Yang, M. Hayes, K. Jenkins, S. Cant

Grid Computing Based Simulations of the Electrical Activity of the Heart 482
J.M. Alonso, V. Hernández, G. Moltó

Artificial Neural Networks and the Grid 486
E. Schikuta, T. Weishäupl

Towards a Grid-Aware Computer Algebra System 490
D. Petcu, D. Dubu, M. Paprzycki

Grid Computing and Component-Based Software Engineering in Computer Supported Collaborative Learning 495
M.L. Bote-Lorenzo, J.I. Asensio-Pérez, G. Vega-Gorgojo, L.M. Vaquero-González, E. Gómez-Sánchez, Y.A. Dimitriadis

An NAT-Based Communication Relay Scheme for Private-IP-Enabled MPI over Grid Environments 499
S. Choi, K. Park, S. Han, S. Park, O. Kwon, Y. Kim, H. Park

A Knowledge Fusion Framework in the Grid Environment 503
J. Gou, J. Yang, H. Qi

A Research of Grid Manufacturing and Its Application in Custom Artificial Joint 507
L. Chen, H. Deng, Q. Deng, Z. Wu

Toward a Virtual Grid Service of High Availability 511
X. Zhi, W. Tong

The Measurement Architecture of the Virtual Traffic Laboratory 515
A. Visser, J. Zoetebier, H. Yakali, B. Hertzberger

Adaptive QoS Framework for Multiview 3D Streaming 519
J.R. Kim, Y. Won, Y. Iwadate

CORBA-Based Open Platform for Processes Monitoring. An Application to a Complex Electromechanical Process 523
K. Cantillo, R.E. Haber, J.E. Jiménez, Á. Alique, R. Galán

An Approach to Web-Oriented Discrete Event Simulation Modeling 527
E. Ochmańska

Query Execution Algorithm in Web Environment with Limited Availability of Statistics 532
J. Jezierski, T. Morzy

Using Adaptive Priority Controls for Service Differentiation in QoS-Enabled Web Servers 537
M.M. Teixeira, M.J. Santana, R.H. Carlucci Santana

On the Evaluation of x86 Web Servers Using Simics: Limitations and Trade-Offs 541
F.J. Villa, M.E. Acacio, J.M. García

MADEW: Modelling a Constraint Awareness Model to Web-Based Learning Environments 545
P. Herrero, A. de Antonio

An EC Services System Using Evolutionary Algorithm 549
W.D. Lin

A Fast and Efficient Method for Processing Web Documents 553
D. Szegő

Online Internet Monitoring System of Sea Regions 557
M. Piotrowski, H. Krawczyk

Modeling a 3G Power Control Algorithm in the MAC Layer for Multimedia Support 561
U. Pineda, C. Vargas, J. Acosta-Elías, J.M. Luna, G. Pérez, E. Stevens

Network Probabilistic Connectivity: Exact Calculation with Use of Chains 565
O.K. Rodionova, A.S. Rodionov, H. Choo

A Study of Anycast Application for Efficiency Improvement of Multicast Trees 569
K.-J. Lee, W.-H. Choi, J.-S. Kim

Performance Analysis of IP-Based Multimedia Communication Networks to Support Video Traffic 573
A.F. Yaroslavtsev, T.-J. Lee, M.Y. Chung, H. Choo

Limited Deflection Routing with QoS-Support 577
H. Kim, S. Lee, J. Song

Advanced Multicasting for DVBMT Solution 582
M. Kim, Y.-C. Bang, H. Choo

Server Mobility Using Domain Name System in Mobile IPv6 Networks 586
H. Sung, S. Han

Resource Reservation and Allocation Method for Next Generation Mobile Communication Systems 590
J. Lee, S.-P. Cho, C. Kang

Improved Location Scheme Using Circle Location Register in Mobile Networks 594
D.C. Lee, H. Kim, I.-S. Hwang

An Energy Efficient Broadcasting for Mobile Devices Using a Cache Scheme 598
K.-H. Han, J.-H. Kim, Y.-B. Ko, W.-S. Yoon

On Balancing Delay and Cost for Routing Paths 602
M. Kim, Y.-C. Bang, H. Choo

Performance of Optical Burst Switching in Time Division Multiplexed Wavelength-Routing Networks 607
T.-W. Um, Y. Kwon, J.K. Choi

On Algorithm for All-Pairs Most Reliable Quickest Paths 611
Y.-C. Bang, I. Hong, H. Choo

Performance Evaluation of the Fast Consistency Algorithms in Large Decentralized Systems 615
J. Acosta-Elías, L. Navarro-Moldes

Building a Formal Framework for Mobile Ad Hoc Computing 619
L. Yan, J. Ni

Efficient Immunization Algorithm for Peer-to-Peer Networks 623
H. Chen, H. Jin, J. Sun, Z. Han

A Secure Process-Service Model 627
S. Deng, Z. Wu, Z. Yu, L. Huang

Multi-level Protection Building for Virus Protection Infrastructure 631
S.-C. Noh, D.C. Lee, K.J. Kim

Parallelization of the IDEA Algorithm 635
V. Beletskyy, D. Burak

A New Authorization Model for Workflow Management System Using the RPI-RBAC Model 639
S. Lee, Y. Kim, B. Noh, H. Lee

Producing the State Space of RC4 Stream Cipher 644
V. Tomašević, S. Bojanić

A Pair-Wise Key Agreement Scheme in Ad Hoc Networks 648
W. Cha, G. Wang, G. Cho

Visual Analysis of the Multidimensional Meteorological Data 652
G. Dzemyda

Using Branch-Grafted R-trees for Spatial Data Mining 657
P. Dubey, Z. Chen, Y. Shi

Using Runtime Measurements and Historical Traces for Acquiring Knowledge in Parallel Applications 661
L. J. Senger, M. J. Santana, R.H.C. Santana

Words as Rules: Feature Selection in Text Categorization 666
E. Montañés, E.F. Combarro, I. Díaz, J. Ranilla, J.R. Quevedo

Proper Noun Learning from Unannotated Corpora for Information Extraction 670
S.-S. Kang

Proposition of Boosting Algorithm for Probabilistic Decision Support System 675
M. Wozniak

Efficient Algorithm for Linear Pattern Separation 679
C. Tadonki, J.-P. Vial

Improved Face Detection Algorithm in Mobile Environment 683
S.-B. Rhee, Y.-H. Lee

Real-Time Face Recognition by the PCA
(Principal Component Analysis) with Color Images 687
J.O. Kim, S.J. Seo, C.H. Chung

Consistency of Global Checkpoints Based on Characteristics
of Communication Events in Multimedia Applications 691
M. Ono, H. Higaki

Combining the Radon, Markov, and Stieltjes Transforms
for Object Reconstruction 695
A. Cuyt, B. Verdonk

Author Index 699

Table of Contents – Part III

Workshop on Programming Grids and Metasystems

High-Performance Parallel and Distributed Scientific Computing with the Common Component Architecture 1
D.E. Bernholdt

Multiparadigm Model Oriented to Development of Grid Systems 2
J.L.V. Barbosa, C.A. da Costa, A.C. Yamin, C.F.R. Geyer

The Effect of the 2^{nd} Generation Clusters: Changes in the Parallel Programming Paradigms 10
J. Porras, P. Huttunen, J. Ikonen

JavaSymphony, a Programming Model for the Grid 18
A. Jugravu, T. Fahringer

Adaptation of Legacy Software to Grid Services 26
B. Baliś, M. Bubak, M. Węgiel

Grid Service Registry for Workflow Composition Framework 34
M. Bubak, T. Gubała, M. Kapałka, M. Malawski, K. Rycerz

A-GWL: Abstract Grid Workflow Language 42
T. Fahringer, S. Pllana, A. Villazon

Design of Departmental Metacomputing ML 50
F. Gava

A Grid-Enabled Scene Rendering Application 54
M. Caballer, V. Hernández, J.E. Román

Rule-Based Visualization in a Computational Steering Collaboratory 58
L. Jiang, H. Liu, M. Parashar, D. Silver

Placement of File Replicas in Data Grid Environments 66
J.H. Abawajy

Generating Reliable Conformance Test Suites for Parallel and Distributed Languages, Libraries, and APIs 74
Ł. Garstecki

A Concept of Replicated Remote Method Invocation 82
J. Brzezinski, C. Sobaniec

Workshop on First International Workshop on Active and Programmable Grids Architectures and Components

Discovery of Web Services with a P2P Network 90
F. Forster, H. De Meer

Achieving Load Balancing in Structured Peer-to-Peer Grids 98
C. Pairot, P. García, A.F.G. Skarmeta, R. Mondéjar

A Conceptual Model for Grid-Adaptivity of HPC Applications and Its Logical Implementation Using Components Technology 106
A. Machì, S. Lombardo

Global Discovery Service for JMX Architecture 114
J. Midura, K. Balos, K. Zielinski

Towards a Grid Applicable Parallel Architecture Machine 119
K. Skala, Z. Sojat

A XKMS-Based Security Framework for Mobile Grid into the XML Web Services 124
N. Park, K. Moon, J. Jang, S. Sohn

A Proposal of Policy-Based System Architecture for Grid Services Management 133
E. Magaña, E. Salamanca, J. Serrat

Self-Management GRID Services – A Programmable Network Approach 141
L. Cheng, A. Galis, A. Savanović, B.J. Blažič, J. Bešter

Application-Specific Hints in Reconfigurable Grid Scheduling Algorithms 149
B. Volckaert, P. Thysebaert, F. De Turck, B. Dhoedt, P. Demeester

Self-Configuration of Grid Nodes Using a Policy-Based Management Architecture 158
F.J. García, Ó. Cánovas, G. Martínez, A.F.G. Skarmeta

Context-Aware GRID Services: Issues and Approaches 166
K. Jean, A. Galis, A. Tan

Security Issues in Virtual Grid Environments 174
J.L. Muñoz, J. Pegueroles, J. Forné, O. Esparza, M. Soriano

Implementation and Evaluation of Integrity Protection Facilities for Active Grids 179
A. Savanović, D. Gabrijelčič, B.J. Blažič, J. Bešter

A Convergence Architecture for GRID Computing and Programmable Networks 187
C. Bachmeir, P. Tabery, D. Marinov, G. Nachev, J. Eberspächer

Programmable Grids Framework Enabling QoS in an OGSA Context 195
J. Soldatos, L. Polymenakos, G. Kormentzas

Active and Logistical Networking for Grid Computing: The E-toile Architecture 202
A. Bassi, M. Beck, F. Chanussot, J.-P. Gelas, R. Harakaly, L. Lefèvre, T. Moore, J. Plank, P. Primet

Distributed Resource Discovery in Wide Area Grid Environments 210
T.N. Ellahi, M.T. Kechadi

Trusted Group Membership Service for JXTA 218
L. Kawulok, K. Zielinski, M. Jaeschke

Workshop on Next Generation Computing

An Implementation of Budget-Based Resource Reservation for Real-Time Linux 226
C.S. Liu, N.C. Perng, T.W. Kuo

Similarity Retrieval Based on SOM-Based R*-Tree 234
K.H. Choi, M.H. Shin, S.H. Bae, C.H. Kwon, I.H. Ra

Extending the Power of Server Based Computing 242
H.L. Yu, W.M. Zhen, M.M. Shen

Specifying Policies for Service Negotiations of Response Time 250
T.K. Kim, O.H. Byeon, K.J. Chun, T.M. Chung

Determination and Combination of Quantitative Weight Value from Multiple Preference Information 258
J.H. Yoo, E.G. Lee, H.S. Han

Forwarding Based Data Parallel Handoff for Real-Time QoS in Mobile IPv6 Networks 266
H. Y. Jeong, J. Lim, J.D. Park, H. Choo

Mobile Agent-Based Load Monitoring System for the Safety Web Server Environment 274
H.J. Park, K.J. Jyung, S.S. Kim

A Study on TCP Buffer Management Algorithm for Improvement of Network Performance in Grid Environment 281
Y. Jeong, M. Noh, H.K. Lee, Y. Mun

Workshop on Practical Aspects of High-Level Parallel Programming (PAPP 2004)

Evaluating the Performance of Skeleton-Based High Level Parallel Programs 289
A. Benoit, M. Cole, S. Gilmore, J. Hillston

Towards a Generalised Runtime Environment for Parallel Haskells 297
J. Berthold

Extending Camelot with Mutable State and Concurrency 306
S. Gilmore

EVE, an Object Oriented SIMD Library 314
J. Falcou, J. Sérot

Petri Nets as Executable Specifications of High-Level Timed Parallel Systems 322
F. Pommereau

Parallel I/O in Bulk-Synchronous Parallel ML 331
F. Gava

Workshop on Parallel Input/Output Management Techniques (PIOMT04)

File Replacement Algorithm for Storage Resource Managers in Data Grids 339
J.H. Abawajy

Optimizations Based on Hints in a Parallel File System 347
M.S. Pérez, A. Sánchez, V. Robles, J.M. Peña, F. Pérez

Using DMA Aligned Buffer to Improve Software RAID Performance 355
Z. Shi, J. Zhang, X. Zhou

mNFS: Multicast-Based NFS Cluster 363
W.-G. Lee, C.-I. Park, D.-W. Kim

Balanced RM2: An Improved Data Placement Scheme for Tolerating Double Disk Failures in Disk Arrays 371
D.-W. Kim, S.-H. Lee, C.-I. Park

Diagonal Replication on Grid for Efficient Access of Data in Distributed Database Systems 379
M. Mat Deris, N. Bakar, M. Rabiei, H.M. Suzuri

Workshop on OpenMP for Large Scale Applications

Performance Comparison between OpenMP and MPI on IA64 Architecture 388
L. Qi, M. Shen, Y. Chen, J. Li

Defining Synthesizable OpenMP Directives and Clauses 398
P. Dziurzanski, V. Beletskyy

Efficient Translation of OpenMP to Distributed Memory 408
L. Huang, B. Chapman, Z. Liu, R. Kendall

ORC-OpenMP: An OpenMP Compiler Based on ORC 414
Y. Chen, J. Li, S. Wang, D. Wang

Workshop on Tools for Program Development and Analysis in Computational Science

Performance Analysis, Data Sharing, and Tools Integration in Grids: New Approach Based on Ontology 424
H.-L. Truong, T. Fahringer

Accurate Cache and TLB Characterization Using Hardware Counters 432
J. Dongarra, S. Moore, P. Mucci, K. Seymour, H. You

A Tool Suite for Simulation Based Analysis of Memory Access Behavior 440
J. Weidendorfer, M. Kowarschik, C. Trinitis

Platform-Independent Cache Optimization by Pinpointing Low-Locality Reuse 448
K. Beyls, E.H. D'Hollander

Teuta: Tool Support for Performance Modeling of Distributed and Parallel Applications 456
T. Fahringer, S. Pllana, J. Testori

MPI Application Development Using the Analysis Tool MARMOT 464
B. Krammer, M.S. Müller, M.M. Resch

Monitoring System for Distributed Java Applications 472
W. Funika, M. Bubak, M. Smętek

Automatic Parallel-Discrete Event Simulation 480
M. Marín

Workshop on Modern Technologies for Web-Based Adaptive Systems

Creation of Information Profiles in Distributed Databases as a n-Person Game 488
J.L. Kulikowski

Domain Knowledge Modelling for Intelligent Instructional Systems 497
E. Pecheanu, L. Dumitriu, C. Segal

Hybrid Adaptation of Web-Based Systems User Interfaces 505
J. Sobecki

Collaborative Web Browsing Based on Ontology Learning from Bookmarks 513
J.J. Jung, Y.-H. Yu, G.-S. Jo

Information Retrieval Using Bayesian Networks 521
L. Neuman, J. Kozlowski, A. Zgrzywa

An Application of the DEDS Control Synthesis Method 529
F. Čapkovič

Using Consistency Measures and Attribute Dependencies for Solving Conflicts in Adaptive Systems 537
M. Malowiecki, N. T. Nguyen, M. Zgrzywa

Logical Methods for Representing Meaning of Natural Language Texts ... 545
T. Batura, F. Murzin

Software Self-Adaptability by Means of Artificial Evolution 552
M. Nowostawski, M. Purvis, A. Gecow

Professor:e – An IMS Standard Based Adaptive E-learning Platform 560
C. Segal, L. Dumitriu

Workshop on Agent Day 2004 – Intelligent Agents in Computing Systems

Towards Measure of Semantic Correlation between Messages in Multiagent System 567
A. Pieczyńska-Kuchtiak, R. Katarzyniak

Modelling Intelligent Virtual Agent Skills with Human-Like Senses 575
P. Herrero, A. de Antonio

Reuse of Organisational Experience Harnessing Software Agents 583
K. Krawczyk, M. Majewska, M. Dziewierz, R. Słota, Z. Balogh, J. Kitowski, S. Lambert

The Construction and Analysis of Agent Fault-Tolerance Model Based on π-Calculus 591
Y. Jiang, Z. Xia, Y. Zhong, S. Zhang

REMARK – Reusable Agent-Based Experience Management and Recommender Framework 599
Z. Balogh, M. Laclavik, L. Hluchy, I. Budinska, K. Krawczyk

Behavior Based Detection of Unfavorable Resources 607
K. Cetnarowicz, G. Rojek

Policy Modeling in Four Agent Economy 615
A. Woźniak

Multi-agent System for Irregular Parallel Genetic Computations 623
J. Momot, K. Kosacki, M. Grochowski, P. Uhruski, R. Schaefer

Strategy Extraction for Mobile Embedded Control Systems Apply the Multi-agent Technology 631
V. Srovnal, B. Horák, R. Bernatík, V. Snášel

Multi-agent Environment for Dynamic Transport Planning and Scheduling 638
J. Kozlak, J.-C. Créput, V. Hilaire, A. Koukam

Agent-Based Models and Platforms for Parallel Evolutionary Algorithms 646
M. Kisiel-Dorohinicki

A Co-evolutionary Multi-agent System for Multi-modal Function Optimization 654
R. Dreżewski

Workshop on Dynamic Data Driven Applications Systems

Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements 662
F. Darema

Distributed Collaborative Adaptive Sensing for Hazardous Weather Detection, Tracking, and Predicting 670
J. Brotzge, V. Chandresakar, K. Droegemeier, J. Kurose, D. McLaughlin, B. Philips, M. Preston, S. Sekelsky

Rule-Based Support Vector Machine Classifiers Applied to Tornado Prediction 678
T.B. Trafalis, B. Santosa, M.B. Richman

Adaptive Coupled Physical and Biogeochemical Ocean Predictions: A Conceptual Basis 685
P.F.J. Lermusiaux, C. Evangelinos, R. Tian, P.J. Haley, J.J. McCarthy, N.M. Patrikalakis, A.R. Robinson, H. Schmidt

Dynamic-Data-Driven Real-Time Computational Mechanics Environment 693
J. Michopoulos, C. Farhat, E. Houstis

A Note on Data-Driven Contaminant Simulation 701
C. C. Douglas, C.E. Shannon, Y. Efendiev, R. Ewing, V. Ginting, R. Lazarov, M.J. Cole, G. Jones, C.R. Johnson, J. Simpson

Computational Aspects of Data Assimilation for Aerosol Dynamics 709
A. Sandu, W. Liao, G.R. Carmichael, D. Henze, J.H. Seinfeld, T. Chai, D. Daescu

A Framework for Online Inversion-Based 3D Site Characterization 717
V. Akçelik, J. Bielak, G. Biros, I. Epanomeritakis, O. Ghattas, L.F. Kallivokas, E.J. Kim

A Note on Dynamic Data Driven Wildfire Modeling 725
J. Mandel, M. Chen, L.P. Franca, C. Johns, A. Puhalskii, J.L. Coen, C.C. Douglas, R. Kremens, A. Vodacek, W. Zhao

Agent-Based Simulation of Data-Driven Fire Propagation Dynamics 732
J. Michopoulos, P. Tsompanopoulou, E. Houstis, A. Joshi

Model Reduction of Large-Scale Dynamical Systems 740
A. Antoulas, D. Sorensen, K.A. Gallivan, P. Van Dooren, A. Grama, C. Hoffmann, A. Sameh

Data Driven Design Optimization Methodology Development and Application 748
H. Zhao, D. Knight, E. Taskinoglu, V. Jovanovic

A Dynamic Data Driven Computational Infrastructure for Reliable Computer Simulations 756
J.T. Oden, J.C. Browne, I. Babuška, C. Bajaj, L.F. Demkowicz, L. Gray, J. Bass, Y. Feng, S. Prudhomme, F. Nobile, R. Tempone

Improvements to Response-Surface Based Vehicle Design Using a Feature-Centric Approach 764
D. Thompson, S. Parthasarathy, R. Machiraju, S. Lawrence

An Experiment for the Virtual Traffic Laboratory: Calibrating Speed Dependency on Heavy Traffic (A Demonstration of a Study in a Data Driven Trafic Analysis) 771
A. Visser, J. Zoetebier, H. Yakali, B. Hertzberger

SAMAS: Scalable Architecture for Multi-resolution Agent-Based Simulation 779
A. Chaturvedi, J. Chi, S. Mehta, D. Dolk

Simulation Coercion Applied to Multiagent DDDAS 789
Y. Loitière, D. Brogan, P. Reynolds

O'SOAP – A Web Services Framework for DDDAS Applications 797
K. Pingali, P. Stodghill

Application of Grid-Enabled Technologies for Solving Optimization Problems in Data-Driven Reservoir Studies 805
M. Parashar, H. Klie, U. Catalyurek, T. Kurc, V. Matossian, J. Saltz, M.F. Wheeler

Image-Based Stress Recognition Using a Model-Based Dynamic Face Tracking System 813
D. Metaxas, S. Venkataraman, C. Vogler

Developing a Data Driven System for Computational Neuroscience 822
R. Snider, Y. Zhu

Karhunen–Loeve Representation of Periodic Second-Order Autoregressive Processes 827
D. Lucor, C.-H. Su, G.E. Karniadakis

Workshop on HLA-Based Distributed Simulation on the Grid

Using Web Services to Integrate Heterogeneous Simulations in a Grid Environment. 835
J.M. Pullen, R. Brunton, D. Brutzman, D. Drake, M. Hieb, K.L. Morse, A. Tolk

Support for Effective and Fault Tolerant Execution of HLA-Based Applications in the OGSA Framework 848
K. Rycerz, M. Bubak, M. Malawski, P.M.A. Sloot

Federate Migration in HLA-Based Simulation 856
Z. Yuan, W. Cai, M.Y.H. Low, S.J. Turner

FT-RSS: A Flexible Framework for Fault Tolerant HLA Federations 865
J. Lüthi, S. Großmann

Design and Implementation of GPDS 873
T.-D. Lee, S.-H. Yoo, C.-S. Jeong

HLA_AGENT: Distributed Simulation of Agent-Based Systems with HLA 881
M. Lees, B. Logan, T. Oguara, G. Theodoropoulos

FedGrid: An HLA Approach to Federating Grids 889
S. Vuong, X. Cai, J. Li, S. Pramanik, D. Suttles, R. Chen

Workshop on Interactive Visualisation and Interaction Technologies

Do Colors Affect Our Recognition Memory for Haptic Rough Surfaces? . . 897
Z. Luo, A. Imamiya

Enhancing Human Computer Interaction in Networked Hapto-Acoustic Virtual Reality Environments on the CeNTIE Network 905
T. Adriaansen, A. Krumm-Heller, C. Gunn

Collaborative Integration of Speech and 3D Gesture for Map-Based Applications 913
A. Corradini

Mobile Augmented Reality Support for Architects Based on Feature Tracking Techniques 921
M. Bang Nielsen, G. Kramp, K. Grønbæk

User Interface Design for a Navigation and Communication System in the Automotive World 929
O. Preißner

Target Selection in Augmented Reality Worlds 936
J. Sands, S. W. Lawson, D. Benyon

Towards Believable Behavior Generation for Embodied Conversational Agents 946
A. Corradini, M. Fredriksson, M. Mehta, J. Königsmann, N.O. Bernsen, L. Johannesson

A Performance Analysis of Movement Patterns 954
C. Sas, G. O'Hare, R. Reilly

On the Motivation and Attractiveness Scope of the Virtual Reality User Interface of an Educational Game 962
M. Virvou, G. Katsionis, K. Manos

A Client-Server Engine for Parallel Computation of High-Resolution Planes 970
D.P. Gavidia, E. V. Zudilova, P.M.A. Sloot

A Framework for 3D Polysensometric Comparative Visualization 978
J.I. Khan, X. Xu, Y. Ma

An Incremental Editor for Dynamic Hierarchical Drawing of Trees 986
D. Workman, M. Bernard, S. Pothoven

Using Indexed-Sequential Geometric Glyphs to Explore Visual Patterns 996
J. Morey, K. Sedig

Studying the Acceptance or Rejection of Newcomers in Virtual Environments 1004
P. Herrero, A. de Antonio, J. Segovia

Open Standard Based Visualization of Complex Internet Computing Systems 1008
S.S. Yang, J.I. Khan

General Conception of the Virtual Laboratory 1013
M. Lawenda, N. Meyer, T. Rajtar, M. Okoń, D. Stokłosa, M. Stroiński, Ł. Popenda, Z. Gdaniec, R. W. Adamiak

Individual Differences in Virtual Environments 1017
C. Sas

Ecological Strategies and Knowledge Mapping 1025
J. Bidarra, A. Dias

Need for a Prescriptive Taxonomy of Interaction for Mathematical Cognitive Tools 1030
K. Sedig

Workshop on Computational Modeling of Transport on Networks

Evolution of the Internet Map and Load Distribution 1038
K.-I. Goh, B. Kahng, D. Kim

Complex Network of Earthquakes 1046
S. Abe, N. Suzuki

Universal Features of Network Topology 1054
K. Austin, G.J. Rodgers

Network Brownian Motion: A New Method to Measure Vertex-Vertex Proximity and to Identify Communities and Subcommunities 1062
H. Zhou, R. Lipowsky

Contagion Flow through Banking Networks 1070
M. Boss, M. Summer, S. Thurner

Local Search with Congestion in Complex Communication Networks 1078
A. Arenas, L. Danon, A. Díaz-Guilera, R. Guimerà

Guided Search and Distribution of Information Flow on Complex Graphs 1086
B. Tadić

Network Topology in Immune System Shape Space 1094
J. Burns, H.J. Ruskin

An Evolutionary Approach to Pickup and Delivery Problem with Time Windows 1102
J.-C. Créput, A. Koukam, J. Kozlak, J. Lukasik

Automatic Extraction of Hierarchical Urban Networks: A Micro-Spatial Approach 1109
R. Carvalho, M. Batty

Workshop on Modeling and Simulation in Super-computing and Telecommunications

Design and Implementation of the Web-Based PSE *GridGate* 1117
K. Kang, Y. Kang, K. Cho

Performance Evaluation of ENUM Directory Service Design 1124
H.K. Lee, Y. Mun

A Single Thread Discrete Event Simulation Toolkit for Java: STSimJ 1131
W. Chen, D. Wang, W. Zheng

Routing and Wavelength Assignments in Optical WDM Networks with Maximum Quantity of Edge Disjoint Paths 1138
H. Choo, V. V. Shakhov

Parallelism for Nested Loops with Non-uniform and Flow Dependences 1146
S.-J. Jeong

Comparison Based Diagnostics as a Probabilistic Deduction Problem 1153
B. Polgár

Dynamic Threshold for Monitor Systems
on Grid Service Environments 1162
E.N. Huh

Multiuser CDMA Parameters Estimation by Particle Filter
with Resampling Schemes 1170
J.-S. Kim, D.-R. Shin, W.-G. Chung

Workshop on QoS Routing

Routing, Wavelength Assignment in Optical Networks Using
an Efficient and Fair EDP Algorithm 1178
P. Manohar, V. Sridhar

Route Optimization Technique to Support Multicast
in Mobile Networks 1185
K. Park, S. Han, B.-g. Joo, K. Kim, J. Hong

PRED: Prediction-Enabled RED 1193
M.G. Chung, E.N. Huh

An Efficient Aggregation and Routing Algorithm Using
Multi-hop Clustering in Sensor Networks 1201
B.-H. Lee, H.-W. Yoon, T.-J. Lee, M.Y. Chung

Explicit Routing for Traffic Engineering in Labeled Optical
Burst-Switched WDM Networks 1209
J. Zhang, H.-J. Lee, S. Wang, X. Qiu, K. Zhu, Y. Huang,
D. Datta, Y.-C. Kim, B. Mukherjee

A Mutual Authentication and Route Optimization Method
between MN and CN Using AAA in Mobile IPv6 1217
M. Kim, H.K. Lee, Y. Mun

Studies on a Class of AWG-Based Node Architectures for Optical
Burst-Switched Networks 1224
Y. Huang, D. Datta, X. Qiu, J. Zhang, H.-K. Park, Y.-C. Kim,
J.P. Heritage, B. Mukherjee

Self-Organizing Sensor Networks. 1233
D. Bein, A.K. Datta

Workshop on Evolvable Hardware

The Application of GLS Algorithm to 2 Dimension Irregular-Shape Cutting Problem 1241
L. Budzyńska, P. Kominek

Biologically-Inspired:
A Rule-Based Self-Reconfiguration of a Virtex Chip 1249
G. Tufte, P.C. Haddow

Designing Digital Circuits for the Knapsack Problem 1257
M. Oltean, C. Groşan, M. Oltean

Improvements in FSM Evolutions from Partial Input/Output Sequences 1265
S. G. Araújo, A. Mesquita, A.C.P. Pedroza

Intrinsic Evolution of Analog Circuits on a Programmable Analog Multiplexer Array 1273
J.F.M. Amaral, J.L.M. Amaral, C.C. Santini, M.A.C. Pacheco, R. Tanscheit, M.H. Szwarcman

Encoding Multiple Solutions in a Linear Genetic Programming Chromosome 1281
M. Oltean, C. Groşan, M. Oltean

Evolutionary State Assignment for Synchronous Finite State Machines . . . 1289
N. Nedjah, L. de Macedo Mourelle

Author Index 1297

Table of Contents – Part IV

Workshop on Advanced Methods of Digital Image Processing

The New Graphic Description of the Haar Wavelet Transform 1
P. Porwik, A. Lisowska

On New Radon-Based Translation, Rotation, and Scaling Invariant Transform for Face Recognition 9
T. Arodź

On Bit-Level Systolic Arrays for Least-Squares Digital Contour Smoothing 18
J. Glasa

Bayer Pattern Demosaicking Using Local-Correlation Approach 26
R. Lukac, K.N. Plataniotis, A.N. Venetsanopoulos

Edge Preserving Filters on Color Images 34
V. Hong, H. Palus, D. Paulus

Segmentation of Fundus Eye Images Using Methods of Mathematical Morphology for Glaucoma Diagnosis..... 41
K. Stąpor, A. Świtonski, R. Chrastek, G. Michelson

Automatic Detection of Glaucomatous Changes Using Adaptive Thresholding and Neural Networks 49
K. Stąpor, L. Pawlaczyk, R. Chrastek, G. Michelson

Analytical Design of 2-D Narrow Bandstop FIR Filters 56
P. Zahradnik, M. Vlček

Analytical Design of Arbitrary Oriented Asteroidal 2-D FIR Filters 64
P. Zahradnik, M. Vlček

A $\{k, n\}$-Secret Sharing Scheme for Color Images 72
R. Lukac, K.N. Plataniotis, A.N. Venetsanopoulos

Workshop on Computer Graphics and Geometric Modelling (CGGM 2004)

Declarative Modelling in Computer Graphics: Current Results and Future Issues 80
P.-F. Bonnefoi, D. Plemenos, W. Ruchaud

Geometric Snapping for 3D Meshes 90
K.-H. Yoo, J.S. Ha

Multiresolution Approximations of Generalized Voronoi Diagrams 98
I. Boada, N. Coll, J.A. Sellarès

LodStrips: Level of Detail Strips 107
J.F. Ramos, M. Chover

Declarative Specification of Ambiance in VRML Landscapes 115
V. Jolivet, D. Plemenos, P. Poulingeas

Using Constraints in Delaunay and Greedy Triangulation for Contour Lines Improvement 123
I. Kolingerová, V. Strych, V. Čada

An Effective Modeling of Single Cores Prostheses Using Geometric Techniques 131
K.-H. Yoo, J.S. Ha

GA and CHC. Two Evolutionary Algorithms to Solve the Root Identification Problem in Geometric Constraint Solving 139
M. V. Luzón, E. Barreiro, E. Yeguas, R. Joan-Arinyo

Manifold Extraction in Surface Reconstruction 147
M. Varnuška, I. Kolingerová

Expression of a Set of Points' Structure within a Specific Geometrical Model 156
J.-L. Mari, J. Sequeira

Effective Use of Procedural Shaders in Animated Scenes 164
P. Kondratieva, V. Havran, H.-P. Seidel

Real-Time Tree Rendering 173
I. Remolar, C. Rebollo, M. Chover, J. Ribelles

A Brush Stroke Generation Using Magnetic Field Model for Painterly Rendering 181
L.S. Yeon, Y.H. Soon, Y.K. Hyun

Reuse of Paths in Final Gathering Step with Moving Light Sources 189
M. Sbert, F. Castro

Real Time Tree Sketching 197
C. Campos, R. Quirós, J. Huerta, E. Camahort, R. Vivó, J. Lluch

Facial Expression Recognition Based on Dimension Model Using Sparse Coding 205
Y.-s. Shin

An Application to the Treatment of Geophysical Images through Orthogonal Projections 213
S. Romero, F. Moreno

A Derivative-Free Tracking Algorithm for Implicit Curves with Singularities 221
J.F.M. Morgado, A.J.P. Gomes

Framework for Simulating the Human Behavior for Intelligent Virtual Agents. Part I: Framework Architecture 229
F. Luengo, A. Iglesias

Framework for Simulating the Human Behavior for Intelligent Virtual Agents. Part II: Behavioral System 237
F. Luengo, A. Iglesias

Point-Based Modeling from a Single Image 245
P.-P. Vázquez, J. Marco, M. Sbert

Introducing Physical Boundaries in Virtual Environments 252
P. Herrero, A. de Antonio

Thin Client Access to a Visualization Environment 258
I. Fudos, I. Kyriazis

Interactive Visualization of Relativistic Effects with the Hardware Acceleration 264
R. Mantiuk, K. Murawko-Wisniewska, D. Zdrojewska

Workshop on Computer Algebra Systems and Applications (CASA 2004)

Design of Interactive Environment for Numerically Intensive Parallel Linear Algebra Calculations 270
P. Luszczek, J. Dongarra

Computer Algebra for Real-Time Dynamics of Robots with Large Numbers of Joints 278
R. Bansevicius, A. Cepulkauskas, R. Kulvietiene, G. Kulvietis

Development of SyNRAC—Formula Description and New Functions 286
H. Yanami, H. Anai

DisCAS: A Distributed-Parallel Computer Algebra System 295
Y. Wu, G. Yang, W. Zheng, D. Lin

A Mathematica Package for Solving and Displaying Inequalities 303
R. Ipanaqué, A. Iglesias

Choleski-Banachiewicz Approach to Systems with Non-positive Definite Matrices with MATHEMATICA® 311
R.A. Walentyński

A Closed Form Solution of the Run-Time of a Sliding Bead along a Freely Hanging Slinky 319
H. Sarafian

Analytical Theory of Motion of a Mars Orbiter 327
J.F. San Juan, S. Serrano, A. Abad

Computing Theta-Stable Parabolic Subalgebras Using LiE 335
A.G. Noël

Graphical and Computational Representation of Groups 343
A. Bretto, L. Gillibert

First Order ODEs: Mathematica and Symbolic-Numerical Methods 351
C. D'Apice, G. Gargiulo, M. Rosanna

Evaluation of the Fundamental Physical Constants in *Mathematica* 358
A.S. Siver

Symbolic Polynomial Interpolation Using Mathematica 364
A. Yazici, I. Altas, T. Ergenc

Constant Weight Codes with Package CodingTheory.m in Mathematica 370
I. Gashkov

Graph Coloring with web *Mathematica* 376
Ü. Ufuktepe, G. Bacak, T. Beseri

Construction of Solutions for Nonintegrable Systems with the Help of the Painlevé Test 382
S. Y. Vernov

Computer Algebra Manipulations in Astronomy 388
T. Ivanova

Workshop on New Numerical Methods for DEs: Applications to Linear Algebra, Control and Engineering

Higher Order Quadrature on Sparse Grids 394
H.-J. Bungartz, S. Dirnstorfer

Application of Extrapolation Methods to Numerical Solution of Fredholm Integral Equations Related to Boundary Value Problems 402
A. Sidi

Extrapolation Techniques for Computing Accurate Solutions of Elliptic Problems with Singular Solutions 410
H. Koestler, U. Ruede

Vandermonde–Type Matrices in Two Step Collocation Methods for Special Second Order Ordinary Differential Equations 418
S. Martucci, B. Paternoster

Direct Optimization Using Gaussian Quadrature and Continuous Runge-Kutta Methods: Application to an Innovation Diffusion Model 426
F. Diele, C. Marangi, S. Ragni

The ReLPM Exponential Integrator for FE Discretizations of Advection-Diffusion Equations 434
L. Bergamaschi, M. Caliari, M. Vianello

Function Fitting Two–Step BDF Algorithms for ODEs 443
L. G. Ixaru, B. Paternoster

Pseudospectral Iterated Method for Differential Equations with Delay Terms 451
J. Mead, B. Zubik-Kowal

A Hybrid Numerical Technique for the Solution of a Class of Implicit Matrix Differential Equation 459
N. Del Buono, L. Lopez

A Continuous Approach for the Computation of the Hyperbolic Singular Value Decomposition 467
T. Politi

Workshop on Parallel Monte Carlo Algorithms for Diverse Applications in a Distributed Setting

Using P-GRADE for Monte Carlo Computations in a Distributed Environment 475
V.N. Alexandrov, A. Thandavan, P. Kacsuk

Calculating Activation Energies in Diffusion Processes Using a Monte Carlo Approach in a Grid Environment 483
M. Calleja, M. T. Dove

Using Parallel Monte Carlo Methods in Large-Scale Air Pollution Modelling 491
V.N. Alexandrov, Z. Zlatev

Parallel Importance Separation for Multiple Integrals and Integral Equations 499
S. Ivanovska, A. Karaivanova

Investigation of the Sensitivity of the Monte Carlo Solution for the Barker-Ferry Equation with Sequential and Parallel Pseudo-Random Number Generators 507
T.V. Gurov, P.A. Whitlock

Design and Distributed Computer Simulation of Thin p^+–i–n^+ Avalanche Photodiodes Using Monte Carlo Model 515
M. Yakutovich

Convergence Proof for a Monte Carlo Method for Combinatorial Optimization Problems 523
S. Fidanova

Monte Carlo Algorithm for Maneuvering Target Tracking and Classification 531
D. Angelova, L. Mihaylova, T. Semerdjiev

Workshop on Modelling and Simulation of Multi-physics Multi-scale Systems

Coupling a Lattice Boltzmann and a Finite Difference Scheme 540
P. Albuquerque, D. Alemani, B. Chopard, P. Leone

Accuracy versus Performance in Lattice Boltzmann BGK Simulations of Systolic Flows 548
A.M. Artoli, L. Abrahamyan, A.G. Hoekstra

Mesoscopic Modelling of Droplets on Topologically Patterned Substrates 556
A. Dupuis, J.M. Yeomans

Soot Particle Deposition within Porous Structures Using a Method of Moments – Lattice Boltzmann Approach 564
B.F.W. Gschaider, C.C. Honeger, C.E.P. Redl

Numerical Bifurcation Analysis of Lattice Boltzmann Models: A Reaction-Diffusion Example 572
P. Van Leemput, K. Lust

Particle Models of Discharge Plasmas in Molecular Gases 580
S. Longo, M. Capitelli, P. Diomede

Fully Kinetic Particle-in-Cell Simulation of a Hall Thruster 588
F. Taccogna, S. Longo, M. Capitelli, R. Schneider

Standard of Molecular Dynamics Modeling and Simulation of Relaxation in Dense Media 596
A. Y. Kuksin, I. V. Morozov, G.E. Norman, V. V. Stegailov

Implicit and Explicit Higher Order Time Integration Schemes for Fluid-Structure Interaction Computations 604
A. van Zuijlen, H. Bijl

Accounting for Nonlinear Aspects in Multiphysics Problems: Application to Poroelasticity 612
D. Néron, P. Ladevèze, D. Dureisseix, B.A. Schrefler

Computational Modelling of Multi-field Ionic Continuum Systems 621
J. Michopoulos

Formation of Dwarf Galaxies in Reionized Universe with Heterogeneous Multi-computer System 629
T. Boku, H. Susa, K. Onuma, M. Umemura, M. Sato, D. Takahashi

A Multi-scale Numerical Study of the Flow, Heat, and Mass Transfer in Protective Clothing 637
M.P. Sobera, C.R. Kleijn, P. Brasser, H.E.A. Van den Akker

Thermomechanical Waves in SMA Patches under Small Mechanical Loadings 645
L. Wang, R.V.N. Melnik

Direct and Homogeneous Numerical Approaches to Multiphase Flows and Applications 653
R. Samulyak, T. Lu, Y. Prykarpatskyy

Molecular Dynamics and Monte Carlo Simulations for Heat Transfer in Micro and Nano-channels 661
A.J.H. Frijns, S.V. Nedea, A.J. Markvoort, A.A. van Steenhoven, P.A.J. Hilbers

Improved Semi-Lagrangian Stabilizing Correction Scheme for Shallow Water Equations 667
A. Bourchtein, L. Bourchtein

Bose-Einstein Condensation Studied by the Real-Time Monte Carlo Simulation in the Frame of Java Applet 673
M. Gall, R. Kutner, A. Majerowski, D. Żebrowski

Workshop on Gene, Genome, and Population Evolution

Life History Traits and Genome Structure: Aerobiosis and G+C Content in Bacteria 679
J.R. Lobry

Differential Gene Survival under Asymmetric Directional Mutational Pressure 687
P. Mackiewicz, M. Dudkiewicz, M. Kowalczuk, D. Mackiewicz, J. Banaszak, N. Polak, K. Smolarczyk, A. Nowicka, M.R. Dudek, S. Cebrat

How Gene Survival Depends on Their Length 694
N. Polak, J. Banaszak, P. Mackiewicz, M. Dudkiewicz, M. Kowalczuk, D. Mackiewicz, K. Smolarczyk, A. Nowicka, M.R. Dudek, S. Cebrat

Super-tree Approach for Studying the Phylogeny of Prokaryotes: New Results on Completely Sequenced Genomes 700
A. Calteau, V. Daubin, G. Perrieère

Genetic Paralog Analysis and Simulations 709
S. Cebrat, J.P. Radomski, D. Stauffer

Evolutionary Perspectives on Protein Thermodynamics 718
R.A. Goldstein

The Partition Function Variant of Sankoff's Algorithm 728
I.L. Hofacker, P.F. Stadler

Simulation of Molecular Evolution Using Population Dynamics Modelling 736
S. V. Semovski

Lotka-Volterra Model of Macro-Evolution on Dynamical Networks 742
F. Coppex, M. Droz, A. Lipowski

Simulation of a Horizontal and Vertical Disease Spread in Population 750
M. Magdoń-Maksymowicz

Evolution of Population with Interaction between Neighbours 758
M. Magdoń-Maksymowicz, A.Z. Maksymowicz

The Role of Dominant Mutations in the Population Expansion 765
S. Cebrat, A. Pękalski

Workshop on Computational Methods in Finance and Insurance

On the Efficiency of Simplified Weak Taylor Schemes for Monte Carlo Simulation in Finance 771
N. Bruti Liberati, E. Platen

Time-Scale Transformations: Effects on VaR Models 779
F. Lamantia, S. Ortobelli, S. Rachev

Environment and Financial Markets 787
W. Szatzschneider, M. Jeanblanc, T. Kwiatkowska

Pricing of Some Exotic Options with *NIG*-Lévy Input 795
S. Rasmus, S. Asmussen, M. Wiktorsson

Construction of Quasi Optimal Portfolio for Stochastic Models of Financial Market 803
A. Janicki, J. Zwierz

Euler Scheme for One-Dimensional SDEs with Time Dependent Reflecting Barriers 811
L. Słomiński, T. Wojciechowski

On Approximation of Average Expectation Prices for Path Dependent Options in Fractional Models 819
B. Ziemkiewicz

Confidence Intervals for the Autocorrelations of the Squares of GARCH Sequences 827
P. Kokoszka, G. Teyssière, A. Zhang

Performance Measures in an Evolutionary Stock Trading Expert System 835
P. Lipinski, J.J. Korczak

Stocks' Trading System Based on the Particle Swarm Optimization Algorithm 843
J. Nenortaite, R. Simutis

Parisian Options – The Implied Barrier Concept 851
J. Anderluh, H. van der Weide

Modeling Electricity Prices with Regime Switching Models 859
M. Bierbrauer, S. Trück, R. Weron

Modeling the Risk Process in the XploRe Computing Environment 868
K. Burnecki, R. Weron

Workshop on Computational Economics and Finance

A Dynamic Stochastic Programming Model for Bond Portfolio Management 876
L. Yu, S. Wang, Y. Wu, K.K. Lai

Communication Leading to Nash Equilibrium without Acyclic Condition (– **S4**-Knowledge Model Case –) 884
T. Matsuhisa

Support Vector Machines Approach to Credit Assessment 892
J. Li, J. Liu, W. Xu, Y. Shi

Measuring Scorecard Performance 900
Z. Yang, Y. Wang, Y. Bai, X. Zhang

Parallelism of Association Rules Mining and Its Application in Insurance Operations 907
J. Tian, L. Zhu, S. Zhang, G. Huang

No Speculation under Expectations in Awareness 915
K. Horie, T. Matsuhisa

A Method on Solving Multiobjective Conditional Value-at-Risk 923
M. Jiang, Q. Hu, Z. Meng

Cross-Validation and Ensemble Analyses on Multiple-Criteria Linear Programming Classification for Credit Cardholder Behavior 931
Y. Peng, G. Kou, Z. Chen, Y. Shi

Workshop on GeoComputation

A Cache Mechanism for Component-Based WebGIS 940
Y. Luo, X. Wang, Z. Xu

A Data Structure for Efficient Transmission of Generalised Vector Maps 948
M. Zhou, M. Bertolotto

Feasibility Study of Geo-spatial Analysis Using Grid Computing 956
Y. Hu, Y. Xue, J. Wang, X. Sun, G. Cai, J. Tang, Y. Luo, S. Zhong, Y. Wang, A. Zhang

An Optimum Vehicular Path Solution with Multi-heuristics 964
F. Lu, Y. Guan

An Extended Locking Method for Geographical Database with Spatial Rules 972
C. Cheng, P. Shen, M. Zhang, F. Lu

Preliminary Study on Unsupervised Classification of Remotely Sensed Images on the Grid 981
J. Wang, X. Sun, Y. Xue, Y. Hu, Y. Luo, Y. Wang, S. Zhong, A. Zhang, J. Tang, G. Cai

Experience of Remote Sensing Information Modelling with Grid Computing 989
G. Cai, Y. Xue, J. Tang, J. Wang, Y. Wang, Y. Luo, Y. Hu, S. Zhong, X. Sun

Load Analysis and Load Control in Geo-agents 997
Y. Luo, X. Wang, Z. Xu

Workshop on Simulation and Modeling of 3D Integrated Circuits

Challenges in Transmission Line Modeling at Multi-gigabit Data Rates 1004
V. Heyfitch

MPI-Based Parallelized Model Order Reduction Algorithm 1012
I. Balk, S. Zorin

3D-VLSI Design Tool 1017
R. Bollapragada

Analytical Solutions of the Diffusive Heat Equation as the Application for Multi-cellular Device Modeling – A Numerical Aspect . . . 1021
Z. Lisik, J. Wozny, M. Langer, N. Rinaldi

Layout Based 3D Thermal Simulations of Integrated Circuits Components 1029
K. Slusarczyk, M. Kaminski, A. Napieralski

Simulation of Electrical and Optical Interconnections for Future VLSI ICs 1037
G. Tosik, Z. Lisik, M. Langer, F. Gaffiot, I. O'Conor

Balanced Binary Search Trees Based Approach for Sparse Matrix Representation 1045
I. Balk, I. Pavlovsky, A. Ushakov, I. Landman

Principles of Rectangular Mesh Generation in Computational Physics . . . 1049
V. Ermolaev, E. Odintsov, A. Sobachkin, A. Kharitonovich, M. Bevzushenko, S. Zorin

Workshop on Computational Modeling and Simulation on Biomechanical Engineering

Inter-finger Connection Matrices 1056
V.M. Zatsiorsky, M.L. Latash, F. Danion, F. Gao, Z.-M. Li, R.W. Gregory, S. Li

Biomechanics of Bone Cement Augmentation with Compression Hip Screw System for the Treatment of Intertrochanteric Fractures 1065
S.J. Lee, B.J. Kim, S.Y. Kwon, G.R. Tack

Comparison of Knee Cruciate Ligaments Models Using Kinematics from a Living Subject during Chair Rising-Sitting 1073
R. Stagni, S. Fantozzi, M. Davinelli, M. Lannocca

Computer and Robotic Model of External Fixation System for Fracture Treatment 1081
Y.H. Kim, S.-G. Lee

Robust Path Design of Biomechanical Systems Using the Concept of Allowable Load Set 1088
J.H. Chang, J.H. Kim, B.M. Kwak

A New Modeling Method for Objects with Branching Problem Using Non-uniform B-Spline 1095
H.S. Kim, Y.H. Kim, Y.H. Choe, S.-M. Kim, T.-S. Cho, J.H. Mun

Motion Design of Two-Legged Locomotion Process of a Man 1103
S. Novikava, K. Miatliuk, K. Jaworek

Adaptive Microcalcification Detection in Computer Aided Diagnosis 1110
H.-K. Kang, S.-M. Kim, N.N. Thanh, Y.M. Ro, W.-H. Kim

Workshop on Information Technologies Enhancing Health Care Delivery

The Impact of Information Technology on Quality of Healthcare Services 1118
M. Duplaga

Computer Generated Patient Plans Based on Patterns of Care 1126
O.M. Winnem

On Direct Comparing of Medical Guidelines with Electronic Health Record 1133
J. Zvárová, A. Veselý, P. Hanzliček, J. Špidlen, D. Buchtela

Managing Information Models for E-health via Planned Evolutionary Pathways 1140
H. Duwe

An Attributable Role-Based Access Control for Healthcare 1148
D. Schwartmann

Aspects of a Massively Distributed Stable Component Space 1156
K. Schmaranz, D. Schwartmann

Demonstrating Wireless IPv6 Access to a Federated Health Record Server 1165
D. Kalra, D. Ingram, A. Austin, V. Griffith, D. Lloyd, D. Patterson, P. Kirstein, P. Conversin, W. Fritsche

Collaborative Teleradiology 1172
K. Zieliński, J. Cała, Ł. Czekierda, S. Zieliński

Workshop on Computing in Science and Engineering Academic Programs

Some Remarks on CSE Education in Germany 1180
H.-J. Bungartz

The Computational Science and Engineering (CS&E) Program at Purdue University 1188
T. Downar, T. Kozlowski

Adapting the CSE Program at ETH Zurich to the Bologna Process 1196
R. Jeltsch, K. Nipp

Computational Engineering and Science Program at the University of Utah 1202
C. DeTar, A.L. Fogelson, C.R. Johnson, C.A. Sikorski, T. Truong

A Comparison of C, MATLAB, and Python as Teaching Languages in Engineering 1210
H. Fangohr

Teaching Computational Science Using VPython and Virtual Reality 1218
S. Roberts, H. Gardner, S. Press, L. Stals

Student Exercises on Fossil Fuels, Global Warming, and Gaia 1226
B. W. Rust

Teaching Scientific Computing 1234
B.A. Shadwick

Creating a Sustainable High-Performance Scientific Computing Course . . . 1242
E.R. Jessup, H.M. Tufo

CSE without Math? A First Course in Modeling and Simulation 1249
W. Wiechert

Author Index 1257

Hierarchical Matrix-Matrix Multiplication Based on Multiprocessor Tasks

Sascha Hunold[1], Thomas Rauber[1], and Gudula Rünger[2]

[1] Fakultät für Mathematik, Physik und Informatik, Universität Bayreuth, Germany
[2] Fakultät für Informatik, Technische Universität Chemnitz, Germany

Abstract. We consider the realization of matrix-matrix multiplication and propose a hierarchical algorithm implemented in a task-parallel way using multiprocessor tasks on distributed memory. The algorithm has been designed to minimize the communication overhead while showing large locality of memory references. The task-parallel realization makes the algorithm especially suited for cluster of SMPs since tasks can then be mapped to the different cluster nodes in order to efficiently exploit the cluster architecture. Experiments on current cluster machines show that the resulting execution times are competitive with state-of-the-art methods like *PDGEMM*.

1 Introduction

Matrix multiplication is one of the core computations in many algorithms of scientific computing and numerical analysis. Many different implementations have been realized over the years, including parallel ones. On a single processor *ATLAS* [7] or PHiPAC [1] create efficient implementations by exploiting the specific memory hierarchy and its properties. Parallel approaches are often based on decomposition, like Cannon's algorithm or the algorithm of Fox. Efficient implementation variants of the latter are *SUMMA* or *PUMMA*, see also [3] for more references. Matrix-matrix multiplication by Strassen or Strassen-Winograd benefits from a reduced number of operations but require a special schedule for a parallel implementation. Several parallel implementations have been proposed in [2,5,4].

Most clusters use two or more processors per node so that the data transfer between the local processors of a node is much faster than the data transfer between processors of different nodes. It is therefore often beneficial to exploit this property when designing parallel algorithms. A task parallel realization based on multiprocessor tasks (M-tasks) is often suited, as the M-tasks can be mapped to the nodes of the system such that the intra-task communication is performed within the single nodes. This can lead to a significant reduction of the communication overhead and can also lead to an efficient use of the local memory hierarchy. Based on this observation, we propose an algorithm for matrix multiplication which is hierarchically organized and implemented with multiprocessor tasks. At each hierarchy level recursive calls are responsible for the computation of different blocks with hierarchically increasing size of the result matrix. The

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 1–8, 2004.
© Springer-Verlag Berlin Heidelberg 2004

processors are split into subgroups according to the hierarchical organization which leads to a minimization of data transfer required. Moreover, only parts of one input matrix are moved to other processors during the execution of the algorithm, i.e., the local parts of the other matrix can be kept permanently in the local cache of the processors.

We have performed experiments on three different platforms, an IBM Regatta p690, a dual Xeon cluster with an SCI interconnection network, and a Pentium III cluster with a fast Ethernet interconnect. For up to 16 processors, the algorithm is competitive with the *PDGEMM* method from *ScaLAPACK* and outperforms this method in many situations. Thus the algorithm is well-suited to be used as a building block for other task parallel algorithms.

The rest of the paper is organized as follows. Section 2 describes the hierarchical algorithm. The implementation of the algorithm is presented in Section 3. Section 4 presents experimental results and Section 5 concludes the paper.

2 Hierarchical Matrix Multiplication

The hierarchical matrix multiplication performs a matrix multiplication $A \cdot B = C$ of an $m \times n$ matrix A and an $n \times k$ matrix B in a recursively blockwise manner on p processors. We assume that $p = 2^i, i \in \mathbb{N}$, and that p divides m and k without remainder. During the entire algorithm the input matrix A is distributed in a row blockwise manner, i.e. processor q stores the rows with indices $((q-1)\cdot s+1, \ldots, q\cdot s), s = m/p, q = 1, \ldots, p$. Input matrix B is distributed columnwise with varying mappings in the computation phases. Initially the distribution is column blockwise, i.e. processor q stores the columns with indices $((q-1)\cdot s'+1, \ldots, q\cdot s')$. The columns are exchanged in later steps, see Figure 1.

The hierarchical matrix multiplication computes the result matrix C in $\log p + 1$ steps and processor q is responsible for the computation of the s rows with indices $((q-1)\cdot s+1, \ldots, q\cdot s)$ of C. The computation is organized so that disjoint processor groups compute the diagonal blocks C_{lk} in parallel, which contain the $(s \cdot 2^{l-1})^2$ entries c_{ij} with $2^{l-1} \cdot (k-1) \cdot s + 1 \leq i, j \leq 2^{l-1} \cdot k \cdot s$. The coarse computational structure is the following:

```
Hierarchical Matrix Multiplication(n,p) =
  for (l = 1 to log p + 1)
    for k = 1, ..., p/2^(l-1) compute in parallel
      compute_block (C_lk);
```

Figure 1, bottom row, illustrates the computation of blocks C_{lk}. A diagonal block C_{lk} is computed by calling compute_block(C_{lk}) which is performed in parallel by all processors of a group. If in one group only a single processor q performs compute_block, i.e. $l = 1$, this processor computes one initial diagonal block C_{lk} by using its local entries of A and B. Otherwise, the computation of the two diagonal sub-blocks $C_{l-1,2k-1}$ and $C_{l-1,2k}$ of C_{lk} have already been completed in the preceding step by two other processor groups and the computation of C_{lk} is completed by computing the remaining sub-blocks $C'_{l-1,2k-1}$ and $C'_{l-1,2k}$ in the following way:

The initial column blocks of B are virtually grouped into larger column blocks according to the hierarchical binary clustering: for $1 \leq l \leq \log p + 1$ and $1 \leq k \leq p/2^{l-1}$, column block B_{lk} contains $s' \cdot 2^{l-1}$ columns of B; these columns have the indices $(2^{l-1}(k-1) \cdot s' + 1, \ldots, 2^{l-1}k \cdot s')$. The first index l of B_{lk} determines the size of the column block, the second index k numbers the column blocks of the same size.

The function compute_block() first exchanges the column blocks $B_{l-1,2k-1}$ and $B_{l-1,2k}$ of matrix B that are needed for the computation of $C'_{l-1,2k-1}$ and $C'_{l-1,2k}$, respectively, between the processors of the corresponding groups. This can be done in parallel since the processors of the group can be grouped into pairs which exchange their data. After the transfer operations the sub-blocks $C'_{l-1,2k-1}$ and $C'_{l-1,2k}$, respectively, are computed in parallel by recursive calls. At any point in time, each local memory needs to store at most s rows of A and s' columns of B and only columns of B are exchanged between the local memories.

3 Task Parallel Implementation

The realization of the task parallel matrix multiplication *(tpMM)* is based on a hierarchy of multiprocessor groups. The resulting implementation uses the run-time library *Tlib* which supports the programming with hierarchically structured M-tasks and provides a tool to handle multiprocessor groups built on top of MPI communicators [6].

The program realizes the recursive structure of the algorithm and uses a description of the block of the result matrix C that is computed in the current recursion step. This description contains the start column and the extent of the sub-block. The implementation exploits that the algorithm fills the basic blocks of C by alternating between basic blocks in the diagonal and the anti-diagonal position, see Figure 1. More precisely, the recursion in each phase subdivides the current block of C into sub-blocks containing 2×2 basic blocks, which are then filled in the diagonal and anti-diagonal direction. The program of *tpMM* uses the functions below. The variables $A, B, C, m, n, k, m_A = m/p$ and $k_B = k/p$ are declared and defined globally.

compute_block(*comm*, *lcc, cc, type*) is the recursive function for computing $C = A \cdot B$. `comm` is the current communicator of the recursion step. `lcc` denotes the leftmost column of C and `cc` specifies the number of columns of C for the next recursion step. `type` $\in$ {DIAGONAL, ANTIDIAGONAL} indicates if compute_block updates a diagonal or anti-diagonal block of C.

multiply(*cc*, *lcc*) performs the actual work of multiplying two sub-matrices and computes one basic block of C. The function is performed on a single processor and is realized by using fast one-processor implementations such as *BLAS* or *ATLAS*.

exchange_columns(*comm*) performs the data exchange between pairs of processors in the current communicator. For each call of the function, each processor participates in exactly one data exchange. The function ensures that

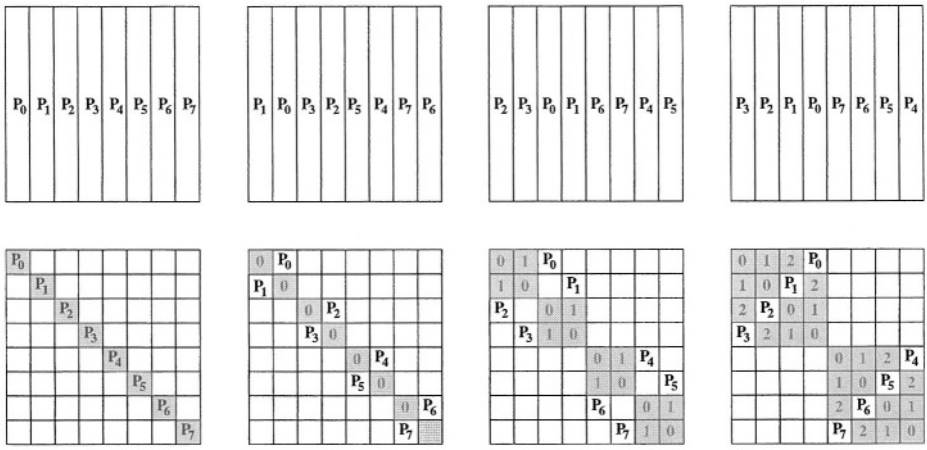


Fig. 1. Data distribution of matrix B (top row) and computation order of the result matrix (bottom row) for processors $P_0, \ldots, P_7$ for the first half of the steps. Each block is labeled with the owning processor. The numbers 0,1,2 denote the phase in which the blocks of C are computed.

processor P_i sends/receives a block of B to/from P_j, $j = (i + \text{sizeof}(comm)/2) \bmod \text{sizeof}(comm)$.

The pseudo-code of `compute_block` is given below. To perform a multiplication the programmer just needs to call `compute_block` and pass the corresponding parameters. The computation phases of *tpMM* reuse the communicators several times according to the recursive structure of the algorithm. Figure 2 illustrates the recursive splitting and the communicator reuse for $p = 8$ processors.

```
function compute_block(comm, lcc, cc, type)
  if ( Comm_size(comm) == 1 ) { multiply(cc, lcc) }
  else {
    subcomm_i = split(comm); /* splitting into subcommunicator i={0,1} */
    if ( type == DIAGONAL )
      { lcc_0 = lcc; lcc_1 = lcc+cc/2 }
    else
      { lcc_0 = lcc+cc/2; lcc_1 = lcc }
    compute_block(subcomm_i, lcc_i, cc/2, DIAGONAL); /* task parallel */
    exchange_columns(comm);
    if ( type == DIAGONAL )
      { lcc_0 = lcc+cc/2; lcc_1 = lcc }
    else
      { lcc_0 = lcc; lcc_1 = lcc+cc/2 }
    compute_block(subcomm_i, lcc_i, cc/2, ANTIDIAGONAL); /* task parallel */
  }
```

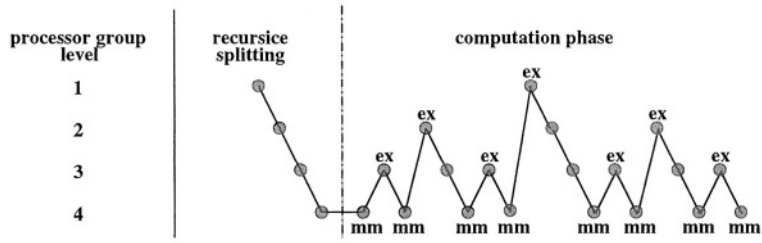


Fig. 2. Usage of processor groups during the computation of *tpMM* for three recursive splittings into sub-groups and four hierarchical levels. The matrices to be multiplied are decomposed into eight blocks of rows and columns, respectively. *mm* denotes the matrix multiplication for a single block, *ex* denotes the exchange of data at the corresponding communicator level.

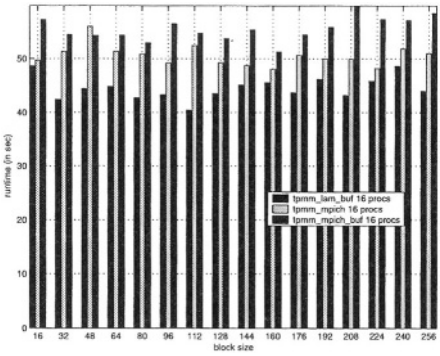


Fig. 3. *tpMM* overlapping tests on *CLiC* for matrix dimension $n = 4096$ and 16 processors.

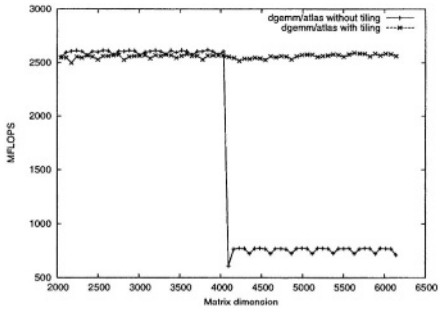


Fig. 4. comparison of *DGEMM* from *ATLAS* with and without tiling enabled (on *dual Beowulf cluster;* m, k=2048; n is varying).

4 Experimental Results

The runtime tests of *tpMM* were performed on a *IBM Regatta p690* (AIX, 6 x 32 processors, Power4+ 1.7GHz, Gigabit Ethernet) operated by the Research Centre Jülich, on a Linux *dual Beowulf cluster* (16 x 2 procs., Xeon 2.0 GHz, SCI network) and the *CLiC* (Chemnitzer Linux Cluster, 528 procs., P3 800 MHz, Fast-Ethernet) at TU Chemnitz.

Minimizing communication costs. The communication overhead of many applications can be reduced by overlapping communication with computation. To apply overlapping to *tpMM,* the block of *B* that each processor holds is not transfered entirely in one block. The blocks are rather send simultaneously in multiple smaller sub-blocks while performing local updates of matrix *C.* This requires non-blocking send and recv operations. Figure 3 shows runtime tests on *CLiC* using mpich and lam. The suffix "buf" refers to `MPI_Ibsend`, the

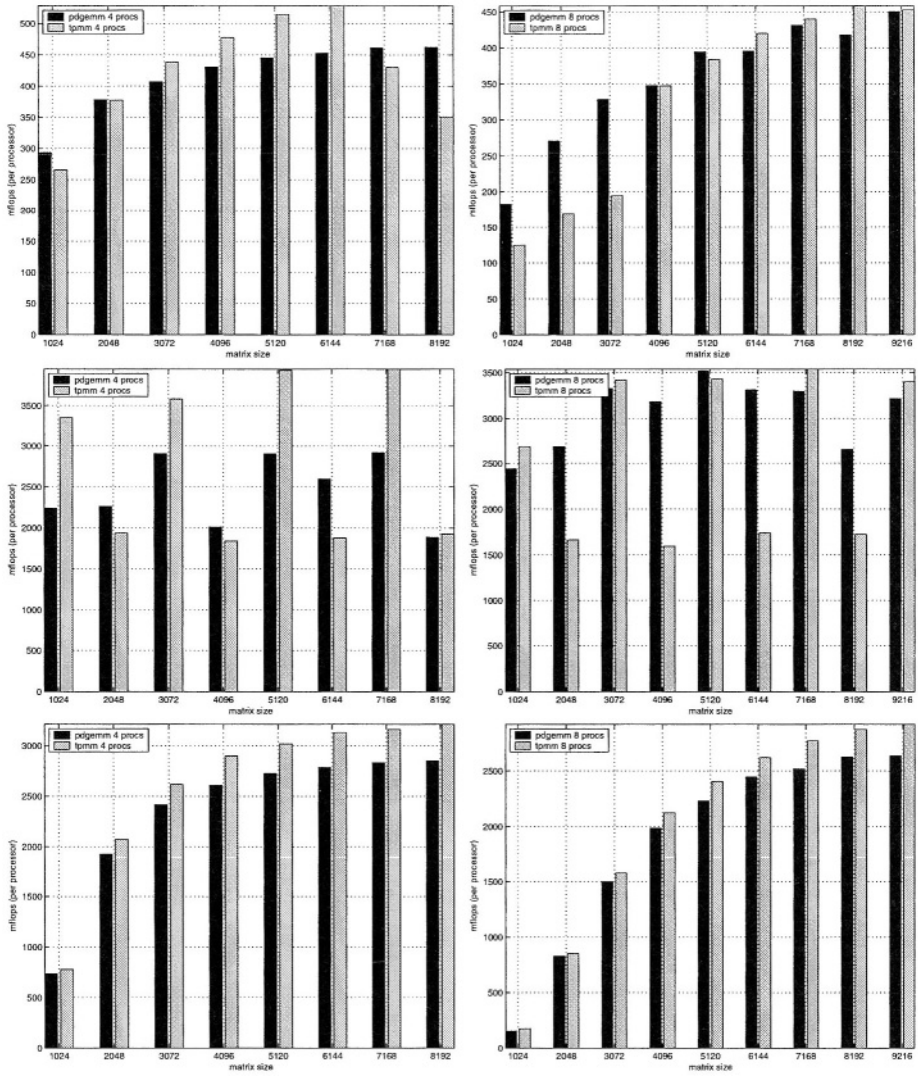


Fig. 5. MFLOPS per node reached by *PDGEMM* and *tpMM* on *CLiC, IBM Regatta p690* and *dual Beowulf cluster* (top to bottom).

buffered version of `MPI_Isend`. For these tests matrices A, B, and C of dimension 4096×4096 and 16 processors are used, so that each processor holds 256 columns of B. In the experiments local updates with block sizes (matrix B) of $4 \leq \text{blocksize} \leq 256$ are performed. For the full block size of 256, no overlapping is achieved and this result can be used for comparison. The experiments show that neither non-blocking nor non-blocking buffered communication leads to a significant and predictable improvement.

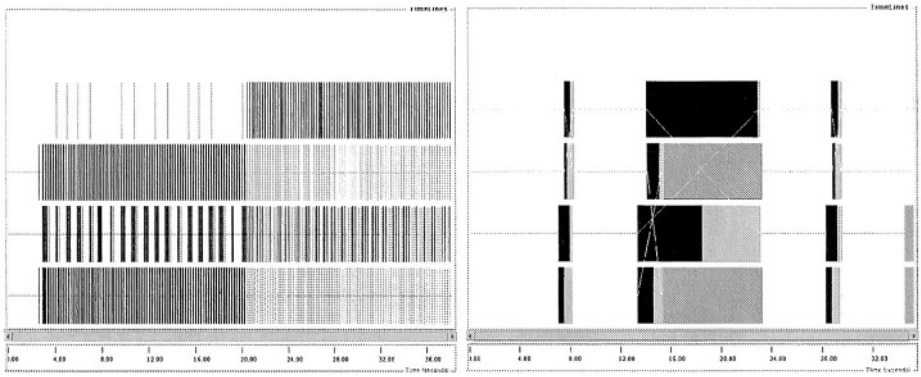


Fig. 6. JUMPSHOT-4 profiles of *PDGEMM* (upper) and *tpMM* (below) recorded on *dual Beowulf cluster* using 4 processors. Darker boxes represent Send-operations and segments in light grey denote either calls to MPI_Recv or MPI_Wait in case of non-blocking communication.

Underlying libraries. Low level matrix-matrix multiplications on one processor (*BLAS* level 3) are performed by *ATLAS* [7] which optimizes itself at compile time to gain maximum performance for a given architecture. Runtime experiments of *tpMM* on *dual Beowulf cluster* with more than 8 processors show a dramatic drop of the MFLOPS rate per node when using larger matrices (> 4096). According to a detailed profiling analysis the performance loss is caused by an internal call to *DGEMM*. Tests with a series of *DGEMM* matrix-matrix multiplications with fixed dimensions of m_A and k_B and variable n are presented in Figure 4. It turned out that when there are more than twice as many rows of B as columns *ATLAS* internally calls a different function which results in poor performance. This situation is likely to happen when executing *tpMM* with large input matrices. One possible work-around is a tiling approach of the original multiplication by dividing the problem into multiple sub-problems. The tiling of the local matrices A and B must ensure that each tile is as big as possible and two tiles must fulfill the requirements to perform a matrix-matrix multiplication (columns of tile t_i^A = rows of t_j^B). With tiling the local matrix-matrix multiplication achieves a similar MFLOPS-rate for all inputs (see Figure 4).

Overall performance evaluation of tpMM. Figures 5 shows the MFLOPS reached by *DGEMM* and *tpMM* on the three test systems considered. Since both methods perform the same number of operations (in different order), a larger MFLOPS rate corresponds to a smaller execution time. The figures show that for 4 processors, *tpMM* leads to larger MFLOPS rates on all three machines for most matrix sizes. For 8 processors, *tpMM* is usually slightly faster than *DGEMM*. For 16 processors, *tpMM* is faster only for the IBM Regatta system. The most significant advantages of *tpMM* can be seen for the IBM Regatta system. For 32 and more processors, *DGEMM* outperforms *tpMM* in most cases.

Figure 6 presents trace profiles of *PDGEMM* and *tpMM*. The profile of *PDGEMM* contains a huge number of communications even though only 4 processors were involved. In contrast, the pattern of *tpMM* shows only a small number of required communication calls. *PDGEMM* is superior if there are many processors involved and the matrix is sufficiently large. In these cases overlapping of computation with communication can be achieved and the block size remains suitable to avoid cache effects and communication overhead. On the other hand, *tpMM* decreases the communication overhead (e.g. numerous startup times) what makes it faster for a smaller group of nodes. Thus, *tpMM* is a good choice for parallel systems of up to 16 processors. For larger parallel systems, *tpMM* can be used as a building block in parallel algorithms with a task parallel structure of coarser granularity.

5 Conclusions

We have proposed a hierarchical algorithm for matrix multiplication which shows good performance for smaller numbers of processors. Our implementation outperforms *PDGEMM* for up to 16 processors on recent machines. Due to the good locality behavior, *tpMM* is well suited as building block in hierarchical matrix multiplication algorithms in which *tpMM* is called on smaller sub-clusters. Experiments have shown that *tpMM* can be combined with one-processor implementations which have been designed carefully to achieve a good overall performance.

References

1. J. Bilmes, K. Asanovic, C.-W. Chin, and J. Demmel. Optimizing matrix multiply using PHiPAC: A portable, high-performance, ANSI c coding methodology. In *International Conference on Supercomputing,* pages 340–347, 1997.
2. Frédéric Desprez and Frédéric Suter. Impact of Mixed-Parallelism on Parallel Implementations of Strassen and Winograd Matrix Multiplication Algorithms. Technical Report RR2002-24, Laboratoire de l'Informatique du Parallélisme (LIP), June 2002. Also INRIA Research Report RR-4482.
3. R. A. Van De Geijn and J. Watts. SUMMA: scalable universal matrix multiplication algorithm. *Concurrency: Practice and Experience,* 9(4):255–274, 1997.
4. Brian Grayson, Ajay Shah, and Robert van de Geijn. A High Performance Parallel Strassen Implementation. Technical Report CS-TR-95-24, Department of Computer Sciences, The Unversity of Texas, 1, 1995.
5. Qingshan Luo and John B. Drake. A Scalable Parallel Strassen's Matrix Multiplication Algorithm for Distributed-Memory Computers. In *Proceedings of the 1995 ACM symposium on Applied computing,* pages 221–226. ACM Press, 1995.
6. T. Rauber and G. Rünger. Library Support for Hierarchical Multi-Processor Tasks. In *Proc. of the Supercomputing 2002,* Baltimore, USA, 2002.
7. R. Clint Whaley and Jack J. Dongarra. Automatically Tuned Linear Algebra Software. Technical Report UT-CS-97-366, University of Tennessee, 1997.

Improving Geographical Locality of Data for Shared Memory Implementations of PDE Solvers

Henrik Löf, Markus Nordén, and Sverker Holmgren

Uppsala University, Department of Information Technology
P.O. Box 337, SE-751 05 Uppsala, Sweden
{henrik.lof,markus.norden,sverker.holmgren}@it.uu.se

Abstract. On cc-NUMA multi-processors, the non-uniformity of main memory latencies motivates the need for co-location of threads and data. We call this special form of data locality, *geographical locality,* as one aspect of the non-uniformity is the physical distance between the cc-NUMA nodes. We compare the well established first-touch strategy to an application-initiated page migration strategy as means of increasing the geographical locality for a set of important scientific applications. The main conclusions of the study are: (1) that geographical locality is important for the performance of the applications, (2) that application-initiated migration outperforms the first-touch scheme in almost all cases, and in some cases even results in performance which is close to what is obtained if all threads and data are allocated on a single node.

1 Introduction

In modern computer systems, *temporal* and *spatial locality* of data accesses is exploited by introducing a memory hierarchy with several levels of cache memories. For large multiprocessor servers, an additional form of locality also has to be taken into account. Such systems are often built as cache-coherent, non-uniform memory access (cc-NUMA) architectures, where the main memory is physically, or *geographically* distributed over several multi-processor nodes. The access time for local memory is smaller than the time required to access remote memory, and the *geographical locality* of the data influences the performance of applications.

The NUMA-ratio is defined as the ratio of the latencies for remote to local memory. Currently, the NUMA-ratio for the commonly used large cc-NUMA servers ranges from 2 to 6. If the NUMA-ratio is large, improving the geographical locality may lead to large performance improvements. This has been recognized by many researchers, and the study of geographical placement of data in cc-NUMA systems is an active research area, see e.g. [1,2,3,4].

In this paper we examine how different data placement schemes affect the performance of two important classes of parallel codes from large-scale scientific computing. The main issues considered are:

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 9–16, 2004.
© Springer-Verlag Berlin Heidelberg 2004

- What impact does geographical locality have on the performance for the type of algorithms studied?
- How does the performance of an application-initiated data migration strategy based on a migrate-on-next-touch feature compare to that of standard data placement schemes?

Most experiments presented in this paper are performed using a Sun Fire 15000 (SF15k) system, which is a commercial cc-NUMA computer. Some experiments are also performed using a Sun WildFire prototype system [5].

Algorithms with static data access patterns can achieve good geographical locality by carefully allocating the data at the nodes where it is accessed. The standard technique for creating geographical locality is based on static first-touch page allocation implemented in the operating system. In a first-touch scheme, a memory page is placed at the node where its first page fault is generated. However, the first-touch scheme also has some well known problems. In most cases, the introduction of pre-iteration loops in the application code is necessary to avoid serial initialization of the data structures, which would lead to data allocation on a single node. For complex application codes, the programming effort required to introduce these loops may be significant. For other important algorithm classes, the access pattern for the main data structures is computed in the program. In such situations it may be difficult, or even impossible, to introduce pre-iteration loops in an efficient way. Instead, some kind of dynamic page placement strategy is required, where misplacement of pages is corrected during the execution by migrating and/or replicating pages to the nodes that perform remote accesses. Dynamic strategies might be explicitly initiated by the application [2], implicitly invoked by software [6], or they may be implicitly invoked by the computer system [7,8,9].

2 Applications

To evaluate different methods for improving geographical locality we study the performance of four solvers for large-scale partial differential equation (PDE) problems. In the discretization of a PDE, a grid of computational cells is introduced. The grid may be structured or unstructured, resulting in different implementations of the algorithms and different types of data access patterns. Most algorithms for solving PDEs could be viewed as an iterative process, where the loop body consists of a (generalized) multiplication of a very large and sparse matrix by a vector containing one or a few entries per cell in the grid. When a structured grid is used, the sparsity pattern of the matrix is pre-determined and highly structured. The memory access pattern of the codes exhibit large spatial and temporal locality, and the codes are normally very efficient. For an unstructured grid, the sparsity pattern of the matrix is unstructured and determined at runtime. Here, the spatial locality is normally reduced compared to a structured grid discretization because of the more irregular access pattern.

We have noted that benchmark codes often solve simplified PDE problems using standardized algorithms, which may lead to different performance results

than for kernels from advanced application codes. We therefore perform experiments using kernels from industrial applications as well as standard benchmark codes from the NAS NPB3.0-OMP suite [10]. More details on the applications are given in [11]. All codes are written in Fortran 90, and parallelized using OpenMP. The following PDE solvers are studied:

NAS-MG. The NAS MG benchmark, size B. Solves the Poisson equation on a $256 \times 256 \times 256$ grid using a multi-grid method.

I-MG. An industrial CFD solver kernel. Solves the time-independent Euler equations describing compressible flow using an advanced discretization on a grid with $128 \times 128 \times 128$ cells. Also here a multi-grid method is used.

NAS-CG. The NAS CG benchmark, size B. Solves a sparse system of equations with an unstructured coefficient matrix using the conjugate gradient method. The system of equations has 75000 unknowns, and the sparse matrix has 13708072 non-zero elements, resulting in a non-zero density of 0.24%.

I-CG. An industrial CEM solver. Solves a system of equations with an unstructured coefficient matrix arising in the solution of the Maxwell equations around an aircraft. Again, the conjugate gradient method is used. This system of equations has 1794058 unknowns, and the non-zero density is only 0.0009%.

3 Results

On the a SF15k system, a dedicated domain consisting of four nodes was used, and the scheduling of threads to the nodes was controlled by binding the threads to Solaris processor sets. Each node contains four 900 MHz UltraSPARC-IIICu CPUs and 4 GByte of local memory. The data sets used are all approximately 500 MByte, and are easily stored in a single node. Within a node, the access time to local main memory is uniform. The nodes are connected via a crossbar interconnect, forming a cc-NUMA system. The NUMA-ratio is only approximately 2, which is small compared to other commercial cc-NUMA systems available today.

All application codes were compiled with the Sun ONE Studio 8 compilers using the flags `-fast -openmp -xtarget=ultra3cu -xarch=v9b`, and the experiments were performed using the 12/03-beta release of Solaris 9. Here, a static first-touch page placement strategy is used and support for dynamic, application-initiated migration of data is available in the form of a migrate-on-next-touch feature [12]. Migration is activated using a call to the `madvise(3C)` routine, where the operating system is advised to reset the mapping of virtual to physical addresses for a given range of memory pages, and to redo the first-touch data placement. The effect is that a page will be migrated if a thread in another node performs the next access to it.

We have also used a Sun WildFire system with two nodes for some of our experiments. Here, each node has 16 UltraSPARC-II processors running at 250 MHz. This experimental cc-NUMA computer has CPUs which are of an earlier generation, but includes an interesting dynamic and transparent page placement optimization capability. The system runs a special version of Solaris 2.6, where

pages are initially allocated using the first-touch strategy. During program execution a software daemon detects pages which have been placed in the wrong node and migrates them without any involvement from the application code. Furthermore, the system also detects pages which are used by threads in both nodes and replicates them in both nodes. A per-cache-line coherence protocol keeps coherence between the replicated cache lines.

We begin by studying the impact of geographical locality for our codes using the SF15k system. We focus on isolating the effects of the placement of data, and do not attempt to assess the more complex issue of the scalability of the codes. First, we measure the execution time for our codes using four threads on a single node. In this case, the first touch policy results in that all application data is allocated locally, and the memory access time is uniform. These timings are denoted UMA in the tables and figures. We then compare the UMA timings to the corresponding execution times when executing the codes in cc-NUMA mode, running a single thread on each of the four nodes. Here, three different data placement schemes are used:

Serial initialization (SI). The main data arrays are initialized in a serial section of the code, resulting in that the pages containing the arrays are allocated on a single node. This is a common situation when application codes are naively parallelized using OpenMP.

Parallel initialization (PI). The main data arrays are initialized in pre-iteration loops within the main parallel region. The first-touch allocation results in that the pages containing the arrays are distributed over the four nodes.

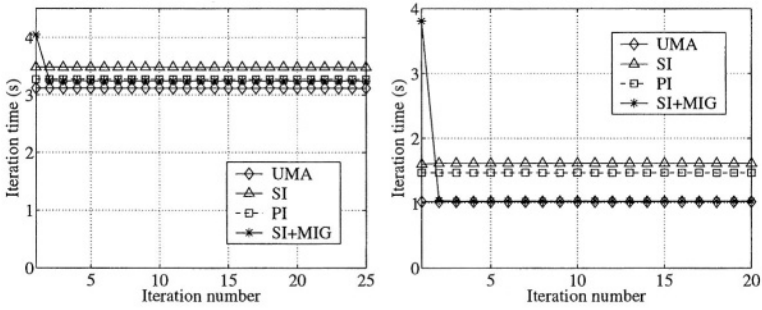
Serial initialization + Migration (SI+MIG). The main arrays are initialized using serial initialization. A migrate-on-next-touch directive is inserted at the first iteration in the algorithm. This results in that the pages containing the arrays will be migrated according to the scheduling of threads used for the main iteration loop.

In the original NAS-CG and NAS-MG benchmarks, parallel pre-iteration loops have been included [10]. The results for PI are thus obtained using the standard codes, while the results for SI are obtained by modifying the codes so that the initialization loops are performed by only one thread. In the I-CG code, the sparse matrix data is read from a file, and it is not possible to include a pre-iteration loop to successfully distribute the data over the nodes using first touch allocation. Hence, no PI results are presented for this code.

In Table 1, the timings for the different codes and data placement settings are shown. The timings are normalized to the UMA case, where the times are given also in seconds. From the results, it is clear that the geographical locality of data does affect the performance for all four codes. For the I-MG code, both the PI and the SI+MIG strategy are very successful and the performance is effectively the same as for the UMA case. This code has a very good cache hit rate, and the remote accesses produced for the SI strategy do not reduce the performance very much either. For the NAS-MG code the smaller cache hit ratio results in that this code is more sensitive to geographical misplacement of data. Also,

Table 1. Timings and fraction of remote memory accesses for the different codes and data placement settings. The timings for the cc-NUMA settings are normalized to the UMA case

Application	Time				Remote accesses			
	UMA	SI	PI	SI+MIG	UMA	SI	PI	SI+MIG
NAS-CG	1.00 (233.9s)	1.12	1.08	1.04	0.0%	75.1%	35.9%	6.2%
NAS-MG	1.00 (20.8s)	1.58	1.43	1.15	0.0%	72.4%	48.5%	11.0%
I-CG	1.00 (39.9s)	1.58	N/A	1.15	0.0%	67.9%	N/A	31.4%
I-MG	1.00 (219.1s)	1.18	1.00	1.01	0.1%	77.1%	4.3%	3.6%



(a) Execution time per iteration of NAS-CG. Only the first 25 iterations are shown in the graph

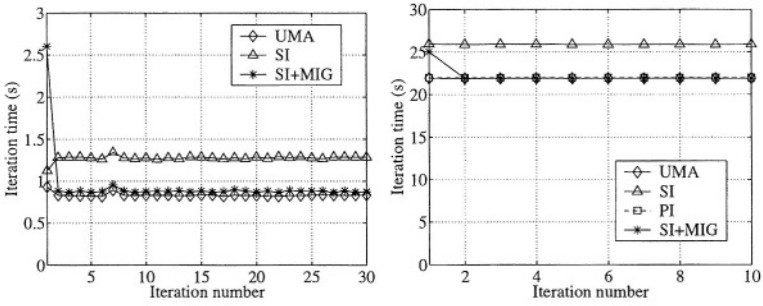
(b) Execution time per iteration of NAS-MG

Fig. 1. Execution time per iteration for NAS-CG and NAS-MG on the SF15K using 4 threads

NAS-MG contains more synchronization primitives than I-MG, which possibly affects the performance when executing in cc-NUMA mode. Note that even for the NAS-MG code, the SI+MIG scheme is more efficient than PI. This shows that sometimes it is difficult to introduce efficient pre-iteration loops also for structured problems.

For the NAS-CG code, the relatively dense matrix results in reasonable cache hit ratio and the effect of geographical misplacement is not very large. Again SI+MIG is more efficient than than PI, even though it is possible to introduce a pre-iteration loop for this unstructured problem. For I-CG, the matrix is much sparser, and the caches are not so well utilized as for NAS-CG. As remarked earlier, it is not possible to include pre-iteration loops in this code. There is a significant difference in performance between the unmodified code (SI) and the version where a migrate-on-next-touch directive is added (SI+MIG).

In the experiments, we have also used the UltraSPARC-III hardware counters to measure the number of L2 cache misses which are served by local and remote



(a) Execution time per iteration of I-CG

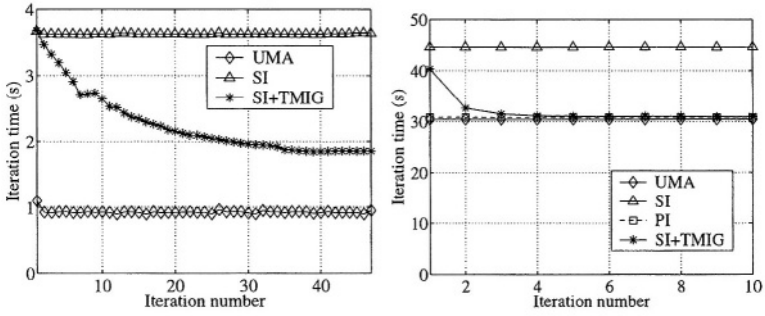
(b) Execution time per iteration of I-MG

Fig. 2. Execution time per iteration for I-CG and I-MG on the SF15K using 4 threads

memory respectively. In Table 1, the fractions of remote accesses for the different codes and data placement settings are shown. Comparing the different columns of Table 1, it is verified that that the differences in overhead between the cc-NUMA cases compared to the UMA timings is related to the fraction of remote memory accesses performed.

We now study the overhead for the dynamic migration in the SI+MIG scheme. In Figures 1(a), 1(b), 2(a), and 2(b), the execution time per iteration for the different codes and data placement settings is shown. As expected, the figures show that the overhead introduced by migration is completely attributed to the first iteration. The time required for migration varies from 0.80 s for the NAS-CG code to 3.09 s for the I-MG code. Unfortunately, we can not measure the number of pages actually migrated, and we do not attempt to explain the differences between the migration times. For the NAS-MG and I-CG codes, the migration overhead is significant compared to the time required for one iteration. If the SI+MIG scheme is used for these codes, approximately five iterations must be performed before there is any gain from migrating the data. For the NAS-CG code the relative overhead is smaller, and migration is beneficial if two iterations are performed. For the I-MG code, the relative overhead from migration is small, and using the SI+MIG scheme even the first iteration is faster than if the data is kept on a single node. A study of the scalability of the SI+MIG scheme is performed in [11].

Finally, we do a qualitative comparison of the SI+MIG strategy to the transparent, dynamic migration implemented in the Sun WildFire system. In Figures 3(a) and 3(b), we show the results for the I-CG and I-MG codes obtained using 4 threads on each of the two nodes in the WildFire system. Here, the SI+TMIG-curves represent timings obtained when migration and replication is enabled, while the SI-curves are obtained by disabling these optimizations and allocating the data at one of the nodes. Comparing the UMA- and SI-curves in Figures



(a) Execution time per iteration of I-CG

(b) Execution time per iteration of I-MG

Fig. 3. Execution time per iteration for I-CG and I-MG on the Sun WildFire using 8 threads

3(a) and 3(b) to the corresponding curves for SF15k in Figures 2(a) and 2(b), we see that the effect of geographical locality is much larger on WildFire than on SF15k. This is reasonable, since the NUMA-ratio for WildFire is approximately three times larger than for SF15k. From the figures, it is also clear that the transparent migration is active during several iterations. The reason is that, first the software daemon must detect which pages are candidates for migration, and secondly the number of pages migrated per time unit is limited by a parameter in the operating system. One important effect of this is that on the WildFire system, it is beneficial to activate migration even if very few iterations are performed.

4 Conclusions

Our results show that geographical locality is important for the performance of our applications on a modern cc-NUMA system. We also conclude that application-initiated migration leads to better performance than parallel initialization in almost all cases examined, and in some cases the performance is close to that obtained if all threads and their data reside on the same node. The main possible limitations of the validity of these results are that the applications involve only sparse, static numerical operators and that the number of nodes and threads used in our experiments are rather small.

Finally, we have also performed a qualitative comparison of the results for the commercial cc-NUMA to results obtained on a prototype cc-NUMA system, a Sun WildFire server. This system supports fully transparent adaptive memory placement optimization in the hardware, and our results show that this is also a viable alternative on cc-NUMA systems. In fact, for applications where the ac-

cess pattern changes dynamically but slowly during execution, a self-optimizing system is probably the only viable solution for improving geographical locality.

References

1. Noordergraaf, L., van der Pas, R.: Performance experiences on Sun's Wildfire prototype. In: Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM), ACM Press (1999) 38
2. Bircsak, J., Craig, P., Crowell, R., Cvetanovic, Z., Harris, J., Nelson, C.A., Offner, C.D.: Extending OpenMP for NUMA machines. Scientific Programming **8** (2000) 163-181
3. Nikolopoulos, D.S., Papatheodorou, T.S., Polychronopoulos, C.D., Labarta, J., Ayguade, E.: A transparent runtime data distribution engine for OpenMP. Scientific Programming **8** (2000) 143–162
4. Bull, J.M., Johnson, C.: Data Distribution, Migration and Replication on a cc-NUMA Architecture. In: Proceedings of the Fourth European Workshop on OpenMP, `http://www.caspur.it/ewomp2002/` (2002)
5. Hagersten, E., Koster, M.: WildFire: A Scalable Path for SMPs. In: Proceedings of the 5th International Symposium on High-Performance Architecture. (1999)
6. Nikolopoulos, D.S., Polychronopoulos, C.D., Ayguadi, E.: Scaling irregular parallel codes with minimal programming effort. In: Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM), ACM Press (2001) 16–16
7. Verghese, B., Devine, S., Gupta, A., Rosenblum, M.: Operating system support for improving data locality on CC-NUMA compute servers. In: Proceedings of the seventh international conference on Architectural support for programming languages and operating systems, ACM Press (1996) 279–289
8. Chandra, R., Devine, S., Verghese, B., Gupta, A., Rosenblum, M.: Scheduling and page migration for multiprocessor compute servers. In: Proceedings of the sixth international conference on Architectural support for programming languages and operating systems, ACM Press (1994) 12–24
9. Corbalan, J., Martorell, X., Labarta, J.: Evaluation of the memory page migration influence in the system performance: the case of the sgi o2000. In: Proceedings of the 17th annual international conference on Supercomputing, ACM Press (2003) 121–129
10. Jin, H., Frumkin, M., Yan, J.: The OpenMP Implementation of NAS Parallel Benchmarks and Its Performance. NAS Technical Report NAS-99-011, NASA Ames Research Center (1999)
11. Löf, H., Nordén, M., Holmgren, S.: Improving geographical locality of data for shared memory implementations of pde solvers. Technical Report 006, Department of Information Technology, Uppsala University (2004)
12. Sun Microsystems `http://www.sun.com/servers/wp/docs/mpo_v7_CUSTOMER.pdf`: Solaris Memory Placement Optimization and Sun Fire servers. (2003)

Cache Oblivious Matrix Transposition: Simulation and Experiment

Dimitrios Tsifakis, Alistair P. Rendell, and Peter E. Strazdins

Department of Computer Science, Australian National University
Canberra ACT0200, Australia
dtsifakis@ieee.org,
{alistair.rendell,peter.strazdins}@anu.edu.au

Abstract. A cache oblivious matrix transposition algorithm is implemented and analyzed using simulation and hardware performance counters. Contrary to its name, the cache oblivious matrix transposition algorithm is found to exhibit a complex cache behavior with a cache miss ratio that is strongly dependent on the associativity of the cache. In some circumstances the cache behavior is found to be worst than that of a naïve transposition algorithm. While the total size is an important factor in determining cache usage efficiency, the sub-block size, associativity, and cache line replacement policy are also shown to be very important.

1 Introduction

The concept of a cache oblivious algorithm (COA) was first introduced by Prokop in 1999 [1] and subsequently refined by Frigo and coworkers [2, 3]. The idea is to design an algorithm that has asymptotically optimal cache performance without building into it any explicit knowledge of the cache structure (or memory architecture) of the machine on which it is running. The basic philosophy in developing a COA is to use a recursive approach that repeatedly divides the data set until it eventually become cache resident, and therefore cache optimal. COA for matrix multiplication, matrix transposition, fast Fourier transform, funnelsort and distribution sort have been outlined (see [4] and references therein).

Although a number of COA have been proposed, to date most of the analyses have been theoretical with few studies on actual machines. An exception to this is a paper by Chatterjee and Sen (C&S) [5] on "Cache-Efficient Matrix Transposition". In this paper C&S outline a number of matrix transposition algorithms and compare their performance using both machine simulation and elapsed times recorded on a Sun UltraSPARC II based system. Their work is of interest in two respects; first their simulations showed that while the cache oblivious transposition algorithm had the smallest number of cache misses for small matrix dimensions, for large dimensions it was actually the worst. Second, their timing runs showed that in most cases the COA was significantly slower than the other transposition algorithms. It was suggested that the poor performance of the cache oblivious matrix transposition algorithm was

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 17–25, 2004.
© Springer-Verlag Berlin Heidelberg 2004

related to the associativity of the cache, although this relationship was not fully explored.

Today virtually all modern processors include a number of special registers that can be programmed to count specific events. These so called "hardware performance counters", coupled with the availability of a number of portable libraries to access them [6, 7] means that it is now possible to gather very detailed information about how a CPU is performing. Examples of the sort of events that can be counted include machine cycles, floating point operations, pipeline stalls, cache misses etc. Using these registers it is therefore possible to directly assess the performance of COA on real machines, and perform details studies comparing theoretical and observed performance. In this respect there have, very recently, appeared a number of studies looking at COA using hardware performance counters, e.g., cache oblivious priority queues [8, 9] and cache oblivious sorting [10, 11].

The primary aim of this paper is to explore further the cache oblivious matrix transposition algorithm with the aim of rationalizing the results of C&S [5]. To achieve this, a combination of machine simulation and hardware performance counters is used, and in this respect the work presented here compliments the other recent studies of COA [8–11].

2 Matrix Transposition

Matrix A of size $m \times n$ is transposed into a matrix B of size $m \times n$ such that:

$$A_{ij} = B_{ji} \quad \forall i \in [1 \cdots m], j \in [1 \cdots n]$$

Frequently the transposition occurs "in-situ", in which case the memory used for storing matrix A and B is identical. For the purpose of this paper the discussion will be restricted to square *(m=n)* in-situ matrix transpositions. Three different algorithms will be consider; cache ignorant, blocked, and cache oblivious.

2.1 Cache Ignorant Matrix Transposition

A naïve implementation of matrix transposition is given by the following C code:

```
for (i = 1; i < n; i++)
       for (j = 0; j < i; j++){
              tmp = A[j][i];
              A[i][j]=A[j][i];
              A[j][i]=tmp;       }
```

In this implementation the statements in the inner loop are executed *n(n-1)/2* times and no special care is made to use the cache efficiently.

2.2 Cache Blocked Matrix Transposition

In the cache blocked transposition algorithm the matrix is effectively divided into a checkerboard of small blocks. Two blocks that are symmetrically distributed with respect to the leading diagonal are identified and their data is copied into cache resident buffers. The buffers are then copied back into the matrix, but in transposed form. Pseudo code illustrating this algorithm is given below:

```
for (i = 0; i < n; i += size)
    for (j = 0; j < i; j += size){
        copy A[i:i+size-1][j:j+size-1] to buf1
        copy A[j:j+size-1][i:i+size-1] to buf2
        transpose buf1 to A[j:j+size-1][i:i+size-1]
        transpose buf2 to A[i:i+size-1][j:j+size-1]}
```

In the above the dimension of the small blocks is given by `size` with the restriction that $2\times$`size`^{2} is less than the cache size, and it has been assumed that `size` perfectly divides the matrix dimension `n`. In contrast to the cache ignorant scheme, each element of the matrix is now loaded into registers twice; once when copying the data from matrix `A` to `buf`, and once when copying each element from `buf` back to `A`.

2.3 Cache Oblivious Matrix Transposition

In the cache oblivious transposition the largest dimension of the matrix is identified and split, creating two sub-matrices. Thus if $n \geq m$ the matrices are partitioned as:

$$A = (A_1 A_2), \quad B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}$$

This process continues recursively until individual elements of A and B are obtained at which point they are swapped.

3 Performance Simulation

To analyse performance a basic cache simulator was written. This assumes a single level of cache, and includes parameters for the cache line size, the number of cache lines, the associativity, and the cache line replacement policy. Code to perform the different matrix transposition algorithms was written and annotated such that the memory address corresponding to every matrix element access was passed to the cache simulator, which then determined whether it was either a cache hit or miss.

When simulating the cache, a number of other issues also need to be considered; notably the initial alignment of the matrix with respect to the cache, the word size of each matrix element, and the dimension of the matrix. For simplicity in the following experiments the first element of the matrix is always aligned perfectly with the start of a cache line, the cache line size is a perfect multiple of the matrix element word size,

and the matrix dimensions are chosen such that different rows of the matrix never share the same cache line.

Before considering the results of the simulator experiments, it is useful to illustrate the typical access patterns of the three matrix transposition algorithms. This is shown in Fig. 1. Of particular interest is the COA. This clearly shows a natural partitioning of the matrix into a hierarchy of square blocks of dimensions 2^x. Thus if the cache line size was sufficient to hold exactly 4 matrix elements and the total cache size was sufficient to hold 8 cache lines, then both of the shaded blocks shown in Fig. 1.c could, in principle, reside in cache simultaneously and the algorithm would therefore be expected to show minimal cache misses.

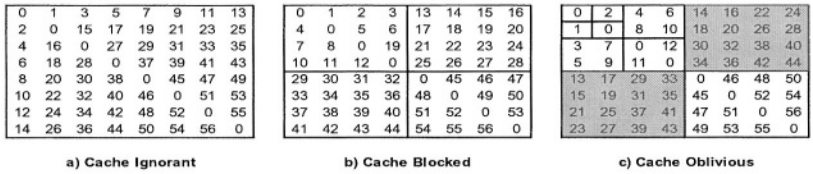


Fig. 1. Typical access patterns for the three transposition algorithms on an 8×8 matrix (A blocking size of 4 is used in the cache blocked algorithm)

In their paper C&S [5] presented a table of cache misses for a variety of different matrix transpositions algorithms and for four different matrix sizes. Their simulated results for the cache ignorant, cache blocked (full copy), and COA are reproduced in table 1. The strange behavior of the COA is immediately obvious; for N=1024 it has the lowest number of cache misses, while for N=8192 it has the largest.

Table 1. Cache misses for three matrix transposition algorithms. Data taken from C&S [5] and obtained by simulating a 16KB direct mapped cache with a 32byte cache line. The matrix is square with a 4byte word size

	Matrix Dimension			
Algorithm	**1024**	**2048**	**4096**	**8192**
Cache ignorant	589795	2362002	9453724	37826712
Full copy cache blocked	275550	1170003	4804808	19493808
Cache oblivious	131226	923295	7101600	56158873

In Fig. 2, the simulations of C&S [5] have been extended to include all matrix dimensions that are less than 10,000 but that are multiples of the cache line size. The figure includes data for the cache ignorant and COA, and also the minimum and maximum number of cache misses. The minimum cache miss ratio assumes all data in a cache line is fully utilized before that cache line is evicted, while the maximum cache miss ratio assumes a cache miss occurs for every read, but the subsequent write is a cache hit. Assuming there are no cache line conflicts between the temporary buffers and the matrix elements then the cache blocked algorithm will essentially give the minimum number of cache misses.

From Fig. 2, it is apparent that the COA is far from cache oblivious. Rather, the cache miss profile shows significant structure. Furthermore the data points chosen by C&S (N=1024, 2048, 4096 and 8192) [5] are actually some of the worst possible values; for many other dimensions the COA achieves close to the minimum.

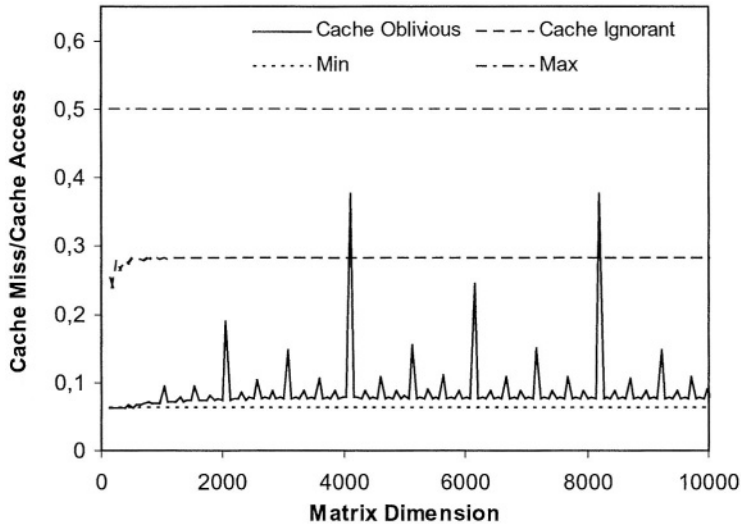


Fig. 2. Simulated cache miss to access ratio for cache oblivious and cache ignorant matrix transposition algorithms, using a 16KB, direct mapped cache with a 32byte line size and 4byte matrix elements. Matrix dimensions are always an exact multiple of the cache line size

The poor performance of the COA for N=4096 and 8192 is due to the fact that for both of these dimensions one row of the matrix is an exact multiple of the cache size. With a direct mapped cache this means that elements in the same column of the matrix map to the same cache line. Inspecting the access pattern for the COA given in Fig. 1 clearly shows that this will be a problem. For example, if a cache line is assumed to hold 4 matrix elements and the matrix is aligned such that accesses {13, 17, 29, 33} correspond to one cache line, then to fully utilize the data in this cache line there must be no cache line conflicts between accesses 13 and 33. However, between these accesses 7 other cache lines will be accessed – corresponding to accesses {15,19,31,35}, {21,25,37,41} {23,27,39,43}, {14,16,22,24}, {18,20,26,28}, {30,32,38,40}, and {34,36,42,44}. The first three of these share the same cache line as the initial access, while the latter 4 will share another cache line. Changing the matrix row size to be, e.g., 1.5 times the cache size will halve the number of cache line conflicts, but will not totally eliminate then. Similar effects occur for other partial multiples giving the complicated structure shown in Fig. 2.

From the above discussion increasing cache line associativity should lead to a decrease in the number of cache line conflicts. This is demonstrated by the simulated results in Fig. 3. It is interesting to note, however, that the reduction in cache misses is not universal for all matrix dimensions. Thus while the cache miss ratio for N=4096 and 8192 decreases in going from a direct to 2-way set associative cache, the cache miss ratio for N=6144 actually increases slightly. This effect is due to the fact that

increasing the cache line associativity while maintaining the same total cache size actually doubles the number of possible cache line conflicts, although also providing two possible cache lines that can be used to resolve each conflict. For example, whereas with the direct mapped cache, a cache line conflict was encountered every 4096 matrix elements and could not be avoided, with a 2-way set associative cache a conflict arises every 2048 elements but there are two possible cache line locations that can be used to remove those conflicts. Thus predicting the overall effect of increasing cache line associativity is hard, although it appears beneficial overall.

Interestingly with an 8-way set associative cache, the data points that originally gave rise to the worst cache miss ratio, i.e. N=4096 and 8192, now give rise to the minimum. This is evident in Fig. 3 as slight dips in the cache miss ratios for these data points. The existence of "magic dimensions" is not surprising; with a 4-way associative cache the cache line conflict discussed above for accesses {13,17,29,33}, {15,19,31,35}, {21,25,37,41} and {23,27,39,43} would be removed. If these assesses also conflicted with those of {14,16,22,24}, {18,20,26,28}, {30,32,38,40}, and {34, 36,42,44}, an 8-way set associative cache would be required to remove the conflict.
This result can be generalized for a cache whose line size (l) is a power of 2. Assuming that each matrix row starts with a new line, a COA will attain minimum misses if its associativity is at least 2^l. This is because it will reach a stage where it will swap two $l \times l$ blocks, which will be stored in $2 \times l$ lines. Providing a least recently used (LRU) replacement policy is used, the cache will be able to hold all of these simultaneously. If matrix rows are not aligned with cache lines, the two sub-blocks will be stored in at most $4 \times l$ lines; in this case, an associativity of $4 \times l$ would be required in order to minimize cache misses.

4 Performance Measurements

Using hardware performance counters cache miss data was gathered for:

- A 167MHz Sun UltraSPARC I system with a 16KB direct mapped L1 data cache with 32-byte cache line and a 512KB level 2 cache
- A 750MHz Sun UltraSPARC III system with a 64KB 4-way set associative L1 data cache with a 32-byte cache line size and an 8MB level 2 cache

The Sun UltraSPARC I system has a direct mapped level 1 cache with identical structure to that used by C&S [5]. The measured and simulated cache misses for the COA are given in table 2. The matrix elements are 4 bytes, with data given for dimensions around N=4096 and 8192. Two different simulated results are shown; for Sim#1 the cache line size is 32bytes while for Sim#2 it is 16bytes. This is done since the 32byte Ultra SPARC I cache line is actually split into two 16byte sub-blocks, and halving the cache line size in the simulated results is an attempt to approximately (but not exactly) account for this effect.

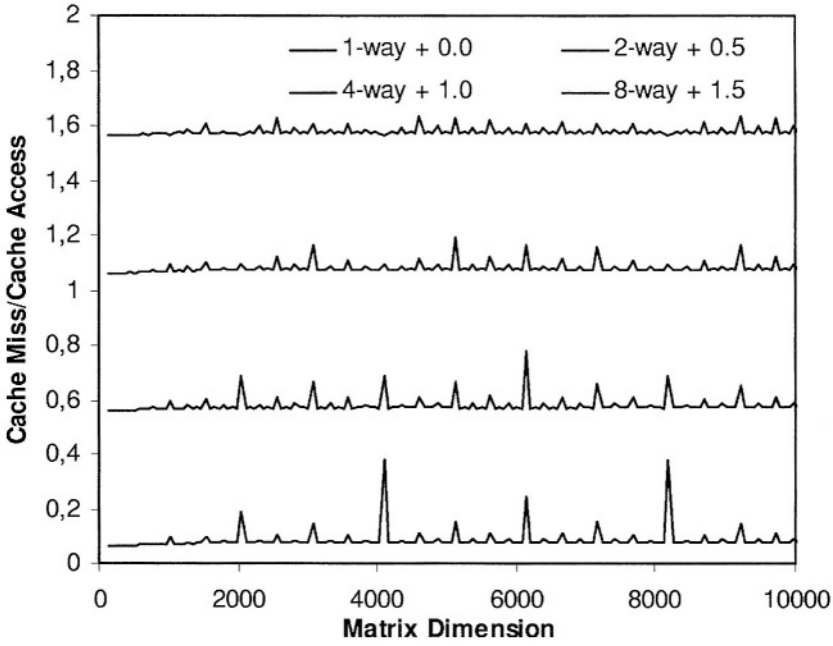


Fig. 3. Simulated cache miss to access ratio as a function of cache line associtivity for the cache oblivious matrix transposition algorithms using a 16KB cache with a 32byte line size and 4byte matrix elements. Matrix dimensions are chosen to be a direct multiple of the cache line size

Table 2. Measured and simulated cache misses on the Ultra SPARC I for square in-situ COA and a variety of matrix dimensions (N). Simulated results reported with both 32byte (Sim#1) and 16byte (Sim#2) cache line size

	Cache Misses				Cache Misses		
N	Ultra I	Sim#1	Sim#2	N	Ultra I	Sim#1	Sim#2
4080	5270060	2391376	4379033	8176	21408816	9646976	17621793
4088	5199068	2316901	4331751	8184	20438710	9301155	17393199
4096	12997199	12615680	12595200	8192	52589636	50479104	50397184
4104	6857957	4176906	4849009	8200	27268615	16677803	19370810
4112	5499487	2550567	4543409	8208	21899111	10117444	18110294

The results as measured by the hardware performance counters clearly show a large number of cache misses at N=4096 and 8192, that decreases markedly for matrix dimensions that are either slightly smaller or larger. At these dimensions both the experimental and simulated results are approximately identical – reflecting the fact that essentially every matrix access results in a cache miss. For other dimensions the simulated results obtained using a 16byte cache line are closest to the experimentally recorded results, with the experimental results showing slightly higher number of cache misses. This is to be expected since the simulated results with a 16kbyte cache and a 16byte cache line has twice the number of cache lines as a 16kbyte cache with a sub-blocked 32byte cache line and is therefore a more flexible cache model. It should

also be noted that the results from the hardware counters show some sensitivity to the choice of compilation flags; the above results were obtained using the `-fast` option and if this is lowered to `-x01` the agreement between the measured and simulated number of cache misses actually improves slightly.

Table 3. Cache misses on an UltraSPARC III system. Simulated results using LRU (Sim#LRU) and random (Sim#Ran) cache replacement policy for the square in-situ COA and a variety of different matrix dimensions (N).

	Cache Misses				Cache Misses		
N	Ultra III	Sim#LRU	Sim#Ran	N	Ultra III	Sim#LRU	Sim#Ran
1000	263477	258290	265990	4072	4361124	4283628	4391355
1024	375028	262983	284788	4096	7751760	4232722	5973761
1048	289398	284628	292332	4120	4464945	4382333	4496696
2024	1075296	1058128	1083519	8168	17577433	17234628	17669072
2048	1923256	1056544	1491917	8192	30873556	16956716	23904911
2072	1128147	1108024	1136952	8216	17791597	17425642	17882604

In table 3, similar cache miss data is given for the Ultra SPARC III platform. On this system there is a 4-way set associative level 1 cache. From the results given in section 3, it might be expected that there would be little difference between the number of cache misses that occurs for N=4096 or 8192 and surrounding values of N. The experimental results show, however, that this not the case; rather, the number of cache misses is roughly double at these values of N compared to those at nearby values of N. This is due to the cache line replacement policy on the UltraSPARC III, which is pseudo random rather than LRU [12]. Simulated results using a random number generator to determine cache line placement are shown as "Sim#Ran" in table 4. These show a considerable increase in the number of cache misses when N=1024, 2048, 4096 and 8192, although still somewhat less than those recorded by the hardware performance counters. Outside these data points there appears to be little difference between the use of an LRU or random cache line replacement policy.

5 Conclusions

The performance of a COA for matrix transposition has be analyzed, with respect to cache misses, via both simulation and use of hardware performance counters on two fundamentally different UltraSPARC systems. The results confirm earlier work by C&S [5] showing very high numbers of cache misses for certain critical matrix dimensions. In general it was shown that the cache miss characteristics of the "cache oblivious" matrix transposition algorithm has significant structure, the form of which depends on a subtle interplay between cache size, matrix dimension, number of matrix elements per cache line, cache line size, cache associativity and the cache line replacement policy. Predicting, a priori, when the COA will perform well and when it will perform poorly is non-trivial, although increased cache line associativity appears overall to be beneficial.

The work presented here has only been concerned with the cache usage characteristics of cache oblivious matrix transposition. The observed performance of any algorithm is of course dependent on other factors as well as efficient cache usage. Details of this will be discussed in a subsequent publication.

Acknowledgements. APR and PES acknowledge support from Australian Research Council Linkage Grant LP0347178 and Sun Microsystems. Discussions with Bill Clarke and Andrew Over are also gratefully acknowledged.

References

1. H. Prokop, *Cache-Oblivious Algoirthms,* MSc Thesis, Dept. Electrical Eng. and Computer Science, Massachusetts Institute of Technology, 1999
2. M. Frigo, C. Leiserson, H. Prokop, and S. Ramachandran, *Cache-Oblivious Algoirthms (extended abstract),* Proceedings of the 40th Annual Symposium on Foundations of Computer Science, IEEE Computer Science Press, 285-297, 1999.
3. M. Frigo, *Portable High Performance Programs,* PhD Thesis, Dept. Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1999.
4. E.D. Demaine, *"Cache-Oblivious Algoirthms and Data Structures",* Lecture notes in Computer Science, BRICS, University of Aarhus, Denmark June 27-July 1, 2002.
5. S. Chatterjee and S. Sen, *Cache-Efficient Matrix Transposition,* Proceedings of the 6th International Conference on High Performance Computing Architecture, 195, 2000
6. Performance Application Programmer Interface (PAPI) http://icl.cs.utk.edu/ projects/papi
7. Performance Counter Library (PCL), http://www.fz-juelich.de/zam/PCL
8. J.H. Olsen and S.C. Skov, *Cache-Oblivious Algoritsms in Practice,* MSc Thesis, Dept Computing, University of Copenhagen, 2002
9. L. Arge, M. Bender, E. Demaine, B. Holland-Minkley and J. Munro, *Cache-Oblivious Priority Queue and Graph Algorithhm Applications,* Submitted to SIAM journal on Computing, May 2003.
10. F. Rønn, *Cache-Oblivious Searching and Sorting,* MSc thesis, Dept Computer Science, University of Copenhagen, July 2003.
11. K. Vinther, *Engineering Cache-Oblivious Sorting Algoirthms,* MSc Thesis, Dept. Computer Science, University of Aarhus, June 2003.
12. D. May, R. Pas and E. Loh, *The RWTH SunFire SMP-Cluster User's Guide (version 3.1),* http://www.rz.rwth-aachen.de/computing/info/sun/primer/ primer_V3.1 .html, July 2003

An Intelligent Hybrid Algorithm for Solving Non-linear Polynomial Systems*

Jiwei Xue[1,2], Yaohui Li[1], Yong Feng[1], Lu Yang[1], and Zhong Liu[1]

[1] Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu 610041, P. R. China
xuejiwei@163.com, {phillip138, mathyfeng}@hotmail.com
[2] Computer Science and Engineering College, Daqing Petroleum Institute, Daqing 163318, P. R. China

Abstract. We give a hybrid algorithm for solving non-linear polynomial systems. It is based on a branch-and-prune algorithm, combined with classical numerical methods, symbolic methods and interval methods. For some kinds of problems, Gather-and-Sift method, a symbolic method proposed by L. Yang, was used to reduce the dependency of variables or occurrences of the same variable, then interval methods were used to isolate the real roots. Besides these, there are some intelligent judgments which can improve the system's efficiency significantly. The algorithm presented here works rather efficiently for some kinds of tests.

1 Introduction

In this paper, we address the problem of finding all solutions to polynomial systems, a fundamental and important problem in the research of real algebra from the viewpoint of algorithm research. This is an old problem and there have been some works concerning this issue. Several interesting methods have been proposed in the past for this task, including two fundamentally different methods: numerical methods[4,5,6,9,10, 11,15] and symbolic methods[1,2,20,21, 22,23,24].

Classical numerical methods start from some approximate trial points and iterate. Thus, there is no way to guarantee correctness (i.e. finding all solutions) and to ensure termination. Interval methods can overcome these two shortcomings but tend to be slow.

Symbolic computation plays important role in applied mathematics, physics, engineering and other areas. But currently it is only possible to solve small examples, because of the inherent complexity of the problems in symbolic computation. Symbolic methods include Ritt-Wu method, Gröbner basis methods or resultant methods [3], but all these methods are time consuming, especially when the number of variables > 10.

In order to improve the efficiency of the system, we propose an *intelligent hybrid* algorithm. *Hybrid* means we combine numerical methods, interval methods

* This research was partially supported by NSFC (10172028).

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 26–33, 2004.
© Springer-Verlag Berlin Heidelberg 2004

and symbolic methods. *Intelligence* means before using interval methods we will use our knowledge to tight the starting box, or use classical numerical methods to approximate the root directly once we can ensure that only one root exists.

The rest of this paper is structured as follows: *Gather-and-Sift* algorithm and some improvements are presented in section 2. Section 3 devotes to univariate and multivariate interval Newton methods. Section 4 presents some of improvements made in our method in order to improve the efficiency. Some examples and their results are given in Section 5. Section 6 concludes the paper.

In this paper, boldface (e.g. $\boldsymbol{x}$, $\boldsymbol{y}$) will denote intervals, lower case (e.g. x, y) will denote scalar quantities, and upper case (e.g. A, B) will denote vectors or matrices, bold upper case (e.g. $\boldsymbol{A}$, $\boldsymbol{B}$) will denote interval vectors (or *boxes*). Brackets "[ ]" will delimit intervals. Underscores $\underline{x}$ will denote lower bounds of intervals and overscores $\bar{x}$ will denote upper bounds of intervals. The set of real intervals will be denoted by $I\!R$.

2 Gather-and-Sift Algorithm

Gather-and-Sift algorithm[23,24], which was proposed by L. Yang et al. in 1995, is a very efficient method in solving nonlinear algebraic equation system both of parametric coefficients and of numeric coefficients. *Gather* means to construct some ascending chains whose zeros contain all the zeros of the original systems. *Sift* means to remove extra zeros from ascending chains such that only the required ones remain. GAS, a MAPLE program based on DIXON resultant, will be called before interval methods are used if the number of variables ≤ 4 in our system. The effect of this modification can be seen from *Example 1*. We will give a sketch of *Gather-and-Sift* method and modifications have been made, for details and further references about *Gather-and-Sift* see [13,23,24].

2.1 A Sketch of *Gather-and-Sift*

Given a system PS consisting of k polynomials in k indeterminates, *Gather-and-Sift* algorithm can be summarized as follows:

Step 1. Regarding x_1 as a parameter, construct a polynomial system *DPS,* which is the *Dixon derived polynomial set* of *PS* with respect to $\{x_2, \cdots, x_k\}$;

Step 2. Transform *DPS* into the following *standard form* (x_1 was regarded as a parameter):

$$DPS: \begin{cases} q_1 = c_{1n}e_n + \cdots + c_{12}e_2 + c_{11}e_1 = 0, \\ q_2 = c_{2n}e_n + \cdots + c_{22}e_2 + c_{21}e_1 = 0, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ q_m = c_{mn}e_n + \cdots + c_{m2}e_2 + c_{m1}e_1 = 0 \end{cases} \tag{1}$$

where $e_n, \cdots, e_1$ represent all the power products of $x_k, x_{k-1}, \cdots, x_2$ appeared in *DPS* sorted into a decreasing order according to a lexicographical order or a degree order;

Step 3. Do a fraction-free Gaussian elimination for the above system *DPS,* which is a linear equation system in $e_n, \cdots, e_1$, then we have:

$$GPS: \begin{cases} h_m = b_{mn}e_n + \cdots + b_{m2}e_2 + b_{m1}e_1 = 0, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ h_2 = b_{22}e_2 + b_{21}e_1 = 0, \\ h_1 = b_{11}e_1 = 0 \end{cases} \tag{2}$$

where $b_{ij} \in K[x_1](i = 1, 2, \cdots, m; j = 1, 2, \cdots, n)$. Now regard x_1 as a indeterminate, *GPS* will be written as follows:

$$GPS: \begin{cases} h_1 = h_1(x_1, x_2, \cdots, x_k) = 0, \\ h_2 = h_2(x_1, x_2, \cdots, x_k) = 0, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ h_m = h_m(x_1, x_2, \cdots, x_k) = 0 \end{cases} \tag{3}$$

The above three steps is called GPS algorithm, and a generic program for this algorithm written in Maple was called GPS program.

Step 4. Select k polynomials from *GPS* to form a *triangular form TS* in $\{x_1, x_2, \cdots, x_k\}$;

Step 5. Establish normal ascending chain *ASC* (or normal ascending chains $ASC_1, \cdots, ASC_l$) from the system *TS* resulting from last step;

Step 6. For every normal ascending chain *ASC,* do *relatively simplicial decomposition* w.r.t. *PS* by using WR method. This step can sift out the extra zeros.

2.2 Some Improvements on the *Gather-and-Sift* Method

From the above algorithm we can see that if a *triangular form* in k indeterminates cannot be found, the efficiency of the *Gather-and-Sift* method will be reduced greatly. The following two improvements have been made to increase the possibility of finding the *triangular form.*

Unknown-Order-Change. By calling GPS program, the output polynomial set GPS has great difference in form if the given sequence of the indeterminates is different. Sometimes, you even cannot find a *triangular form* in all variables from GPS directly. The possibility of finding the *triangular form* can be increased by Z. Liu's method, i.e., unknown-order-change. It can be described as follows:

```
BEGIN
   FOR i FROM 1 TO n DO
      regard the ith arrange of the given indeterminates as the current sequence;
      regard the first element of current sequence as parameter and call GPS;
      select a triangular form TS in all indeterminates from GPS;
      IF such TS can be found THEN
         return TS;
      END IF;
   ENDDO
END
```

Extension of The Polynomial Set. By experiments we also found such a fact: sometimes a *triangular form TS* cannot be found just because of shortage of polynomial in some indeterminates in GPS, but such a polynomial can easily be found in the original polynomial set.

Z. Liu proposed an extension of the polynomial set method to further increase the possibility of finding the *triangular form.* The method can be summarized briefly as: after running the GPS algorithm, add the original polynomial set *PS* into GPS and the result still be denoted by GPS, then try to select a *triangular form* in all indeterminates from GPS. If succeed, output the *TS,* otherwise try to use the next arrange of the indeterminates to redo the above steps.

3 Interval Newton Methods

Modern development of interval arithmetic began with R. E. Moore's dissertation in 1962. Since then thousands of research articles and numerous books have appeared on the subject. For details and further references about interval arithmetic, see[4,5,6,7,9,10,11,14,15,16,18,19].

The classical Newton method does not mathematically guarantee to find *all* roots within a given domain. Computational result obtained by finite precision arithmetics may not be reliable both mathematically and computationally. To overcome these problems, extensive studies on interval Newton methods e.g. [8, 10,12,16,17,19] have been done.

Interval Newton methods combine the *classical Newton method,* the *mean value theorem* and *interval analysis.* These methods may be used both to discard root free subintervals, and to replace subintervals by smaller ones via a rapidly converging iteration scheme.

3.1 Univariate Interval Newton Methods

Suppose $f : \boldsymbol{x} = [\underline{x}, \bar{x}] \to R$ has a continuous first derivative on $\boldsymbol{x}$, suppose that there exists $x^* \in \boldsymbol{x}$ such that $f(x^*) = 0$, and suppose that $\check{x} \in \boldsymbol{x}$. Then, since the *mean value theorem* implies $0 = f(x^*) = f(\check{x}) + f'(\xi)(x^* - \check{x})$ we have $x^* = \check{x} - f(\check{x})/f'(\xi)$ for some $\xi \in \boldsymbol{x}$. If $\boldsymbol{f}'(\boldsymbol{x})$ is any interval extension of the derivative of f over $\boldsymbol{x}$, then

$$x^* \in \check{x} - f(\check{x})/\boldsymbol{f}'(\boldsymbol{x}) \text{ for any } x^* \in \boldsymbol{x}. \tag{4}$$

From equation (4) we can get the *univariate interval Newton operator:*

$$\boldsymbol{N}(f, \boldsymbol{x}, \check{x}) = \check{x} - f(\check{x})/\boldsymbol{f}'(\boldsymbol{x}) \tag{5}$$

It is well known that $\boldsymbol{N}(f, \boldsymbol{x}, \check{x})$ has the following properties:

1. If $x^* \in \boldsymbol{x}$ and $f(x^*) = 0$, then $x^* \in \boldsymbol{N}(f, \boldsymbol{x}, \check{x})$;
2. If $\boldsymbol{x} \cap \boldsymbol{N}(f, \boldsymbol{x}, \check{x}) = \emptyset$, then $\forall x \in \boldsymbol{x}$, $f(x) \neq 0$;
3. If $\boldsymbol{N}(f, \boldsymbol{x}, \check{x}) \subset \boldsymbol{x}$, then $\exists x^* \in \boldsymbol{x}$, $f(x^*) = 0$.

3.2 Multivariate Interval Newton Methods

Multivariate interval Newton methods are analogous to univariate ones, the iteration step is as follows:

$$\begin{cases} \boldsymbol{F}'(\boldsymbol{X}_k)(\tilde{\boldsymbol{X}}_k - X_k) = -F(X_k) \\ \boldsymbol{X}_{k+1} = \boldsymbol{X}_k \cap \tilde{\boldsymbol{X}}_k \end{cases} \tag{6}$$

where $k=0,1,2,\cdots$, $\boldsymbol{F}'(\boldsymbol{X}_k)$ is a suitable interval extension of the Jacobian matrix over the box $\boldsymbol{X}_k$ (with $\boldsymbol{X}_0 = \boldsymbol{X}$), and where $X_k \in \boldsymbol{X}_k$ represents a predictor or initial guess point.

4 Some Improvements Made in Our Method Besides *Gather-and-Sift*

4.1 Intelligence+Numerical Method+Interval Arithmetic

The classical numerical method's disadvantages include: incorrectness (i.e. finding all solutions) and unreliability. For some applications each variable's degree is *one,* which means that there is only one solution to the equation. So before use *Interval Newton methods,* first judge if the equation only has one solution by collecting the maximal degree of the variables appeared in the polynomial systems. If the maximal degree is equal to one, then we can use the following method to isolate the root. Because there is only one solution so the correctness can be guaranteed, while numerical reliability is obtained by using interval arithmetic. From *Example 2,* it can be seen that this intelligent method can greatly improve the system's performance.

4.2 Numerical Method+Interval Arithmetic

As we have realized that *classical Newton method* can be made more efficient if the initial value was chosen close to the root. This is also true to *interval Newton methods.* Until now, most people choose the midpoint of the interval as the initial value, for lots of tests, this will need many times of *Interval Newton* iteration. In our method, after knowing that there is a root in a certain interval, we will use an ordinary numerical method (e.g. *classical Newton method, classical Quasi-Newton method*) to compute the approximation to the root, then use *Interval Newton method* to bound the *error.*

5 Examples and Results

In this section we report the performance of our method on some examples. All results were obtained by running our system on a PC (Pentium 566MHz CPU, 256Mb of main memory) with Maple 9.

Example 1. The system $\begin{cases} x_1^2 + x_2^2 - 1 = 0 \\ x_1^2 - x_2 = 0 \end{cases}$ can be found in many papers. Given initial interval vector $[-10^8, 10^8]^2$, tolerance $\varepsilon = 10^{-8}$, the comparison results without and with calling GAS are given as follows:

1. Without calling GAS, the following two intervals are achieved after 0.641s:

[[-0.7861513793, -0.7861513766], [0.6180339869, 0.6180339912]]
[[0.7861513766, 0.7861513793], [0.6180339869, 0.6180339912]]

2. While if GAS is called firstly, after 0.010s it gives the following result:

$$\begin{cases} -x_1^2 + 1 - x_1^4 = 0 \\ x_1^2 - x_2 = 0 \end{cases}$$

Then do as without calling GAS, we will get the following result after another 0.100s, i. e., it will cost 0.110s totally to do the same task.

[[-0.7861513779, -0.7861513776], [0.6180339885, 0.6180339890]]
[[0.7861513776, 0.7861513779], [0.6180339885, 0.6180339890]]

Example 2. The following system is an examples given by Moore and Jones [6].

$$\begin{cases} x_1 - 0.25428722 - 0.18324757x_4x_3x_9 = 0 \\ x_2 - 0.37842197 - 0.16275449x_1x_{10}x_6 = 0 \\ x_3 - 0.27162577 - 0.16955071x_1x_2x_{10} = 0 \\ x_4 - 0.19807914 - 0.15585316x_7x_1x_6 = 0 \\ x_5 - 0.44166728 - 0.19950920x_7x_6x_3 = 0 \\ x_6 - 0.14654113 - 0.18922793x_8x_5x_{10} = 0 \\ x_7 - 0.42937161 - 0.21180486x_2x_5x_8 = 0 \\ x_8 - 0.07056438 - 0.17081208x_1x_6x_7 = 0 \\ x_9 - 0.34504906 - 0.19612740x_6x_8x_{10} = 0 \\ x_{10} - 0.42651102 - 0.21466544x_1x_4x_8 = 0 \end{cases}$$

Given starting box $[-1, 1]^{10}$ and tolerance $\varepsilon = 10^{-8}$, it costs 0.180s gives the following interval as the result: $[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}] =$

[[0.2578333932, 0.2578333943], [0.3810971543, 0.3810971550],
[0.2787450173, 0.2787450174], [0.2006689638, 0.2006689647],
[0.4452514243, 0.4452514254], [0.1491839199, 0.1491839201],
[0.4320096989, 0.4320096991], [0.07340277776, 0.07340277779],
[0.3459668268, 0.3459668269], [0.4273262759, 0.4273262760]]

Without changing the tolerance, if the starting box was taken as $[-10, 10]^{10}$, the traditional method (without intelligence) does not terminate after running 7200s. But by running the intelligent analyzing module in our method, we know that if this equation has root in the given interval it will only has one root. So we can use numerical method combined with interval method to get the result interval, it only costs 0.190s. Furthermore, if GAS is called firstly, it does not terminate after running 3600s.

From *Example 1* and *Example 2,* we can get the following conclusion: Hybrid method without any consideration sometimes will not improve the systems's efficiency, contrarily, it may make the situation even worse.

Example 3. This is another standard benchmark given by Moore and Jones [5].

$$\begin{cases} x_1 - 0.25428722 - 0.18324757x_4^3x_3^3x_9^3 + x_3^4x_9^7 = 0 \\ x_2 - 0.37842197 - 0.16275449x_1^3x_{10}^3x_6^3 + x_{10}^4x_6^7 = 0 \\ x_3 - 0.27162577 - 0.16955071x_1^3x_2^3x_{10}^3 + x_2^4x_{10}^7 = 0 \\ x_4 - 0.19807914 - 0.15585316x_7^3x_1^3x_6^3 + x_1^4x_6^7 = 0 \\ x_5 - 0.44166728 - 0.19950920x_7^3x_6^3x_3^3 + x_6^4x_3^7 = 0 \\ x_6 - 0.14654113 - 0.18922793x_8^3x_5^3x_{10}^3 + x_5^4x_{10}^7 = 0 \\ x_7 - 0.42937161 - 0.21180486x_2^3x_5^3x_8^3 + x_5^4x_8^7 = 0 \\ x_8 - 0.07056438 - 0.17081208x_1^3x_6^3x_7^3 + x_7^4x_6^7 = 0 \\ x_9 - 0.34504906 - 0.19612740x_6^3x_8^3x_{10}^3 + x_6^4x_8^7 = 0 \\ x_{10} - 0.42651102 - 0.21466544x_1^3x_4^3x_8^3 + x_8^4x_1^7 = 0 \end{cases}$$

Given starting box $[-1, 1]^{10}$ and tolerance $\varepsilon = 10^{-8}$, it costs 0.381s gives the following interval as the result: $[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}] =$
$[[0.2542852239, 0.2542852240], [0.3784225742, 0.3784225743],$
$[0.2715848389, 0.2715848390], [0.1980797710, 0.1980797711],$
$[0.4416682234, 0.4416682235], [0.1464438740, 0.1464438741],$
$[0.4293719571, 0.4293719572], [0.07056502913, 0.07056502914],$
$[0.3450490767, 0.3450490768], [0.4265110279, 0.4265110280]]$

6 Conclusion and Future Work

In this paper, we have studied a hybrid method for isolating real solutions of polynomial systems. On the one hand, we use interval Newton methods in conjunction with bisection methods to overcome classical numerical methods' shortcomings; on the other hand, we use classical numerical methods to remedy interval methods' deficiency (i. e., slow). But there are also some problems deserve further study.

1. We use classical Quasi-Newton method, a superlinear convergence method, to approximate the root. Next, we can use some high-order convergence methods to further increase the efficiency of the algorithm.

2. It is computationally very expensive for polynomials with multiple occurrences of the same variables. Next, we will use more symbolic methods (e.g., Gröbner basis, Wu-method) to reduce the dependency of variables or occurrences of the same variables. But it is well known that all symbolic methods are time consuming (e.g., *Example 2*), so we must further study how to cooperate different methods and the extent of cooperation.

3. We will further study human knowledge which can be used in our method to increase the system's performance.

References

1. Collins, G.E., Loos, R.: Real Zeros of Polynomials. Computer Algebra:Symbolic and Algebraic Computation (1983)

2. Collins, G.E., Johnson, J.R., Krandick, W.: Interval Arithmetic in Cylindrical Algebraic Decomposition. Journal of Symbolic Computation. 34 (2002) 145-157
3. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms. Springer-Verlag, New York, USA (1992)
4. Hentenryck, P.V., Michel, L., Benhamou, F.: Newton: Constraint Programming over Nonlinear Constraints. Science of Computer Programing. 30(1-2)(1998) 83-118
5. Hentenryck, P.V., McAllester, D., Kapur, D.: Solving Polynomial Systems Using a Branch and Prune Approach. SIAM Journal on Numerical Analysis. 34(2)(1997) 797-827
6. Herbort, S., Ratz, D.: Improving the Efficiency of a Nonlinear Systems Solver Using a Componentwise Newton Method. http://citeseer.nj.nec.com/herbort97improving.html (1997)
7. Hickey, T., Ju, Q., van Emden, M.H.: Interval Arithmetic: From Principles to Implementation. Journal of ACM. 48(5) (2001) 1038-1068
8. Hu, C.Y.: Reliable Computing with Interval Arithmetic. Proceeding of the International Workshop on Computational Science and Engineering '97.
9. Kearfott, R.B., Hu, C.Y., Novoa III, M.: A Review of Preconditioners for the Interval Gauss-Seidel Method. Interval Computations. 1(1)(1991) 59-85
10. Kearfott, R.B.: Interval Computations: Introduction, Uses and Resources. Euromath Bulletin. 2(1)(1996) 95-112
11. Kearfott, R.B., Shi, X.F.: Optimal Preconditioners for Interval Gauss-Seidel Methods. Scientific Computing and Validated Numerics, Akademie Verlag (1996) 173-178
12. Kearfott, R.B., Walster, G.W.: Symbolic Preconditioning with Taylor Models: Some Examples. Reliable Computing. 8(6)(2002) 453-468
13. Liu, Z.: Gather-and-Sift Software GAS Based on DIXON Resultant. Chengdu Institute of Computer Applications, Chinese Academy of Sciences (2003)(Dissertation)
14. Moore, R.E., Yang, C.T.: Interval Analysis I. (1959) 1-49 (Technical document)
15. Ratz, D., Karlsruhe.: Box-splitting Strategies for the Interval Gauss-Seidel Step in a Global Optimization Method. Computing. 53 (1994) 337-353
16. Ratz, D.: On Extended Interval Arithmetic and Inclusion Isotonicity. Institut für Angewandte Mathmatik, Universität Karlsruhe (1996)
17. Revol, N.: Reliable an Daccurate Solutions of Linear and Nonlinear Systems. SIAM Conference on Optimization, Toronto, Ontario, Canada, 20-22 May, 2002.
18. Schichl, H., Neumaier, A.: Interval Analysis - Basics. In: http://solon.cma.univie.ac.at/ neum/interval.html (2003)
19. Stahl, V.: Interval Methods for Bounding the Range of Polynomials and Solving Systems of Nonlinear Equations (1995) (Dissertation)
20. Wu, W.T.: On Zeros of Algebraic Equations-An Application of Ritt Principle. Kexue Tongbao. 31 (1986) 1-5
21. Xia, B.C., Yang, L.: An Algorithm for Isolating the Real Solutions of Semi-algebraic Systems. Journal of Symbolic Computation. 34 (2002) 461-477
22. Xia, B.C., Zhang, T.: Algorithm for Real Root Isolation Based on Interval Arithmetic (2003) (draft)
23. Yang, L., Hou, X.R.: Gather-And-Shift: a Symbilic Method for Solving Polynomial Systems. Proceedings for First Asian Technology Conference in Mathemtics,18-21 December 1995, Singapore (1995) 771-780
24. Yang, L., Zhang, J.Z., Hou, X.R.: Nonlinear Algebraic Equation System and Automated Theorem Proving. ShangHai Scientific and Technological Education Publishing House, ShangHai (1996) (in Chinese)

A Jacobi–Davidson Method for Nonlinear Eigenproblems

Heinrich Voss

Section of Mathematics, Hamburg University of Technology, D – 21071 Hamburg
voss@tu-harburg.de,http://www.tu-harburg.de/mat/hp/voss

Abstract. For the nonlinear eigenvalue problem $T(\lambda)x = 0$ we consider a Jacobi–Davidson type iterative projection method. The resulting projected nonlinear eigenvalue problems are solved by inverse iteration. The method is applied to a rational eigenvalue problem governing damped vibrations of a structure.

1 Introduction

In this paper we consider the nonlinear eigenvalue problem

$$T(\lambda)x = 0 \tag{1}$$

where $T(\lambda) \in \mathbb{C}^{n\times n}$ is a family of large and sparse matrices depending on a parameter $\lambda \in D \subset \mathbb{C}$. Problems of this type arise in damped vibrations of structures, vibrations of rotating structures, stability of linear systems with retarded argument, lateral buckling problems or vibrations of fluid-solid structures, to name just a few. As in the linear case $T(\lambda) = \lambda I - A$ a parameter λ is called an eigenvalue of $T(\cdot)$ if problem (1) has a nontrivial solution $x \neq 0$ which is called a corresponding eigenvector.

For linear sparse eigenproblems iterative projection methods such as the Lanczos, Arnoldi or Jacobi–Davidson methods are very efficient. In these approaches one determines approximations to the wanted eigenvalues and corresponding eigenvectors from projections of the large eigenproblem to low-dimensional subspaces which are generated in the course of the algorithm. The small projected eigenproblems are solved by standard techniques.

Similar approaches for general nonlinear eigenproblems were studied in [2], [4], [7], and for symmetric problems allowing maxmin characterizations of the eigenvalues in [1] and [8].

Ruhe in [4] (with further modifications and improvements in [2]) linearized the nonlinear problem (1) by regula falsi and applied an Arnoldi type method to the varying sequence of linear problems thus constructing a sequence of search spaces V_k and Hessenberg matrices H_k which approximate the projection of $T(\sigma)^{-1}T(\lambda_k)$ to V_k. Here λ_k denotes an approximation to the wanted eigenvalue and σ a shift close to that eigenvalue. Then a Ritz vector of H_k corresponding to an eigenvalue of small modulus approximates an eigenvector of the nonlinear

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 34–41, 2004.
© Springer-Verlag Berlin Heidelberg 2004

problem (1) from which a new approximation to the corresponding eigenvalue is obtained. Hence, in this approach the two numerical subtasks reducing the large dimension to a low one and solving the projected nonlinear eigenproblem are attacked simultaneously.

In this paper we suggest an iterative projection method for the nonlinear eigenproblem where the two subtasks mentioned in the last paragraph are handled separately. If V_k denotes a subspace of $\mathbb{C}^n$ of small dimension k constructed in the course of the algorithm we solve the projected nonlinear eigenvalue problem $V_k^H T(\lambda) V_k z = 0$ by a dense solver to obtain an approximate eigenvalue λ_k and eigenvector $x_k = V_k z$. After that we expand the space V_k. Similarly as in the Jacobi–Davidson method for linear eigenproblems the expansion direction v_{k+1} of V_k is chosen such that $x_k + \alpha v_{k+1}$ for some $\alpha \in \mathbb{C}$ has a high approximation potential for the eigenvector we are just aiming at. The projection step and the expansion step are repeated alternately until convergence.

Here we consider a method of this type where the search space V_k is expanded by an approximate solution of a correction equation

$$\left(I - \frac{T'(\lambda_k)x_k x_k^H}{x_k^H T'(\lambda_k) x_k}\right) T(\lambda_k) \left(I - \frac{x_k x_k^H}{x_k^H x_k}\right) v = -T(\lambda_k)x_k$$

in a Jacobi–Davidson like manner. In [7] we proposed an expansion of the search space by $v_{k+1} = T(\sigma)^{-1} T(\lambda_k) x_k$ generalizing the residual inverse iteration for dense nonlinear eigenproblems.

The paper is organized as follows. Section 2. discusses the expansion of the search space in a Jacobi–Davidson type way. In particular we discuss the approximate solution of the correction equation by a preconditioned Krylov subspace method. Section 3. reviews solvers of dense nonlinear eigenproblems with special emphasis on the fact that nonlinear problems are often small perturbations of linear problems which can be exploited in the solution process. Section 4. contains the Jacobi–Davidson method for nonlinear eigenproblems and Section 5 demonstrates its numerical behavior for a finite element model of a structure.

2 Expanding the Search Space by Jacobi–Davidson

The Jacobi–Davidson method was introduced by Sleijpen and van der Vorst (cf. [6]) for the linear eigenproblem $Ax = \lambda x$, and generalized in a series of papers with different co-authors to general and to polynomial eigenvalue problems (cf. [5]). Its idea is to construct a correction for a given eigenvector approximation x in a subspace orthogonal to x. Namely, if V is the current search space and (θ, u), $\|u\| = 1$, is a Ritz pair of $Ax = \lambda x$ corresponding to V then V is expanded by a solution t of the so called correction equation

$$(I - uu^H)(A - \theta I)(I - uu^H)t = -(A - \theta I)u, \quad t \perp u.$$

If the correction equation is solved exactly then it is easily seen that the new search space $[V, t]$ contains the vector $\tilde{t} = (A - \theta I)^{-1} u$ obtained by one step

of shifted inverse iteration, and therefore one can expect quadratic (and in the Hermitean case even cubic) convergence.

A natural generalization to the nonlinear eigenproblem (1) which was already suggested in [5] for polynomial eigenvalue problems is the following one: Suppose that the columns of $V \subset \mathbb{C}^k$ form an orthonormal basis of the current search space, and let (u, θ) be a Ritz pair of (1) with respect to V, i.e. $V^H T(\theta) V y = 0$, $u = Vy$. Then we consider the correction equation

$$\left(I - \frac{pu^H}{u^H p}\right) T(\theta) \left(I - \frac{uu^H}{u^H u}\right) t = -r, \quad t \perp u \tag{2}$$

where $p := T'(\theta)u$ and $r := T(\theta)u$.

Equation (2) can be rewritten as $T(\theta)t - \alpha p = -r$ where α has to be chosen such that $t \perp u$. Solving for t we obtain

$$t = -u + \alpha T(\theta)^{-1} p = -u + \alpha T(\theta)^{-1} T'(\theta) u,$$

and $u = Vy$ yields $\tilde{t} := T(\theta)^{-1} T'(\theta) u \in \text{span}[V, t]$.

Hence, as in the linear case the new search space span$[V, t]$ contains the vector obtained by one step of inverse iteration with shift θ and initial vector u, and we may expect quadratic or even cubic convergence of the resulting iterative projection method, if the correction equation (2) is solved exactly.

It has been observed by Sleijpen and van der Vorst for linear problems that the correction equation does not have to be solved accurately but fast convergence of the projection method is maintained if the search space is expanded by an approximate solution, and the same holds true for nonlinear problems. For the linear problem they suggested to apply a few steps of a Krylov solver with an appropriate preconditioner.

In the correction equation (2) the operator $T(\theta)$ is restricted to map the subspace $u^\perp$ to $T'(\theta)u^\perp$. Hence, if $K \approx T(\theta)$ is a preconditioner of $T(\theta)$ then a preconditioner for an iterative solver of (2) should be modified correspondingly to

$$\tilde{K} := (I - \frac{pu^H}{u^H p}) K (I - \frac{uu^H}{u^H u}).$$

With left-preconditioning equation (2) becomes

$$\tilde{K}^{-1} \tilde{T}(\theta) t = -\tilde{K}^{-1} r, \quad t \perp u. \tag{3}$$

where

$$\tilde{T}(\theta) := (I - \frac{pu^H}{u^H p}) T(\theta) (I - \frac{uu^H}{u^H u}).$$

We apply a Krylov solver to equation (3) with initial guess $t = 0$. For the linear case this was already discussed in [6], and the transfer to equation (3) is straightforward.

Since the operator $\tilde{K}^{-1}\tilde{T}(\theta)$ maps the space $u^\perp$ into itself, and since the initial guess $t = 0$ is an element of $u^\perp$, all iterates are contained in this space,

and therefore in each step we have to perform one matrix-vector product $y = \tilde{K}^{-1}\tilde{T}(\theta)v$ for some $v \in u^{\perp}$. To this end we first multiply v by $\tilde{T}(\theta)$ which yields

$$\tilde{y} = (I - \frac{pu^H}{u^H p})T(\theta)v = T(\theta)v - \frac{u^H T(\theta)v}{u^H p}p,$$

and then we solve $\tilde{K}y = \tilde{y}, \quad y \perp u$.

This equation can be rewritten as $Ky - \alpha p = \tilde{y}$, where α is determined from the condition $y \perp u$. Thus, we finally obtain

$$y = K^{-1}\tilde{y} - \frac{u^H K^{-1}\tilde{y}}{u^H K^{-1}p}K^{-1}p \tag{4}$$

which demonstrates that taking into account the projectors in the preconditioner, i.e. using $\tilde{K}$ instead of K, raises the cost of the preconditioned Krylov solver only slightly. To initialize one has to solve the linear system $K\tilde{p} = p$ and to determine the scalar product $\alpha := u^H\tilde{p} = u^H K^{-1}p$. These computations have to be executed just once. Afterwards in each iteration step one has to solve only one linear system $Kw = \tilde{y}$ for w, one has to compute the scalar product $\beta := u^H w = u^H K^{-1}\tilde{u}$, and to perform one axpy $y = w - (\beta/\alpha)\tilde{y}$ to expand the Krylov space of $\tilde{K}^{-1}\tilde{T}(\theta)$.

3 Solving Projected Nonlinear Eigenproblems

Since the dimensions of the projected eigenproblems are usually small they can be solved by any method for dense nonlinear eigenproblems like inverse iteration or residual inverse iteration.

If $T(\lambda)$ is symmetric or Hermitean such that the eigenvalues are real and can be characterized as minmax values of a Rayleigh functional then the projected problem inherits this property, and the eigenvalues can be determined one after the other by safeguarded iteration. This approach which was discussed for the Jacobi–Davidson method in [1] and for the Arnoldi method in [8] has the advantage that it is most unlikely that the method converges to an eigenvalue that has already been found previously.

In the general case the following strategy is similar to safeguarded iteration. Assume that we want to determine all eigenvalues of problem (1) in the vicinity of a given parameter $\sigma_0 \in D$, and that already $m - 1$ eigenvalues closest to σ_0 have been determined. Assume that $\tilde{\mu}$ is an approximation to the eigenvalue wanted next.

A first order approximation of problem (1) is

$$T(\lambda)x \approx (T(\tilde{\mu}) - \theta T'(\tilde{\mu}))x = 0, \ \theta = \tilde{\mu} - \lambda. \tag{5}$$

This suggests the method of successive linear problems in Algorithm 1 which was introduced by Ruhe [3], and which converges quadratically.

Of course this method is not appropriate for a sparse problem (1), but in an iterative projection method the dimension of the projected problem which has

Algorithm 1 Method of successive linear problems

1: Start with an approximation μ_1 to the m-th eigenvalue of (1)
2: **for** $\ell = 1, 2, \ldots$ until convergence **do**
3: solve the linear eigenproblem $T(\mu_\ell)u = \theta T'(\mu_\ell)u$
4: choose the eigenvalue θ such $|\sigma_0 - (\mu_\ell - \theta)|$ is the m–smallest among the eigenvalues
5: $\mu_{\ell+1} = \mu_\ell - \theta$
6: **end for**

to be solved in step 3. usually is quite small, and every standard solver for dense eigenproblems applies.

Quite often the nonlinear eigenvalue problem under consideration is a (small) perturbation of a linear eigenvalue problem. As a numerical example we will consider a finite element model of a vibrating structure with nonproportional damping. Using a viscoelastic constitutive relation to describe the behavior of a material in the equations of motions yields a rational eigenvalue problem for the case of free vibrations. A finite element model obtains the form

$$T(\omega) := \left(\omega^2 M + K - \sum_{j=1}^{J} \frac{1}{1 + b_j \omega} \Delta K_j\right) x = 0. \tag{6}$$

If the damping is not too large the eigenmodes of the damped and the undamped problem do not differ very much although the eigenvalues do. Therefore, in step 3. of Algorithm 2 it is reasonable to determine an eigenvector y of the undamped and projected problem $(\omega^2 V^H M V + V^H K V)y = 0$ corresponding to the m-smallest eigenvalue ω_m^2, determine an approximate eigenvalue $\tilde{\omega}$ of the nonlinear projected problem from the complex equation $y^H V^H T(\omega) V y = 0$ or $e^H V^H T(\sigma)^{-1} T(\omega) V y = 0$, and correct it by (residual) inverse iteration.

4 Jacobi–Davidson Method for Nonlinear Eigenproblems

A template for the Jacobi–Davidson method for the nonlinear eigenvalue problem (1) is given in Algorithm 2.

Remarks on some of its steps are inorder:

1. In V preinformation about the wanted eigenvectors (which may be gained from previous solutions of similar problems) can be introduced into the method.
   If we are interested in eigenvalues close to a given parameter σ and no information on eigenvectors is at hand we can start the Jacobi–Davidson method with an orthogonal basis V of an invariant subspace of the linear eigenproblem $T(\sigma)u = \theta u$ (or $T(\sigma)u = \theta T'(\sigma)u$) corresponding to eigenvalues θ which are small in modulus.
8. As the subspaces expand in the course of the algorithm the increasing storage and the computational cost for solving the projected eigenproblems may

Algorithm 2 Nonlinear Jacobi–Davidson method

1: Start with an initial basis V, $V^H V = I$; $m = 1$
2: determine a preconditioner $K \approx T(\sigma)^{-1}$, σ close to the first wanted eigenvalue
3: **while** $m \leq$ number of the wanted eigenvalues **do**
4: compute an approximation to the m-th wanted eigenvalue λ_m and corresponding eigenvector x_m of the projected problem $V^H T(\lambda) V x = 0$
5: determine the Ritz vector $u = V x_m$ and the residual $r = T(\lambda_m) u$
6: **if** $\|r\|/\|u\| < \epsilon$ **then**
7: accept approximate eigenpair (λ_m, u); increase $m = m + 1$;
8: reduce the search space V if necessary
9: choose an approximation (λ_m, u) to the next eigenpair
10: compute the residual $r = T(\lambda_m) u$;
11: **end if**
12: Find an approximate solution of the correction equation

$$(I - \frac{T'(\lambda_m) u u^H}{u^H T'(\lambda_m) u}) T(\sigma) (I - \frac{u u^H}{u^H u}) t = -r \tag{7}$$

13: orthogonalize $t = t - V V^H t$, $v = t/\|t\|$, and expand the subspace $V = [V, v]$
14: determine a new preconditioner $K \approx T(\lambda_m)$ if necessary
15: update the projected problem
16: **end while**

make it necessary to restart the algorithm and to purge some of the basis vectors. Since a restart destroys information on the eigenvectors and particularly the one the method is just aiming at we restart only if an eigenvector has just converged. A reasonable search space after restart is the space spanned by the already converged eigenvectors (or a space slightly larger).

12. The correction equation can be solved by a preconditioned Krylov solver, e.g.
13. The first two statements represent the classical Gram–Schmidt process. It is advisable to repeat this orthogonalization step once if the norm of t is reduced by more than a modest factor, say $\|t\|/\|v_{\text{old}}\| < 0.25$, e.g.
14. We solved the correction equation (7) by a few steps of preconditioned GMRES where we kept the preconditioner for a couple of eigenvalues. We terminated the solver of (7) in the k-th outer iteration for the m-th eigenvalue if the residual was reduced by at least $\tau_k = 2^{-k}$, and we allowed at most 10 steps of the solver. If the required accuracy τ_k was not met after at most 5 iteration steps we updated the preconditioner. However, we allowed at most one update for every eigenvalue λ_m.

5 Numerical Experiments

To test the Jacobi–Davidson method we consider the rational eigenvalue problem (6) governing damped vibrations of a column

$$\{(x, y, z) : 0.8 < \sqrt{x^2 + y^2} < 1,\ 0 < z < 5\}$$

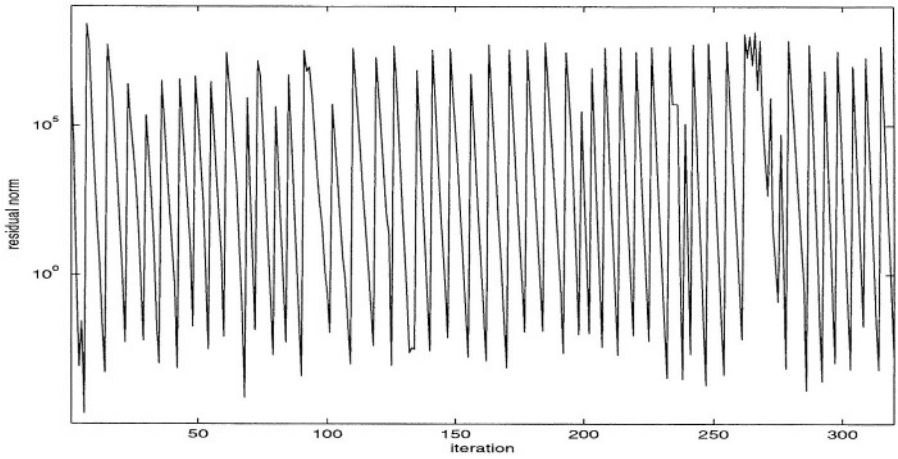


Fig. 1. Convergence history without restarts

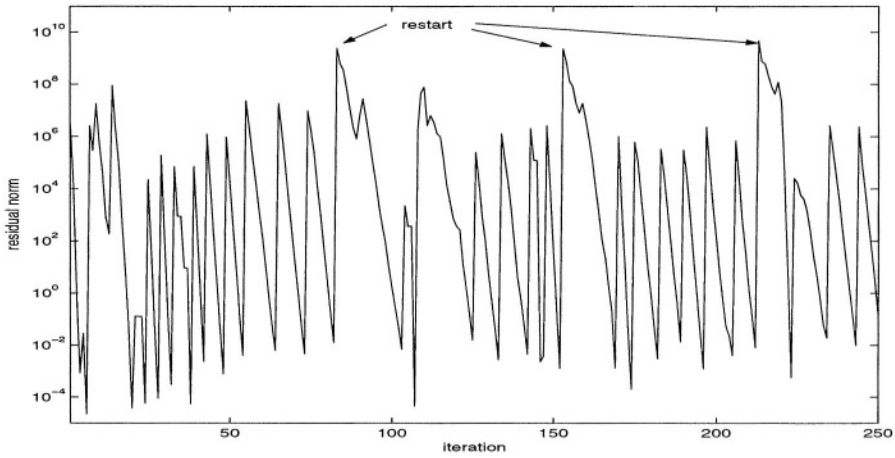


Fig. 2. Convergence history with restarts (first 250 iterations)

which is clamped at its bottom $z = 0$. The instantaneous Young's modulus is set to $E = 2.06 \times 10^{11}$, the instantaneous Poisson's rate is $\nu = 0.3$, and the density is set to $\rho = 7800$. For the nonproportional damping we use in addition the following parameters, $\Delta\nu = 0.27$, and $\Delta E = 6 \times 10^{10}$ for $0 < x < 2.5$, and $\Delta E = 5 \times 10^{10}$ for $2.5 < x < 5$. The relaxation constant is set to $b = 10^{-4}$.

Discretizing this problem by linear Lagrangian elements we obtained the rational eigenproblem (6) of dimension 11892, and the bandwidth of the stiffness matrix K was after reducing it by reverse Cuthill–McKee algorithm still 665. For symmetry reasons we determined only eigenvalues with negative imaginary part, and we computed 50 of them one after another with decreasing imaginary part.

The nonlinear projected eigenproblems were solved by inverse iteration with an initial guess obtained from the corresponding undamped projected problem as explained at the end of Section 3.

The experiments were run under MATLAB 6.5 on a Pentium 4 processor with 2 GHz and 1 GB RAM. We preconditioned by the LU factorization of $T(\sigma)$, and terminated the iteration if the norm of the residual was less than 10^{-2}.

Starting with an eigenvector of the linear eigenproblem $Kx = \lambda Mx$ corresponding to the smallest eigenvalue the algorithm without restarts needed 320 iteration steps, i.e. an average of 6.4 iterations per eigenvalue, to approximate all 50 eigenvalues (including double eigenvalues) with maximal negative imaginary part. To solve the correction equations a total of 651 GMRES steps were needed, and 6 updates of the preconditioner were necessary. Fig. 1. contains the convergence history.

Restarting the Jacobi–Davidson process if the dimension of the research space exceeded 80 the method needed 7 restarts. Again all 50 eigenvalues were found by the method requiring 422 iterations, 840 GMRES steps, and 16 updates of the preconditioner. The convergence history in Fig. 2. looks very similar to the one without restarts, however, after a restart the speed of convergence was reduced considerably. After a restart an average of 17.1 iterations was necessary to gather enough information about the search space and to make the method converge, whereas for the other iteration steps the average number of steps for convergence was 7.0.

References

1. T. Betcke and H. Voss. A Jacobi–Davidson–type projection method for nonlinear eigenvalue problems. *Future Generation Computer Systems,* 20(3):363 – 372, 2004.
2. P. Hager. *Eigenfrequency Analysis. FE-Adaptivity and a Nonlinear Eigenvalue Problem.* PhD thesis, Chalmers University of Technology, Göteborg, 2001.
3. A. Ruhe. Algorithms for the nonlinear eigenvalue problem. *SIAM J. Numer. Anal.,* 10:674 – 689, 1973.
4. A. Ruhe. A rational Krylov algorithm for nonlinear matrix eigenvalue problems. *Zapiski Nauchnyh Seminarov POMI,* 268:176 – 180, 2000.
5. G.L. Sleijpen, G.L. Booten, D.R. Fokkema, and H.A. van der Vorst. Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems. *BIT,* 36:595 – 633, 1996.
6. G.L. Sleijpen and H.A. van der Vorst. A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM J.Matr.Anal.Appl.,* 17:401 – 425, 1996.
7. H. Voss. An Arnoldi method for nonlinear eigenvalue problems. Technical Report 56, Section of Mathematics, Hamburg University of Technology, 2002. To appear in BIT Numerical Mathematics.
8. H. Voss. An Arnoldi method for nonlinear symmetric eigenvalue problems. In *Online Proceedings of the SIAM Conference on Applied Linear Algebra, Williamsburg,* http://www.siam.org/meetings/laa03/, 2003.

Numerical Continuation of Branch Points of Limit Cycles in MATCONT

Annick Dhooge[1], Willy Govaerts[1], and Yuri A. Kuznetsov[2]

[1] Department of Applied Mathematics and Computer Science, Gent University, Krijgslaan 281-S9,B-9000 Gent, Belgium
{Annick.Dhooge,Willy.Govaerts}@UGent.be
[2] Mathematical Institute, Utrecht University, Budapestlaan 6, 3584 CD Utrecht, The Netherlands
Yu.Kuznetsov@math.uu.nl

Abstract. MATCONT is a MATLAB continuation package for the interactive numerical study of a range of parameterized nonlinear problems. We discuss a recent addition to the package, namely the continuation of branch points of limit cycles in three parameters which is not available in any other package. This includes the exact location of the BPC points and branch switching. The algorithm is important in the numerical study of symmetry and we illustrate it in the case of the famous Lorenz model for the atmospheric circulation.

1 Introduction

Numerical continuation is a technique to compute a sequence of points which approximate a branch of solutions to $F(x) = 0$ where $F : \mathbb{R}^{N+1} \to \mathbb{R}^N$.

In particular, we consider a dynamical system of the form

$$\frac{dx}{dt} = f(x, \alpha) \tag{1}$$

with $x \in \mathbb{R}^n$, $f(x, \alpha) \in \mathbb{R}^n$, and α a vector of parameters. In this setting equilibria, limit points, limit cycles etcetera can be computed.

MATCONT provides a continuation toolbox for (1) which is compatible with the standard MATLAB representation of ODEs. The package is freely available at: `http://allserv.UGent.be/~ajdhooge/research.html`. It requires MATLAB 6.*. In [4] we describe the implementation in MATCONT of the continuation of the Fold bifurcation of limit cycles, using a minimal extended system, i.e. we only append a scalar equation to the definition of limit cycles [6]. Here we discuss the continuation in three parameters of branch points of limit cycles, an algorithm which is not available in any other package. For general background on dynamical systems we refer to [8,9]; for the algorithms that involve BPC we refer to [7].

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 42–49, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Mathematical Background on Limit Cycles and Their Branch Points

2.1 Limit Cycles and Their Branch Points

A limit cycle is an isolated closed orbit that corresponds to a periodic solution of (1) with period T, i.e. $x(0) = x(T)$. Since T is not known in advance, it is customary (cf AUTO [5], CONTENT [10]) to use an equivalent system defined on the unit interval [0,1] by rescaling time:

$$\begin{cases} \dfrac{dx}{dt} - Tf(x,\alpha) = 0, \\ x(0) = x(1). \end{cases} \tag{2}$$

To obtain a unique solution the following integral constraint is often used [5,10]:

$$\int_0^1 \langle x(t), \dot{x}_{old}(t)\rangle dt = 0, \tag{3}$$

where $\dot{x}_{old}(t)$ is the derivative vector of a previously calculated limit cycle and is therefore known, $\langle x, v\rangle$ is just a different notation for $x^{\mathrm{T}}v$. The left-hand side of (3) will be sometimes denoted by $Int_{\dot{x}_{old}}(x)$.

If, say, α_1 is the control parameter in (1) then a branch point of limit cycles (BPC) is a solution to (2)–(3) in $(x(t), T, \alpha_1)$-space where the null space of the derivative operator of (2)–(3) with respect to $x(t), T, \alpha_1$ has dimension greater than one. Generically, it then has a two-dimensional null space and the solution to (2)–(3) has two intersecting branches in $(x(t), T, \alpha_1)$-space. The complete BVP denning a BPC point using the minimal extended system is

$$\begin{cases} \frac{dx}{dt} - Tf(x,\alpha) & = 0 \\ x(0) - x(1) & = 0 \\ \int_0^1 \langle x(t), \dot{x}_{old}(t)\rangle dt & = 0 \\ G[x, T, \alpha] & = 0 \end{cases} \tag{4}$$

where $G = (G^{1T}\ G^{2T})^T$ is defined by requiring

$$L \begin{pmatrix} v^{11} & v^{21} \\ v^{12} & v^{22} \\ v^{13} & v^{23} \\ G^1 & G^2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{5}$$

Here v^{1k} and v^{2k} are vector functions, G^1 and G^2 are scalars and

$$L = \begin{bmatrix} D - Tf_x(x(\cdot),\alpha) & -f(x(\cdot),\alpha) & -Tf_{\alpha_1}(x(\cdot),\alpha) & w_0^1 \\ \delta_1 - \delta_0 & 0 & 0 & w_0^2 \\ Int_{\dot{x}_{old}(\cdot)} & 0 & 0 & w_0^3 \\ v_0^{11} & v_0^{12} & v_0^{13} & 0 \\ v_0^{21} & v_0^{22} & v_0^{23} & 0 \end{bmatrix} \tag{6}$$

where the bordering operators v_0^{11}, v_0^{21}, function w_0^1, vector w_0^2 and scalars v_0^{12}, v_0^{22}, v_0^{13}, v_0^{23} and w_0^3 are chosen so that L is nonsingular [6,7]. To interpret (6) correctly, note that $\delta_a v = v(a)$. The defining system composed of (5) and (6) can be used to continue the BPC in three control parameters.

3 Numerical Continuation of Limit Cycles

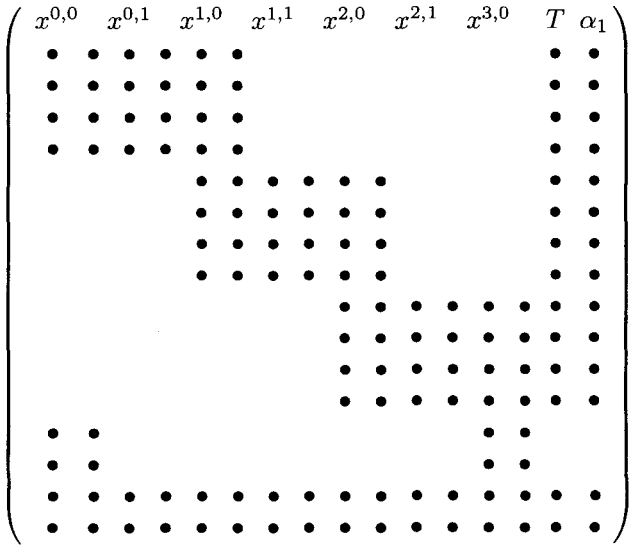
For the numerical continuation of a limit cycle with respect to a control parameter we discretize the system consisting of (2) and (3); to use a Newton-like method the Jacobi matrix of the discretized system is also needed. We exploit the sparsity by using the MATLAB routines for sparse matrices.

Using the orthogonal collocation described, for example, in [4] we obtain the discretized BVP (2)–(3) in the form:

$$\begin{cases} \left(\sum_{j=0}^{m} x^{i,j} l'_{i,j}(\zeta_{i,k})\right) - Tf(\sum_{j=0}^{m} x^{i,j} l_{i,j}(\zeta_{i,k}), \alpha) = 0 \\ x^{0,0} - x^{N-1,m} = 0 \\ \sum_{i=0}^{N-1} \sum_{j=0}^{m-1} \sigma_{i,j} \langle x^{i,j}, \dot{x}_{old}^{i,j} \rangle + \sigma_{N,0} \langle x^{N,0}, \dot{x}_{old}^{N,0} \rangle = 0 \end{cases}$$

The first equation in fact consists of Nm equations, one for each combination of $i = 0, 1, 2, ..., N-1$ and $k = 1, 2, ..., m$.

In the Newton iterations during the continuation process a system consisting of theJacobi matrix and an extra row (the tangent vector) is solved. For $N = 3$ test intervals, $m = 2$ collocation points and $\dim(x) = 2$ this matrix has the following sparsity structure (•'s are generically non-zero). This is explained in more detail in [4].



(7)

4 Continuation of BPC Cycles

4.1 Discretization of the BPC Equations

The last equation in (4) expresses that the operator

$$\begin{bmatrix} D - Tf_x(x(\cdot),\alpha) & -f(x(\cdot),\alpha) & -Tf_{\alpha_1}(x(\cdot),\alpha) \\ \delta_1 - \delta_0 & 0 & 0 \\ Int_{\dot{x}old(\cdot)} & 0 & 0 \end{bmatrix} \tag{8}$$

that appears as a block in (6) is rank deficient. In the numerical implementation in MATCONT and CL_MATCONT we replace this by the condition that the discretized operator of (8) is rank deficient: $G_d^1 = G_d^2 = 0$. To find G_d^i, we solve

$$L^d \begin{pmatrix} v_d^{11} & v_d^{21} \\ v_d^{12} & v_d^{22} \\ v_d^{13} & v_d^{23} \\ G_d^1 & G_d^2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{9}$$

where $L^d =$

$$\begin{bmatrix} [\ D - Tf_x(x(\cdot),\alpha)\ ]_d & [-f(x(\cdot),\alpha)]_d & [-Tf_{\alpha_1}(x(\cdot),\alpha)]_d & w_{0d}^1 \\ I_{\dim(x)}\ 0_{n\times(Nm-1)\dim(x)}\ -I_{\dim(x)} & 0 & 0 & w_{0d}^2 \\ Int_{[\dot{x}_{old}(\cdot)]_d} & 0 & 0 & w_{0d}^3 \\ v_{0d}^{11\mathrm{T}} & v_{0d}^{12} & v_{0d}^{13} & 0 \\ v_{0d}^{21\mathrm{T}} & v_{0d}^{22} & v_{0d}^{23} & 0 \end{bmatrix}, \tag{10}$$

where the bordering vectors $v_{0d}^{11}, v_{0d}^{21}, w_{0d}^1$ and w_{0d}^2 and scalars $v_{0d}^{12}, v_{0d}^{22}, v_{0d}^{13}, v_{0d}^{23}$ and w_{0d}^3 are chosen so that L^d is nonsingular. The structure is similar to that of (7); however, the bordering rows and columns have a different meaning.

To continue a solution branch of the discretized equations (4), the Jacobi matrix of the system is needed, which means that the derivatives of G_d with respect to the unknowns of the system, i.e., with respect to $x^{i,j}, T,$ and the control parameters α, have to be calculated.

The derivative with respect to z (being a component of $x^{i,j}, T$ or α) is obtained from

$$L^d \begin{pmatrix} v_{dz}^{11} & v_{dz}^{21} \\ v_{dz}^{12} & v_{dz}^{22} \\ v_{dz}^{13} & v_{dz}^{23} \\ G_{dz}^1 & G_{dz}^2 \end{pmatrix} + J_z \begin{pmatrix} v_d^{11} & v_d^{21} \\ v_d^{12} & v_d^{22} \\ v_d^{13} & v_d^{23} \\ G_d^1 & G_d^2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Simplifying gives

$$L^d \begin{pmatrix} v_{dz}^{11} & v_{dz}^{21} \\ v_{dz}^{12} & v_{dz}^{22} \\ v_{dz}^{13} & v_{dz}^{23} \\ G_{dz}^1 & G_{dz}^2 \end{pmatrix} = \begin{pmatrix} J_1 & J_2 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

where $J_i = [Tf_x(x(t),\alpha]_{dz}v_d^{i1} + [f(x(t),\alpha)]_{dz}v_d^{i2} + [Tf_{\alpha_1}(x(t),\alpha]_{dz}v_d^{i3}, \quad i = 1,2.$ Instead of solving this for every z we solve the transposed equations

$$(w_d^{1\mathrm{T}}, w_d^{2\mathrm{T}}, w_d^3, w_d^4, w_d^5)L^d = (0,0,0,1) \tag{11}$$

where w_d^1 is a $dim(x) \times Nm$ vector, w_d^2 a $dim(x)$ vector and w_d^3, w_d^4 and w_d^5 are scalars. Combining (9) and (11) we find

$$G_{dz}^i = w_d^{1\mathrm{T}}([Tf_x(x(t),\alpha]_{dz}v_d^{i1} + [f(x(t),\alpha)]_{dz}v_d^{i2} + Tf_{\alpha_1}(x(t),\alpha]_{dz}v_d^{i3}). \tag{12}$$

So in each iteration step we solve three systems with the structure of (7) or its transpose.

4.2 Initialization and Adaptation of the Borders

The bordering vectors in (10) must be such that the matrix L^d is nonsingular. We choose them in such a way that L^d is as well conditioned as possible. This involves an initialization of the borders when the continuation is started and a subsequent adaptation during the continuation.

During the initialization the borders must be chosen so that the extension L^d of O =

$$\begin{bmatrix} [\ D\ -\ Tf_x(x(\cdot),\alpha)\ ]_d & & [-f(x(\cdot),\alpha)]_d & [-Tf_{\alpha_1}(x(\cdot),\alpha)]_d \\ I_{\dim(x)}\ \ 0_{\dim(x)\times(Nm-1)\dim(x)}\ \ -I_{\dim(x)} & & 0 & 0 \\ Int_{f(x(\cdot),\alpha)_d} & & 0 & 0 \end{bmatrix}$$

has full rank. We first perform an QR orthogonal-triangular decomposition with column pivoting. The MATLAB command *[Q,R,E] = QR(full(O))* produces a permutation matrix *E,* an upper triangular matrix *R* of the same dimension as *O* and an unitary matrix *Q* so that *OE = QR.* The column pivoting guarantees that the QR decomposition is rank revealing and in particular that abs(diag(*R*)) is decreasing. Since *O* has rank defect 1, the last element on the diagonal and the bottom right element of *R* should be zero (up to approximation). The borders $[v_{0d}^{11\mathrm{T}}v_{0d}^{12}v_{0d}^{13}]$ and $[v_{0d}^{21\mathrm{T}}v_{0d}^{22}v_{0d}^{23}]$ in (10) are chosen as an orthogonal base for the null space of *O.* If p is a two-column matrix that spans this null space, then from $Op = 0$ follows that $RE^{\mathrm{T}}p = 0$. Setting the bottom right element and the last element on the diagonal of *R* to zero, we obtain

$$\left(\begin{array}{ccccc|cc} * & * & * & \dots & * & * & * \\ 0 & * & * & \dots & * & * & * \\ 0 & 0 & * & \dots & * & * & * \\ & & & \dots & & \dots & * \\ 0 & 0 & 0 & \dots & * & * & * \\ \hline 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{array}\right) E^{\mathrm{T}}p = \left(\begin{array}{c} 0\ 0 \\ \\ \\ \hline 0\ 0 \end{array}\right)$$

By imposing some structure on $E^{\mathrm{T}}p$ we get

$$\left(\begin{array}{c|cc} R_1 & b_1 & b_2 \\ \hline 0 & 0 & 0 \end{array}\right) \left(\begin{array}{c|c} p_1 & p_2 \\ \hline 1 & 0 \\ 0 & 1 \end{array}\right) = \left(\begin{array}{c} 0\ 0 \\ \hline 0\ 0 \end{array}\right).$$

or

$$R_1(p_1\ p_2) = -(b_1\ b_2)$$

where R_1 is a nonsingular square upper triangular matrix. So $[v_{0d}^{11\mathrm{T}} v_{0d}^{12} v_{0d}^{13}]$ and $[v_{0d}^{21\mathrm{T}} v_{0d}^{22} v_{0d}^{23}]$ in (10) are initially chosen as the normalization and orthogonalization of $E[(R_1 \backslash [-b_1, -b_2]); eye(2)]$ where $eye(2)$ is the 2-by-2 identity matrix. We choose this column as the bordering column $[w_{0d}^{1\mathrm{T}} w_{0d}^{2\mathrm{T}} w_{0d}^{3}]^{\mathrm{T}}$ in (10). This choice of the borders in (10) makes the bordered matrix nonsingular.

The borders $[v_{0d}^{11\mathrm{T}} v_{0d}^{12} v_{0d}^{13}]$ and $[v_{0d}^{21\mathrm{T}} v_{0d}^{22} v_{0d}^{23}]$ are adapted by replacing them by the normalized and orthogonalized $[v_d^{11\mathrm{T}} v_d^{12} v_d^{13}]$ and $[v_d^{21\mathrm{T}} v_d^{22} v_d^{23}]$ in (9). The borders w_{0d}^1, w_{0d}^2 and w_{0d}^3 in (10) are adapted by solving the transposed equations and replacing them respectively by the normalized and orthogonalized w_d^1, w_d^2 and w_d^3 in (11).

5 BPC Cycles on a Curve of Limit Cycles

Generically, i.e. if no symmetry is present, then BPC are not expected on curves of limit cycles. However, they are common if the system has symmetry. The location and processing of BPC in that case requires a special treatment.

5.1 Branch Point Locator

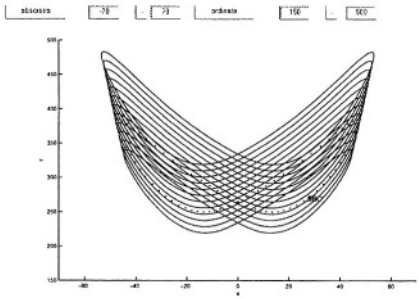
Location of BPC points in as zeros of some test functions causes numerical difficulties because no local quadratic convergence can be guaranteed (see [3] in the case of equilibria). This difficulty is avoided by introducing an additional unknown $\beta \in \mathbb{R}$ and considering the minimally extended system:

$$\begin{cases} \frac{dx}{dt} - Tf(x,\alpha) + \beta p_1 & = 0 \\ x(0) - x(1) + \beta p_2 & = 0 \\ \int_0^1 \langle x(t), \dot{x}_{old}(t) \rangle dt + \beta p_3 & = 0 \\ G[x, T, \alpha] & = 0 \end{cases} \tag{13}$$

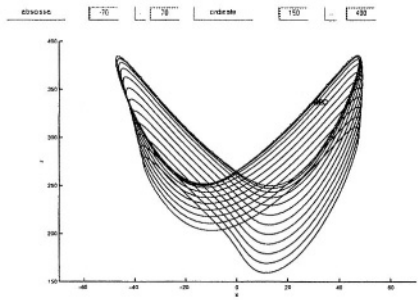
where G is defined as in (5) and $[p_1^{\mathrm{T}} p_2^{\mathrm{T}} p_3]^{\mathrm{T}}$ is the bordering vector $[w_{0d}^{1\mathrm{T}}; w_{0d}^{2\mathrm{T}}; w_{0d}^{3}]^{\mathrm{T}}$ in (10) calculated as in §4.2. We solve this system with respect to x, T, α_1 and β by Newton's method with initial $\beta = 0$. A branch point (x, T, α_1) corresponds to a regular solution $(x, T, \alpha_1, 0)$ of system (13) (see [1], p. 165).

5.2 Processing of the BPC

The tangent vector v_{1st} at the BPC singularity is approximated as $v_{1st} = \frac{v_1 + v_2}{2}$ where v_1 is the tangent vector in the continuation point previous to the BPC and v_2 is the one in the next point. To start the continuation of the secondary cycle branch passing through the BPC point, we need an approximation of the tangent vector of the secondary branch. We choose the vector which is in the space spanned by $p = [p_1 p_2]$ obtained in §4.2 as an orthogonal base for the null space of O and orthogonal to the tangent vector to the primary branch.

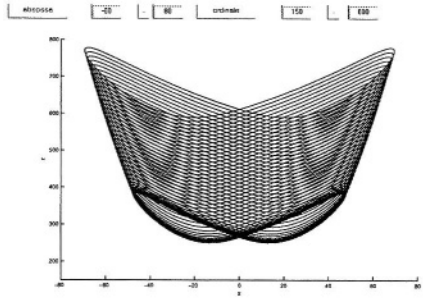


(a) Continuation of symmetric limit cycle

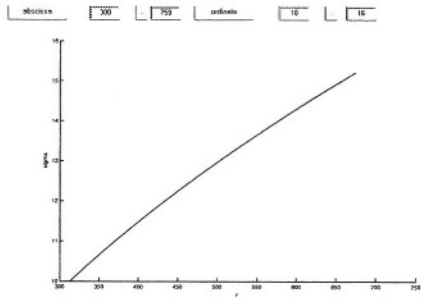


(b) Continuation of asymmetric limit cycle

Fig. 1.



(a) Continuation of BPC curve in phase space.



(b) Continuation of BPC curve in (r,σ)-space.

Fig. 2.

6 An Example

Consider the Lorenz model [11] where x, y and z are state variables and r, σ and b are parameters:

$$\begin{cases} \dot{x} = \sigma(-x + y) \\ \dot{y} = rx - y - xz \\ \dot{z} = -bz + xy. \end{cases} \tag{14}$$

This problem satisfies the equivariance relation with respect to a group of two transformations, i.e. $\{I_3, S\}$, where S = Diag(–1, –1,1). As in the Tutorial to CONTENT[10], we compute an orbit starting from the point (0,50,600) at $\sigma = 10$, $r = 400$ and $b = \frac{8}{3}$ and start a limit cycle continuation with respect to the control parameter r from the converged closed orbit. This is clearly a branch of S-symmetric periodic solutions of (14), see Fig. 1(a). We detect a BPC at $r = 312.9735\ldots$. We continue in r the secondary cycle branch passing through the BPC point. From Fig. 1(b) it is clear that for the secondary cycle the S-symmetry is broken. To compute the branch of BPC points with respect to r through the BPC point with control parameters r, σ we need to introduce

an additional free parameter that breaks the symmetry. We choose to introduce a parameter ϵ and extend the system (14) by simply adding the term ϵ to the right-hand side of the first equation in (14). For $\epsilon = 0$ this reduces to (14) while for $\epsilon \neq 0$ the symmetry is broken. Using the code for the continuation of generic BPC points with three free parameters r, σ, ϵ we continue the curve of non-generic BPC points, where ϵ remains close to zero $(1e-8)$. The picture in Fig. 2 clearly shows that the symmetry is preserved.

References

1. Beyn, W.J., Champneys, A.R., Doedel, E., Govaerts, W., Kuznetsov, Yu.A., Sandstede, B.: Numerical continuation and computation of normal forms. In: B. Fiedler, ed. "Handbook of Dynamical Systems, Vol 2", Elsevier 2002, 149–219.
2. Dhooge, A., Govaerts, W., Kuznetsov Yu.A.: MATCONT: A MATLAB package for numerical bifurcation analysis of ODEs, *ACM TOMS* **29**(2) (2003), 141–164.
3. Dhooge, A., Govaerts, W., Kuznetsov, Yu.A., Mestrom, W., Riet, A.M. : A Continuation Toolbox in MATLAB, Manual (2003): `http://allserv.UGent.be/~ajdhooge/doc_cl_matcont.zip`
4. Dhooge, A., Govaerts, W., Kuznetsov, Yu.A.: Numerical continuation of fold bifurcations of limit cycles in MATCONT, Proceedings of the ICCS 2003, Part I. Springer Verlag Lecture Notes in Computer Science, Vol. 2657 (May 2003) (eds. P.M.A. Sloot, D. Abramson, A.V. Bogdanov, J.J. Dongarra, A.Y. Zomaya and Y.E. Gorbachev), 701–710.
5. Doedel, E.J., Champneys, A.R., Fairgrieve, T.F., Kuznetsov, Yu.A., Sandstede, B., Wang, X.J., AUTO97-AUTO2000 : Continuation and Bifurcation Software for Ordinary Differential Equations (with HomCont), User's Guide, Concordia University, Montreal, Canada (1997–2000): `http://indy.cs.concordia.ca`.
6. Doedel, E.J., Govaerts W., Kuznetsov, Yu.A.: Computation of periodic solution bifurcations in ODEs using bordered systems, *SIAM J. Numer. Anal.* **41**(2) (2003), 401–435.
7. Doedel, E.J., Govaerts, W., Kuznetsov, Yu.A., Dhooge A.: Numerical continuation of branch points of equilibria and periodic orbits, Preprint 1280, Department of Mathematics, Utrecht University, The Netherlands (2003)
8. Govaerts, W.: Numerical Methods for Bifurcations of Dynamical Equilibria, SIAM, Philadelphia (2000).
9. Kuznetsov, Yu.A.: Elements of Applied Bifurcation Theory, 2nd edition, Springer-Verlag, New York (1998)
10. Kuznetsov, Yu.A., Levitin, V.V.: CONTENT: Integrated Environment for Analysis of Dynamical Systems. CWI, Amsterdam (1997): `ftp://ftp.cwi.nl/pub/CONTENT`
11. Lorenz, E.: Deterministic non-periodic flow, *J. Atmos. Science* **20** (1963), 130–141.

Online Algorithm for Time Series Prediction Based on Support Vector Machine Philosophy

J.M. Górriz[1], C.G. Puntonet[2], and M. Salmerón[2]

[1] E.P.S. Algeciras, Universidad de Cádiz,
Avda. Ramón Puyol s/n, 11202 Algeciras Cádiz, Spain
juanmanuel.gorriz@uca.de
[2] E.S.I., Informática, Universidad de Granada
C/ Periodista Daniel Saucedo, 69042 Granada, Spain
{carlos, moises}@atc.ugr.es

Abstract. In this paper we prove the analytic connection between *Support Vector Machines* (SVM) and *Regularization Theory* (RT) and show, based on this prove, a new on-line parametric model for time series forecasting based on Vapnik-Chervonenkis (VC) theory. Using the latter strong connection, we propose a regularization operator in order to obtain a suitable expansion of radial basis functions (RBFs) and expressions for updating neural parameters. This operator seeks for the "flattest" function in a feature space, minimizing the risk functional. Finally we mention some modifications and extensions that can be applied to control neural resources and select relevant input space.

1 Introduction

The purpose of this work is twofold. It introduces the foundations of SVM [4] and its connection with RT [1]. Based on this connection we show the new on-line algorithm for time series forecasting. SVMs are learning algorithms based on the structural risk minimization principle [2] (SRM) characterized by the use of the expansion of support vector (SV) "admissible" kernels and the sparsity of the solution. They have been proposed as a technique in time series forecasting [3] and they have faced the overfitting problem, presented in classical neural networks, thanks to their high capacity for generalization. The solution for SVM prediction is achieved solving the constrained quadratic programming problem. thus SV machines are nonparametric techniques, i.e. the number of basis functions are unknown before hand. The solution of this complex problem in *real-time applications* can be extremely uncomfortable because of high computational time demand.

SVMs are essentially Regularization Networks (RN) with the kernels being Green's function of the corresponding regularization operators [4]. Using this connection, with a clever choice of regularization operator (based on SVM philosophy), we should obtain a parametric model being very resistant to the overfitting problem. Our parametric model is a *Resource allocating Network* [5]

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 50–57, 2004.
© Springer-Verlag Berlin Heidelberg 2004

characterized by the control of neural resources and by the use of matrix decompositions, i.e. Singular Value Decomposition (SVD) and QR Decomposition with pivoting to input selection and neural pruning [6].

We organize the essay as follows. SV algorithm and its analytic connection to RT Theory will be presented in section 2. The new on-line algorithm will be compare to a previous version of it and to the standard SVM in section 4. Finally we state some conclusions in section 5.

2 Analytic Connection between SVM and RT

The SV algorithm is a nonlinear generalization of the generalized portrait developed in the sixties by Vapnik and Lerner in [10]. The basic idea in SVM for regression and function estimation, is to use a mapping Φ from the input space $\mathcal{X}$ into a high dimensional feature space $\mathcal{F}$ and then to apply a linear regression. Thus the standard linear regression transforms into:

$$f(x) = \langle \omega \cdot \Phi(x) \rangle + b. \tag{1}$$

where $\Phi : \mathcal{X} \to \mathcal{F}$, b is a bias or threshold and $\omega \in \mathcal{F}$ is a vector defining the function class. The target is to determinate ω, i.e. the set of parameters in the neural network, minimizing the regularizated risk expressed as:

$$R_{reg}[f] = R_{emp}[f] + \lambda ||\omega||^2. \tag{2}$$

thus we are enforcing "flatness" in feature space, that is we seek small ω. Note that equation 2 is very common in RN with a certain second term. SVM algorithm is a way of solving the minimization of equation 2, which can be expressed as a quadratic programming problem using the formulation stated in [11]:

$$minimize \quad \frac{1}{2}||\omega||^2 + C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*). \tag{3}$$

given a suitable Loss function L(·)[1] a constant $C \geq 0$ and with slack variables $\xi_i, \xi_i^* \geq 0$. The optimization problem is solve constructing a Lagrange function by introducing dual variables, using equation 3 and the selected loss function. Once it is uniquely solved, we can write the vector ω in terms of the data points as follows:

$$\omega = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \Phi(x_i). \tag{4}$$

where α_i, α_i^* are the solutions of the mentioned quadratic problem. Once this problem, characterized by a high computational demand [2], is solved we use equation 4 and 1, obtaining the solution in terms of dot products:

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \langle \Phi(x_i) \cdot \Phi(x) \rangle + b. \tag{5}$$

[1] For example Vapnik's ϵ insensitive loss function [11].

[2] This calculation must be compute several times during the process

At this point we use a trick to avoid computing the dot product in high dimensional feature space in equation 5, replacing it by a kernel function that satisfies Mercer's condition. Mercer's Theorem [12] guarantees the existence of this kernel function:

$$f(x) = \sum_{i=1}^{\ell} h_i \cdot k(x_i, x) + b. \tag{6}$$

where $h_i \equiv (\alpha_i - \alpha_i^*)$ and $k(x_i, x) = \langle \Phi(x_i) \cdot \Phi(x) \rangle$. Finally we note, regarding the sparsity of the SV expansion 5, that only the elements satisfying $|f(x_i) - y_i| \geq \epsilon$, where ϵ is the standard deviation of $f(x_i)$ from y_i (see selected loss function), have nonzero Lagrange multipliers α_i, α_i^*. This can be proved applying Karush-Kuhn-Tucher (KKT) conditions [13] to the SV dual optimization problem.

2.1 Regularization Theory

RT appeared in the methods for solving *ill posed problems* [1]. In RN we minimize a expression similar to equation 2. However, the search criterium is enforcing smoothness (instead of flatness) for the function in input space (instead of feature space). Thus we get:

$$R_{reg}[f] = R_{emp}[f] + \frac{\lambda}{2} ||\hat{P} f||^2. \tag{7}$$

where $\hat{P}$ denotes a regularization operator in the sense of [1], mapping from the Hilbert Space H of functions to a dot product Space D such as $\langle f, g \rangle \quad \forall f, g \in H$ is well defined. Applying Fréchet's differential[3] to equation 7 and the concept of Green's function of $\hat{P}^* \hat{P}$:

$$\hat{P}^* \hat{P} \cdot G(x_i, x_j) = \delta(x_i - x_j). \tag{8}$$

(here δ denotes the Dirac's δ, that is $\langle f, \delta(x_i) \rangle = f(x_i)$), we get [6]:

$$f(x) = \lambda \sum_{i=1}^{\ell} [y_i - f(x_i)]_\epsilon \cdot G(x, x_i). \tag{9}$$

The correspondence between SVM and RN (equations 6 and 9) is proved if and only if the Green's function G is an "admissible" kernel in the terms of Mercer's theorem [12],i.e. we can write G as:

$$G(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle \quad with \quad \Phi : x_i \rightarrow (\hat{P} G)(x_i, .). \tag{10}$$

Prove: Minimizing $||\mathbf{P} f||^2$ can be expressed as:

$$||\mathbf{P} f||^2 = \int dx (\mathbf{P} f)^2 = \int dx f(x) \mathbf{P}^* \mathbf{P} f(x) \tag{11}$$

we can expand f in terms of Green's function associated to **P**, thus we get:

[3] Generalized differentiation of a function: $dR[f] = \left[\frac{d}{d\rho} R[f + \rho h]\right]$, where $h \in H$.

$$\begin{aligned} ||\mathbf{P}f||^2 &= \sum_{i,j}^{N} h_i h_j \int dx G(x, x_i)\mathbf{P}^*\mathbf{P}G(x, x_j) \\ &= \sum_{i,j}^{N} h_i h_j \int dx G(x, x_i)\delta(x - x_j) \\ &= \sum_{i,j}^{N} h_i h_j G(x_j, x_i) \end{aligned} \tag{12}$$

then only if G is Mercer Kernel it correspond to a dot product in some feature space. Then minimizing 7 is equivalent to minimize 2†.

A similar prove of this connection can be found in [4]. Hence given a regularization operator, we can find an admissible kernel such that SV machine using it will enforce flatness in feature space and minimize the equation 7. Moreover, given a SV kernel we can find a regularization operator such that the SVM can be seen as a RN.

3 Online Endogenous Learning Machine Using Regularization Operators

In this section we show a new on-line RN based on "Resource Allocating Network" algorithms (RAN) [4] [5] which consist of a network using RBFs, a strategy for allocating new units (RBFs), using two part novelty condition [5]; input space selection and neural pruning using matrix decompositions such as SVD and QR with pivoting [6]; and a learning rule based on SRM as discuss in the previous sections. The pseudo-code of the new on-line algorithm is presented in section 3.1. Our network has 1 layer as is stated in equation 6. In terms of RBFs the latter equation can be expressed as:

$$f(x) = \sum_{i=1}^{N(t)} h_i \cdot \exp\left(-\frac{||x(t) - x_i(t)||^2}{2\sigma_i^2(t)}\right) + b. \tag{13}$$

where $N(t)$ is the number of neurons, $x_i(t)$ is the center of neurons and $\sigma_i(t)$ the radius of neurons, at time "t".

In order to minimize equation 7 we propose a regularization operator based on SVM philosophy. We enforce flatness in feature space, as described in section 2, using the regularization operator $||\hat{P}f||^2 \equiv ||\omega||^2$, thus we get:

$$R_{reg}[f] = R_{emp}[f] + \frac{\lambda}{2} \sum_{i,j=1}^{N(t)} h_i h_j k(x_i, x_j). \tag{14}$$

We assume that $R_{emp} = (y - f(x))^2$ we minimize equation 14 adjusting the centers and radius (gradient descend method $\Delta\chi = -\eta\frac{\partial R[f]}{\partial \chi}$, with simulated annealing [14]):

$$\Delta x_i = -2\frac{\eta}{\sigma_i}(x - x_i)h_i(f(x) - y)k(x, x_i) + \alpha \sum_{i,j=1}^{N(t)} h_i h_j k(x_i, x_j)(x_i - x_j). \tag{15}$$

and

[4] The principal feature of these algorithms is sequential adaptation of neural resources.

$$\Delta h_i = \tilde{\alpha}(t) f(x_i) - \eta(f(x) - y) k(x, x_i). \tag{16}$$

where $\alpha(t), \tilde{\alpha}(t)$ are scalar-valued "adaptation gain", related to a similar gain used in the stochastic approximation processes [15], as in these methods, it should decrease in time. The second summand in equation 15 can be evaluated in several regions inspired by the so called "divide-and-conquer" principle and used in unsupervised learning, i.e. competitive learning in self organizing maps [16] or in SVMs experts [17]. This is necessary because of volatile nature of time series, i.e. stock returns, switch their dynamics among different regions, leading to gradual changes in the dependency between the input and output variables [18]. Thus the super-index in the latter equation is redefined as:

$$N_c(t) = \{s_i(t) : ||x(t) - x_i(t)|| \leq \rho\}. \tag{17}$$

that is the set of neurons close to the current input.

3.1 Program Pseudo-Code

The structure of the algorithm is shown below as pseudo-code:

```
program online-algorithm
    (Note: to denotes current iteration; k denotes
    prediction horizon);
    Initialize parameters and variables
    Build input Toeplitz matrix A using (3W-1)
        input values
    Input space selection: determinate Np relevant
        lags L using SVD y QR_wp [8]
    Determinate input vector: x= x(to-k-L(1))
    while (true)
        if (n_rbfs > 0)
            Compute f(x)
            Find nearest RBF:|x-x_dmin|
        else
            f(x)=x(to-k-1)
        Calculate error: e=|f(x)-x(to)|
        if (e>epsilon and |x-x_d_min|>delta) [7]
            Add RBF with parameters:
            x_i=x, sigma_i=kappa*|x-c_dmin}|, h=e
        else
            Execute pruning (SVD & QR_wp to neural activations)[8]
            Update parameters minimizing actual risk
            (15}(16}
        if (e>theta*epsilon and n_inps<max_inps)
            n_inps = n_inps+1
            Determinate new lags: L=[L_1,L_2,...,L_Np]
            Add rbf_add RBFs
        to = to+1
    end
```

4 Experiments

The application of our network is to predict complex time series. We choose the high-dimensional chaotic system generated by the Mackey-Glass delay differential equation:

$$\frac{dx(t)}{dt} = -b \cdot x(t) + a \cdot \frac{x(t-\tau)}{1 + x^{10}(t-\tau)} \quad . \tag{18}$$

with $b = 0.1$, $a = 0.2$ and delay $t_d = 17$. This equation was originally presented as a model of blood regulation [19] and became popular in modelling time series benchmark. We add two modifications to equation 18: Zero-mean gaussian noise with standard deviation equal to 1/4 of the standard deviation of the original series and dynamics changes randomly in terms of delay (between 100-300 time steps) $t_d = 17, 23, 30$.

We integrated the chaotic model using MatLab software on a Pentium III at 850MHZ obtaining 2000 patterns. For our comparison we use 100 prediction results from SVM_online (presented in this paper), standard SVM (with ϵ-insensitive loss) and NAPA_PRED (RAN algorithm using matrix decompositions being one of the best on-line algorithms to date[6]).

Table 1. Evolution of NRMSE (Normalized Root Mean Square Error). Xth-S detones the Xth-step prediction NRMSE on the test set.

Method	1st-S	25th-S	50th-S	75-th	100th-S
NAPA_PRED	1.1982	0.98346	0.97866	0.91567	0.90985
Standard SVM	0.7005	0.7134	0.7106	0.7212	0.7216
SVM_online	0.7182	0.71525	0.71522	0.72094	0.7127

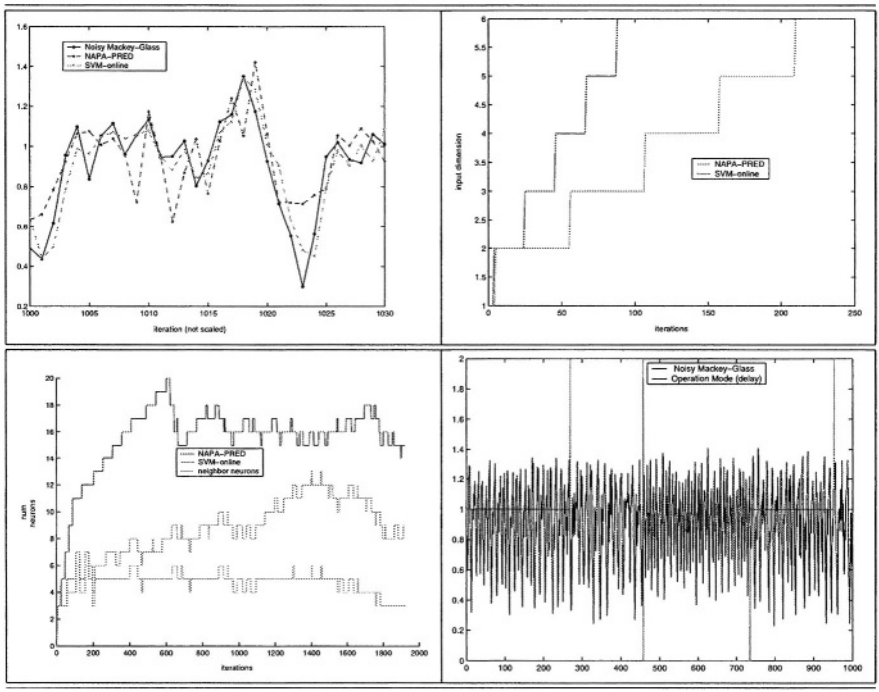
Clearly there's a remarkable difference between previous on-line algorithm and SVM philosophy. Standard SVM and SVM_online achieve similar results for this set of data at the beginning of the process. In addition, there's is noticeable improvement in the last iterations because of the volatile nature of the series. The change in time delay t_d, leads to gradual changes in the dependency between the input and output variables and, in general, it's hard for a single model including SVMs to capture such a dynamic input-output relationship inherent in the data. Focussing our attention on the on-line algorithm, we observe the better performance of the new algorithm as is shown in table 2.

5 Conclusions

Based on SRM and the principle of "divide and conquer" , a new online algorithm is developed by combining SVM and SOM using a resource allocating network and matrix decompositions. Minimizing the regularizated risk functional, using an operator the enforce flatness in feature space, we build a hybrid model

that achieves high prediction performance, comparing with the previous on-line algorithms for time series forecasting. This performance is similar to the one achieve by SVM but with lower computational time demand, essential feature in real-time systems. The benefits of SVM for regression choice consist in solving a – uniquely solvable – quadratic optimization problem, unlike the general RBF networks, which requires suitable non-linear optimization with danger of getting stuck in local minima. Nevertheless the RBF networks used in this paper, join various techniques obtaining high performance, even under extremely volatile conditions, since the level of noise and the change of delay operation mode applied to the chaotic dynamics was rather high.

Table 2. Figures: 1) The predicted and actual values in Noisy Mackey-Glass data for online algorithms. 2) Evolution of the input space dimension in SVM_online vs. NAPA_PRED. The higher capacity of generalization controls input space dimension. 3) Evolution of the neural resources in SVM_online vs. NAPA_PRED and number of neighboring neurons. 4) Noisy Mackey-Glass Series with changing operation mode.



References

1. Tikhonov, A.N., Arsenin, V.Y.:, Solutions of Ill-Posed Problems, Winston. Washington D.C., U.S.A. Berlin Heidelberg New York (1997) 415–438

2. Vapnik, V., Chervonenkis, A.: Theory of Pattern Recognition [in Russian]. Nauka, Moscow (1974).
3. Muller, K.R, Smola A.J., Ratsch, G., Scholkopf, B., Kohlmorgen, J: Using Support Vector Machines for time series prediction, In Advances in kernel Methods- Spport Vector Learning, MIT Press, Cambridge, MA. (1999) 243–254
4. Smola, A.J., Scholkopf, B., Muller, K.R: The connection between regularization operators and support vector kernels. Neural Networks, 11, 637–649
5. Platt, J.: A resource-allocating network for function interpolation. Neural Computation, 3, (1991) 213–225
6. Salmerón-Campos, M.: Predicción de Series Temporales con Rede Neuronales de Funciones Radiales y Técnicas de Descomposición Matricial. PhD Thesis, University of Granada, DEpartamento de Arquitectura y Tecnología de Computadores (2001)
7. Kolmogorov, A.N.: On the representation of continuous fucntions of several variables by superposition of continuos fucntions of one variable and addition. Dokl. Akad. Nauk USSR, vol 114, 953–956 (1957)
8. Muller, K.R, Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An Introduction to Kernel-Based Learning Algorithms. IEEE Transactions on neural Networks, vol 12, num 2, 181–201 (2001)
9. Poggio, T., Girosi, F.,: Regularization algorithms for learning that are equivalent to multilayer networks. Science, vol 247, 978–982 (1990)
10. Vapnik, V., Lerner, A.: Pattern Recognition using Generalized Portrait Method. Automation and Remote Control, vol 24, issue 6,(1963)
11. Vapnik, V.: Statistical Learning Theory. Wiley, N.Y. (1998)
12. Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. Philos. Trans. Roy. Soc. London , num 209, vol A, 415–446 (1909)
13. Kuhn, H.W.,Tucker, A.W.: Nonlinear Programming. In 2^{nd} Symposium on Mathematical Statistics and Probabilistics,University of California Press, 481–492 (1951)
14. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science, vol 220, num 4598, 671-680, (1983)
15. Kushner, H.J., Yin, G.:Stochastic Approximation Algorithms and Applications. Springer-Verlag, New York, U.S.A. (1997)
16. Kohonen, T.: The Self-Organizing Map. Proceedings of the IEEE, num 9, vol 78, 1464–1480 (1990)
17. Cao, L.: Support Vector Machines Experts for Time Series Forecasting- Neurocomputing, vol 51, 321–339 (2003)
18. Górriz, J.M. : Algorítmos Híbridos para la Modelización de Series Temporales con Técnicas AR-ICA. In Press PhD Thesis, University of Cádiz (2003)
19. Mackey, M.C., Glass, L., Science, 197 287–289 (1977)

Improved A-P Iterative Algorithm in Spline Subspaces*

Jun Xian[1], Shi-Ping Luo[2], and Wei Lin[1]

[1] Department of Mathematics, Sun Yat-sen University, Guangzhou, 510275, China, xianjun11@sohu.com
[2] Department of Mathematics, South China Normal University, Guangzhou, 510631, China, stslw@zsu.edu.cn

Abstract. In this paper, we improve A-P iterative algorithm, and use the algorithm to implement the reconstruction from weighted samples, and obtain explicit convergence rate of the algorithm in spline subspaces.

1 Introduction

For a bandlimited signal of finite energy, it is completely described by the famous classical Shannon sampling theorem. This classical theorem has broad application in signal processing and communication theory and has been generalized to many other forms. However, in many real applications sampling points are not always regularly. It is well-known that in the sampling and reconstruction problem for non-bandlimited spaces, signal is often assumed to belong to a shift-invariant spaces[l, 2, 4, 5, 7, 8, 9, 10]. As the special shift-invariant spaces, spline subspaces yield many advantages in their generation and numerical treatment so that there are many practical applications for signal or image processing[1, 2, 3,9],

For practical application and computation of reconstruction, Goh et al., show practical reconstruction algorithm of bandlimited signals from irregular samples in [11], Aldroubi et al., present a A-P iterative algorithm in [5]. In this paper, we improve the A-P iterative algorithm in spline subspaces. The improved algorithm occupies better convergence than the old one.

2 Improved A-P Iterative Algorithm in V_N

Aldroubi presented A-P iterative algorithm in [5]. In this section, we will improve the algorithm. The improved algorithm occupies faster convergence. We will discuss the cases of non-weighted samples and weighted samples, respectively.

We define some symbols. $V_N = \{\sum_{k\in Z} c_k\varphi_N(\cdot - k) : \{c_k\} \in \ell^2\}$ is spline subspace generated by $\varphi_N = \chi_{[0,1]} * \cdots * \chi_{[0,1]}$ (N convolutions), $N \geq 1$.

* This work is supported in part by the China-NSF, the Guangdong-NSF and the Foundation of Sun Yat-sen University Advanced Research Centre.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 58–64, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Let $\chi_{j,1}(x) = \chi_{[x_j, \frac{x_j+x_{j+1}}{2})}(x), \chi_{j,2}(x) = \chi_{[\frac{x_j+x_{j+1}}{2}, x_{j+1})}(x), \forall x_j \in X$ (X is sampling set),

$$W(L^p) = \{f \in L^p : \| f \|_{W(L^p)} = (\sum_{k\in Z^d} esssup_{x\in[0,1]^d} |f(x+k)|^p)^{(1/p)} < \infty\}$$

if $1 \le p < \infty$,

$$W(L^\infty) = \{f \in L^\infty : \| f \|_{W(L^p_v)} = \sup_{k\in Z^d} esssup_{x\in[0,1]^d} |f(x+k)| < \infty\}$$

if $p = \infty$. Let oscillation $osc_\delta(f)(x) = \sup_{|y|\le\delta} |f(x+y) - f(x)|$.

We show some lemmas that will be used in the proof of Theorem 2.1 , 2.2, and 2.3.

Lemma 2.1[5] *If φ is continuous and has compact support, then for any $f \in V^p(\varphi) = \{\sum_{k\in\mathbb{Z}} c_k\varphi(\cdot - k) : (c_k) \in \ell^p\}$ the conclusions (i)-(ii) hold:*

i. $\|f\|_{L^p} \approx \|c\|_{\ell^p} \approx \|f\|_{W(L^p)}$,

ii. $V^p(\varphi) \subset W_0(L^p) \subset W_0(L^q) \subset W(L^q) \subset L^q(\mathbb{R}) (1 \le p \le q \le \infty)$.

Lemma 2.2 *If $f \in V_N$, then for any $0 < \delta < 1$ we have* $\|osc_\delta(f)\|^2_{L^2} \le (3N\delta)^2 \sum_{k\in\mathbb{Z}} |c_k|^2$.

Proof. We have the following equalities and inequalities:

$$\|osc_\delta(f)\|_2^2 = \int_{-\infty}^{\infty} |\sup_{|y|\le\delta} |\sum_{n\in\mathbb{Z}} c_n(\varphi_N(x+y-n) - \varphi_N(x-n))||^2 dx$$

$$\le \int_{-\infty}^{\infty} (\sum_{n\in\mathbb{Z}} |c_n| \sup_{|y|\le\delta} |\varphi_N(x+y-n) - \varphi_N(x-n)|)^2 dx$$

$$= \int_0^1 \sum_{k\in\mathbb{Z}} (\sum_{m\in\mathbb{Z}} |c_{k-m}| \sup_{|y|\le\delta} |\varphi_N(x+y+m) - \varphi_N(x+m)|)^2 dx.$$

By induction method and properties of φ_N, we can easily check that

$$\sum_{k\in\mathbb{Z}} \sup_{x\in[0,1]} |\varphi'_N(x+k)| \le N \qquad (\triangle)$$

From $(\triangle)$ and properties of φ_N, we can obtain the following estimate:

$$\|osc_\delta(f)\|_2^2 \le \int_0^1 \sum_{k\in\mathbb{Z}} (\sum_{m\in\mathbb{Z}} |c_{k-m}|^2 \sup_{|y|\le\delta} |\varphi_N(x+y+m) - \varphi_N(x+m)|^2) dx$$

$$\le \sum_{k\in\mathbb{Z}} |c_k|^2 \int_0^1 (\sum_{m\in\mathbb{Z}} \sup_{|y|\le\delta} |\varphi_N(x+y+m) - \varphi_N(x+m)|)^2 dx$$

$$\leq \delta^2 \sum_{k\in\mathbb{Z}} |c_k|^2 \int_0^1 (\sum_{m\in\mathbb{Z}} \sup_{|y|\leq\delta} |\varphi_N'(x+\theta y+m)|)^2 dx$$

$$\leq \delta^2 \sum_{k\in\mathbb{Z}} |c_k|^2 \int_0^1 (\sum_{m\in\mathbb{Z}} \sup_{t\in[-1,2]} |\varphi_N'(t+m)|)^2 dx$$

$$\leq (3N\delta)^2 \sum_{k\in\mathbb{Z}} |c_k|^2.$$

The last inequality of the above inequalities derives from $(\triangle)$. From $x \in [0,1]$,$0 < \theta < 1$ and $0 < \delta < 1$, we have the forth inequality of the above inequalities.

Lemma 2.3[5] *For any $f \in V^p(\varphi)$ the following conclusions hold:*

1. $\|osc_\delta(f)\|_{W(L^p)} \leq \|c\|_{\ell^p} \|osc_\delta(\varphi)\|_{W(L^1)}$,

2. $\|\sum_{k\in\mathbb{Z}} c_k \varphi(\cdot - k)\|_{W(L^p)} \leq \|c\|_{\ell^p} \|\varphi\|_{W(L^1)}$.

Lemma 2.4 *If $X = \{x_n\}$ is real sequence with $\sup_i(x_{i+1} - x_i) = \delta < 1$, then for any $f = \sum_{k\in\mathbb{Z}} c_k \varphi_N(\cdot - k) \in V_N$ we have $\|Qf\|_{L^2} \leq \|Qf\|_{W(L^2)} \leq (3+\delta)\|c\|_{\ell^2} \|\varphi\|_{W(L^1)}$.*

Proof. : For $f = \sum_{k\in\mathbb{Z}} c_k \varphi_N(\cdot - k)$ we have

$$|f(x) - (Qf)(x)| = |\sum_{j\in\mathbb{Z}} (f(x) - f(x_j))\chi_{j,1}(x) + \sum_{j\in\mathbb{Z}} (f(x) - f(x_{j+1}))\chi_{j,2}(x)|$$

$$\leq \sum_{j\in\mathbb{Z}} |f(x) - f(x_j)|\chi_{j,1}(x) + \sum_{j\in\mathbb{Z}} |f(x) - f(x_{j+1})|\chi_{j,2}(x)$$

$$\leq \sum_{j\in\mathbb{Z}} osc_{\frac{\delta}{2}}(f)(x)\chi_{j,1}(x) + \sum_{j\in\mathbb{Z}} osc_{\frac{\delta}{2}}(f)(x)\chi_{j,2}(x)$$

$$= osc_{\frac{\delta}{2}}(f)(x).$$

From this pointwise estimate and Lemma 2.3 we get

$$\|f - Qf\|_{W(L^2)} \leq \|osc_{\frac{\delta}{2}}(f)\|_{W(L^2)}$$

$$\leq \|c\|_{\ell^2} \|osc_{\frac{\delta}{2}}(\varphi_N)\|_{W(L^1)}.$$

And by the results of [7] or [8] we know

$$\|osc_{\frac{\delta}{2}}(\varphi_N)\|_{W(L^1)} \leq (2+\delta)\|\varphi_N\|_{W(L^1)}.$$

Putting the above discuss together, we have

$$\|Qf\|_{L^2} \leq \|Qf\|_{W(L^2)} \leq \|f - Qf\|_{W(L^2)} + \|f\|_{W(L^2)}$$

$$\leq (2+\delta)\|c\|_{\ell^2} \|\varphi_N\|_{W(L^1)} + \|\sum_{k\in\mathbb{Z}} c_k \varphi_N(\cdot - k)\|_{W(L^2)}$$

$$\leq (2+\delta)\|c\|_{\ell^2} \|\varphi_N\|_{W(L^1)} + \|c\|_{\ell^2} \|\varphi_N\|_{W(L^1)}$$

$$\leq (3+\delta)\|c\|_{\ell^2} \|\varphi_N\|_{W(L^1)}.$$

The following Theorem 2.1 is one of our main theorems in this paper.

Theorem 2.1 *Let P be an orthogonal projection from $L^2(\mathbb{R})$ to V_N and $Q[f(x)] = \sum_j f(x_j)\chi_{j,1}(x) + \sum_j f(x_{j+1})\chi_{j,2}(x)$. If sampling set $X = \{x_n\}$ is a real sequence with $\sup_i(x_{i+1} - x_i) = \delta < 1$ and $\frac{3N\delta}{2\sqrt{\sum_k |\hat{\varphi}_N(\pi+2k\pi)|^2}} < 1$, then any $f \in V_N$ can be recovered from its samples $\{f(x_j) : x_j \in X\}$ on sampling set X by the iterative algorithm*

$$\begin{cases} f_1 = PQf, \\ f_{n+1} = PQ(f - f_n) + f_n. \end{cases}$$

The convergence is geometric, that is,

$$\|f_{n+1} - f\|_{L^2} \le \left(\frac{3N\delta}{2\sqrt{\sum_k |\hat{\varphi}_N(\pi + 2k\pi)|^2}}\right)^n \|f_1 - f\|_{L^2}.$$

Proof. By $Pf = f$ and $\|P\|_{op} = 1$, for any $f = \sum_{k\in\mathbb{Z}} c_k\varphi_N(\cdot - k) \in V_N$ we have

$$\begin{aligned} \|(I - PQ)f\|_{L^2}^2 &= \|Pf - PQf\|_{L^2}^2 \le \|P\|_{op}^2\|f - Qf\|_{L^2}^2 = \|f - Qf\|_{L^2}^2 \\ &= \int |\sum_{j\in\mathbb{Z}} f(x)\chi_{j,1}(x) + \sum_{j\in\mathbb{Z}} f(x)\chi_{j,2}(x) - \sum_{j\in\mathbb{Z}} f(x_j)\chi_{j,1}(x) \\ &\qquad - \sum_{j\in\mathbb{Z}} f(x_{j+1})\chi_{j,2}(x)|^2 dx \\ &\le \int (\sum_{j\in\mathbb{Z}} |f(x) - f(x_j)|\chi_{j,1}(x) + \sum_{j\in\mathbb{Z}} |f(x) - f(x_{j+1})|\chi_{j,2}(x))^2 dx \\ &= \int \sum_{j\in\mathbb{Z}} |f(x) - f(x_j)|^2\chi_{j,1}(x) + \sum_{j\in\mathbb{Z}} |f(x) - f(x_{j+1})|^2\chi_{j,2}(x)dx \\ &\le \int \sum_{j\in\mathbb{Z}} |osc_{\frac{\delta}{2}}(f)(x)|^2\chi_{j,1}(x) + \sum_{j\in\mathbb{Z}} |osc_{\frac{\delta}{2}}(f)(x)|^2\chi_{j,2}(x)dx \\ &= \|osc_{\frac{\delta}{2}}(f)\|_{L^2}^2 \le (3N\frac{\delta}{2})^2 \sum_{k\in\mathbb{Z}} |c_k|^2 = (3N\frac{\delta}{2})^2\|c\|_{\ell^2}^2 \\ &\le \left(\frac{3N\delta}{2\sqrt{\sum_k |\hat{\varphi}_N(\pi + 2k\pi)|^2}}\right)^2 \|f\|_{L^2}^2. \end{aligned}$$

The third equality is from property $\sum_j(\chi_{j,1}(x) + \chi_{j,2}(x)) = 1$. The forth equality derives from property $supp\chi_{j,i} \cap supp\chi_{k,l} = \emptyset (j \ne k, i \ne l)$. By Lemma 2.2, the forth inequality holds. And we have

$$\|f_{n+1} - f\|_{L^2} = \|f_n + PQ(f - f_n) - f\|_{L^2} = \|PQ(f - f_n) - (f - f_n)\|_{L^2}$$

$$\leq \|I - PQ\|\|f - f_n\|_{L^2} \leq \cdots \leq \|I - PQ\|^n \|f - f_1\|_{L^2}.$$

Combining with the estimate of $\|I - PQ\|$, we can imply

$$\|f_{n+1} - f\|_{L^2} \leq \left(\frac{3N\delta}{2\sqrt{\sum_k |\hat{\varphi}_N(\pi + 2k\pi)|^2}}\right)^n \|f_1 - f\|_{L^2}.$$

Taking assumption $\frac{3N\delta}{2\sqrt{\sum_k |\hat{\varphi}_N(\pi+2k\pi)|^2}} < 1$,we know the algorithm is convergent.

In the following, we will show improved A-P iterative algorithm from weighted samples.

Theorem 2.2 *Let P be an orthogonal projection from $L^2(\mathbb{R})$ to V_N and weight function satisfy the following three conditions (i)-(iii):*
(i) $supp\varphi_{x_j} \subset [x_j - \frac{a}{2}, x_j + \frac{a}{2}]$
(ii) *there exist $M > 0$ such that* $\int |\varphi_{x_j}(x)|dx \leq M$,
(iii) $\int \varphi_{x_j}(x)dx = 1$.
Let $Af(x) = \sum_j \langle f, \varphi_{x_j}\rangle \chi_{j,1}(x) + \sum \langle f, \varphi_{x_{j+1}}\rangle \chi_{j,2}(x)$. *If sampling set* $X = \{x_n\}$ *is a real sequence with* $0 < \sup_i(x_{i+1} - x_i) = \delta < 1$ *and we choice proper δ and a such that* $\frac{3N}{2\sqrt{\sum_k |\hat{\varphi}_N(\pi+2k\pi)|^2}}(\delta + a(3+a)M) < 1$, *then any $f \in V_N$ can be recovered from its weighted samples $\{\langle f, \varphi_{x_j}\rangle : x_j \in X\}$ on sampling set X by the iterative algorithm*

$$\begin{cases} f_1 = PAf, \\ f_{n+1} = PA(f - f_n) + f_n. \end{cases}$$

The convergence is geometric, that is,

$$\|f_{n+1} - f\|_{L^2} \leq \left(\frac{3N\delta}{2\sqrt{\sum_k |\hat{\varphi}_N(\pi + 2k\pi)|^2}}(\delta + a(3+a)M)\right)^n \|f_1 - f\|_{L^2}.$$

Proof. By $Pf = f$ and $\|P\|_{op} = 1$, for any $f = \sum_{k\in\mathbb{Z}} c_k \varphi_N(\cdot - k) \in V_N$ we have

$$\|f - PAf\|_{L^2} = \|f - PQf + PQf - PAf\|_{L^2}$$

$$\leq \|f - Qf\|_{L^2} + \|Qf - Af\|_{L^2} \qquad (1).$$

From the proof of Theorem 3.1, we have the following estimate for $\|f - Qf\|_{L^2}$:

$$\|f - Qf\|_{L^2} \leq \left(\frac{3N\delta}{2\sqrt{\sum_k |\hat{\varphi}_N(\pi + 2k\pi)|^2}}\right)\|f\|_{L^2} \qquad (2).$$

For the second term $\|Qf - Af\|_{L^2}$ of (1) we have the pointwise estimate

$$|(Qf - Af)(x)|$$

$$= |\sum_j (f(x_j) - \langle f, \varphi_{x_j}\rangle)\chi_{j,1}(x) + \sum_j (f(x_{j+1}) - \langle f, \varphi_{x_{j+1}}\rangle)\chi_{j,2}(x)|$$

$$= |\int \sum_j (f(x_j) - f(\xi))\varphi_{x_j}(\xi)\chi_{j,1}(x)$$

$$+ \sum_j (f(x_{j+1}) - f(\xi))\varphi_{x_{j+1}}(\xi)\chi_{j,2}(x)d\xi|$$

$$\leq M(\sum_j osc_{\frac{a}{2}}(f)(x_j)\chi_{j,1}(x) + \sum_j osc_{\frac{a}{2}}(f)(x_{j+1})\chi_{j,2}(x))$$

$$= MQ(\sum_{k\in\mathbb{Z}} |c_k| osc_{\frac{a}{2}}(\varphi_N)(x-k)).$$

The above second equality derives from $\int \varphi_{x_j}(x)dx = 1$. By $\int |\varphi_{x_j}(x)|dx \leq M$ and $supp\varphi_{x_j} \subset [x_j - \frac{a}{2}, x_j + \frac{a}{2}]$, we know the above first inequality.

From this pointwise estimate and Lemma 2.4, it follows that:

$$\|Qf - Af\|_{L^2} \leq M(3+a)\|c\|_{\ell^2}\|osc_{\frac{a}{2}}(\varphi_N)\|_{W(L^1)}$$

$$\leq M(3+a)\frac{\|osc_{\frac{a}{2}}(\varphi_N)\|_{W(L^1)}}{\sqrt{\sum_k |\hat{\varphi}_N(\pi+2k\pi)|^2}}\|f\|_{L^2}$$

$$\leq M(3+a)\frac{3Na}{2\sqrt{\sum_k |\hat{\varphi}_N(\pi+2k\pi)|^2}}\|f\|_{L^2} \qquad (3).$$

By combining (1),(2) and (3), we can obtain

$$\|I - PA\|_{L^2} \leq \frac{3N}{2\sqrt{\sum_k |\hat{\varphi}_N(\pi+2k\pi)|^2}}(\delta + a(3+a)M).$$

Similar to the procedure in the proof of Theorem 3.1, we have

$$\|f_{n+1} - f\|_{L^2} \leq (\frac{3N}{2\sqrt{\sum_k |\hat{\varphi}_N(\pi+2k\pi)|^2}}(\delta + a(3+a)M))^n \|f_1 - f\|_{L^2}.$$

Remark 1. Term $(\frac{1}{2})^n$ is added in the expression of convergence rate. This improves the velocity of convergence. From the construction of operator Q and A, we know why it appears in the expression of convergence rate.

The reconstruction algorithm in Theorem 2.1 and 2.2 require the existence of orthogonal projection from L^2 onto V_N. For this purpose, the following Theorem 2.3 will construct the orthogonal projection. We can find the similar proof of Theorem 2.3 in [5, 10].

Theorem 2.3 *Let* $X = \{x_n\}$ *be a real sequence such that* $0 < \sup_i(x_{i+1} - x_i) = \delta < 1$. *Then*

$$Pf = \sum_{x_j \in X} \langle f, k_{x_j} \rangle \tilde{k}_{x_j}$$

is orthogonal projection from L^2 *onto* V_N, *where* $\{k_{x_j}\}$ *and* $\{\tilde{k}_{x_j}\}$.

Remark 2. : The above improved A-P iterative algorithm maybe be generalized to the case of $L^p(\mathbb{R})$ and $V^p(\varphi)$ whenever generator φ belongs to $W_0(L^1)$. We will study it in future work.

3 Conclusion

In this paper we pay main attention on the weighted sampling and reconstruction in spline subspaces. We give some reconstruction methods from different weighted sampling in spline subspaces. The improved A-P iterative algorithm performs better than the old A-P algorithm. And we obtain the explicit convergence rate of the improved A-P iterative algorithm in spline subspaces.

References

1. Aldroubi, A., Gröchenig, K.: Beurling-Landau-type theorems for non-uniform sampling in shift invariant spline spaces. *J. Fourier. Anal. Appl,* 6(1) (2000) 93-103.
2. Sun, W. C., Zhou, X. W.: Average sampling in spline subspaces. *Appl. Math. Letter,* 15(2002)233-237.
3. Wang, J.: Spline wavelets in numerical resolution of partial differential equations, *International Conference on Wavelet Analysis and its application, AMS/IP Studies in Advanced Mathematics,* Vol 25(2002)257-277.
4. Chen, W., Itoh, S., Shiki, J.: On sampling in shift invariant spaces. *IEEE Trans. Information. Theory* 48(10) (2002)2802-2810.
5. Aldroubi, A., Gröchenig, K.: Non-uniform sampling and reconstruction in shift-invariant spaces. *SIAM Rev* 43(4) (2001)585-620.
6. Chui, C. K.: An introduction to Wavelet, Academic Press, New York,1992.
7. Aldroubi, A.: Non-uniform weighted average sampling and reconstruction in shift-invariant and wavelet spaces. *Appl. Comput. Harmon. Anal* 13(2002)156-161.
8. Aldroubi, A., Feichtinger,H.: Exact iterative reconstruction algorithm for multivate irregular sampled functions in spline-like spaces: The L_p theory. *Proc. Amer. Math. Soc* 126(9)(1998)2677-2686.
9. Xian, J., Lin, W.: Sampling and reconstruction in time-warped spaces and their applications. to appear in *Appl. Math. Comput,* 2004.
10. Xian, J., Qiang, X. F.: Non-uniform sampling and reconstruction in weighted multiply generated shift-invariant spaces. *Far. East. J. Math. Sci* 8(3)(2003)281-293
11. Goh, S. S., Ong, I. G. H.: Reconstruction of bandlimited signals from irregular samples. *Signal. Processing* 46(3)(1995)315-329.

Solving Differential Equations in Developmental Models of Multicellular Structures Expressed Using L-systems

Pavol Federl and Przemyslaw Prusinkiewicz

University of Calgary, Alberta, Canada

Abstract. Mathematical modeling of growing multicellular structures creates the problem of solving systems of equations in which not only the values of variables, but the equations themselves, may change over time. We consider this problem in the framework of Lindenmayer systems, a standard formalism for modeling plants, and show how parametric context-sensitive L-systems can be used to numerically solve growing systems of coupled differential equations. We illustrate our technique with a developmental model of the multicellular bacterium *Anabaena.*

1 Introduction

Recent advances in genetics have sparked substantial interest in the modeling of multicellular organisms and their development. Modeling information transfer through cell membranes is a vital aspect of these models. Diffusion of chemicals is one example of a transfer mechanism, and can be mathematically expressed as a system of ordinary differential equations (ODEs). Due to the developmental nature of the models, this system changes as the cells in the organism divide. Such *dynamically evolving systems of equations* are not easily captured by standard mathematical techniques [2].

The formalism of L-systems [6] lends itself well to modeling developmental processes in organisms. Prusinkiewicz et al. introduced *differential L-systems* (dL-systems) [10] as a notation for expressing developmental models that include growing systems of ODEs, but left open the problem of solving these equations. From the viewpoint of software organization these equations can be solved either using an external solver or within the L-system formalism itself. The first technique induces substantial overhead due to repetitive transfers of large amounts of data to and from the solver in each simulation step. As an alternative, we present a mechanism where the system of ODEs is internally maintained, updated, and solved by an L-system. We adapt to this end an implicit (Crank-Nicholson) integration scheme, whereas previous approaches only used simpler, explicit methods [2,7]. We illustrate our solution by revisiting the diffusion-based developmental model of the blue-green alga *Anabaena catenula* [1,8,11], defined using a dL-system [10].

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 65–72, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 L-systems and the L+C Language

In the formalism of L-systems [6], growing biological structures are represented as strings of *modules* [11]. The initial structure is the *axiom.* An L-system describes the development of this structure in terms of *rewriting rules* or *productions.* Each production replaces its the *predecessor* module by zero, one, or more *successor* modules. For example, the production $A \rightarrow AB$ replaces module A by a structure consisting of a new module A and a new module B. In general, productions can be *context free,* with productions matched only to the predecessor module, or *context-sensitive,* with productions matched also to the predecessor's neighbors. The context-sensitive productions make it possible to simulate information transfer within developing structures.

The algorithms presented in this paper are specified in the L+C programming language [4,5], which combines the declarative programming style of L-systems with C++ constructs. The L+C modules are declared by the keyword `module`, e.g. `module B(int, double)`. The initial string is preceded by the keyword `axiom`, e.g. `axiom: B(1,7.0)`. The body of a production, delimited by curly braces, may include any valid C++ statement. An example of a context-sensitive production is:

```
A(n) < B(i,j) > C() : { if (i < j) produce D(i+n,j-n); }
```

The body of this production is executed for every module `B` that has a module `A` on its left and `C` on its right side. If the parameter `i` of module `B` is less than its parameter `j`, the module `B` will be replaced by a module `D` with updated parameters. This is denoted by the keyword `produce` inside the `if` statement.

Although L-systems have been defined as a *parallel* rewriting mechanism, they are commonly implemented by *sequentially* scanning the predecessor string to obtain the successor string. In L+C we take advantage of this fact. The scanning direction is chosen at each derivation step by calling functions `Forward()` or `Backward()`. As the successor string is being generated, the *newly created* modules in the string can be used for context matching, using the symbols '`<<`' for the *new left context* and '`<<`' for the *new right context.* For example, consider a string of modules `A` with integer parameters `A(1)A(2)A(3)A(4)A(5)` and a production

```
A(nc) << A(n) : produce A(nc+n);
```

This string will be transformed into `A(1)A(3)A(6)A(10)A(15)` in a single derivation step.

Productions in L+C can be arranged into *groups* using the construct `group <n>: <productions>`, where `n` is an integer identifier. The active group of productions can be selected dynamically by calling the function `UseGroup(n)`. The L+C constructs `Start` and `StartEach` are used to denote blocks of C++ code that are executed at the beginning of the simulation and at the beginning of each derivation step. Both constructs are useful when the derivation of the string is divided into separate phases.

3 The *Anabaena* Model

Anabaena is a bacterium that forms non-branching multicellular filaments consisting of two classes of cells: *vegetative cells* and *heterocysts* [8]. A vegetative cell may divide into two vegetative cells, or differentiate into a heterocyst. In spite of the cell division, the spacing between heterocysts is relatively constant, with approximately 10 vegetative cells separating the heterocysts on the average Mathematical models of this phenomenon [1,7,11] postulate that the distribution of heterocysts is regulated by a compound produced by the heterocysts, diffusing from cell to cell along the filament, and decaying in the vegetative cells. If the compound concentration in a vegetative cell falls below a threshold level, this cell differentiates into a heterocyst.

A model operating in continuous time according to this description can be captured by the following dL-system [10]:

axiom: $F_h(s_{max}, c_{max})F_v(s_{max}, c_{max})F_h(s_{max}, c_{max})$
$F(s_l, c_l) < F_v(s, c) > F(s_r, c_r)$:
  if $s < s_{max}$ & $c > c_{min}$
    solve $dc/dt = D \cdot (c_l + c_r - 2c) - \mu c$ and $ds/dt = rs$
  if $s = s_{max}$ & $c > c_{min}$
    produce $F_v(ks_{max}, c)F_v((1-k)s_{max}, c)$
  if $c = c_{min}$
    produce $F_h(s, c)$
$F_h(s, c)$:
    solve $ds/dt = r_s(s_{max} - s)$ and $dc/dt = r_c(c_{max} - c)$

According to this model, a vegetative cell F_v or heterocyst F_h is characterized by its length s and compound concentration c. If the vegetative cell length s is below the maximum value s_{max} and the compound concentration c is above the threshold c_{min}, the concentration changes according to the equation $dc/dt = D \cdot (c_l - 2c + c_r) - \mu c$. This equation combines diffusion of the compound c according to Fick's law with the decay of c. The second differential equation characterizing F_v describes exponential elongation of cells according to the equation: $ds/dt = rs$.

In addition to these differential equations, two productions describe the behavior of a vegetative cell. If the cell reaches maximum length s_{max} while the concentration c is still above the threshold c_{min}, the cell will asymmetrically divide into two vegetative cells of length ks_{max} and $(1-k)s_{max}$, with the compound concentration c inherited from their parent cell. Otherwise, if the concentration c drops down to the threshold c_{min}, the cell will differentiate into a heterocyst. The last line of the dL-system specifies the behavior of heterocysts. Their length and compound concentration converge exponentially to the limit values of s_{max} and c_{max}, according to the equations: $dc/dt = r_c(c_{max} - c)$ and $ds/dt = r_s(s_{max} - s)$. The heterocysts do not undergo any further division.

4 Solving the dL-systems

A dL-system can be viewed as a formal statement of a problem that requires a solution. The simplest technique for numerically solving the ODEs from the previous section is the forward Euler method, where time derivatives are replaced with a forward finite difference approximation. This approach to solving differential equations in the framework of L-systems was presented in [3]; for the *Anabaena* dL-system it leads to the following system of equations:

$$(C^{i+1} - C^i)/\Delta t = D \cdot \left(C_l^i - 2C^i + C_r^i\right) - \mu C^i \ , \tag{1}$$

$$(S^{i+1} - S^i)/\Delta t = rS^i \ , \tag{2}$$

$$(C^{i+1} - C^i)/\Delta t = r_c(c_{max} - C^i) \ , \tag{3}$$

$$(S^{i+1} - S^i)/\Delta t = r_s(s_{max} - S^i) \ . \tag{4}$$

Here S^i and C^i denote the approximations of the cell length s and compound concentration c at time step i; time intervals have size Δt. Equations (1)-(4) define explicitly how the new values S^{i+1} and C^{i+1} are calculated using the previous values S^i and C^i. Subscripts l and r refer to the left and right neighbors of a given cell.

Although explicit integration is simple and can be easily expressed in L+C as a part of developmental models, it is inaccurate and prone to numerical instabilities. This is especially true for stiff differential equations, such as Equation (1) with a high diffusion constant D. Implicit schemes are much more appropriate for stiff equations. Discretizing the same ODEs using the implicit Crank-Nicholson scheme [9] yields:

$$(C^{i+1} - C^i)/\Delta t = D\frac{C_l^i - 2C^i + C_r^i + C_l^{i+1} - 2C^{i+1} + C_r^{i+1}}{2} - \mu\frac{C^i + C^{i+1}}{2} \ , \tag{5}$$

$$(S^{i+1} - S^i)/\Delta t = r(S^i + S^{i+1})/2 \ , \tag{6}$$

$$(C^{i+1} - C^i)/\Delta t = r_c\left(c_{max} - (C^i + C^{i+1})/2\right) \ , \tag{7}$$

$$(S^{i+1} - S^i)/\Delta t = r_s\left(s_{max} - (S^i + S^{i+1})/2\right) \ . \tag{8}$$

The relationships between the new and old values are now defined implicitly, by a system of linear equations. The coefficient matrix representing Equations (5) is tridiagonal, which leads to an efficient solution using a modified Gaussian elimination. Below we show that this elimination can be efficiently implemented using L-systems and L+C.

4.1 Solving Tridiagonal Systems of Linear Equations with L-systems

Let us consider a system of n linear equations with n unknowns described by a tridiagonal coefficient matrix. Such a system can be expressed as:

$$a_i X_{i-1} + b_i X_i + c_i X_{i+1} = y_i \ \text{ for } \ 1 \le i \le n \ , \tag{9}$$

where a_i, b_i, c_i and y_i are the coefficients of the equations, and X_i's denote the unknowns. This system of LEs can also be written in matrix form as $AX = Y$, where $X = [X_1, \cdots, X_n]^T$ is the column vector representing the unknowns, $Y = [y_1, \cdots, y_n]^T$ represents the constant terms, and A is the coefficient matrix:

$$A = \begin{pmatrix} b_1 & c_1 & 0 & 0 & \cdots & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & \cdots & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & b_{n-1} & c_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & a_n & b_n \end{pmatrix} \tag{10}$$

We solve this system of LEs in linear time using Gaussian elimination optimized to perform row subtractions only between non-zero coefficients. This process consists of two phases [9]. In the first phase, the coefficients below the diagonal are eliminated. Since the only coefficients of A below the diagonal are the coefficients a_i, this phase corresponds to calculating new values for coefficients a_i, b_i, c_i and y_i such that $a_i = 0$. To this end, the coefficients of each row are recalculated using the following substitutions:

$$a_i' \leftarrow 0 \ , \ b_i' \leftarrow b_i - \frac{a_i}{b_{i-1}'} c_{i-1}' \ , \ c_i' \leftarrow c_i \text{ and } y_i' \leftarrow y_i - \frac{a_i}{b_{i-1}'} y_{i-1}' \ . \tag{11}$$

These substitutions are applied iteratively, for i increasing from 2 to n. In the second phase the coefficients above the diagonal c_i are eliminated. This is accomplished by applying the following substitutions:

$$a_i' \leftarrow 0 \ , \ b_i' \leftarrow b_i \ , \ c_i' \leftarrow 0 \text{ and } y_i' \leftarrow y_i - \frac{c_i}{b_{i+1}'} y_{i+1}' \ . \tag{12}$$

These substitutions are applied in reverse order, for i decreasing from $n-1$ to 1. At the end of the second phase the coefficient matrix has non-zero entries strictly on the diagonal and the solution can be trivially found as $X_i = y_i / b_i$.

To implement this two-phase process using an L-system, we represent the system of tridiagonal LEs by a string of modules `C`. Each module `C` has a single parameter `p` of type `struct CD` that holds the non-zero entries of one row:

```
struct CD { double a, b, c, y; };
module C (struct CD);
```

The following L+C productions implement the two phases specified by Equations (11) and (12):

```
group 1: C(pl) << C(p) : { p.y=p.y-pl.y*p.a/pl.b;
           p.b=p.b-pl.c*p.a/pl.b; produce C(p); }
group 2: C(p) >> C(pr) : { p.y=p.y-pr.y*p.c/pr.b;
           produce C(p); }
```

Each phase is performed in a single sequential derivation step. We use a global variable `phase` to set the direction of processing at the beginning of each derivation step and to decide which group of productions will be applied during that phase:

```
int phase;
Start: { phase=1; }
StartEach: {
    if (phase == 1) { Forward (); UseGroup (1); phase=2; }
    else { Backward (); UseGroup (2); phase=1; } }
```

4.2 Implementation

The complete developmental model of *Anabaena* based on Equations (5)-(8) is obtained by complementing the solution to the diffusion Equation (5), presented above, with the solution to Equations (6)-(8). To solve Equation (5) we transform it to the form of Equation (9) using the substitutions:

$$a_0 = c_0 = a_n = c_n = 0 \ , \ b_0 = b_n = 1 \ , \tag{13}$$

$$y_0 = y_n = \frac{2C^i + 2s_{\max} r_s \Delta t - C^i \Delta t r_s}{2 + r_s \Delta t} \ , \tag{14}$$

$$a_i = c_i = -D \ , \ b_i = \frac{1}{\Delta t} + 2D + \mu \ , \tag{15}$$

$$y_i = \frac{2C^i}{\Delta t} + D \cdot \left(C_l^i - 2C^i + C_r^i\right) - \mu C^i \ . \tag{16}$$

for $1 < i < n$. The resulting system of LEs is solved using the method outlined in Sec. 4.1. Since no information transfer is involved in Equations (6)-(8), they can be expressed in closed form as:

$$S_{veg}^{i+1} = \frac{2 + r\Delta t}{2 - r\Delta t} S^i \ , \tag{17}$$

$$C_{het}^{i+1} = \frac{2C^i + 2c_{max} r_c \Delta t - C^i r_c \Delta t}{2 + r_c \Delta t} \ , \tag{18}$$

$$S_{het}^{i+1} = \frac{2S^i + 2s_{max} r_s \Delta t - S^i r_s \Delta t}{2 + r_s \Delta t} \ . \tag{19}$$

To implement the solution based on these formulas using an L-system, we represent the growing filament again as a string of modules `C`. The parameter of the module `C` is of type `struct CD`, declared as:

```
struct CD { double s, con, a, b, c, y; bool h; };
```

The fields `s` and `con` denote the size of the cell and the concentration of the diffusing compound, respectively. The fields `a, b, c` and `y` represent the coefficients of the system of LEs described in Sec. 4.1. We use five logical phases to update the state of this filament in a time step Δt. In the first phase the new cell sizes are calculated using (17), and the system of LEs is initialized according to Equations (13)-(16). This phase is implemented by two productions:

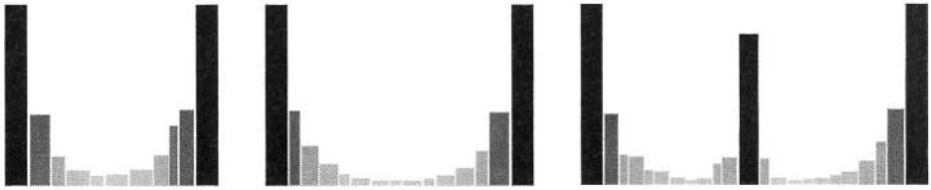


Fig. 1. Simulation results obtained using the implementation based on the Crank-Nicholson integration scheme. The following simulation constants were used: $s_{max} = 1$, $c_{max} = 255$, $c_{min} = 5$, $D = \mu = 0.03$, $r = 0.01$, $k = 0.37$, $r_s = 0.1$, $r_c = 0.15$, $\Delta t = 28.57$. Each cell is represented by a rectangle whose width and height correspond to the cell's size and concentration of the compound, respectively. The tallest dark bars represent the heterocysts, the lighter shaded bars represent vegetative cells.

```
group 1: C(pl) < C(p) > C(pr): {
    if (p.h) {
        p.y=(2*p.con+2*cmax*dt*rc-p.con*dt*rc)/(2+dt*rc);
        p.s=(2*p.s+2*smax*dt*rs-p.s*dt*rs)/(2+dt*rs);
        p.a=0; p.b=1; p.con=0; produce C(p); }
    else {
        p.a=-D; p.b=2/dt+2*D+mu; p.c=-D;
        p.y=2*p.con/dt+D*(cl.con-2*p.con+cr.con) - mu*p.con;
        p.s=(2+r*dt)/(2-r*dt)*p.s; produce C(p); } }
  C(p) : { p.a=p.c=0; p.b=1; p.y=p.con; produce C(p); }
```

The first production is applied to every module except the first and last. The second production is applied to the first and last modules. The LEs are then solved in the next two phases, using productions identical to those given in the previous section. In the fourth phase the solution is extracted and the fifth phase implements division and differentiation of cells:

```
group 4: C(p): { p.con=p.y/p.b; produce C(p); }
group 5: C(pl) < C(p) > C(pr) : {
    if (p.h) { produce C(p); }
    else if (p.s >= smax && p.con > cmin) {
        CD p1=p; p1.s=k*p.s; CD p2=p; p2.s=(1-k)*p.s;
        produce C(p1) C(p2); }
    else if (p.con < cmin) { p.h=true; produce C(p); }
}
```

The switching between the five phases is accomplished by extending the method discussed in Sec. 4.1. The initial steps of a simulation obtained using the above L-system are shown in Fig.1.

5 Conclusions

The modeling of developing organisms raises the problem of formulating mathematical models in which the set of variables that describes the system, and the

system of equations that relate these variables, dynamically change [2]. We presented a method for automatically modifying and solving these equations using the formalism of L-systems. The underlying numerical integration method is the Crank-Nicholson method [9]. We illustrated our approach using a model of a filamentous bacterium *Anabaena.* This organism captures the essential elements of the development of a multicellular structure: division and differentiation of cells, and signaling between cells. Although we have only considered a very simple model for illustrative purposes, the use of the implicit method for solving the ODEs is essential to extensions of this model. For example, the process of gradual cell division, during which the diffusion constants between incompletely divided cells are large, can easily be incorporated into the model. Other potential applications of the proposed method include biomechanical and functional-structural models of plant architecture.

Acknowledgments. We thank Brendan Lane, Lynn Mercer and Colin Smith for editorial help. The support from the Human Frontier Science Program and the National Sciences and Engineering Research Council of Canada is gratefully acknowledged.

References

1. C. G. de Koster and A. Lindenmayer. Discrete and continuous models for heterocyst differentiation in growing filaments of blue-green bacteria. *Acta Biotheoretica,* 36:249–273, 1987.
2. J.-L. Giavitto and O. Michel. Modeling the topological organization of cellular processes. BioSystems 2003, 70:149-163.
3, M. Hammel and P. Prusinkiewicz. Visualization of developmental processes by extrusion in space-time. *Proceedings of Graphics Interface '96,* pp 246–258, 1996.
4. R. Karwowski. *Improving the process of plant modeling: The L+C modeling language.* PhD thesis, University of Calgary, 2002.
5. R. Karwowski and P. Prusinkiewicz. Design and implementation of the L+C modeling language. *Electronic Notes in Theoretical Computer Science,* 86.2, 2003.
6. A. Lindenmayer. Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology,* 18:280–315, 1968.
7. A. Lindenmayer. Adding continuous components to L-systems. In G. Rozenberg and A. Salomaa, editors, *L Systems,* Lecture Notes in Computer Science 15, pages 53–68. Springer-Verlag, Berlin, 1974.
8. G. J. Mitchison and M. Wilcox. Rules governing cell division in *Anabaena. Nature,* 239:110–111, 1972.
9. W. H. Press, S. A. Teukolsky, and W. T. Wetterling. *Numerical recipes in C: The art of scientific computing. Second edition.* Cambridge University Press, 1988.
10. P. Prusinkiewicz, M. Hammel, and E. Mjolsness. Animation of plant development. Proceedings of SIGGRAPH 93 (Anaheim, California, August 1–6, 1993). ACM SIGGRAPH, New York, 1993, pp. 351–360.
11. P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants.* Springer-Verlag, New York, 1990. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.

On a Family of A-stable Collocation Methods with High Derivatives

Gennady Y. Kulikov[1], Arkadi I. Merkulov[2], and Ekaterina Y. Khrustaleva[2]

[1] School of Computational and Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, Johannesburg, South Africa
gkulikov@cam.wits.ac.za
[2] Ulyanovsk State University, L. Tolstoy Str. 42, 432970 Ulyanovsk, Russia
merkul@vda.ru, shabalkina@mail.ru

Abstract. In this paper we develop a family of A-stable one-step methods with high derivatives by means of a collocation technique. We present construction details and a theory to justify such sort of methods. We also concentrate on an effective way of their practical implementation.

1 Introduction

In this paper we study one-step numerical methods for ordinary differential equations (ODEs) of the form

$$x'(t) = g\big(t, x(t)\big), \quad t \in [t_0, t_0 + T], \quad x(t_0) = x^0 \tag{1}$$

where $x(t) \in \mathbf{R}^n$ and $g : D \subset \mathbf{R}^{n+1} \to \mathbf{R}^n$ is a sufficiently smooth function. Problem (1) is quite usual in theoretical research and in practical engineering (see, for example, [2], [4], [6], [7], [12]). Therefore efficient methods for finding its solution with any set accuracy (up to round-off errors) are important as in theory as in practice.

Here, we present a family of one-step collocation methods with high derivatives. Note that such topic of research proved its efficiency. So, there are many papers in this field (see [5], [7], [8], [11] and so on). Especially, we pay attention to the general linear methods developed by Butcher [4] which also include high derivatives. In general, all that methods are implicit. Therefore the most difficult task is their correct and effective implementation because we have to involve an additional iterative scheme.

The outline of this paper is organized as follows: We present the family of one-step collocation methods with high derivatives in Sect. 2. There, we also study convergence and A-stability of these methods. An effective implementation based on Newton-like iterations is given in Sect. 3. The last section of the paper is devoted to the problem of step size selection. Theoretical results are clearly confirmed by numerical experiments.

2 Collocation Methods with High Derivatives

As in [1], we use the collocation technique to construct numerical methods with high derivatives. First, we fix the points t_k, $t_{k+1/2}$ and t_{k+1} on an arbitrary

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 73–80, 2004.
© Springer-Verlag Berlin Heidelberg 2004

interval $[t_k, t_{k+1}]$ as collocation ones. Second, we assume that we know derivatives of the right-hand part of problem (1) at the nodes t_k and t_{k+1} up to order p. Now we can construct the Hermit interpolation formula (see, for example, [3]) and substitute it into the right-hand part of system (1) instead of $g\big(t, x(t)\big)$. Integrating the newly obtained system on the interval $[t_k, t_{k+1}]$ we come to the numerical method

$$x_{k+1/2} = x_k + \tau_k \sum_{r=0}^{p} \tau_k^r \Big(a_1^{(r)} g_k^{(r)} + a_3^{(r)} g_{k+1}^{(r)} \Big) + \tau_k a_2 g_{k+1/2}^{(0)}, \tag{2a}$$

$$x_{k+1} = x_k + \tau_k \sum_{r=0}^{p} \tau_k^r \Big(b_1^{(r)} g_k^{(r)} + b_3^{(r)} g_{k+1}^{(r)} \Big) + \tau_k b_2 g_{k+1/2}^{(0)}, \tag{2b}$$

$k = 0, 1, \ldots, K-1$, where

$$a_j^{(r)} = \sum_{i=0}^{p-r} \frac{1}{i!r!} \left[\frac{(\theta - (j-1)/2)^{p+1}}{\Psi(\theta)} \right]^{(i)}_{\theta=(j-1)/2} \int_0^{1/2} \frac{\Psi(\theta)}{(\theta - (j-1)/2)^{p-i-r+1}} d\theta,$$

$$b_j^{(r)} = \sum_{i=0}^{p-r} \frac{1}{i!r!} \left[\frac{(\theta - (j-1)/2)^{p+1}}{\Psi(\theta)} \right]^{(i)}_{\theta=(j-1)/2} \int_0^{1} \frac{\Psi(\theta)}{(\theta - (j-1)/2)^{p-i-r+1}} d\theta$$

when $j = 1, 3$ and $r = 0, 1, \ldots, p$,

$$a_2 = \frac{(-1)^{p+1}}{4^{p+1}} \int_0^{1/2} \theta^{p+1} (\theta - 1)^{p+1} d\theta, \quad b_2 = \frac{(-1)^{p+1}}{4^{p+1}} \int_0^{1} \theta^{p+1} (\theta - 1)^{p+1} d\theta,$$

$x_0 = x^0$, $\theta = (t - t_k)/\tau_k$, $\Psi(\theta) = \theta^{p+1}(\theta - 1/2)(\theta - 1)^{p+1}$, $g_k^{(r)}$ denotes the r-th derivative[1] of the right-hand part of problem (1) with respect to t evaluated at the point t_k, and τ_k is a step size which may be fixed or variable.

The results presented above give a way to derive

Theorem 1 *Let the right-hand part of ODE* (1) *possess continuous derivatives up to order* $2p+5$ *in a neighborhood of the solution* $x(t)$ *on the interval* $[t_0, t_0+T]$, *where* p *is a nonnegative integer number. Then method* (2) *is convergent, it has stage order* $2p+3$ *and classical order* $2p+4$, *and its coefficients satisfy*

$$a_1^{(r)} = \frac{p+1}{r!2^{p+r+2}} \sum_{i=0}^{p-r} \sum_{l=0}^{i+r} \sum_{j=0}^{p+1} \sum_{q=0}^{i} \frac{(-1)^l (i+r)!(p+q)!}{l!(i+r-l)!j!(p+1-j)!(l+j+2)q!2^q}, \tag{3a}$$

$$a_2 = \frac{(p+1)!}{2} \sum_{l=0}^{p+1} \frac{(-1)^l}{l!(p+1-l)!(2l+1)}, \tag{3b}$$

[1] Here and further the zero-derivative implies the original function.

$$a_3^{(r)} = \frac{(-1)^{r+1}(p+1)}{r!2^{p+r+2}} \sum_{i=0}^{p-r} \sum_{l=0}^{i+r} \sum_{j=0}^{p+1} \frac{(-1)^j (i+r)!}{l!(i+r-l)!j!(p+1-j)!(l+j+2)} \times \sum_{q=0}^{i} \frac{(p+q)!}{q!2^q}, \quad r = 0, 1, \ldots, p, \tag{3c}$$

$$b_1^{(r)} = a_1^{(r)} + (-1)^r a_3^{(r)}, \quad b_2 = 2a_2, \quad b_3^{(r)} = (-1)^r a_1^{(r)} + a_3^{(r)}. \tag{3d}$$

Theorem 2 *One-step method* (2) *is A-stable for any integer* $p \geq 0$.

Further we call methods of the form (2) as *E-methods with high derivatives,* for short. As an example of E-methods, we present the following numerical method of order 8 obtained by Theorem 1 when $p = 2$:

$$x_{k+1/2} = x_k + \frac{689}{2240}\tau_k g_k^{(0)} + \frac{169}{4480}\tau_k^2 g_k^{(1)} + \frac{17}{8960}\tau_k^3 g_k^{(2)} - \frac{81}{2240}\tau_k g_{k+1}^{(0)} + \frac{41}{4480}\tau_k^2 g_{k+1}^{(1)} - \frac{19}{26880}\tau_k^3 g_{k+1}^{(2)} + \frac{8}{35}\tau_k g_{k+1/2}^{(0)}, \tag{4a}$$

$$x_{k+1} = x_k + \frac{57}{210}\tau_k \left(g_k^{(0)} + g_{k+1}^{(0)}\right) + \frac{1}{35}\tau_k^2 \left(g_k^{(1)} - g_{k+1}^{(1)}\right) + \frac{1}{840}\tau_k^3 \left(g_k^{(2)} + g_{k+1}^{(2)}\right) + \frac{16}{35}\tau_k g_{k+1/2}^{(0)}. \tag{4b}$$

3 Implementation of E-methods with High Derivatives

When implementing method (2) in practice, we have to involve some additional iterative scheme to treat the corresponding nonlinear algebraic system. We applied the simple (fixed-point) iteration and the modified (or full) Newton one in [10] to obtain the following combined algorithms:

E-method with the simple iteration (*SI-method*):

$$X_{k+1}^i = \bar{G}_k^\tau X_{k+1}^{i-1}, \quad i = 1, 2, \ldots, N, \tag{5a}$$

$$X_{k+1}^0 = \left(\Pi(t_{k+1/2})^T, \Pi(t_{k+1})^T\right)^T \in \mathbf{R}^{2n}, \quad k = 0, 1, \ldots, K-1, \tag{5b}$$

where $X_{k+1} \stackrel{\text{def}}{=} \left((x_{k+1/2})^T, (x_{k+1})^T\right)^T \in \mathbf{R}^{2n}$, the mapping $\bar{G}_k^\tau : \mathbf{R}^{2n} \to \mathbf{R}^{2n}$ presents the right-hand part of system (4), $\bar{x}_k$ is an approximate solution of problem (2) obtained after N iterations; i.e., $\bar{x}_k \stackrel{\text{def}}{=} x_k^N$, $\bar{g}_k^{(r)} \stackrel{\text{def}}{=} g^{(r)}(t_k, \bar{x}_k)$, and $\Pi(t)$ implies a predictor which is an interpolation formula based on the values of numerical solution $\bar{x}_k$ ($\Pi(t) \equiv \bar{x}_k$ in the trivial case).

E-method with the Newton iteration (N-method):

$$X_{k+1}^i = X_{k+1}^{i-1} - \partial \bar{F}_k^\tau (X_{k+1}^{i-1})^{-1} \bar{F}_k^\tau X_{k+1}^{i-1}, \quad i = 1, 2, \ldots, N, \tag{6a}$$

$$X_{k+1}^0 = \left(\Pi(t_{k+1/2})^T, \Pi(t_{k+1})^T\right)^T \in \mathbf{R}^{2n}, \quad k = 0, 1, \ldots, K-1, \tag{6b}$$

where $\bar{F}_k^\tau \stackrel{\text{def}}{=} E_{2n} - \bar{G}_k^\tau$ (E_{2n} is the identity operator in $\mathbf{R}^{2n}$) and $\partial\bar{F}_k^\tau(X_{k+1}^{i-1})$ is the Jacobian of the mapping $\bar{F}_k^\tau$ evaluated at the point X_{k+1}^{i-1}.

E-method with the modified Newton iteration (MN-method):

$$X_{k+1}^i = X_{k+1}^{i-1} - \partial\bar{F}_k^\tau(X_{k+1}^0)^{-1}\bar{F}_k^\tau X_{k+1}^{i-1}, \quad i = 1, 2, \ldots, N, \tag{7a}$$

$$X_{k+1}^0 = \left(\Pi(t_{k+1/2})^T, \Pi(t_{k+1})^T\right)^T \in \mathbf{R}^{2n}, \quad k = 0, 1, \ldots, K-1. \tag{7b}$$

Let us test our combined algorithms with the underlying method (4) on the restricted three body problem

$$x_1''(t) = x_1(t) + 2x_2'(t) - \mu_1\frac{x_1(t)+\mu_2}{y_1(t)} - \mu_2\frac{x_1(t)-\mu_1}{y_2(t)}, \tag{8a}$$

$$x_2''(t) = x_2(t) - 2x_1'(t) - \mu_1\frac{x_2(t)}{y_1(t)} - \mu_2\frac{x_2(t)}{y_2(t)}, \tag{8b}$$

$$y_1(t) = \left((x_1(t)+\mu_2)^2 + x_2(t)^2\right)^{3/2}, \quad y_2(t) = \left((x_1(t)-\mu_1)^2 + x_2(t)^2\right)^{3/2}, \tag{8c}$$

where $t \in [0, T]$, $T = 17.065216560157962558891$, $\mu_1 = 1 - \mu_2$ and $\mu_2 = 0.012277471$. The initial values of problem (8) are: $x_1(0) = 0.994$, $x_1'(0) = 0$, $x_2(0) = 0$, $x_2'(0) = -2.0015851063790825224 0$. The solution path of this problem is a periodic one with the period T (see, for example, [6]).

Table 1. Global errors of the EN-algorithm with trivial predictor for E-method (4)

N	K				
	$1.000 \cdot 10^{+04}$	$2.000 \cdot 10^{+04}$	$4.000 \cdot 10^{+04}$	$8.000 \cdot 10^{+04}$	$1.600 \cdot 10^{+05}$
1	$2.068 \cdot 10^{+00}$	$1.571 \cdot 10^{+00}$	$2.066 \cdot 10^{+00}$	$2.000 \cdot 10^{+00}$	$2.612 \cdot 10^{+00}$
2	$2.193 \cdot 10^{-02}$	$6.138 \cdot 10^{-05}$	$1.587 \cdot 10^{-07}$	$4.408 \cdot 10^{-10}$	$1.043 \cdot 10^{-12}$
3	$2.582 \cdot 10^{-04}$	$2.059 \cdot 10^{-07}$	$6.728 \cdot 10^{-10}$	$4.039 \cdot 10^{-13}$	$7.865 \cdot 10^{-14}$
4	$2.582 \cdot 10^{-04}$	$2.059 \cdot 10^{-07}$	$6.727 \cdot 10^{-10}$	$3.974 \cdot 10^{-13}$	$7.865 \cdot 10^{-14}$

Table 2. Global errors of the EMN-algorithm with trivial predictor for E-method (4)

N	K				
	$1.000 \cdot 10^{+04}$	$2.000 \cdot 10^{+04}$	$4.000 \cdot 10^{+04}$	$8.000 \cdot 10^{+04}$	$1.600 \cdot 10^{+05}$
1	$2.068 \cdot 10^{+00}$	$1.571 \cdot 10^{+00}$	$2.066 \cdot 10^{+00}$	$2.000 \cdot 10^{+00}$	$2.612 \cdot 10^{+00}$
2	$1.841 \cdot 10^{+00}$	$2.938 \cdot 10^{-01}$	$8.422 \cdot 10^{-03}$	$2.615 \cdot 10^{-04}$	$8.146 \cdot 10^{-06}$
3	$3.716 \cdot 10^{-03}$	$2.582 \cdot 10^{-05}$	$1.192 \cdot 10^{-07}$	$4.959 \cdot 10^{-10}$	$2.401 \cdot 10^{-12}$
4	$4.660 \cdot 10^{-04}$	$3.322 \cdot 10^{-07}$	$1.264 \cdot 10^{-10}$	$3.025 \cdot 10^{-13}$	$5.085 \cdot 10^{-14}$
5	$2.589 \cdot 10^{-04}$	$2.060 \cdot 10^{-07}$	$6.726 \cdot 10^{-11}$	$4.651 \cdot 10^{-13}$	$4.150 \cdot 10^{-14}$

Table 3. Global errors of the ESI-algorithm with trivial predictor for E-method (4)

N	K				
	$1.000 \cdot 10^{+04}$	$2.000 \cdot 10^{+04}$	$4.000 \cdot 10^{+04}$	$8.000 \cdot 10^{+04}$	$1.600 \cdot 10^{+05}$
3	$5.689 \cdot 10^{+00}$	$1.254 \cdot 10^{+00}$	$2.098 \cdot 10^{-01}$	$2.661 \cdot 10^{-02}$	$3.416 \cdot 10^{-03}$
4	$9.384 \cdot 10^{-01}$	$1.360 \cdot 10^{-01}$	$8.004 \cdot 10^{-03}$	$4.604 \cdot 10^{-04}$	$2.738 \cdot 10^{-05}$
5	$1.971 \cdot 10^{-01}$	$6.712 \cdot 10^{-03}$	$2.045 \cdot 10^{-04}$	$6.314 \cdot 10^{-06}$	$1.964 \cdot 10^{-07}$
6	$8.074 \cdot 10^{-03}$	$1.655 \cdot 10^{-04}$	$2.605 \cdot 10^{-06}$	$3.924 \cdot 10^{-08}$	$5.952 \cdot 10^{-10}$
7	$1.900 \cdot 10^{-03}$	$1.944 \cdot 10^{-05}$	$1.742 \cdot 10^{-07}$	$1.427 \cdot 10^{-09}$	$1.155 \cdot 10^{-11}$
8	$1.225 \cdot 10^{-03}$	$4.420 \cdot 10^{-06}$	$1.622 \cdot 10^{-08}$	$6.113 \cdot 10^{-11}$	$5.988 \cdot 10^{-13}$
9	$3.920 \cdot 10^{-04}$	$5.595 \cdot 10^{-07}$	$8.233 \cdot 10^{-10}$	$1.136 \cdot 10^{-12}$	$1.060 \cdot 10^{-13}$
10	$2.817 \cdot 10^{-04}$	$2.309 \cdot 10^{-07}$	$8.464 \cdot 10^{-11}$	$3.899 \cdot 10^{-13}$	$1.206 \cdot 10^{-13}$

Tables 1–3 exhibit that all the methods under testing are convergent for problem (8) and their orders depend on the number of iterations per grid point. We also see that the full (or modified) Newton iteration demonstrates the maximum order convergence with fewer iteration steps than the simple one. Unfortunately, it may be too expensive to compute the exact Jacobian of method (2) because of the high derivatives. Now we show how to simplify it.

First of all we replace the exact Jacobian $\partial \bar{F}_k^\tau$ in method (6) with the following matrix:

$$A_k \stackrel{\text{def}}{=} \begin{pmatrix} 1 - \tau_k a_2^{(0)} \partial g_{k+1/2} & -\tau_k a_3^{(0)} \partial g_{k+1} \\ -\tau_k b_2^{(0)} \partial g_{k+1/2} & 1 - \tau_k b_3^{(0)} \partial g_{k+1} \end{pmatrix},$$

and obtain *the E-method with the simplified Newton iteration (ESN-method).* If we test the new method on our problem (8) we come to Table 4. It is obvious that the ESN-method is cheaper than the previous Newton-type methods, but we see that it acts like the EMN-method. The corresponding theoretical result is given by

Table 4. Global errors of the ESN-algorithm with trivial predictor for E-method (4)

N	K				
	$1.000 \cdot 10^{+04}$	$2.000 \cdot 10^{+04}$	$4.000 \cdot 10^{+04}$	$8.000 \cdot 10^{+04}$	$1.600 \cdot 10^{+05}$
1	$2.102 \cdot 10^{+00}$	$1.862 \cdot 10^{+00}$	$2.056 \cdot 10^{+00}$	$1.989 \cdot 10^{+00}$	$1.839 \cdot 10^{+00}$
2	$2.976 \cdot 10^{+00}$	$2.467 \cdot 10^{-01}$	$1.024 \cdot 10^{-02}$	$3.774 \cdot 10^{-04}$	$1.304 \cdot 10^{-05}$
3	$2.347 \cdot 10^{-02}$	$3.725 \cdot 10^{-04}$	$5.377 \cdot 10^{-06}$	$8.578 \cdot 10^{-08}$	$1.393 \cdot 10^{-09}$
4	$2.066 \cdot 10^{-04}$	$1.274 \cdot 10^{-07}$	$9.618 \cdot 10^{-11}$	$1.803 \cdot 10^{-12}$	$1.140 \cdot 10^{-13}$
5	$2.571 \cdot 10^{-04}$	$2.059 \cdot 10^{-07}$	$6.739 \cdot 10^{-11}$	$4.583 \cdot 10^{-13}$	$4.428 \cdot 10^{-14}$

Theorem 3 *Let the right-hand part of ODE* (1) *possess continuous derivatives up to order* $2p+5$ *in a neighborhood of the solution* $x(t)$ *on the interval* $[t_0, t_0+T]$,

where p is a nonnegative integer number. Then the ESN-algorithm based on method (2) *is convergent, and its error satisfies*

$$\|x(t_k) - \bar{x}_k(N)\| \le C_\mu \tau^\mu, \quad k = 1, 2, \ldots, K, \tag{9}$$

where $\mu = \min\{(\xi+1)(N+1) - 2, 2p+4\}$, $\xi = \min\{\zeta+1, 2p+3\}$, ζ is the order of predictor $\Pi(t)$ which is used to compute an initial approximation in the simplified Newton iteration, $\tau \stackrel{\text{def}}{=} \max\{\tau_k\}$ is a diameter of the grid, and C_μ is a constant.

4 Step Size Control for E-methods with High Derivatives

Now we consider the problem of step size selection for E-methods with high derivatives. Let us assume that the exact solution of problem (1) is known at the point t_k; i. e., $x_k = x(t_k)$, and the local error tolerance ϵ_l is given. The notation $\hat{x}_{k+1}$ denotes the numerical solution calculated by two steps of method (2) with the size $\tau_k/2$. Then the algorithm to control the local error of E-methods is presented as follows:

Algorithm 1: Local step size control

Step 1. Compute $\tilde{x}_{k+1}$, $\hat{x}_{k+1}$.
Step 2. Calculate $\|\Delta\tilde{x}_{k+1}\| = \|\hat{x}_{k+1} - \tilde{x}_{k+1}\|/(1 - 1/2^{2p+4})$.
Step 3. $\tilde{\tau}_k := \theta\left(\epsilon_l/\|\Delta\tilde{x}_{k+1}\|\right)^{1/(2p+5)} \tau_k$.
Step 4. If $\|\Delta\tilde{x}_{k+1}\| > \epsilon_l$ then $\tau_k := \tilde{\tau}_k$ and go to *Step 1*,
else $\tilde{x}_{k+1} := \tilde{x}_{k+1} + \Delta\tilde{x}_{k+1}$ and $\tau_{k+1} := \tilde{\tau}_k$.
Step 5. $t_k := t_k + \tau_k$, $k := k+1$ and go to *Step 1*.

Here, $\theta \in (0, 1)$ is a safety factor.

If we want to apply the local error control mechanism to iterative methods (5)–(7) we have to provide sufficient iterations at each grid point as the following estimates indicate:

$$N \ge \log_2\left(\frac{2p+7}{\xi+1}\right) \qquad \text{for the EN-method,} \tag{10}$$

$$N \ge \frac{2p+7}{\xi+1} - 1 \quad \text{for the EMN- and ESN-methods,} \tag{11}$$

$$N \ge 2p + 6 - \xi \qquad \text{for the ESI-method.} \tag{12}$$

Let us test our methods with the local step size control on problem (8). If we use the combined algorithms mentioned above to solve this problem we come to the data presented in Table 5. The number of iterations per grid point for each type of the iterative E-methods was calculated by formulas (10)–(12), respectively. These numbers are given in parentheses at the first column of Table 5.

Table 5. Global errors of variable step size iterative algorithms based on E-method (4) with Algorithm 1 to control step size and with extrapolation

Type of iteration	required accuracy $\epsilon_l = 10^{-03}$	$\epsilon_l = 10^{-04}$	$\epsilon_l = 10^{-05}$	$\epsilon_l = 10^{-06}$	$\epsilon_l = 10^{-07}$
SI(9)	$3.67 \cdot 10^{-03}$	$5.25 \cdot 10^{-04}$	$3.85 \cdot 10^{-06}$	$1.90 \cdot 10^{-06}$	$1.44 \cdot 10^{-07}$
N(3)	$1.46 \cdot 10^{-03}$	$1.94 \cdot 10^{-04}$	$2.53 \cdot 10^{-05}$	$9.16 \cdot 10^{-06}$	$3.73 \cdot 10^{-07}$
MN(5)	$1.51 \cdot 10^{-03}$	$1.74 \cdot 10^{-04}$	$2.74 \cdot 10^{-05}$	$2.08 \cdot 10^{-06}$	$3.00 \cdot 10^{-07}$
SN(5)	$2.92 \cdot 10^{-03}$	$6.67 \cdot 10^{-04}$	$2.36 \cdot 10^{-05}$	$2.55 \cdot 10^{-06}$	$1.34 \cdot 10^{-07}$

Table 6. Global errors of variable step size iterative algorithms based on E-method (4) with Algorithm 2 to control step size and with extrapolation ($\epsilon_l = \epsilon_g$)

Type of iteration	required accuracy $\epsilon_g = 10^{-03}$	$\epsilon_g = 10^{-04}$	$\epsilon_g = 10^{-05}$	$\epsilon_g = 10^{-06}$	$\epsilon_g = 10^{-07}$
SI(9)	$1.54 \cdot 10^{-04}$	$1.95 \cdot 10^{-05}$	$1.71 \cdot 10^{-06}$	$2.00 \cdot 10^{-07}$	$1.96 \cdot 10^{-08}$
N(3)	$2.89 \cdot 10^{-04}$	$3.60 \cdot 10^{-05}$	$3.15 \cdot 10^{-06}$	$4.44 \cdot 10^{-07}$	$4.27 \cdot 10^{-08}$
MN(5)	$2.16 \cdot 10^{-04}$	$2.85 \cdot 10^{-05}$	$3.28 \cdot 10^{-06}$	$4.79 \cdot 10^{-07}$	$4.02 \cdot 10^{-08}$
SN(5)	$1.63 \cdot 10^{-04}$	$3.38 \cdot 10^{-05}$	$2.80 \cdot 10^{-06}$	$5.04 \cdot 10^{-07}$	$4.09 \cdot 10^{-08}$

We see that the local step size selection is quite efficient for all the types of iterations, but, unfortunately, it does not allow to find automatically the numerical solution with the accuracy set by the user. To improve it, we have to involve a global error control in the step size selection procedure.

By this reason, we apply the local-global step size selection developed with the aim of controlling the global error of Runge-Kutta formulas in [9] to our E-methods with high derivatives. First, we suppose that the numerical solution $\tilde{x}_k$ has been computed at the point t_k with an accuracy of $O(\tau_k^{2p+5})$. Second, the local and global errors tolerances ϵ_l and ϵ_g have been given. Third, we assume that some step size τ_k has been fixed. Then the algorithm of the local-global step size selection for method (2) is presented as follows:

Algorithm 2: Local-global step size control

Step 1. By Algorithm1,find estimates $\Delta\tilde{x}_{k+1}$ $\Delta\tilde{x}_{k+2}$ of the local error of method (2) at the points t_{k+1}, t_{k+2} and step sizes $\tilde{\tau}_k$, $\tilde{\tau}_{k+1}$ for the tolerance ϵ_l (change the step size τ_k, if necessary).

Step 2. Determine the coefficient of the principal term of the local error $\hat{\psi}_{2p+5}(t_{k+1})$.

Step 3. Compute the coefficient of the principal term of the global error $\psi_{2p+4}(t_{k+1})$ by the formula

$$\psi_{2p+4}(t_{k+1}) := \Big(E_n - \tau_k \partial_x g(t_{k+1}, \tilde{x}_{k+1})\Big)^{-1} \Big(\tau_k \hat{\psi}_{2p+5}(t_{k+1})\Big). \qquad (14)$$

Step 4. Find an estimate $\Delta\psi_{2p+4}(t_{k+1})$ of the local error of method (14) by the Richardson extrapolation or by two one-step methods of different orders (see [9], for details).

Step 5. If $\|\Delta\psi_{2p+4}(t_{k+1})\| > \epsilon_g/100$ then calculate a newstep size τ_k as shown in Algorithm 1 and go to *Step 1*,

else $q := \left[\tau_k \left(\|\psi_{2p+4}(t_{k+1})\| / \epsilon_g \right)^{1/(2p+4)}\right] + 1.$

Step 6. If $q > 1$ then $\tau_k := \tau_k/q$ and go to *Step 1*,

else $\tilde{x}_{k+1} := \tilde{x}_{k+1} + \Delta\tilde{x}_{k+1}$, $\Delta\tilde{x}_{k+1} := \Delta\tilde{x}_{k+2}$ and $\tau_{k+2} := \tilde{\tau}_{k+1}$.

Step 7. $t_k := t_k + \tau_k$, $k := k + 1$ and go to *Step 1*.

Here, the square brackets mean an integer part of the number. A fuller description of Algorithm 2 for the iterative E-methods with high derivatives will appear in [10].

If we now test our iterative E-methods with Algorithm 2 to select a step size on problem (8) we come to Table 6. These data show the great advantage of the local-global step size control (Algorithm 2) over the local one (Algorithm 1), if we compare the global errors obtained in the last numerical experiment with Table 5 . Thus, E-methods presented in the paper together with the local-global step size control can be a good computational technique to solve many practical problems including stiff ones.

References

1. Aul'chenko, S. M., Latypov, A. F., Nikulichev, Yu. V.: A method for the numerical integration of systems of ordinary differential equations using Hermite interpolation polynomials. *(in Russian)* Zh. Vychisl. Mat. Mat. Fiz. **38** (1998) No. 10, 1665–1670; *translation in* Comput. Math. Math. Phys. **38** (1998) No. 10, 1595–1601
2. Bakhvalov, N.S.: Numerical methods. *(in Russian)* Nauka, Moscow, 1975
3. Berezin, I.S., Zhidkov, N.P.: Methods of computations. V. 1. *(in Russian)* Gos. izd-vo fiz.-mat. lit-ry, Moscow, 1962
4. Butcher, J.C.: Numerical methods for ordinary differential equations. John Wiley & Son, Chichester, 2003
5. Fehlberg, E.: New high-order Runge-Kutta formulas with step size control for systems of first and second order differential equations. ZAMM. **44** (1964) T17–T19
6. Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations I: Nonstiff problems. Springer-Verlag, Berlin, 1987
7. Hairer, E., Wanner, G.: Solving ordinary differential equations II: Stiff and differential-algebraic problems. Springer-Verlag, Berlin, 1996
8. Kastlunger, K.H., Wanner, G.: Runge-Kutta processes with multiple nodes. Computing. **9** (1972) 9–24
9. Kulikov, G.Yu.: A local-global version of a stepsize control for Runge-Kutta methods. Korean J. Comput. Appl. Math. **7** (2000) No. 2, 289–318
10. Kulikov, G.Yu., Merkulov, A.I.: On one-step collocation methods with high derivatives for solving ordinary differential equations. *(in Russian)* Zh. Vychisl. Mat. Mat. Fiz. (to appear); *translation in* Comput. Math. Math. Phys. (to appear)
11. Nørsett, S.P.: One-step methods of Hermite type for numerical integration of stiff systems. BIT. **14** (1974) 63–77
12. Ortega, J.M., Poole, W.G.: An introduction to numerical methods for differential equations. Pitman Publishing Inc., 1981

Local Sampling Problems

Shou-Yuan Yang[1] and Wei Lin[2*]

[1] Department of Mathematics, Sun Yat-Sen University,
Guangzhou 510275, PR China
yshouy@263.net
[2] Department of Mathematics, Sun Yat-Sen University,
Guangzhou 510275, PR China
stslw@zsu.edu.cn

Abstract. The main purpose of this paper is to investigate the local error for the sampling problem in diverse situations. We find that the local error is heavily depending on the asymptotic behavior of the sampling function. By virtue of evaluating the decay of the sampling function, we give a local error estimation for uniform and non-uniform sampling in multiresolution analysis (MRA) and in shift-invariant spaces.

1 Introduction and Preliminaries

In digital signal and image processing and digital communications, a continuous function (signal) is usually represented by its discrete samples. Then how to reconstruct a continuous function from its discrete samples is a fundamental problem. Many mathematicians and engineers contribute to this area, e. g., see[1],[3],[13]. Most of their work is mainly concerned with the whole domain cases, i.e., the sampling domain is the whole Euclidean space $\mathbb{R}^d$ and we want to recover the function on the whole space. But in many real applications, sampling may be limited to a bounded domain, and we need only to recover the function in a local area. So finding a suitable method to recover a function from its discrete samples locally is significant both in theory and application. The earliest results on local sampling may be found in [7,9]. In [2], N. Atreas et al. investigated the local error of the reconstruction formula in a multiresolution analysis (MRA). In [14], S. Y. Yang find a new method to evaluate the asymptotic rate of decay of the sampling function, and generalized their result to higher dimensional cases. In this paper, we shall investigate the local error for sampling problem in diverse situations, including sampling in shift-invariant spaces and in MRA, uniform sampling and non-uniform sampling.

1.1 Notations

Before proceeding with our discussion, we must introduce some notations first. $\mathbb{N}$, $\mathbb{Z}$,$\mathbb{R}$,$\mathbb{C}$ denote the set of natural numbers, integers, real numbers and complex numbers, respectively. $\mathbb{R}^d$ and Λ denote the d-dimensional Euclidean space

* Two authors both supported by the Natural Science Fund of China and the Natural Science Fund of Guangdong province.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 81–88, 2004.
© Springer-Verlag Berlin Heidelberg 2004

and unit lattice respectively. $\mathbb{T}^d = \mathbb{R}^d/\Lambda$ is the d-dimensional torus. $l^1(\Lambda)$ denotes the well-known Banach space of all absolutely summable sequences on Λ. The Fourier transform of a function $f \in L^1(\mathbb{R}^d)$ is defined by

$$\hat{f}(u) = \int_{\mathbb{R}^d} f(x)e^{-i2\pi\langle x,u\rangle}dx,$$

this transform maps s onto itself and extends to all tempered distributions s' by duality, here s denotes the class of all infinitely differentiable functions with rapid decay at infinity. We shwll also use the Wiener Amalgam space $W(L^p)$ defined by

$$W(L^p) = \left\{ f \in L^p(\mathbb{R}^d) : \|f\|_{W(L^p)} = \left(\sum_{k\in\Lambda} \sup_{x\in[0,1]^d} |f(x+k)|^p \right)^{1/p} < \infty \right\},$$

and we use $W_0(L^p)$ to denote its closed subspace of continuous functions. Endowed with the norm $\|\cdot\|_{W(L^p)}$, then $W(L^p)$ and $W_0(L^p)$ are Banach spaces; e.g., see [5,6]. $V^p(\varphi)(1 \le p \le \infty)$ is the shift-invariant space defined by

$$V^p(\varphi) = \left\{ \sum_{k\in\Lambda} c(k)\varphi(\cdot - k) : c \in l^p(\Lambda) \right\},$$

where φ is the so-called generator. For such a shift-invariant space to be well defined, we need the generator $\varphi \in W_0(L^1)$ and satisfies the following condition

$$0 < m \le \sum_{k\in\Lambda} |\hat{\varphi}(\xi + k)|^2 \le M < \infty$$

for some $m, M > 0$. Under these conditions, the shift-invariant space $V^p(\varphi)$ is a closed subspace of $L^p(\mathbb{R}^d)(1 \le p \le \infty)$. Moreover, we have

$$m_p\|c\|_{l^p(\Lambda)} \le \|\sum_{k\in\Lambda} c(k)\varphi(\cdot - k)\|_{L^p(\mathbb{R}^d)} \le M_p\|c\|_{l^p(\Lambda)}, \quad \forall\, c \in l^p(\Lambda) \tag{1}$$

for $1 \le p \le \infty$ and some positive constants m_p, M_p depending on p, e. g., see [1]. In this situation, the generator φ is said to be stable. By the way, we also need the the following notation

$$\hat{\varphi}* = \sum_{k\in\Lambda} \varphi(k)\exp(i2\pi\langle k, \cdot\rangle) \tag{2}$$

defined on $\mathbb{T}^d$.

Multiresolution analysis (MRA) is a fundamental concept in wavelet theory; e.g., see [4,11,12].

Definition 1. *A dyadic multiresolution analysis (MRA) of $L^2(\mathbb{R}^d)$ is a sequence $\{V_m : m \in \mathbb{Z}\}$ of closed subspaces of $L^2(\mathbb{R}^d)$ with the following properties:*

(i) $\forall\, m \in \mathbb{Z}, V_m \subseteq V_{m+1}$,
(ii) $\bigcap_m V_m = \{0\}$, $\overline{\bigcup_m V_m} = L^2(\mathbb{R}^d)$,
(iii) $f \in V_m \Leftrightarrow f(2^{-m}\cdot) \in V_0$,
(iv) $\exists\, \varphi \in V_0$, *a scaling function, such that* $\{\tau_n\varphi = \varphi(\cdot - n) : n \in \Lambda\}$ *is an orthonormal basis for* V_0.

Throughout this paper we shall use the standard notation $\varphi_{m,n} = 2^{md/2}\varphi(2^m \cdot - n)$ for $m \in \mathbb{Z}$, $n \in \Lambda$.

Definition 2. *A closed subspace* V *of* $L^2(\mathbb{R}^d)$ *is said to be a reproducing kernel Hilbert space (RKHS) with reproducing kernel* $K(x,y)$ *if and only if*

$$\forall\, f \in V, \quad x \in \mathbb{R}^d, \quad f(x) = \langle K(x,\cdot), f\rangle.$$

Let $X = \{x_j : j \in J\} \subseteq \mathbb{R}^d$. Then X is said to be separated if $\inf_{i\neq j} |x_i - x_j| > 0$; X is said to be γ_0-dense in $\mathbb{R}^d$ if $\mathbb{R}^d = \bigcup_{j\in J} B_\gamma(x_j)$ holds for all $\gamma > \gamma_0$, where $B_\gamma(x_j)$ denotes the open ball centered at x_j with diameter γ.

1.2 Reconstruction Formula for Sampling Problems

The basic idea of sampling is to recover a function $f(x)$ from its discrete samples $\{f(x_j) : j \in J\}$. Of course, only special kind of functions can be reconstructed in this way, moreover, the sampling set $x = \{x_j : j \in J\}$ must satisfy certain density condition. For the functions in a shift-invariant space $V^2(\varphi)$, we can recover them from their samples on the unit lattice Λ. the following proposition is well-known (e. g., see[13,14]).

Proposition 1. *Suppose the generator* $\varphi \in W_0(L^1)$ *is stable and* $\hat{\varphi}*$ *is non-vanishing on* $\mathbb{T}^d$, *where* $\hat{\varphi}*$ *is defined by (2) in subsection 1.1. Then*

a. $\{\varphi(\cdot - n) : n \in \Lambda\}$ *is a Riesz basis for* $V^2(\varphi)$;
b. $V^2(\varphi)$ *is an RKHS with the reproducing kernel* $K(x,y)$ *determined by*

$$K(x,y) = \sum_{k\in\Lambda} \overline{\varphi(x-k)}\tilde{\varphi}(y-k), \tag{3}$$

where $\tilde{\varphi}$ *is the dual of* φ *determined by*

$$\langle \varphi(\cdot - k), \tilde{\varphi}(\cdot - l)\rangle = \delta(k-l), \quad \forall\, k,l \in \Lambda; \tag{4}$$

c. There exist a unique function $s \in W_0(L^1)$ *such that*

$$f = \sum_{k\in\Lambda} f(k)s(\cdot - k), \quad \forall\, f \in V^2(\varphi), \tag{5}$$

s *is the so-called sampling function determined by*

$$\hat{s} = \hat{\varphi}/\hat{\varphi}*\,. \tag{6}$$

For MRA cases, we also have the following well-known proposition, e. g., see[2].

Proposition 2. *Let $\{V_m\}$ be an MRA of $L^2(\mathbb{R}^d)$ with scaling function $\varphi \in W_0(L^1)$. If φ is stable and $\hat{\varphi}*$ is non-vanishing on $\mathbb{T}^d$, then*

a. For each m, V_m is a RKHS, the corresponding reproducing kernel K_m is uniquely determined by

$$K_m(x,y) = \sum_{n\in\Lambda} \overline{\varphi_{m,n}(x)}\varphi_{m,n}(y); \tag{7}$$

b. There is a unique function $s \in V_0$ with the property that

$$f = \frac{1}{2^{dm/2}} \sum_{n\in\Lambda} f(\frac{n}{2^m}) s_{m,n}, \quad \forall\, f \in V_m \tag{8}$$

the convergence is in $L^2(\mathbb{R}^d)$ and is absolute, and therefore also uniform on $\mathbb{R}^d$. Where s is the so-called sampling function determined by (6) in Proposition 1.

For non-uniform sampling, we need the sampling set X satisfy certain density condition. The following proposition are borrowed from[1].

Proposition 3. *Suppose $\varphi \in W_0(L^1)$ is a stable generator, $X = \{x_j : j \in J\} \subseteq \mathbb{R}^d$ is a separated set, $K(x,y)$ is the reproducing kernel defined by (3) in Proposition 1. Then there exists some $\gamma_0 > 0$ such that if X is γ_0-dense in $\mathbb{R}^d$, then $\{K(x_j,\cdot) : j \in J\}$ is a frame for $V^2(\varphi)$. hence for any $f \in V^2(\varphi)$, we have the following reconstruction formula*

$$f(x) = \sum_{j\in J} \langle f, K(x_j,\cdot)\rangle \widetilde{K(x_j,\cdot)} = \sum_{j\in J} f(x_j)\widetilde{K(x_j,\cdot)}, \tag{9}$$

where $\{\widetilde{K(x_j,\cdot)} : j \in J\}$ is the dual frame of $\{K(x_j,\cdot) : j \in J\}$. In this situation, X is said to be a set of sampling

1.3 Local Error for Sampling Problem

Suppose we want to recover a function $f \in V^2(\varphi)$ in a bounded domain $\Omega \subseteq \mathbb{R}^d$. Then we must sample in a neighborhood of Ω. For simplicity, we suppose the sampling domain is $\Omega_\delta = \left\{x \in \mathbb{R}^d : dist(x,\Omega) < \delta\right\}$, where $dist(x,\Omega) = \inf_{y\in\Omega} |x-y|$ denotes the distance between x and Ω, and we shall use $X_{\Omega,\delta}$ to denote the set $X \bigcap \Omega_\delta$.

Definition 3. *Let X be a set of sampling (see Proposition 3). Then the local error of the reconstruction formula (9) is defined by*

$$E_{\Omega,\delta} f(x) = f(x) - \sum_{x_j \in X_{\Omega,\delta}} f(x_j)\widetilde{K(x_j,x)}, \quad x \in \Omega. \tag{10}$$

Since the reconstruction formulas (5) and (8) are just special cases of the reconstruction formula (9), we can define the local error of these reconstruction formulas similarly, i.e.

$$E_{\Omega,\delta}f(x) = f(x) - \sum_{n\in\Lambda\bigcap\Omega_\delta} f(n)s(x-n), \quad x\in\Omega \tag{11}$$

for (5) and

$$E^m_{\Omega,\delta}f(x) = f(x) - \sum_{n\in\Lambda:2^{-m}n\in\Omega_\delta} f\left(\frac{n}{2^m}\right)s(2^m x-n), \quad x\in\Omega \tag{12}$$

for (8).

In general, the localization of these reconstruction formulas can be measured by the asymptotic rate of decay of the local error $E_{\Omega,\delta}$ (or $E^m_{\Omega,\delta}$) as δ increases. If the local error decays very fast, then the reconstruction formula is well localized, and we can recover a function from its samples locally. But as we shall see, the asymptotic rate of decay of the local error is heavily depending on the asymptotic behavior of the sampling function $s(x)$ (in uniform case) or the dual frame $\{\widetilde{K(x_j,\cdot)} : j\in J\}$ (in non-uniform case), so we must evaluate the asymptotic rate of decay of the aforementioned sampling function and dual frame first. Fortunately, we have already many results on this topic at hand.

Here is a brief outline of this paper. In section 2, we evaluate the local error for uniform sampling problems, Theorem 1, Theorem 2, Theorem 3 and Theorem 4 are the main results in this section; In section 3, we evaluate the local error for the non-uniform sampling problem, Theorem 5 is the main result in this section.

2 Local Error Estimation for Uniform Sampling

This section mainly deals with the uniform case. We shall give several estimates for the local error in different situations. Through out this section, we always assume the generator (or scaling function in MRA case) φ satisfies one of the following conditions

(C1) $\varphi(x) = O((1+|x|)^{-\beta})$ as $|x|\longrightarrow\infty$ for some $\beta > \frac{d}{2}$;

(C2) $\varphi(x) = O(\exp(-\alpha\cdot|x|))$ as $|x|\longrightarrow\infty$ for some $\alpha > 0$.

If the generator (or scaling function in MRA case) φ satisfies condition (C1), then S. Y. Yang has already give the following result (see[14])

Theorem 1. *Assume the generator $\varphi\in W_0(L^1)$ is stable and satisfies condition (C1) for some $\beta > \frac{d}{2}$. Then the local error $E_{\Omega,\delta}f(x)$ defined by (11) in subsection 1.3 satisfies*

$$\sup_{x\in\Omega}|E_{\Omega,\delta}f(x)| = O(\delta^{d/2-\beta})\cdot\|f\|_{L^2(\mathbb{R}^d)} \quad \text{as} \quad \delta\longrightarrow\infty, \quad \forall\, f\in V^2(\varphi). \tag{13}$$

Theorem 2. *Assume the scaling function $\varphi \in W_0(L^1)$ is stable and satisfies condition (C2) for some $\beta > \frac{d}{2}$. Then the local error $E^m_{\Omega,\delta} f(x)$ defined by (12) in subsection 1.3 satisfies*

$$\sup_{x\in\Omega} |E^m_{\Omega,\delta} f(x)| = O(\delta^{d/2-\beta}) \cdot \|f\|_{L^2(\mathbb{R}^d)} \quad \text{as} \quad \delta \longrightarrow \infty, \quad \forall f \in V_m(\varphi). \tag{14}$$

Now we assume the generator (or scaling function in MRA case) is continuous and satisfies condition (C2), then φ automatically belongs to $W_0(L^1)$. We first give the following lemma, which will also be used in the next section, its proof is direct, one can see[15], Lemma 3.1.

Lemma 1. *Let $I_1(\delta,\alpha) = \int_{|y|\geq\delta} |y|^{-d-\alpha} dy$, $I_2(\delta,\alpha) = \int_{|y|\geq\delta} \exp(-\alpha \cdot |y|) dy$, then we have*

$$I_1(\delta,\alpha) = 2\pi^{d/2}\Gamma\left(\frac{d}{2}\right)^{-1} \cdot \frac{1}{\alpha}\delta^{-\alpha} \tag{15}$$

and

$$I_2(\delta,\alpha) = 2\pi^{d/2}\Gamma\left(\frac{d}{2}\right)^{-1} \cdot e^{-\alpha\delta} \cdot \left(\sum_{k=0}^{d-1} \frac{(d-1)!}{(d-1-k)!} \cdot \alpha^{-k-1} \cdot \delta^{d-1-k}\right). \tag{16}$$

To evaluate the asymptotic rate of decay of the sampling function $s(x)$, we need the following lemma, its proof is found in[10].

Lemma 2. *Suppose $\varphi \in W_0(L^1)$ satisfies condition (C2) for some $\alpha > 0$.*
a. If φ is the generator of a shift-invariant space $V^2(\varphi)$, then the sampling function $s(x)$ determined by (6) in Proposition 1 satisfies

$$s(x) = O(\exp(-\alpha'|x|)) \quad \text{as} \quad |x| \longrightarrow \infty \tag{17}$$

for some $\alpha' \in (0,\alpha)$;
b. If φ is the scaling function of an MRA, then the sampling function $s(x)$ determined by (6) also satisfies (17) for some $\alpha' \in (0,\alpha)$.

Now we can evaluate the decay of the local error. The following theorem can be proved in the same way as Theorem 3.1 in[14], so we omit its proof.

Theorem 3. *Assume the generator φ is continuous, stable and satisfies condition (C2) for some $\alpha > 0$. Then for any $f \in V^2(\varphi)$, the local error $E_{\Omega,\delta} f(x)$ defined by (11) in subsection 1.3 satisfies*

$$\sup_{x\in\Omega} |E_{\Omega,\delta} f(x)| = O(\delta^{(d-1)/2} \cdot \exp(-\alpha'\delta)) \cdot \|f\|_{L^2(\mathbb{R}^d)} \quad \text{as} \quad \delta \longrightarrow \infty \tag{18}$$

for some $\alpha' \in (0,\alpha)$.

For the MRA case, similar result also holds. The following theorem can be proved in the same way as theorem 3, so we also omit its proof.

Theorem 4. *Assume the scaling function φ is continuous, stable and satisfies condition (C2) for some $\alpha > 0$. Then for any $f \in V^2(\varphi)$, the local error $E^m_{\Omega,\delta} f(x)$ defined by (12) in subsection 1.3 satisfies*

$$\sup_{x\in\Omega} |E^m_{\Omega,\delta} f(x)| = O(\delta^{(d-1)/2} \cdot \exp(-\alpha'\delta)) \cdot \|f\|_{L^2(\mathbb{R}^d)} \quad \text{as} \quad \delta \longrightarrow \infty \tag{19}$$

for some $\alpha' \in (0, \alpha)$.

3 Local Error Estimation for Non-uniform Sampling

This section mainly deals with the non-uniform case. Through out this section, we assume the generator φ either satisfy condition (C2) defined in section 2 or satisfy the following condition

(C3) $\varphi(x) = O((1 + |x|)^{-\beta})$ as $|x| \longrightarrow \infty$ for some $\beta > d$.

Hence if moreover φ is continuous, then in both cases we can conclude that $\varphi \in W_0(L^1)$. By Proposition 3, if X is a set of sampling, then the reconstruction formula (9) holds. It is easy to observe that the localization of this reconstruction formula is determined by the off-center decay of the dual frame $\{\widetilde{K(x_j, \cdot)} : j \in J\}$, because if each $\widetilde{K(x_j, \cdot)}$ is concentrated in a neighborhood of x_j, then $f(x)$ is mainly determined by few samples which are near the point x. So evaluating the off-center decay of every $\widetilde{K(x_j, \cdot)}$ is important. Fortunately, S. Jaffard investigated the off-diagonal decay of the infinite matrices, got the conclusion that the inverse matrix has almost the same off-diagonal decay as the original matrix (see [8]). Recently, K. Gröchenig using Jaffard's result to evaluate the off-center decay of the dual frame $\{\widetilde{K(x_j, \cdot)} : j \in J\}$, e. g. see[10]. The following lemma is one of his main results.

Lemma 3. *Assume the generator φ is continuous and stable, $K(x, y)$ is the corresponding reproducing kernel defined by (3) in Proposition 1, $X = \{x_j : j \in J\}$ is a set of sampling, $\{\widetilde{K(x_j, \cdot)} : j \in J\}$ is the dual frame of $\{K(x_j, \cdot) : j \in J\}$.*
a. If moreover φ satisfies condition (C3), then we have

$$|K(x_j, y)| \le C(X, \varphi)(1 + |y - x_j|)^{-\beta}, \quad \forall\, j \in J\,, \tag{20}$$

where $C(X, \varphi)$ is a constant depending on X and φ, independent of j.
b. If moreover φ satisfies condition (C2) for some $\alpha > 0$, then there exists some $\alpha' \in (0, \alpha)$ such that

$$|K(x_j, y)| \le C(X, \varphi) \exp(-\alpha'|y - x_j|), \quad \forall\, j \in J\,, \tag{21}$$

where $C(X, \varphi)$ has the same property as in a.

By virtue of the above result, we now can give an evaluation for the local error $E_{\Omega,\delta} f(x)$ defined by (10) in subsection 1.3.

Theorem 5. *Suppose the generator φ is continuous and stable, $X = \{x_j : j \in J\}$ is a set of sampling satisfying $\inf_{i \neq j} |x_i - x_j| = \mu > 0$.*
a. If moreover φ satisfies condition (C3), then the local error $E_{\Omega,\delta} f(x)$ defined by (10) in subsection 1.3 satisfies

$$\sup_{x \in \Omega} |E_{\Omega,\delta} f(x)| = O(\delta^{d/2-\beta}) \cdot \|f\|_{L^2(\mathbb{R}^d)} \quad \text{as} \quad \delta \longrightarrow \infty, \quad \forall f \in V^2(\varphi). \tag{22}$$

b. If moreover φ satisfies condition (C2) for some $\alpha > 0$, then there exists some $\alpha' \in (0, \alpha)$ such that

$$\sup_{x \in \Omega} |E_{\Omega,\delta} f(x)| = O(\delta^{(d-1)/2} \cdot \exp(-\alpha' \delta)) \cdot \|f\|_{L^2(\mathbb{R}^d)} \quad \text{as} \quad \delta \longrightarrow \infty, \quad \forall f \in V^2(\varphi). \tag{23}$$

Because of page limit, we omit the proof of this theorem, but we will present it in another paper.

References

1. A. Aldroubi, and K. Gröchenig, Nonuniform sampling and reconstruction in shift-invariant spaces. SIAM Rev. **43**(4)(2001) 585-620
2. N. Atreas, J. J. Benedetto, and C. Karanikas, Local sampling for regular wavelet and Gabor expansions(to appear)
3. John J. Benedetto, Irregular sampling and frames, in: C. K. Chui (Ed.), Wavelets: A Tutorial in Theory and Applications. (1992) 445-507
4. I. Daubechies, Ten Lectures on Wavelets. CBMS-NSP Series in Applied Math. SIAM Philadelphia. 1992.
5. H. G. Feichtinger, Generalized amalgams, with application to Fourier transform. Cand. J. Math. **42**(1990) 395-409
6. H. G. Feichtinger, Wiener amalgams over Euclidean spaces and some of their applications. In Proc. Conf. Function Spaces, K. Jarosz, ed., Lecture Notes in Math. 136, Springer-Verlag, New York, 1991, pp.123-137
7. H. D. Helms and J. B. Thomas, Truncation error of sampling theorem expansions. Proc. IRE **50**(1962) 179-184
8. S. Jaffard, Propriétés des matrices "bien localisées" prés de leur diagonale et quelques applications. Ann. Inst. H. Poincaré Anal. Non Linéaire **7**(5) (1990) 461–476
9. D. Jagerman, Bounds for truncation error of the sampling expansions. SIAM J. Appl. Math. **14**(1966) 714-723
10. K. Gröchenig, Localization of frames, Banach frames, and the invertibility of the frame operator. J. Fourier Anal. Appl.(to appear)
11. S. Mallat, A Wavelet Tour of Signal Processing. Academic Press, Boston, 1998
12. Y. Meyer, Ondelettes Et Opérateurs. Hermann, Paris, 1990
13. Michael Unser, Sampling–50 years after Shannon. Procdings of the IEE. **88** (4) (2000) 569-587
14. S. Y. Yang, Local error estimation for samling problems, Appl. Math. Comp.(to appear)
15. S. Y. Yang, The local property of several operators on sampling, Applicable Anal. (to appear)
16. K. Yao, Application of reproducing kernel Hilbert spaces–bandlimited signal models. Inform. and control. **11** (1967) 429-444

Recent Advances in Semi-Lagrangian Modelling of Flow through the Strait of Gibraltar

Mohammed Seaïd[1], Mofdi El-Amrani[2], and Ahmed Machmoum[3]

[1] Fachbereich Mathematik AG8, TU Darmstadt, 64289 Darmstadt, Germany
seaid@mathematik.tu-darmstadt.de
[2] Dpto. Matemáticas, Univ. Rey Juan Carlos, 28933 Mostoles-Madrid, Spain
mofdi@escet.urjc.es
[3] Faculté des Sciences, LIMI, Université Ibn Zohr, B.P. 28/S Agadir, Morocco

Abstract. Two aspects of work are addressed in this paper. The first is concerned with a mathematical model for mean flow and hydraulics in the strait of Gibraltar. The model is based on the two-dimensional shallow water equations. The second aspect of work is devoted to developing a robust numerical method for solving such equations. We introduce a fully implicit semi-Lagrangian method which maintains stability even if large time steps are used in computations and reduces artificial numerical dispersion. Preliminary results obtained for a dam-break problem show that our method is able to provide stable and accurate solutions.

1 Introduction

During the last decade, enormous amount of work on mathematical modelling has been performed. The advent of more capable computation machines has proved the way to the use of mathematical models in all aspects of engineering, including ocean circulation, hydraulics and also floods. In this work, we try to build a numerical method to approximate solutions for the partial differential equations used to model the mean flow of water in the strait of Gibraltar and, consequently, determining their dynamics.

The strait of Gibraltar connects the Atlantic ocean with the Mediterranean sea. The differences on density, salinity and temperature of the two water bodies lead to a flow exchange through the strait. This flow exchange consists of two counter-flowing layers: (i) an upper layer of Atlantic water flowing into the Mediterranean sea and (ii) a lower layer of Mediterranean water flowing into the Atlantic ocean. For comprehensive contributions on oceanography of the strait of Gibraltar we refer the reader to the proceeding book for a seminar on studies conducted in the strait of Gibraltar [2]. In this paper, we are interested on the mean flow exchange in the water free surface to which the shallow water equations can be a suitable model. Particularly, we are concerned with numerical study of inflow contributed by the Atlantic ocean into the Mediterranean sea which takes place on the free water surface.

The main difficulties in solving numerically the shallow water equations used to model mean flow in the strait of Gibraltar are the nonlinearity structure of

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 89–96, 2004.
© Springer-Verlag Berlin Heidelberg 2004

the equations and the coupling through a source function. Furthermore, it is the presence of the convective term in the shallow water equations that makes finding solutions to these equations difficult, this convective term is a source of serious computational problems and oscillations.

Eulerian methods use fixed grids and incorporate some upstream weighting in their formulations to stabilize the schemes. Among the class of Eulerian methods are the Petrov-Galerkin methods, the streamline diffusion methods and also include many other methods such as the high resolution methods from computational fluid dynamics, in particular, the Godunov methods and the essentially non-oscillatory methods. All of these Eulerian methods are easy to formulate and to implement. However, time truncation errors dominate their solutions that introduce numerical diffusion and are subject to the CFL stability conditions that put a restriction on the size of the time steps taken in numerical simulations.

Semi-Lagrangian methods, on the other hand, make use of the transport nature of the governing equations. They combine the fixed Eulerian grids with a particle tracking along the characteristic curves of the governing equations. The Lagrangian treatment in these methods greatly reduces the time truncation errors in the Eulerian methods. In addition, these methods alleviate the restrictions on the Courant number, thus allowing for large time steps in the simulations and reduce artificial numerical dispersion. Obviously, any explicit term in semi-Lagrangian formulation is computed at the characteristic feet by interpolation that might introduce extra dissipation. In order to eliminate this drawback, we propose a fully implicit semi-Lagrangian method. The discretized system can be formulated as a compact fixed point problem in flow alone. As a nonlinear solver we apply a Newton-Gmres algorithm.

The purpose of this work is to develop a robust semi-Lagrangian method to approximate solutions to the equations governing mean flow in the strait of Gibraltar. In the present paper, we discuss only the formulation of a fully implicit semi-Lagrangian method for mean flow model and results are given for a dam-break problem. As part of an ongoing project, this method will be implemented for the full model and obtained results will be compared to measurements done in the strait of Gibraltar.

2 Mathematical Model

In this section, we briefly set the governing equations for mean flow in the strait of Gibraltar. For more details on the modelling and related physical studies we refer to [2] and further references are cited therein. The derivation of the equations is based on Navier-Stokes problem with the assumption that the dynamics of mean flow on continental shelves is such that the vertical advection of momentum can be assumed to be small and the pressure to be approximately hydrostatic. If these hypotheses are applied to the Navier-Stokes equations, we obtain the well-known shallow water equations in two space dimensions given by the continuity and momentum equations:

$$\partial_t \eta + \partial_x\big((\eta + h)U\big) + \partial_y\big((\eta + h)V\big) = 0, \tag{1}$$

$$\partial_t U + U\partial_x U + V\partial_y U - fV = -g\partial_x \eta - \frac{r}{\eta + h} U\sqrt{U^2 + V^2} + K_H \nabla^2 U, \qquad (2)$$

$$\partial_t V + U\partial_x V + V\partial_y V + fU = -g\partial_y \eta - \frac{r}{\eta + h} V\sqrt{U^2 + V^2} + K_H \nabla^2 U, \qquad (3)$$

where η is the free surface height, $\mathbf{U} = (U, V)^T$ is the vertically integrated velocity, h is the water depth measured from the mean sea level, g is the gravity acceleration, K_H is the horizontal eddy viscosity, r denotes the drag coefficient on the bottom, f is the Coriolis parameter defined by $f = 2\omega \sin\phi$, with ω is the angular velocity of the earth and ϕ is the geographic latitude, and ∇^2 denotes the two-dimensional Laplace operator.

Equations (1)-(3) are defined in a spatial domain bounded by Tangier-Barbate line at the west and Sebta-Gibraltar at the east as shown in Fig. 1. This domain contains the Camarinal Sill (interface that separates the Mediterranean sea and the Atlantic ocean) where exchange of the water body takes place.

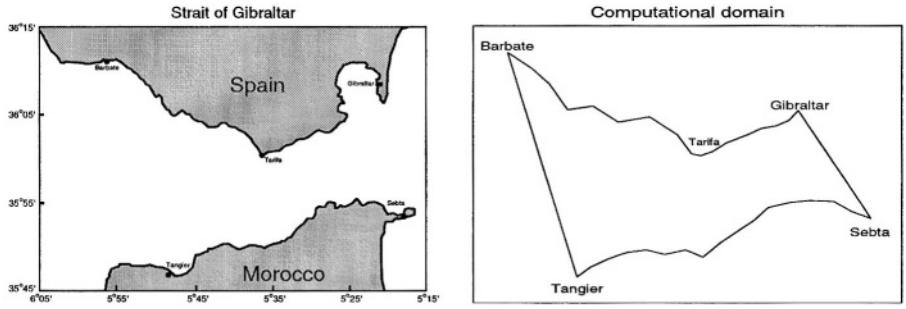


Fig. 1. Schematic diagrams for the strait of Gibraltar and the computational domain

In order to obtain a well-posed problem, initial data has to be provided and boundary conditions have to be imposed. A detailed mathematical study on these issues can be found in [2]. The overall problem is complicated one and it includes certain physical processes which are not yet fully understood. Also, the numerical methods for simulating solutions of the problem which are of key importance have not yet been well established. Theoretical considerations can provide some ideas, concerning stability, convergence rates, restriction on time stepsizes, or qualitative behavior of the solution, but a complete quantitative analysis is not possible today. Therefore, the only way to make a judgment is to perform numerical tests, at least for some problems which seem to be representative. However, looking into the literature, it seems that there have not been many studies of this type which can give satisfactory answers.

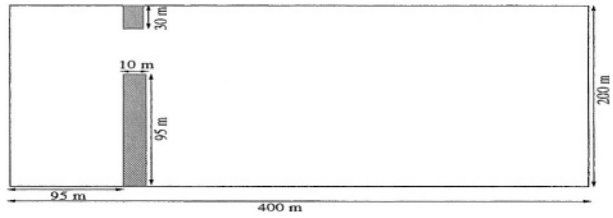


Fig. 2. Geometry of the rectangular strait in the dam-break problem

3 Numerical Method

The semi-Lagrangian method we consider in this paper consists on two fractional steps. The first step is the Lagrangian interpretation of the advective part in equations (1)-(3) by the modified method of characteristics, while the second step uses the Eulerian coordinates for discretization of the remaining part. In vector-valued form, the equations (1)-(3) can be rewritten as

$$\begin{aligned} &D_t\eta + \mathbf{U}\cdot\nabla h + \gamma\nabla\cdot\mathbf{U} = 0,\\ &D_t\mathbf{U} + g\nabla\cdot\eta - K_H\nabla^2\mathbf{U} = \mathbf{S}(\gamma,\mathbf{U}), \end{aligned} \tag{4}$$

where the total depth $\gamma = \eta + h$, the source term is defined by

$$\mathbf{S}(\gamma,\mathbf{U}) = \left(fV - \frac{r}{\gamma}U\sqrt{U^2+V^2}, -fU - \frac{r}{\gamma}V\sqrt{U^2+V^2}\right)^T, \tag{5}$$

the gradient vector $\nabla = (\partial_x, \partial_y)^T$, and $D_t w = \partial_t w + \mathbf{U}\cdot\nabla w$ is the material derivative of the function w in direction of the flow $\mathbf{U}$. The fundamental idea of semi-Lagrangian methods is to impose a regular grid at the new time level, and to backtrack the flow trajectories to the previous time level. At the old time level, the quantities that are needed are evaluated by interpolation from their known values on a regular grid.

Let the time interval be divided into subintervals $[t_n, t_{n+1}]$ of length Δt such that $t_n = n\Delta t$. Following [3], the characteristics curves of the equations (4) are the solution of initial value problem

$$\begin{aligned} &\frac{d\mathbf{X}(\tau;t_{n+1},\mathbf{x})}{d\tau} = \mathbf{U}\left(\tau,\mathbf{X}(\tau;t_{n+1},\mathbf{x})\right), \quad \tau\in[t_n,t_{n+1}],\\ &\mathbf{X}(t_{n+1};t_{n+1},\mathbf{x}) = \mathbf{x}. \end{aligned} \tag{6}$$

Note that $\mathbf{X}(\tau;t_{n+1},\mathbf{x}) = (X(\tau;t_{n+1},\mathbf{x}), Y(\tau;t_{n+1},\mathbf{x}))^T$ is the departure point at time τ of a fluid particle that will arrive at $\mathbf{x} = (x,y)^T$ at time $\tau = t_{n+1}$. The semi-Lagrangian methods do not follow the flow particles forward in time, as the Lagrangian schemes do, instead their trace backwards the position at time t_n of particles that will reach the points of a fixed mesh at time t_{n+1}. By so

doing, the semi-Lagrangian methods avoid the grid distortion difficulties that the conventional Lagrangian schemes have. In this paper, to solve the ordinary differential equations (6), we used a method first proposed in the context of semi-Lagrangian schemes to integrate the weather prediction equations [5].

Once the characteristics feet $\mathbf{X}(t_n; t_{n+1}, \mathbf{x})$ are known, the material derivatives in (4) are approximated by

$$D_t\mathbf{U} = \frac{\mathbf{U}^{n+1} - \hat{\mathbf{U}}^n}{\Delta t}, \qquad D_t\eta = \frac{\eta^{n+1} - \hat{\eta}^n}{\Delta t}, \tag{7}$$

where $\mathbf{U}^{n+1} = \mathbf{U}(t_{n+1}, \mathbf{x})$, $\eta^{n+1} = \eta(t_{n+1}, \mathbf{x})$; $\hat{\mathbf{U}}^n = \mathbf{U}\left(t_n, \mathbf{X}(t_n; t_{n+1}, \mathbf{x})\right)$ and $\hat{\eta}^n = \eta\left(t_n, \mathbf{X}(t_n; t_{n+1}, \mathbf{x})\right)$. In general, the departure points $\mathbf{X}(t_n; t_{n+1}, \mathbf{x})$ do not coincide with the spatial position of a grid point. A requirement is then that the scheme to compute $\mathbf{X}(t_n; t_{n+1}, \mathbf{x})$ be provided with a search-locate algorithm to find the host element where such point is located. A general, efficient and easy to implement scheme to perform this step in arbitrary grids is presented in [1].

Assuming that a suitable approximation is made for $\mathbf{X}(t_n; t_{n+1}, \mathbf{x})$, then the functions $\hat{\mathbf{U}}^n$ and $\hat{\eta}^n$ in (7) must be obtained by interpolation from known values at the grid points. The interpolation procedure we used in this paper is the bicubic spline most commonly used in practice.

Using the assumption that the total water depth γ is approximately constant within the time interval $[t_n, t_{n+1}]$, and an implicit time integration method, the semi-discretization of (4) is

$$\frac{\eta^{n+1} - \hat{\eta}^n}{\Delta t} + \mathbf{U}^{n+1} \cdot \nabla h + \gamma^n \nabla \cdot \mathbf{U}^{n+1} = 0, \tag{8}$$

$$\frac{\mathbf{U}^{n+1} - \hat{\mathbf{U}}^n}{\Delta t} + g\nabla \cdot \eta^{n+1} - K_H \nabla^2 \mathbf{U}^{n+1} = \mathbf{S}(\gamma^n, \mathbf{U}^{n+1}). \tag{9}$$

Note that the bottom, h, is assumed to be time-independent (so there is no time-level superscript associated with it). A simple way to solve the equations (8)-(9), is to use the first equation (8) to eliminate the divergence term $\nabla \cdot \eta^{n+1}$ from the second equation (9). These procedure yields to a fixed point problem

$$\mathbf{U} = \mathcal{H}(\mathbf{U}), \tag{10}$$

in $\mathbf{U}$ only. In (10), we have dropped the $n+1$ superscript for ease of notation. Newton's method applied to (10) results in the following iteration

$$\mathbf{U}^{(k+1)} = \mathbf{U}^{(k)} - \mathcal{R}'\left(\mathbf{U}^{(k)}\right)^{-1} \mathcal{R}\left(\mathbf{U}^{(k)}\right), \tag{11}$$

where $\mathcal{R}(\mathbf{U}) = \mathbf{U} - \mathcal{H}(\mathbf{U})$ is the nonlinear residual and $\mathcal{R}'$ is the system Jacobian approximated by a forward difference quotient of the form

$$\mathcal{R}'\left(\mathbf{U}^{(k)}\right)\mathbf{w} \approx \frac{\mathcal{R}\left(\mathbf{U}^{(k)} + \delta\mathbf{w}\right) - \mathcal{R}\left(\mathbf{U}^{(k)}\right)}{\delta}. \tag{12}$$

If Gmres method is used to compute the Newton direction then, at each time step, the following algorithm (Newton-Gmres) has to be called in the time loop:

Algorithm 1: Ngmres($\mathcal{H}, \mathbf{U}^{(0)}, \varepsilon$)
Given $\mathcal{H}$, tolerance ε and initial guess $\mathbf{U}^{(0)}$, the Newton-Gmres algorithm for solving (10) uses the following steps: (we denote by Gmres($\mathbf{A}$,$\mathbf{b}$,$\mathbf{z}^{(0)}$,ε) the result of Gmres algorithm applied to linear system $\mathbf{Az = b}$ with initial guess $\mathbf{z}^{(0)}$ and tolerance ε)
do $k = 0, 1, \ldots$
Compute the residual $\mathcal{R}^{(k)} = \mathbf{U}^{(k)} - \mathcal{H}(\mathbf{U}^{(k)})$
Solve using Gmres, $\mathbf{d}^{(k)} = \text{Gmres}\left(\mathcal{R}'(\mathbf{U}^{(k)}), -\mathcal{R}(\mathbf{U}^{(k)}), \mathbf{d}^{(0)}, \varepsilon^{(k)}\right)$
Update the solution $\mathbf{U}^{(k+1)} = \mathbf{U}^{(k)} + \xi \mathbf{d}^{(k)}$
If $\left(\|\mathbf{U}^{(k+1)}\|_{L^2} \leq \varepsilon\right)$ stop
end do

The free surface height η^{n+1} can be updated by back-substituting $\mathbf{U}^{n+1}$ in the first equation (8) as

$$\eta^{n+1} = \hat{\eta}^n - \Delta t \mathbf{U}^{n+1} \cdot \nabla h - \gamma^n \Delta t \nabla \cdot \mathbf{U}^{n+1}. \tag{13}$$

To discretize the equations (4) in space, we use the finite difference discretization. Thus, the spatial domain is divided into cells with nonuniform sizes and centred at grid points $(x_i = i(\Delta x)_i, y_j = j(\Delta y)_j)$. We use the notation $w^n_{i,j} = w(t_n, x_i, y_j)$ and we define the following centred difference operators

$$\mathcal{D}_x w_{i,j} = \frac{w_{i+1,j} - w_{i-1,j}}{2(\Delta x)_i}, \qquad \mathcal{D}^2_x w_{i,j} = \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)_i{}^2},$$

similarly for $\mathcal{D}_y w$ and $\mathcal{D}^2_y w$. Using the semi-discrete formulation (8)-(9), the fully-discrete approximation for the equations (4) can be directly written as

$$\begin{aligned}
&\frac{\eta^{n+1}_{i,j} - \hat{\eta}^n_{i,j}}{\Delta t} + U^{n+1}_{i,j}\mathcal{D}_x h_{i,j} + V^{n+1}_{i,j}\mathcal{D}_y h_{i,j} + \gamma_{i,j}\left(\mathcal{D}_x U^{n+1}_{i,j} + \mathcal{D}_y V^{n+1}_{i,j}\right) = 0,\\
&\frac{U^{n+1}_{i,j} - \hat{U}^n_{i,j}}{\Delta t} + g\mathcal{D}_x \eta^{n+1}_{i,j} - K_H \tilde{\mathcal{D}}^2_{\mathbf{x}} U^{n+1}_{i,j} - S_1(\gamma^n_{i,j}, \mathbf{U}^{n+1}_{i,j}) = 0, \qquad (14)\\
&\frac{V^{n+1}_{i,j} - \hat{V}^n_{i,j}}{\Delta t} + g\mathcal{D}_y \eta^{n+1}_{i,j} - K_H \tilde{\mathcal{D}}^2_{\mathbf{x}} V^{n+1}_{i,j} - S_2(\gamma^n_{i,j}, \mathbf{U}^{n+1}_{i,j}) = 0,
\end{aligned}$$

where S_1, S_2 are the entries of the source term $\mathbf{S}$ in (5) and $\tilde{\mathcal{D}}^2_{\mathbf{x}} = \mathcal{D}^2_x + \mathcal{D}^2_y$. The implementation of the semi-Lagrangian algorithm to solve (4) is carried out in the following steps. Given $\{\eta^n, \mathbf{U}^n\}$, we compute $\{\eta^{n+1}, \mathbf{U}^{n+1}\}$ via:

Algorithm 2: Semi-Lagrangian algorithm
Step 1. *Compute the departure point* $\mathbf{X}(t_{n+1}; t_n, \mathbf{x})$ *in (6) and identify the element of the grid where such a point is located*
Step 2. *Evaluate the approximations* $\hat{\eta}^n$ *and* $\hat{\mathbf{U}}^n$ *employing the bicubic spline interpolation from the values of* η^n *and* $\mathbf{U}^n$*, respectively*
Step 3. *Formulate the fixed point problem (10) for the flow* $\mathbf{U}^{n+1}$ *only*
Step 4. *Call the Newton-Gmres algorithm Ngmres*$(\mathcal{H}, \mathbf{U}^n, \varepsilon)$ *to obtain* $\mathbf{U}^{n+1}$
Step 5. *Update the free surface height* η^{n+1} *using (13)*

It is worth saying that, the time integration method used in (8)-(9) is first order accurate. A second order time discretization can be formulated in similar way by considering the Crank-Nicolson method.

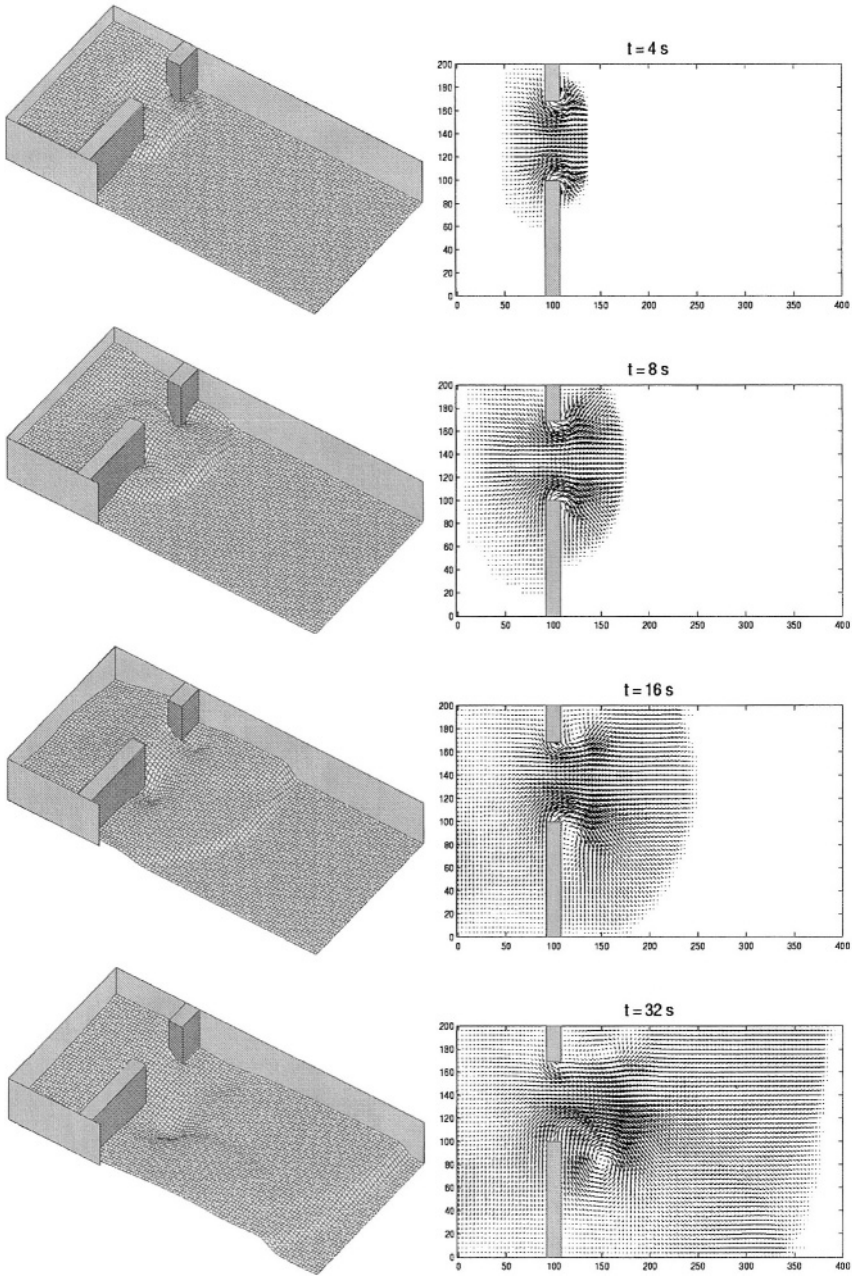


Fig. 3. Water surface elevation (left column) and velocity field (right column)

4 Test Problem

To validate our algorithm we consider a prototype test of dam-break problem formulated by the equations (4). We consider a 200 m long and 400 m wide rectangular strait with two different constant levels of water separated by a dam. At $t = 0$ s part of the dam breaks instantaneously. The dam is 10 m thick and the breach is assumed to be 75 m wide. Fig. 2 shows a sketch of the strait and dimensions of the dam. The initial water level is 10 m at upstream and 5 m at downstream and water is initially at rest. For the other parameters in equations (4) we use similar values as those provided for mean flow in the strait of Gibraltar [2]. Thus, the gravity acceleration $g = 9.81\ m/s^2$, the friction of the bottom $r = 3 \times 10^{-3}$, the Coriolis parameter $f = 8.55 \times 10^{-5}\ s^{-1}$, and the viscosity coefficient $K_H = 4 \times 10^{-4}\ kg/ms$.

In the left column of Fig. 3 we plot the water surface elevation, while the right column contains the corresponding velocity vectors. All computations are made on an uniform Cartesian grid with $\Delta x = \Delta y = 40\ m$ and fixed time step $\Delta t = 0.5\ s$. We used a tolerance $\varepsilon = 10^{-5}$ to stop iterations in Newton algorithm (`Algorithm 1`), and we have observed that the number of iterations required to reach this tolerance was between 8 and 13 during the time integration process. The duration of simulation is 32 s.

The semi-Lagrangian scheme captured correctly the discontinuity and the shock without need for very fine mesh. The monotonicity of the scheme is preserved and no nonphysical oscillations or extra numerical diffusion have been detected during the computations. Note that the performance of our semi-Lagrangian scheme is very attractive since the computed solution remains, stable, monotone and highly accurate even on coarse grids without requiring special front tracking techniques. We can see that the small complex structures of the water flow being captured by our semi-Lagrangian scheme.

References

1. Allievi, A., Bermejo, R.: A Generalized Particle Search-Locate Algorithm for Arbitrary Grids. J. Comp. Physics, **132** (1997) 157–166
2. Almazán, J.I., Bryden, H., Kinder, T., Parrilla, G. (eds.): Seminario Sobre la Oceanografía Física del Estrecho de Gibraltar. SECEG, Madrid (1988)
3. Robert, A.: A stable Numerical Integration Scheme for the Primitive Meteorological Equations. Atmos. Ocean **19** (1981) 35–46
4. Seaïd, M.: Semi-Lagrangian Integration Schemes for Viscous Incompressible Flows. J. Comp. Methods in App. Math. **4** (2002) 392–409
5. Temperton, C., Staniforth, A.: An Efficient Two-Time-Level Semi-Lagrangian Semi-Implicit Integration Scheme. Quart. J. Roy. Meteor. Soc. **113** (1987) 1025–1039

Efficiency Study of the "Black-Box" Component Decomposition Preconditioning for Discrete Stress Analysis Problems

M.D. Mihajlović [1]* and S. Mijalković [2]

[1] Department of Computer Science, The University of Manchester, Manchester, UK
[2] Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands

Abstract. Efficiency of the preconditioning methodology for an iterative solution of discrete stress analysis problems is studied in this article. The preconditioning strategy is based on space decomposition and subspace correction framework. The principle idea is to decompose a global discrete system into a sequence of scalar subproblems, corresponding to the different Cartesian coordinates of the displacement vector. The scalar subproblems can be treated by a host of direct and iterative techniques, however we restrict ourselves to a "black-box" application of the direct sparse solvers and the scalar algebraic multigrid (AMG) method. The subspace correction is performed in either block diagonal or block lower triangular fashion. The efficiency and potential limitations of the proposed preconditioning methodology are studied on stress analysis for 2D and 3D model problems from microfabrication technology.

1 Introduction

Stress analysis is an essential part of modelling and design of complex physical and mechanical systems that consist of a variety of structural elements with different mechanical properties and intrinsic built-in stress distributions. The underlying multi-layer material geometries (with potentially very complex shapes) are usually partitioned by general unstructured grids. The governing stress equations are discretised by the finite element method. This procedure leads to the solution of large sparse linear algebraic systems with a non-regular sparsity pattern and poor equations scaling.

The problem of designing efficient solution methods for stress analysis has been studied by many authors. Some of this work covers specific topics, such as methods for thin domains in 3D [12], mixed and penalty methods for incompressible problems [8], and the AMLI methods for compressible problems [1]. The optimal and nearly optimal preconditioners for Krylov subspace methods can often be designed within a general framework of space decomposition and

* This article was produced during the first author's visit to Delft University of Technology. The financial support from the Leonardo da Vinci Staff Exchange Programme is greatly acknowledged.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 97–104, 2004.
© Springer-Verlag Berlin Heidelberg 2004

subspace correction (SSC) methods [19]. The idea behind the SSC strategy is to decompose, in a stable manner, the global finite dimensional space into a set of local subspaces. The global solution is obtained by the suitable combination of the local subspace problem solutions. The basis for the space decomposition for stress analysis is formed from the Cartesian components of the displacement vector. In this way, the local subspace problems represent material displacement along a single Cartesian coordinate, provided that displacements in other directions are fixed [1].

The local subspace problems have the properties of the scalar PDEs. Thus, there exists a variety of efficient methods for solving and preconditioning of these discrete linear systems. In [2] approximate factorisation is used to solve the subproblems. In [13] the local subproblems are approximated by the additive multilevel method (AMLI). In our study we adopt two different strategies for the solution of the local subproblems: direct sparse factorisation and the approximation by the scalar algebraic multigrid (AMG) solver. The main idea behind this choice is to implement the whole solver in a "black-box" fashion by using publicly available codes for both the Krylov solver and the local subproblems' solvers/preconditioners. Direct sparse solvers are not (asymptotically) the optimal choice in this context, both in terms of the execution time and storage requirements. On the other hand, AMG method offer the prospect of optimal scaling with problem size [16], [17]. It is developed essentially for stiffness matrices corresponding to scalar elliptic PDEs (M-matrices). Thus, the scalar AMG solver is a suitable choice for preconditioning of the local subproblems.

This article is organised as follows. In Section 2 we introduce the physical problem and give details of the finite element discretisation. Section 3 describes our preconditioning strategy. In Section 4 we present numerical examples from microfabrication technology modelling. The aim is to demonstrate efficiency of the proposed methodology, as well as to outline some of its limitations and potential problems.

2 Problem Formulation

Deformation of a continuous material body occupying a bounded domain $\Omega \subset \Re^d$ $(d = 2, 3)$ is described by the following boundary value problem

$$-\nabla \cdot \sigma(\bar{u}) = \bar{f} \quad \text{in } \Omega; \qquad \bar{u} = \bar{u}_D \quad \text{on } \partial\Omega_D; \qquad \sigma(\bar{u})\hat{n} = \bar{u}_N \quad \text{on } \partial\Omega_N. \tag{1}$$

In (1) $\sigma(\bar{u})$ denotes the Cauchy stress tensor, $\bar{u}$ is the displacement vector, and $\bar{f}$ = is the body force vector. The vector $\bar{u}_D$ represents the prescribed displacement of the boundary segment $\partial\Omega_D$, while $\bar{u}_N$ is the surface traction of the boundary segment $\partial\Omega_N$ with outward unit normal vector $\hat{n}$ $(\partial\Omega_D \cap \partial\Omega_N = \emptyset$, $\partial\Omega_D \cup \partial\Omega_N = \partial\Omega)$. For linear elasticity, the stress tensor $\sigma(\bar{u})$ is related to the strain tensor $\varepsilon(\bar{u})$ by Hooke's law

$$\sigma(\bar{u}) = 2\mu\varepsilon(\bar{u}) + \lambda(\nabla \cdot \bar{u})I$$

where I is the identity tensor, and $\mu > 0$ and $\lambda > 0$ are Lamé's coefficients, which, in turn, depend on Young's modulus and the Poisson ratio [8]. Note that

in this study we consider only the cases of compressible linear elasticity, when ν is bounded away from 0.5.

The variational stress analysis problem reads as follows: Find $\bar{u} \in [V(\Omega)]^d$ such that

$$E(\bar{u}, \bar{v}) = F(\bar{v}) \qquad \forall \bar{v} \in [V_0(\Omega)]^d, \tag{2}$$

where $V_0(\Omega) = H_0^1(\Omega)$ is the standard Sobolev space of functions defined on Ω with the homogeneous Dirichlet boundary conditions on $\partial\Omega_D$. It is possible to prove that problem (2) has a unique solution (see [5]). In (2) $E(\bar{u}, \bar{v})$ represents the bilinear energy functional, and $F(\bar{v})$ is the load vector functional (see [10]).

The discrete stress analysis problem can be obtained from (2) by applying the Galerkin projection of $V(\Omega)$ onto $V_h(\Omega) \subset V(\Omega)$. In this way we obtain the problem: find $\bar{u}_h \in [V_h(\Omega)]^d$ such that

$$E(\bar{u}_h, \bar{v}_h) = F(\bar{v}_h) \qquad \forall \bar{v}_h \in [V_{h0}(\Omega)]^d.$$

In our case, $V_h(\Omega)$ is the space of piecewise linear polynomials that correspond to a given partition of the domain Ω into disjoint triangles/tetrahedra (2D/3D). If we introduce the nodal basis set $V_h(\Omega) = \text{span}\{\phi_r\}_{r=1}^N$, the discrete displacement vector component u_i can be written as $u_i = \sum_{r=1}^{N} u_i^r \phi_r$, $i = 1, \ldots, d$, with u_i^r being the unknown displacement nodal coefficients along the ith Cartesian coordinate. By adopting for the discrete test space $V_{h0}(\Omega)$ the same basis set as for $V_h(\Omega)$, we obtain a system of $d \cdot N$ linear algebraic equations

$$\sum_{j=1}^{d} \sum_{s=1}^{N} A_{ij}^{rs} u_j^s = f_i^r, \qquad i = 1, \ldots, d; \quad r = 1, \ldots, N, \tag{3}$$

with

$$A_{ij}^{rs} = \mu \sum_{k=1}^{d} (\partial_k \phi_r, \partial_k \phi_s)\delta_{ij} + \mu(\partial_j \phi_r, \partial_i \phi_s) + \lambda(\partial_j \phi_s, \partial_i \phi_r)$$
$$f_r^i = (f_i, \phi_r) + \text{b.t.},$$

where $(\cdot, \cdot)$ denotes the usual scalar product in $L_2(\Omega)$.

3 Preconditioning Methodology

The large, sparse linear system (3) can be solved by a host of iterative and direct techniques [3]. In this article we study the efficiency of the preconditioned Krylov subspace methods. In preconditioning, the aim is to design a matrix M (called the preconditioner) which is spectrally close to the coefficient matrix A, but simple to assemble and compute the action of its inverse. The aim is that the preconditioned matrix $M^{-1}A$ has a small number of distinct eigenvalues with large multiplicity, or that the spectrum of $M^{-1}A$ is tightly clustered and

bounded by the quantities that are (ideally) independent of the discretisation parameter h and other problem parameters.

In order to solve the system (3) by the SSC method, it is necessary to decompose the discrete finite element space $V_h(\Omega)$ into a sum of subspaces $V_h^{(i)}(\Omega)$. Furthermore, the robust and effective implementation of SSC methods requires an efficient solution of the subspace problems. In this article we study two techniques: direct sparse solution of the subproblems, and the approximate solution of the subproblems by the scalar AMG. In our methodology we adopt a component-wise space decomposition, where for $\bar{u} \in V_h^{(i)}(\Omega)$ we have $u_j = 0$ if $i \neq j$ [1]. In the case of block diagonal subspace correction the preconditioner is defined as $M_D = D$, while in the case of block lower triangular subspace correction we have $M_L = D + L$ (L is the block lower triangular part of A).

We comment briefly on the suitability of the AMG as an approximate solver for the local subproblems. The local subspace operators

$$\mathcal{D}(u, v) = \mu \sum_i (\nabla u_i, \nabla v_i) + (\lambda + \mu) \sum_i (\partial_i u_i, \partial_i v_i) \tag{4}$$

are anisotropic as the second sum contains div-type operators. In our approach we apply scalar AMG as a "black-box" solver to the entire discrete local subproblems (4). Because of the coarsening strategy adopted in AMG (coarsening is performed in the directions in which the error varies smoothly [17]), we expect efficient performance of the AMG with respect to anisotropies and discontinuities along the different material interfaces. The possible scenarios where the AMG strategy may encounter difficulties are the cases of nearly incompressible materials ($\lambda \to \infty$), very thin domain structures (with large ratios between the principal dimensions), and highly non-convex domains (see Example 2 in Section 4).

4 Numerical Examples

A component based preconditioning strategy is practically tested on stress analysis problems in 2D and 3D that arise in microfabrication technology. However, the suggested methodology is not restricted in any way to this particular application. Microfabrication technology is concerned with the design and production of microelectronic and micromechanical components [9].

Having established superiority of the component based preconditioning over the standard ILU(0) preconditioning [10], we now examine the efficiency (in terms of wall clock time and memory requirements) of our methodology when various strategies are employed for the solution of local subproblems. For completeness we compare our results with a general direct sparse solver from the HSL library [6]. First we give some implementation details. We adopt BiCGSTAB(2) as a representative Krylov subspace method (see [15]). In our implementation, in line with the strategy of developing a fully "black-box" solver, we take a BiCGSTAB(ℓ) implementation from the NAG library [11]. The exact version of

the SSC preconditioner (with sparse direct factorisation of the blocks) is implemented using the code HSL_MA42 from the HSL library [6]. This code implements the frontal Gaussian elimination and is suitable for non-symmetric linear systems [4]. We also use the MSRO strategy for reordering the equations within each diagonal block [14], as implemented in the routine MC62 from the HSL library [6]. An inexact version of the SSC preconditioner (with diagonal blocks preconditioned by a small fixed number of AMG cycles) is implemented using the well-established and publicly available code AMG1R5 [16]. The approximate AMG solving procedure consists of one V(1,1) cycle. Two model problems are studied (one in 2D and one in 3D). The discrete linear systems (3) are generated by the Taurus simulator [18]. The right hand side vector is chosen randomly (although the elements of $\mathbf{f}$ are scaled appropriately in order to simulate the realistic situation of a randomly distributed load).

Next we describe our experiment environment. All tests are performed on an Origin 3400 architecture with 16 R12000 processors (although no parallelism is considered). A single R12000 processor has a clock speed of 400MHz, corresponding to a theoretical peak performance of 800MFlops. The system has 4GB of RAM. All our codes are written in Fortran 90 and optimised using the high compiler optimisation.

Example 1. We consider the model problem in 3D. The domain represents one quarter of the intermediate structure (the domain is restricted by using suitable symmetries) during fabrication of the Fin Field Effect Transistor (FinFet) (see [18]). The FinFet technology offers the possibility for further miniaturisation of the transistor structure to nanoscale dimensions (< 50nm). The present structure consists of four different material domains, which have different coefficients of thermal expansion. During production the whole structure is subject to high temperature variations. This process causes the build-up of stress in the structure (see [7]). The domain is discretised by unstructured grids of various sizes. The idea of studying this domain geometry is to test the robustness of the proposed iterative technique in the case of relatively "thin" domain geometries in 3D. The standard discretisation and solution techniques may encounter difficulties in this context [12]. Table 1 gives the number of iterations (N_{it}) and the total execution time (T_t) required by the preconditioned BiCGSTAB(2) algorithm to reduce the initial residual below 10^{-6} for both the exact (E) and inexact (I) version of the SSC preconditioner. In Table 1, n denotes the total dimension of the problem and M_D and M_L stand for the block diagonal and the block triangular versions of the preconditioner.

The asterix in the last column of Table 1, part (E) indicates that the matrix size is beyond the memory capacity of a particular architecture (we did not use direct access files for storing partially factorised matrices, see [6]). It can be concluded that this implementation gives a spectrally independent preconditioner, however, neither the execution time nor the storage requirements are optimal.

Next we report the results obtained for the inexact version of the SSC preconditioner. Part (I) of Table 1 presents the iteration counts and the execution time of the BiCGSTAB(2) algorithm with the inexact SSC preconditioner. In this case we observe a very moderate growth in the iteration counts. However, AMG

is an optimal solver both in terms of execution time and memory requirements. The total storage for the largest case from Table 1, part (I) is approximately 900MB. The optimality in terms of execution time is reflected in the rows T_t of Table 1, part (I). These times scale near linearly with the problem size and are much smaller than in the exact case.

Finally, for completeness, we contrast the timing results from parts (E) and (I) in Table 1 with the execution times required by a general sparse direct solver to complete this task (part (D) of Table 1). In this context we employ HSL_MA42 code with the MSRO reordering of the equations (MC62 code) [6].

Table 1. Iteration counts and the execution time (in seconds) of the BiCGSTAB(2) algorithm for the case of the exact (E) SSC preconditioner (direct factorisation of the discrete local subproblems by HSL_MA42) and the inexact SSC preconditioner (I) (preconditioning of the local subproblems by one V(1,1) AMG cyle). Execution time (in seconds) of the direct (D) sparse solver HSL_MA42.

n			21987	38550	51120	113085	224292
E	M_D	N_{it}	6	6	6	6	*
		T_t	17.16	32.12	65.90	698.31	*
	M_L	N_{it}	4	4	4	4	*
		T_t	16.51	30.00	62.32	675.98	*
I	M_D	N_{it}	8	10	10	12	14
		T_t	4.54	9.57	12.57	37.37	82.04
	M_L	N_{it}	6	8	6	8	12
		T_t	4.38	9.42	10.58	30.73	87.75
D		T_t	47.76	86.77	223.48	*	*

Example 2. We examine the efficiency of our methodology when applied to a model problem in 2D. The structure represents a trench obtained by etching the silicon substrate and depositing the nitride over it (see [18]). In microfabrication technology trenches are frequently used for the electromagnetic isolation of integrated active and passive components [20]. The main feature of this problem, as opposed to Example 1 and the example from [10] is that the domain Ω is not convex. This can potentially cause some difficulties to the iterative solver using the AMG preconditioner. In this example, stress in the structure is caused by the deposition of a material (nitride in this case) with intrinsic built-in stress. In the following experiments we are following exactly the same procedure as in Example 1. Table 2 presents the iteration counts and the execution time of the preconditioned BiCGSTAB(2) algorithm with the exact (E) and inexact (I) version of the SSC preconditioner needed to reduce the infinite norm of the residual below 10^{-6}.

Table 2. Iteration counts and the execution time of the BiCGSTAB(2) algorithm for the case of the exact (E) version SSC preconditioner (direct factorisation of the discrete local subproblems by HSL_MA42) and the inexact SSC preconditioner (I) (preconditioning of the local subproblems by one V(1,1) AMG cyle). Execution time (in seconds) of the direct (D) sparse solver HSL_MA42.

n			4000	9860	17464	34988	69316	136770
E	M_D	N_{it}	20	24	20	20	24	22
		T_t	1.08	4.07	8.42	24.37	67.05	212.58
	M_L	N_it	14	16	16	18	20	18
		T_t	0.89	3.13	7.55	23.50	61.52	200.14
I	M_D	N_{it}	50	58	72	84	92	98
		T_t	1.15	3.66	9.37	25.09	59.80	135.95
	M_L	N_{it}	38	46	58	66	72	76
		T_t	0.91	3.38	8.19	21.40	51.86	115.20
D		T_t	0.56	2.44	9.98	27.52	118.92	*

From Table 2, part (E) we can conclude that the exact SSC preconditioner exhibits nearly optimal convergence characteristics in this context (although a higher number of iterations is required than in Example 1 to achieve the prescribed tolerance). From Table 2, part (I) it can be concluded that the inexact version of the SSC preconditioner encounters some problems with the convergence in this context. Moreover, the convergence characteristics exhibit an irregular pattern. Although the inexact version of the SSC preconditioner is not vastly superior in terms of the execution time as in Example 1 (compared to the exact version), it still has the advantage of substantially lower memory requirements (for the case $n = 136770$ the total memory requirement is 260MB, compared to 3.7GB for the exact case, if the direct access files are not used). Finally, we contrast the timing results of the iterative method with the time required by the sparse direct solver HSL_MA42 (Table 2, part (D)) to complete the same task. Again, direct access files are not used in this case.

From Table 2 it can be concluded that in the case of the 2D problem with non-convex domain geometry the iterative method is still faster than the direct solver, but the superiority is not as comprehensive as in the 3D case. Of course, the storage requirements argument is still on the side of the iterative solver. The potential reason for slow convergence of the inexact solver is that the coarse level approximations of the local subproblems do not reflect the domain geometry with sufficient accuracy. This problem can be partially alleviated by restricting the number of coarse levels in the automatic coarsening process [16]. However, this can cause relatively large problem sizes at the coarsest level (with fairly dense coefficient matrices). This can potentially reduce the number of iterations, but will lead to increase of the execution times.

References

1. Axelsson, O.: On iterative solvers in structural mechanics; separate displacement orderings and mixed variable methods, Math. Comp. Simulat., **50**(1999), 11–30.
2. O. Axelsson, O., Gustafsson, I.: Iterative methods for the solution of the Navier equations of elasticity, Comp. Meth. Appl. Mech. Engng, **15**(1978), 241-258.
3. Dongarra, J.J., Duff, I.S., Sorensen, D.C., van der Vorst, H.A.: Numerical Linear Algebra for High-Performance Computers, SIAM, Philadelphia, PA, 1998.
4. Duff, I.S., Scott, J.A.: The design of a new frontal code for solving sparse unsymmetric systems, ACM Trans. Math. Soft., **22**(1)(1996), 30–45.
5. Fichera, G.: Existence theorems in elasticity, In: Handbuch der Physik, Vol. 6a/2 (S. Flüge, C. Truesdell eds.), Springer, Berlin, 1972, 347–389.
6. HSL: A collection of Fortran codes for large-scale scientific computation, 2002, (see `http://hsl.rl.ac.uk/`) .
7. Hu, S.M.: Stress related problems in silicon technology, Int. J. Appl. Phys., **15**(1991), R53–R80.
8. Hughes, T.J.R.: The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Prentice–Hall, Englewood–Cliffs, NJ, 1987.
9. Joppich, W., Mijalković, S.: Semiconductor Process Modelling, in: Wiley Encyclopedia of Electrical and Eletronics Engineering (J.G. Webster ed.), Wiley, New York, 1999, 127–139.
10. Mihajlović, M.D., Mijalković, S.: A component decomposition preconditioning for 3D stress analysis problems, Numer. Linear Algebra Appl., **9**(2002), 567–583.
11. Numerical Algorithms Group: NAG Manual, Fortran Library, Mark 20, Oxford, 2002.
12. Ovtchinikov, E.E., Xanthis, L.S.: Iterative subspace correction methods for thin elastic structures and Korn's type inequality in subspaces, Proc. Roy. Soc. (London), Ser. A **454**(1998), 2023–2039.
13. Padiy, A.: On a robust multilevel method applied for solving large-scale linear elasticity problems, Commun. Numer. Meth. Engng, **15**(1999), 153–165.
14. Scott, J.A.: A new row ordering strategy for frontal solvers, RAL Technical Report, RAL-TR-1998-056, 1998.
15. Sleipen, G.L.G., Fokkema, D.R.: BiCGSTAB(ℓ) for linear equations involving unsymmetric matrices with complex spectrum, ETNA **1**(1993), 11–32.
16. Stüben, K.: Algebraic multigrid (AMG): experiences and comparisons, Appl. Maths and Comput., **13**(1983), 419–452.
17. Stüben, K.: A review of algebraic multigrid, J. Comput. Appl. Maths, **128**(2001), 281–309.
18. Taurus Process and Device: Multi-Dimensional Process and Device Modelling Program, Version 2002.4, User's Manual, Synopsis, 2002.
19. Xu, J.: Iterative methods by space decomposition and subspace correction, SIAM Review **34**(1992), 581–613.
20. van Zant, P.: Microchip Fabrication: A Practical Guide to Semiconductor Processing, McGraw Hill, 2000.

Direct Solver Based on FFT and SEL for Diffraction Problems with Distribution

Hideyuki Koshigoe

Department of Urban Environment System, Chiba University,
Inage, Chiba, 263-8522, Japan
koshigoe@faculty.chiba-u.jp

Abstract. A direct solver for diffraction problems is presented in this paper. The solver is based on the fast Fourier transform (FFT) and the successive elimination of lines which we call SEL. In the previous paper, we showed the numerical algorithm by use of SEL and proved that the limit function of approximate solutions satisfied the diffraction problem in the sense of distribution. In this paper, the above numerical algorithm is improved with FFT and we show that the calculation speed is faster than the previous one.

1 Introduction

This paper is devoted to study the construction of finite difference solutions based on the direct method which we call SEL and establish the numerical algorithm by use of FFT.

Let Ω be a rectangular domain in R^2, Ω_1 be an open subset of Ω and $\Omega_2 = \Omega \backslash \overline{\Omega_1}$, the interface of them be denoted by $\Gamma (= \overline{\Omega_1} \cap \overline{\Omega_2})$. The diffraction problem considered here is the followings.

Problem I. For $f \in L^2(\Omega)$, $\sigma \in L^2(\Gamma)$ and $g \in H^{1/2}(\partial\Omega)$, find $\{u_1, u_2\} \in H^1(\Omega_1) \times H^1(\Omega_2)$ such that

$$-\epsilon_1 \triangle u_1 = f \quad \text{in} \quad \Omega_1 , \tag{1}$$

$$-\epsilon_2 \triangle u_2 = f \quad \text{in} \quad \Omega_2 , \tag{2}$$

$$u_1 = u_2 \quad \text{on} \quad \Gamma , \tag{3}$$

$$\epsilon_1 \frac{\partial u_1}{\partial \nu} - \epsilon_2 \frac{\partial u_2}{\partial \nu} = \sigma \quad \text{on} \quad \Gamma . \tag{4}$$

$$u_2 = g \quad \text{on} \quad \partial\Omega , \tag{5}$$

Here ν is the unit normal vector on Γ directed from Ω_1 to Ω_2 and ϵ_i is a positive parameter $(i = 1, 2)$.

The sysytem consisting of equations (1)-(5) is called the diffraction problem ([4]) and the equation (3)-(4) imposed on the surface Γ is also called the transmission conditions ([5]). The diffraction problems are arisen in various sciences. One of such examples can be found in the context of electricity and $\{\epsilon_1, \epsilon_2\}$ is corresponding to the dielectric constant of the material $\{\Omega_1, \Omega_2\}$.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 105–112, 2004.
© Springer-Verlag Berlin Heidelberg 2004

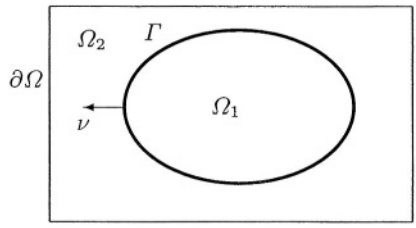


Fig. 1. Interface Γ and unit normal ν

From the view point of numerical analysis, one gives $w \in H^{1/2}(\Gamma)$, solves the Dirichlet problems in each domain Ω_i $(i = 1, 2)$:

$$\begin{cases} -\epsilon_1 \triangle u_1(w) = f & \text{in} \quad \Omega_1 , \\ u_1(w) = w & \text{on} \quad \Gamma , \end{cases} \tag{6}$$

$$\begin{cases} -\epsilon_2 \triangle u_2(w) = f & \text{in} \quad \Omega_2 , \\ u_2(w) = w & \text{on} \quad \Gamma, \\ u_2(w) = g & \text{on} \quad \partial\Omega , \end{cases} \tag{7}$$

and finds $w_0 \in H^{1/2}(\Gamma)$ satisfying

$$\epsilon_1 \frac{\partial u_1(w_0)}{\partial \nu} - \epsilon_2 \frac{\partial u_2(w_0)}{\partial \nu} = \sigma \quad \text{on} \quad \Gamma . \tag{8}$$

Hence introducing the Dirichlet-Neumann map T defined by

$$T : H^{1/2}(\Gamma) \ni w \to \epsilon_1 \frac{\partial u_1(w)}{\partial \nu} - \epsilon_2 \frac{\partial u_2(w)}{\partial \nu} \in H^{-1/2}(\Gamma),$$

Problem I is reduced to find w_0 satisfying

$$Tw_0 = \sigma \tag{9}$$

(see also [6]). Therefore one of purposes of this paper is to present a direct method for solving (9).

This paper is organized as follows. Section 2 describes the finite difference scheme with distribution. Section 3 is devoted to study the construction of approximate solutions from the viewpoint of the successive elimination of lines. Finally we shall present a numerical algorithm by use of FFT and SEL in Section 4.

2 Finite Difference Scheme with Distribution

2.1 Distribution Formulation

Before proceeding to the finite difference scheme, we reform Problem I as follows:
Problem II. For $f \in L^2(\Omega)$, $\sigma \in L^2(\Gamma)$ and $g \in H^{1/2}(\partial\Omega)$, find $u \in H^1(\Omega)$ such that

$$-\ \mathrm{div}\,(a(x,y)\,\nabla u) \ = \ f + \sigma\ \delta_\Gamma \quad \text{in} \quad D'(\Omega)\ , \tag{10}$$

$$u \ = \ g \quad \text{on} \quad \partial\Omega\ . \tag{11}$$

Here $a(x,y) = \epsilon_1\,\chi_{\Omega_1}(x,y) + \epsilon_2\,\chi_{\Omega_2}(x,y)$ where χ_{Ω_i} $(i = 1,2)$ isdefined by

$$\chi_{\Omega_i}(x,y) = \begin{cases} 1 & if\ (x,y) \in \Omega_i \\ 0 & if\ (x,y) \notin \Omega_i \end{cases}$$

and δ_Γ is the distribution with the support on Γ.
In fact, since the equation

$$\mathrm{div}\,(a(x,y)\,\nabla u) = a(x)\ \Delta u - (\epsilon_1\ \frac{\partial u_1}{\partial \nu} - \ \epsilon_2\ \frac{\partial u_2}{\partial \nu}) \quad \text{in} \quad D'(\Omega)$$

holds for any $u \in H^1(\Omega)$, it follows that Problem I is equivalent to Problem II ([1], [7]).
Hereafter the discretization of Problem II will be used in stead of Problem I .

2.2 Finite Difference Scheme with Distribution

Without loss of generality we assume that $g = 0$ and that Ω is the unit square in R^2, i.e., $\Omega = \{(x,y) |\ 0 < x, y < 1\ \}$. Let $h \in R$ be a mesh size such that $h = 1/n$ for an integer n and set $\Delta x = \Delta y = h$. We associate with it the set of the grid points:

$$\overline{\Omega}_h = \{m_{i,j} \in R^2 \mid m_{i,j} = (i\ h,\ j\ h),\ 0 \le i, j \le n\},$$
$$\Omega_h = \{m_{i,j} \in R^2 \mid m_{i,j} = (i\ h,\ j\ h),\ 1 \le i, j \le n-1\}.$$

With each grid point $m_{i,j}$ of $\overline{\Omega}_h$, we associate the panel $\omega^0_{i,\,j}$ with center $m_{i,j}$:

$$\omega^0_{i,j} \equiv \ \Big(\ (i-1/2)h,\ (i+1/2)h\ \Big] \times \Big(\ (j-1/2)h,\ (j+1/2)h\ \Big], \tag{12}$$

and the cross $\omega^1_{i,j}$ with center $m_{i,j}$:

$$\omega^1_{i,j} = \ \omega^0_{i+1/2,j} \cup\ \omega^0_{i-1/2,j} \cup\ \omega^0_{i,j+1/2} \cup\ \omega^0_{i,j-1/2} \tag{13}$$

where e_i denotes the i th unit vector in R^2 and we set

$$\omega^0_{i\pm 1/2,j} = \ \omega^0_{i,j} \pm \frac{h}{2}\ e_1, \quad \omega^0_{i,j\pm 1/2} = \ \omega^0_{i,j} \pm \frac{h}{2}\ e_2. \tag{14}$$

Moreover using the datum in Problem I, we define

$$\begin{cases} a^E_{i,j} = \frac{1}{\Delta x \Delta y} \int_{\omega^0_{i+1/2,j}} a(x,y)\, dxdy, & a^W_{i,j} = \frac{1}{\Delta x \Delta y} \int_{\omega^0_{i-1/2,j}} a(x,y)\, dxdy, \\ a^N_{i,j} = \frac{1}{\Delta x \Delta y} \int_{\omega^0_{i,j+1/2}} a(x,y)\, dxdy, & a^S_{i,j} = \frac{1}{\Delta x \Delta y} \int_{\omega^0_{i,j-1/2}} a(x,y)\, dxdy, \\ f_{i,j} = \frac{1}{\Delta x \Delta y} \int_{\omega^0_{i,j}} f(x,y)\, dxdy, & \sigma_{i,j} = \frac{1}{\Delta\, l_{i,j}} \int_{\Gamma \cap \omega^0_{ij}} \sigma(s)\, ds, \\ \Delta\, l_{i,j} = \int_{\Gamma \cap \omega^0_{i,j}} ds. \end{cases} \tag{15}$$

We then define the discrete equation of Problem II as follows.

Problem F. Find $\{u_{i,j}\}$ $(1 \le i,\ j \le n-1)$ such that

$$\begin{aligned} &-\frac{1}{\Delta x}\left(a^E_{i,j} \frac{u_{i+1,j} - u_{ij}}{\Delta x} - a^W_{i,j} \frac{u_{ij} - u_{i-1,j}}{\Delta x} \right) \\ &-\frac{1}{\Delta y}\left(a^N_{i,j} \frac{u_{i,j+1} - u_{i,j}}{\Delta y} - a^S_{i,j} \frac{u_{i,j} - u_{i,j-1}}{\Delta y} \right) \\ &\quad = f_{i,j} + \frac{\Delta l_{i,j}}{\Delta x \Delta y} \sigma_{i,j}, \quad 1 \le i,j \le n-1. \end{aligned} \tag{16}$$

Now introducing the step function $\theta_{i,j}$:

$$\theta_{i,j}(x,y) = \begin{cases} 1, & (x,y) \in \omega^0_{i,j} \\ 0, & (x,y) \notin \omega^0_{i,j} \end{cases},$$

and let us define the piecewise functions σ_h and u_h by

$$\begin{aligned} \sigma_h &= \textstyle\sum_{i,j=1}^{n-1} \frac{\Delta l_{i,j}}{\Delta x \Delta y}\, \sigma_{i,j}\, \theta_{i,j}(x,y), \\ u_h &= \textstyle\sum_{i,j=1}^{n-1} u_{i,j}\, \theta_{i,j}(x,y) \end{aligned} \tag{17}$$

respectively. We then have ([3])

Theorem 1. *(i)* $\sigma_h \to \sigma \cdot \delta_\Gamma$ *in* $D'(\Omega)$,
(ii) $u_h \to u$ *weakly in* $L^2(\Omega)$, $u \in H^1(\Omega)$, *and*
(iii) u *is the solution of Problem II in the sense of distribution.*

3 Construction of Approximate Solutions (SEL)

3.1 Vector Valued Equations

In this section we state the direct method which we call the successive elimination of lines. Instead of the $(n-1)^2$ unknowns $u_{i,j}$ in the discrete equation (16), we introduce the line vectors such that

$$U_i = {}^t[u_{i,1},\ u_{i,2}, \cdots,\ u_{i,n-1}\ ] \quad (1 \le i \le n-1).$$

Then Problem F *w.r.t* $\{u_{i,j}\}$ is reduced to Problem M *w.r.t* $\{U_i\}$ from the equations (16),

Problem M. Find U_i $(1 \le i \le n-1)$ satisfying

$$A^e_i U_i = A^W_i U_{i-1} + A^E_i U_{i+1} + F_i \quad (1 \le i \le n-1) \tag{18}$$

where $U_0 = 0, U_n = 0$, F_i is given by the data $\{f, \sigma, g\}$, A_i^ϵ is a tridiagonal matrix defined by

$$A_i^\epsilon = \begin{pmatrix} a_{i,1}^\epsilon & -a_{i,1}^N & 0 & \cdots & \cdots & \cdots & 0 \\ -a_{i,2}^S & a_{i,2}^\epsilon & -a_{i,2}^N & 0 & \vdots & \vdots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 & \vdots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & 0 & -a_{i,n-2}^S & a_{i,n-2}^\epsilon & -a_{i,n-2}^N \\ 0 & \cdots & \cdots & \cdots & 0 & -a_{i,n-1}^S & a_{i,n-1}^\epsilon \end{pmatrix} \tag{19}$$

A_i^W, A_i^E are the diagonal matrices given by

$$A_i^W = \mathrm{diag}[a_{i,j}^W]_{1\le j\le n-1} \text{ and } A_i^E = \mathrm{diag}[a_{i,j}^E]_{1\le j\le n-1} \tag{20}$$

and

$$a_{i,j}^\epsilon = a_{i,j}^W + a_{i,j}^E + a_{i,j}^S + a_{i,j}^N.$$

Moreover considering the geometry of the domain Ω and the interface Γ we first introduce sets of interface lattice points Γ_h and boundary lattice points $\partial\Omega_h$ as follows;

(i) $\Gamma_h = \{P_{i,j} = (ih, jh) \mid \Gamma \cap \omega_{i,j}^1 \neq \emptyset\}$,

(ii) $\partial\Omega_h = \overline{\Omega}_h \setminus \Omega_h$.

Using the above notion, we divide the unknown vector $\{U_i\}$ into two parts.

For each $U_i = \{u_{i,j}\}_{1\le j\le n-1}$, we define $U_i' = \{u_{i,j}'\}_{1\le j\le n-1}$ and $W_i = \{w_{i,j}\}_{1\le j\le n-1}$ as follows;

$$u_{i,j}' = \begin{cases} 0 & \text{if } P_{i,j} \in \Gamma_h \\ u_{i,j} & \text{if } P_{i,j} \in \Omega_h \backslash \Gamma_h, \end{cases} \qquad w_{i,j} = \begin{cases} u_{i,j} & \text{if } P_{i,j} \in \Gamma_h \\ 0 & \text{if } P_{i,j} \in \Omega_h \backslash \Gamma_h \end{cases} \tag{21}$$

and devide U_i into two parts by

$$U_i = U_i' + W_i. \tag{22}$$

We then introduce the new vector $\{V_i\}$ defined by

$$V_i = A_i^W U_i' \ (= A_i^E U_i') \qquad (1 \le i \le n-1). \tag{23}$$

From the definition of $\{U_i'\}$ and $\{V_i\}$, we get ([2])

Lemma 1. *$A_i^\epsilon U_i' = B V_i$, $A_i^w U_{i-1}' = V_{i-1}$ and $A_i^E U_{i+1}' = V_{i+1}$ hold $(1 \le i \le n-1)$. Here B is a block tridiagonal matrix in the discretization of the Laplace operator in Ω with homogeneous Dirichlet boundary conditions. i.e., $B = [b_{ij}]$ is an $(n-1) \times (n-1)$ tridiagonal matrix such that $B = \mathrm{tridiag}[-1, 4, -1]$.*

Therefore the following equations are derived from (18),(22),(23) and Lemma 1.

Problem PN. Find $\{V_i , W_i\}$ such that for $i (1 \le i \le n-1)$,

$$B V_i = V_{i-1} + V_{i+1} + F_i + \left(A_i^W W_{i-1} - A_i^\epsilon W_i + A_i^E W_{i+1} \right) \tag{24}$$

where $V_0 = V_n = W_0 = W_n = 0$.

3.2 Formulation of SEL

Applying the principle of the successive elimination of lines to (24), we have

Theorem 2. *$\{W_i\}_{1\le i\le n-1}$ in (24) is uniquely determined as follows.*

$$\begin{aligned} &\textstyle\sum_{i=1}^{k-1} {}^tP_l\ D(n-k,i)\ P\Big(-A_i^W W_{i-1} + A_i^\epsilon W_i - A_i^E W_{i+1} \Big) \\ +\ &\textstyle\sum_{i=k}^{n-1} {}^tP_l\ D(k,n-i)\ P\Big(-A_i^W W_{i-1} + A_i^\epsilon W_i - A_i^E W_{i+1} \Big) \\ =\ &{}^tP_l \Big(\textstyle\sum_{i=1}^{k-1} D(n-k,i)\ P\ F_i + \sum_{i=k}^{n-1} D(k,n-i)\ P\ F_i \Big) \end{aligned} \tag{25}$$

for (k,l) such that $m_{k,l}\in \Gamma_h$.

Here $P(=(p_{i,j})_{1\le i,j\le n-1})$ is the othogonal matrix such that

$$p_{i,j} = \sqrt{\frac{2}{n}}\ sin\left(\frac{ij\pi}{n}\right)\ (1\le i,j\le n-1) \tag{26}$$

and $D(\ l,\ i)\ (1\le l,i\le n-1)$ is a diagonal matrix de*fi*ned by

$$D(l,i) = \mathrm{diag}\left[\Big(\sinh(l\ \lambda_j)\ \sinh(i\ \lambda_j)\Big)/\Big(\sinh(n\ \lambda_j)\ \sinh(\lambda_j)\Big)\right]_{1\le j\le n-1} \tag{27}$$

where $\lambda_j = \mathrm{arccosh}(2-\cos(j\pi/n))$.

Remark 1. $\{W_i\}$ corresponds to the approximate solution of (9).

Remark 2. Linear system w.r.t to $\{W_i\}$ is dramatically less than the total system w.r.t $\{u_{i,j}\}$. For example, let $\Omega=(-0.5,0.5)\times(-0.5,0.5)$ and $\Gamma: x^2+y^2=(1/4)^2$. Then using the notation $\{w_{ij}\}/\{u_{i,j}\}$ which means 'ratio' that is the percentage of the number of unknowns $\{w_{ij}\}$ to the total number of unknowns $\{u_{ij}\}$, we get the following table.

Table 1. Ratio

Grids	$\{w_{ij}\}/\{u_{i,j}\}$
128^2	2.68%
256^2	1.34%

On the other hand, the remainder part $\{V_k\}_{1\le k\le n-1}$ of $\{U_i\}_{1\le i\le n-1}$ is calculated by the following linear algebra in stead of solving the linear systems.

Theorem 3. *V_k is determined by*

$$\begin{aligned} V_k\ =&\textstyle\sum_{i=1}^{k-1} P\ D(n-k,i)\ P\Big(A_i^W\ W_{i-1} - A_i^\epsilon\ W_i + A_i^E\ W_{i+1} \Big) \\ &+\textstyle\sum_{i=k}^{n-1} P\ D(k,n-i)\ P\Big(A_i^W\ W_{i-1} - A_i^\epsilon\ W_i + A_i^E\ W_{i+1} \Big) \\ &+\ P\Big(\textstyle\sum_{i=1}^{k-1} D(n-k,i)\ P\ F_i + \sum_{i=k}^{n-1} D(k,n-i)\ P\ F_i \Big). \end{aligned} \tag{28}$$

From now on a new numerical algorithm using FFT is derived from Theorem 2 and 3.

4 Hybrid Numerical Algorithm Based on FFT and SEL

4.1 Hybrid Numerical Algorithm

Recalling Theorem 3 in the previous section, we notice that the essential part is the calculation of $P\alpha$ for any vector $\alpha =^t (\alpha_1, \cdots, \cdots, \alpha_{n-1})$ and that the row vectors of P are bases of the discrete sine transformation.
In fact,

$$P\alpha = \sqrt{\frac{2}{n}} \begin{pmatrix} sin(\frac{1}{n}\pi) & sin(\frac{2}{n}\pi) & \cdots & \cdots & sin(\frac{n-1}{n}\pi) \\ \vdots & \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & \vdots & \\ sin(\frac{n-1}{n}\pi) & sin(\frac{2(n-1)}{n}\pi) & \cdots & \cdots & sin(\frac{(n-1)(n-1)}{n}\pi) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \vdots \\ \alpha_{n-1} \end{pmatrix}$$

from which the i-th component β_i of $P\alpha$ has the form of

$$\beta_i = \sum_{j=1}^{n-1} p_{i,j}\alpha_j = \sum_{j=1}^{n-1} \sqrt{\frac{2}{n}} \sin(\frac{ij}{n}\pi) \cdot \alpha_j \ . \tag{29}$$

This means that $\{\beta_i\}$ is exactly the discrete sine transform for data α.
Hence we are able to establish a numerical algorithm coupled with FFT and SEL since it is possible to make a program of the discrete sine transform via FFT(see [8]).
Therefore we summarize our numerical algorithm as follows.

Hybrid Numerical Algorithm
1st step: Calculate the solution $\{W_i\}$ on Γ_h for the linear system (25).
2nd step: Compute $\{V_k\}$ on $\Omega_h \backslash \Gamma_h$ by use of the formulation in Theorem 3 and FFT we stated here.

4.2 Comparison of Calculation Speed

Let A and B denote the actual computing time using the matrix calculation in Theorem 3 and the above hybrid calculation respectively. Then we get the table under my computer condition as follows.

Table 2. Calculation speed

Grids	A/B
128^2	4.1
256^2	7.1

Table 3. Computer condition

Pentium 4	2.4GHz
Memory	512MB
Soft	Maple 8

5 Conclusions and Further Works

In this paper we described how to calculate numerically the diffraction problem by use of FFT. The formula of SEL was a mathematical approach and we showed that the hybrid calculation coupled with FFT and SEL was efficient to solve the diffraction problem.
In near future we will apply this method to the heat problem near earth's surface in environments which is described by the systems of diffusion equations with transmission conditions.

References

1. H. Kawarada, Free boundary problem - theory and numerical method - , Tokyo University Press (1989) (in Japanese) .
2. H. Koshigoe and K. Kitahara, Numerical algorithm for finite difference solutions constructed by fictitious domain and successive eliminations of lines, Japan SIAM, Vol.10, No.3 (2000), 211-225 (in Japanese).
3. H. Koshigoe, Direct method for solving a transmission problem with a discontinuous coefficient and the Dirac distribution, Computational Science-ICCS 2003, Part III (2002), 388-400.
4. O.A. Ladyzhenskaya, The boundary value problems of mathematical physics, 49, Springer-Verlarg (1985).
5. J.L. Lions, Optimal control of systems governed by partial differential equations, 170, Springer-Verlarg (1971).
6. G.I. Marchuk,Y.A. Kuznetsov and A.M. Matsokin, Fictitious domain and domain decomposition methods, Sov.J.Numer.Anal.Math.Modelling,Vol.1,No.1 (1986) 3-35.
7. S. Mizohata, The theory of partial differential equations, Cambridge at the University Press, 1973.
8. W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in C, Cambridge University Press, 1988.

Non-negative Matrix Factorization for Filtering Chinese Document*

Jianjiang Lu[1,2,3], Baowen Xu[1,2], Jixiang Jiang[1], and Dazhou Kang[1]

[1] Department of Computer Science and Engineering, Southeast University, Nanjing, 210096, China
[2]Jiangsu Institute of Software Quality, Nanjing, 210096, China
[3]PLA University of Science and Technology, Nanjing, 210007, China
jjlu@seu.edu.cn

Abstract. There are two nasty classical problems of synonymy and polysemy in the filtering systems of Chinese documents. To deal with these two problems, we would ideally like to represent documents not by words, but by the semantic relations between words. Non-negative matrix factorization (NMF) is applied to dimensionality reduction of the words space. NMF is distinguished from the latent semantic indexing (LSI) by its non-negativity constraints. These constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. Also, NMF computation is based on the simple iterative algorithm; it is therefore advantageous for applications involving large sparse matrices. The experimental results show that, comparing with LSI, NMF method not only improves filtering precision markedly, but also has the merits of fast computing speed and less memory occupancy.

1 Introduction

Automatic filtering of information from document sources has become increasingly important in recent years. Information filtering systems are designed to shift through large quantities of dynamically generated documents and display only those which may be relevant to a user's interests [1]. Two major types of filtering systems have been proposed: content-based filtering [2] and collaborative filtering. Collaborative filtering selects documents based on user's evaluations of the documents. On the other hand, content-based filtering selects documents based on the contents of documents and each user's preference.

There exist several types of content-based filtering systems. In the vector space model [3], user profiles and document profiles are represented as weighted vectors of the words in the system. The relevance of each document to each user is calculated according to the similarity between the user profile vector and the document profile vector. There are two nasty classical problems of synonymy and polysemy in the vector space model. To deal with these two and other similar problems, we would

* This work was supported in part by the Young Scientist's Fund of NSFC (60373066, 60303024), National Grand Fundamental Research 973 Program of China (2002CB312000), National Research Foundation for the Doctoral Program of Higher Education of China.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 113–120, 2004.
© Springer-Verlag Berlin Heidelberg 2004

ideally like to represent documents not by words, but by the semantic relations between words.

Latent semantic indexing (LSI) analysis based on singular-value decomposition (SVD) [4, 5] is an information retrieval method that attempts to capture the semantic relations by using techniques from linear algebra. LSI constructs a low-dimensional semantic space wherein words and documents that are closely associated are placed near one another. SVD allows the arrangement of the space to reflect the major associative patterns in the data, and ignore the smaller, less important influences. However, the cost of SVD computation will be prohibitive when matrices become large. In addition, SVD is lack of intuitive notion.

In this paper, a method based on non-negative matrix factorization (NMF) [6, 7] for constructing Chinese user profile is presented. This method proposes to apply NMF to dimensionality reduction of the document vectors. NMF can decompose a non-negative matrix into two non-negative matrices. One of the decomposed matrices can be regarded as the basis vectors. The dimensionality reduction can be performed by projecting the document vectors onto the lower dimensional space which is formed by these basis vectors. NMF is distinguished from LSI by its non-negativity constraints. These constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. Also, NMF computation is based on the simple iterative algorithm; it is therefore advantageous for applications involving large sparse matrices.

The remainder of this paper is organized as follows. In section 2, we briefly review how to represent a set of unstructured Chinese documents as a vector space model. In section 3, we introduce non-negative matrix factorization. In section 4, a NMF method for constructing Chinese user profile is presented. In section 5, the experimental results of NMF method are compared with LSI. Finally, section 6 gives the conclusions.

2 Vector Space Models for Documents

Let $D_1 = \{d_1^1, d_2^1, \cdots, d_{n_1}^1\}$ be a set of Chinese topic documents let $D_2 = \{d_1^2, d_2^2, \cdots, d_{n_2}^2\}$ be a set of non-topic documents, $n_1 + n_2 = n$, $D = D_1 \cup D_2$. We briefly review how to represent a set of unstructured Chinese documents as a vector space model. The pre-processing is as following.

(1) Chinese documents are written as characters strings with no spaces between words, so we first use word segmentation algorithm [8] to segment the Chinese documents.

(2) After word segmentation, we eliminate non-content-bearing "stopwords".

(3) Using heuristic or information-theoretic criteria, eliminate non-content-bearing "high-frequency" and "low-frequency" words. Such words and the stopwords are both known as "function" words. Eliminating function words removes little or no information, while speeding up the computation.

(4) After above elimination, suppose m unique words remain, let be $T = \{t_1, t_2, \cdots, t_m\}$. We use normalized word frequency-inverse document frequency

scheme [9] to obtain word-document matrix $X = (x_{ij})_{m\times n}$. The i^{th} element x_{ij} of the document vector $x_j = (x_{1j}, x_{2j}, \cdots, x_{mj})^T$ is given by

$$x_{ij} = \#(x_j, t_i)\log\frac{n}{h_i}$$

where $\#(x_j, t_i)$ denotes the number that the word t_i appears in the document x_j, h_i denotes the number of the documents in which the word t_i appears, n is the total document number. Document vectors are usually normalized to a unit vector, that is,

$$x_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}, i = 1,2,\cdots,m$$

Intuitively, the effect of normalization is to retain only the direction of the document vectors. This ensures that documents dealing with the same subject matter (that is, using similar words), but differing in length lead to similar document vectors.

After the preprocessing, Chinese documents are represented as m dimensional document vectors $x_j, j = 1,2,\cdots,n$. Let $X_1 = \{x_1^1, x_2^1, \cdots, x_{n_1}^1\}$ be topic document vectors, $X_2 = \{v_1^2, v_2^2, \cdots, v_{n_2}^2\}$ be non-topic document vectors, $X = X_1 \cup X_2$. These document vectors make up of word-document matrix $X = (x_{ij})_{m\times n}$.

3 Non-negative Matrix Factorization

Given a non-negative matrix $X = (x_{ij})_{m\times n}$, NMF finds the non-negative $m\times r$ matrix $U = (u_{ij})_{m\times r}$ and the non-negative $r\times n$ matrix $V = (v_{ij})_{r\times n}$ such that

$$X \approx UV \tag{1}$$

The r is generally chosen to satisfy $(n + m)\, r < nm$, so that the product UV can be regarded as a compressed form of the data in X.

The equation (1) can be rewritten column by columns as

$$x \approx Uv \tag{2}$$

where x and v are the corresponding columns of X and V. Each vector x is approximated by a linear combination of the columns of U, weighted by the components of v. Therefore, U can be regarded as containing a basis vector that is optimized for the linear approximation of the vector in X. Since relatively few basis vectors are used to represent many vectors, good approximation can only be achieved if the basis vectors discover structure that is latent in the vectors.

Here, we introduce an algorithm based on iterative estimation of U and V. At each iteration of the algorithm, the new value of U and V is found by multiplying the current value by some factor that depends on the quality of the approximation in equation (1). Repeated iteration of the update rules is guaranteed to converge to a locally optimal matrix factorization.

The update rules given in the next equations [7].

$$v_{ij} \leftarrow \sqrt{v_{ij} \sum_k u_{ki} \frac{x_{ki}}{y_{kj}}} \tag{3}$$

$$u_{ij} \leftarrow u_{ij} \sum_k \frac{x_{ik}}{y_{ik}} v_{jk} \tag{4}$$

$$u_{ij} \leftarrow \frac{u_{ij}}{\sum_k u_{kj}} \tag{5}$$

where U and V are initial stochastic matrices.

The update rules maximize the following objective function:

$$F(X,Y) = \sum_{i,j} \left(x_{ij} \log \frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij} \right) + \alpha \sum_{i,j} a_{ij} - \beta \sum_i b_{ii} \tag{6}$$

where a_{ij} is the components of $U^T U$, b_{ii} is the diagonal components of VV^T, $\alpha, \beta > 0$ are some constants, $Y = UV = (y_{ij})_{m \times n}$.

4 Constructing Chinese User Profile

After word-document matrix X is decomposed by the NMF in section 3, the m dimensional document vectors are projected into the r dimensional vectors. Let $V_1 = \{v_1^1, v_2^1, \cdots, v_{n_1}^1\}$ be projecting of the topic document vectors, $V_2 = \{v_1^2, v_2^2, \cdots, v_{n_2}^2\}$ be projecting of the non-topic document vectors. We can compute the mean vector of topic document vectors.

$$O = (o_1, o_2, \cdots, o_m) = \frac{1}{n_1} \sum_{j=1}^{n_1} x_j^1$$

In the same way, we can compute the mean vector of projecting of the topic document vectors.

$$O^1 = (o_1^1, o_2^1, \cdots, o_r^1) = \frac{1}{n_1} \sum_{j=1}^{n_1} v_j^1$$

According to equation (2), we can easily obtain.

$$O \approx UO^1 = u_1 o_1^1 + u_2 o_2^1 + \cdots + u_r o_r^1 \tag{7}$$

A simply way is to select UO^1 as a user profile, but this user profile may be ineffective. Next, we define the class discriminative degree of the basis vectors in order to obtain an effective user profile.

Definition1. Let $v_j^1 = (v_{1j}^1, v_{2j}^1, \cdots, v_{rj}^1)^T$, $j = 1,2,\cdots,n_1$, $v_j^2 = (v_{1j}^2, v_{2j}^2, \cdots, v_{rj}^2)^T$, $j = 1,2,\cdots,n_2$ Class discriminative degree of the basis vector u_s to topic documents is defined as follows:

$$d_s = \frac{1}{n_1}\sum_{j=1}^{n_1} v_{sj}^1 - \frac{1}{n_2}\sum_{j=1}^{n_2} v_{sj}^2 \ , \ s = 1,2,\cdots,r \tag{8}$$

If the average weight $\frac{1}{n_1}\sum_{j=1}^{n_1} v_{sj}^1$ of the basis vector u_s in the topic documents is big, and the average weight $\frac{1}{n_2}\sum_{j=1}^{n_2} v_{sj}^2$ of the basis vector u_s in the non-topic documents is small, then the class discriminative degree d_s is large. That is to say, the basis vector u_s has strong discriminative ability between topic documents and the non-topic documents. We select *k* basis vectors with big class discriminative degrees, simply let be $u_1, u_2, \cdots, u_k, k \le r$. According to equation (7), we use these k basis vectors to construct a *m* dimensional vector of the words as follows.

$$u_1 o_1^1 + u_2 o_2^1 + \cdots + u_k o_k^1 \tag{9}$$

Then, we sort the components of the *m* dimensional vector by the value, and select *l* components with big values as the user profile.

$$User_Profile = \{< t_1, g_1 >, < t_2, g_2 >, \cdots, < t_l, g_l >\}$$

where t_i is a word, g_i is the component value with respect to word t_i, $i = 1,2,\cdots,l$.

5 Experimental Results

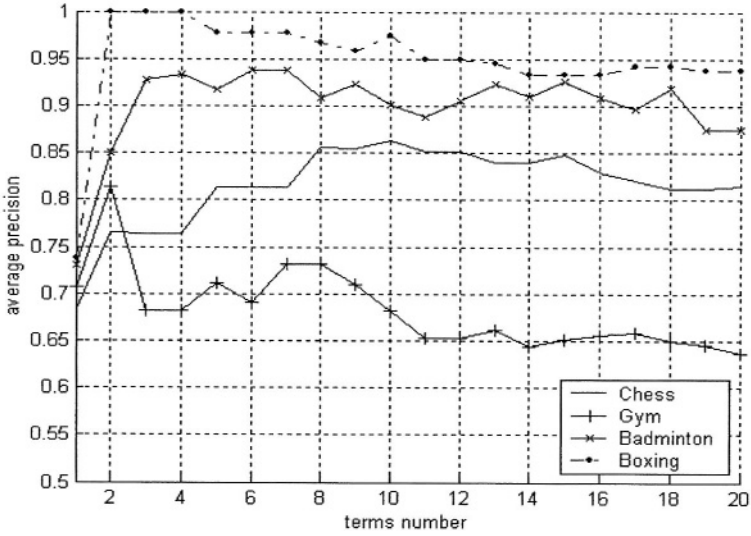
All documents in the experiment are downloaded from http://www.sina.com.cn. The topic documents include: Chess (277), Gym (149), Badminton (110), Box (69), Ping-Pong (177), Volleyball (171), Racing (95), Swimming (126), Tennis ball (208), Baseball (60), Skating (155), Golf (111), Track and field (147), Billiards (58) and Martial art (43). In addition, there have 2044 non-topic documents.

In order to test 15 topic documents, we partition each topic documents into four groups, and select one group as testing documents, other groups as training documents. When recall was set to 0%, 10%, 20%, 30%, ..., 90%, 100%, the average precision at 11 points of filtering systems based on NMF and LSI [5] are compared.

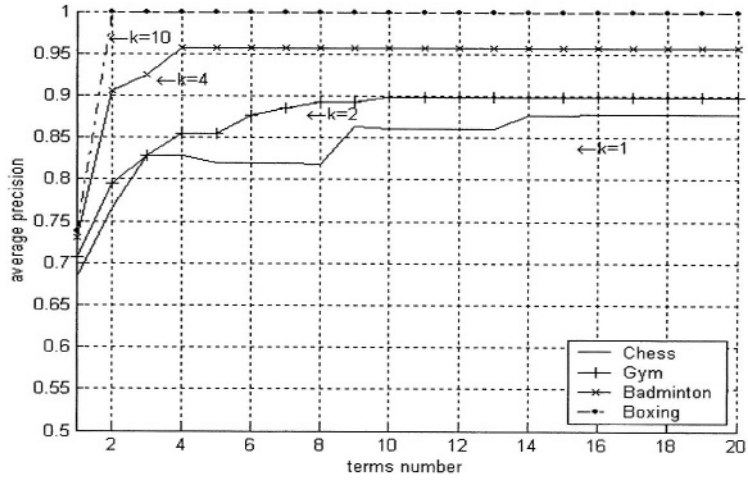
We select 2843 words in the experiment, let r = 200, figure 1 shows the experimental results on the four topic documents. Figure 1(a) and figure 1(b) show the average precisions of filtering systems based on NMF and LSI with the selecting words respectively. In the filtering systems based on NMF, we select k basis vectors that have strong discriminative ability between topic documents and the non-topic documents. Figure 2 shows the average precisions of 15 topic documents.

The experimental results show that the average precision of filtering systems based on NMF is better than LSI. In addition, the memory occupancy of NMF is lesser than LSI. For example, the memory occupancy of the left singular matrix in LSI is

126.3M, and the memory occupancy of the right singular matrix is 33.0M. Whereas the memory occupancy of the left non-negative matrix in NMF is 6.9M, and the memory occupancy of the right non-negative matrix is 6.7M. Furthermore, NMF only needs 30-40 iterative times, so NMF costs less computation time than LSI. In the experiment, using the SVDPACK from http://www.netlib.org/svdpack, NMF only costs about half time of LSI.



(a) LSI (r=200)



(b) NMF (r=200)

Fig. 1. Comparing the precision

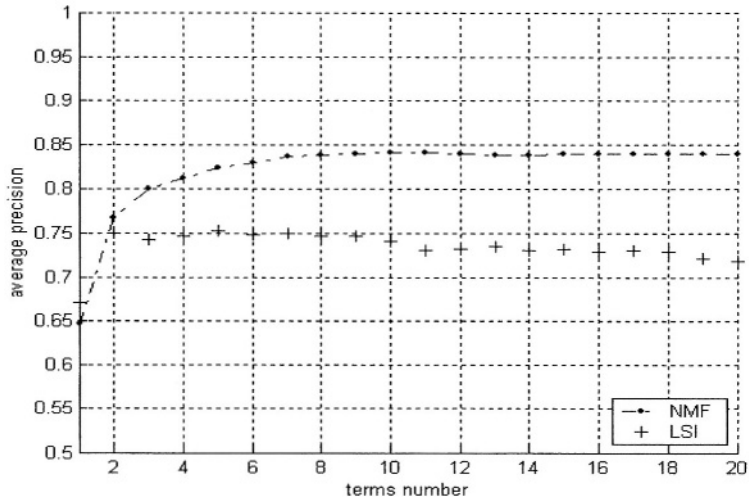


Fig. 2. Comparing average precision

6 Conclusion

Automatic filtering of information from document sources has become increasingly important as the volume of electronically accessible documents has exploded in recent years. In this paper, a method based on NMF for constructing Chinese user profile is presented. This method proposes to apply NMF to dimensionality reduction of the document vectors in the word-document matrices. NMF decomposes a non-negative matrix into two non-negative matrices. One of the decomposed matrices can be regarded as the basis vectors. The dimensionality reduction can be performed by projecting the document vectors onto the lower dimensional space which is formed by these basis vectors. NMF is distinguished from LSI by its non-negativity constraints. These constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. Also, NMF computation is based on the simple iterative algorithm, it is therefore advantageous for applications involving large sparse matrices. The experimental results show that, comparing with LSI, NMF method not only improves filtering precision markedly, but also has the merits of fast computing speed and less memory occupancy. In the future work, we will discuss how to use NMF in the Chinese document clustering and classification.

References

1. Belkin, N.J., Croft, W. B.: Information Filtering and Information Retrieval: two sides of the same coin. Communication of ACM, 35(12). (1992) 29-38
2. Chen, L., Katia, S.: WebMate: A Personal Agent for Browsing and Searching. ACM AGENTS'98, Proceedings of the International Conference on Autonomous Agents, Minneapolis (1998) 132-139
3. Yart, T. W., Garcia-Molina, H.: Index Structures for Information Filtering under the Vector Space Model. Proceedings of the 10th International Conference on Data Engineering, Alamitos, CA, IEEE (1994) 337-347
4. Papadimitriou, C. H., Raghavan, P., Tamaki, H.: Latent Semantic Indexing: A Probabilistic Analysis. Proceedings of PODS'98, Seattle, WA (1998)159-168
5. Lu, Z., Lu, H., Li, Y.: FDS Expressive Method in Information Filtering. Journal of Tsinghua University (science and technology), 39(9). (1999)118-121
6. Lee, D. D., Seung, H. S.: Learning the Parts of Objects by Non-negative Matrix Factorization. Nature, 401. (1999) 788-791
7. Li, S. Z., Hou, X. W., Zhang, H. J.: Learning Spatially Localized Parts-based Representation. Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, Hawaii (2001) 207 –212
8. Chen, G. L., Wang, Y. C., HAN, K. S., Wang, G.: An Improved Fast Algorithm for Chinese Word Segmentation. Journal of Computer Research and Development, 37(4). (2000) 418–424
9. Kolda, T. G.: Limited–Memory Matrix Methods with Applications. Ph.D. thesis, The Applied Mathematics Program, University of Maryland, College Park, Mayland (1977)

On Highly Secure and Available Data Storage Systems*

Sung Jin Choi[1], Hee Yong Youn[1], and Hyung Soo Lee[2]

[1] School of Information and Communications Engineering
Sungkyunkwan University, Suwon, Korea
{choisj, youn}@ece.skku.ac.kr
[2] Ubiquitous System Research Center,
Korea Electronics Technology Institute, Pyoungtaek, Korea
hslee@keti.re.kr

Abstract. Rapid technological advances are resulting in a greater use of data intensive applications. For this reason and that of the alarming growth in electronic crime, security and availability are critical factors that must be considered in designing a system. This paper proposes a novel data dispersal/encryption scheme to improve both the availability and security of distributed storage system by using the Singular Values Decomposition (SVD) theorem. It handles data represented by any size matrix, and it also allows complete recovery of original data even though stored data are partially damaged. Analysis shows that it improves the availability about 10% compared with an efficient existing scheme for both read and write operation, while it allows secure storage simultaneously.

1 Introduction

Rapid technological advances are resulting in a greater use of distributed storage system in world-wide organizations for communication, transport, banking, education, manufacturing, medicine, etc., leading to the handling of large quantities of data that are especially important for these organizations on many occasions. For this reason and that of the alarming growth in electronic crime, security becomes a serious aspect that must be considered specifically for distributed storage system. At the same time, due to harsh and unpredictable distributed environment, the system needs to be designed to provide continuous service even in the presence of failures. Indeed, both security and availability are two important criteria for distributed storage systems [1].

The main issues focused in typical distributed systems including database systems mainly have been with the following aspects: confidentiality, integrity, and availability. The range of possible threats that affect each one of the three factors is so wide that they cannot be tackled as a whole. Data distribution is one of the key techniques developed for achieving a desired level of security and availability, and it involves data encoding and partitioning algorithm. There exist many such algorithms applicable to distributed storage system including encryption, replication, striping, secret

* This work was supported by 21C Frontier Ubiquitous Computing and Networking, Korea Research Foundation Grant (KRF - 2003 - 041 - D20421) and Brain Korea 21 Project in 2004. Corresponding author : Hee Yong Youn

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 121–129, 2004.
© Springer-Verlag Berlin Heidelberg 2004

sharing, and various combinations of them. They offer different levels of performance (throughput), availability (probability that data can be accessed), and security (effort required to compromise the confidentiality or integrity of stored data). Development of new efficient data distribution and protection scheme is inevitable for allowing highly available but secure system at the same time without significantly affecting the performance [2].

In distributed storage system security and availability of data are greatly influenced by the policy how the data are dispersed. High security and availability are usually conflicting requirement. This paper, however proposes a new scheme that can allow high security and availability of distributed storage system. It is achieved by using the SVD theorem which is an important matrix decomposition that applicable to any size matrix. A scheme targeting the same goal was presented in [3] using matrix eigenvalue. However, it can handle data represented by only square matrix and eigenvalues are not guaranteed to be found, while the proposed scheme solves both the problems. It also allows complete recovery of original data even though stored data are partially damaged. Analysis shows that the availability is improved about 10% compared with information dispersal scheme, while it allows secure storage since decryption without the key is absolutely impossible.

The rest of the paper is organized as follows. Section 2 discusses existing approaches for data encryption and distribution, and Section 3 presents the proposed scheme. Section 4 evaluates and compares the performance of the proposed scheme with the earlier schemes, and finally concluding remark is given in Section 5.

2 Related Work

There is a wide array of data distribution algorithms including encryption, replication, striping, erasure-resilient coding, information dispersal, and secret sharing. Threshold algorithms, characterized by three parameters *(p, m,* and *n),* represent a large set of these algorithms. In a *p-m-n* threshold scheme, data is encoded into *n* shares such that any *m* of the shares can reconstruct the data and less than *p* reveal no information about the encoded data. Thus, a stored value is available if at least *m* of the *n* shares can be retrieved. Attackers must compromise at least *p* storage nodes before it is even theoretically possible to determine any part of the encoded data. Table I lists a number of well-known threshold schemes [4].

Table 1. Threshold schemes

p-m-n	Scheme
1-1-n	Replication
1-n-n	Striping
n-n-n	Splitting
1-m-n	Information Dispersal
m-m-n	Secret Sharing
p-m-n	Ramp Schemes

The simplest example of threshold scheme is n-way replication, which is a *1-1-n* threshold scheme. That is, out of the *n* replicas that are stored, any single replica provides the original data (*m = 1*), and each replica reveals information about the encoded data (*p = 1*). Another example is striping (or decimation, as in disk arrays), wherein a large block of data is partitioned into *n* sub-blocks, each containing one partition of the data (so, *p = 1* and *m = n*). At the other end of the example is splitting, an *n-n-n* threshold scheme that consists of storing *n-1* random values and one value is the exclusive-or of the original value and those *n-1* values; *p = m = n* for splitting, since all *n* shares are needed to recover the original data. Replication, decimation, and splitting schemes have a single tunable parameter, *n*, which affects their place in the trade-off space. With more mathematics, the full range of *p-m-n* threshold schemes becomes available. For example, secret sharing schemes are *m-m-n* threshold schemes. Shamir's implementation of secret sharing is based on interpolating points on a polynomial in a finite field. The secret values along with *m-1* randomly generated values uniquely determine the encoding polynomial of order *m-1*. Each share is generated by evaluating the polynomial at distinct points.

Information dispersal algorithm, a *1-m-n* threshold scheme, uses the same polynomial based mathematics as Shamir's secret sharing, but no random number; *m* secret values are used to determine the unique encoding polynomial. Thus, each share reveals partial information about the *m* simultaneously encoded values, but the encoding is much more space efficient. Ramp schemes are *p-m-n* threshold schemes, and they can also be implemented with the same approach. The points used to uniquely determine the encoding polynomial are *p-1* random values and *m-(p-1)* secret values. Ramp schemes thus theoretically offer confidentiality of up to *p-1* shares. They are also more space efficient than secret sharing (so long as *m>p*). For *p=1*, ramp schemes are equivalent to information dispersal; for *p=m,* they are equivalent to secret sharing [5].

3 The Proposed Scheme

We first present the theorem and definition required to explain the proposed encryption/dispersal scheme.

3.1 SVD's Theorem and Singular Values (Finding V, D, and U)

Theorem 1: Let A be an $m \times n$ matrix and $\sigma_1, ..., \sigma_r$ be all its nonzero singular values. Then there are orthogonal matrices U ($m \times m$) and V ($n \times n$) and an $m \times n$ matrix D of the form such that

$$A = UDV^T. \tag{1}$$

Proof 1: U, V, and D (of the indicated sizes) have been already explicitly defined. Moreover, U and V are orthogonal. It remains to show only that $A = UDV^T$. It suffices to show that AV = UD, because $V^T = V^{-1}$.

$$u_i = \frac{1}{\sigma_i} A v_i \text{ for } i = 1,...,r . \tag{2}$$

By Eq. (2) $\sigma_i u_i = A v_i$ for i = 1,....,r and $\|Av_i\| = \sigma_i = 0$ for i = r+1,...,n. So, $Av_i = 0$ for i= r+1,...,n . Therefore,

$$AV = [Av_1 \cdots Av_n] = [\sigma_1 u_1 \cdots \sigma_r u_r 0 \cdots 0]$$
$$= [u_1 \cdots u_m] \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \sigma_r & \vdots \\ 0 & \cdots & 0 \end{bmatrix} . \tag{3}$$

Let A be an n × n matrix. A nonzero vector v is an eigenvector of A if Eq. (4) holds for some scalarλ. λ is called an eigenvalue of A corresponding to the eigenvector v. Eigenvalues are also known as characteristic, proper values, or latent roots.

$$Av = \lambda v . \tag{4}$$

We define V, and then find the σ_i's along the diagonal of D. Consider the n × n symmetric matrix $A^T A$. By the spectral theorem, $A^T A$ is orthogonally diagonalizable and has eigenvalues, say $\lambda_1,...,\lambda_n$. Let $v_1,...,v_n$ be the corresponding eigenvectors so that they form an orthonormal basis of R^n . V is simply

$$V = [v_1 v_2 \cdots v_n] . \tag{5}$$

Next, we observe that all the eigenvalues are nonnegative (so $A^T A$ is positive semidefinite). Because $(A^T A)v_i = \lambda_i v_i$ and $\|v_i\| = 1$, we have $0 \le \|Av_i\|^2 = (Av_i)^T Av_i = V_i^T A^T Av_i = v_i^T \lambda_i v_i = \lambda_i \|v_i\|^2 = \lambda_i$. Hence, $\lambda_i \ge 0$ for i=1,...,n. By renumbering, if necessary, we order the λ_i values from largest to smallest and take their square roots, such that $\sigma_1 = \sqrt{\lambda_1} \ge \cdots \ge \sigma_n = \sqrt{\lambda_n} \ge 0$. So, $\sigma_i = \|Av_i\|$, i = 1,.....,n .The numbers $\sigma_1,...,\sigma_n$ are called the singular values of A, and they carry important information on A. These are the diagonal entries of D [6].

3.2 Encryption/Dispersal Scheme

First, we find the singular values of the elements of D. Therefore, we need a general method for finding eigenvalue and eigenvector by using Eq. (4) and it is the Power Method. It computes the dominant eigenvalue and an eigenvector corresponding to the dominant eigenvalue. Without loss of generality, it is necessary to assume that A has the following two properties:

i. There is a single eigenvalue of maximum modulus.
ii. There is a linearly independent set of n eigenvectors.

According to the first assumption, the eigenvalues can be labeled such that $|\lambda_1| > |\lambda_2| \ge |\lambda_3| \ge \cdots \ge |\lambda_n|$. According to the second assumption, there is a basis $\{u^{(1)}, u^{(2)},\ldots,u^{(n)}\}$ for C^n such that

$$Au^{(j)} = \lambda_j u^{(j)} \ (1 \le j \le n) . \tag{6}$$

Let $x^{(0)}$ be an element of C^n such that when $x^{(0)}$ is expressed as a linear combination of the basis elements $u^{(1)}, u^{(2)}, ..., u^{(n)}$, the coefficient of $u^{(1)}$ is not 0. Thus,

$$x^{(0)} = a_1 u^{(1)} + a_2 u^{(2)} + \cdots + a_n u^{(n)} \ (a_1 \neq 0) . \tag{7}$$

We form then $x^{(1)}=Ax^{(0)}$, $x^{(2)}=Ax^{(1)}$,L,$x^{(k)}=Ax^{(k-1)}$ to have

$$x^{(k)} = A^k x^{(0)} . \tag{8}$$

In the following analysis there is no loss of generality in absorbing all the coefficients a_j in the vectors $u^{(j)}$ that they multiply. Hence, we may rewrite Eq. (7) as

$$x^{(0)} = u^{(1)} + u^{(2)} + \cdots + u^{(n)} . \tag{9}$$

By this equation and (8), we have $x^{(k)} = A^k u^{(1)} + A^k u^{(2)} + \cdots + A^k u^{(n)}$. Using Eq. (6), we arrive at $x^{(k)} = \lambda_1^k \left[u^{(1)} + \left(\frac{\lambda_2}{\lambda_1}\right)^k u^{(2)} + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^k u^{(n)} \right]$. Since $|\lambda_1| > |\lambda_j|$ for $2 \le j \le n$, we see that the coefficients $\left(\frac{\lambda_j}{\lambda_1}\right)^k$ tend to 0 and the vector within the brackets converges to $u^{(1)}$ as $k \to \infty$. To simplify the notation, we write $x^{(k)}$ in the form $x^{(k)} = \lambda_1^k \left[u^{(1)} + \varepsilon^{(k)} \right]$, where $\varepsilon^{(k)} \to 0$ as $k \to \infty$. In order to be able to take ratios, let φ be any linear functional on C^n for which $\varphi(u^{(1)}) \neq 0$. Then $\varphi(x^{(k)}) = \lambda_1^k \left[\varphi(u^{(1)}) + \varphi(\varepsilon^{(k)}) \right]$. Consequently, the following ratios converges to λ_1 as $k \to \infty$: $r_k \equiv \frac{\varphi(x^{(k+1)})}{\varphi(x^{(k)})} = \lambda_1 \left[\frac{\varphi(u^{(1)}) + \varphi(\varepsilon^{(k+1)})}{\varphi(u^{(1)}) + \varphi(\varepsilon^{(k)})} \right] \to \lambda_1$.

This constitutes the Power Method for computing λ_1. Since the direction of the vector $x^{(k)}$ aligns more and more with $u^{(1)}$ as $k \to \infty$, the method can also give us the eigenvector, $u^{(1)}$. The eigenvectors found are

$$v^{(1)} = [s_1 \cdots s_n] \ s_n \in R, \ v^{(2)} = [t_1 \cdots t_n] \ t_n \in R, \ldots\ldots, \ v^{(n)} = [z_1 \cdots z_n] \ z_n \in R . \tag{10}$$

Therefore, by Eq. (2) and Eq. (5), U ($U = \frac{1}{\sigma_i} A[v^{(1)} v^{(2)} \cdots v^{(n)}]$) and V ($V = [v^{(1)} v^{(2)} \cdots v^{(n)}]$) are actually stored data and matrix D consisting of σ becomes Decryption Key.

Example 1: If the original data is $A = \begin{bmatrix} -2 & 1 & 2 \\ 6 & 6 & 3 \end{bmatrix}$, we first calculate $A^T A$. Then we derive λ_1=81, $\lambda_2 = 9$, $\lambda_3 = 0$ by using Theorem 2. Eigenvectors that correspond to each λ become[2s 2s s], [-2t t 2t], [u -2u 2u] s,t,u$\in R$. U is calculated by us-

ing $U = \frac{1}{\sigma_i} A[v^{(1)} v^{(2)} \cdots v^{(n)}]$. The data actually stored in each node and Decryption Key D and U are as follows. Here s=1, t=2 and u=3 are randomly selected.

Node1=[2 2 1], Node2=[-4 2 4],Node3=[3 -6 6], $U=\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $D=\begin{bmatrix} 9 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}$

An important property of the proposed scheme is that it allows secret dispersal as a general threshold scheme and encryption of data at the same time. That is, one cannot extract original data even though the stored data U and V are available since it is NP-hard problem to decide the original matrix using U and V. Refer to the example above. There exist infinite ways to form V matrix by arbitrarily deciding the s, t, and u values. As a result, the proposed scheme offers high availability and security as well as complete recovery of original data even though data are partially damaged.

3.3 Data Recovery Scheme

We next show how the original data is recovered using Gram-Schmidt Process.

Theorem 2: Any subspace W of R^n has at least one orthogonal basis and at least one orthonormal basis. If $B = \{a_1, ..., a_n\}$ is a basis of W, then $B' = \{b_1, ..., b_n\}$ is an orthogonal basis, where

$$b_1 = a_1, ..., b_k = a_k - \frac{a_k \cdot b_1}{b_1 \cdot b_1} b_1 - \frac{a_k \cdot b_2}{b_2 \cdot b_2} b_2 \cdots - \frac{a_k \cdot b_{k-1}}{b_{k-1} \cdot b_{k-1}} b_{k-1} \tag{11}$$

Proof 2: An orthonormal Basis B'' is obtained by normalizing B':

$$B'' = \left(\frac{b_1}{\|b_1\|}, \ldots, \frac{b_k}{\|b_k\|}\right). \tag{12}$$

Let V $(v_1, ..., v_n)$ be the corresponding eigenvectors, so that they form an orthonormal basis of R^n. V^T is as follows, by Eq. (12).

$$V^T = \left[\frac{v_1}{\|v_1\|} \cdots \frac{v_k}{\|v_k\|}\right] . \tag{13}$$

Therefore, the original data can be recovered from U, V that were stored in the storage nodes by Eq. (1).

Example 2: We obtain V^T using Gram-Schmidt orthonormal Process after reading the data stored in Node1, Node2 and Node3. Therefore,

$v_1=[2, 2, 1]$, $v_2=[-4, 2, 4]$, $v_3=[3, -6, 6]$

$$V_1 = \frac{v_1}{\|v_1\|} = \frac{1}{3} v_1 = \begin{bmatrix} 2/3 \\ 2/3 \\ 1/3 \end{bmatrix}, \quad V_2 = \frac{v_2}{\|v_2\|} = \frac{1}{6} v_2 = \begin{bmatrix} -2/3 \\ 1/3 \\ 2/3 \end{bmatrix}, \quad V_3 = \frac{v_3}{\|v_3\|} = \frac{1}{6} v_3 = \begin{bmatrix} 1/3 \\ -2/3 \\ 2/3 \end{bmatrix},$$

V^T is $\begin{bmatrix} 2/3 & -2/3 & 1/3 \\ 2/3 & 1/3 & -2/3 \\ 1/3 & 2/3 & 2/3 \end{bmatrix}$. We get the original data by equation $A = UDV^T$.

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 9 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}\begin{bmatrix} 2/3 & -2/3 & 1/3 \\ 2/3 & 1/3 & -2/3 \\ 1/3 & 2/3 & 2/3 \end{bmatrix} = \begin{bmatrix} -2 & 1 & 2 \\ 6 & 6 & 3 \end{bmatrix}$$

4 Performance Evaluation

Different parameter values for the threshold scheme (i.e. the values of *n, m* and *p*) create a large class of schemes. Each scheme has different property, and the best scheme needs different parameter values according to the given condition. In order to select an optimal scheme, we need to be able to evaluate and compare the performance of different schemes. In quantifying the schemes, the primary metric is availability.

Availability is defined as the probability that a file can be accessed at any given time. With threshold schemes, files are encoded into *n* shares, of which *m* or more are sufficient to fully reconstruct the file. We assume the failures of storage nodes are independent. The general availability read and write model are

$$\begin{aligned} \text{Availability}_{\text{read}} &= \left(\sum_{i=m}^{n}\binom{n}{i}\times p_a^i\times(1-p_a)^{n-i}\right)^m \\ \text{Availability}_{\text{write}} &= \left(\sum_{i=m}^{N}\binom{N}{i}\times p_a^i\times(1-p_a)^{N-i}\right)^m \end{aligned} \tag{14}$$

Here P_a is the node availability, and it is assumed to be 0.9. Also, $N = n+1$. Figure 1 and 2 show the comparison of read and write availability of the proposed scheme along with striping and information dispersal scheme. Here up to 100 nodes were tested. Note that the m value for striping, information dispersal, and proposed scheme are n, 5, 6, respectively. The figures reveal that the proposed scheme display substantially higher availability than striping while it is consistently better than the information dispersal scheme for about 10% for both read and write operation.

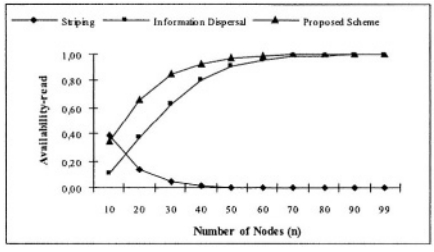


Fig. 1. Comparison of read availabilities.

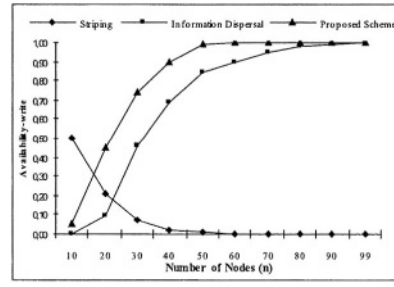


Fig. 2. Comparison of write availabilities.

Figure 3 shows that how availability of the proposed scheme varies as m and n value change. Form the figure we can see that availability gets higher as m decreases and n increases.

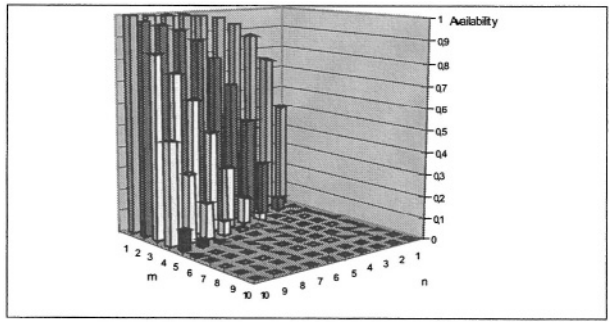


Fig. 3. The availability as m changes from 1 to 10

5 Conclusion and Future Work

Most earlier schemes apply data encryption and distribution separately, which result in some vulnerability and inefficiency. In this paper thus we propose a new approach which integrates data encryption and distribution process into one process to improve both the security and availability by using Singular Values Decomposition theorem. As a result, the proposed scheme offers high availability and security as well as complete recovery of original data even though partial damage occurs to the data. The proposed scheme can handle data represented by any size matrix, and eigenvalues are guaranteed to be found. Analysis shows that it improves the availability about 10% compared with information dispersal scheme for both read and write operation, while it allows secure storage since decryption without the key is absolutely impossible. A new model considering not only availability but also security in a more formal way will be developed.

References

1. Mehmet Bakkaloglu, Jay J. Wylie, Chenxi Wang, Gregory R. Ganger.: On Correlated Failures in Survivable Storage Systems: School of Computer Science Carnegie Mellon University, Pittsburgh, PA15213 (2002)
2. Jay J. Wylie, Michael W. Bigrigg, John D. Strunk, Gregory R. Ganger, Han Kiliccote, Pradeep K. khosla.: Survivable Information Storage systems: IEEE Computer (2000)
3. Sung Jin Choi, Hee Yong Youn, Bo Kyung Lee.: An Efficient Dispersal and Encryption Scheme for Secure Distributed Information Storage: ICCS2003, LNCS2660, Springer (2003) 958-967

4. A. De Santis and B. Masucci.: Multiple Ramp Schemes: IEEE Trans. Information Theory (1999) 1720-1728
5. R. Cannetti, R. Gennaro, S. Jarecki, H. Krawcxyk and T. Rabin.: Adaptive Security for Threshold Cryptosystems: In Advances in Cryptology-Crypto '99, LNCS, Springer (1999) 98-115
6. George Nakos, David Joyner.: Linear Algebra with Applications, Brooks/Cole USA (1998) 562-569

A Numerical Adaptive Algorithm for the Obstacle Problem

F.A. Pérez, J.M. Cascón, and L. Ferragut

Departamento de Matemática Aplicada, Universidad de Salmanca,
37008 Salamanca, Spain
{franc, casbar, ferragut}@usal.es
http://matapli.fis.usal.es

Abstract. This paper concerns an adaptive finite element method for the elliptic obstacle problem. We consider the formulation of the problem as an elliptic variational inequation. The adaptive algorithm (modified Uzawa adaptive method)we construct is based on a combination of the Uzawa method associated with the corresponding multivalued operator and a convergent adaptive method for the linear problem. As our main result we show that if the adaptive method for the linear problem is convergent, then the adaptive modified Uzawa method is convergent as well. A numerical experiment shows the studied properties of the method.

1 Introduction

Adaptive mesh refinement based on a posteriori error estimates is an essential instrument for efficient numerical solving of Partial Differential Equations, in particular nonlinear problems.

The obstacle problem may be considered as a model problem for variational inequalities (see, e.g,[1],[2], [5], [6],[7]). The obstacle problem belongs to a class of problems which have found applications in such diverse fields as elasticity and fluid dynamics(see,e.g,[4],[5],[6],[7]). These problems are characterized by free boundaries that are a priori unknown. An issue to consider is that in their formulation the solution is sought in a convex and not necessarily linear subset of some vector space.

In this paper we construct an adaptive finite element method for the elliptic obstacle problem (adaptive modified Uzawa method). We formulate the variational inequality in terms of a multivalued operator. As it is well known, the Uzawa algorithm consists in solving in each iteration a linear problem and a nonlinear adaptation of the Lagrange multiplier associated with the multivalued equation. As our main result we show that if the adaptive method applied for the linear equation is convergent then, the adaptive modified Uzawa method is convergent as well.

The convergence is proved with respect to a discrete solution in the space corresponding to a sufficiently refined mesh. In order to assure the convergence of the Lagrange multiplier, the space of piecewise constant finite element functions is extended with bubble functions.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 130–137, 2004.
© Springer-Verlag Berlin Heidelberg 2004

We get the following convergence result:

Main result. *Let (U_j, P_j) the sequence of finite element solutions of the linear problem and the corresponding Lagrange multiplier produced by the adaptive modified Uzawa algorithm. There exist positive constants C and $\delta < 1$ such that*

$$\|u - U_j\| + |p - P_j| \leq C\delta^j$$

where $|\cdot|$ is the $L^2(\Omega)$ norm and $\|\cdot\| := |\nabla \cdot|$ is the energy norm; u and p are the discrete solutions on a sufficiently refined mesh.

Several examples show the convergence of the method. The numerical experiments have been developed with the finite element toolbox ALBERT [11], extended with new function basis including bubble functions.

2 Continuous Problem

Let Ω a bounded, polyhedral domain in $\mathbb{R}^d (d = 2, 3)$. $L^2(\Omega)$ the space of Lebesgue-measurable functions that are square-integrable. Set $\mathbb{W} := H^1(\Omega)$ the Sobolev space of functions in $L^2(\Omega)$ with weak derivatives of first order in $L^2(\Omega)$. $\mathbb{V} := H_0^1(\Omega)$ is the subspace of $\mathbb{W}$ whose functions vanish on boundary $\partial\Omega$. In the following, $(\cdot, \cdot)$ denotes the inner product in L^2 and $< \cdot, \cdot >$ the duality pairing between the corresponding spaces. Moreover, consider the bilinear and linear forms:

$$a(u, v) = (\nabla u, \nabla v), < Au, v >= a(u, v), \forall u, v \in \mathbb{V} \quad (1)$$

$$< F, v >= (f, v) \quad (2)$$

The rest of the data is given by a force density $f \in L^2(\Omega)$ and an obstacle $\psi \in \mathbb{W}$ with $\psi \leq 0$ on $\partial\Omega$. The obstacle ψ is associated with the nonempty, $\mathbb{V}$-closed, and convex set of admisible displacements:

$$\mathcal{K} := \{v \in \mathbb{V} | v \geq \psi \text{ a.e. in } \Omega\} \quad (3)$$

The continuous problem reads as follows:

Continuous Problem. *Given Ω, f, ψ as above, find $u \in \mathcal{K}$ such that the following variational inequality holds:*

$$a(u, v - u) \geq (f, v - u) \quad \forall v \in \mathcal{K} \quad (4)$$

It is well known that the above problem admits a unique solution u, see, e.g. [3]. The unilateral constraint $u \geq \psi$ yields a line singularity (free boundary) that is the internal boundary of the contact set:

$$\mathcal{C} := \{x \in \Omega : u(x) = \psi(x)\} \quad (5)$$

The free boundary location is a priori unknown and a prime computational objective.

3 Formulation with Multivalued Operator

Consider the functional on $\mathbb{V}$ that characterizes the convex $\mathcal{K}$:

$$I_{\mathcal{K}}(v) = \begin{cases} 0 & v \in \mathcal{K} \\ \infty & v \notin \mathcal{K} \end{cases} \tag{6}$$

The problem (4) is equivalent to: *Find* $u \in \mathbb{V}$ such that

$$Au + p = F \text{ in } \mathbb{V}' \tag{7}$$

$$p \in \partial I_{\mathcal{K}}(u) \text{ in } \mathbb{V}' \tag{8}$$

where $\partial I_{\mathcal{K}}$ is the subdiferential of $I_{\mathcal{K}}$, which is a multivalued operator, and $\mathbb{V}'$ is the dual space of $\mathbb{V}$. Set Π_λ the Yosida approximation of $\partial I_{\mathcal{K}}$. The solution is then characterized by the existence of $p \in \mathbb{V}$, such that the pair $(u, p) \in \mathbb{V} \times \mathbb{V}$ holds:

$$Au + p = F \text{ in } \mathbb{V}' \tag{9}$$

$$p = \Pi_\lambda(u + \lambda p) \text{ in } \mathbb{V}' \tag{10}$$

The operator $\Pi_\lambda = \frac{1}{\lambda}(Id - P_{\mathcal{K}})$, where $P_{\mathcal{K}}$ is the projection operator on $\mathcal{K}$. Moreover, Π_λ is a Lipschitz operator with constant $1/\lambda$

4 Discretization

Let $\mathcal{T}_h$ be a uniformly regular triangulation of Ω characterized by the diameter h. Let $\mathbb{V}_h \subset \mathbb{V}$ indicate the space of continuous piecewise linear finite element functions and extended with the *bubble* functions over $\mathcal{T}_h$, i.e., $\mathbb{V}_h := \{v_h \in \mathcal{C}^0(\bar{\Omega}) | \quad \forall T \in \mathcal{T}_h, \quad v_{h|T} \in P_1(T) + B(T)\}$, where $B(T)$ is generated by the product $\lambda_1\lambda_2\lambda_3$ of the barycentric coordinates. Let $\mathbb{M}_h \subset L^2(\Omega)$ be the space of piecewise constant finite element functions over $\mathcal{T}_h$.

We consider the operator $\Pi_{\lambda,h} = \frac{1}{\lambda}(Id - P_{\mathcal{K}_h})$, where $\mathcal{K}_h$ is a suitable approximation of $\mathcal{K}$, that is, $\mathcal{K}_h = \{q \in \mathbb{M}_h | \quad q_{|T} \geq \psi(\mathfrak{b}_T) \quad \forall T \in \mathcal{T}_h\}$, being $\mathfrak{b}_T$ the barycentre of T . The discrete problem reads as follows:

Find $(u_h, p_h) \in \mathbb{V}_h \times \mathbb{M}_h$ *such that*

$$Au_h + R^t p_h = F \tag{11}$$

$$p_h = \Pi_{\lambda,h}(Ru_h + \lambda p_h) \tag{12}$$

where R denotes the orthogonal projection operator in $L^2(\Omega)$ norm from $\mathbb{V}_h$ onto $\mathbb{M}_h$ and R^t is its transposed operator.

The Uzawa algorithm iterations are written:

For any ${p_h}^{(0)}$, *obtained* ${p_h}^{(i-1)}$

$$A{u_h}^{(i)} = F - R^t {p_h}^{(i-1)} \tag{13}$$

$${p_h}^{(i)} = \Pi_{\lambda,h}(R{u_h}^{(i)} + \lambda {p_h}^{(i-1)}) \tag{14}$$

It is well known that the above algorithm is convergent: for some $\lambda > 0$ we have

$$\lim_{i\to\infty} \|u_h{}^{(i)} - u_h\| = 0 \tag{15}$$

Remark 1. Notice that the above algorithm may be considered as a fixed point iteration for the application defined over M_h:

$$q \to \Pi_\lambda(\lambda(I - \frac{1}{\lambda} RA^{-1}R^t)q + RA^{-1}F) \tag{16}$$

Since $\mathbb{V}_h$ and $\mathbb{M}_h$ are finite dimensional spaces and $\mathbb{V}_h$ has bubble functions, the kernel of R^t is null. Thus, the application

$$\begin{aligned} \mathbb{M}_h &\to \mathbb{R} \\ q &\to |R^t q| \end{aligned}$$

defines a norm in M_h.We can now choose λ such that:

$$|I - \frac{1}{\lambda} RA^{-1}R^t| = 1 - \beta(h) < 1 \tag{17}$$

hence,

$$|p_h{}^i - p_h| \leq (1 - \beta(h))|p_h{}^{i-1} - p_h| \tag{18}$$

and

$$\lim_{h\to 0} \beta(h) = 0 \tag{19}$$

5 Adaptive Algorithm

In this section we describe the adaptive-modified Uzawa method. To simplify notations let us assume that $\mathcal{T}_j$ stands for the mesh obtained from $\mathcal{T}_{j-1}$ by refining and the corresponding sets of finite element functions are denoted by $(\mathbb{V}_j, \mathbb{M}_j)$ and $(\mathbb{V}_{j-1}, \mathbb{M}_{j-1})$). Consider a pair of successions:

$$\begin{aligned} \mathbb{V}_0 \subset \mathbb{V}_1 \subset \ldots \quad \subset \mathbb{V}_j \subset \mathbb{V}_{j+1} \subset \ldots \subset \mathbb{V}_J = \mathbb{V}_h \\ \mathbb{M}_0 \subset \mathbb{M}_1 \subset \ldots \subset \mathbb{M}_j \subset \mathbb{M}_{j+1} \subset \ldots \subset \mathbb{M}_J = \mathbb{M}_h \end{aligned}$$

For any $P_0 \in \mathbb{M}_0$, $\varepsilon_0 > 0$, $0 < \gamma < 1$. Given $P_{j-1} \in \mathbb{M}_{j-1}$, let $u_j \in V_J$ denote the solution of

$$a(u_j, v) = < F, v > -(P_{j-1}, v) \quad \forall v \in \mathbb{V}_J \tag{20}$$

An adaptive FEM method is applied to find $U_j \in V_j$ such that

$$\|U_j - u_j\| \leq C\varepsilon_j, \quad |(R_j - R_J)U_j| < C\varepsilon_j \tag{21}$$

where $\varepsilon_j < \gamma\varepsilon_{j-1}$. This procedure is denoted by

$$(\mathcal{T}_j, U_j) \leftarrow \mathsf{ELLIPTIC}(\mathcal{T}_{j-1}, U_{j-1}, \varepsilon_j, f) \tag{22}$$

We, finally, actualize the multiplier:

$$P_j = \Pi_{\lambda,j}(R_j U_j + \lambda P_{j-1}) \tag{23}$$

The following box describes the algorithm:

Adaptive Modified Uzawa Algorithm AMUA.

Choose parameters $\lambda > 0$, $0 < \gamma < 1, \varepsilon_0 > 0$; set $j = 1$.
1. Select any initial mesh $\mathcal{T}_0$ and any $P_0 \in \mathbb{M}_0$.
2. Update $\varepsilon_j \leftarrow \gamma \varepsilon_{j-1}$.
3. Compute $(\mathcal{T}_j, U_j) \leftarrow \mathsf{ELLIPTIC}(\mathcal{T}_{j-1}, U_{j-1}, P_{j-1}, \varepsilon_j, f)$.
4. Compute $P_j \leftarrow \mathsf{UPDATE}(\mathcal{T}_j, U_{j-1}, P_{j-1}, \lambda)$.
5. Update $j \leftarrow j + 1$.
6. Go to step 2.

With the hypothesis above, we have the following convergence theorem for the algorithm:

Theorem 1. *There exist positive constants C and $\delta < 1$ such that the iterative solutions (U_j, P_j) produced by the adaptive-modified Uzawa method satisfy:*

$$\|u_J - U_j\| + |p_J - P_j| \leq C\delta^j \tag{24}$$

where $(u_J, p_J) \in \mathbb{V}_J \times \mathbb{M}_J$ denote the solution of the problem (11)-(12).

Sketch of the proof in the case $\psi = 0$:

$$P_j = \Pi_{\lambda,j}(R_j U_j + \lambda P_{j-1}) = \tag{25}$$

$$= \Pi_{\lambda,j}(R_J u_j + \lambda P_{j-1} + R_J(U_j - u_j) + (R_j - R_J)U_j) \tag{26}$$

The solution u_j of (20) may be written as $u_j = A^{-1}(F - R_J^t P_{j-1})$. Observe that in the case $\psi = 0$ we have $\Pi_j(z) = \Pi_J(z)$ if $z \in \mathbb{M}_j$. Hence

$$P_j = \Pi_{\lambda,J}(\lambda(I - \frac{1}{\lambda} R_J A^{-1} R_J^t) P_{j-1} + R_J A^{-1} F + R_J(U_j - u_j) + (R_j - R_J)U_j) \tag{27}$$

If we write p_J, the solution of (12), as follows

$$p_J = \Pi_{\lambda,J}(\lambda(I - \frac{1}{\lambda} R_J A^{-1} R_J^t) p_J + R_J A^{-1} F) \tag{28}$$

Then, subtracting (27) and (28), applying norms, we find an upper bound, for different constants C

$$|p_J - P_j| \leq (1 - \beta(h))|p_J - P_{j-1}| + \frac{C}{\lambda}\|U_j - u_j\| + \frac{C}{\lambda}|(R_j - R_J)U_j|$$

$$|p_J - P_j| \leq (1 - \beta(h))|p_J - P_{j-1}| + \frac{C}{\lambda}\varepsilon_0 \gamma^j$$

As in [9], by induction arguments we obtain

$$|p_J - P_j| \leq |p_J - P_0|\eta^j + \frac{1}{\lambda}\varepsilon_0 j\eta^j \leq C\delta^j \tag{29}$$

where $\eta = max\{(1-\beta(h)), \gamma\}$ and $\eta < \delta < 1$.

To find an error bound for $\|u_J - U_j\|$, observe that $\|u_J - u_j\| \leq C|p_J - P_j|$. Hence,

$$\|u_J - U_j\| \leq \|u_J - u_j\| + \|u_j - U_j\| \leq C(|p_J - P_j| + \varepsilon_j) \tag{30}$$

which proves the result.

For the case $\psi \neq 0$ we need to add $|\psi - \psi_j| < C\varepsilon_j$ to the condition (21) where ψ_j is the interpolated function in $\mathbb{M}_j$.

6 Numerical Experiment

Consider $\Omega = \{x \in \mathbb{R}^2 : \|x\| < 1\}$, $f(x) = 0$, and $\psi(x) = max\{0, 1 - 4\|x\|\}, \forall x \in \Omega$. The solution of the problem $u(x) = 1 - \|x\|$, for all $x \in \Omega$ and $\mathcal{C}(u) = \{0\}$

Let us assume an initial triangulation $\mathcal{T}_0$ of Ω, and the posteriori error estimate

$$\|u - u_h\| \leq \eta(u_h) = (\sum_{T \in \mathcal{T}_0} \eta_T(u_h)^2)^{1/2} \tag{31}$$

where(see [10] for H^1 error estimate for the linear problem $-\Delta u = f(x)$)

$$\eta_T(u_h)^2 = C_0^2 h_T^2 \| - \Delta u_h + f(\cdot)\|^2_{L^2(T)} + \\ + C_1^2 \sum_{\partial T \cap \Omega} h_T \|[\![\nu \cdot \nabla u_h]\!]\|^2_{L^2(\Gamma)}$$

If *tol* is a given allowed tolerance for the error, and $\eta(u_h) > tol$, we refine the mesh while $\eta > tol$. For the Maximum strategy (see [11]), a threshold $\gamma \in (0,1)$ is given, and all elements $T \in \mathcal{T}_j$ with

$$\eta_T > \gamma \max_{T' \in \mathcal{T}} \eta_{T'}$$

are marked for refinement.

Set $\gamma = 0.8$, $\varepsilon_0 = tol = 1.0$ and the Yosida parameter $\lambda = 1.0$. Figure 1 shows the behaviour of the true error in H^1-norm as a function of the number of u_h-degrees of freedom (DOF). We observe the improvement in applying our adaptive method (in solid linestyle) compared with the results obtained with uniform refinement (in dashed linestyle). Figure 2 shows mesh in the final step and the solution isolines.

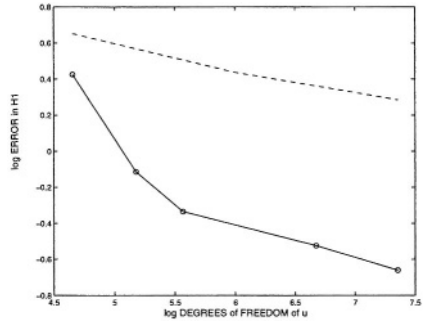


Fig. 1. log-log error and DOF for $\varepsilon_0 = 1.0$

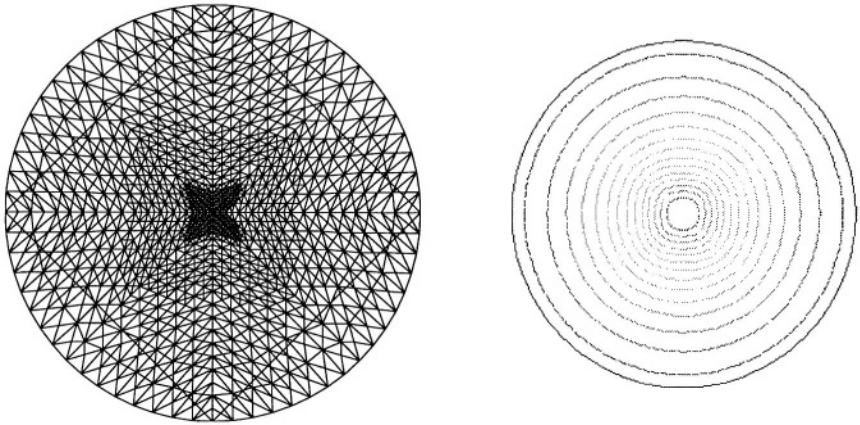


Fig. 2. Mesh and solution isolines

7 Conclusions

We have developed an adaptive Uzawa algorithm to solve the obstacle problem which is a modification of the classical Uzawa method. We justify the use of a a-posteriori error estimation from the linear elliptic problems for this kind of non-linear problems. The numerical results asserts the validity of the theoretical analysis and the efficiency of the algorithm. A better improvement should be obtained with a finest control of the interpolation error of the obstacle function ψ. This will be done in a future research.

Acknowledgements. Research partially supported by REN2001-0925-03-03, Ministerio de Ciencia y Tecnología (Spain) and SA089/01, Junta de Castilla y León (Spain).

References

1. A. Friedman, Variational Principles and Free-Boundary Problems, Pure Appl. Math.,John Wiley, New York, 1982.
2. J. Haslinger, I. Hlaváček and J. Nečas, Numerical Methods for unilateral problems in solid mechanics, Handbook of Numerical Analysis. Vol. IV, P. G. Ciarlet and J. L. Lions, eds., North-Holland, Amsterdam, 1996, pp.313-485.
3. P. G. Ciarlet, Basic Error Estimates for Elliptic Problems, Handbook of Numerical Analysis. Vol II, P. G. Ciarlet and J. L. Lions, eds., North-Holland, Amsterdam, 1991,pp.24-25.
4. G. Duvaut and J.L. Lions, Inequalities in Mechanics and Physics, Grundlehren Mathematischen Wiss, Springer-Verlag, Berlin, Heidelberg, New York, 1976.
5. N. Kikuchi and J. T. Oden, Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods, SIAM Stud. Appl.Math. 8,SIAM, Philadelphia, 1988.
6. D. Kinderlehrer and G. Stampacchia,An Introduction to Variational Inequalities and Their Applications, Pure Appl. Math. 88, Academic Press, New York, 1980.
7. J.F. Rodrigues, Obstacle Problems in Mathematical Physics, North-Holland Math. Stud. 134, North-Holland, Amsterdam, 1987.
8. J. L. Lions, Quelques méthodes de résolution de problèmes aux limites non linéaires, Dunod, Paris, 1969.
9. E. Bänsch, P.Morin and R.H. Nochetto, An adaptive Uzawa fem for the Stokes problem: Convergence without the inf-sup condition, SIAM J. Numer. Anal., 40 (2002), 1207-1229.
10. E. Bänsch, Local mesh refinement in 2 and 3 dimensions, Impact Comput.Sci.Engrg.,3(1991), 181-191.
11. A. Schmidt and K.G. Siebert, ALBERT: An adaptive hierarchical finite element toolbox, Preprint 06/2000, Freiburg (2000).

Finite Element Model of Fracture Formation on Growing Surfaces

Pavol Federl and Przemyslaw Prusinkiewicz

University of Calgary, Alberta, Canada

Abstract. We present a model of fracture formation on surfaces of bi-layered materials. The model makes it possible to synthesize patterns of fractures induced by growth or shrinkage of one layer with respect to another. We use the finite element methods (FEM) to obtain numerical solutions. This paper improves the standard FEM with techniques needed to efficiently capture growth and fractures.

1 Introduction and Background

We consider fracture pattern formation on differentially growing, bi-layered surfaces. The top layer, called the *material layer,* is assumed to grow slower than the bottom *background layer.* Through the attachment of the material layer to the background layer, such differential growth produces increasing stresses in the material layer. Eventually, the stresses exceed the material's *threshold stress,* which leads to formation of a fracture. As this process continues, a pattern of fractures develops. Here we present a method for simulating this pattern formation.

In our method, fracture mechanics [1] is combined with the framework of the finite element method (FEM) to form computer simulations that can predict whether and how a material will fail. The FEM is a numerical technique for solving partial differential equations [10], widely used in mechanical engineering to analyze stresses in materials under load [10]. Given some initial configuration of a structure, coupled with boundary conditions and a set of external forces, the FEM determines the shape of the deformed structure. The deformed shape represents the *equilibrium state,* since the sum of internal and external forces at any point in the structure is zero. Our method is most closely related to that of O'Brien and Hodgins [3], in that it treats fracture formation in the context of continuum mechanics and the finite element method. In contrast to their work, however, we are interested in patterns of fractures, rather than the breaking of brittle materials.

We consider formation of crack patterns in bark as an example of pattern formation due to *expansion* of one material layer with respect to another, and formation of crack patterns in mud as an example of pattern formation due to *shrinking* of one layer with respect to another. Tree bark consists of dead conductive tissue, phloem, which is expanded by the radial growth of cambium inside the trunk [6]. As a result of this expansion, the bark stretches until it reaches its limit of deformation and cracks. In our simulation we use a simplified,

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 138–145, 2004.
© Springer-Verlag Berlin Heidelberg 2004

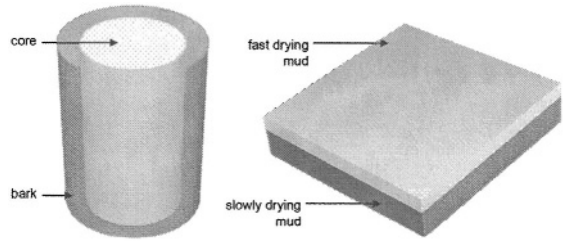


Fig. 1. The two-layered models of bark and drying mud.

two-layer model of a growing tree trunk (Fig. 1, left). The inside core grows radially and does not break, while the outer layer, which represents the bark, may break.

As water evaporates from mud, the mud shrinks. Since water evaporates faster from the layers closest to the surface, different layers shrink at different rates with respect to each other. This non-uniform shrinkage of the various layers leads to material stress and, consequently, to the formation of cracks. We model drying mud using two layers (Fig. 1, right). The background layer is assumed to be static, representing either mud that dries very slowly or the surface on which the drying mud rests. The material layer represents the drying mud and is attached to the background layer.

We use linear elastic fracture mechanics [1], and approximate the stress field near a crack tip using the theory of linear elasticity [8]. A fracture occurs where the *maximum principal stress* exceeds material's *threshold stress* (*maximum principal stress criterion* [8]). The direction of the newly formed fracture is perpendicular to the direction of this maximum principal stress. We also use the maximum principal stress criterion to establish the propagation direction of an existing fracture. We terminate the propagation of a fracture using the Griffith energy approach [1]. It states that a fracture propagates as long as the potential energy released by the fracture exceeds the energy required to form the fracture. An overview of our algorithm is given in Fig. 2.

2 Fracture Simulation Algorithm

Discretization. We model the material layer as a single layer of three dimensional 6-node wedge elements (prisms) (Fig. 3) [2]. The material layer in which the cracks are formed is attached to the background layer at *attachment points,* which are the bottom three nodes of each wedge element. The attachment points are randomly placed on the plane or a cylindrical surface, then repelled using a particle repelling algorithm [7] to obtain more uniform distribution. The resulting points are connected into a mesh using Delaunay triangulation.

Growth modeling. The growth of the background layer is modeled by adjusting the positions at which the wedge elements are attached to it. The trajectory of each attachment point is defined by its initial position and its velocity vector.

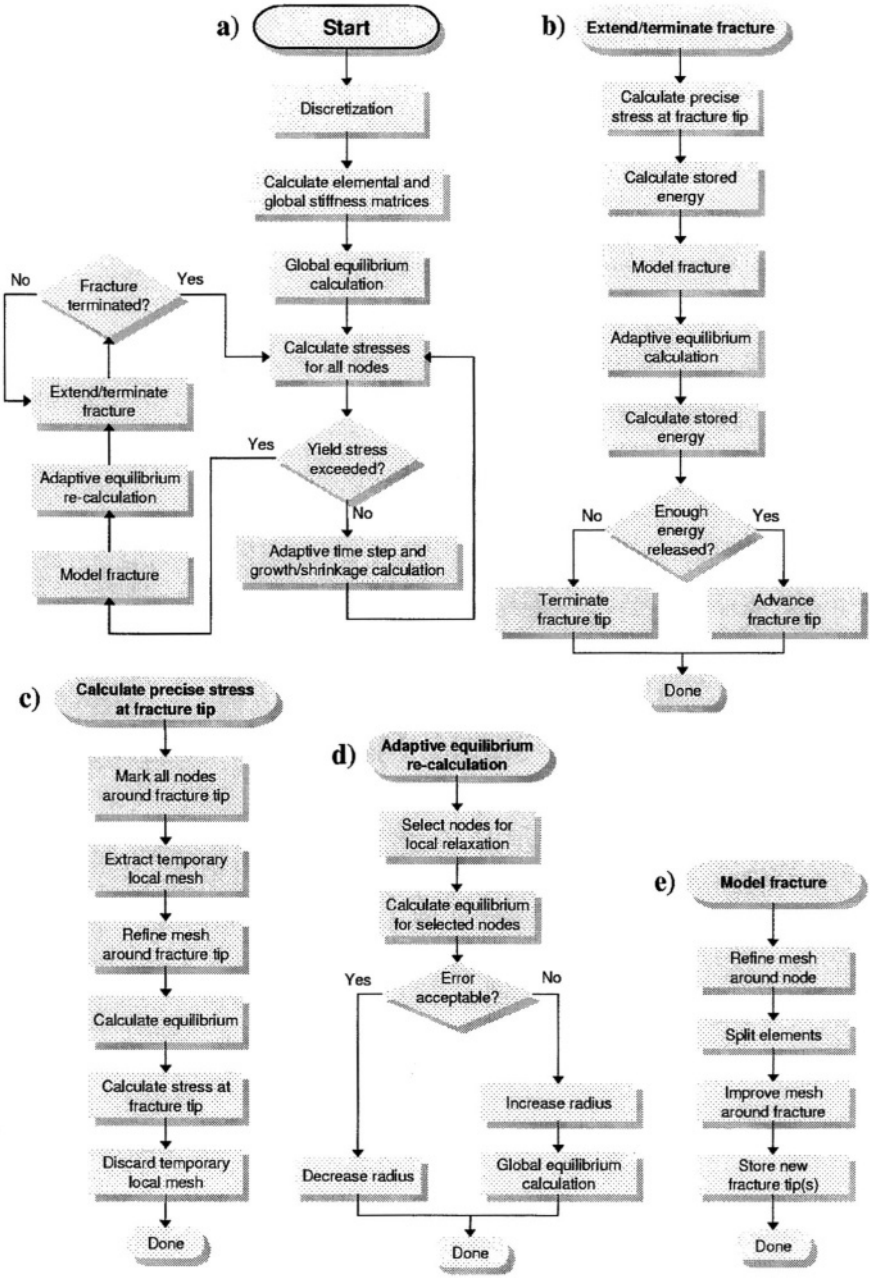


Fig. 2. Structure of our fracture simulation algorithm.

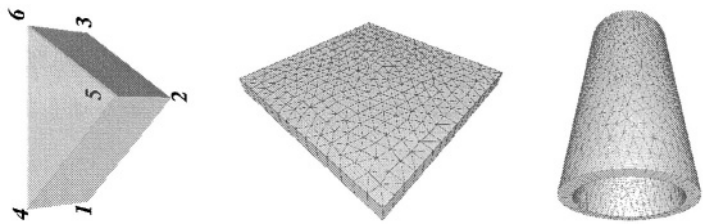


Fig. 3. The wedge element and the resulting representations of flat and cylindrical surfaces.

We consider both *isotropic* and *anisotropic* growth [6]. Shrinkage of the material layer is simulated by adjusting the reference shapes of the wedge elements.

Global stiffness matrix calculation. We calculate the equilibrium state of the mesh using the finite element method [10]. First, we calculate the elemental stiffness matrices K_e using 9-point Gaussian quadrature for each prism element, as described by Keeve et al. [2]. Next, we assemble the elemental stiffness matrices into the global stiffness matrix K_g, which represents the coefficients of a set of linear equations $K_g Q = F$. Here Q is the vector of nodal displacements and F is the vector of nodal forces.

Equilibrium calculation. At equilibrium, the total force acting on any free node is equal to zero. The calculation of the equilibrium is therefore performed by setting $F = 0$, imposing boundary conditions, and solving the resulting system of equations for Q. In our case, the boundary conditions consist of the known nodal displacements values of the fixed nodes, determined from the positions of the attachment points. We solve the resulting system of equations using the iterative conjugate gradient algorithm [4].

When a change is made to the geometry of the model, the equilibrium state of the model needs to be recalculated. Many of these changes, such as in fracture formation, mesh refinement, or node repositioning during mesh smoothing, are confined to small regions, and have negligible effect on more distant parts of the mesh. We take advantage of this locality by recalculating the equilibrium state adaptively, only in the regions of interest (*local relaxation*). These regions are detected by checking for large unbalanced nodal forces.

Modeling fracture behavior. Once the equilibrium state of the material layer is calculated, we compute the stress tensor at each node [11], and we use it to calculate the maximum principal stress s_1. If s_1 exceeds the threshold stress of the material, we mark the corresponding node n as a possible candidate for fracture initiation. In most cases there is only a single candidate. Having more than one candidate typically means that a too large a time step was used for simulating growth. We address this issue by advancing the simulation time with adaptive time step control (Fig. 2a).

Once a single fracture candidate node n has been identified, we extend the fracture at this node and adjust the finite element mesh accordingly. We use the

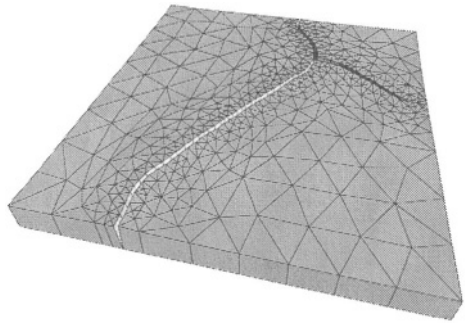


Fig. 4. The mesh is refined only around the fracture.

same procedure to both incorporate the onset of a new fracture and to propagate an existing fracture (Fig. 2e). The input to this procedure is the location of the fracture, specified by a fracture node n and the corresponding nodal stress tensor σ. The fracture plane p is determined from σ: its normal is the eigenvector of σ corresponding to the maximum principal stress s_1.

Modeling fracture extension. The first step consists of refining the elements sharing the fracture node n so that each element is smaller than a user-defined constant λ_{max}. The constant λ_{max} effectively denotes the maximum distance a fracture can extend before the nodal stress at its fracture tip is recalculated. Imposing the limit on the length of the fracture extension is important when the fractures turn rapidly.

We refine the elements with a version of the triangular mesh dynamic refinement algorithm proposed by Rivara and Inostroza [5]. This refinement step allows us to discretize the surface using a coarse global mesh, and subdivide it only where needed, leading to smaller memory requirements and faster simulations. An example of a mesh that has been dynamically refined around a fracture is shown in Fig. 4.

The next step is to create a new copy n' of the fracture node n. All elements that contain node n are then adjusted according to their locations with respect to the fracture plane p. The elements situated entirely on one side of the plane are assigned the original node n, while the elements on the other side are assigned the new copy n'. The remaining elements, sharing the node n, are split by the fracture plane. If a T-junction is formed by this process, the adjacent element is also subdivided to remove it.

When the fracture plane intersects an element close to one of its edges, a degenerate wedge may be formed as a result of splitting. The solution proposed by O'Brien and Hodgins [3] is not to allow degenerate elements to be created; this is accomplished by rotating the fracture plane by a small amount to align it with an edge in the mesh. This approach suffers from fracture directions being occasionally influenced by the geometry of the surface subdivision. We adopted a reverse approach: instead of snapping the fracture plane to a near parallel edge, we snap the edge to the fracture plane, as illustrated in Fig.5.

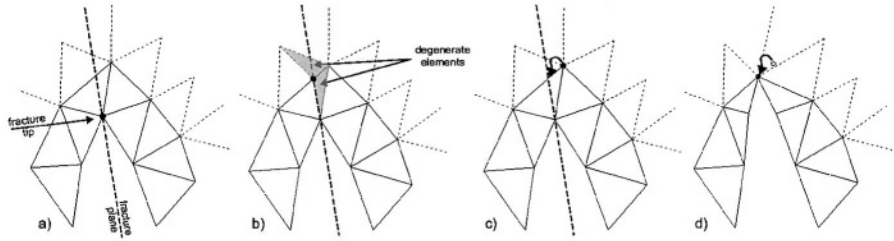


Fig. 5. Example of snapping a node to a fracture plane. a) The node and fracture plane are identified, b) simple node insertion can lead to degenerate elements, c) our approach is to snap the node to the fracture plane, d) the resulting mesh does not contain degenerate elements.

The accuracy of the nodal stress calculation depends highly on the shapes of the elements [10]. The closer the top faces of elements are to equilateral triangles, the more precise are the stress calculations. Unfortunately, even though the edge-snapping technique prevents formation of degenerate elements, the introduction of a fracture into the mesh can produce elements of sub-optimal shapes. To further improve the mesh around a fracture after it has been extended, we employ the angle smoothing algorithm developed by Zhou and Shimada [9]. Since a global application of mesh smoothing would require re-computation of all elemental stiffness matrices, we only apply the smoothing to the mesh nodes around the fracture tips.

Local multi-resolution calculation of nodal stress at crack tips. The elements around a fracture tip must be very small in order to calculate the stress at the fracture tip correctly. On the other hand, once the nodal stress has been evaluated, the need for such small elements disappears. To reconcile these requirements, we evaluate nodal stresses at fracture tips using a local multi-resolution method (Fig 2c). First, we extract a sub-model from the original model, consisting of the mesh in the neighborhood of the fracture tip. This sub-model is then refined around the fracture tip to a user-controlled level of detail with the algorithm of Rivara and Inostroza [5]. The equilibrium state of the refined mesh is calculated next; this is followed by the computation of the nodal stress at the fracture tip. The refined sub-model is then discarded. The end-result is the original mesh and a more accurate approximation of the stress at the fracture tip. This process is illustrated in Fig. 6.

3 Results and Discussion

Sample bark and mud patterns synthesized using the presented method are shown in illustrated in Figs. 7 and 8. The different patterns were obtained by varying simulation parameters, including the thickness of the material layer, rate of growth and shrinkage, Young's modulus, threshold stress of the material, fracture toughness, etc. The average size of the models used to generate these

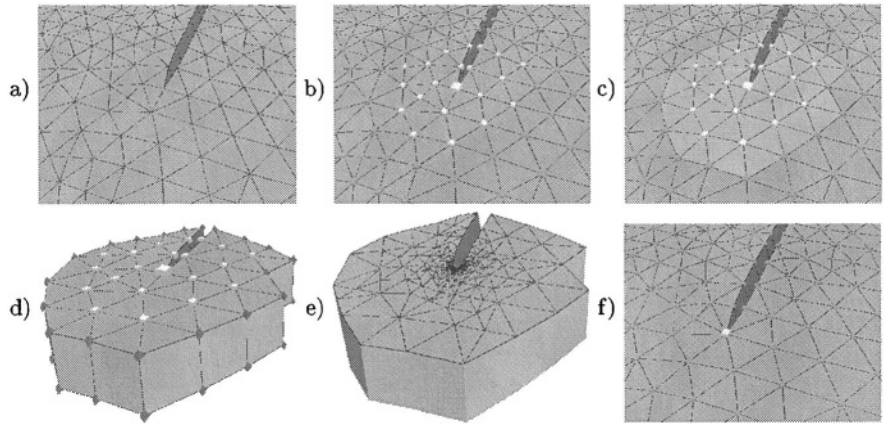


Fig. 6. Illustration of the local multi-resolution calculation of stresses and fracture propagation. a) View of a fracture before it is extended. b) Nodes close to the fracture tip are identified. c) All elements sharing the selected nodes are identified. d) A sub-mesh with the selected elements is created. The nodes on its boundary are treated as fixed. e) This mesh is refined and the stress at the fracture tip is computed with increased precision. f) The sub-model is discarded and the calculated stress at the fracture tip is used to extend the fracture.

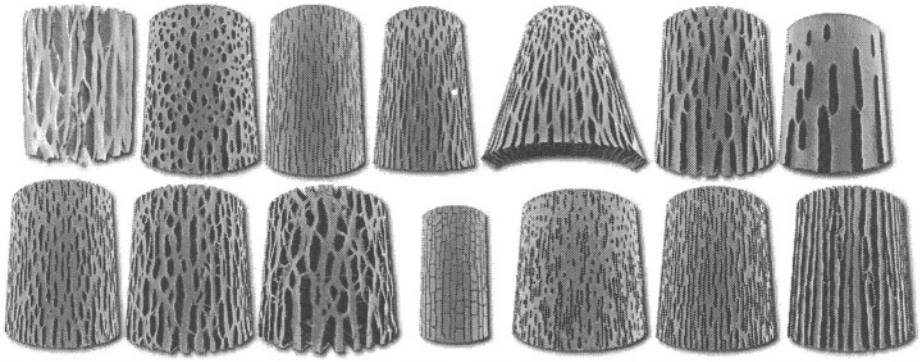


Fig. 7. A variety of bark-like patterns generated by the proposed method.

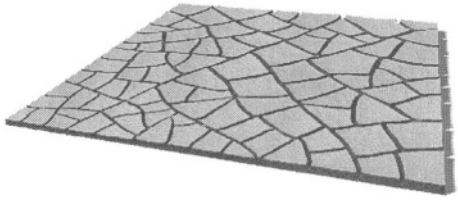


Fig. 8. Generated fracture pattern in dried mud.

patterns was between 60 and 150 thousand elements. The running times were of the order of few hours on a 1.4GHz Pentium IV computer.

We found that the largest performance improvement was achieved due to the dynamic subdivision of elements around the fractures. The local equilibrium (relaxation) calculation algorithm also improves the simulation efficiency. For example, the mud pattern in Fig. 8 was generated in approximately two hours using the local relaxation algorithm. The same pattern took almost eight hours to synthesize when the local relaxation was turned off. This large improvement in the simulation time is due to the fact that fractures reduce the global effects of localized changes.

In conclusion, this paper shows that the finite element method is a viable tool for modeling not only individual fractures, but also fracture patterns. The acceleration techniques presented in this paper, taken together, decrease the computation time an order of magnitude, compared to the non-accelerated method.

Acknowledgments. We thank Brendan Lane and Colin Smith for editorial help. The support of the Nationial Science and Engineering Research Council is gratefully acknowledged.

References

1. Anderson T. L. Fracture Mechanics: Fundamentals and Applications. CRC Press, Boca Raton, second edition, 1995.
2. Keeve E., Girod S., Pfeifle P, Girod B. Anatomy-Based Facial Tissue Modeling Using the Finite Element Method. Proceedings of Visualization'96, 1996.
3. O'Brien J. F., Hodgins J. K. Graphical Modeling and Animation of Brittle Fracture. Proceedings of ACM SIGGRAPH'99, 1999.
4. Press W. H., Teukolsky S. A., Wetterling W. T., Flannery B. P. Numerical recipes in C: the art of scientific computing. Second edition. Cambridge University Press.
5. Rivara M. and Inostroza P. Using Longest-side Bisection Techniques for the Automatic Refinement of Delaunay Triangulations. The 4th International Meshing Roundtable, Sandia National Laboratories, pp.335-346, October 1995.
6. Romberger J. A., Hejnowicz Z. and Hill J. F. Plant Structure: Function and Development. Springer-Verlag,1993.
7. Witkin A. P. and Heckbert P. A. Using particles to sample and control implicit surfaces. SIGGRAPH'94, pp. 269-277, July 1994.
8. Zhang L. C. Solid Mechanics for Engineers, Palgrave, 2001.
9. Zhou T. and Shimada K. An Angle-Based Approach to Two-Dimensional Mesh Smoothing. The 9th International Meshing Roundtable, pp.373-84, 2000.
10. Zienkiewicz O. C. and Taylor R. L. Finite element method: Volume 2 - Solid Mechanics. Butterworth Heinemann, London, 2000.
11. Zienkiewicz O. C. and Zhu J. Z. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. International Journal for Numerical Methods in Engineering, 33:1331-1364, 1992.

An Adaptive, 3-Dimensional, Hexahedral Finite Element Implementation for Distributed Memory

Judith Hippold[1]*, Arnd Meyer[2], and Gudula Rünger[1]

[1] Chemnitz University of Technology, Department of Computer Science,
09107 Chemnitz, Germany
{juh,ruenger}@informatik.tu-chemnitz.de
[2] Chemnitz University of Technology, Department of Mathematics
09107 Chemnitz, Germany
a.meyer@mathematik.tu-chemnitz.de

Abstract. Finite elements are an effective method to solve partial differential equations. However, the high computation time and memory needs, especially for 3-dimensional finite elements, restrict the usage of sequential realizations and require efficient parallel algorithms and implementations to compute real-life problems in reasonable time. Adaptivity together with parallelism can reduce execution time significantly, however may introduce additional difficulties like hanging nodes and refinement level hierarchies. This paper presents a parallel adaptive, 3-dimensional, hexahedral finite element method on distributed memory machines. It reduces communication and encapsulates communication details like actual data exchange and communication optimizations by a modular structure.

1 Introduction

Finite element methods (FEM) are popular numerical solution methods to solve partial differential equations. The fundamentals are a discretization of the physical domain into a mesh of finite elements and the approximation of the unknown solution function by a set of shape functions on those elements. The numerical simulation of real-life problems with finite elements has high computation time and high memory needs. Adaptive mesh refinement has been developed to provide solutions in reasonable time. However, there is still need for parallel implementations, especially for 3-dimensional problems as considered in this paper.

The basis of an efficient parallel implementation is a sophisticated algorithmic design offering a trade-off between minimized data exchange, computation overhead due to parallel realization, and memory needs. The actual parallel implementation furthermore requires optimized communication mechanisms to

* Supported by DFG, SFB393 *Numerical Simulation on Massively Parallel Computers*

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 146–154, 2004.
© Springer-Verlag Berlin Heidelberg 2004

achieve good performance. The main problems to address for adaptive, hexahedral FEM are irregular structured meshes and hanging nodes: An adaptively refined mesh spread out across the address spaces of several processes requires to keep information about the different refinement levels of neighboring volumes owned by different processes. Furthermore, hanging nodes caused by hexahedral finite elements require several projections during the solution process. Both characteristics lead to high communication needs with irregular behavior. The parallelization approach presented in this paper reduces the number of sent messages by a special numerical design for solving the system of equation which we adopt from [1], [2], and [3] and by a specific communication mechanism. The advantages of the proposed parallel realization are: (a) a reduced number of messages due to the separation of communication and computation and duplicated data storage and (b) the possibility for internal optimizations without modifying the original FEM implementation which is reached by a modular structure. An interface for using the communication mechanism is provided.

The paper is organized as follows: Section 2 gives a brief overview of the FEM implementation. The parallel numerical and implementation approaches are introduced in Section 3. Section 4 presents our parallel realization in detail. Experimental results are given in Section 5 and Section 6 concludes.

2 Adaptive, 3-Dimensional Finite Element Method

The software package SPC-PM3AdH [4] implements the adaptive, 3-dimensional finite element method with hexahedral elements and solves 2nd order elliptic partial differential problems like the Poisson equation (1) or the Lamé system of linear elasticity (2).

$$Lu := -\nabla \cdot (A(x)\nabla u) + cu = f \ in \ \Omega \subset I\!R^3, \qquad A(x) = diag(a_i)_{i=1}^3 \tag{1}$$

$$u = u_0 \quad on \ \partial\Omega_1, \qquad n^t A(x)\nabla u = g \quad on \ \partial\Omega_2$$

$$-\mu\Delta\underline{u} - (\lambda+\mu)\, grad\, div\, \underline{u} = \underline{f} \quad in \ \Omega \subset I\!R^3, \qquad \underline{u} = (u^{(1)}, u^{(2)}, u^{(3)})^t \tag{2}$$

$$u^{(i)} = u_0^{(i)} \quad on \ \partial\Omega_1^{(i)}, \qquad t^{(i)} = g^{(i)} \quad on \ \partial\Omega_2^{(i)}, \quad i = 1,2,3$$

The program uses h-version finite element analysis where refinement of the elements is done according to the estimated error per hexahedron. Finite elements with linear, quadratic, and tri-quadratic shape functions are realized. The finite element method implemented by SPC-PM3AdH is composed of 5 phases:

Phase I: The first phase creates the initial mesh from an input file. A mesh consists of a hierarchy of structures. The most coarse-grained structure is the *volume* which represents a hexahedral finite element. Volumes are geometrically formed by 6 *faces* and each face is composed of 4 *edges*. Edges connect two vertices and a mid-node. *Nodes* are the most fine-grained data structure. They store information about coordinates and the solution vector. To keep track of the development of the adaptively refined mesh there is an additional hierarchy implemented for faces and edges to express the *parent-child* relation.

Phase II: Volumes are subdivided into 8 children according to the estimated error and geometrical conditions. Adaptive refinement may lead to different subdivision levels. The difference of those levels for neighboring volumes is restricted to one which causes additional iterative refinement.
Phase III: To facilitate a parallel implementation the global stiffness matrix is subdivided and an element stiffness matrix is assigned to each volume. The element stiffness matrices are assembled for newly created volumes by the third phase of the program.
Phase IV: The system of equations is solved with the preconditioned conjugate gradient method (PCGM). For preconditioning a Jacobi, an Yserentant [5], or a BPX [6] preconditioner can be selected.
Phase V: In the last phase the error is estimated with a residual based error estimator [7]. If the error for a volume deviates within a predefined threshold value from the maximum error, it is labeled for refinement.

3 Parallelization Approach

The parallelization approach assigns finite elements to processes. Thus the corresponding data for each volume representing a finite element are distributed among the address spaces of the different processes. For the parallel realization three main problems have to be solved: the management of shared data structures, the minimization of communication needs, and the consistency of different refinement levels.

3.1 Shared Data Structures

Neighboring volumes share faces, edges, and nodes. If two neighboring volumes are situated in different address spaces, the shared data structures are duplicated and exist within the memory of each owner process, which allows fast computation with minimal communication. Vector entries for duplicated nodes exist several times (see Figure 1) and contain only subtotals which have to be accumulated to yield the total result. [8] presents an approach distributing the nodes exclusively over the address spaces.

Computations on duplicated data structures require the unique identification of the different duplicates. For that reason we introduce the tuple *Tup(Identifier, Process). Identifier* denotes a local data structure of type face, edge, or node and *Process* denotes the number of the process that owns the duplicate. The tuple *Tup* is used to implement *coherence lists.* Each duplicated data structure is tagged with a coherence list which contains the identification tuples of all existing duplicates of that structure.

3.2 Minimization of Communication Needs

Numerical approach: The discretization of Formulas (1) or (2) with nodal shape functions yields the linear system $V\underline{u} = \underline{b}$ where the global stiffness

matrix V and the global right-hand-side vector $\underline{b}$ contain problem describing data and the solution vector $\underline{u}$ has to be calculated. Each process owns only element stiffness matrices V_s and element right-hand-side vectors $\underline{b}_s$ of its volumes, parts of the solution vector, and parts of the main diagonal which is necessary for applying the preconditioners. As introduced in Subsection 3.1 data shared by different processes require global accumulation of partial results. To keep the communication overhead low especially during solving the system of equation we distinguish between data requiring accumulation, e. g. the main diagonal and the solution vector, and data which can be used for independent calculations performed by the distinct processes as V_s and $\underline{b}_s$ (see also [2]). Our parallel preconditioned conjugate gradient algorithm working mainly on unaccumulated data and therefore reducing communication is described in the following:

Start:

Produce $\underline{u}_s$ from $\underline{u}$,
$\underline{r}_s := V_s\underline{u}_s - \underline{b}_s$,
$\gamma_{old} := 0.0$

Iterate until convergence:

(1) $\underline{w} := C^{-1}\underline{r}$

(2) $\gamma_s := \underline{r}_s^T\underline{w}_s$, $\gamma := \sum_{s=1}^{p} \gamma_s$, $if\ \gamma_{old} > 0 : \beta := \gamma/\gamma_{old}$, $\gamma_{old} := \gamma$

(3) $\underline{s}_s := \underline{w}_s + \beta\underline{s}_s$

(4) $\underline{w}_s := V_s\underline{s}_s$

(5) $\alpha_s := \underline{w}_s^T\underline{s}_s \qquad \alpha := \sum_{s=1}^{p} \alpha_s, \quad \alpha := -\gamma/\alpha$

(6) $\underline{u}_s := \underline{u}_s + \alpha\underline{s}_s$

(7) $\underline{r}_s := \underline{r}_s + \alpha\underline{w}_s$

Communication mechanism: Due to the special algorithmic design the exchange of data within a computational phase can be delayed and performed at the end of that phase, thus separating computation from communication. The resulting collect&get communication mechanism is the following: During computation each process collects information about necessary data exchanges with different *collect* functions which are adapted to the algorithmic needs. Such a function examines the coherence list for a given local data structure and in case

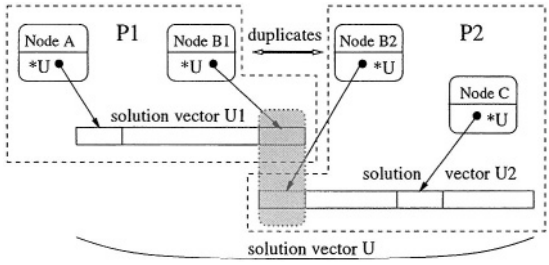


Fig. 1. Solution vector $\underline{u}$ spread over the address spaces of processes P1 and P2. Entries for the node B shared by P1 and P2 are duplicated and contain only subtotals after a computational phase.

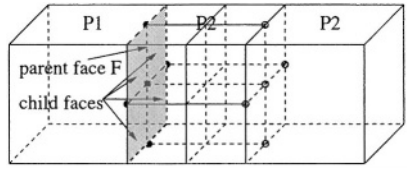


Fig. 2. Illustration of hanging nodes. Projections for grey-shaded hanging nodes access local data in the address space of process P2. Black hanging nodes require duplicated storage of face and edge parent-child hierarchies.

of duplicates it stores the remote identifiers and additional values in a send buffer for later exchange. After the computations the gathered values are sent to the corresponding processes extracted from the coherence lists. This data exchange is initialized by the first call of a *get* function. Further calls return an identifier of a local data structure and the received values for this structure from the receive buffer in order to perform specific actions.

3.3 Consistence of Refinement Levels

Adaptivity causes irregularly structured meshes with different refinement levels for neighboring volumes. Thus *hanging nodes* arise for hexahedral volumes (see Figure 2). Hanging nodes need several projections during the solution process which requires accesses to the parent-child hierarchy of the corresponding faces and edges. If the parent and child data structures are situated in different address spaces, as illustrated in Figure 2 for parent face *F,* the projections either require explicit communication for loading the remote data or the duplicated storage of face and edge hierarchies. Our parallelization approach stores the face and edge hierarchies in the address space of each owner process because this reduces communication and improves performance. For this reason the explicit refinement of duplicated faces and edges within the refinement phase and the creation of coherence lists for these data structures is necessary to keep data consistent. (see Section 4, Phase II).

4 Parallel Implementation

This section describes the parallel realization with regard to the necessary data exchanges within the different algorithmic phases.

Phase I – Creation of the Initial Mesh. In the first phase the initial mesh is read from an input file. The distribution of data structures is done according to a computed initial partitioning. First the entire mesh exists on each cluster node in order to reduce the communication effort necessary to create the coherence lists. The functions collect_dis and get_dis are provided to determine the duplicated structures and their remote identifiers and owner processes.

Phase II – Iterative Mesh Refinement. The parallel execution of the iterative refinement process requires the remote subdivision of duplicated faces and edges in order to keep data structures und coherence lists consistent. For that reason the refinement process is split into 2 steps: The first step iteratively subdivides local volumes and investigates them for duplicated faces and edges. For these faces and edges the identifiers of the children and the identifiers of the connected, newly created edges and nodes are collected with the function collect.ref. The remotely subdivided faces and edges are received using the function get_ref. In the second step the local refinement of those faces and edges and the creation of coherence lists is done. To update the coherence lists at the process initiating the remote refinement the collection and exchange of identifiers is necessary again. Refinement is performed until no further subdivision of volumes is done by any process. A synchronization step ensures convergence.

Projections of hanging nodes during the solution process require to access the corresponding faces and edges. Parallel execution needs explicit communication because processes do not have information about the current refinement levels of neighboring volumes. We reduce the number of sent messages by extracting the necessary information for faces during remote refinement and by using our collect&get communication mechanism for edges.

Phase III – Assembling the Element Stiffness Matrices. The entire main diagonal and the global right-hand-side vector are re-computed after assembling the element stiffness matrices for the new volumes. For the main diagonal, containing accumulated values, the global summation of subtotals for duplicated nodes is necessary and is supported by the functions collect_val and get_val.

Phase IV – Solving the System of Equation. Figure 3 outlines the conjugate gradient method for solving the system of equation in parallel. There are 3 communication situations to distinguish: calculation of scalar products, accumulation of subtotals, and projections for hanging nodes.

To determine the global scalar product each process computes subtotals which have to be accumulated. Duplicated nodes do not require special consideration because computation is done on unaccumulated vectors only. To create a uniform start vector $\underline{u}$ and to provide a uniform residual vector $\underline{r}$ for the preconditioner, partial results for duplicated nodes have to be accumulated using collect_val and get_val. Hanging nodes require several projections. If there are modifications of values for duplicated nodes, communication can be necessary to send the results to the other owner processes. To perform this the functions collect_own and get_own are provided.

Phase V – Error Estimation. The parallel error estimator determines the global maximum error by investigating the set of volumes with the maximum local error. To determine the error per volume calculations for the faces of the volumes are necessary. If a face is shared between two volumes, the overall result for this face is composed of the partial results computed by the different owners.

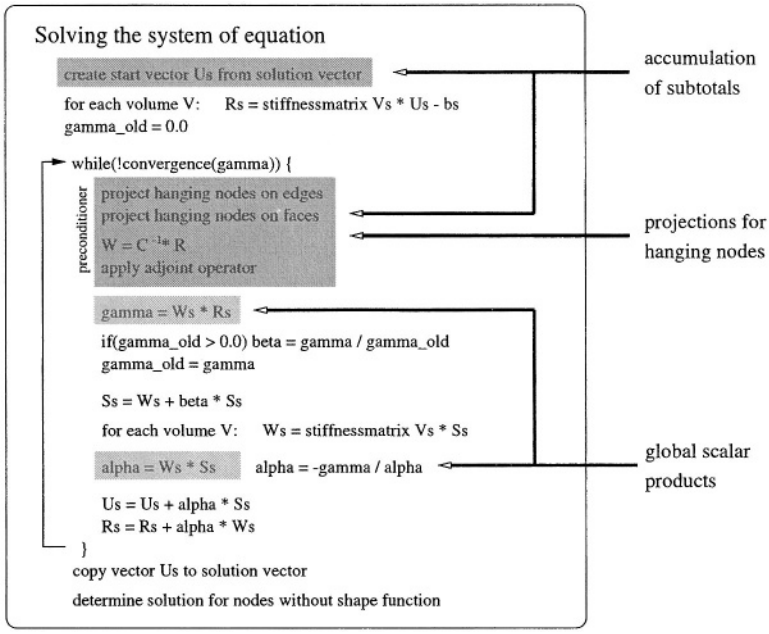


Fig. 3. Solving the system of equation with the parallel PCGM. Shaded areas indicate global data exchange. Capital letters denote vectors.

5 Experimental Results

To gain experimental results two platforms have been used: XEON a 16x2 SMP cluster of 16 PCs with 2.0 GHz Intel Xeon processors running Linux and SB1000 a 4x2 SMP cluster of 4 SunBlade 1000 with 750 MHz UltraSPARC3 processors running Solaris. One process is assigned to each cluster node which enforces network communication. For parallel and sequential measurements linear finite elements and the Jacobi preconditioner have been used. We consider three examples: *layer3* a boundary layer for the convection-diffusion equation $(-\epsilon\Delta u + u = 1 \quad in\ \Omega = (0,1)^3;\quad u = 0\ for\ x,y,z = 0;\quad \frac{\partial u}{\partial n} = 0\ for\ x,y,z = 1)$, *ct01* representing the Lamé equation (2) $(\Omega = (0,2) \times (0,1) \times (0,4);\quad \partial\Omega_1^{(i)} = (0,1) \times (0,1) \times \{0\},\ i = 1,2,3;\quad \partial\Omega_2^{(3)} = (0,2) \times (0,1) \times \{4\},\ g^3 = 1000)$, and *torte4d* a layer near a non-convex edge $(-\Delta u = 0\ in\ \Omega;\ \Omega = (0,3) \times (0, \frac{3}{2}\Pi) \times (0,1);\quad u = 100\ on\ \Gamma_1 = \partial\Omega)$.

The advantages of adaptivity illustrate the volume refinement histories for adaptive and regular subdivision: e. g. 36 vs. 521; 554 vs. 262,144; 1884 vs. 134,217,728 volumes after $3, 6, 9$ program iterations for *ct01*. The number of initial volumes might be less than the number of parallel processes. Therefore regular refinement is performed at program start until a satisfying number of

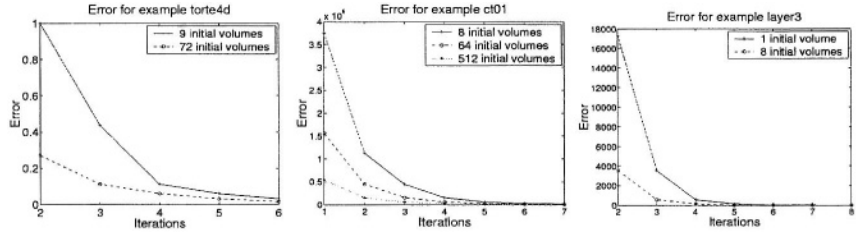


Fig. 4. Error for *torte4d, ct01,* and *layer3* using different initial numbers of volumes.

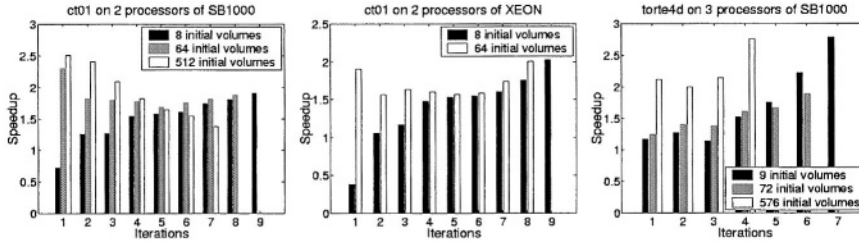


Fig. 5. Speedups for example *ct01* on 2 processors of SB1000 and XEON and for example *torte4d* on 3 processors of SB1000.

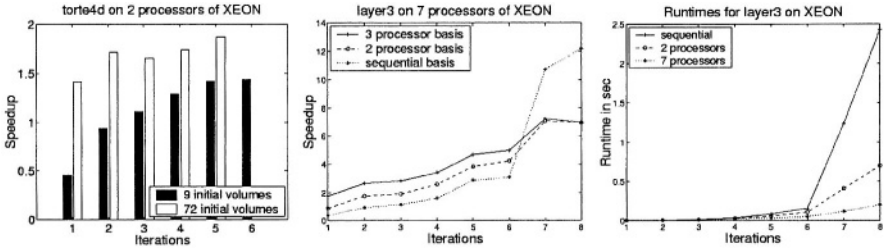


Fig. 6. Speedups for example *torte4d* on 2 and *layer3* on 7 processors of XEON. Comparison of runtimes for example *layer3*.

volumes is reached. Figure 4 compares the development of the maximum error for different initial numbers of volumes.

Figure 5 and Figure 6 (left) depict speedups on SB1000 and XEON for the examples *ct01* and *torte4d* using different initial numbers of volumes. In general speedups increase with growing number of program iterations because the communication overhead compared to the computation effort is reduced. For larger initial numbers of volumes, speedups are in most cases better than for smaller numbers. This is caused by the better computation-communication ratio and by cache effects due to the huge amount of data to process by the sequential program. If the initial number of volumes is too high and many nodes are shared

between the processors, speedup decrease is possible with proceeding refinement (see example *ct01* on SB1000).

On the right of Figure 6 sequential and parallel runtimes on XEON are compared for *layer3*. After 6 iterations runtimes increase extremely due to a rapid increase of volumes. Thus cache effects largely influence the achievable speedups (strongly superlinear). Speedups with different calculation bases (sequential, 2, 3 processors) are shown in the middle of Figure 6.

6 Conclusion

We have presented a parallel implementation for adaptive, hexahedral FEM on distributed memory. The numerical algorithm and the parallel realization have been designed to reduce communication effort. The modular structure of the implementation allows internal optimizations without modifying the original algorithm. Tests for three examples deliver good speedup results.

References

1. Meyer, A.: A parallel preconditioned conjugate gradient method using domain decomposition and inexact solvers on each subdomain. Computing **45** (1990) 217–234
2. Meyer, A.: Parallel Large Scale Finite Element Computations. In Cooperman, G., Michler, G., Vinck, H., eds.: LNCIS 226. Springer Verlag (1997) 91–100
3. Meyer, A., Michael, D.: A modern approach to the solution of problems of classic elasto–plasticity on parallel computers. Num. Lin. Alg. with Appl. **4** (1997) 205–221
4. Beuchler, S., Meyer, A.: SPC-PM3AdH v1.0, Programmer's Manual. Technical Report SFB393/01-08, Chemnitz University of Technology (2001)
5. Yserentant, H.: On the multi-level-splitting of the finite element spaces. Numerical Mathematics **49** (1986) 379–412
6. Bramble, J., Pasciak, J., J.Xu: Parallel multilevel preconditioners. Mathematics of Computation **55** (1991) 1–22
7. Kunert, G.: A posteriori error estimation for anisotropic tetrahedral and triangular finite element meshes., Phd Thesis, TU-Chemnitz, Logos Verlag Berlin (1999)
8. Gross, L., Roll, C., Schoenauer, W.: Nonlinear Finite Element Problems on Parallel Computers. In: Proc. of PARA'94. (1994) 247–261

A Modular Design for Parallel Adaptive Finite Element Computational Kernels

Krzysztof Banaś

Section of Applied Mathematics ICM, Cracow University of Technology, Warszawska 24, 31-155 Kraków, Poland, Krzysztof.Banas@pk.edu.pl

Abstract. The paper presents modular design principles and an implementation for computational kernels of parallel adaptive finite element codes. The main idea is to consider separately sequential modules and to add several specific modules for parallel execution. The paper describes main features of the proposed architecture and some technical details of implementation. Advanced capabilities of finite element codes, like higher order and discontinuous discretizations, multi-level solvers and dynamic parallel adaptivity, are taken into account. A prototype code implementing described ideas is also presented.

1 Introduction

The often used model for parallelization of finite element codes is to consider a library of communication routines that handle transfer of finite element data structures, taking into account complex inter-relations between them [1]. After the transfer of e.g. an element data structure, all required connectivities (such as, for example, constituting faces and vertices, neighboring elements, children and father elements) must be restored, either directly from transfered data or by suitable computations. In such a model, main modules of a finite element code, most importantly mesh manager, must handle parallelism explicitly, by calling respective transfer procedures. As a result, despite the splitting between a communication library and a finite element code, both have to be aware of finite element technical details and parallel execution details.

In the second popular model [2] standard communication routines are employed. Then, parallelization concerns the whole code (or its main parts). This effectively means that sequential parts are replaced by new parallel components.

In the present paper an alternative to both approaches is proposed. The main modules of sequential finite element codes (except linear solver) remain unaware of parallel execution. Additional modules are added that fully take care of parallelism. These modules are tailored to the needs of parallelization of sequential parts, in order to achieve numerical optimality and execution efficiency.

The paper is organized as follows. In Sect. 2 some assumptions on finite element approximation algorithms and codes, that are utilized in parallelization process, are described. The next section concerns assumptions on a target environment for which parallel codes are designed. Algorithms fitting the proposed

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 155–162, 2004.
© Springer-Verlag Berlin Heidelberg 2004

model of parallel execution are described in Sect. 4. Section 5 presents an architecture of parallel codes, with main parallel modules specified, while Sect. 6 considers in more detail the main tasks performed by parallel modules. Section 7 concerns implementation of parallel modules. Section 8 describes some numerical experiments. Conclusions are presented in Sect. 9.

2 Sequential Algorithms and Codes

The model of parallelization presented in the paper is applicable to a broad class of finite element codes, including complex adaptive codes for coupled multi-physics problems. It is assumed that several meshes and several approximation fields may be present in a simulation. Meshes may be adaptive and non-conforming. Approximation fields may be vector fields and may provide higher order approximation. All types of adaptivity, including anisotropic and *hp,* can be handled. The interface between the finite element code and a linear solver allows for the use of multi-level (multigrid) solvers.

In a prototype implementation, described in later sections, it is assumed that the finite element code is split into four fundamental modules, based on four separate data structures [3]: mesh manipulation module with mesh data structure, approximation module with finite element approximation data structure, linear equations solver (or interface to an external solver) with multi-level matrix data structure and problem dependent module with all problem specific data. Although this splitting is not necessary in order to apply parallelization process described in the paper, it facilitates the presentation of the process as well as its practical implementation.

3 Target Parallel Execution Environment

The architecture is developed for the most general to-date execution environment, a system with message passing. Any hardware system that supports message passing may be used as a platform for computations.

Naturally for PDEs, the problem and program decomposition is based on spatial domain decomposition. The computational domain is partitioned into subdomains and main program data structures are split into parts related to separate subdomains. These data structures are distributed among processes executed on processors with their local memories. Processes are obtained by the Single Program Multiple Data (SPMD) strategy and realize main solution tasks in parallel. The most natural and efficient is the situation where there is one-to-one correspondence between processes and processors in a parallel machine, but other mappings are not excluded. In the description it is assumed that there is a unique assignment: subdomain–process–processor–local memory.

4 Parallel Algorithms

From the three main phases of adaptive finite element calculations, creating a system of linear equations, solving the system and adapting the mesh, only solving the system is not "embarrassingly" parallel. Numerical integration, system matrix aggregation, error estimation (or creation of refinement indicators), mesh refinement/derefinement are all local processes, on the level of a single mesh entity or a small group of entities (e.g. a patch of elements for error estimation).

Thanks to this, with a proper choice of domain decomposition, it is possible to perform all these local (or almost local) tasks by procedures taken directly from sequential codes. There must exist however, a group of modules that coordinate local computations spread over processors.

The only part of computational kernels that involve non-local operations is the solution of systems of linear equations. However, also here, the choice of Krylov methods with domain decomposition preconditioning guarantees optimal complexity with minimal number of global steps.

5 An Architecture for Parallel Codes

Fig. 1 presents an architecture for parallel adaptive finite element computational kernels. Four fundamental sequential modules are separated from additional, parallel execution modules. The structure of interfaces between all modules is carefully designed to combine maintainability, that require minimal interfaces, with flexibility and efficiency, for which more intensive module interactions are often necessary.

The main module to handle tasks related to parallel execution is called domain decomposition manager, according to the adopted strategy for parallelization. It has a complex structure that reflects the complex character of performed operations.

6 Main Parallel Solution Tasks

Main tasks related to parallel execution of finite element programs include:

- mesh partitioning
- data distribution
- overlap management
- maintaining mesh and approximation data coherence for parallel adaptivity
- load balancing and associated data transfer
- supporting domain decomposition algorithms

Mesh partitioning, algorithms and strategy, is not considered in the current paper. It is assumed that there exist an external module that provides non-overlapping mesh partitioning according to specified criteria. The criteria must include the standard requirements for keeping load balance and minimizing the

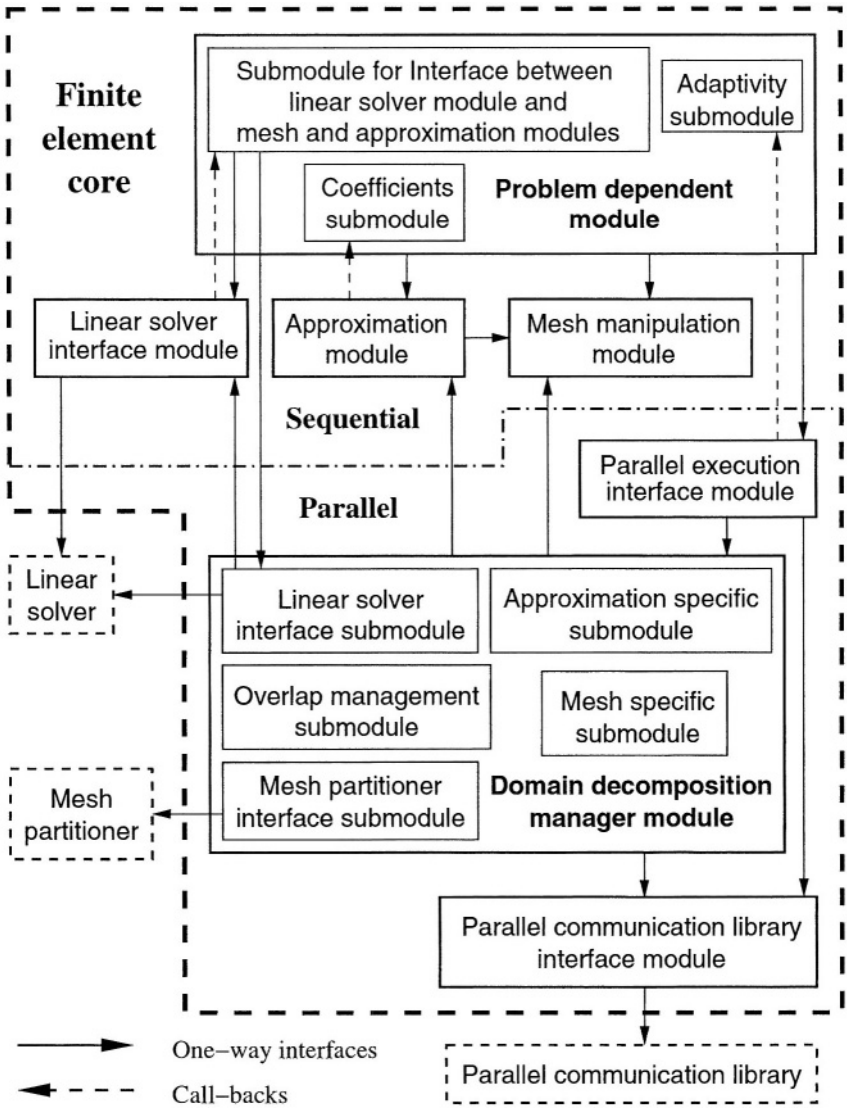


Fig. 1. Diagram of the proposed modular architecture for computational kernels of parallel adaptive finite element codes

extent of inter-subdomain boundary. Keeping load balance for all stages of computations, especially taking into account multi-level linear equations solvers [4], may be a difficult, if not impossible, task. Usually some compromise is postulated among requirements posed by different phases of computations.

Each mesh entity (and in consequence the related approximation data structure) is assigned to a single submesh (subdomain). Subdomains are distributed among processes (processors, local memories), creating an ownership relation between mesh entities and processes (processors, local memories). Each local memory stores all data related to owned entities and each processor performs main solution tasks operating on owned entities.

The existence of overlap (i.e. storing in local memory not owned, "ghost", mesh entities) is advantageous for several tasks in the solution procedure. These tasks include obviously multi-level overlapping domain decomposition preconditioning. Also error estimation, mesh refinement and derefinement benefit from storing data on neighbors of owned entities. The existence of overlap allows for utilizing more local operations and reduces the inter-processor communication. In exchange, more storage is required locally and some operations are repeated on different processors.

The amount of overlap depends on the profits achieved from local storage, which further depends not only on utilized algorithms, but also on computer architectures and interconnection networks employed. For implementation it is assumed that the amount of created overlap is indicated by the maximal extent of data, not available in the initial non-overlapping decomposition, necessary for *any* task operating on local data. Such a choice was made to adapt codes to slower parallel architectures based on networks.

It is a task of domain decomposition manager to create an overlap and to ensure that the overlap data is in a coherent state during computations. Proper values have to be provided, despite the fact that different modules and routines use and modify different parts of overlap data at different times. This task is important for parallel mesh modifications, especially when irregular (non-confirming) meshes are allowed.

Mesh modifications create load imbalance in the form of improper distribution of mesh and approximation entities between subdomains. It is assumed that in the code there is a special, possibly external, module that computes "proper" data distribution. The original mesh partitioner or a separate repartitioner can be used. Additionally to standard partitioning requirements, the module should also aim at minimizing data transfer between processors when regaining balance.

Taking the new partition supplied by the repartitioning module as an input, the domain decomposition module performs mesh transfer. To minimize data traffic, mesh entities must not be transfered separately, but grouped together, to form a patch of elements. Necessary parts of data structure, related to whole patches, are then exchanged between indicated pairs of processors.

Supporting domain decomposition algorithms consist in performing standard vector operations in parallel (such as scalar product or norm) and exchanging data on degrees of freedom close to inter-subdomain boundary between processors assigned to neighboring subdomains. Once again the operations can be cast into the general framework of keeping overlap data (approximation data in this case) stored in local memories in a coherent state. A proper coordination of data exchange with multi-level solution procedure has to be ensured.

7 Implementation

The basis for parallel implementation is formed by an assumption that every mesh entity (together with associated approximation data structure containing degrees of freedom) is equipped with a global (inter-processor) identifier (IPID). This identifier can be understood as a substitute for a global address space used in sequential codes. The IPID is composed of a processor (subdomain) number and a local (to a given processor) identifier. IPIDs are not known to sequential modules of the code. The domain decomposition manager creates an overlap and assigns IPIDs to all mesh entities. Whenever data not stored locally is necessary for computations, domain decomposition manager can find its owning processor and requests the data using suitable calls. With this implementation, keeping the local data structures in a coherent state means keeping a unique assignment of IPIDs to all mesh and approximation entities and data structures.

According to the design assumptions, the changes in the sequential routines are kept minimal. During refinements, children entities remain local to the same processor as their parents. During derefinements, all children entities are either already present locally or are transferred to one chosen processor (e.g. if multi-level computations are performed, the chosen processor may be the one assigned to a parent entity). To assign IPIDs to newly created entities, their lists are passed from mesh manipulation module to domain decomposition manager.

For the linear solver, additional routines are created for performing global vector operations and for exchanging data on overlap DOFs. In the proposed implementation these routines are simple wrappers for domain decomposition manager routines that perform actual operations.

7.1 Interfaces with Communication Libraries

It is assumed that codes use a set of generic send/receive and group communication operations. Additionally, initialization and finalization procedures are specified. All these have to be implemented for various *standard* communication libraries. In the example implementation a model of buffered send/receive operations is employed. The data to be sent are first packed into a buffer and then the whole buffer is sent. Procedures in that model can easily be implemented for MPI standard, as well as packages like PVM.

8 Numerical Examples

Two, simple from numerical point of view but demanding from the point of view of technical difficulties, computational examples are presented as a proof of concept. Both use a prototype implementation of the presented architecture in a discontinuous Galerkin *hp*-adaptive parallel code for 3D simulations.

The first example is a pure convection problem, with a rectangular pattern traveling through a 3D medium. Dynamic adaptivity is employed in this case with two levels of refinement, 1-irregular meshes and adaptations performed after

each time step. To minimize interprocessor communication for small fluctuations of subdomain sizes, load imbalance (measured by the ratio of the maximal or minimal number of degrees of freedom to the average number of degrees of freedom in a subdomain) up to 10% is allowed. When this limit is exceeded, repartitioning takes place and the balance is regained through the transfer of mesh entities.

In the example run, four processors and four subdomains were used that resulted in the average number of degrees of freedom around 5000 per subdomain. Mesh transfers were performed on average after each three steps. As a hardware platform a 100Mbit Ethernet network of PCs was used. PCs were equipped with 1.6 GHz Pentium 4 processors and 1 GByte memory. An average mesh transfer involved several thousand mesh entities. The overall speedup for four processors was equal to 2.67, taking into account times for repartitioning and mesh transfer.

Table 1. Parallel performance for 10 iterations of the preconditioned GMRES method and discontinuous Galerkin approximation used for solving Laplace's equation in a box domain (description in the text).

Single level preconditioner						
N_{DOF}	N_{proc}	Error*10^9	Rate	Time	Speed up	Efficiency
3 129 344	2	48.041	0.738	70.76	1.00	100%
	4	47.950	0.738	35.63	1.98	99%
	8	48.748	0.739	17.71	3.99	100%
Three level preconditioner						
N_{DOF}	N_{proc}	Error*10^9	Rate	Time	Speed up	Efficiency
3 129 344	2	0.027	0.350	111.16	1.00	100%
	4	0.027	0.350	57.76	1.92	96%
	8	0.027	0.348	33.15	3.35	84%

The second example is Laplace's equation in the box $[0,1]^3$ with assumed known exact solution. Results of two experiments are presented. In the first experiment the same network of PCs as for convection problem was used. The experiment consisted in solving the problem for a mesh with 3 129 344 degrees of freedom, obtained by consecutive uniform refinements of an initial mesh. Single level and three level multigrid preconditioning for the GMRES solver with Schwarz methods as smoothers was employed for solving linear equations. Table 1 presents results for *10 iterations* of the preconditioned GMRES method, to focus on the efficiency of parallel implementation of the code. N_{proc} is the number of workstations solving the problem. *Error* is the norm of residual after 10 GMRES iterations and *Rate* is the total GMRES convergence rate during solution. Execution time *Time* is a wall clock time, that includes generation of linear systems (numerical integration) as well. Speed-up and efficiency are computed in the standard way. The run with 2 PCs is taken as a reference since the problem was too large to fit into the memory of a single PC.

The second experiment for the second example was undertaken to test the scalability of the system. The experiment was performed on a cluster of 32 Pentium 3 PCs with 512 MByte memory each and 100 Mbit Ethernet interconnection. The mesh was obtained by another uniform refinement of the mesh from the previous experiment yielding 25 034 752 degrees of freedom. The data structure occupied 4.5 GBytes of memory and parallel adaptations were *necessary* to reach this problem size. Because of memory constraints (16 GBytes) a single level Schwarz preconditioning for GMRES was used, resulting in convergence rate equal to 0.9. The error reduction of 10^{-9} was obtained in 200 iterations that took 20 minutes to perform. Despite the increase in the number of iterations, the scalability of parallel implementation (related to the time of a single iteration) was maintained.

9 Conclusions

The presented model allows for relatively easy parallelization of existing finite element codes, with much of sequential parts of codes retained. The results of numerical experiments with the prototype implementation show good efficiency, making a model feasible solution for migrating finite element codes to high performance parallel environments.

Acknowledgments. The author would like to thank Prof. Peter Bastian from IWR at the University of Heidelberg for invitation to IWR and granting access to IWR's computational resources, used in the last described numerical experiment. The support of this work by the Polish State Committee for Scientific Research under grant 7 T11F 014 20 is also gratefully acknowledged.

References

1. Bastian, P., Birken, K., Johannsen, K., Lang, S., Neuss, N., Rentz-Reichert, H., Wieners, C.: UG - a flexible software toolbox for solving partial differential equations. Computing and Visualization in Science **1** (1997) 27–40
2. J.-F.Remacle, O.Klaas, J.E.Flaherty, M.S.Shephard: A Parallel Algorithm Oriented Mesh Database. Report 6, SCOREC (2001)
3. K.Banaś: On a modular architecture for finite element systems. I. Sequential codes. Computing and Visualization in Science (2004) accepted for publication.
4. Bastian, P.: Load balancing for adaptive multigrid methods. SI AM Journal on Scientific Computing **19** (1998) 1303–1321

Load Balancing Issues for a Multiple Front Method

Christophe Denis[1], Jean-Paul Boufflet[1], Piotr Breitkopf[2], Michel Vayssade[2], and Barbara Glut[3]*

[1] Department of Computing Engineering, UMR 6599 Heudiasyc,
Compiègne University of Technology, BP 20529
F-60205 Compiègne cedex, France
[2] Department of Mechanical Engineering, UMR 6066 Roberval
Compiègne University of Technology, BP 20529
F-60205 Compiègne cedex, France
{Christophe.Denis,Jean-Paul.Boufflet,
Piotr.Breitkopf,Michel.Vayssade}@utc.fr
[3] Institute of Computer Science
AGH University of Science and Technology
Cracow, Poland

Abstract. We investigate a load balancing strategy that uses a model of the computational behavior of a parallel solver to correct an initial partition of data.

1 Introduction

We deal with linear systems $K \cdot u = f$ issued from finite elements. The frontal approach interleaves assembly and elimination avoiding to directly manage the entire matrix K. A variable is eliminated when its corresponding equation is fully summed (I. Duff et al [1,2]). Rather than parallelize an existing code (P.R. Amestoy et al [3]), one can perform tasks in independent modules like in J. Scott [4] *MP42* solver based on the frontal code by I. Duff and J. Reid [2]. We use an implementation of a multiple front parallel method in the context of our academic software *SIC* [5,6].

The domain is partitioned using *METIS* [7] and *CHACO* [8]. This initial partition tends to minimize the communications and to balance the subdomain amount of data assuming that the computation cost is proportional to the number of vertices of the subgraph and that the order of assembly does not matter.

[9] seems to confirm the analysis presented by B. Hendrickson [10,11]: equipartitioning of data volumes does not result systematically in well balanced computational times.

We design a load balancing process transferring finite elements between subdomains to improve the initial partition. Test data are from the PARASOL project (http://www.parallab.uib.no/parasol).

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 163–170, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Problem Formulation

We use a non-overlapping domain decomposition:

1. the graph G_{elem} associated with the finite element mesh is partitioned into N_s subdomains $SD^{(j)}$;
2. each $SD^{(j)}$ is partially condensed in parallel;
3. an interface problem is built and then treated.

This process for an equivalent assembled matrix K can be block ordered:

$$\begin{pmatrix} K_{ii}^{(1)} & & & K_{ib}^{(1)} \\ & \ddots & & \vdots \\ & & K_{ii}^{(N_s)} & K_{ib}^{(N_s)} \\ K_{bi}^{(1)} & \dots & K_{bi}^{(N_s)} & K_{bb} = \sum_{j=1}^{N_s} K_{bb}^{(j)} \end{pmatrix}$$

Subscript i indicates "internal" and b"boundary". $K_{ii}^{(j)}$ are the terms of K associated with the internal variables of $SD^{(j)}$. The terms of $K_{bi}^{(j)}$ (resp. $K_{ib}^{(j)}$) correspond to the interactions between internal variables of $SD^{(j)}$ and the boundary ones. For each $SD^{(j)}$, we build the following matrix and a partial LU condensation

$$\begin{pmatrix} K_{ii}^{(j)} & K_{ib}^{(j)} \\ K_{bi}^{(j)} & K_{bb}^{(j)} \end{pmatrix} = \begin{pmatrix} L_{ii}^{(j)} & 0 \\ L_{bi}^{(j)} & I \end{pmatrix} \cdot \begin{pmatrix} U_{ii}^{(j)} & U_{ib}^{(j)} \\ 0 & S_{bb}^{(j)} \end{pmatrix} \tag{1}$$

The block $S_{bb}^{(j)}$ denotes the local Schur complement of $SD^{(j)}$.

$$S = \sum_{j=1}^{N_s} S_{bb}^{(j)} = \sum_{j=1}^{N_s} \left(K_{bb}^{(j)} - (L_{bi}^{(j)} \cdot U_{ib}^{(j)}) \right) \tag{2}$$

Using (1) we get the global Schur complement matrix

$$S = \sum_{j=1}^{N_s} \left(K_{bb}^{(j)} - K_{bi}^{(j)} \cdot (K_{ii}^{(j)})^{-1} \cdot K_{ib}^{(j)} \right) \tag{3}$$

We use a frontal method to partially condense the matrices associated with each $SD^{(j)}$ and to treat the interface problem. The nested multiple front approach is based on the treatment of groups of $(K_{bb}^{(j)} - K_{bi}^{(j)} \cdot (K_{ii}^{(j)})^{-1} \cdot K_{ib}^{(j)})$. Matrices $S_{bb}^{(j)}$ can be viewed as a super-elements that can be assembled in a frontal matrix and partially condensed. The computational scheme we consider is a tree of tasks (1).

Definition 1. *A computation tree $\mathcal{A}^{N_s,y}$ has N_s leaves and y levels.*

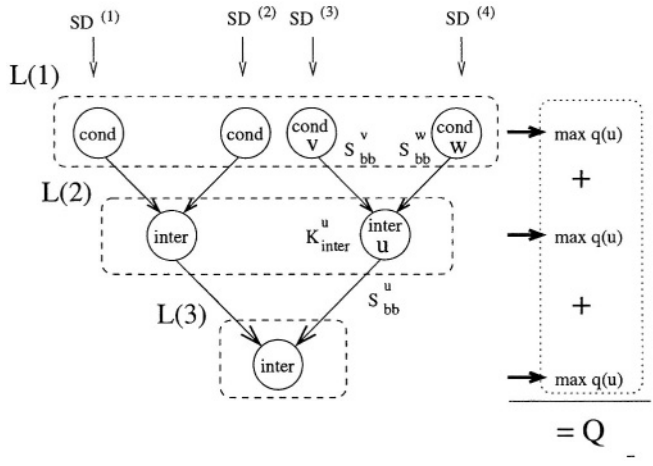


Fig. 1. The computation tree $\mathcal{A}^{4,3}$ and the principle of the estimation of the computation

A task u is associated with a vertex u of the computation tree and $q(u)$ is an estimation of the number of operations. Let $L(i)$ be the set of vertices of $\mathcal{A}^{N_s,y}$ at level iThe leaves in $L(1)$ correspond to the partial condensations of the $SD^{(j)}$. For a task u associated with an internal vertex of $\mathcal{A}^{N_s,y}$ we define:

- S^u_{bb} obtained by partial condensation on $SD^{(j)}$ or on the assembly K^u_{inter} of two matrices $S^{(i)}_{bb}$;
- K^u_{inter} the interface matrix obtained by assembling two matrices $S^{(i)}_{bb}$.

On the computation tree $\mathcal{A}^{4,3}$ of Fig. (1), subdomains $SD^{(3)}$ and $SD^{(4)}$ are partially condensed by tasks v and w. We obtain two matrices S^v_{bb} and S^w_{bb}. They are assembled in the interface matrix K^u_{inter}. The boundary variables between subdomains $SD^{(3)}$ and $SD^{(4)}$ correspond to fully summed rows and columns of K^u_{inter}. We obtain S^u_{bb} by partially condensing K^u_{inter}. The interface problem of level $L(3)$ is then solved and individual variables are obtained by successive back substitutions.

We use a coarse grain parallel approach where tasks are the partial condensation and interface problems. The communication times and the back restitution times are negligible.

The goal is to correct an initial partition of the graph G_{elem}. An estimator of the number of operations of the frontal method is applied on $SD^{(j)}$:

$$Q_1(V_e^{(j)}, SD^{(j)}) = \sum_{\gamma \in assembling} |F_\gamma| + \sum_{\gamma \in elimination} |F_\gamma|^2 \qquad (4)$$

where $V_e^{(j)}$ is the reordering vector of the finite elements of $SD^{(j)}$. Q_1 [12] counts operations and gives 10% error between the estimated time for $SD^{(j)}$ and the actual time $T_{SD^{(j)}}$.

The second estimator counts the number of operations for the partial condensation of K^u_{inter}.

We evaluate then the maximum number of operations $\max q(u)$ for each level $L(i)$ as shown in Fig. (1). The sum $Q = \sum \max q(u)$ provides an estimation of the cost. In an ideal case of equal tasks at each level, Q is a tight estimation, otherwise it gives an upper bound.

We consider balanced trees obtained with multi-level tools [7]. First a unique task of $L(1)$ is assigned per processor. Then, $S^{(i)}_{bb}$ are sent to processors computing the associated tasks according to the computation tree.

Table 1. The PARASOL data and the finite element meshes used for our experiments

name	No. elts	order	name	No. elts	order
MT1	5 328	97 578	SUSPEN_D1	18 171	14 517
SHIPSEC8	35 280	114 919	C1	42 689	34 707
X104	6 019	108 384	MISSILE4	27 804	166 824

3 Principle of the Heuristics

The initial partition $\mathcal{P}$ is first computed using [7]. Then we apply the following heuristics:

1. for each subdomain $SD^{(j)}$ compute first a $V_e^{(j)}$, then $Q_1(V_e^{(j)}, SD^{(j)})$;
2. select $SD^{(max)}$ with maximum estimated number of operations;
3. determine the set $\mathcal{N}_{max}$ of indices of subdomains that are neighbors to $SD^{(max)}$;
4. virtually aggregate the subdomains of $\mathcal{N}_{max}$;
5. compute $Q^{\mathcal{N}_{max}}_{moy}$ the average number of operation of these subdomains;
6. compute Q_{trans} the number of operations to be transferred from $SD^{(max)}$;
7. compute m_t the number of elements to be transferred;
8. transfer a subset of m_t finite elements from $SD^{(max)}$ to the virtual subdomain.

The volume Q_{trans} is half the difference between the maximum estimated number of operations and $Q^{\mathcal{N}_{max}}_{moy}$ and m_t is ratio Q_{trans} over the number of operations per element. By applying this process k times we improve the initial partition. For our experiments we set $k = 100$ and select the best result.

A transfer primitive chooses finite elements near the common boundary in order to limit the growth of the interface. Consider examples from Fig. (2) to Fig. (4).

In Fig. (2) $SD^{(1)}$ has the maximum estimated number of operations. Grey elements are near the boundary between $SD^{(1)}$ and the virtual subdomain $[SD^{(2)},SD^{(3)}]$ ($\mathcal{N}_1 = \{2,3\}$).

Table 2. Q the estimated amount of computation, T_{glob} the real computing time (in s), the obtained gain (in %), and Δ^{mesu} the load balancing criterion

name	tree	method	Q	T_{glob} (s)	gain (%)	Δ^{mesu} (%)
MT1	A	Pmetis	$1,17.10^{11}$	94,4		96
	A	PmetisC	$1,00.10^{11}$	89,6	5,1	98
	B	Pmetis	$1,09.10^{11}$	66,7		71
	B	PmetisC	$8,05.10^{10}$	59.1	11,4	80
	C	Pmetis	$4,26.10^{10}$	43,0		54
	C	PmetisC	$3,15.10^{10}$	36,9	14,2	72
SHIPSEC8	A	Pmetis	$4,02.10^{11}$	266,6		81
	A	PmetisC	$3,85.10^{11}$	239,1	10,3	89
	B	Pmetis	$2,75.10^{11}$	160,5		71
	B	PmetisC	$1,75.10^{11}$	112,7	29,8	89
	C	Pmetis	$1,40.10^{11}$	108,3		71
	C	PmetisC	$1,20.10^{11}$	101,1	6,6	79
X104	A	Pmetis	$2,01.10^{11}$	162,9		98
	A	PmetisC	$2,01.10^{11}$	162,9	0	98
	B	Pmetis	$7,84.10^{10}$	70,1		76
	B	PmetisC	$6,39.10^{10}$	57,4	18.1	91
	C	Pmetis	$4,53.10^{10}$	48,5		46
	C	PmetisC	$4,48.10^{10}$	46,8	3.5	48
SUSPEN_D1	A	Pmetis	$2,67.10^{9}$	21,1		88
	A	PmetisC	$1,19.10^{9}$	9,9	53,1	73
	B	Pmetis	$1,20.10^{9}$	9,9		50
	B	PmetisC	$9,28.10^{8}$	4,1	58,6	80
	C	Pmetis	$7,75.10^{8}$	6,7		44
	C	PmetisC	$4,05.10^{8}$	3,8	43,8	73
C1	A	Pmetis	$1,78.10^{10}$	129,9		80
	A	PmetisC	$8,87.10^{9}$	84,2	35,2	88
	B	Pmetis	$9,23.10^{9}$	99,4		54
	B	PmetisC	$3,38.10^{9}$	30,1	69,7	90
	C	Pmetis	$3,11.10^{9}$	24,7		54
	C	PmetisC	$1,74.10^{9}$	15,4	37,7	90
MISSILE4	A	Pmetis	$1,44.10^{11}$	1220		71
	A	PmetisC	$3,91.10^{10}$	395,3	67,6	99
	B	Pmetis	$1,04.10^{11}$	891,6		65
	B	PmetisC	$4,81.10^{10}$	406,6	54,4	67
	C	Pmetis	$6,13.10^{10}$	448,3		59
	C	PmetisC	$2,26.10^{10}$	204,5	54,4	93

We compute a level structure through $SD^{(1)}$ from the boundary elements of $[SD^{(2)},SD^{(3)}]$. We apply the *BFS* algorithm on the element graph of $SD^{(1)}$ initializing its queue with boundary elements corresponding to level 0. We obtain a spanning tree where level l contains elements at a distance of l edges to level 0. It may be seen as using a virtual vertex r connecting $SD^{(1)}$ and its associated virtual subdomain (Fig. (3)). We assume m_t=4. We then transfer selected elements to the neighbor subdomains (Fig. (4)).

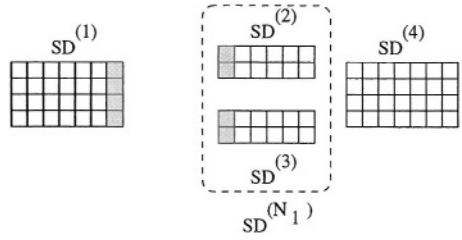


Fig. 2. The initial partition of the domain into 4 subdomains

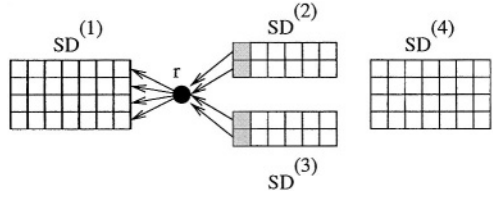


Fig. 3. Initialisation of the root r of the level structure in order to select the finite elements to be transfered from $SD^{(1)}$ to the virtually aggregated subdomain

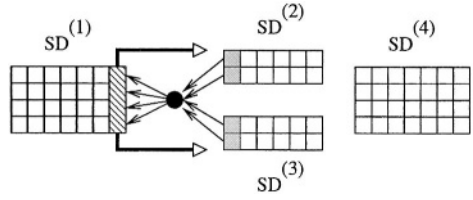


Fig. 4. m_t finite element are transfered

4 Results

The experiments were performed on a 10 Athlon 2200+, 1Gb bi-processors cluster, running LINUX Red Hat 7.1, with a 1 Gbit/s Ethernet network. Table (1) gives the sizes of the PARASOL data, and of some arbitrary meshes. The *order* column gives the size of the assembled matrix.

Three types of computation tree were used, and we define the labels: *A* for $\mathcal{A}^{2,2}$, 2 subdomains and 2 levels; *B* for $\mathcal{A}^{4,3}$, 4 subdomains and 3 levels and *C* for $\mathcal{A}^{8,4}$, 8 subdomains and 4 levels.

Table (2) presents the results: estimates Q, and measures T_{glob}. *Pmetis* is the original *METIS* decomposition and *PmetisC* is the corrected one. $T_{SD^{(j)}}$ is measured for each $SD^{(j)}$ along with the load balancing criterion:

$$\Delta^{mesu} = \frac{1}{N_s} \frac{\sum_{j=1}^{N_s} T_{SD^{(j)}}}{\max_j T_{SD^{(j)}}}$$

In the ideal case the $T_{SD^{(j)}}$ are equal and $\Delta^{mesu} = 1$.

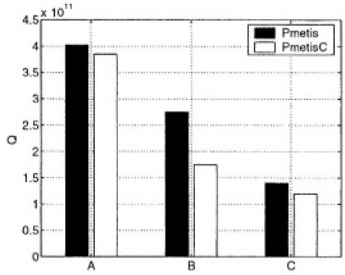


Fig. 5. Q : the estimated amount of computation before and after applying the heuristics for the SHIPSEC8 data

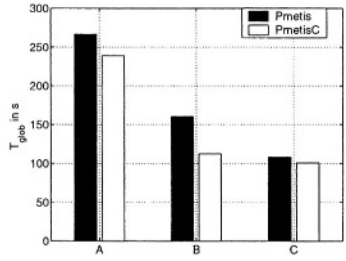


Fig. 6. T_{glob} : the real computing time (in s) before and after applying the heuristics for the SHIPSEC8 data

Table (2) shows that Δ^{mesu} is improved. The transfer primitive was modified in order to limit the number of interface nodes.

Figs. (5) and (6) show a good correlation between Q and T_{glob}. However, we do not obtain a perfect balance, because the estimations do not reflect exactly the real computations. Moreover, moving elements influences the ordering and consequently the computation time. It is therefore difficult to attain $\Delta^{mesu} = 1$. As a rule, fewer than 10 iterations of the heuristics provide the maximum gain reported in Table (2).

5 Conclusion

We propose a heuristics to correct an initial domain decomposition based on equal volumes of data, in order to balance the estimated number of operations of a multiple-front method. With this coarse-grained parallel approach, the preliminary results obtained on the benchmark improve computing time. The modification of the boundary due to the transfer of finite elements can increase the number of interface nodes and the size of the interface problem.

References

1. Duff, I., Erisman, A., Reid, J.: Direct Methods for Sparse Matrices. Monographs on Numerical Analysis. Clarendon Press - Oxford (1986)
2. Duff, I.S., Scott, J.A.: MA42 – A new frontal code for solving sparse unsymmetric systems, technical report ral 93-064. Technical report, Chilton, Oxon, England (1993)
3. P.R. Amestoy, I.S. Duff, J.Y.L., Koster, J.: A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM J. Matrix Anal. Appl. **23** (2001) 15–41
4. Scott, J.: The design of a parallel frontal solver,technical report ral-tr99-075. Technical report, Rutherford Appleton Laboratory (1999)

5. Escaig, Y., Vayssade, M., Touzot, G.: Une méthode de décomposition de domaines multifrontale multiniveaux. Revue Européenne des Eléments Finis **3** (1994) 311–337
6. Breitkopf, P., Escaig, Y.: Object oriented approach and distributed finite element simulations. Revue Européenne des Eléments Finis **7** (1998) 609–626
7. Karypis, G., Kumar, V.: Metis : A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Technical report, University of Minnesota, Department of Computer Science (1998)
8. Hendrickson, B., Leland, R.: The chaco user's guide, version 2.0. Technical report, Sandia National Laboratories (1995)
9. Boufflet, J., Breitkopf, P., Denis, C., Rassineux, A., Vayssade, M.: Optimal element numbering schemes for direct solution of mechanical problems using domain decomposition method. In: 4th ECCOMAS Solid Mechanics Conference. (2000) Espagne.
10. Hendrickson, B.: Graph partitioning and parallel solvers: Has the emperor no clothes? In: Irregular'98, Lecture Notes in Computer Science. Volume 1457. (1998) 218–225
11. Hendrickson, B.: Load balancing fictions, falsehoods and fallacies. Applied Mathematical Modelling **25** (2000) 99–108
12. Boufflet, J., Breitkopf, P., Denis, C., Rassineux, A., Vayssade, M.: Equilibrage en volume de calcul pour un solveur parallèle multi-niveau. In: 6ème Colloque National en Calcul des Structures. (2001) 349–356 Giens, France.

Multiresolutional Techniques in Finite Element Method Solution of Eigenvalue Problem

Marcin Kamiński

Chair of Mechanics of Materials, Technical University of Łódź
Al. Politechniki 6, 93-590 Łodz, POLAND, tel/fax 48-42-6313551
marcin@kmm-lx.p.lodz.pl, marcinka@p.lodz.pl

Abstract. Computational analysis of unidirectional transient problems in multiscale heterogeneous media using specially adopted homogenization technique and the Finite Element Method is described below. Multiresolutional homogenization being the extension of the classical micro-macro traditional approach is used to calculate effective parameters of the composite. Effectiveness of the method is compared against previous techniques thanks to the FEM solution of some engineering problems with real material parameters and with their homogenized values. Further computational studies are necessary in this area, however application of the multiresolutional technique is justified by the natural multiscale character of composites.

1 Introduction

Wavelet analysis [1] perfectly reflects the very demanding needs of composite materials computational modeling. It is due to the fact that wavelet functions like Haar, sinusoidal (harmonic), Gabor, Morlet or Daubechies, for instance, relating neighboring scales in the medium analysed can efficiently model a variety of heterogeneities preserving composites periodicity, for instance. It is evident now that wavelet techniques may serve for analysis in the finest scale by various numerical techniques [2,4,5] as well as using multiresolutional analysis (MRA) [3,5,6,8]. The first method leads to the exponential increase of the total number of degrees of freedom in the model, because each new decomposition level almost doubles this number, while an application of the homogenization method is connected with determination of effective material parameters.

Both methodologies are compared here in the context of eigenvalue problem solution for a simply supported linear elastic Euler-Bernoulli beam using the Finite Element Method (FEM) computational procedures. The corresponding comparison made for a transient heat transfer has been discussed before in [5]. Homogenization of a composite is performed here through (1) simple spatial averaging of composite properties, (2) two-scale classical approach [7] as well as (3) thanks to the multiresolutional technique based on the Haar wavelets. An application of the symbolic package MAPLE guarantees an efficient integration of algebraic formulas defining effective properties for a composite with material properties given by some wavelet functions.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 171–178, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Multiresolutional Homogenization Scheme

MRA approach uses the algebraic transformation between various scales provided by the wavelet analysis to determine the fine-scale behavior and introduce it explicitly into the macroscopic equilibrium equations. The following relation:

$$\Omega_0 \subset \Omega_{-1} \subset \Omega_{-2} \subset ... \tag{1}$$

defines the hierarchical geometry of the scales and this chain of subspaces is so defined that Ω_j is "finer" than Ω_{j+1}. Further, let us note that the main assumption on general homogenization procedure tor transient problems is a separate averaging of the coefficients from the governing partial differential equation responsible for a static behavior and of the unsteady component. The problem can be homogenized only if its equilibrium can be expressed by the following operator equation:

$$BT + u + \lambda = L(AT + v) \tag{2}$$

This equation in the multiscale notation can be rewritten at the given scale j as

$$B_0^{(j)} T_0^{(j)} + u_0^{(j)} + \lambda = L_0\left(A_0^{(j)} T_0^{(j)} + v_0^{(j)}\right), \tag{3}$$

with the recurrence relations used j times to compute $B_0^{(j)}, A_0^{(j)}, u_0^{(j)}, v_0^{(j)}$. MRA homogenization theorem is obtained as a limit for $j \to -\infty$

$$B_0^{(-\infty)} T_0^{(-\infty)} + u_0^{(-\infty)} + \lambda = L_0\left(A_0^{(-\infty)} T_0^{(-\infty)} + v_0^{(-\infty)}\right), \tag{4}$$

which enables to eliminate infinite number of the geometrical scales with the reduced coefficients $B_0^{(-\infty)}, A_0^{(-\infty)}$. If the limits defining the matrices $B_0^{(-\infty)}$ and $A_0^{(-\infty)}$ exist, then there exist constant matrices B^h, A^h and forcing terms u^h, v^h, such that the reduced coefficients and forcing terms are given by $B_0^{(-\infty)}, A_0^{(-\infty)}, u_0^{(-\infty)}, v_0^{(-\infty)}$. The homogenized coefficients are equal to

$$A^h = A_0^{(-\infty)}, \; B^h = A^h \tilde{A}^{-1} - I, \tag{5}$$

$$u^h = u_0^{(-\infty)} + \left(I - \tfrac{1}{2}\tilde{A} - \tilde{A}\left(\exp\left(\tilde{A} - I\right)^{-1} A^h\right)^{-1} v^h\right), \tag{6}$$

where

$$\tilde{A} = \log\left(I + \left(I + B_0^{(-\infty)} - \tfrac{1}{2}A^h\right)^{-1} A^h\right). \tag{7}$$

As the example let us review the static equilibrium of elastic Euler-Bernoulli beam

$$-\frac{d}{dx}\left(E(x)\frac{d}{dx}u(x)\right) = f(x); \; x \in [0,1], \tag{8}$$

where E(x), defining material properties of the heterogeneous medium, varies arbitrarily on many scales. The unit interval denotes here the Representative Volume Element (RVE), called also the periodicity cell. This equation can represent linear elastic behavior of unidirectional structures as well as unidirectional heat conduction and other related physical fields. A periodic structure with a small parameter $\varepsilon>0$,

tending to 0, relating the lengths of the periodicity cell and the entire composite, is considered in a classical approach. The displacements are expanded as

$$u(x) = u^{(0)}(x,y) + \varepsilon^1 u^{(1)}(x,y) + \varepsilon^2 u^{(2)}(x,y) + \dots , \tag{9}$$

where $u^{(i)}(x,y)$ are also periodic; the coordinate x is introduced for macro scale, while y - in micro scale. Introducing these expansions into classical Hooke's law, the homogenized elastic modulus is obtained as [6]

$$E^{(eff)} = \left(\int_\Omega \frac{dy}{E(y)} \right)^{-1} . \tag{10}$$

The method called multiresolutional starts from the following decomposition:

$$\begin{cases} \dfrac{d}{dx} u(x) = \dfrac{v(x)}{E(x)} \\ \dfrac{d}{dx} v(x) = -f(x) \end{cases} \tag{11}$$

to determine the homogenized coefficient $E^{(eff)}$ constant for $x \in [0,1]$. Therefore

$$\begin{pmatrix} u(x) \\ v(x) \end{pmatrix} - \begin{pmatrix} u(0) \\ v(0) \end{pmatrix} = \int_0^x \left(\begin{pmatrix} 0 & E(t)^{-1} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u(t) \\ v(t) \end{pmatrix} + \begin{pmatrix} 0 \\ -f(t) \end{pmatrix} \right) dt . \tag{12}$$

The reduction algorithm between multiple scales of the composite consists in determination of such effective tensors $B^{(eff)}$, $A^{(eff)}$, $p^{(eff)}$ and $q^{(eff)}$, such that

$$\left(\mathbf{I} + B^{(eff)} \right) \begin{pmatrix} u(x) \\ v(x) \end{pmatrix} + q^{(eff)} + \lambda = \int_0^x \left(A^{(eff)} \begin{pmatrix} u(t) \\ v(t) \end{pmatrix} + p^{(eff)} \right) dt , \tag{13}$$

where **I** is an identity matrix. In our case we apply

$$B^{(eff)} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}; A^{(eff)} = \begin{pmatrix} 0 & C_1 - 2C_2 \\ 0 & 0 \end{pmatrix}, \quad C_1 = \int_0^1 \frac{dt}{E(t)}; C_2 = \int_0^1 \frac{\left(t - \frac{1}{2}\right) dt}{E(t)} \tag{14}$$

Furthermore, for f(x)=0 there holds $p^{(eff)} = q^{(eff)} = 0$, while, in a general case, $B^{(eff)}$ and $A^{(eff)}$ do not depend on p and q.

3 Multiresolutional Finite Element Method

Let us consider the governing equation

$$-e\nabla^2 u = f , \; x \in \Omega \tag{15}$$

with

$$u = 0 , \; x \in \Gamma_u \subset \partial\Omega . \tag{16}$$

Variational formulation of this problem for the multiscale medium at the scale k is given as

$$\int_\Omega e \nabla u_k \nabla \varphi_k \, d\Omega + \int_\Omega \gamma u_k \varphi_k \, d\Omega = \int_\Gamma f \varphi_k \, d\Gamma , \; x \in \Omega . \tag{17}$$

Solution of the problem must be found recursively by using some transformation between the neighboring scales. Hence, the following nonsingular n x n wavelet transform matrix $\mathbf{W}_k$ is introduced [2]:

$$\mathbf{W}_k = \mathbf{T}_k \begin{bmatrix} \mathbf{T}_{k-1} & 0 \\ 0 & \mathbf{I}_{k-1} \end{bmatrix}, \tag{18}$$

and

$$\boldsymbol{\psi}_k = \mathbf{W}_k^T \boldsymbol{\varphi}_k . \tag{19}$$

$\mathbf{T}_k$ is a two-scale transform between the scales k-1 and k, such that

$$\begin{Bmatrix} \boldsymbol{\varphi}_{k-1} \\ \boldsymbol{\psi}_k \end{Bmatrix} = \mathbf{T}_k^T \boldsymbol{\varphi}_k , \tag{20}$$

with

$$\psi_k^j = \varphi_k^{2j-1}, \; j=1,\ldots,N_k \tag{21}$$

N_k denotes here the total number of the FEM nodal points at the scale k. Let us illustrate the wavelet-based FEM idea using the example of 1D linear two-noded finite element with the shape functions [9]

$$\mathbf{N}^T = \begin{Bmatrix} N_1 \\ N_2 \end{Bmatrix} = \begin{Bmatrix} \frac{1}{2}(1-\xi) \\ \frac{1}{2}(1+\xi) \end{Bmatrix}, \tag{22}$$

where N_1 is valid for ξ=-1 and N_2 – for ξ=1 in local coordinates system of this element. The scale effect is introduced on the element level by inserting new extra degrees of freedom at each new scale. Then, the scale 1 corresponds to first extra multiscale DOF per the original finite element, scale 2 – next two additional multiscale DOFs and etc. It may be generally characterized as

$$\psi_k(\xi) = \psi_k\left(2^{k-1}(1+\xi) - 2j - 1\right) \tag{23}$$

where

$$\begin{cases} 2^{2-k}j - 1 \le \xi \le 2^{2-k}j + 2^{1-k} - 1 \\ 2^{2-k}j + 2^{1-k} - 1 \le \xi \le 2^{2-k}j + 2^{2-k} - 1 \end{cases} \tag{24}$$

The value of k defines the actual scale. The reconstruction algorithm starts from the original solution for the original mesh. Next, the new scales are introduced using the formula

$$\mathbf{u}_k^{2+2^{k-1}+j} = \sum_{i=1}^{N_{old}} \mathbf{N}_i \mathbf{u}_0^i + \sum_{i=1}^{N_{new}} \boldsymbol{\psi}_k^{2+2^{k-1}+j} \Delta \mathbf{u}_k^{2+2^{k-1}+j} . \tag{25}$$

The wavelet algorithm for stiffness matrix reconstruction starts at scale 0 with the smallest rank stiffness matrix

$$\mathbf{K}_0 = \frac{e}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \tag{26}$$

where h is the node spacing parameter. Then, the diagonal components of the stiffness matrix for any k>0 are equal to

$$K_k^{2+2^{K-1}+J} = \frac{2^{k+1}e}{h}. \tag{27}$$

It should be underlined that the FEM so modified reflects perfectly the needs of computational modeling of multiscale heterogeneous media. The reconstruction algorithm can be applied for such n, which assure a sufficient mesh zoom on the smallest scale in the composite.

4 Finite Element Method Equations of the Problem

The following variational equation is proposed to study dynamic equilibrium for the linear elastic system:

$$\int_\Omega \rho \ddot{u}_i \delta u_i d\Omega + \int_\Omega C_{ijkl} \varepsilon_{ij} \delta\varepsilon_{kl} d\Omega = \int_\Omega \rho f_i \delta u_i d\Omega + \int_{\partial\Omega_\sigma} \hat{t}_i \delta u_i d(\partial\Omega) \tag{28}$$

and u_i represents displacements of the system Ω with elastic properties and mass density defined by the elasticity tensor $C_{ijkl}(x)$ and $\rho=\rho(\mathbf{x})$; the vector $\hat{t}_i$ denotes the stress boundary conditions imposed on $\partial\Omega_\sigma \subset \partial\Omega$. Analogous equation for the homogenized medium has the following form:

$$\int_\Omega \rho^{(eff)} \ddot{u}_i \delta u_i d\Omega + \int_\Omega C_{ijkl}^{(eff)} \varepsilon_{ij} \delta\varepsilon_{kl} d\Omega = \int_\Omega \rho^{(eff)} f_i \delta u_i d\Omega + \int_{\partial\Omega_\sigma} \hat{t}_i \delta u_i d(\partial\Omega) \tag{29}$$

where all material properties of the real system are replaced with the effective parameters. As it is known [9], classical FEM discretization returns the following equations for real heterogeneous and homogenized systems are obtained:

$$M_{\alpha\beta}\ddot{q}_\beta + C_{\alpha\beta}\dot{q}_\beta + K_{\alpha\beta}q_\beta = Q_\alpha, \quad M_{\alpha\beta}^{(eff)}\ddot{\overline{q}}_\beta + C_{\alpha\beta}^{(eff)}\dot{\overline{q}}_\beta + K_{\alpha\beta}^{(eff)}\overline{q}_\beta = Q_\alpha. \tag{30}$$

The R.H.S. vector equals to 0 for free vibrations and then an eigenvalue problem is solved using the following matrix equations:

$$\left(K_{\alpha\beta} - \omega_{(\alpha)} M_{\alpha\beta}\right)\Phi_{\beta\gamma} = 0, \quad \left(K_{\alpha\beta}^{(eff)} - \overline{\omega}_{(\alpha)} M_{\alpha\beta}^{(eff)}\right)\overline{\Phi}_{\beta\gamma} = 0. \tag{31}$$

5 Computational Illustration

First, simply supported periodic composite beam is analyzed, where Young modulus E(x) and mass density in the periodicity cell are given by the following wavelets:

$$h(x) = \begin{cases} 20.0E9; 0 \le x \le 0.5 \\ 2.0E9\ 0.5 < x \le 1 \end{cases}, \quad m(x) = 2 + \frac{1}{\sqrt{2\pi\sigma^3}} \frac{x^2}{\sigma^2 - 1} \exp\left(\frac{-x^2}{2\sigma^2}\right), \sigma=-0.4; \tag{32}$$

$$E(x) = 10.0 \cdot h(x) + 2.0E9 \cdot m(x). \tag{33}$$

$$\tilde{h}(x) = \begin{cases} 200; 0 \le x \le 0.5 \\ 20; 0.5 < x \le 1 \end{cases}, \quad \rho(x) = 0.5 \cdot \tilde{h}(x) + 0.5 \cdot m(x). \tag{34}$$

The composite specimen is discretized using each time 128 2-noded linear finite elements with unitary inertia moments. The comparison starts from a collection of the eigenvalues reflecting different homogenization techniques given in Tab. 1. Further, the eigenvalues for heterogeneous beams are given for 1st order wavelet projection in Tab. 2, for 2nd order projection – in Tab. 3, 3rd order - in Tab. 4.

The eigenvalues computed for various homogenization models approximate the values computed for the real composite models with different accuracy - the weakest efficiency is detected in case of spatially averaged composite and the difference in relation to the real structure results increase together with the eigenvalue number and the projections order. The results obtained thanks to MRA projection are closer to those relevant to MRA homogenization for a single RVE in composite; classical homogenization is more effective for increasing number of the cells in this model.

Table 1. Eigenvalues for the simply supported homogenized composite beams

Eigenvalue	Spatial averaging	Classical approach	Multiresolutional model
1	1,184 E12	3,665 E11	6,228 E11
2	1,895 E13	5,864 E12	9,965 E12
3	9,592 E13	2,969 E13	5,045 E13
4	3,032 E14	9,383 E13	1,594 E14
5	7,401E14	2,291 E14	3,893 E14
6	1,535 E15	4,750 E14	8,072 E14

Table 2. Eigenvalues for the simply supported composite beam, 1st order wavelet projection

64 RVEs	32 RVEs	16 RVEs	8 RVEs	4 RVEs	2 RVEs	1 RVE
3,534 E11	3,535 E11	3,537 E11	3,550 E11	3,599 E11	3,829 E11	4,529 E11
5,656 E12	5,660 E12	5,679 E12	5,760 E12	6,137 E12	7,887 E12	2,593 E13
2,864 E13	2,870 E13	2,892 E13	2,991 E13	3,512 E13	4,973 E13	7,317 E13
9,056 E13	9,087 E13	9,216 E13	9,828 E13	1,315 E14	4,867 E14	3,512 E14
2,212 E14	2,224 E14	2,275 E14	2,536 E14	3,758 E14	6,743 E14	6,241 E14
4,591 E14	4,627 E14	4,786 E14	5,655 E14	8,448 E14	1,347 E15	1,678 E15

Table 3. Eigenvalues for the simply supported composite beam, 2nd order wavelet projection

32 RVEs	16 RVEs	8 RVEs	4 RVEs	2 RVEs	1 RVE
3,636 E11	3,639 E11	3,652 E11	3,703 E11	3,936 E11	4,604 E11
5,823 E12	5,842 E12	5,925 E12	6,309 E12	8,006 E12	2,603 E13
2,952 E13	2,975 E13	3,075 E13	3,605 E13	5,090 E13	7,420 E13
9,348 E13	9,480 E13	1,010 E14	1,334 E14	4,875 E14	3,531 E14
2,288 E14	2,340 E14	2,605 E14	3,846 E14	6,803 E14	6,292 E14
4,760 E14	4,921 E14	5,805 E14	8,641 E14	1,362 E15	1,690 E15

Table 4. Eigenvalues for the simply supported composite beam, 3rd order wavelet projection

16 RVEs	8 RVEs	4 RVEs	2 RVEs	1 RVE
3,662 E11	3,674 E11	3,726 E11	3,964 E11	4,664 E11
5,879 E12	5,962 E12	6,354 E12	8,121 E12	2,600 E13
2,993 E13	3,096 E13	3,637 E13	5,174 E13	7,479 E13
9,540 E13	1,017 E14	1,354 E14	4,876 E14	3,529 E14
2,355 E14	2,626 E14	3,903 E14	6,839 E14	6,341 E14
4,954 E14	5,857 E14	8,796 E14	1,373 E15	1,691 E15

Free vibrations for 2 and 3-bays periodic beams are solved using classical and homogenization-based FEM implementation. The unitary inertia momentum is taken in all computational cases, ten periodicity cells compose each bay, while material properties inserted in the numerical model are calculated by spatial averaging, classical and multiresolutional homogenization schemes and compared against the real structure response. First 10 eigenvalues changes for all these beams are contained in Figs. 1,2 – the resulting values are marked on the vertical axes, while the number of eigenvalue being computed – on the horizontal ones.

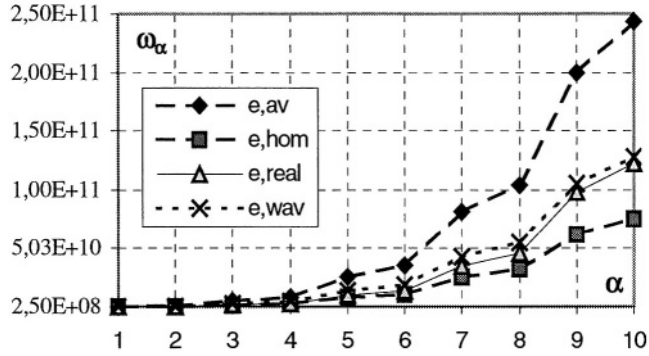


Fig. 1. Eigenvalues progress for various two-bays composite structures

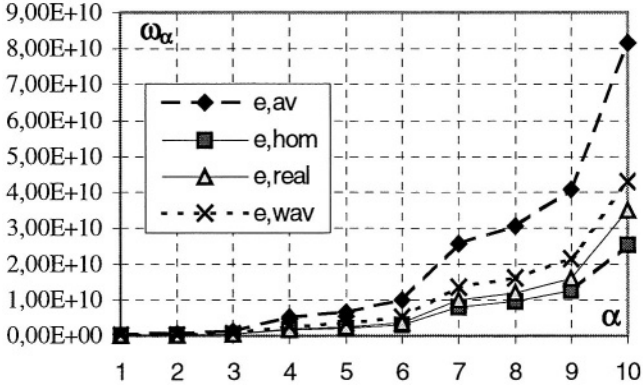


Fig. 2. Eigenvalues progress for various three-bays composite structures

Eigenvalues obtained for various homogenization models approximate the values computed for the real composite with different accuracy - the worst efficiency in eigenvalues modeling is detected in case of spatially averaged composite and the difference in relation to the real structure results increase together with the eigenvalue number. Wavelet-based and classical homogenization methods give more accurate results – the first method is better for smaller number of the bays, and classical homogenization approach is recommended in case of increasing number of the bays and the RVEs. The justification of this observation comes from the fact, that the wavelet function is less important for the increasing number of the periodicity cells in

the structure. Another interesting result is that the efficiency of approximation of the maximum deflections for a multi-bay periodic composite beam by the deflections encountered for homogenized systems increase together with an increase of the total number of the bays.

6 Conclusions

The most important result of the homogenization-based Finite Element modeling of the periodic unidirectional composites is that the real composite behavior is rather well approximated by the homogenized model response. MRA homogenization technique giving more accurate approximation of the real structure behavior is decisively more complicated in numerical implementation since necessity of usage of the combined symbolic-FEM approach. The technique introduces new opportunities to calculate effective parameters for the composites with material properties approximated by various wavelet functions. A satisfactory agreement between the real and homogenized structures models enables the application to other transient problems with deterministic as well as stochastic material parameters.

Multiresolutional homogenization procedure has been established here using the Haar basis to determine complete mathematical equations for homogenized coefficients and to make implementation of the FEM-based homogenization analysis. As it was documented above, the Haar basis approximation gives sufficient approximation of various mathematical functions describing most of possible spatial distributions of composites physical properties.

References

1. Al-Aghbari, M., Scarpa, F., Staszewski, W.J., On the orthogonal wavelet transform for model reduction/synthesis of structures. J. Sound & Vibr. 254(4), pp. 805-817, 2002.
2. Christon, M.A. and Roach, D.W., The numerical performance of wavelets for PDEs: the multi-scale finite element, *Comput. Mech.,* 25, pp. 230-244, 2000.
3. Dorobantu M., Engquist B., Wavelet-based numerical homogenization, *SIAM J. Numer. Anal.,* 35(2), pp. 540-559, 1998.
4. Gilbert, A.C., A comparison of multiresolution and classical one-dimensional homogenization schemes, *Appl. & Comput. Harmonic Anal.,* 5, pp. 1-35, 1998.
5. Kamiński, M., Multiresolutional homogenization technique in transient heat transfer for unidirectional composites, *Proc. 3rd Int. Conf. Engineering Computational Technology,* Topping, B.H.V. and Bittnar Z., Eds, Civil-Comp Press, 2002.
6. Kamiński, M., Wavelet-based finite element elastodynamic analysis of composite beams, *WCCM V,* Mang, H.A., Rammerstorfer, F.G. and Eberhardsteiner, J., Eds, Vienna 2002.
7. Sanchez-Palencia, E., *Non-homogeneous Media and Vibration Theory. Lecture Notes in Physics,* vol. 127, Springer-Verlag, Berlin, 1980.
8. Steinberg, B.Z. and McCoy, J.J., A multiresolution homogenization of modal analysis with application to layered media, *Math. Comput. & Simul.,* 50, pp. 393-417, 1999.
9. Zienkiewicz, O.C. and Taylor, R.L., *The Finite Element Method.* Heinemann-Butterworth, 2000.

Self-Organizing Multi-layer Fuzzy Polynomial Neural Networks Based on Genetic Optimization

Sung-Kwun Oh[1], Witold Pedrycz[2], Hyun-Ki Kim[3], and Jong-Beom Lee[1]

[1] Department of Electrical Electronic and Information Engineering, Wonkwang University, 344-2, Shinyong-Dong, Iksan, Chon-Buk, 570-749, South Korea
{ohsk, ipower}@wonkwang.ac.kr
http://autosys.wonkwang.ac.kr

[2] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2G6, Canada
and Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
pedrycz@ee.ualberta.ca

[3] Department of Electrical Engineering, University of Suwon, South Korea
hkkim@suwon.ac.kr

Abstract. In this paper, we introduce a new topology of Fuzzy Polynomial Neural Networks (FPNN) that is based on a genetically optimized multilayer perceptron with fuzzy polynomial neurons (FPNs) and discuss its comprehensive design methodology involving mechanisms of genetic optimization, especially genetic algorithms (GAs). The proposed FPNN gives rise to a structurally optimized structure and comes with a substantial level of flexibility in comparison to the one we encounter in conventional FPNNs. The structural optimization is realized via GAs whereas in case of the parametric optimization we proceed with a standard least square method-based learning. Through the consecutive process of such structural and parametric optimization, an optimized and flexible fuzzy neural network is generated in a dynamic fashion. The performance of the proposed gFPNN is quantified through experimentation that exploits standard data already used in fuzzy modeling. These results reveal superiority of the proposed networks over the existing fuzzy and neural models.

1 Introduction

Recently, a lots of attention has been directed towards advanced techniques of complex system modeling. While neural networks, fuzzy sets and evolutionary computing as the technologies of Computational Intelligence (CI) have expanded and enriched a field of modeling quite immensely, they have also gave rise to a number of new methodological issues and increased our awareness about tradeoffs one has to make in system modeling [1-4]. The most successful approaches to hybridize fuzzy systems with learning and adaptation have been made in the realm of CI. Especially neural fuzzy systems and genetic fuzzy systems hybridize the approximate inference method of fuzzy systems with the learning capabilities of neural networks and evolutionary algorithms [5]. As one of the representative design approaches which are advanced tools, a family of fuzzy polynomial neuron (FPN)-based SOPNN(called “FPNN” as a

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 179–187, 2004.
© Springer-Verlag Berlin Heidelberg 2004

new category of neuro-fuzzy networks)[6] were introduced to build predictive models for such highly nonlinear systems. The FPNN algorithm exhibits some tendency to produce overly complex networks as well as a repetitive computation load by the trial and error method and/or the repetitive parameter adjustment by designer like in case of the original GMDH algorithm. In this study, in addressing the above problems with the conventional SOPNN (especially, FPN-based SOPNN called "FPNN" [6, 9]) as well as the GMDH algorithm, we introduce a new genetic design approach; as a consequence we will be referring to these networks as GA-based FPNN (to be called "gFPNN"). The determination of the optimal values of the parameters available within an individual FPN (viz. the number of input variables, the order of the polynomial, and input variables) leads to a structurally and parametrically optimized network.

2 The Architecture and Development of Fuzzy Polynomial Neural Networks (FPNN)

2.1 FPNN Based on Fuzzy Polynomial Neurons (FPNs)

The FPN consists of two basic functional modules. The first one, labeled by **F**, is a collection of fuzzy sets that form an interface between the input numeric variables and the processing part realized by the neuron. The second module (denoted here by **P**) is about the function – based nonlinear (polynomial) processing. This nonlinear processing involves some input variables. In other words, FPN realizes a family of multiple-input single-output rules. Each rule reads in the form

$$\text{If } x_p \text{ is } \boldsymbol{A}_l \text{ and } x_q \text{ is } \boldsymbol{B}_k \text{ then } z \text{ is } P_{lk}(x_i, x_j, \boldsymbol{a}_{lk}) \tag{1}$$

where $\boldsymbol{a}_{lk}$ is a vector of the parameters of the conclusion part of the rule while $\boldsymbol{P}_{lk}(x_i, x_j, \boldsymbol{a}_{lk})$ denotes the regression polynomial forming the consequence part of the fuzzy rule which uses several types of high-order polynomials besides the constant function forming the simplest version of the consequence; refer to Table 1. The activation levels of the rules contribute to the output of the FPN being computed as a weighted average of the individual condition parts (functional transformations) $\boldsymbol{P}_K$ (note that the index of the rule, namely *"K"* is a shorthand notation for the two indexes of fuzzy sets used in the rule (1), that is $K = (l, k)$).

$$z = \sum_{K=1}^{all\ rules} \mu_K P_K(x_i, x_j, \boldsymbol{a}_K) \Big/ \sum_{K=1}^{all\ rules} \mu_K = \sum_{K=1}^{all\ rules} \tilde{\mu}_K P_K(x_i, x_j, \boldsymbol{a}_K) \tag{2}$$

$$\tilde{\mu}_K = \frac{\mu_K}{\sum_{L=1}^{all\ rules} \mu_L} \tag{3}$$

Table 1. Different forms of the regression polynomials standing in the consequence part of the fuzzy

No. of inputs / Order of the polynomial	1	2	3
0 (Type 1)	Constant	Constant	Constant
1 (Type 2)	Linear	Bilinear	Trilinear
2 (Type 3)	Quadratic	Biquadratic-1	Triquadratic-1
2 (Type 4)		Biquadratic-2	Triquadratic-2

1: Basic type, 2: Modified type

2.2 Genetic Optimization of FPNN

GAs is a stochastic search technique based on the principles of evolution, natural selection, and genetic recombination by simulating "survival of the fittest" in a population of potential solutions(individuals) to the problem at hand [7]. For the optimization of the FPNN model, GA uses the serial method of binary type, roulette-wheel used in the selection process, one-point crossover in the crossover operation, and a binary inversion (complementation) operation in the mutation operator. To retain the best individual and carry it over to the next generation, we use elitist strategy [8].

3 The Algorithms and Design Procedure of Genetically Optimized FPNN

The framework of the design procedure of the Fuzzy Polynomial Neural Networks (FPNN) based on genetically optimized multi-layer perceptron architecture comprises the following steps.

[Step 1] *Determine system's input variables*

[Step 2] *Form training and testing data*

The input-output data set $(\boldsymbol{x}_i, y_i)=(x_{1i}, x_{2i}, \ldots, x_{ni}, y_i)$, $i=1,2,\ldots, N$ is divided into two parts, that is, a training and testing dataset.

[Step 3] *Decide initial information for constructing the FPNN structure*

[Step 4] *Decide FPN structure using genetic design*

When it comes to the organization of the chromosome representing (mapping) the structure of the FPNN, we divide the chromosome to be used for genetic optimization into three sub-chromosomes. The 1st sub-chromosome contains the number of input variables, the 2nd sub-chromosome involves the order of the polynomial of the node, and the 3rd sub-chromosome (remaining bits) contains input variables coming to the corresponding node (FPN). All these elements are optimized when running the GA.

[Step 5] *Carry out fuzzy inference and coefficient parameters estimation for fuzzy identification in the selected node (FPN)*

Regression polynomials (polynomial and in the specific case, a constant value) standing in the conclusion part of fuzzy rules are given as different types of Type 1, 2, 3, or 4, see Table 1. In each fuzzy inference, we consider two types of membership

functions, namely triangular and Gaussian-like membership functions. The consequence parameters are produced by the standard least squares method

[Step 6] *Select nodes (FPNs) with the best predictive capability and construct their corresponding layer*

The generation process can be organized as the following sequence of steps

Sub-step 1) We set up initial genetic information necessary for generation of the FPNN architecture.

Sub-step 2) The nodes (FPNs) are generated through the genetic design.

Sub-step 3) We calculate the fitness function. The fitness function reads as

$$F(\text{fitness function}) = 1/(1+EPI) \tag{4}$$

where *EPI* denotes the performance index for the testing data (or validation data).

Sub-step 4) To move on to the next generation, we carry out selection, crossover, and mutation operation using genetic initial information and the fitness values obtained via ***sub-step 3***.

Sub-step 5) We choose several FPNs characterized by the best fitness values. Here, we use the pre-defined number W of FPNs with better predictive capability that need to be preserved to assure an optimal operation at the next iteration of the FPNN algorithm. The outputs of the retained nodes (FPNs) serve as inputs to the next layer of the network. There are two cases as to the number of the retained FPNs, that is

(i) If $W^* < W$, then the number of the nodes retained for the next layer is equal to z.

Here, W^*denotes the number of the retained nodes in each layer that nodes with the duplicated fitness values are moved.

(ii) If $W^* \geq W$, then for the next layer, the number of the retained nodes is equal to W.

Sub-step 6) For the elitist strategy, we select the node that has the highest fitness value among the selected nodes *(W)*.

Sub-step 7) We generate new populations of the next generation using operators of GAs obtained from ***Sub-step 4***. We use the elitist strategy. This sub-step carries out by repeating ***sub-step 2-6***.

Sub-step 8) We combine the nodes (W populations) obtained in the previous generation with the nodes (W populations) obtained in the current generation.

Sub-step 9) Until the last generation, this sub-step carries out by repeating ***sub-step 7-8***.

[Step 7] *Check the termination criterion*

As far as the performance index is concerned (that reflects a numeric accuracy of the layers), a termination is straightforward and comes in the form,

$$F_1 \leq F_* \tag{5}$$

Where, F_1 denotes a maximal fitness value occurring at the current layer whereas F_* stands for a maximal fitness value that occurred at the previous layer. In this study, we use a measure (performance index) that is the Root Mean Squared Error (RMSE).

$$E(PI \ \ or \ \ EPI) = \sqrt{\frac{1}{N}\sum_{p=1}^{N}(y_p - \hat{y}_p)^2} \tag{6}$$

[Step 8] *Determine new input variables for the next layer*
If (5) has not been met, the model is expanded. The outputs of the preserved nodes (z_{1i}, z_{2i}, …, z_{wi}) serves as new inputs to the next layer (x_{1j}, x_{2j}, …, x_{wj})(j=i+1). This is captured by the expression

$$x_{1j} = z_{1i}, x_{2j} = z_{2i}, \ldots, x_{wj} = z_{wi} \tag{7}$$

The FPNN algorithm is carried out by repeating steps 4-8.

4 Experimental Studies

The performance of the GA-based FPNN is illustrated with the aid of well-known and widely used dataset of the chaotic Mackey-Glass time series [10-17].
The time series is generated by the chaotic Mackey-Glass differential delay equation comes in the form

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{8}$$

To obtain the time series value at each integer point, we applied the fourth-order Runge-Kutta method to find the numerical solution to (8). From the Mackey-Glass time series $x(t)$, we extracted 1000 input-output data pairs in the following format:

$$[x(t\text{-}18), x(t\text{-}12), x(t\text{-}6), x(t)\ ;\ x(t\text{+}6)]$$

where, t=118 to 1117. The first 500 pairs were used as the training data set while the remaining 500 pairs formed the testing data set. To come up with a quantitative evaluation of the network, we use the standard RMSE performance index as given by (6). Table 2 summarizes the list of parameters used in the genetic optimization of the network.

Table 2. Summary of the parameters of the genetic optimization

	Parameters	1st layer	2nd to 5th layer
GA	Maximum generation	100	100
	Total population size	60	60
	Selected population size (W)	30	30
	Crossover rate	0.65	0.65
	Mutation rate	0.1	0.1
	String length	3+3+30	3+3+30
FPNN	Maximal no.(Max) of inputs to be selected	$1 \le l \le$ Max(2~5)	$1 \le l \le$ Max(2~5)
	Polynomial type (Type T) of the consequent part of fuzzy rules	$1 \le$T≤ 4	$1 \le$T≤ 4
	Consequent input type to be used for Type T (*)	Type T*	Type T
	Membership Function (MF) type	Triangular	Triangular
		Gaussian	Gaussian
	No. of MFs per input	2	2

l, P, Max: integers, T* means that entire system inputs are used for the polynomial in the conclusion part of the rules.

Table 3 summarizes the performance of the 1st and the 2nd layer of the network when changing the maximal number of inputs to be selected; here Max was set up to 2 through 5.

Table 3. Performance index of the network of each layer versus the increase of maximal number of inputs to be selected

(a) Triangular MF

Max	1st layer					2nd layer				
	Node			T	PI	EPI	Node			T	PI	EPI
2	1	2		3	0.0056	0.0054	7	29		4	0.0050	0.0049
3	1	2	3	3	0.0040	0.0040	2	16	25	2	0.0033	0.0034
4	1	2	3	4	3	0.0017	0.0016	9	12	22	28	3	0.0015	0.0014
5	1	2	3	4	0	3	0.0017	0.0016	8	24	25	27	28	2	0.0014	0.0013

(b) Gaussian-like MF

1st layer						2nd layer					
Node			T	PI	EPI	Node			T	PI	EPI
1	2		3	0.0028	0.0027	19	26		3	0.0026	0.0025
1	2	4	3	0.0014	0.0013	15	25	27	4	0.0012	0.0011
1	2	3	4	3	0.0005	0.0006	10	23	24	29	2	0.0004	0.0006
1	2	3	4	0	3	0.0005	0.0006	7	12	18	22	29	4	0.0004	0.0006

Fig. 1 depicts the performance index of each layer of gFPNN according to the increase of maximal number of inputs to be selected. Fig. 2(a) illustrates the detailed optimal topologies of gFPNN for 1 layer when using Max=4 and triangular MF. And also Fig. 2(b) illustrates the detailed optimal topology of gFPNN for 1 layer in case of using Max= 3 and Gaussian-like MF.

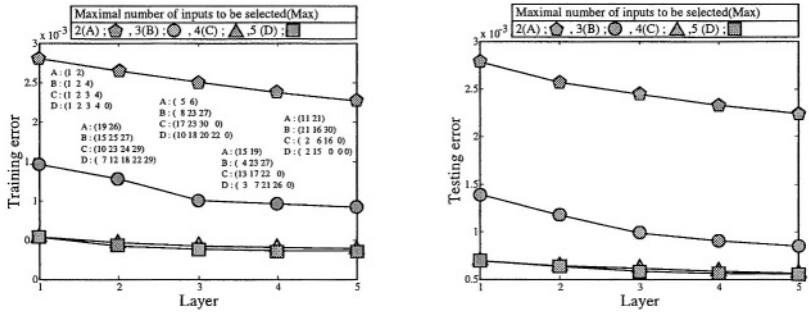
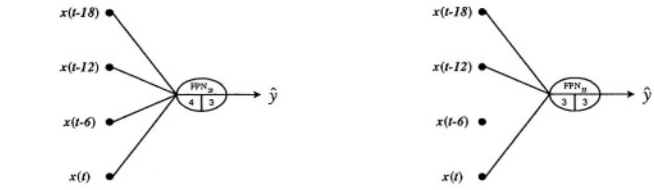


Fig. 1. Performance index of gFPNN with respect to the increase of number of layers (Gaussian-like MF)



(a) Triangular MF (1 layer) (b) Gaussian-like MF

Fig. 2. Genetically optimized FPNN(gFPNN) architecture

Table 4 gives a comparative summary of the network with other models.

Table 4. Comparative analysis of the performance of the network; considered are models reported in the literature

Model			Performance index			
			PI	PI_t	EPI_t	NDEI*
Wang's model[10]			0.044			
			0.013			
			0.010			
Cascaded-correlation NN[11]						0.06
Backpropagation MLP[11]						0.02
6th-order polynomial[11]						0.04
ANFIS[12]				0.0016	0.0015	0.007
FNN model[13]				0.014	0.009	
Recurrent neural network[14]			0.0138			
SONN**[15]	Basic (5^{th} layer)	Case 1		0.0011	0.0011	0.005
		Case 2		0.0027	0.0028	0.011
	Modified (5^{th} layer)	Case 1		0.0012	0.0011	0.005
		Case 2		0.0038	0.0038	0.016
Proposed gFPNN	Triangular	2^{nd} layer(Max=4)	0.0014	0.0015	0.0014	0.0061
		5^{th} layer(Max=4)	0.0010	0.0011	0.0011	0.0045
		2^{nd} layer(Max=5)	0.0011	0.0014	0.0013	0.0051
		5^{th} layer(Max=5)	0.0007	0.0009	0.0010	0.0031
	Gaussian	1^{st} layer(Max=3)	0.0014	0.0014	0.0013	0.0064
		1^{st} layer(Max=4)	0.0005	0.0005	0.0006	0.0023
		1^{st} layer(Max=5)	0.0005	0.0005	0.0006	0.0023

*Non-dimensional error index (NDEI) as used in [16] is defined as the root mean square errors divided by the standard deviation of the target series. ** is called "conventional optimized FPNN".

5 Concluding Remarks

In this study, the GA-based design procedure of Fuzzy Polynomial Neural Networks (FPNN) along with their architectural considerations has been investigated. The design methodology comes as a hybrid structural optimization and parametric learning being viewed as two fundamental phases of the design process. The GMDH method is now comprised of both a structural phase such as a self-organizing and an evolutionary algorithm (rooted in natural law of survival of the fittest), and the ensuing parametric phase of the Least Square Estimation (LSE)-based learning. The comprehensive experimental studies involving well-known datasets quantify a superb performance of the network in comparison to the existing fuzzy and neuro-fuzzy models. Most importantly, through the proposed framework of genetic optimization we can efficiently search for the optimal network architecture (structurally and parametrically optimized network) and this becomes crucial in improving the performance of the resulting model.

Acknowledgements. This work has been supported by EESRI(R-2003-B-274), which is funded by MOCIE (Ministry of Commerce, Industry and Energy).

References

1. V. Cherkassky, D. Gehring, F. Mulier.: Comparison of adaptive methods for function estimation from samples. IEEE Trans. Neural Networks, 7 (1996) 969-984
2. J. A. Dickerson, B. Kosko.: Fuzzy function approximation with ellipsoidal rules. IEEE Trans. Syst., Man, Cybernetics. Part B. 26 (1996) 542-560
3. V. Sommer, P. Tobias, D. Kohl, H. Sundgren, L. Lundstrom.: Neural networks and abductive networks for chemical sensor signals: A case comparison. Sensors and Actuators B. 28 (1995) 217-222
4. S. Kleinsteuber, N. Sepehri.: A polynomial network modeling approach to a class of large-scale hydraulic systems. Computers Elect. Eng. 22 (1996) 151-168
5. O. Cordon, et al.: Ten years of genetic fuzzy systems: current framework and new trends. Fuzzy Sets and Systems. 2003(in press)
6. S.-K. Oh, W. Pedrycz.: Self-organizing Polynomial Neural Networks Based on Polynomial and Fuzzy Polynomial Neurons: Analysis and Design. Fuzzy Sets and Systems. 142(2) (2003) 163-198
7. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
8. D. Jong, K. A.: Are Genetic Algorithms Function Optimizers?. Parallel Problem Solving from Nature 2, Manner, R. and Manderick, B. eds., North-Holland, Amsterdam
9. S.-K. Oh, W. Pedrycz.: Fuzzy Polynomial Neuron-Based Self-Organizing Neural Networks. Int. J. of General Systems, 32(2003) 237-250
10. L. X. Wang, J. M. Mendel.: Generating fuzzy rules from numerical data with applications. IEEE Trans. Systems, Man, Cybern. 22 (1992) 1414-1427
11. R. S. Crowder III.: Predicting the Mackey-Glass time series with cascade-correlation learning. In: D. Touretzky, G. Hinton, and T. Sejnowski, editors, Proceedings of the 1990 Connectionist Models Summer School, Carnegie Mellon University, (1990) 117-123
12. J. S. R. Jang.: ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Trans. System, Man, and Cybern. 23 (1993) 665-685
13. L. P. Maguire, B. Roche, T. M. McGinnity, L. J. McDaid.: Predicting a chaotic time series using a fuzzy neural network. Information Sciences. 112(1998) 125-136
14. C. James Li, T. -Y. Huang.: Automatic structure and parameter training methods for modeling of mechanical systems by recurrent neural networks. Applied Mathematical Modeling. 23 (1999) 933-944
15. S.-K. Oh, W. Pedrycz, T.-C. Ahn.: Self-organizing neural networks with fuzzy polynomial neurons. Applied Soft Computing. 2 (2002) 1-10
16. A. S. Lapedes, R. Farber.: Non-linear Signal Processing Using Neural Networks: Prediction and System Modeling. Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, (1987)
17. M. C. Mackey, L. Glass.: Oscillation and chaos in physiological control systems. Science, 197 (1977) 287-289
18. S.-K. Oh, W. Pedrycz.: The design of self-organizing Polynomial Neural Networks. Information Science. 141 (2002) 237-258

19. S.-K. Oh, W. Pedrycz, B.-J. Park.: Polynomial Neural Networks Architecture: Analysis and Design. Computers and Electrical Engineering. 29 (2003) 703-725
20. B.-J. Park, D.-Y. Lee, S.-K. Oh: Rule-Based Fuzzy Polynomial Neural Networks in Modeling Software Process Data. International Journal of Control, Automation and Systems, 1(3) (2003) 321-331
21. H.-S. Park, S.-K. Oh: Rule-based Fuzzy-Neural Networks Using the Identification Algorithm of GA hybrid Scheme. International Journal of Control, Automation and Systems, 1(1) (2003) 101-110

Information Granulation-Based Multi-layer Hybrid Fuzzy Neural Networks: Analysis and Design

Byoung-Jun Park[1], Sung-Kwun Oh[1], Witold Pedrycz[2], and Tae-Chon Ahn[1]

[1] School of Electrical, Electronic and Information Engineering,
Wonkwang University, Korea
{lcap, ohsk, tcahn}@wonkwang.ac.kr
[2] Department of Electrical and Computer Engineering, University of Alberta,
Edmonton, AB T6G 2G6, Canada
and Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
pedrycz@ee.ualberta.ca

Abstract. In this study, a new architecture and comprehensive design methodology of genetically optimized Hybrid Fuzzy Neural Networks (gHFNN) are introduced and a series of numeric experiments are carried out. The gHFNN architecture results from a synergistic usage of the hybrid system generated by combining Fuzzy Neural Networks (FNN) with Polynomial Neural Networks (PNN). FNN contributes to the formation of the premise part of the overall network structure of the gHFNN. The consequence part of the gHFNN is designed using PNN.

1 Introductory Remarks

Efficient modeling techniques should allow for a selection of pertinent variables and a formation of highly representative datasets. The models should be able to take advantage of the existing domain knowledge and augment it by available numeric data to form a coherent data-knowledge modeling entity. The omnipresent modeling tendency is the one that exploits techniques of Computational Intelligence (CI) by embracing fuzzy modeling [1-6], neurocomputing [7], and genetic optimization [8].

In this study, we develop a hybrid modeling architecture, called genetically optimized Hybrid Fuzzy Neural Networks (gHFNN). In a nutshell, gHFNN is composed of two main substructures driven to genetic optimization, namely a fuzzy set-based fuzzy neural network (FNN) and a polynomial neural network (PNN). From a standpoint of rule-based architectures, one can regard the FNN as an implementation of the antecedent part of the rules while the consequent is realized with the aid of a PNN. The role of the FNN is to interact with input data, granulate the corresponding input spaces. In the first case (Scheme I) we concentrate on the use of simplified fuzzy inference. In the second case (Scheme II), we take advantage of linear fuzzy inference. The role of the PNN is to carry out nonlinear transformation at the level of the fuzzy sets formed at the level of FNN. The PNN that exhibits a flexible and versatile structure [9] is constructed on a basis of Group Method of Data Handling (GMDH [10])

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 188–195,2004.
© Springer-Verlag Berlin Heidelberg 2004

method and genetic algorithms (GAs). The design procedure applied in the construction of each layer of the PNN deals with its structural optimization involving the selection of optimal nodes (polynomial neurons; PNs) with specific local characteristics (such as the number of input variables, the order of the polynomial, and a collection of the specific subset of input variables) and addresses specific aspects of parametric optimization. To assess the performance of the proposed model, we exploit a well-known time series data. Furthermore, the network is directly contrasted with several existing intelligent models.

2 Conventional Hybrid Fuzzy Neural Networks (HFNN)

The architectures of conventional HFNN [11,12] result as a synergy between two other general constructs such as FNN and PNN. Based on the different PNN topologies, the HFNN distinguish between two kinds of architectures, namely basic and modified architectures. Moreover, for the each architecture we identify two cases. In the connection point, if input variables to PNN used on the consequence part of HFNN are less than three (or four), the generic type of HFNN does not generate a highly versatile structure. Accordingly we identify also two types as the generic and advanced. The topologies of the HFNN depend on those of the PNN used for the consequence part of HFNN. The design of the PNN proceeds further and involves a generation of some additional layers. Each layer consists of nodes (PNs) for which the number of input variables could the same as in the previous layers or may differ across the network. The structure of the PNN is selected on the basis of the number of input variables and the order of the polynomial occurring in each layer.

3 The Architecture and Development of Genetically Optimized HFNN (gHFNN)

The gHFNN emerges from the genetically optimized multi-layer perceptron architecture based on fuzzy set-based FNN, GAs and GMDH. These networks result as a synergy between two other general constructs such as FNN [13] and PNN [9].

3.1 Fuzzy Neural Networks and Genetic Optimization

We use FNN based on two types of fuzzy inferences, that is, simplified (Scheme I) and linear fuzzy inference-based FNN (Scheme II) as shown in Fig. 1. The notation used in Fig. 1 requires some clarification. The "circles" denote units of the FNN while "N" identifies a normalization procedure applied to the membership grades of the input variable x_i. The output of the "$\sum$" neuron is described by a nonlinear function $f_i(x_i)$. Finally, the output of the FNN is governed by the following expression.

$$\hat{y} = f_1(x_1) + f_2(x_2) + \cdots + f_m(x_m) = \sum_{i=1}^{m} f_i(x_i) \tag{1}$$

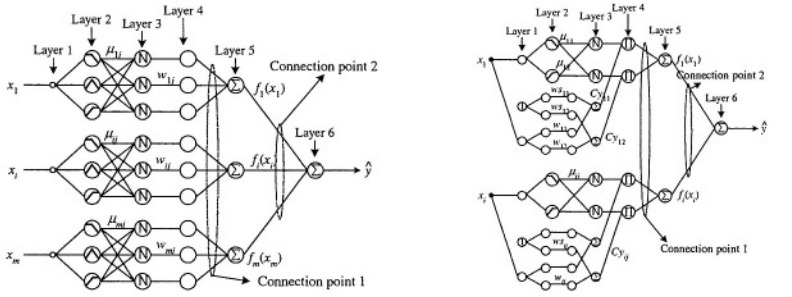


Fig. 1. Topologies of fuzzy set-based FNN

We can regard each $f_i(_{xi})$ given by (1) as the following mappings (rules).

$$\text{Scheme I} - R^j : \textit{If } x_i \textit{ is } A_{ij} \textit{ then } Cy_{ij}=w_{ij} \tag{2}$$

$$\text{Scheme II} - R^j : \textit{If } x_i \textit{ is } A_{ij} \textit{ then } Cy_{ij}=ws_{ij} + w_{ij}\, x_i\, R_j \tag{3}$$

R^j is the *j-th* fuzzy rule while A_{ij} denotes a fuzzy variable of the premise of the fuzzy rule and represents a membership function μ_{ij}. w_{ij} is a constant in (2), and ws_{ij} is a constant and w_{ij} is an input variable consequence of the fuzzy rule in (3). They express a connection existing between the neurons as visualized in Fig. 1. Mapping from x_i to $f_i(x_i)$ in (2) is determined by the fuzzy inferences and a standard defuzzification.

$$f_i(x_i) = \sum_{j=1}^{z} \mu_{ij}(x_i) \cdot w_{ij} \Big/ \sum_{j=1}^{z} \mu_{ij}(x_i) \tag{4}$$

The learning of FNN is realized by adjusting connections of the neurons and as such it follows a standard Back-Propagation (BP) algorithm [14]. For the simplified fuzzy inference-based FNN, the update formula of a connection in Scheme I is as follow.

$$\Delta w_{ij} = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \mu_{ij}(x_i) + \alpha(w_{ij}(t) - w_{ij}(t-1)) \tag{5}$$

Where, y_p is the *p-th* target output data, $\hat{y}_p$ stands for the *p-th* actual output of the model for this specific data point, η is a positive learning rate and α is a momentum coefficient constrained to the unit interval. The inference result and the learning algorithm in linear fuzzy inference-based FNN use the mechanisms in the same manner as discussed above.

Genetic algorithms (GAs) are optimization techniques based on the principles of natural evolution. In essence, they are search algorithms that use operations found in natural genetics to guide a comprehensive search over the parameter space [8]. In order to enhance the learning of the FNN and augment its performance of a FNN, we use GAs to adjust learning rate, momentum coefficient and the parameters of the membership functions of the antecedents of the rules.

3.2 Genetically Optimized PNN (gPNN)

When we construct PNs of each layer in the conventional PNN [9], such parameters as the number of input variables (nodes), the order of polynomial, and input variables available within a PN are fixed (selected) in advance by the designer. This could have

frequently contributed to the difficulties in the design of the optimal network. To overcome this apparent drawback, we introduce a new genetic design approach; especially as a consequence we will be referring to these networks as genetically optimized PNN (to be called "gPNN"). The overall genetically-driven optimization process of PNN is shown in Fig. 2.

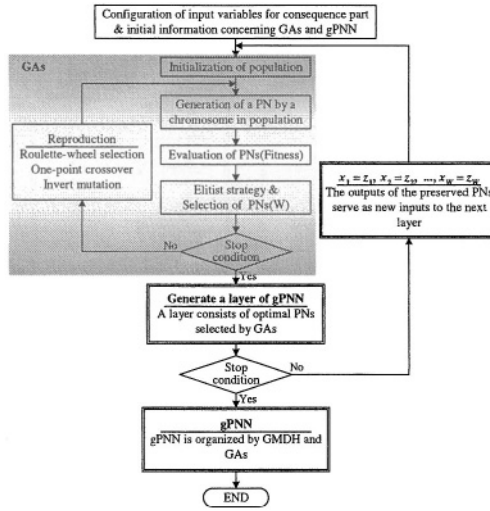


Fig. 2. Overall genetically-driven optimization process of PNN

4 The Algorithms and Design Procedure of gHFNN

The premise of gHFNN: FNN *(Refer to Fig. 1)*

[Layer 1] Input layer.

[Layer 2] Computing activation degrees of linguistic labels.

[Layer 3] Normalization of a degree activation (firing) of the rule.

[Layer 4] Multiplying a normalized activation degree of the rule by connection. If we choose Connection point 1 for combining FNN with gPNN as shown in Fig. 1, a_{ij} is given as the input variable of the gPNN.

$$a_{ij} = \overline{\mu}_{ij} \times Cy_{ij} = \mu_{ij} \times Cy_{ij} \tag{6}$$

$$\begin{cases} \text{Simplified} : Cy_{ij} = w_{ij} \\ \text{Linear} \quad : Cy_{ij} = ws_{ij} + w_{ij} \cdot x_i \end{cases} \tag{7}$$

[Layer 5] Fuzzy inference for the fuzzy rules. If we choose Connection point 2, f_i is the input variable of gPNN.

[Layer 6; Output layer of FNN] Computing output of a FNN.

The consequence of gHFNN: gPNN *(Refer to Fig. 2)*

[Step 1] Configuration of input variables.

If we choose the first option (Connection point 1), $x_1=a_{11}$, $x_2=a_{12}$,$\cdots$, $x_n=a_{ij}$ $(n=i\times j)$. For the second option (Connection point 2), we have $x_1=f_1$, $x_2=f_2$,$\cdots$, $x_n=f_m$ $(n=m)$.

[Step 2] Decision of initial information for constructing the gPNN.
[Step 3] Initialization of population.
[Step 4] Decision of PNs structure using genetic design. We divide the chromosome to be used for genetic optimization into three sub-chromosomes as shown in Fig. 4(a).

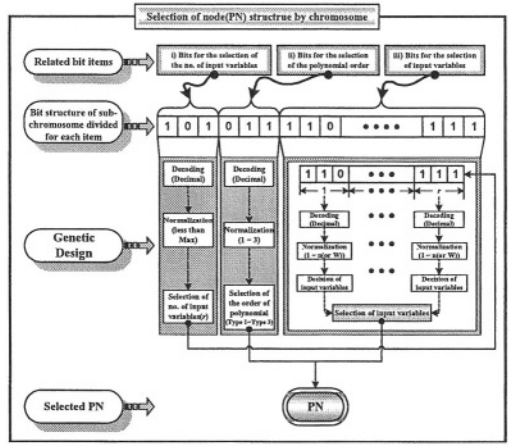


Fig. 3. The PN design using genetic optimization

Table 1. Different forms of regression polynomial forming a PN

Number of inputs / Order of the polynomial	2	3	4
1 (Type 1)	Bilinear	Trilinear	Tetralinear
2 (Type 2)	Biquadratic-1	Triquadratic-1	Tetraquadratic-1
2 (Type 3)	Biquadratic-2	Triquadratic-2	Tetraquadratic-2

[Step 5] Evaluation of PNs.
[Step 6] Elitist strategy and selection of PNs with the best predictive capability.
[Step 7] Reproduction.
[Step 8] Repeating Step 4-7.
[Step 9] Construction of their corresponding layer.
[Step 10] Check the termination criterion (performance index).

$$E(PI \text{ or } EPI) = \frac{1}{n}\sum_{p=1}^{n}(y_p - \hat{y}_p)^2 \tag{8}$$

[Step 11] Determining new input variables for the next layer.
The gPNN algorithm is carried out by repeating Steps 4-11.

5 Experimental Studies

The performance of the gHFNN is illustrated with the aid of a time series of gas furnace [14]. The delayed terms of methane gas flow rate, *u(t)* and carbon dioxide density, *y(t)* are used as system input variables. We utilizes 3 system input variables such as *u(t-2), y(t-2),* and *y* (t-1). The output variable is *y(t).*

Table 2. Performance index of gHFNN for the gas furnace

Premise part				CP	Consequence part							PI	EPI
Fuzzy Inference	No. of rules (MFs)	PI	EPI		Layer	No. of inputs	Input No.				T		
Simplified	6 (2+2+2)	0.0248	0.126	01	1	4	5	2	1	3	3	0.0220	0.135
					2	4	15	28	26	3	2	0.0209	0.135
					3	4	5	1	25	16	3	0.0205	0.131
					4	4	1	2	28	26	2	0.0190	0.128
					5	4	18	6	28	1	2	0.0175	0.125
				02	1	3	2	3	2	·	3	0.0221	0.135
					2	4	11	2	15	13	2	0.0194	0.126
					3	4	28	2	11	17	2	0.0190	0.116
					4	4	2	6	23	27	1	0.0188	0.114
					5	4	1	12	7	26	3	0.0182	0.112
Linear	6 (2+2+2)	0.0256	0.143	01	1	4	6	3	1	5	3	0.0218	0.136
					2	4	6	24	16	30	3	0.0197	0.124
					3	3	4	16	26	·	1	0.0196	0.121
					4	4	22	24	1	13	1	0.0193	0.119
					5	3	11	18	21	·	3	0.0191	0.117
				02	1	3	1	2	3	·	3	0.0232	0.130
					2	4	12	15	13	6	2	0.0196	0.120
					3	2	19	30	·	·	2	0.0194	0.115
					4	4	2	21	11	5	1	0.0188	0.113
					5	4	13	3	26	25	1	0.0184	0.110

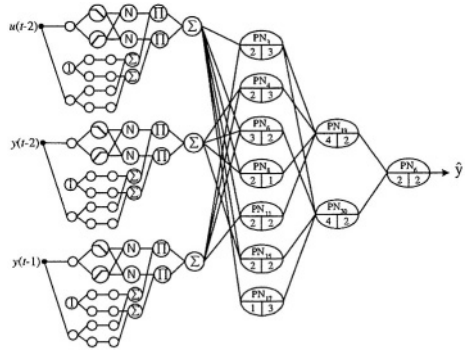


Fig. 4. Optimal topology of genetically optimized HFNN for the gas furnace (in case of linear fuzzy inference)

Table 2 summarizes the results of the optimized architectures according to connection points based on each fuzzy inference method. In Table 2, the values of the performance index of output of the gHFNN depend on each connection point based on the individual fuzzy inference methods. The optimal topology of gHFNN is shown in Fig. 4. Fig. 5 illustrates the optimization process by visualizing the performance index in successive cycles. It also shows the optimized network architecture when taking into consideration gHFNN based on linear fuzzy inference and connection point (CP) 2, refer to Table 2. Table 3 contrasts the performance of the genetically developed network with other fuzzy and fuzzy-neural networks studied in the literatures.

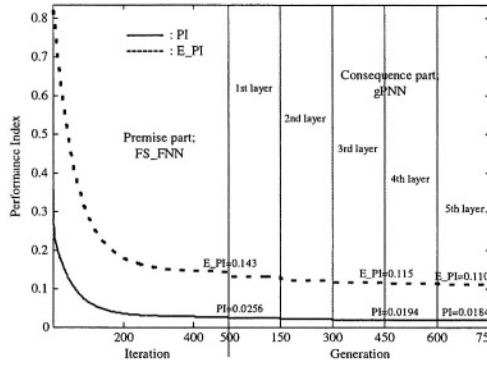


Fig. 5. Optimization procedure of gHFNN by BP learning and GAs

Table 3. Comparison of performance with other modeling methods

Model			PI	EPI	No. of rules
Box and Jenkin's model [14]			0.710		
Pedrycz's model [1]			0.320		
Xu and Zailu's model [2]			0.328		
Sugeno and Yasukawa's model [3]			0190		
Kim, et al.'s model [15]			0.034	0.244	2
Lin and Cunningham's mode [16]			0.071	0.261	4
Fuzzy	Complex [4]	Simplified	0.024	0.328	4(2×2)
		Linear	0.023	0.306	4(2×2)
	Hybrid [6] (GAs+Complex)	Simplified	0.024	0.329	4(2×2)
		Linear	0.017	0.289	4(2×2)
	HCM+GAs [5]	Simplified	0.022	0.333	6(3×2)
		Linear	0.020	0.264	6(3×2)
FNN [13]		Simplified	0.043	0.264	6(3+3)
		Linear	0.037	0.273	6(3+3)
SOFPNN		Generic [11]	0.017	0.250	4 rules/5th layer
		Advanced [12]	0.019	0.264	6 rules/5th layer
Proposed model (gHFNN)		Simplified	0.018	0.254	4 rules/5th layer
			0.018	0.112	6 rules/5th layer
		Linear	0.015	0.258	4 rules/5th layer
			0.018	0.110	6 rules/5th layer

6 Concluding Remarks

The comprehensive design methodology comes with the parametrically as well as structurally optimized network architecture. 1) As the premise structure of the gHFNN, the optimization of the rule-based FNN hinges on genetic algorithms and back-propagation (BP) learning algorithm: The GAs leads to the auto-tuning of vertexes of membership function, while the BP algorithm helps obtain optimal parameters of the consequent polynomial of fuzzy rules through learning. And 2) the gPNN that is the consequent structure of the gHFNN is based on the technologies of the extended GMDH and GAs: The extended GMDH is comprised of both a structural phase such as a self-organizing and evolutionary algorithm, and a parametric phase of

least square estimation-based learning, moreover the gPNN architecture is driven to genetic optimization, in what follows it leads to the selection of the optimal nodes. In the sequel, a variety of architectures of the proposed gHFNN driven to genetic optimization have been discussed. The experiments helped compare the network with other intelligent models – in all cases the previous models came with higher values of the performance index.

Acknowledgement. This work was supported by Korea Research Foundation Grant (KRF-2003-002-D00297)

References

1. W. Pedrycz: An identification algorithm in fuzzy relational system. Fuzzy Sets and Systems, 13 (1984) 153-167
2. C. W. Xu, Y. Zailu: Fuzzy model identification self-learning for dynamic system. IEEE Trans. on Syst. Man, Cybern., SMC-17(4) (1987) 683-689
3. M. Sugeno, T. Yasukawa: A Fuzzy-Logic-Based Approach to Qualitative Modeling. IEEE Trans. Fuzzy Systems, 1(1) (1993) 7-31
4. S.-K. Oh, W. Pedrycz: Fuzzy Identification by Means of Auto-Tuning Algorithm and Its Application to Nonlinear Systems. Fuzzy Sets and Systems, 115(2) (2000) 205-230,.
5. B.-J. Park, W. Pedrycz, S.-K. Oh: Identification of Fuzzy Models with the Aid of Evolutionary Data Granulation. IEE Proceedings-Control theory and application, 148(5) (2001) 406-418
6. S.-K. Oh, W. Pedrycz, B.-J. Park: Hybrid Identification of Fuzzy Rule-Based Models. International Journal of Intelligent Systems, 17(1) (2002) 77-103
7. K. S. Narendra, K. Parthasarathy: Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks. IEEE Transactions on Neural Networks, 2 (1991) 252-262
8. Z. Michalewicz: Genetic Algorithms + Data Structures = Evolution Programs. (1996) Springer-Verlag, Berlin Heidelberg
9. S.-K. Oh, W. Pedrycz, B.-J. Park: Polynomial Neural Networks Architecture: Analysis and Design. Computers and Electrical Engineering, 29(6) (2003) 653-725
10. A. G. Ivahnenko: The group method of data handling: a rival of method of stochastic approximation. Soviet Automatic Control, 13(3) (1968) 43-55
11. B.-J. Park, D.-Y. Lee, S.-K. Oh: Rule-Based Fuzzy Polynomial Neural Networks in Modeling Software Process Data. International Journal of Control, Automation and Systems, 1(3) (2003) 321-331
12. H.-S. Park, S.-K. Oh: Rule-based Fuzzy-Neural Networks Using the Identification Algorithm of GA hybrid Scheme. International Journal of Control, Automation and Systems, 1(1) (2003) 101-110
13. S.-K. Oh, W. Pedrycz, H.-S. Park: Hybrid Identification in Fuzzy-Neural Networks. Fuzzy Sets and Systems, 138(2) (2003) 399-426
14. D. E. P. Box, G. M. Jenkins: Time Series Analysis, Forecasting, and Control, 2nd edition (1976) Holden-Day, SanFrancisco
15. E. Kim, H. Lee, M. Park, M. Park: A Simply Identified Sugeno-type Fuzzy Model via Double Clustering. Information Sciences, 110 (1998) 25-39
16. Y. Lin, G. A. Cunningham III: A new Approach to Fuzzy-neural Modeling. IEEE Transaction on Fuzzy Systems, 3(2) 190-197

Efficient Learning of Contextual Mappings by Context-Dependent Neural Nets

Piotr Ciskowski

Wroclaw University of Technology
Department of Electronics
Institute of Engineering Cybernetics
Wybrzeze Wyspianskiego 27
Wroclaw 50-370, Poland
Piotr.Ciskowski@pwr.wroc.pl

Abstract. The paper addresses the problem of using contextual information by neural nets solving problems of contextual nature. The model of a context-dependent neuron is unique in the fact that allows weights to change according to some contextual variables even after the learning process has been completed. The structures of context-dependent neural nets are outlined, the Vapnik-Chervonenkis dimension of the single context-dependent neuron and multilayer net shortly discussed, decision boundaries are analyzed and compared with the traditional nets. The main goal of the article is to present highly effective contextual training algorithms for the considered models of neural nets.

1 Introduction

The phenomena and objects in real world can rarely be considered in isolation. The parameters describing them often change according to some environmental conditions, called the context. A learning machine, dealing with such systems should take the information about the context into consideration in order to improve its performance obtained using only the primary features of the analyzed object. The paper presents a novel way of context-sensitive learning for the neural network approach. A brief discussion will be presented on how context-dependent neural nets (CD-nets) use contextual data (analysis of various contextual approaches towards machine learning may be found in [1], the definitions of contextual variables and methods of managing contextual information may be found in [2,3]).

Many algorithms have been developed for improving the training process of neural nets, their generalization abilities or optimizing the nets' architecture. However, few of them intrude the neuron's model itself in order to enrich its processing abilities. Although neural nets are proved to be able to perform any input-output mapping with the desired accuracy, provided they are complex enough and given enough training data, the need is clear for designing less complex networks and more appropriate for the problems to be solved.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 196–203, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The model of a context-dependent neural net, introduced in [4,5], developed in [1], and presented in this paper proves to be useful for applications of neural nets in problems showing contextual dependencies among data. It is based on the ideas of contextual learning of nonlinear mappings, presented in [6], [7], [8] and [9], and applied for tasks of robot arm control and classification of ECG signals. It is a generalization of the traditional neuron's model, however, as shown in the paper:

1. more appropriate for contextual learning,
2. of higher Vapnik-Chervonenkis dimension (better processing abilities),
3. using its abilities more efficiently than just considering the growth with the number of parameters,
4. using the Kronecker product of the vectors of primary inputs and contextual basis functions, what simplifies the standard training algorithms and allows developing highly effective contextual learning algorithms.

Generally speaking we define the **context** as *all the factors that should influence (improve) the decision making* (or other kind of information processing) performed by the learning algorithm, *but other from the data on which the decision is taken* (or which are directly processed). Contrary to the traditional models of neural nets, the contextual approach makes distinction between these two groups of features and takes contextual dependencies between them into consideration in order to improve the process of machine learning. The division of features into these two groups is a separate problem and subject of research, not covered in this paper.

2 Models of Context-Dependent Nets

2.1 Model of a Context Dependent Neuron

Unlike the traditional neural net, the CD-net does not transform the primary and contextual features equally. It groups primary and context-sensitive inputs in the primary input vector $\overline{\mathbf{X}}$, operates on them similarly as the traditional net, although making the weights to the primary inputs functionally dependent on the vector of contextual inputs $\overline{\mathbf{Z}}$. Consider a neuron model of the form

$$y = \Phi\left[w_0\left(\overline{\mathbf{Z}}\right) + \sum_{s=1}^{S} w_s\left(\overline{\mathbf{Z}}\right) x_s\right] \tag{1}$$

where x_s denotes the s-th primary input, y is the neuron's output, $\overline{\mathbf{Z}}$ denotes the vector of contextual variables, while Φ is the activation function. The neuron's weights to the primary inputs, and the offset, depend on the vector of contextual variables as follows:

$$w_s = w_s\left(\overline{\mathbf{Z}}\right) = \sum_{m=1}^{M} a_{s,m} v_m\left(\overline{\mathbf{Z}}\right) = \overline{\mathbf{A}}_s^T \overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right) \tag{2}$$

$s = 0, 1, \ldots, S$, where $\overline{\mathbf{V}}(\overline{\mathbf{Z}})$ is the vector of independent basis functions, common for all the net, spanning the weights' dependence on the context, and $\overline{\mathbf{A}}_s$ is the vector of coefficients for weight $w_s(\overline{\mathbf{Z}})$. The model is presented in fig. 1, where also the Kronecker product of primary input vector and the vector of basis functions of contextual inputs is shown.

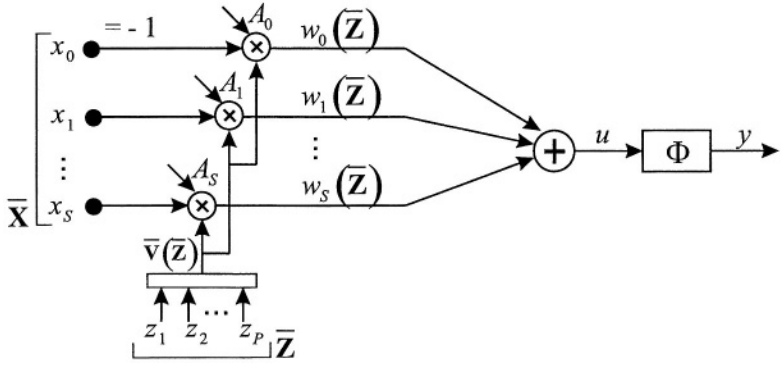


Fig. 1. Context-dependent neuron

2.2 Architectures of Context-Dependent Nets

A neural net may be used to solve a problem characterized both by primary and context-sensitive features. The former are useful for solving the task even when considered in isolation, regardless of their context. The latter require adapting the way of processing them to the contextual data. Therefore one may think of a hybrid net, in which some weights (for connections to context-sensitive inputs) are context-dependent and others (for connections to primary inputs) are traditional (constant after training, that is the same in all the contexts). If some additional knowledge of the problem being solved suggests using traditional weights, the net's designer may simply set some of the basis functions or coefficients for these weights to zero. It is also advisable to analyze the weights' change throughout the contexts when the learning process is completed in order to recognize the weights which should be constant. Different structures of hybrid nets are presented in [1].

2.3 Context-Dependent Feedforward Net

As in traditional nets, one can build more complicated structures made of single context-dependent neurons, particularly the common feedforward net. The parameters of context-dependent nets are stored in the vectors of coefficients for

each weight. It is advisable to construct the weight matrix for the whole layer of neurons in the following way:

$$\mathbf{A} = \left[\overline{\mathbf{A}}_1\ \overline{\mathbf{A}}_2 \cdots \overline{\mathbf{A}}_K\right] = \begin{bmatrix} \overline{\mathbf{A}}_{1,0} & \overline{\mathbf{A}}_{2,0} & & \overline{\mathbf{A}}_{K,0} \\ \overline{\mathbf{A}}_{1,1} & \overline{\mathbf{A}}_{2,1} & \cdots & \overline{\mathbf{A}}_{K,1} \\ \vdots & \vdots & & \vdots \\ \overline{\mathbf{A}}_{1,S} & \overline{\mathbf{A}}_{2,S} & & \overline{\mathbf{A}}_{K,S} \end{bmatrix}_{(S+1)M \times K} \tag{3}$$

where $\overline{\mathbf{A}}_k$ is the vector of coefficients for the k-th neuron, built of vectors $\overline{\mathbf{A}}_{k,s}$ of coefficients for the weight $w_{k,s}\left(\overline{\mathbf{Z}}\right)$ (that is k-th neuron's weight to the s-th input). Such an organization of the net's parameters makes it possible to write the output of the net's layer $\overline{\mathbf{Y}}$ in a clear form as:

$$\overline{\mathbf{Y}}\left(\overline{\mathbf{X}}, \overline{\mathbf{Z}}\right) = \Phi\left\{\overline{\mathbf{A}}^T \cdot \left[\overline{\mathbf{X}} \otimes \overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right)\right]\right\} \tag{4}$$

3 Properties of Context-Dependent Nets

The properties of traditional and context-dependent nets shall be compared in this section. The Vapnik-Chervonenkis dimension of both types of learning machines and the discriminating boundaries produced by them in their input space will be considered.

3.1 The Vapnik-Chervonenkis Dimension of CD-Nets

The Vapnik-Chervonenkis dimension is one of the quantities used in machine learning theory for estimating their generalization abilities and the effectiveness of learning. The idea of the growth function and VC-dimension is well presented in [10]. The parallel theory of the VC-dimension for context-dependent nets is developed in [1]. Here we shall present the main results and conclusions.

The VC-dimension of a traditional perceptron using S primary inputs is equal to the number of its adjustable parameters, that is $S + 1$.

In non-contextual approach the traditional perceptron is supplied with only primary features of the data (it is common for the donors of many datasets to exclude contextual information as irrelevant). Then a neuron with S inputs is able to dichotomize $S + 1$ points in its S-dimensional input space. If we also supply the neuron with information about the context of these points (encoded on P inputs and added to the S primary inputs already being used), then the neuron is able to dichotomize $S + P + 1$ points in the joint input space. In both cases the neuron's parameter space is of the same dimensionality as its input space.

The Vapnik-Chervonenkis dimension of a real weight context-dependent perceptron with $S \in N$ real primary inputs, using the basis function vector $\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right)$ consisting of M independent basis functions $v_m\left(\overline{\mathbf{Z}}\right)$, $m = 1, 2, \ldots, M$, is given by

$$\mathrm{VCdim}\left(H^{\mathrm{cont}}\right) = M\left(S + 1\right) \tag{5}$$

This result follows from the fact that for the above context-dependent neuron it is possible to find $M(S + 1)$ (at most $S + 1$ in the same context, that is with the same contextual vector $\overline{\mathbf{Z}}$) points in its input space, for which the neuron is able to perform all the dichotomies.

3.2 Separating Power of Traditional and CD-Nets

The separating abilities of both traditional and context-dependent neurons are strictly related to their parameter space. The dimensionality of the traditional neuron's parameter space is equal to the number of weights W. Analogically, for the context-dependent neuron it is equal to the number of coefficients $A = MW$.

The context-dependent neuron produces a discriminating hyperplane in its A-dimensional, that is $M(S + 1)$-dimensional *parameter* space. This hyperplane is in fact a hypersurface in the $S + P + 1$-dimensional *input* space. This leads to much more adjustable decision boundaries which context-dependent multilayer nets are able to produce in the joint input space. Therefore they are more powerful in classification tasks of contextual nature. Examples of discriminating hypersurfaces in a joint input space (including two primary and one contextual features) are presented in fig. 2. The hypersurfaces are lines for a fixed value of the contextual variable, as context-dependent neuron is a generalization of the traditional one and in the fixed context has the same properties. These discriminating lines may however much more flexibly change from one context to another.

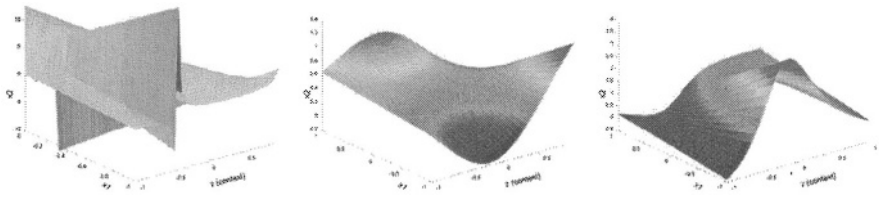


Fig. 2. Examples of discriminating hypersurfaces of context-dependent nets

The discriminating boundaries of traditional neurons are only hyperplanes in the joint input space. Nonlinear boundaries, more complicated in the contextual domain, must be modeled by a mixture of hyperplanes, that is by more neurons in the hidden layer of multilayer traditional net. Therefore both the decision boundaries, as well as their change through the contexts, have to be modeled by hidden layer's neurons of a traditional net. In context-dependent multilayer nets these two tasks are divided between the primary structure of the net (the weights) and the contextual structure (dependence of weights on the context).

Another advantage of context-dependent nets is the fact that contextual information modifies the weights of all the layers, modifying the behavior of the whole net according to the context. In traditional ones contextual data is supplied only to the first layer.

The Vapnik-Chervonenkis dimension of multilayer CD-nets grows similarly to traditional nets, that is with the number of net's parameters. However, the separating abilities of context-dependent nets are more evenly distributed between different contexts than for a traditional net of the same VC-dimension. The net's designer may also increase the VC-dimension by adding more basis functions. Not the growth of the VC-dimension should however be emphasized, but the effective use of the parameters space in order to obtain good generalization abilities of context-dependent nets in all the contexts.

As the interest in improving the learning machines performance by using contextual information grows, the models of context-dependent neural nets prove to be effective in solving tasks with contextual dependencies between the input data. An example of using a context-dependent neural net to modeling a highly nonlinear magnetorheological damper may be found in [11].

4 Training Context-Dependent Nets

Two algorithms for training multi-layer feedforward context-dependent nets are presented in this section: the backpropagation rule and contextual version of the Levenberg-Marquardt algorithm based on calculation of the error function's Hessian. A more thorough analysis of neural nets' training algorithms may be found in [1].

4.1 Context-Dependent Training with Backpropagation

Due to the appropriate construction of the net's coefficient matrix, the gradient based training algorithms are clear and their computational complexity is comparable to the one of traditional nets. For the net's output given by (4) the gradient (with respect to the coefficient matrix) of the following error function

$$E_{(\overline{\mathbf{X}},\overline{\mathbf{Z}},\overline{\mathbf{Y}})}\left[\Phi^{-1}\left(\overline{\mathbf{Y}}\right)-\mathbf{W}^{T}\left(\overline{\mathbf{Z}}\right)\cdot\overline{\mathbf{X}}\right]^{2}=E_{(\overline{\mathbf{X}},\overline{\mathbf{Z}},\overline{\mathbf{Y}})}\left\{\Psi\left(\overline{\mathbf{Y}}\right)-\mathbf{A}^{\mathbf{T}}\cdot\left[\overline{\mathbf{X}}\otimes\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right)\right]\right\}^{2} \tag{6}$$

where E is the expectation with respect to random variables specified below this operator, is given by

$$\nabla_{\mathbf{A}}E_{(\overline{\mathbf{X}},\overline{\mathbf{Z}},\overline{\mathbf{Y}})}=-\left\{\Psi\left(\overline{\mathbf{Y}}\right)-\mathbf{A}^{\mathbf{T}}\cdot\left[\overline{\mathbf{X}}\otimes\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right)\right]\right\}\cdot\left[\overline{\mathbf{X}}\otimes\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right)\right] \tag{7}$$

The backpropagation algorithm of training multi-layer context-dependent nets is similar to the one for traditional nets. It is derived and analyzed in detail in [1]. Here we shall give the final formula for the standard external error function's gradient (actual for the output layer's neurons and estimated for the hidden layers' neurons). The gradient is given as a vector for all neurons in the l-th

layer of the net (gradient of the error function E with respect to the l-th matrix coefficient matrix $A^{(l)}$, given by (3) for each layer):

$$\overline{\nabla}_{\mathbf{A}^{(l)}} E = \frac{\partial E}{\partial \mathbf{A}^{(l)}} = -\frac{1}{K_L}\left[\overline{\mathbf{D}}^{(l)} \circ \overline{\varphi'}^{(l)}\right] \otimes \left[\overline{\mathbf{X}}^{(l)} \otimes \overline{\mathbf{V}}\left(\overline{\mathbf{Z}}^{(l)}\right)\right] \tag{8}$$

where for the output layer $\overline{\mathbf{D}}^{(L)} = \overline{\mathbf{Y}}_d^{(L)} - \overline{\mathbf{Y}}^{(L)}$ and for the hidden layer $\overline{\mathbf{D}}^{(l)} = \mathbf{W}_{|\text{nobias}}^{(l+1)}\left(\overline{\mathbf{Z}}^{(l+1)}\right) \cdot \overline{\mathbf{D}}^{(l)} \circ \overline{\varphi'}^{(l+1)}$ L is the number of layers in the net, while K_l is the number of neurons in the l-th layer and $\overline{\mathbf{Y}}_d^{(L)}$ is the vector of desired outputs, $\overline{\varphi'}^{(l)}$ is the vector of l-th layer neurons' activation functions' derivative.

4.2 Efficient Training with Hessian-Based Algorithms

The use of the Kronecker product in the above formulas is the key for developing effective Hessian-based training algorithms for context-dependent nets. One of them is the contextual version of Levenberg-Marquardt algorithm derived in [1]. Here we shall present the main results. The approximation of the inverse of the Hessian matrix (with respect to the k-th neuron's coefficient vector) of the error function given by (6) for a sequence of N training examples may be calculated as

$$\mathbf{H}_{\mathbf{A}_{k,N}}^{-1} = \frac{1}{N}\left\{\sum_{n=1}^{N} e_{k,n}^2 \cdot \left[\overline{\mathbf{X}}_n \otimes \overline{\mathbf{V}}\left(\overline{\mathbf{Z}}_n\right)\right] \cdot \left[\overline{\mathbf{X}}_n \otimes \overline{\mathbf{V}}\left(\overline{\mathbf{Z}}_n\right)\right]^T\right\}^{-1} \tag{9}$$

where subscript n denotes a value for the n-th training example, $e_{k,n}$ is the k-the neuron's error. If all the examples from the sequence come from the same context group (have the same or similar values of the vector of basis functions, here denoted as $\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}_N\right)$), we may approximate the Hessian's inverse as

$$\mathbf{H}_{\mathbf{A}_{k,N}}^{-1} = \frac{1}{N}\left\{\sum_{n=1}^{N} e_{k,n}^2 \cdot \left[\overline{\mathbf{X}}_n \cdot \overline{\mathbf{X}}_n^T\right]\right\}^{-1} \otimes \left[\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}_N\right) \cdot \overline{\mathbf{V}}^T\left(\overline{\mathbf{Z}}_N\right)\right]^{-1} \tag{10}$$

The algorithm allows to reduce the computational complexity of training from $\mathcal{O}\left(W^3 M^3\right)$ to $\mathcal{O}\left(W^3 + M^3\right)$. As in most nets $M \ll W$, the computational effort of context-dependent net's learning is a little higher than for the traditional net of the same number of weights (thus the M times lower VC-dimension). The estimations give the possible training improvement of 10 to 1000 times (compared with traditional nets of the same VC-dimension, that is processing abilities) depending on the net's structure in traditional and contextual domain (the proportion of the number of weights to the number of basis functions). Let us notice that this computational gain is achievable only by applying the appropriate training algorithms and does not include the better convergence of net's parameters, expected from the better fitting of net's model to the model of the problem.

The presented algorithm is simple to apply as it is based just on the proper organization of training data into contextual groups containing examples of the

same (or comparable) value of contextual variables. Using samples from the same group in each training epoch (while mixing the examples from different groups between epochs) we may substitute the most operation demanding process of inverting the Hessian matrix with the inversion of two smaller matrices.

5 Conclusions

It has been shown that context-dependent neural nets, being the generalization of traditional nets, have better transformation abilities and are able to use their Vapnik-Chervonenkis dimension more efficiently than just with the growth of the net's size. The gradient-based training algorithms presented in the paper are of the comparable computational complexity than the ones for traditional nets. The original algorithms, using the properties of the Kronecker product for Hessian inverse calculation, improve the efficiency of training.

Acknowledgment. The work is supported by KBN grant in years 2002-2005

References

1. P. Ciskowski. *Learning of context-dependent neural nets.* PhD thesis, Wroclaw University of Technology, 2002.
2. P. Turney. The identification of context-sensitive features: A formal definition of context for concept learning. In *Proc. of 13th International Conference on Machine Learning (ICML96), Workshop on Learning in Context-Sensitive Domains,* Bari, Italy, 1996.
3. P. Turney. The management of context-sensitive features: A review of strategies. In *Proc. of 13th International Conference on Machine Learning (ICML96), Workshop on Learning in Context-Sensitive Domains,* Bari, Italy, 1996.
4. E. Rafajłowicz. Context dependent neural nets – problem statement and examples. In *Proc. of the Third Conference Neural Networks and Their Applications,* Zakopane, Poland, May 1999.
5. E. Rafajłowicz. Learning context dependent neural nets. In *Proc. of the Third Conference Neural Networks and Their Applications,* Zakopane, Poland, May 1999.
6. D. Yeung and G. Bekey. Using a context-sensitive learning for robot arm control. In *Proc. IEEE International Conference on Robotics and Automation,* pages 1441–1447, Scottsdale, Arizona, May 14-19 1989.
7. S. Lee and G. A. Bekey. Application of neural networks to robotics. *Control and Dynamic Systems,* 39:1–67, 1991.
8. D.-T. Yeung and G.A. Bekey. On reducing learning time in context-dependent mappings. *IEEE Transactions on Neural Networks,* 4(1):31–42, 1993.
9. R. Watrous and G. Towell. A patient-adaptive neural network ECG patient monitoring algorithm. In *Computer in Cardiology,* Vienna, Austria, September 10-13 1995.
10. M. Anthony and P.L. Bartlett. *Neural Network Learning: Theoretical Foundations.* Cambridge University Press.
11. P. Ciskowski. Context-dependent neural nets in contextual modelling. In *Proc. International Conference on Signals and Electronic Systems, ICSES 2002,* Swieradow-Zdroj, Poland, 2002.

An Unsupervised Neural Model to Analyse Thermal Properties of Construction Materials

Emilio Corchado[1], Pedro Burgos[1], María del Mar Rodríguez[2], and Verónica Tricio[2]

[1] Department of Civil Engineering. University of Burgos,
09006 Burgos, Spain
escorchado@ubu.es, pburgos@ubu.es
[2] Department of Physics. University of Burgos,
09001 Burgos, Spain
foulquie@arquinex.es, vtricio@ubu.es

Abstract. This study proposes a new approach to feature selection and the identification of underlaying factors. The goal of this method is to visualize and extract information from complex and high dimensional data sets. The model proposed is an extension of Maximum Likelihood Hebbian Learning [14], [5], [15] based on a family of cost functions, which maximizes the likelihood of identifying a specific distribution in the data while minimizing the effect of outliers [7], [10]. We demonstrate a hierarchical extension method which provides an interactive method for identifying possibly hidden structure in the dataset. We have applied this method to study the thermal evolution of several construction materials under different thermal and humidity environmental conditions.

1 Introduction

We introduce a method which is closely related to exploratory projection pursuit. It is an extension of a neural model based on the Negative Feedback artificial neural network [3]. This method is called Maximum-Likelihood Hebbian learning (ML) [5], [14], [12].

In this paper we provide a hierarchical extension to the ML method. This extension allows the dynamic investigation of a data set in which each subsequent layer extracts structure from increasingly smaller subsets of the data. The method reduces the subspace spanned by the data as it passes through the layers of the network therefore identifying the lower dimensional manifold in which the data lies.

The Negative Feedback artificial neural network has been linked to the statistical techniques of Principal Component Analysis (PCA)[3], Factor Analysis [2] and Exploratory Projection Pursuit (EPP) [11]. The originality of this paper is the development and application of Hierarchical Maximum Likelihood Hebbian Learning (HML) to provide a novel approach different than those found by the contemporary statistical technique of PCA.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 204–211, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 The Negative Feedback Neural Network

We introduce the Negative Feedback Network [3].

Consider an N-dimensional input vector, **x**, and a M-dimensional output vector, **y** , with W_{ij} being the weight linking input j to output i and let η be the learning rate.

The initial situation is that there is no activation at all in the network. The input data is fed forward via weights from the input neurons (the **x**-values) to the output neurons (the **y**-values) where a linear summation is performed to give the activation of the output neuron. We can express this as:

$$y_i = \sum_{j=1}^{N} W_{ij} x_j, \forall i \tag{1}$$

The activation is fed back through the same weights and subtracted from the inputs (where the inhibition takes place):

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \forall j \tag{2}$$

After that simple Hebbian learning is performed between input and outputs:

$$\Delta W_{ij} = \eta e_j y_i \tag{3}$$

The effect of the negative feedback is to stabilise the learning in the network. Because of that it is not necessary to normalise or clip the weights to get convergence to a stable solution. Note that this algorithm is clearly equivalent to Oja's Subspace Algorithm[6] since if we substitute equation (2) in equation (3) we get:

$$\Delta W_{ij} = \eta e_j y_i = \eta \left(x_j - \sum_k W_{kj} y_k \right) y_i \tag{4}$$

This network is capable of finding the principal components of the input data [3] in a manner that is equivalent to Oja's Subspace algorithm [6], and so the weights will not find the actual Principal Components but a basis of the Subspace spanned by these components.Writing the algorithm in this way gives us a model of the process which allows us to envisage different models which would otherwise be impossible [5,6].

3 A Family of Learning Rules

It has been shown [8] that the nonlinear PCA rule

$$\Delta W_{ij} = \eta \left(x_j f(y_i) - f(y_i) \sum_k W_{kj} f(y_k) \right) \tag{5}$$

can be derived as an approximation to the best non-linear compression of the data. Thus we may start with a cost function

$$J(W) = 1^T E\left\{\left(\mathbf{x} - Wf\left(W^T\mathbf{x}\right)\right)^2\right\} \tag{6}$$

which we minimise to get the rule(5). [10] used the residual in the linear version of (6) to define a cost function of the residual

$$J = f_1(\mathbf{e}) = f_1(\mathbf{x} - W\mathbf{y}) \tag{7}$$

where $f_1 = \|.\|^2$ is the (squared) Euclidean norm in the standard linear or nonlinear PCA rule. With this choice of $f_1(\ )$, the cost function is minimized with respect to any set of samples from the data set on the assumption that the residuals are chosen independently and identically distributed from a standard Gaussian distribution [13]. We may show that the minimization of J is equivalent to minimizing the negative log probability of the residual, $\mathbf{e}$, if $\mathbf{e}$ is Gaussian.

$$\text{Let } p(\mathbf{e}) = \left(1/_Z\right)\exp(-\mathbf{e}^2) \tag{8}$$

The factor Z normalizes the integral of $p(\mathbf{y})$ to unity.

Then we [5] can denote a general cost function associated with this network as

$$J = -\log p(\mathbf{e}) = (\mathbf{e})^2 + K \tag{9}$$

where K is a constant. Therefore performing gradient descent on J we have

$$\Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial J}{\partial \mathbf{e}}\frac{\partial \mathbf{e}}{\partial W} \approx \mathbf{y}(2\mathbf{e})^T \tag{10}$$

where we have discarded a less important term (see [9] for details).

In general [7], the minimisation of such a cost function may be thought to make the probability of the residuals greater dependent on the probability density function (pdf) of the residuals. Thus if the probability density function of the residuals is known, this knowledge could be used to determine the optimal cost function.

[10] investigated this with the (one dimensional) function:

$$p(\mathbf{e}) = \left(1/_{2+\varepsilon}\right)\exp\left(-|\mathbf{e}|_\varepsilon\right) \tag{11}$$

where

$$|\mathbf{e}|_\varepsilon = \begin{cases} 0_ \forall|\mathbf{e}| < \varepsilon \\ |\mathbf{e}| - \varepsilon_ otherwise \end{cases} \tag{12}$$

with ε being a small scalar ≥ 0.

[10] described this in terms of noise in the data set. However we feel [12] that it is more appropriate to state that, with this model of the pdf of the residual, the optimal $f_1(\ )$ function is the ε-insensitive cost function:

$$f_1(\mathbf{e}) = |\mathbf{e}|_\varepsilon \tag{13}$$

In the case of the Negative Feedback Network, the learning rule is

$$\Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial f_1(\mathbf{e})}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial W} \tag{14}$$

which gives:

$$\Delta W_{ij} = \begin{cases} 0 \text{ if } |e_j| < \varepsilon \\ \eta y_i (sign(e_j)) \text{ otherwise} \end{cases} \tag{15}$$

In a more general situation let the residual after feedback has a probability density function (pdf)[5,12,14]:

$$p(\mathbf{e}) = \frac{1}{Z} \exp(-|\mathbf{e}|^p) \cdot \tag{16}$$

Then we [5] can denote a general cost function associated with this network as

$$J = -\log p(\mathbf{e}) = |\mathbf{e}|^p + K \tag{17}$$

where K is a constant. Therefore performing gradient descent on *J* we have

$$\Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial J}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial W} \approx y(p\,|\mathbf{e}|^{p-1}\, sign(\mathbf{e}))^T \tag{18}$$

where T denotes the transpose of a vector. We would expect that for leptokurtotic residuals (more kurtotic than a Gaussian distribution), values of p<2 would be appropriate, while for platykurtotic residuals (less kurtotic than a Gaussian), values of p>2 would be appropriate.Therefore the network operation is:

$$\text{Feedforward: } y_i = \sum_{j=1}^{N} W_{ij} x_j, \forall_i \tag{19}$$

$$\text{Feedback: } e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i \tag{20}$$

$$\text{Weight change: } \Delta W_{ij} = \eta . y_i . sign(e_j) |\, e_j \,|^{p-1} \tag{21}$$

[10] described their rule as performing a type of PCA, but this is not strictly true since only the original (Oja) ordinary Hebbian rule actually performs PCA. It might be more appropriate to link this family of learning rules to Principal Factor Analysis since this method makes an assumption about the noise in a data set and then removes the assumed noise from the covariance structure of the data before performing a PCA. We are doing something similar here in that we are basing our PCA-type rule on the assumed distribution of the residual. By maximising the likelihood of the residual with respect to the actual distribution, we are matching the learning rule to the pdf of

the residual. Maximum Likelihood Hebbian Learning (ML) [5,12,14] has been linked to the standard statistical method of EPP [11,14].

4 Hierarchical Maximum Likelihood Hebbian Learning (HML)

There may be cases where the structure of the data may not be captured by a single linear projection. In such cases a hierarchical scheme may be beneficial.

This can be done in two ways, firstly by projecting the data using the ML [5], [12], [14] method, select the data points which are interesting and re-run the ML network on the selected data. Using this method only the projections are hierarchical.

A second more interesting adaptation is to use the resulting projected data of the previous ML network as the input to the next layer. Each subsequent layer of the network identifying structure among fewer data points in a lower dimensional subspace.

The HML method can reveal structure in the data which would not be identified by a single ML projection as each subsequent projection can analyse different sections of the subspace spanned by the data. This method has an additional advantage that with each sub-projection the number of samples and dimensionality of the data is reduced. As this is a linear method, by the linear combination of all previous layers we can identify the manifold in which the data points of interest lie.

5 Real Data Set and Experimental Results

The data used to illustrate our method is obtained from an experimental work which goal is the analysis of the thermal transmission of several construction materials [16]. We want to analysis their similarities and differences of behaviour. For that reason we have studied, in an independent way, the thermal transmission of these materials in a stationary state and in a transitory state. The data is collected from a "thermal house" with a heat source inside the house controlled by a thermostat outside of it. The house was inside of an inhabited building. It is a closed cube, in which all of the four vertical faces are made of different materials:

A piece of plywood (3 cm of thickness),	a simple glass (1,8 cm of thickness),
A piece of plaster (1,5 cm of thickness),	a piece of polystyrene (4 cm of thickness)

The experiment has been designed to register representative variables and them have been measured inside and outside the house. So we have measured the inside and outside environmental temperature, the inside and outside relative humidity, the inside and outside surface temperature of each vertical face and the thermal flux of each vertical face.

The initial situation is an equilibrium state between the inside and outside temperature for both cases. These are the surface temperature and the environmental one. Then the heat source is switched on until it reaches a stationary state. Once this state is reached, we kept the heating on for a while and then we switched it off. The data

recorded belong to a period which starts before the heating is switched on and finishes when the equilibrium state is reached after the heating is switched off.

We performs two series of measures with the same inside heat source. We do not provide humidity in the first serie and we do it during the second serie of measures. We have performs the same experiment several times.

In the following experiments we analyse the performance of the neural method (HML) and highlight the differences in the projections obtained by PCA and ML. We demonstrate also the HML method.

Comparison of PCA and ML on the Construction Materials Data Set

Figure 1 and Figure 2 show the behaviour of some thermal properties (the inside and outside environmental temperatures, the inside and outside relative humidity and the inside and outside surface temperature) of the four vertical faces during the ascending transitory state.

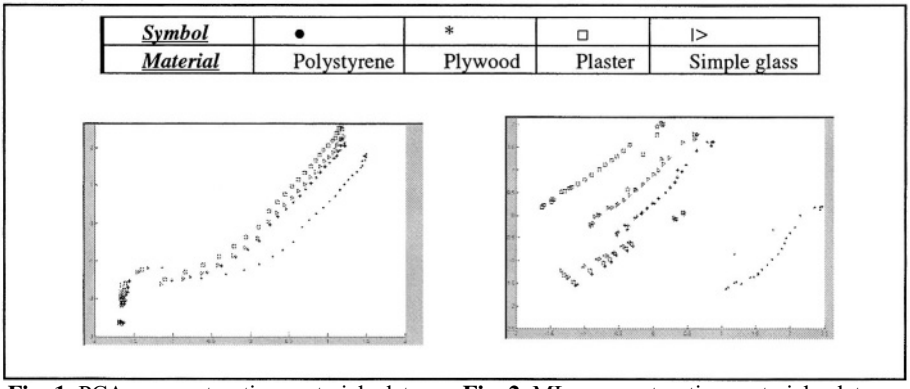


Fig. 1. PCA on construction materials data. **Fig. 2.** ML on construction materials data

In Figure 1 and Figure 2 we show the comparison of PCA and ML projections of the construction material data. ML (Fig.2) clearly shows greater separation in the behaviour of different materials than is achieved with PCA (Fig.1). Figure 2 (ML) shows that Polystyrene behaves very different than the other three materials. These three materials (plywood, glass and plaster) have a similar behaviour. We could go further and say that plywood and glass have some times almost the same behaviour.

Behaviour of Two Vertical Faces, One Made of Glass and the Other One Made of Wood

In Figure 3 we show a representation of the behavior of some properties (inside and outside temperature, relative inside and outside humidity, the inside and outside surface temperature of each vertical face and their thermal flux) of two vertical faces, one made of glass and the other one made of wood.

If we analyse Fig.3 we could see that these two construction materials have quite the same behaviour in terms of temperature, but the thermal fluxes are different. These logic similarities and differences can not be observed by other classic models used by physicists [16].

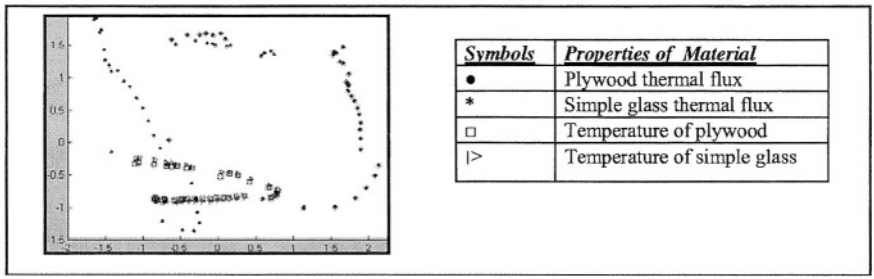


Fig. 3. ML on construction materials data.

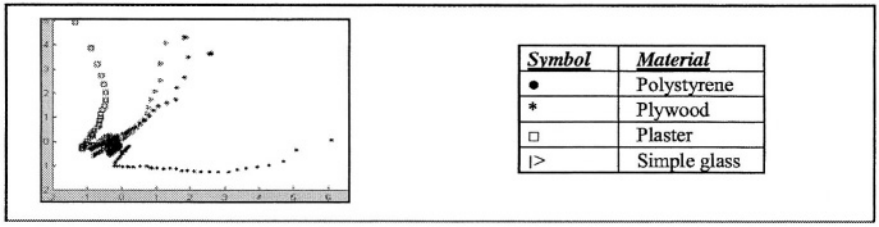


Fig. 4. ML on construction materials data.

5.1 Results Using Hierarchical Maximum Likelihood Hebbian Learning

Fig.4. Representation of inside and outside temperature, relative inside and outside humidity and the inside and outside surface temperature of each vertical face versus time. The values correspond to the descendant transitory state. Fig.4 represents the results obtained using ML.

An important advantage of the hierarchical model is that provides more information in terms of inflexion points. This allows a better identification of different states in the studied transitory state and the identification of changes of state.as we could see in Fig.5.

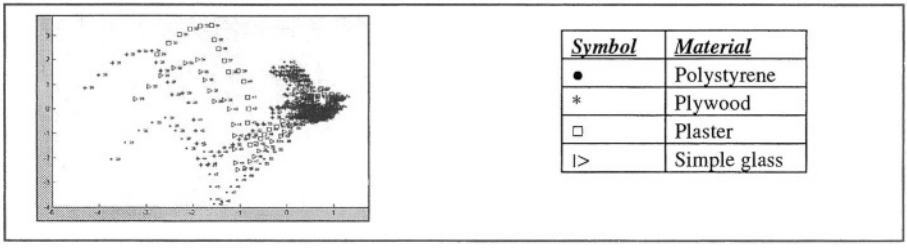


Fig. 5. Result of HML on left cluster in Fig.4

In Fig.5 we can see that HML identifies the inflexion points for the construction materials in a very clear way and gives us a new projection. It shows that the inflexion points of the four materials take places at the same time. If we compare Fig.5 and the temperature evolution plot (Fig.4) we may see some minimal temperature and relative humidity fluctuations in an inhabited place.

6 Conclusions

We have reviewed the recent method of ML, extend it to develop a hierarchical method and compare it with a typical statistical technique. The method proposed in this paper provides a novel way to extract more information from a particular good projection and it gives a very clear method to analyse the thermal behavior of construction materials. Future work will investigate further more complex data sets in order to study materials with an unpredictable behavior and in extreme conditions.

References

1. Bertsekas, D.P. Nonlinear Programming. Athena Scientific, Belmont, MA, 1995.
2. Charles, D. Unsupervised Artificial Neural Networks for the Identification of Multiple Causes in Data. PhD thesis, University of Paisley. 1999
3. Fyfe, C. Negative Feedback as an Organising Principle for Artificial Neural Networks, PhD Thesis, Strathclyde University. 1995
4. Fyfe, C. A. Neural Network for PCA and Beyond, Neural Processing Letters, 6:33-41. 1996.
5. Fyfe, C. and Corchado, E., Maximum Likelihood Hebbian Rules. European Symposium on Artificial Neural Networks. 2002.
6. Oja, E. Neural Networks, Principal Components and Subspaces, International Journal of Neural Systems, 1:61-68. 1989.
7. Smola, A.J. and Scholkopf, B. A Tutorial on Support Vector Regression. Technical Report NC2-TR-1998-030, NeuroCOLT2 Technical Report Series. 1998.
8. Xu, L. Least Mean Square Error Reconstruction for Self-Organizing Nets", Neural Networks, Vol. 6, pp. 627-648. 1993.
9. Karhunen, J. and Joutsensalo, J. Representation and Separation of Signals Using Nonlinear PCA Type Learning, Neural Networks, 7:113-127. 1994.
10. Fyfe, C. and MacDonald, D. ε-Insensitive Hebbian learning", Neuro Computing, 2002.
11. Friedman, J. and Tukey, J. A Projection Pursuit Algorithm for Exploratory Data Analysis. IEEE Transaction on Computers, (23): 881-890. 1974.
12. Corchado, E. and Fyfe, C. Orientation Selection Using Maximum Likelihood Hebbian Learning, International Journal of Knowledge-Based Intelligent Engineering Systems Volume 7 Number 2, April 2003. ISSN: 1327-2314. Brighton, United Kingdom.
13. Bishop, C.M., Neural Networks for Pattern Recognition, Oxford.,1995.
14. Corchado, E. MacDonald, D. and Fyfe, C. Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit, Data Mining and Knowledge Discovery, Kluwer Academic Publishing, (In-Press).
15. Corchado, E. and Fyfe, C. Connectionist Techniques for the Identification and Suppression of Interference Underlying Factors. International Journal of Pattern Recognition and Artificial Intelligence. (IJPRAI). Vol. 17, No. 8 (Dec 2003).
16. Tricio, V.and Viloria, R. Microclimatic Study in a Historical Building: Protection and Conservation of the Cultural Heritage of the Mediterranean Cities. Topic: Environmental parameters of the Mediterranean Basin. 2002. Pg: 67-70.

Intrusion Detection Based on Feature Transform Using Neural Network

Wonil Kim[1], Se-Chang Oh[2], and Kyoungro Yoon[3]*

[1] Sejong University, Dept. of Digital Contents, 98 Gunja-Dong, Gwangjin-Gu, Seoul, Korea 143-747
wikim@sejong.ac.kr

[2] Sejong Cyber University, 111-1 Gunja-Dong, Gwangjin-Gu, Seoul, Korea 143-747
scoh713@naver.com

[3] Konkuk University, Dept. Computer Science and Engineering, 1 Hwayang-Dong, Gwangjin-Gu, Seoul, Korea 143-701
yoonk@konkuk.ac.kr

Abstract. In this paper, a novel method for intrusion detection is presented. The presented method uses a clustering method based on the transformed features, which can enhance the effectiveness of the clustering. Clustering is used in anomaly detection systems to separate attack and normal samples. In general, separating attack and normal samples in the original input space is not an easy task. For better separation of the samples, a transformation that maps input data into a different feature space should be performed. In this paper, we propose a novel method for obtaining proper transformation function that reflects the characteristics of the given domain. The transformation function is obtained from the hidden layer of the trained three-layer neural network. Experiments over network connection records from KDD CUP 1999 data set are used to evaluate the proposed method. The result of the experiment clearly shows the outstanding performance of the proposed method.

1 Introduction

The goal of an Intrusion Detection System (IDS) is to detect any kind of attack effectively, whereas reducing false alarm of legitimate accesses. There have been many researches on this field and published various methods to achieve this goal, but not many of the previous systems reported satisfactory results on achieving the goal of separating attacks and legitimate accesses. There are two categories for intrusion detection method. One is misuse detection method and the other is anomaly detection method. In the misuse detection method, a model called a signature is developed based on the attack patterns and any activity similar to the defined signature is regarded as an intrusion [1]. On the other hand, in the anomaly detection method, a model called a profile is developed based on the normal access patterns, and any activity that deviates from the defined profile is regarded as intrusion [2].

* Author for Correspondence, +82-2-450-4129. This work was supported by the faculty research fund of Konkuk University in 2003.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 212–219, 2004.
© Springer-Verlag Berlin Heidelberg 2004

There are several ways to model the legitimate behavior, such as statistical method, data mining, and clustering. In statistical method, statistical values like mean and variance are calculated for features of data, and the set of these values is used as a profile. In [3][4], a statistical distribution is assumed and the parameters for this distribution is estimated from data. Statistical method is too simple to model complex distributions of data. Another approach is using data mining technique. It finds common patterns in given data, after which a decision tree or a rule set is constructed from the discovered common patterns [5]. However, the complexity of the method is too high to be used in a real-time application. Finally, clustering selects several representative centroids for normal samples in the input space [6]. Then the selected centroids are used as the profiles. The clustering can be performed in unsupervised manner since most of the data is normal [7]. This method can model any data of complex distribution, and the computational cost is acceptable. The problem of reported clustering approaches [6][7] is that the distribution of normal and attack samples is not easily separable and in some cases, the normal data themselves are not well clustered.

This paper presents a novel approach based on the clustering methods to solve the problems, in which the normal and attack samples are not separable and the samples themselves are not well clustered. We solve the clustering problem by using transformed features instead of the original input fields (attributes). The original data are transformed to a new feature space so that the attack and normal samples are well clustered and clearly separable. The feature transform is performed using the hidden layer nodes of trained neural network as explained in section 3.2 and 4.3..

The main problem and main idea are described in section 2 and section 3 respectively. The experimental results using KDD cup data are shown and evaluated in section 4. Finally, the conclusion and discussion are given in section 5.

2 Problem Description

The problem is to find a reasonable transformation function that separates samples from different classes and categorizes the samples into the same class.

In the proposed approach, the feature space of the original given data is transformed into a different feature space. There are several reasons for this transformation. First, there may be highly correlated data fields in the raw data [8]. Principal Component Analysis (PCA) [9] and Independent Component Analysis (ICA) [10] that find independent features in the given data are some of the most popular methods to find such data fields in the raw data. Second, several data fields dominate the rest of the data fields as the raw data is usually not normalized. Generally, balancing the fields by weighting is done manually in preprocessing. Third, the most radical reason is that samples from different classes, in general, are not separable in the input space. This appears commonly in many problem domains, and makes the clustering difficult and inefficient. The first goal of the presented approach is finding a transformed feature space in which the samples are well separable and grouped.

3 Data Transformation and Anomaly Detection

There are two steps in this data transformation. The first step is finding fields irrelevant in discriminating two classes of data, i.e. attack and normal classes, and excluding the irrelevant fields. Of course, a proper transformation function would ignore these fields even if they are not excluded beforehand. But if we can exclude them beforehand, it would reduce the computational costs. The second step is finding a transformation function itself. In this paper, we propose a method for finding a transformation function using neural networks. It exploits the fact that a trained neural network has capability of discriminating between classes. We discuss about how to use a neural network trained with a given set of labeled samples for data transformation in the following sections. Through the two steps, input data are expected to be transformed into a well separable feature space.

After the input data are transformed into the desired feature space, we use *k*-means algorithm [11] to define clusters of normal class data including the centroid and radius of each cluster. If new data (data under test) are not within the radius of these clusters, they will be considered as attack data.

3.1 Field Selection

In this process, the fields whose values are scarcely or randomly changed are removed from the feature space using three steps, as they are either meaningless in the aspect of information theory, or not helpful for discriminating between classes.

In the first step, the fields whose values hardly change are selected to be removed, as the fields with relatively fixed value do not give any discriminating information. In the second step, the fields whose values change randomly are selected to be removed, as the random behavior makes it very hard to find rules for discriminating classes. To measure the randomness, entropy of each field is calculated. Fields with high entropy values are considered as random fields. In the third step, Bayes decision theory is used to evaluate the contribution of each field to the discrimination between classes. For this step, the Bayesian threshold value for each field, is selected based on the normal and attack samples from the labeled training data set from KDD Cup 1999 [12]. Using the selected threshold value, the error probability of using each field for the discrimination factor is calculated. Finally, the fields with high probability of error are removed.

3.2 Input Data Transformation

In this step, we employ neural network to obtain the function τ for transforming data. The well-known back propagation algorithm is used to train the neural network [13]. Using the weights in the trained neural network, we can obtain the transform function. The transformation is a process for obtaining the feature vector $\boldsymbol{b}$. As a result, we obtain the transformation function $\boldsymbol{b} = \tau(\boldsymbol{a})$ where $\boldsymbol{a}$ is the input vector (selected

features from the original input space) and $\boldsymbol{b}$ is the output vector acquired from input to the output node of the trained neural network.

In other words, a neural network with hidden nodes is trained, and the values of the hidden nodes in the trained neural network provide the parameters for the feature transformation.

As the training of the neural network is performed with approximately the same number of normal samples and attack samples, less than 2 percent difference to be exact, the trained neural network models both normal pattern and attack pattern. Therefore, the proposed method is not biased to either normal samples or attack samples in the process of developing the transform function, and provides effective discrimination power of normal and attack samples.

3.3 Clustering and Class Labeling

Once the transformation function is acquired, i.e. the neural network is trained, profile of the normal data is ready to be developed using clustering method.

There are various clustering methods available, and we use the *k*-means algorithm for our clustering method [11]. Assuming that most of the network connections are normal patterns in practice, we can build the clusters only with normal dataset. When *k* clusters are built, we remove clusters with relatively small number of samples. In the process of cluster removal, the assumption is that the normal samples in the transformed feature space are well grouped together and most of the clusters are practically empty when *k* is large enough. The appropriate *k* (number of initial clusters) and the number of remaining clusters after removal can be decided experimentally. The definitions of the remained clusters become the profile for the anomaly detection.

When a connection record is received from the network, we can acquire transformed features of the connection record using the trained neural network, i.e. using the developed transform function. The transformed feature vector is compared to the cluster definitions in the profile. If it does not belong to any of the clusters in the profile, the connection is considered anomalous. In other words, any samples which do not belong to any of the developed clusters of the normal samples are considered attack samples. This idea of using limited number of clusters in the transformed feature space outperforms any known anomaly detection system as shown in the experimental results.

4 Experimental Results

4.1 Data Description

We evaluated our proposed method over the KDD Cup 1999 data [12]. It contains a wide variety of intrusions simulated in a military network environment. Each sample in the data is a record of extracted features from a network connection gathered during the simulated intrusions. A connection is a sequence of TCP packets to and from various IP addresses. The TCP packets were assembled into connection records using

the Bro program [14]. A connection record consists of 42 fields. It contains basic features about TCP connection as duration, protocol type, and number of bytes transferred. It contains domain specific features as number of file creation, number of failed login attempts, and whether root shell was obtained. Also it has the last field for labeling it as either normal or one specific kind of attack. Following is a typical example of connection record.

(0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,254,1.00,0.01,0.00,0.00,0.00,0.00,0.00,0.00,normal)

We used three data files for experiments: the file named "kdd-train-nor" and "kdd-train-att" for training the neural network and for developing normal clusters; and the file named "kdd-test" for evaluation. The origins of these files are "kddcup.data_10_percent" and "corrected" downloaded from KDD Cup 1999 site. The file "kdd-train-nor" consists of 97,278 normal samples selected from the file "kddcup.data_10_percent". The file "kdd-train-att" consists of 95,649 attack samples selected from the file "kddcup.data_10_percent" so as to make a balance in size with the training set of normal samples. Finally, the file "kdd-test" which is the same with the file "corrected" consists of labeled 60,593 normal and 250,436 attack samples.

4.2 Field Selection

Among the 42 fields of connection record, the symbolic fields such as protocol_type, service, and flag are difficult to represent in numerals. A Bayesian classifier is trained and tested for each of 38 fields except three symbolic fields and the label field. Fig. 1 shows the error rate of Bayesian classifier for each field. Any field with the error rate higher than 20%, as a rule of thumb, is removed and 12 fields are selected.

4.3 Data Transformation

A three-layered neural network was used to acquire transformation function for the simplicity of computation. The input layer of the neural network consists of 12 nodes to present 12 field values as input. The number of nodes in the middle layer (hidden layer) should be decided depending on the complexity of the problem. As the number of node increases, the complexity of the network increases. Also large number of nodes makes the train time longer and the resulting neural network could be over-specialized. If the number is too small on the other hand, the neural network cannot be trained as we wanted. In this intrusion detection system, ten nodes are employed for the middle layer, based on the experiments. Finally, the output layer has only one node because the number of classes to be distinguished is two (normal or attack). In this case, if the value of output node is close to 1, the given data is accepted to belong to the normal class and if the value is close to 0, the given data is accepted to belong to the attack class. The neural network is trained using the data set "kdd-train-nor" and "kdd-train-att" as explained in section 4.1.

After the training of the neural network, each sample is transformed before clustering is performed. The equation $\boldsymbol{b} = \tau(\boldsymbol{a})$ is computed using the weights of the

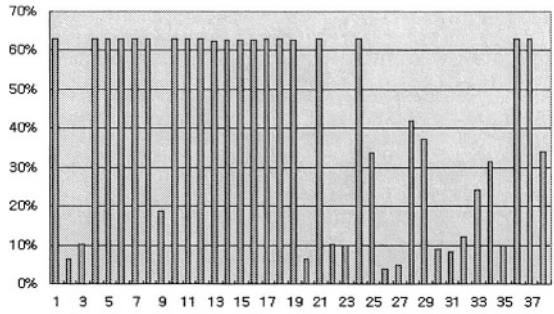


Fig. 1. The error rate of Bayesian classifier for each field is shown. Among 38 fields, Bayesian classifiers of only 12 fields have error rate less than 20% and those 12 fields are selected as input space features.

first two layers from the trained neural network to transform a sample. In other words, the two layer neural network with 10 output nodes (the hidden nodes in the original three layer neural network), which is acquired by removing the single node output layer from the trained neural network, represents the transformation function.

4.4 Clustering and Anomaly Detection

We used the file "kdd-train-nor" as the training set to build up the cluster set, since we assumed that most of the samples are normal in practice, following the anomaly detection approach. As a result, each sample $\boldsymbol{a}$ of 12-dimensional vector from the file "kdd- train-nor" is transformed to a 10-dimensional feature vector $\boldsymbol{b}$. Then the k-means algorithm [11] is used for clustering the set of transformed vectors. It is possible to measure the size of each cluster by counting the number of samples classified as the cluster. The clusters with small size are removed from the cluster set, after which the remaining clusters contain 98% of the original samples. The centroids and radii of the remaining clusters are used as the profile of the normal data.

Fig. 2 shows the number of remaining clusters after removing small clusters. The horizontal axis stands for the value k which is the number of initial clusters. The result of clustering with original data shows that, as the number of clusters increase, more normal samples are covered by the clusters. In other words, the samples are scattered in the original input feature space. On the other hand, the result of clustering with transformed data shows that the number of clusters stabilizes rapidly. This means that the samples are well grouped in the transformed feature space.

Fig. 3 (a) and Fig. 3 (b) show the classification result with and without transformation. Two indicators are used to measure the performance: the attack detection rate and the false positive rate. The file "kdd-test" that contains both normal samples and attack samples are used as the test set. Cluster information (profile of normal data) acquired using the k-means algorithm with "kdd-train-nor" data set in the previous stage is used for classification task. Fig. 3 (a) shows that it is hard to increase the

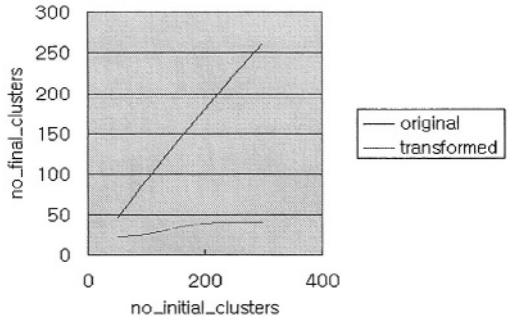


Fig. 2. Even if the number of initial clusters *(k)* increases, most of the samples from the normal connection stay within very limited number of clusters in the transformed feature space. In this experiment, less than 50 of the clusters cover most of the normal samples from the KDD Cup dataset.

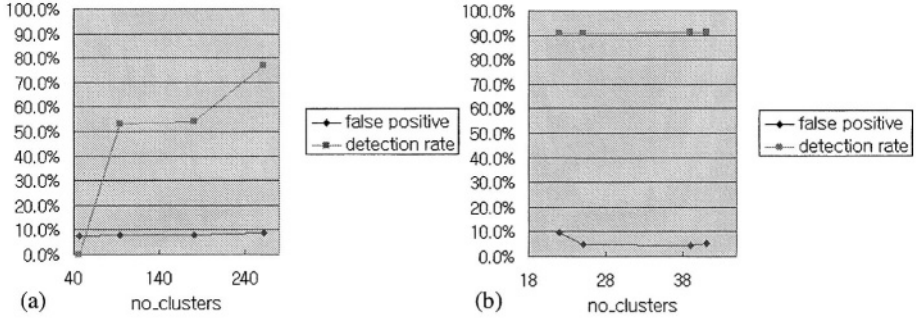


Fig. 3. Theses figures compare the performance of anomaly detection with various numbers of clusters, (a) and (b) show the performances when the clustering is performed in the input-feature space and the transformed feature space, respectively. In (a), we can clearly see that the detection rate depends on the number of clusters used. In other words, the samples are scattered in the original input feature space and they cannot be covered by a small number of clusters. On the other hand, in (b), we can clearly see that less than 40 clusters are enough to cover most of the normal samples and the normal samples are well clustered together. Furthermore, when the feature space transformation is performed as presented, the detection rate greatly increases, while the false positive rate is kept low using only a small number of clusters.

detection rate, even if we increase the number of clusters, when the transformation is not performed. On the other hand, Fig. 3 (b) clearly shows that the detection rate stabilizes to a higher level with relatively small number of clusters, whereas the false positive rate is kept very low, when the transformed data are used.

5 Conclusion

In this paper, we proposed a neural network based feature transformation for intrusion detection using clustering. First, we explained how the characteristics of data can affect the clustering results. Then we discussed how the raw data should be trans-

formed in order to achieve efficient clustering. To evaluate the proposed method, the performance of anomaly detection based on the clustering in the transformed feature space was compared with the performance of anomaly detection based on the clustering in the input space. The result shows that the number of generated clusters is decreased and the correctness is increased dramatically in the transformed feature space. In practice, the number of clusters critically affects the efficiency of intrusion detection engines. The proposed method, in this sense, clearly improves the efficiency as well as the correctness of the intrusion detection system. This method can be easily extended to many problems in which the samples in the original feature space is not easily separable.

References

1. Lunt, A., Tamaru, A., Gilham, F.: IDES: A Progress Report. Proceedings of the 6th Annual Computer Security Applications Conference Tucson, AZ (1990)
2. Denning, D.: An Intrusion Detection Model. Proceedings of the Seventh IEEE Symposium on Security and Privacy (1986) 119-131
3. Ye, Nong: A Markov chain model of temporal behavior for anomaly detection. Proceedings of the 2000 IEEE Systems, Man, and Cybernetics; Information Assurance and Security Workshop (2000)
4. Eskin, E., Lee, W., Stolfo, S. J.: Modeling system calls for intrusion detection with dynamic window sizes. Proceedings of DARPA Information Survivability Conference and Exposition II (DISCEX II) (2001)
5. Lee, W., Stolfo, S. J., Mok, K.: A Data Mining Framework for Building Intrusion Detection Models. Proceedings of 1999 IEEE Symposium on Security and Privacy (1999)
6. Portnoy, L., Eskin, E., Stolfo, S.: Intrusion detection with unlabeled data using clustering. Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA 2001) (2001)
7. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. Data Mining for Security Applications, Kluwer (2002)
8. Dura, R. O., Hart, P. E.: Pattern Classification and Scene Analysis. Wiley-Interscience Publication (1973) 247
9. Tou, J., Heydorn, R.: Some Approaches to Optimum Feature Extraction. Computer and Information Sciences II, Academic Press, New York (1967)
10. Lee, Te-Won: Independent Component Analysis: Theory and Applications. Kluwer Academic Publishers (1998)
11. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Data. Proceedings of the 5th Berkeley Symposium on Probability and Statistics, University of California Press, Berkeley (1967)
12. The third international knowledge discovery and data mining tools competition dataset KDD99-Cup. Available at `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html` (1999)
13. K.Mehrotra, C.K.Mohan, and S.Ranka, Elements of Artificial Neural Networks, MIT Press (1997)
14. Paxon, V.: Bro: A system for detecting network intruders in real-time. Proceedings of the 7th USENIX Security Symposium, San Antonio, TX (1998)

Accelerating Wildland Fire Prediction on Cluster Systems*

Baker Abdalhaq, Ana Cortés, Tomás Margalef, and Emilio Luque

Departament d'Informàtica, E.T.S.E, Universitat Autònoma de Barcelona, 08193-Bellaterra (Barcelona) Spain
baker@aows10.uab.es,
{ana.cortes,tomas.margalef,emilio.luque}@uab.es

Abstract. Classical prediction fire schemes do not match the real fire propagation, basically, because of the complexity of the physical models involved, the need for a great amount of computation and the difficulties of providing accurate input parameters. We describe an enhanced prediction scheme, which uses recent fire history and optimization techniques to predict near future propagation. The proposed method takes advantage of the computational power offered by distributed systems to accelerate the optimization process at real time.

1 Introduction

Wildland fire is an important hazard from the ecological, economical and social points of view. Considerable effort from different ambits (physics, biology, computer science...) are often joined in order to develop fire propagation models that can help fire prevention and mitigation. Classical prediction schemes rely on these fire propagation simulators, however, in most of cases, the results provided by simulation tools do not match real propagation. One of the most common sources of deviation in fire simulation spread from that of real-fire propagation is imprecision in input model parameters. A way of overcoming this input parameter uncertainty during real fires lies in exploiting to the limit all available data about the recent fire history in order to predict its near-future evolution [1]. We propose an enhanced prediction scheme, which, basically, works as follows. Once the initial fire line and a real propagation after a certain time interval are available, an optimization technique will be applied in order to determine a set of input parameters, providing the best match between simulation and real-fire behavior. These values will then be used to predict fire behavior in the next interval. Whilst feed-back from the field is supplied, this process is repeated so as to correct the prediction by using the available information. This enhanced prediction scheme reflects the dynamics of the fire environment (wind, moisture content, etc.), since the real data is updated periodically.

* This work has been supported by the *Comisión Interministerial de Ciencia y Tecnología* (CICYT) under contract TIC2001-2592 and by the European Commission under contract EVG1-CT-2001-00043 SPREAD.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 220–227, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Parameter optimization and prediction must be carried out faster than real-time propagation so that the prediction can be useful in deciding which actions need to be taken in tackling the emergency. However, the number of parameters is quite large and the resulting search space for the optimization strategy becomes enormous. It is not, therefore, feasible to assess the whole search space, which needs to be reduced by applying certain techniques. The current state of the art in the computational field offers the required background that should be applied. On the one hand, evolutionary computing is a well-established field with several techniques in the literature that are widely accepted (for example, Genetic Algorithms [2]). These techniques can be applied to guiding the search over the whole space, so that only certain cases are tested. On the other hand, computing systems based on parallel and distributed platforms offer the required computing power to apply these techniques and to provide successful results in an acceptable time. Typically, these techniques work in an iterative way by improving the obtained solution at each iteration. Therefore, a clear way of saving time consists of increasing the convergence speed of the optimization technique. For this purpose, we propose applying a sensitivity analysis to the input parameters in order to asses their impact on output and, consequently, to determine which parameters are worth spending time on tuning and which are not, maintaining the latter at an estimated value. In order to be more effective in tuning the most sensitive parameters, we also propose introducing a certain degree of knowledge during the optimization process. This knowledge will consist of limiting the range of the tuned parameters around an estimated value (which may be the real measurement) for those parameters.

The rest of this paper is organized as follows. Section 2 introduces the parallel implementation of the enhanced prediction scheme. Section 3 is devoted to the sensitivity analysis carried out. Section 4 reports on the experimental study carried out, and the results obtained. Finally, section 5 presents the main conclusions to this work.

2 Parallel Wildland Fire Prediction Method

The parallel enhanced prediction method is a step forward with respect to the classical methodology. It introduces the idea of applying an optimisation scheme to calibrate the set of input parameters of a given fire simulator, with the aim of finding an "optimal" set of inputs, which improves the results provided by the fire-spread simulator. Figure 1 shows how the proposed prediction method works. The FS box corresponds to the underlying fire simulator (ISStest in our case [3]). This simulator needs to be fed with some input parameters, which will form the input parameters set to be optimised. In our particular case, the number of input parameters required are 9. In section 3, we will describe in more detail each one of these parameters. In order to evaluate the good ness of the results provided by the simulator, a fitness function (FF box) has been included, which determines the degree of matching between the predicted fire line and the real fire line. Both FS and FF boxes will be evaluated for a large group of input parameter sets in order to exploit the inherent parallelism of some optimisation techniques (OPT box). The optimisation process will be repeated until a "good" solution is found or until a predetermined number of iterations has been reached. At that point, an "optimal" set of inputs should be found, which will be used

as the input set for the fire simulator, in order to obtain the position of the fire front in the very near future. Obviously, this process will not be useful if the time incurred in optimizing is superior to the time interval between two consecutive updates of the real-fire spread information. For this reason, it is necessary to accelerate the optimization process as much as possible.

For optimization purposes, we used the optimization framework called BBOF (Black-Box Optimization Framework) [10], which consists of a set of C++ abstract-based classes that must be re-implemented in order to fit both the particular function being optimized and the specific optimization technique. BBOF uses the master-worker programming paradigm working in an iterative scheme. In particular, since the evaluation of the function to be optimized for each guess is independent, the guesses can be identified as the work (tasks) done by the workers. The responsibility for collecting all the results from the different workers and for generating the next set of guesses by applying a given optimization technique will be taken on by the master process. For the purposes of master-worker communication, BBOF uses the MPICH (Message Passing Interface) library [9]. In our case, the master will execute a GA [2] as optimisation strategy and will also be in charge of distributing the guesses of the candidate solutions to the workers. The workers themselves will execute, the underlying fire-spread simulator and will also provide the prediction error (evaluation of the fitness function).

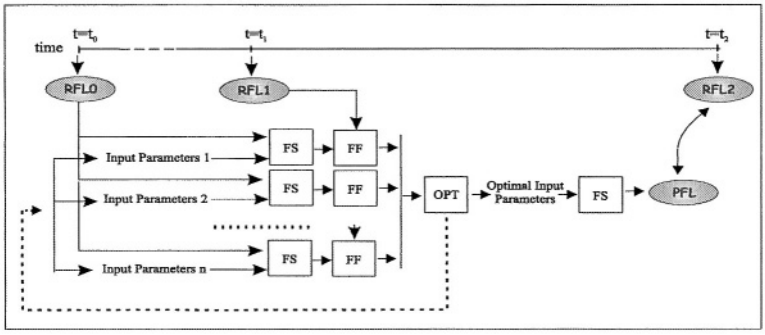


Fig. 1. Enhanced wildland fire prediction method on a master-worker scheme.

Since our objective consists of finding the combination of input parameters that minimizes the deviation of the simulator prediction from the real scenario as fast as possible, we need to compare the simulated fire lines against the real fire line and, according to the results of this comparison, assign a quality measurement to the underlying scenario. Each fireline describes a burned area. To compare the simulated and the real firelines we used the area of the XOR between the real and simulated burned areas (FF box). This XOR includes the areas that are burned in one of the propagations but not in the other one. If there is a perfect match between the real and simulated fire, this XOR is zero.

The optimization technique (GA) will be iterated until either a preset number of iteration is executed (1000 in our case). In each iteration a generation size of 20 individuals have been used.

3 Sensitivity Analysis

Sensitivity Analysis (SA) classically aims to ascertain how the model depends upon the information fed into it (input model/simulator parameters) [5]. The method we used here is based on nominal range sensitivity analyses, which are also known as local sensitivity analysis or threshold analysis [5]. Basic nominal sensitivity analysis evaluates the effect on the model output exerted by individually varying only one of the model inputs across its entire range of possible values, while holding all other inputs at their nominal or base-case values. The difference in the model output due to the change in the input variable is referred to as the sensitivity or swing weight of the model to that particular input variable, in that given case. However, there may be interdependencies among the parameters. The nominal sensitivity analysis must therefore be repeated for each parameter for all possible cases and combinations of all the other parameters.

The sensitivity of the parameters, in our case, depends on the fire propagation model used in the core of the objective function. For a generic study, we studied the effect of the parameters of the model in one dimension on propagation speed, thus the wind has only one scalar value, which is the speed of the wind in the direction of the fire propagation, however, in the experimental study both wind speed and direction has been considered because the fire simulator works with a two dimensional space.

Let V_{ik} be the effect of varying factor i from its minimum to its maximum (difference of the speed of the minimum and the speed of the maximum) at case k. The total effect of parameter i is defined as follows:

$$V_i = \sum V_{ik} \tag{1}$$

where k is all the possible cases (combinations of input factors). Thus, V_i will be our index of sensitivity for the parameter i. This index not only reflects the effect of the parameter but also the effect of its range. Higher parameter ranges mean greater uncertainty in the measurement of that parameter.

Table 1 outlines each one of these parameters and their corresponding minimum and maximum values according to [6], also showing the calculated index. Using the value of the index, we can classify input parameters by their sensitivity. This table shows that the two most important parameters are the load parameters (W0, β); the third is wind speed (U), followed by humidity (Mf). The parameters with weakest effect are metal content (St, Se) and heating content (h). This result concords with the results obtained by [7], which also uses the Rothermel set of equations as a forest fire propagation model.

Table 1. Ranges used to calculate the sensitivity index

parameter	W0	β	σ	St	Se	Mx	Mf	h	U
Min	0,1	0,01	315	0,001	0,0001	0,1	0	18571429	0
Max	4	0,11	11500	0,08	0,07	0,4	Mx	22000000	15
Index	0,77	0,86	0,56	0,03	0,16	0,28	0,61	0,13	0,71

Since sensitivity analysis implies a high number of simulations, we have also used the master/worker programming paradigm to evaluate all sensitivity indexes.

4 Experimental Study

The experimental study was carried out on a Linux cluster composed of 21 PC's with Intel Celeron processor 433 MHz, each one with 32 MB RAM, connected to a Fast Ether Net 100 Mb.

To properly evaluate the XOR area after executing the ISStest simulator for each guess, we need to have a reference fire line for comparison. For this purpose, a synthetic fire line was obtained by setting the values of all inputs to certain known values. We assumed homogeneous vegetation through the whole terrain, which consisted of a flat area. Once the synthetic real fire line was obtained, it was dismissed and was only used as a comparative member for calculating the XOR during the optimization process.

As we have commented, a GA was used as the optimization technique. Since the genetic algorithm has random factors, the global optimization process was performed 10 times and all results were averaged. All the values reported in this study therefore correspond to the mean values of the corresponding 10 different experiments conducted.

As we have commented, we are interested in calibrating the parameters that have a greater sensitivity index while we do not know their real values. Since calibrating the parameters that have little effect on the result will not improve the simulator results significantly, it is not worth to spent processing time on tuning them. We suppose that fewer parameters to be optimized will make the convergence faster and, at the same time, fixing certain unimportant parameters to a given value with a reasonable error will not deviate the optimization process too far from the global minimum, because of the reduction of the search space size.

This experiment is designed to observe the effect of removing the parameters that have a small sensitivity index on the convergence of the optimization process.

As estimated values for the parameters that are to be fixed, we have their real values plus 10% of its full range. Table 2 shows the real value of the less sensitive parameters, and their corresponding estimated values, when applying this estimation error (10%).

Table 2. The real and estimated values of the fixed parameters.

Parameter	Real value	Estimated value
St	0.04	0.04799
h	18971429	19314270
Se	0.02	0.0269
Mx	0.3	0.33

Figure 2(a) shows the convergence of the optimizing process by reducing the number of optimized parameters. Each curve differs from the other by omitting one parameter each time. We applied statistical hypothesis testing [8] to the results in order to asses whether or not the two observed behaviors can be considered statistically different. We can observe in figure 3(a) that at iteration 500, in general, for each case (fixing certain number of parameters) the corresponding mean value lays inside the confidence intervals of the other cases. That means that there is no statistical evidence that the differences between the means due to fixing some parameters at that iteration are relevant. However, figure34(b) shows that the means

have a trend to diverge one from the other at iteration 1000. In the same way we have applied the statistical hypotheses testing for the iterations 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000. We found that there is no statistical difference between the means before iteration 500; consequently, it is irrelevant to discus the behavior of the curves during the first phase of the optimization process. However, at iteration 1000, the results show a statistical difference between optimizing all parameters as opposed to fixing 1, 2, 3 and 4 parameters because the mean value for the case '10 parameters' clearly lays out of the other confidence intervals. Furthermore, we can observe that, as the number of parameters fixed increases, a statistical difference also appears (i.e., 8 and 9 versus 6 and 7). The mean values of the objective function (XOR area) at the end of the optimization process (iteration 1000) is shown in figure 2(b). As we can see, the objective function for the case of '6 parameters' is one third of the mean value obtained for the case of '10 parameters'. Table 3 shows the search space reduction for the cases plotted in figure 2(a). As we can observe, the search space reached a size reduction of one third from the '10 parameters' case to the '6 parameters' case. This reduction is directly related to the improvement obtained in the objective function (one third, approximately, in both cases).

Therefore, since the statistical study has shown that the convergence improvement for the case of '6 parameters' is not a matter of chance, we can conclude that reducing the search space is a good policy to speed up the optimization convergence.

However, we should bearing in mind that, these results are obtained using an error of estimation equal to 10%. If the error is greater, the practice of fixing the value of the parameters to estimated values will not be good. This method therefore assumes a "good" estimation of the real parameter value.

Table 3. Search space size for different number of parameters.

N° parameters	10	9	8	7	6
Search space size	1E+77	1E+69	1E+61	1E+53	1E+45

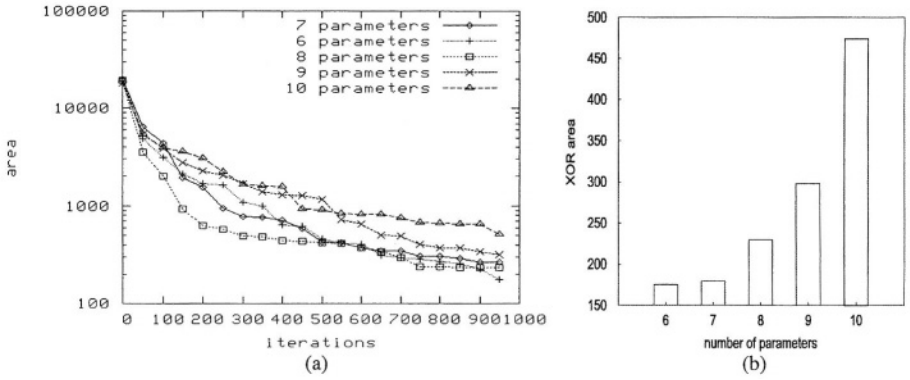


Fig. 2. Optimization Convergence Changing the Number of Parameters (a) and the mean value of the XOR area at iteration 1000 (b)

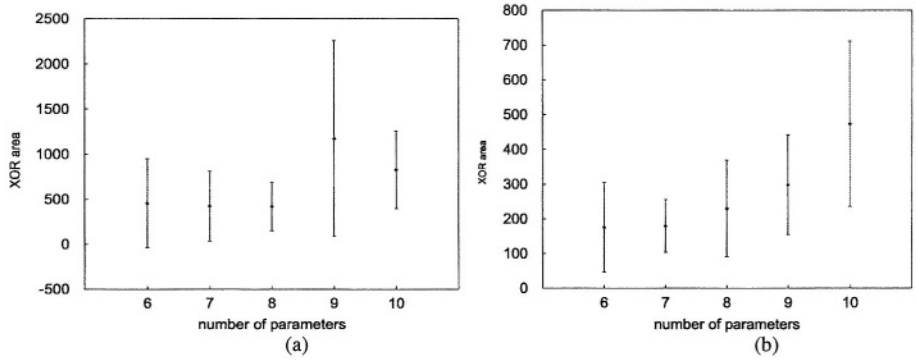


Fig. 3. 95% confidence intervals at iteration 500 (a) and 1000 (b).

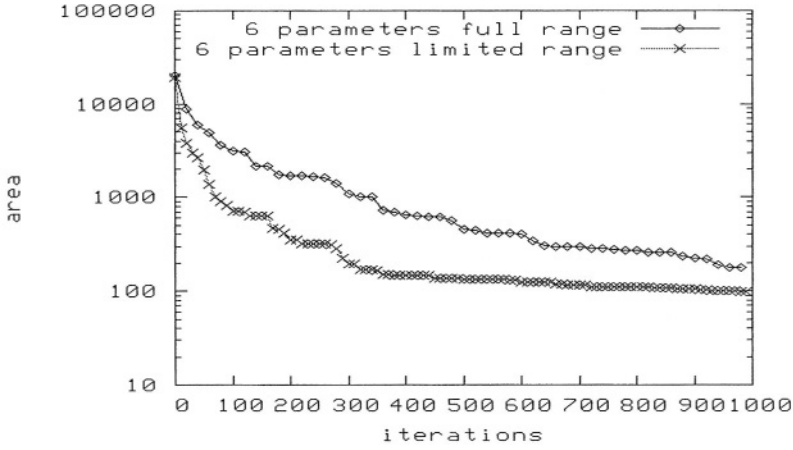


Fig. 4. Optimization convergence comparison using both the full and limited ranges.

Once we have observed that fixing 4 parameters to a certain estimated value provides a considerable improvement in optimization convergence, we focus on this case to introduce a certain degree of knowledge of the optimized parameters in order to further improve such convergence. We assume that we have some knowledge about the limits within which a parameter can vary, therefore it is not necessary to search within its full possible range. For the purpose of this experiment, we limited the range of the parameter to 15% above and below its "known value" so as to simulate the expected range. Figure 4 shows the optimization convergence when optimizing 6 parameters using either their full range or a limited searching range. As we can observe, cutting the range of the parameters significantly accelerates optimization convergence. Although from the figure it seems that, at iteration 1000, both situations provide similar results, the limited range at the end of the optimization process provides an objective function (XOR area) equal to 98.71, on average, whereas the final value is 175.47, using the full range.

5 Conclusions

One of the most common sources of fire spread simulation deviation from real fire propagation is imprecision in input simulator parameters. This problem can be approached by applying an evolutionary optimization such as the Genetic Algorithm so as to calibrate input-simulator parameters. Since this approach is a time-demanding task, we have proposed a global sensitivity analysis to accelerate optimization convergence. This technique reduces the search space screened by fixing the less sensitive parameters to an estimated value and by focusing optimization on the most sensitive parameters. We have also reduced the range of each optimized parameter by introducing some degree of knowledge of each of them. This was considered by limiting the variation of these parameters around a known value (field measurement). Both techniques were carried out on a Linux cluster composed of 21 PC's. We used a master/worker programming paradigm, where the master and worker processes communicate with each other using MPI. The results show that, combining both accelerating strategies, the convergence improvement obtained is quite significant.

References

1. Baker Abdalhaq, Ana Cortés, Tomàs Margalef, Emilio Luque, "Optimization of Fire Propagation Model Inputs: A Grand Challenge Application on Metacomputers". LNCS 2400, pp. 447-451. (2002).
2. Coley David A.:" An Introduction to Genetic Algorithms for Scientists and Engineers", World Scientific, 1999.
3. Jorba J., Margalef T., Luque E., J. Campos da Silva Andre, D. X Viegas "Parallel Approah to the Simulation Of Forest Fire Propagation". Proc. 13 Internationales Symposium "Informatik fur den Umweltshutz" der Gesellshaft Fur Informatik. pp. 69-81, (1999)
4. Rothermel, R. C, "A mathematical model for predicting fire spread in wildland fuels", USDA FS, Ogden TU, Res. Pap. INT-115, (1972).
5. Satelli, A., K. Chan, M. Scott, Editors. "Sensitivity analysis". John Wiley & Sons publishers, Probability and Statistics series. (2000).
6. André, J. "Uma Teoria Sobre a propagaçao de Frentes de Fogos Florestais de Superficie". PhD. Dissertaion. Coimbra. (1996).
7. Salvador, R., Piñol, P, Tarantola, S. And Pla, E. "Global Sensitivity Analysis and Scale Effects of a Fire Propagation Model used Over Mediterranean Shrub lands". Elsevier, Ecological Modelling 136 pp. 175-189, (2001).
8. Wadsworth, Harrison M. "Handbook of statistical methods for engineers and scientists", McGraw.Hill, Inc. (1990).
9. W. Gropp and E. Lusk and N. Doss and A. Skjellum,"A high-performance, portable implementation of the MPI message passing interface standard", "Parallel Computing", volume22-6, pp.789-828, sep,1996.
10. Baker Abdalhaq, Ana Cortés, Tomàs Margalef, Emilio Luque: "Evolutionary Optimization Techniques on Computational Grids.", International Conference on Computational Science, pp. 513-522, (1) 2002.

High Precision Simulation of Near Earth Satellite Orbits for SAR-Applications

Marc Kalkuhl[1], Katharina Nöh[1], Otmar Loffeld[2], and Wolfgang Wiechert[1]

[1] University of Siegen, FOMAAS,
[2] University of Siegen, ZESS
Paul-Bonatz-Str. 9-11, D-57068 Siegen, Germany
wiechert@simtec.mb.uni-siegen.de
http://www.simtec.mb.uni-siegen.de

Abstract. In a high resolution SAR-satellite mission it is very important to predict the flight trajectory of the satellite with a very high precision. Typically a SAR-satellite operates in near earth orbits. At this altitude the orbit is influenced by several physical phenomena (gravity, atmosphere, and air friction). The model to simulate these orbits should represent these influences sufficiently well. A detailed model has been build on the basis of the newest CHAMP gravity model and the MSIS86 atmosphere model. As a result of the complexity and the mathematical characteristics of this model it is necessary to implement a high order ODE-solver. Three different kinds of error are investigated to evaluate the predictive power of the model: numerical error due to the approximation scheme, power series truncation error on the gravity model, and statistical error due to parameter sensitivity. To this end an extensive sensitivity analysis of thousands of model parameters is carried out.

1 Introduction

Elementary models for satellite orbits or celestial mechanics based on Newtons Law of gravitation are well known text book examples for simulation courses and numerical mathematics [1]. They assume an ideal model of the gravitational field with spherical equipotential surfaces whereas air friction is neglected. In contrast the modeling and simulation of real satellite orbits based on realistic gravitational fields and including air friction is still a problem that is not satisfactorily solved.

The development of high precision orbit models becomes necessary when satellites aviate in low altitudes of less than 1000 km and additionally a very precise tracking of the satellite flight trajectory is required. An important application domain is the synthetic aperture radar (SAR) technique for remote sensing of the earth surface [2]. If it is driven in the interferometric mode SAR can even produce three dimensional maps with a height resolution of some few meters. However, in order to achieve this high resolution the position of the SAR sensor and receiver must be known with a high precision. The baseline which is a vector and reflects the difference in the sensor positions should be precisely

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 228–235, 2004.
© Springer-Verlag Berlin Heidelberg 2004

known, i.e. depending on the geometry of the configuration with an error smaller than a few millimeters (length) and a few (or even a fraction of an) arcsec (roll angle). An error of 1 millimeter in this difference results in a meter difference in the height resolution of the map [3].

This precision can only be achieved by a combination of direct position measurements with flight trajectory computations. Typically GPS signals are used for navigation. However, it is a well known fact that even the best available GPS-system (DGPS - Differential GPS) has only a precision of 1-6 meters. But in use with SAR-satellites this system is not the first choice for reasons that can't be discussed in this paper. All other GPS-systems have a precision in a magnitude of 10 meters. Thus, measurements are taken in regular time intervals in order to obtain redundant position information. At the same time the flight trajectory between two measurements is predicted by a mathematical model. The fusion of both sources of information is done by extended Kalman filter algorithms with alternating trajectory prediction and position update steps [3]. Clearly, the better the underlying trajectory model of the Kalman filter is the better will be the quality of the position estimate. The central question of this contribution is how precise a near earth satellite orbit can be predicted with state of the art knowledge about gravitation and air friction.

2 Coordinate Systems

Consulting the literature on the different physical influences on the satellite orbit (gravitation, atmosphere, air-friction) it turns out that models from different sources (geodesy, meteorology) are formulated for different reference coordinate systems. Here the coordinate systems given in Table 1 are relevant. Details about these coordinate systems and their precise definition can be found in [3].

Table 1. Used Coordinate Systems. Letters x (point of intersection between 0° longitude and equator), z (point at the North Pole), y (completion of the Cartesian System) indicate Cartesian Coordinates. Letters λ (longitude), δ (latitude) and r (distance between point of origin and emission point) resp. h (height above the earth surface) signify the coordinates in (modified) Spherical Coordinate Systems.

Name	Shortcut	Coordinates	Characteristics
Inertial System	IS	$x_{\mathrm{IS}}, y_{\mathrm{IS}}, z_{\mathrm{IS}}$	stationary clockwise Cartesian CS
Earth Coordinate System	ECS	$x_{\mathrm{ECS}}, y_{\mathrm{ECS}}, z_{\mathrm{ECS}}$	earth-bound clockwise Cartesian CS
Geocentric Coordinate System	GCS	$\lambda_{\mathrm{GCS}}, \delta_{\mathrm{GCS}}, r_{\mathrm{GCS}}$	earth-bound modified Spherical CS related to ECS
World Geodetic System	WGS	$\lambda_{\mathrm{WGS}}, \delta_{\mathrm{WGS}}, h_{\mathrm{WGS}}$	earth-bound modified Spherical CS represents earth-ellipsoid

Explicit formulas are available to transform each coordinate system into the other. Figure 1 shows the different coordinate transformations. Some of them have well established names in the literature. By applying the chain rule each coordinate system can be transformed into each other. All these coordinate transformations have been implemented in MATLAB.

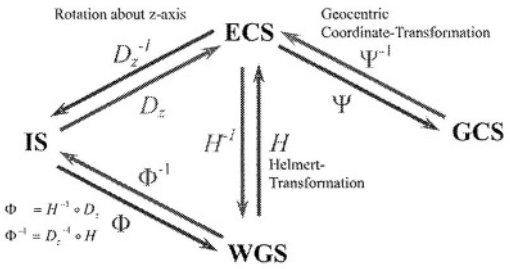


Fig. 1. Transformations between the different used Coordinate Systems.

3 Gravitation Model

Several gravitational influences must be taken into account for high precision flight simulation [4]:

- the gravitational field of the earth,
- the direct gravity influence of the moon, sun and planets, and
- the indirect gravity influence of the tide.

For simplicity the following remarks concentrate on the earth gravitational field without any time dependent influences on the flight trajectories.

The earth's gravitational field is described by a potential function $U(\lambda, \delta, r)$ where λ, δ, r are given in GCS coordinates (see Table 1). As an example the classical Newton potential is given by $U = \mu/r$ with μ - gravitational constant and r - distance between point of origin and emission point.

The most recent and most detailed gravitational model also takes the variation of the gravitational potential in dependency of the longitude and latitude into account. The CHAMP model of the GeoForschungsZentrum Potsdam [5] is based on high precision measurements with the CHAMP-satellite [6]. Here the normalized Legendre functions $\bar{P}_{n,m}$ are used for a series development:

$$U = \sum_{n=0}^{\infty} \frac{\mu \cdot a_e^n}{r^{n+1}} \cdot \sum_{m=0}^{n} \bar{P}_{n,m}(\sin(\delta)) \cdot (c_{nm} \cdot \cos(m \cdot \lambda) + s_{nm} \cdot \sin(m \cdot \lambda)) \quad (1)$$

with μ - gravitational constant, a_e - earth semi-major axis, $\bar{P}_{n,m}$ - normalized Legendre function and c_{nm}, s_{nm} - harmonic coefficients, r, λ, δ - see Table 1.
In general the precision of the gravitation model can be increased by adding further terms to the series development. Currently the parameters are available up to the 120th degree (e.g. $n = 0, 1, ..., 120$) [5]. For example there are 6642 coefficients at level 80 and 14762 for degree 120.

4 Atmosphere Model

Earth atmosphere is subdivided into several layers from which the thermosphere (90-500 km) and the exosphere (> 500 km) are important in this context. Satellite orbits are influenced by air friction up to a height of about 1000 km. Remote sensing satellites typically operate in near earth orbit (e.g. ERS in 750 km altitude [4]).

In contrast to gravitation the earth atmosphere is subject to several time dependent influences like the seasons and the sun activity. Frequently the MSIS model [7] is used to calculate the atmospheres density. The MSIS model family is formulated in the WGS coordinate system. The MSIS86 model used here takes the following parameters into account:

D	date	STL	local apparent solar time
UT	universal time	$F107A$	3 month average of F10.7 flux
h	altitude above earth surface	$F107$	daily F10.7 flux for prev. day
δ	geodetic latitude	Ap	magnetic index
λ	geodetic longitude		

This model can be obtained as a FORTRAN source code from [8]. Its details are much too involved to present them in this paper.

5 Air Friction Model

In addition to the air density the geometry of the satellite determines the total deceleration due to air friction. In particular the front surface of the satellite depends on its current orientation relative to the flight trajectory. In the used model a constant surface is assumed and the total deceleration is assumed to be proportional to the square of the current velocity which is a common model for very low gas densities close to molecular flow conditions:

$$\vec{d} = -\frac{1}{2} \cdot c_D \cdot \frac{A}{m} \cdot \rho \cdot V_a \cdot \vec{V}_a$$

with m - mass of satellite, ρ - air density, c_D - drag coefficient, A - aerodynamic active satellite cross sectional area, V_a - norm of track speed vector, $\vec{V}_a$ - track speed vector relative to atmosphere.

6 Flight Trajectory Model and Implementation

In total there are three sets of parameters in the present model which influence the flight trajectory:

- degree and coefficients of the gravitational model,
- coefficients of the air density model, and
- satellite geometry coefficients.

The dynamics of a satellite can now be described by combining gravitational and atmospheric influences into Newtons law of motion:

$$m \cdot \ddot{\vec{r}} = F_{grav.} + F_{fricl.} = m \cdot \nabla U + m \cdot \vec{d}. \tag{2}$$

Notice, that the two forces must be given with respect to the same coordinate system which is chosen here to be the GCS coordinate system. The chain rule must be applied to compute the gradient ∇U in the GCS system. More details can be found in [9].

The general model (2) has been implemented in MATLAB. To this end the coefficients of the gravitation model are imported from a file obtained directly from the GFZ Potsdam [5]. The MSIS86 model has been converted from the FORTRAN code to a C code by using the f2c translator and then coupled to MATLAB via the MEX-interface.

It turned out to be very important to implement the computation of the normalized Legendre function $\bar{P}_{n,m}$ in a numerical stable way. Therefore a recursion formula [9] should be used instead of the explicit formula, which is subject to strong numerical errors and also much slower to compute.

Because the resulting ordinary differential equation system (2) has only 6 dimensions and stiffness is not expected the high order explicit integration algorithm DOPRI8 with step size control and dense output [10] has been implemented in MATLAB. It is currently one of the most frequently used high order Runge-Kutta methods. The average time to simulate one full orbit of a satellite in 800 km altitude with the CHAMP model of degree 40 was about 30 seconds on an AMD 1200MHz computer.

7 Numerical Precision

Before the influence of the parameter uncertainties on the prediction of the flight trajectories can be evaluated it must be known how precise the numerical solution of the differential equation is. To this end a series of test runs was undertaken.

A good test case can be produced by omitting the friction force from equation (2). A first test was the classical Newton model for which exact solutions are available. If the starting conditions of the system are chosen appropriately the flight trajectory is known to be an ellipse [1]. Thus the closing of the trajectory can be checked numerically. The absolute position deviation in the IS system after one full orbit is less than 2 millimeters in each coordinate. Also omitting the friction force in the second test case the system must be conservative with respect to mechanical energy [1]. The test investigates the change in the computed energies in relation to the energy at the starting position for simulations with the classical Newton model and CHAMP model of degree 40 for one full orbit. It turned out that for the classical Newton model a numerical precision of 10^{-13} is obtained, whereas for the most complex model only a numerical precision of at least four digits is achieved.

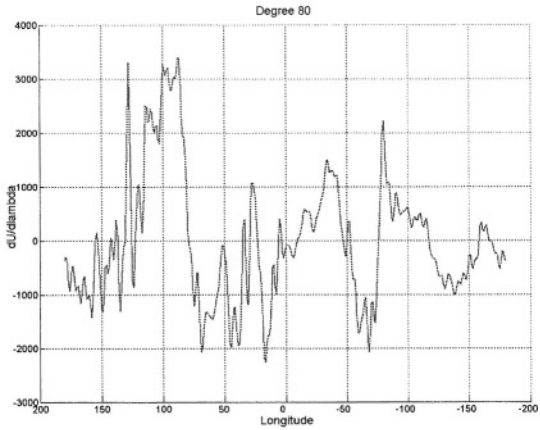


Fig. 2. Smoothness of the right hand side of the differential equation system. Here the longitude component of ∇U (degree 80) for 0° latitude is shown.

In order to understand the obvious difference between Newton and CHAMP model a closer look is be taken to the smoothness of the right hand side of the differential equation system (2). Clearly, the higher the degree of the CHAMP model is the more high frequent sine and cosine terms are included in (1). This increasingly reduces the smoothness of the model. Figure 2 shows the longitude component of ∇U for degree 80 in the right hand side of the ODE for 0° latitude. In contrary to the classical Newton model the real gravitational field is not very smooth. Thus algorithms with an excellent behavior for smooth systems like for example Taylor solvers need not perform well in this case.

8 Sensitivity Analysis

Having an estimate of the numerical precision it is now possible to carry out parameter variations in order to judge the model prediction uncertainty.

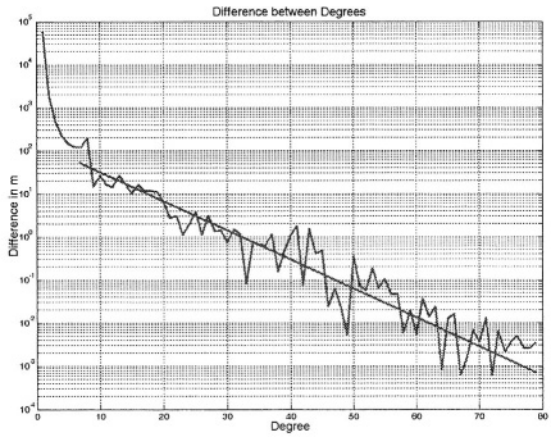


Fig. 3. Absolute deviation in space between two consecutive degrees.

To this end a rigorous sensitivity analysis was carried out for a sample trajectory with realistic starting conditions of a satellite in an altitude of 800 km. The parameters for the atmosphere and air friction model have been set to meaningful average values. Before the parameter sensitivities are computed it is tested how large the series truncation error is, when all parameters are assumed to be exact. Figure 3 shows the absolute deviation in space between two consecutive degrees. A log-linear decrease

can be extrapolated. As a result of this consideration it can be ascertained that the series truncation error of the gravitation model is in the magnitude of 7 decimeters for degree 40 and 1 millimeter for degree 80.

The quantities for which the sensitivities with respect to a parameter p were computed were the x, y and z position of the satellite after one full orbit around the earth. To obtain a more condensed measure the total deviation in space, that is

$$S = \sqrt{(\partial x/\partial p)^2 + (\partial y/\partial p)^2 + (\partial z/\partial p)^2}$$

was taken in the following figures. Sensitivities are computed from first order differential quotients which gives rise to 6642 different simulation runs for a gravitation model of degree 80.

Absolute sensitivities in general are not very meaningful in a practical application context because they have different physical units and thus cannot be directly compared to each other. In order to obtain comparable sensitivities each sensitivity is scaled with the precision σ_p of the respective parameter p:

$$\bar{S}^x = (\partial x/\partial p) \cdot \sigma_p, \quad \bar{S}^y = (\partial y/\partial p) \cdot \sigma_p, \quad \bar{S}^z = (\partial z/\partial p) \cdot \sigma_p, \quad \bar{S} = S \cdot \sigma_p.$$

Whereas the precision of the gravitational coefficients can be obtained from the GFZ institute, the precision of most other coefficients (air density, satellite geometry) have been roughly estimated. In case of doubt a pessimistic assumption was made (e.g. up to 30% relative tolerance for some atmosphere parameters). Figure 4 shows the results for the CHAMP model (degree 80) in combination with the atmosphere and air friction sensitivities. In order to condense the information all sensitivities of the parameters in the same polynomial degree were summarized by taking a mean value. Interestingly, the more detailed the gravitational model is the more imprecise the prediction becomes. This effect is based on the fact, that the deviation magnitude σ of the gravitation parameters c_{nm}, s_{nm} for degrees greater than 40 becomes equal resp. larger than the parameter himself. However, all these sensitivities are in the same order of magnitude and up

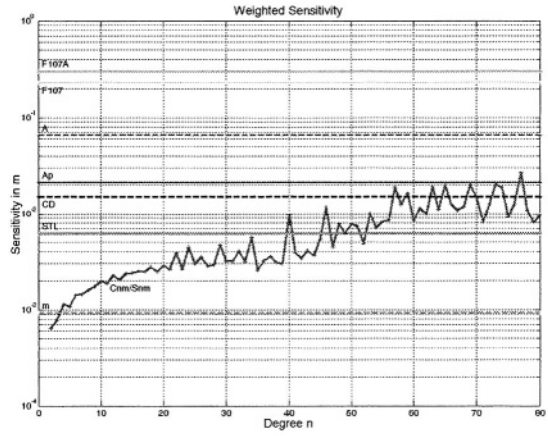


Fig. 4. Computed weighted sensitivities of the CHAMP (degree 80), atmosphere and air friction model parameters after one full orbit. The curve shows the sensitivity of the CHAMP model parameters. The horizontal lines display the different atmosphere (solid) and air friction (dashed) parameter sensitivities.

to approximately degree 65 are well below the truncation error of the gravitational model. Above this degree the sensitivities exceed the truncation error which means that the model causes more error with its parameters than the series truncation. In comparison the sensitivities of the air friction and satellite geometry parameters are also shown in Figure 4. It turns out that these sensitivities are mostly in the same order of magnitude as the gravitational parameters except for the two parameters $F107$ and $F107A$ which have a significantly higher influence. Unfortunately, these are exactly those parameters where only little knowledge is available today.

9 Conclusion

Assuming that the state of the art models for gravitation and air friction are correct representations of reality and the uncertainties in these parameters are well estimated the following conclusions can be drawn. According to the three kinds of error consideration the gravitation model need and should be computed to a maximal degree of 60 for the requirements of this contribution. A higher degree will not yield a higher precision. A consequence out of this is, that the computing time can be reduced compared to a model of higher degrees significantly and numerical errors can be avoided. In addition to that the sensitivity analysis gives also another important result: it's a prime importance to get the atmosphere parameters with a very high precision, because they have a great influence on the whole model.

Future investigation will be made on the consideration of other effects (e.g. moon or sun gravity) in the flight trajectory model. Also the enhancement of the model by reproducing the satellites geometry and inertia is intended.

References

1. Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration, Springer, 2002
2. Franceschetti, G., Lanari, R., Lanari, R.: Sythetic Aperture Radar Processing, CRC Press, 1999
3. Knedlik, S.: Auf Kalman-Filtern basierende Verfahren zur Erzielung genauerer Höhenmodelle in der SAR-Interferometrie, PhD. Thesis, University of Siegen, 2003
4. Montenbruck, O., Gill, E.: Satellite Orbits, Springer, 2000
5. GeoForschungsZentrum-Potsdam: `http://www.gfz-potsdam.de`
6. Reigber, Ch., Lühr, H., Schwintzer, P. (eds.): First CHAMP Mission Results for Gravity, Magnetic and Atmospheric Studies. Springer, 120-127, 2003.
7. Hedin, A.E.: MSIS-86 Thermospheric Model, J. Geophys. Res., 1987
8. MSIS86 model description and code download: `http://uap-www.nrl.navy.mil/models_web/msis/msis_home.htm`
9. Kalkuhl, M.: Erdnahe Orbitsimulation eines Interferometrischen Cart-Wheels, Diploma Thesis, University of Siegen, 2003
10. Hairer, E., Norsett, S.P., Wanner, G.: Solving Ordinary Differential Equations I, 1st edition, Springer, 1993

Hybrid Approach to Reliability and Functional Analysis of Discrete Transport System

Tomasz Walkowiak and Jacek Mazurkiewicz

Institute of Engineering Cybernetics, Wroclaw University of Technology,
ul. Janiszewskiego 11/17, 50-372 Wroclaw, Poland
{twalkow, jmazur}@ict.pwr.wroc.pl

Abstract. This paper describes a novel approach of combining Monte Carlo simulation and neural nets. This hybrid approach is applied to model discrete transportation systems, with the accurate but computationally expensive Monte Carlo simulation used to train a neural net. Once trained the neural net can efficiently, but less accurately provide functional analysis and reliability predictions. No restriction on the system structure and on a kind of distribution is the main advantage of the proposed approach. The results of reliability and functional analysis can be used as a basis for economic aspects discussion related to the discrete transport system. The presented decision problem is practically essential for defining an organization of vehicle maintenance.

1 Introduction

Modern transportation systems often have a complex network of connections. From the reliability point of view [2] the systems are characterized by a very complex structure. The main issue of reliability considerations is to model the influence of these faults at a satisfactory level of detail. This analysis can only be done if there is a formal model of the transport logistics, i.e. there are deterministic or probabilistic rules on how the transport is redirected in every possible combination of connection faults and congestion. The classical models used for reliability analysis are mainly based on Markov or Semi-Markov processes [2] which are idealized and it is hard to reconcile them with practice. The typical structures with reliability focused analysis are not complicated and use very strict assumptions related to the life or repair time and random variables distributions of the analysed system elements. The proposed solution is to use a time event simulation with Monte Carlo analysis [1], [5] to train a neural net. Once trained, the neural net can efficiently provide functional analysis and reliability predictions. One advantage of this approach it supports the computation of any point wise parameters. However, it also supports estimating the distributions of times when the system assumes a particular state or set of states.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 236–243, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Discrete Transport System Model

The basic entities of the system are as follows: store-houses of tradesperson, roads, vehicles, trans-shipping points and store-houses of addressee and the commodities transported. An example system is shown in Fig. 1. The commodities are taken from store-houses of tradesperson and transported by vehicles to trans-shipping points. Other vehicles transport commodities from trans-shipping points to next trans-shipping points or to final store-houses of addressees. Moreover, in time of transportation vehicles dedicated to commodities could fail and then they are repaired. In general, a system does not need to be equipped with any trans-shipping points. However, all system configurations need at least: one store-house of tradesperson, one road, single vehicle and one store-house of addressee [6], [7].

2.1 Commodities

The media transported in the system are called commodities. Different commodities are characterized by common attributes which can be used for their mutual comparison. The presented analysis uses the capacity (volume) of commodities as such attribute. The following assumptions related to the commodities are taken: it is possible to transport n different kinds of commodities in the system and each kind of commodity is measured by its capacity.

2.2 Roads

A road is an ordered pair of system elements. The first element must be a store-house of tradesperson or trans-shipping point, the second element must be a trans-shipping point or store-house of addressee. Moreover, each road is described by following parameters: length, the number of vehicle maintenance crews (at a given time only one vehicle could be maintained by a single crew) and the number of vehicles moving on the road. The number of maintain crews ought to be understand as the number of vehicles which can be on a single road maintained simultaneously.

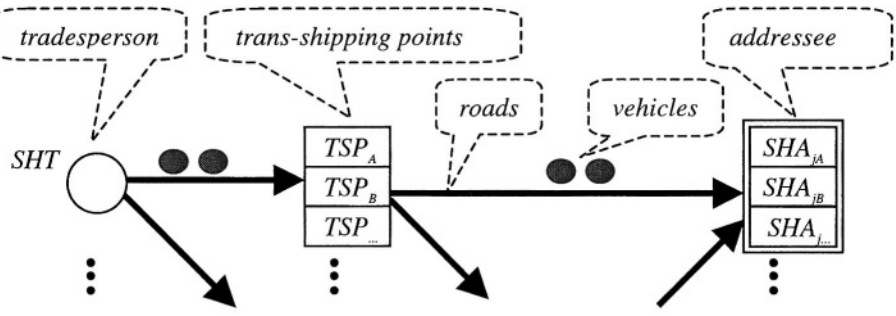


Fig. 1. Exemplar model of discrete transport system

2.3 Vehicles

A single vehicle transports commodities from the start to end point of a single road, after which the empty vehicle returns and the whole cycle is repeated. Our vehicle model makes the following assumptions. Each vehicle can transport only one kind of commodity at a time. Vehicles are universal – are able to transport different kinds of commodity. Moreover, the vehicle is described by following parameters: capacity, mean speed ofjourney (both when hauling the commodity and when empty), journey time (described by its distribution parameters), time to vehicle failure (also described an distribution), time of vehicle maintenance (described by distribution). The choice of distribution for the random variables is flexible provided that we know both a method and the parameters needed to generate random numbers with that distribution.

2.4 Store-Houses of Tradesperson

The store-house of tradesperson is the source of commodities. It can be only a start point of the road. Each store-house of tradesperson is an infinity source of single kind of commodity.

2.5 Trans-shipping Points

The trans-shipping point can be used as a start or end point of a single road. This is a transition part of the system which is able to store the commodity. The trans-shipping point is described by following parameters: global capacity *C,* initial state described by capacity vector of commodities stored when the system observation begins, delivery matrix ***D.*** This matrix defines which road is chosen when each kind of commodity leaves the shipping point (1 means that a given commodity is delivered to a given road). On contradictory to previously described systems ([6], [7], [8]) in this case a commodity could be routed to more then one road (direction). The dimensions of the delivery matrix are: number of commodities *x* number of output roads.

Input algorithm:

only one vehicle can be unloaded at a time, if the vehicle can be unloaded the commodity is stored in trans-shipping point, if not – the vehicle is waiting in the input queue, there is only one input queue serviced by FIFO algorithm.

Output algorithm:

only one vehicle can be loaded at a time, if the vehicle can be loaded, i.e. the proper commodity is presented in trans-shipping point, (a commodity which could be routed to a given road), the state of trans-shipping is reduced, if not – the vehicle is waiting in the output queue; each output road has its own FIFO queue.

2.6 Store-House of Addressee

The store-house of addressee can be used only as the end point of a single road. The main task of this component of the system is to store the commodity as long as the

medium is spent by recipient. The store-house of addressee is described by following parameters: global capacity *C,* initial state described as for the trans-shipping point, function or rule which describes how each kind of commodity is spent by recipients. Input algorithm is exactly the same as for trans-shipping point. Output algorithm can be described as: stochastic process, continuous deterministic or discrete deterministic one. The model assumes that the capacity of the commodity can not be less than zero, "no commodity state" – is generated when there is a lack of required kind of commodity (marked as τ on Fig. 2).

3 System Structure

The simulation program generates a description of all changes in the system during simulation (with all events). It is a base for calculation of any functional and reliability measures. The most valuable results of statistical analysis are: time percentage when the vehicle is present in each state, time percentage when the store-house of addressee is present in each state, mean time when the store-house of addressee is empty - this way we can say if "no commodity state" is prolonged or only momentary (Fig. 2.). We also propose a quantile calculation of time when the store-house of addressee is empty. This is the answer if "no commodity state" situation sometimes lasts significantly longer than the mean time of empty store-house. Moreover, it is possible to observe the influence of changes related to single parameter or a set of parameters – vehicle repair time for example – for other system characteristics – as vehicle utilization level, or commodity accessible in store-houses. The calculated reliability and functional measures could be a base of developing economic measures [8]. Such layered approach allows a high level, economic analysis of the system. It is necessary to check different variants of maintenance organization and to choose the less expensive among them if the reliability criteria are satisfied. It could be done by subsequent Monte-Carlo analysis and calculation of the required economic or functional measures for a set of analyzed parameters.

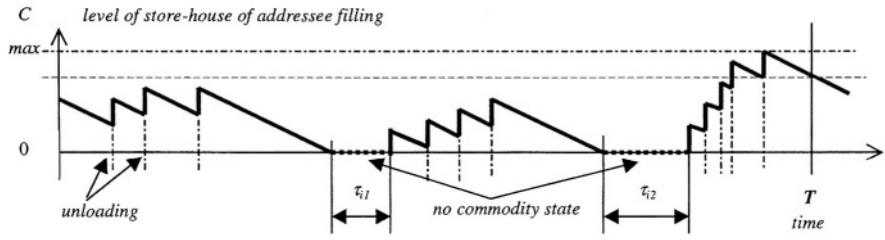


Fig. 2. Single store-house of addressee filling in time-period *T*

The system model described in previous sections is a subject of computer simulation. A special software package for simulation of the discrete transport system has been developed. The transport system is described in specially designed script language

(with syntax similar to XML) [4]. It is an input for simulator programme (written in C++) performing Monte-Carlo simulation [1], [5].

Monte Carlo simulation has an advantage in that it does not constrain the system structure or kinds of distributions used [4]. However, it requires proper data pre-processing, enough time to realize the calculations and efficient calculation engine.

4 Hybrid Approach

The problem of speeding up functional and reliability analysis of discrete transport system we propose to solve by hybrid system using simulation and neural nets. In many tasks, i.e. in decision systems, there is a need to give an answer in a short time. However Monte-Carlo simulation requires quite a lot of time to realize calculation for a given set of system parameters. To solve this problem we have proposed a use of artificial neural networks [9]. The use of neural network is motivated by its universal approximation capability [3]. Knowing that most of output system parameters are continues we can expect that neural network can approximate any unknown function based on a set of examples. The time needed to get an output from learnt neural network is very short. Solution generated by net seems to be satisfactory [9], because we do not need very precise results - time is the most important attribute of the solution. The neural network ought to substitute the simulation process. As it is presented in Fig. 3 the neural net module is added to developed simulation software. The aim of this module is to generate an answer how to select the best system parameters (i.e. the maintenance agreements - the average time of vehicle repair) based on the achieved system functional parameters (i.e. the average time of "no commodity" in the store-house of addressee).

The process of data analysis will be as follows:

1. set the input parameters for model of discrete transport system;
2. give a range of analyzed free parameter (parameters);
3. perform initial Monte-Carlo analysis for a few parameters from a given range - calculate all required functional and reliability parameters;
4. build a neural network classification tool:
   - use multilayer perceptron;
   - the input to the network are analyzed free parameters;
   - the outputs are functional and reliability measures;
5. build the answer about the maintenance agreement based on the output of the neural network and the proper economic measures;
6. communicate with a user:
   - play with functional and reliability data, goto 4.

   If more accurate analysis of economic parameter in a function of free parameter is required goto 3 – perform more Monte-Carlo analysis.

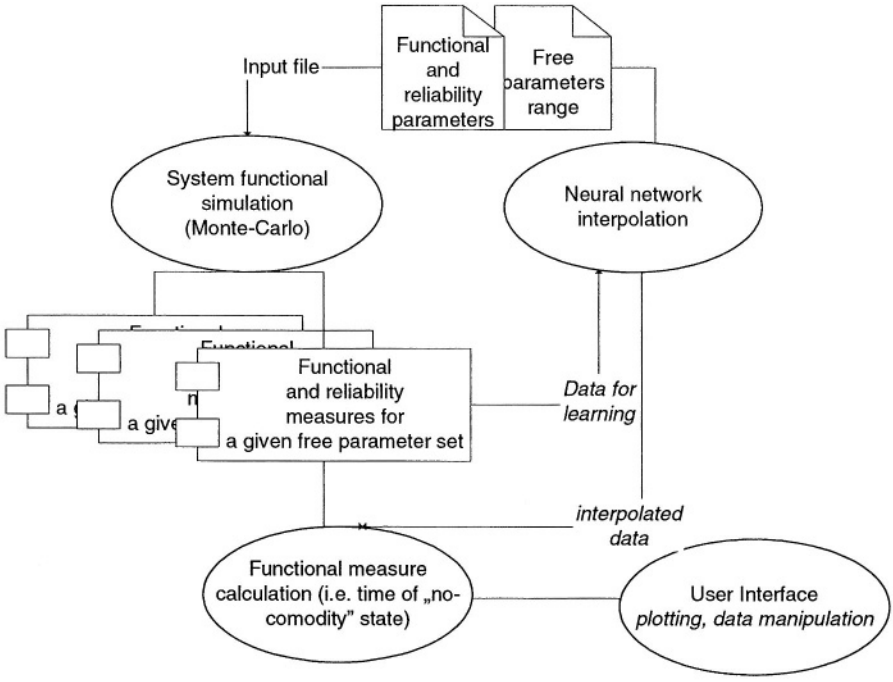


Fig. 3. Hybrid system overview

5 Case Study

To show possibilities of the proposed model and developed software we have analyzed an exemplar transport network presented on Fig. 4. The network consists of two store-houses of tradesperson (each one producing its own commodity, marked as A and B), one trans-shipping point (with one storehouse for both commodities) and two store-houses of addressee (each one with one storehouse). The commodities are spent by each recipient. The process is continuous deterministic as presented on Fig. 2, the amount of consumption in time unit is marked by u with subscripts corresponding to store-houses of addressee and commodity id. It's exemplar values are presented in Fig. 4. Having lengths of the roads (see Fig. 4), the amount of commodity consumption in time unit for each store-house of addressee, the capacity of each vehicle (15), vehicle speed (50 and 75 in empty return journey) the number of vehicles for each road could be easy calculated. We have take into account some redundancy [8] due to the fact of car failure (we assumed that the time between failures is 2000 time units) what results in following number of vehicles: road one n_1=40, road two n_2=12, road three n_3=18(A)+6(B)=24 and road four n_4=16(A)+8(B)=24. The analysis time T was equal to 20000.

We have analyzed maintains and service level agreement (SLA) dependency. From one side the transport network operator has to fulfill some service level agreement, i.e. have to deliver commodity in such way that a "no commodity state" is lower then a given stated level. Therefore the analyzed functional measure was a summary time of "no commodity state" during the analyzed time period. It could be only done if a proper maintenance agreement is signed. Therefore the argument of analyzed dependency was a average time of repair of vehicles. We assumed that we have four separated maintenance agreement, one for each for each road (roads 1 and 2 with one maintains crew, and 3 and 4 with two maintains crews). Also the exponential distribution of repair time was assumed. Therefore, we have four free parameters with values spanning from 1 to 1200. The system was simulated in 1500 points. For each repair time values set the simulation was repeated 25 times to allow to get some information of summary time of "no commodity" distribution. Two measures were calculated: average time of summary of "no commodity state" and its 4% quantile (i.e. the value of summary "no commodity" time that with probability 96% could be not higher).

The achieved date from simulation was divided randomly into two sets: learning and testing. We have used the multilayer perceptron architecture with 4 input neurons which correspond to repair time for each road, 10 hidden layer neurons and 2 output neurons.

The number of neurons in the hidden layer was chosen experimentally. Such network produced best results and higher numbers did not give any improvement. The tan-sigmoid was used as a transfer function in hidden layer and log-sigmoid output layer. Besides that, the output values have been weighted due to the fact the log-sigmoid has values between 0 and 1. The network presented above was trained using the Levenberg-Marquardt algorithm [3].

The achieved results, the mean of absolute value of difference between network results (multiplied by time range: 20 000) and results from simulation, for testing data set is 364 time units and 397 respectively for average time of summary of "no commodity state" and its 4% quantile. It is in range of 1-2% of analyzed transport system time. We have also tested the simulation answer stability, i.e. the difference between two different runs of simulation (25 of them each time) for both functional measures (average time of summary of "no commodity state" and its 5% quantile) is 387 time units in average.

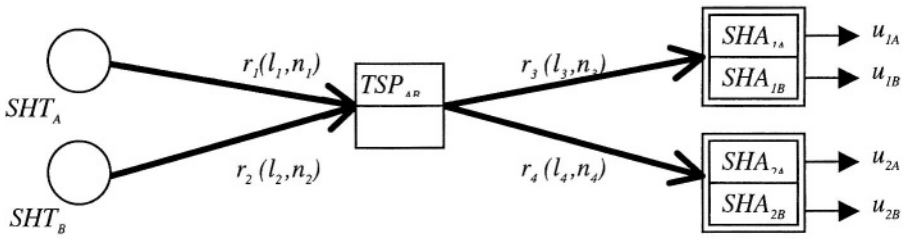


Fig. 4. Structure of case study discrete time system
(parameters: l_1=120, l_2=90, l_3=90, l_4=120, u_{1A}=60, u_{1B}=20, u_{2A}=40, u_{2B}=20)

6 Conclusion

Results of functional and reliability analysis of exemplar discrete transport system are very promising. Time necessary for whole neural network training is less (in average 4 times) then time necessary for a single training vector preparation (run of 25 simulations for a single set of free parameters). An error related to the network answer - when the already trained network is tested by the input data which are not used during training - is in the range of disperse related to results of simulation. Of course there is an important aspect of avoiding over fitting or under training by neural network. At this stage of work it was done manually by observing the global error in function of training epochs and stopping training when the curve stops to decrease.

The other interesting aspect of presented approach is the scalability projections. Increasing the number of modeled vehicles or system elements increases the Monte Carlo simulation time significantly. In case of training time of neural network (classification time is negligible) increasing a number of simulated entities has not direct influence. However, if one wants to analyze more sophisticated relation between input parameters and output measures, i.e. increases the number of input parameters, it results in an increase of input neurons, therefore needs a larger number of training data and results in a longer training time. Future work is planned on checking the extrapolation features of the neural network. We are going to analyze the answer of the network for input data with range outside the training set.

Acknowledgement. Work reported in this paper was sponsored by a grant No. 5 T12C 021 25, (years: 2003-2006) from the Polish Committee for Scientific Research (KBN).

References

1. Banks, J., Carson, J.S., Nelson, B.N.: Discrete-Event System Simulation, 2nd Edition. Prentice Hall, Upper Saddle River, NJ (1996)
2. Barlow, R., Proschan, F.: Mathematical Theory of Reliability. Society for Industrial and Applied Mathematics, Philadelphia (1996)
3. Bischop, Ch.: Neural Networks for Pattern Recognition. Clarendon Press Oxford (1996)
4. Caban, D., Walkowiak, T.: Computer simulation of discrete transport system (in Polish). XXX Winter School of Reliability, Poland, Szczyrk (2002) 93-103
5. Fishman: Monte Carlo: Concepts, Algorithms, and Applications. Springer-Verlag, New York (1996)
6. Jarnicki, J., Mazurkiewicz, J., Zamojski, W.: Model of discrete transport system (in Polish). XXX Winter School of Reliability, Poland, Szczyrk (2002) 149-157
7. Kaplon, K., Mazurkiewicz, J., Walkowiak, T.: Economic analysis of discrete transport systems. Risk Decision and Policy, Vol. 8, No. 3. Taylor & Francis Inc. (2003) 179-190
8. Kaplon, K., Walkowiak, T.: Economic aspects of redundancy in discrete transport systems (in Polish). XXXII Winter School of Reliability, Poland, Szczyrk (2004) 142-153
9. Mazurkiewicz, J., Walkowiak, T.: Neural Network for Reliability Parameters Analysis – Case Study. V Conference Neural Networks and Soft Computing, Poland, Zakopane (2000) 687-692

Mathematical Model of Gas Transport in Anisotropic Porous Electrode of the PEM Fuel Cell

Eugeniusz Kurgan and Paweł Schmidt

AGH University of Science and Technology, Department of Electrical Engineering,
al. Mickiewicza 30, 30-059 Krakow, Poland,
{kurgan,pschmidt}@agh.edu.pl

Abstract. In this paper a gas mixture model is developed to study anisotropic hydrogen and water vapour flow in anode of the PEM fuel cell. Dependence of the distribution of concentrations and fluxes of the of gas components in anisotropic porous layer is investigated. First full partial differential equations describing mass transport for permeability and diffusivity tensors based on Darcy's and Fick's laws are developed. Next this set of nonlinear equations together with appropriate nonlinear boundary conditions using finite element method was solved. At the end an illustrative example is given.

1 Introduction

The Proton Exchange Membrane (PEM) fuel cell consists of two gas diffusion layers (GDL) separated by PEM. Between each GDL and PEM thin platinum catalyst layer is located. Numerical simulation of all aspects of GDL performance is very important from practical point of view because, most of the working parameters is very difficult to measure. This is caused mainly by small physical dimensions of single cell. Typical cell electrodes are made of carbon fibre paper, consists of single carbon fibres. Because of this, GDL diffusion and convection coefficients are not constant numbers but tensor values.

One of the first publication on simulation of the PEM fuel cell based on the fluid flow approach, started after publication [1]. In this publication authors described equations governing gas distribution in one dimension in different regions of membrane-electrode assembly, the transport of mass species inside GDL and electrochemical reactions. We decouple GDL from the rest of the assembly by appropriate boundary conditions, occurring in the gas channel and on the catalyst layer.

Anisotropic properties of the GDL were investigated by many authors, but few publications with numerical models of anisotropic fuel cell were presented. In [2] authors simulate gas distribution in anisotropic porous electrodes but they do not show full equations which flow fields should fulfil. In this article authors present full equations describing gas distribution in anode of the PEM fuel cell and extend results presented in [2] to the full anisotropic case, where anisotropic properties of the material are described by full permeability and diffusivity tensors. At the end some illustrative example is given.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 244–251, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Numerical Model of the Anisotropic Electrode

At the Fig. 1 cross section of the anode of the PEM fuel cell is shown. Gas mixture is supplied by the inlet flow channel and distributed to the electrochemical reaction sites by the anisotropic porous layer. At the PEM electrolyte boundary, where very thin catalyst layer is present, hydrogen molecules H_2 are split into hydrogen ions H^+ and free electrons in electrochemical reaction. The hydrogen ions flow farther in direction of the cathode and electrons flow back to the interdigitated graphite plate, where graphite is carrying them out to the external load. Through the polymer electrolyte from the cathodic side to the anode-electrolyte boundary, water molecules are transported. In this publication we assume that temperature of the fuel cell is high enough for all this water to evaporate. Thus, it is further assumed that water occurs only as a vapour.

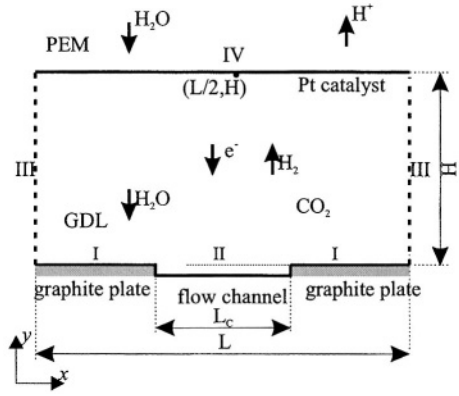


Fig. 1. Cross section of the PEM anode together with geometrical dimensions.

For practical reasons the hydrogen is obtained from hydrocarbons after catalytic reforming process. As a result in the gas mixture, with which cell is supplied, carbon dioxide is present. It is an inert gas but influences distribution of hydrogen and water vapour distribution, it should be taken into account during calculations.

Mass transport of the reactant gases obeys two fundamental laws, namely Darcy's and Fick's laws. The first law defines convection flow which is proportional to the pressure difference and the second diffusion flow which depends on concentration gradients. Both laws are correct for homogeneous and isotropic mediums, described by constant coefficients.

3 Mathematical Model of the Anisotropic Porous Layer

The overall flow of the gas mixture $\mathbf{N} = C\mathbf{U}$ is governed by the mass conservation law which in our case can be written in the form

$$\nabla(C\mathbf{U}) = 0 \tag{1}$$

where **U** is a gas molar average velocity and C is a gas mixture concentration. It is a sum of the concentration of all three species: $C = C_1 + C_2 + C_3$, where C_1 is concentration of hydrogen, C_2 is concentration of water vapour and C_3 is concentration of carbon dioxide. We assume further that CO_2 concentration is a fixed part of the mixture concentration C, and is equal $(1 - \gamma)C$. Molar average velocity is described by Darcy's law:

$$\mathbf{U} = -\frac{1}{\varepsilon_0 \mu} \bar{\bar{K}} \nabla P \tag{2}$$

where $\bar{\bar{K}}$ is a permeability tensor given by a matrix

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \tag{3}$$

Ideal gas law relates pressure with the gas mixture concentration C

$$P = CRT \tag{4}$$

Taking into account above relation and defining convection matrix $\mathbf{\Gamma}$ as

$$\mathbf{\Gamma} = \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix} = \frac{RT}{\varepsilon_p \mu} \mathbf{K} \tag{5}$$

the Darcy's Law can be formulated for the anisotropic case as

$$\begin{bmatrix} U_x \\ U_y \end{bmatrix} = -\begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix} \begin{bmatrix} \dfrac{\partial C}{\partial x} \\ \dfrac{\partial C}{\partial y} \end{bmatrix} \tag{6}$$

Introducing (6) into (1) we get the first partial differential equations governing the process of mass transport in the electrode

$$-\gamma_{11} \frac{\partial}{\partial x}\left(C \frac{\partial C}{\partial x}\right) - \gamma_{12} \frac{\partial}{\partial x}\left(C \frac{\partial C}{\partial y}\right) - \gamma_{21} \frac{\partial}{\partial y}\left(C \frac{\partial C}{\partial x}\right) - \gamma_{22} \frac{\partial}{\partial y}\left(C \frac{\partial C}{\partial y}\right) = 0 \tag{7}$$

Also hydrogen flux $\mathbf{N}_1 = C_1 \mathbf{U}_1$ has to fulfil mass conservation law:

$$\nabla \mathbf{N}_1 = 0 \tag{8}$$

Hydrogen flux consists of two parts: convective flux $C_1\mathbf{U}$ and diffusive flux $\mathbf{J}_1$:

$$\mathbf{N}_1 = C_1 \mathbf{U} + \mathbf{J}_1 \tag{9}$$

Convective flux is related with concentration gradients by Fick's law:

$$\mathbf{J}_1 = -C\bar{\bar{D}}\nabla x_1 \tag{10}$$

where tensor $\bar{\bar{D}}$ is given by matrix

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \tag{11}$$

Thus Fick's law for anisotropic case has following form:

$$\begin{bmatrix} J_{1x} \\ J_{1y} \end{bmatrix} = -C \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \begin{bmatrix} \dfrac{\partial x_1}{\partial x} \\ \dfrac{\partial x_1}{\partial y} \end{bmatrix} \tag{12}$$

Introducing equations (6) and (12) into (8) we get and eliminating C_1 by introducing $C_1 = x_1 C$ we get second partial differential equation describing mass transport in anode of the PEM fuel cell:

$$\begin{aligned} &-d_{11}\frac{\partial}{\partial x}\left(C\frac{\partial x_1}{\partial x}\right) - d_{12}\frac{\partial}{\partial x}\left(C\frac{\partial x_1}{\partial y}\right) - d_{21}\frac{\partial}{\partial y}\left(C\frac{\partial x_1}{\partial x}\right) - d_{22}\frac{\partial}{\partial y}\left(C\frac{\partial x_1}{\partial y}\right) - \\ &-C\left(\gamma_{11}\frac{\partial x_1}{\partial x}\frac{\partial C}{\partial x} + \gamma_{12}\frac{\partial x_1}{\partial x}\frac{\partial C}{\partial y} + \gamma_{21}\frac{\partial x_1}{\partial y}\frac{\partial C}{\partial x} + \gamma_{22}\frac{\partial x_1}{\partial y}\frac{\partial C}{\partial y}\right) = 0 \end{aligned} \tag{13}$$

Equations (7) and (13) form complete set of equation in total mixture concentration C and molar fraction of hydrogen x_1 variables.

4 Boundary Conditions

Equations (7) and (13) have to fulfil adequate boundary conditions. Each of them needs different types of Dirichlet's and Neuman's conditions for four dinstinct boundary sections.

Section I – Graphite Plate

Along this part of boundary graphite plate contacts with carbon fibre material and throughout this section can be no flux of any species. Because fluxes of all species at this boundary section are parallel to the boundary surface, the fluxes entering the graphite plate are equal zero.

$$\frac{\partial x_1}{\partial n} = 0 \quad \text{and} \quad \frac{\partial C}{\partial n} = 0 \tag{14}$$

Section II–Inlet Flow Channel
On this part of the boundary fibrous carbon material contacts with flow channel and reactant species enter the anode. This results following Dirichlet's conditions:

$$C = C_g \quad \text{and} \quad x_1 = x_{1g} \tag{15}$$

where C_g is given molar concentration of mixture and x_{1g} given molar fraction of hydrogen in inlet channel.

Section III – Left and Right Sides of the Calculation Domain
We assume that solution is periodic with respect to the calculation variables, and thus all fluxes are directed parallel to the *y* axis. This causes that boundary conditions in this section are the same as for section I.

Section IV – Platinum Catalyst Layer
Gas diffusion layer contacts with catalytic material along this part of the boundary. Here hydrogen atoms lose their electrons during electrochemical reaction. Those electrons flow back to graphite plate and next through external circuitry to the cathode. Hydrogen ions flow further through proton membrane. Here hydrogen flux crossing boundary is proportional to its molar concentration difference on both sides of boundary. We assume that electrochemical reaction efficiency is equal 100% and no hydrogen atoms enter proton membrane. Thus

$$\mathbf{N}_1 \cdot \mathbf{n} = N_1^n = r_H \left(C_1 - C_{1\text{out}}\right) = r_H \left(C_1 - 0\right) = r_H C_1 \tag{16}$$

where the mass transfer coefficient r_H models electrochemical reaction which takes place in catalyst layer. It relates hydrogen flux with its molar concentration and can be determined from averaged current density flowing throughout catalytic region.
The water in fuel cell is produced on the cathode side of the membrane and its majority is transported from this region to cathodic gas channel. However, some part of water diffuses throughout membrane to anode, crosses anodic catalytic layer and enters anode. This flux of water vapour is strictly related to hydrogen flux flowing in opposite direction, because from every two hydrogen atoms one water molecule is produced. This relation between hydrogen flux and water vapour flux is established by return coefficient *v* which determines direction and magnitude water vapour flux $\mathbf{N}_2$:

$$\mathbf{N}_2 \cdot \mathbf{n} = (\nu - 1)\mathbf{N}_1 \cdot \mathbf{n} = (\nu - 1) r_h C x_1 \tag{17}$$

Equations (16) and (17) are starting points for derivation Neuman's boundary conditions on catalyst layer for molar mixture concentration *C* and molar hydrogen fraction x_1. From (16) we get:

$$x_1 \gamma_{21} \frac{\partial C}{\partial x} + x_1 \gamma_{22} \frac{\partial C}{\partial y} + d_{21} \frac{\partial x_1}{\partial x} + d_{22} \frac{\partial x_1}{\partial y} = -r_h x_1 \tag{18}$$

Carrying out the same manipulations as for hydrogen from (17) we get relation between partial normal derivative for hydrogen and mixture concentration

$$(\gamma_r - x_1)\gamma_{21}\frac{\partial C}{\partial x} + (\gamma_r - x_1)\gamma_{22}\frac{\partial C}{\partial y} - d_{21}\frac{\partial x_1}{\partial x} - d_{22}\frac{\partial x_1}{\partial y} = -(\nu - 1)r_h x_1 \tag{19}$$

It is reasonable to assume that fluxes on this boundary enter catalyst layer perpendicularly and that no tangential fluxes are present. This means that partial derivatives of all calculation variables in directions tangential to the boundary are equal zero, thus, after simplification and solving system of equations, we get Neuman's boundary conditions for C and x_1 variables:

$$\frac{\partial C}{\partial y} = -\frac{\nu r_h x_1}{\gamma_r \gamma_{22}} \quad \text{and} \quad \frac{\partial x_1}{\partial y} = \frac{r_h x_1 (\nu x_1 - \gamma_r)}{d_{22}\gamma_r} \tag{20}$$

5 An Illustrative Example

The geometry of numerical example are defined in fig. 1 .Numerical values was presented in [3] and will not be repeated here.
Let us assume further that main axes of anisotropy overlap with geometrical coordinates. This results in following form of anisotropy for convective and diffusive matrixes

$$\mathbf{\Gamma} = \Gamma\begin{bmatrix} \chi_g & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \mathbf{D} = D\begin{bmatrix} \chi_d & 0 \\ 0 & 1 \end{bmatrix} \tag{21}$$

Let us further assume that both anisotropic ratios are equal $\chi_g = \chi_d = \chi$. For this simplified assumptions equations (7) and (13) will take following form:

$$\chi\frac{\partial}{\partial x}\left(C\frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(C\frac{\partial C}{\partial y}\right) = 0 \tag{22}$$

$$\chi\frac{\partial}{\partial x}\left(C\frac{\partial x_1}{\partial x}\right) + \frac{\partial}{\partial y}\left(C\frac{\partial x_1}{\partial y}\right) + \frac{C\Gamma}{D}\left(\chi\frac{\partial x_1}{\partial x}\frac{\partial C}{\partial x} + \frac{\partial x_1}{\partial y}\frac{\partial C}{\partial y}\right) = 0 \tag{23}$$

For anisotropic ratio $\chi = 1$ the above equations reduce to the usual equations for isotropic and homogeneous case [4]. The above set of mutually coupled and nonlinear equations was solved in two dimensions by finite element method.

The dependencies between fluxes of species and anisotropy ratio were investigated during the calculations. The calculations were carried out at point (L/2,H) in the middle of top domain boundary, across the GDL at y=H/2. In the fig. 2 and 3 the total flux of the hydrogen is shown for two highly different anisotropy ratios. We can see differences between directions of the fluxes caused by difference in anisotropic ratio of carbon fibres assumed for each calculation. In the figures 4 and 5 dependence

between anisotropic ratio and diffusion flux of hydrogen is shown. For anisotropy ratio $\chi = 1$, the numerical results are consistent with general equations [5].

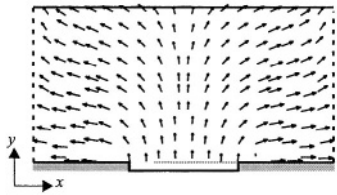


Fig. 2. Total flux of hydrogen $\mathbf{N}_1$ for anisotropic ratio $\chi = 0.2$.

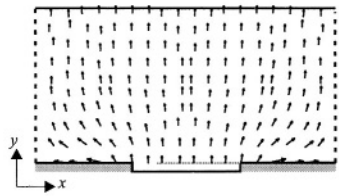


Fig. 3. Total flux of hydrogen $\mathbf{N}_1$ for anisotropic ratio $\chi = 1.8$.

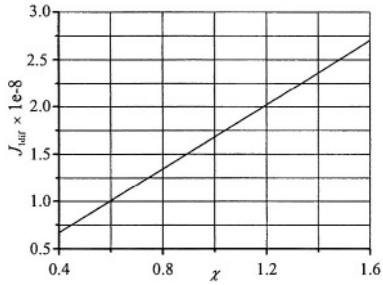


Fig. 4. Dependence between diffusion modulus of the flux of hydrogen J_{1diff} and anisotropic ratio χ at point, placed on middle point of the catalytic layer (L/2, H).

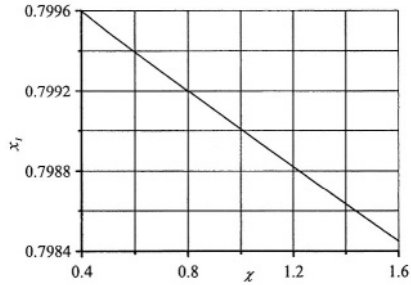


Fig. 5. Dependence between molar fraction of hydrogen x_1 and anisotropic ratio χ at calculation point, placed on middle point of the catalytic layer (L/2, H).

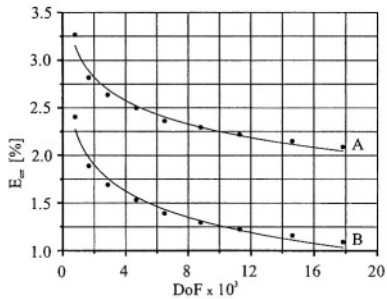


Fig. 6. Dependency between DoF and E_{err} coefficient for normal components of the water vapour flux $\mathbf{N}_2$ **(A)** and hydrogen flux $\mathbf{N}_1$ **(B)**.

In Fig. 4, dependence of modulus of the hydrogen diffusion flux is presented. In the middle of catalytic layer for $\chi < 1$ less hydrogen reaches middle parts of the cathode-electrolyte boundary. This means that distribution of hydrogen on the catalyst layer is more uniform. This in turn results in greater effectiveness of the electrochemical reaction of the splitting hydrogen molecules into hydrogen ions and electrons, what

gives greater electric cell performance. It is obvious that there exists an optimal value of the anisotropic ratio for which electrochemical reaction effectiveness on this electrode attains maximal value. To calculate this optimal value of χ one has to use any optimization method. In this publication this problem will be not further considered.

Convergence of the solution of the problem was investigated by analysing dependence between error coefficient E_{err} defined by (24) and degrees of freedom (DoF) of discretized form of problem equations. Analyse was done for normal component of fluxes of each gas component $\mathbf{N}_1$ and $\mathbf{N}_2$. Results of this investigation are shown in fig. 6. We can see, that increasing of DoF decreases percentage error coefficient, what means, that problem is convergent.

$$E_{err} = \frac{\oint_L \mathbf{N}_i \cdot d\mathbf{l}}{\oint_L |\mathbf{N}_i \cdot \mathbf{n}| \cdot d\mathbf{l}} 100\% \tag{24}$$

where $i = 1$ or 2.

6 Conclusions

In this paper we present a mathematical model of multicomponent gas transport in the GDL, which includes the full anisotropy of porous material. We assumed that anisotropy tensors for both convective and diffusive fluxes for hydrogen and water vapour are equal each to other. Generally it is not the case. Further work should be carried to take into account pressure as the third independent variable, and to include physical aspects of anisotropic coefficients and their mutual relation.

Acknowledgement. This work was supported by the AGH University of Science and Technology, under grant 11.11.120.183.

References

1. Bernardi D.M., Verbrugge , M.W.: Mathematical model of a gas diffusion electrode bonded to a polymer electrode, J. Electrochem. Soc. vol. 139, no 9 (1992) 2477–2490
2. Stockie J., Promislow K., Wetton B.: A finite volume method for multicomponent gas transport in a porous fuel cell electrode, Int. J. Numer. Methods in Fluids, vol. 41 (2003) 577 – 599.
3. Promislow K., Stockie J.: Adiabatic relaxation of convective-diffusive gas transport in porous fuel cell electrode, SIAM J. Appl. Math, vol. 62, no 1 (2001) 180 – 205
4. Kurgan E., Schmidt P.: Transport of Gas Components in the Cathode of PEM Fuel Cell, Sixth Int. Conference on Advanced Methods in the Theory of Electrical Engineering, Pilsen, Czech Republic, (2003) 5 – 10.
5. Kurgan E., Schmidt P.: Time Dependence of Species Concentrations in Cathode of The PEM Fuel Cell, The Second Polish-Slovenian Joint Seminar on Computational and Applied Electromagnetics, Kraków, (2003) 27 – 30

Numerical Simulation of Anisotropic Shielding of Weak Magnetic Fields

Eugeniusz Kurgan

AGH University of Science and Technology, Department of Electrical Engineering,
al. Mickiewicza 30, 30-059 Krakow, Poland,
kurgan@agh.edu.pl

Abstract. In this paper the method of computation of weak magnetic fields in the presence of anisotropic shields it is described. The formulation is based on vector magnetic potential and finite element formulation. Investigation of influence of anisotropic ratio on shielding effectiveness of low level magnetic fields is investigated. At the end some illustrative example in 3D is given and numerical results are compared with experimental data.

1 Introduction

In the last time there is an increasing interest in low-frequency magnetic shielding. Generally, the shielding effectiveness for low-frequency fields can be obtained by solving Maxwell's equations with appropriate assumptions and boundary conditions [1]. However, the complexity of real shield and source geometries and the anisotropy of the medium do not allow a solution to be easily obtained, unless numerical methods are exploited. Furthermore, even if an analytical solution can be achieved, it might be so complex to be of no practical use for shielding design. Nevertheless, the analysis of magnetic shields by standard numerical methods, for example, the finite element method, gives sufficient tool for design of practical shields, especially when the number of layers is low [2,3].

One means for reducing magnetic fields in given region is to make use of some properties of materials, as a way for altering the spatial distribution of such fields from field source. When a shielding magnetic material separates wires with currents, that are sources for a magnetic field from regions, where reduction of field induction **B** is required, the shielding sheets cause a change in the distribution of the magnetic field, directing lines of magnetic flux away from the shielded domain [4].

A quantitative measure of the magnetic shield in reducing the magnetic induction at a given place is the shielding coefficient. It is defined as the ratio of magnitude of magnetic induction at given point when the shield is present, to the magnitude of magnetic induction at the same point, when the shielding material is absent. In general shielding factor is a function of material properties, position at which it is measured, distance of the shield from the field source and magnitude of the excitation [5]. If the magnetic permeability of a shielding material depends significantly from flux

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 252–259, 2004.
© Springer-Verlag Berlin Heidelberg 2004

density within magnetic material, the shielding factor is dependent on excitation value. Problem geometry also plays very important role, both from theoretical and practical point of view [3].

Special difficulties arise in the case of shielding of weak magnetic fields. As it was pointed in [1] by induction of the order of 10 μT ferromagnetic materials behave fast as paramagnetics. As a result shielding effectiveness becomes very low. In [1] author suggested to use anisotropy effect to increase shielding coefficient. He gives experimental data which show that such shielding method for very weak fields can be to some extend effective.

In this paper author describes numerical simulation of anisotropic shielding of weak electromagnetic fields in the case of magnetic field. The formulation is based on vector magnetic potential. At the end some illustrative example is given.

2 Main Equations

Material equation for anisotropic magnetic material can be written in general form as:

$$\mathbf{B} = \overline{\overline{\mu}}\mathbf{H} \tag{1}$$

When spatial coordinate axes are chosen to be coincident with main anisotropic axes of the magnetic material, the constitutive relation (1) has the form:

$$\begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{bmatrix} \cdot \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} \mu_1 H_x \\ \mu_2 H_y \\ \mu_3 H_z \end{bmatrix} \tag{2}$$

It is assumed here that anisotropic axes overlap geometrical axes. Here μ_1, μ_2 and μ_3 are permeability coefficients for *x, y,* and *z* axes, respectively. This equation can be written in simpler form as:

$$B_x = \mu_1 H_x \quad B_y = \mu_2 H_y \quad B_z = \mu_3 H_z \tag{3}$$

After introducing vector magnetic material **A** given by

$$\mathbf{B} = \mathrm{rot}\,\mathbf{A} \tag{4}$$

and utilizing equations (3), vector components of magnetic field strength **H** have the form:

$$H_x = \frac{1}{\mu_1}\left(\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z}\right) \tag{5}$$

$$H_y = \frac{1}{\mu_2}\left(\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x}\right) \tag{6}$$

$$H_z = \frac{1}{\mu_3}\left(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}\right) \tag{7}$$

Ampere's law

$$\mathrm{rot}\,\mathbf{H} = \mathbf{J} \tag{8}$$

gives relation between vector components of magnetic field strength and current density vector

$$\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} = J_x \tag{9}$$

$$\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = J_y \tag{10}$$

$$\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = J_z \tag{11}$$

Introducing into above equations usual gauge condition for magnetostatic field [8]:

$$\frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y} + \frac{\partial A_z}{\partial z} = 0 \tag{12}$$

we get the final equations for anisotropic case:

$$\begin{aligned}\frac{\partial}{\partial x}\left[\frac{1}{\mu_2}\left(\frac{\partial A_x}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left[\frac{1}{\mu_3}\left(\frac{\partial A_x}{\partial y}\right)\right] + \\ + \frac{\partial}{\partial z}\left[\frac{1}{\mu_2}\left(\frac{\partial A_x}{\partial z}\right)\right] + \frac{\partial}{\partial x}\left[\left(\frac{1}{\mu_2} - \frac{1}{\mu_3}\right)\left(\frac{\partial A_y}{\partial y}\right)\right] = -J_x\end{aligned} \tag{13}$$

$$\begin{aligned}\frac{\partial}{\partial x}\left[\frac{1}{\mu_3}\left(\frac{\partial A_y}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left[\frac{1}{\mu_3}\left(\frac{\partial A_y}{\partial y}\right)\right] + \\ + \frac{\partial}{\partial z}\left[\frac{1}{\mu_1}\left(\frac{\partial A_y}{\partial z}\right)\right] + \frac{\partial}{\partial y}\left[\left(\frac{1}{\mu_3} - \frac{1}{\mu_1}\right)\left(\frac{\partial A_z}{\partial z}\right)\right] = -J_y\end{aligned} \tag{14}$$

$$\begin{aligned}\frac{\partial}{\partial x}\left[\frac{1}{\mu_2}\left(\frac{\partial A_z}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left[\frac{1}{\mu_1}\left(\frac{\partial A_z}{\partial y}\right)\right] + \\ + \frac{\partial}{\partial z}\left[\frac{1}{\mu_1}\left(\frac{\partial A_z}{\partial z}\right)\right] + \frac{\partial}{\partial z}\left[\left(\frac{1}{\mu_1} - \frac{1}{\mu_2}\right)\left(\frac{\partial A_x}{\partial x}\right)\right] = -J_z\end{aligned} \tag{15}$$

To solve the last three partial differential equations for vector potential **A**, standard Galerkin's nodal finite element method was used [5].

3 An Illustrative Example

As an illustrative example let us consider a rectangular current carrying wire with rectangular cross section, which is shielded by ferromagnetic plate, as it is given in fig.1. Rectangular wire is 400 mm wide in both *x* and *y* direction and wire cross section has value 10 × 10 mm. Over the wire in *z* direction there is a permeable plate, which is 360 mm away from it.

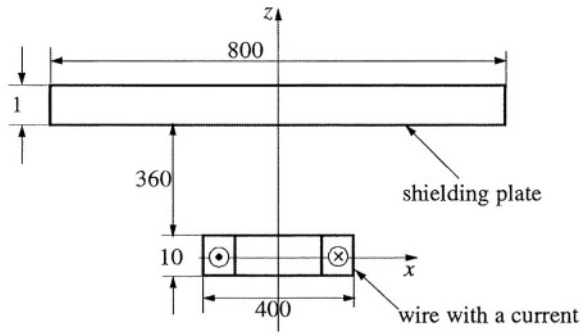


Fig. 1. Geometry description of simple shielding problem. All dimensions are given in mm.

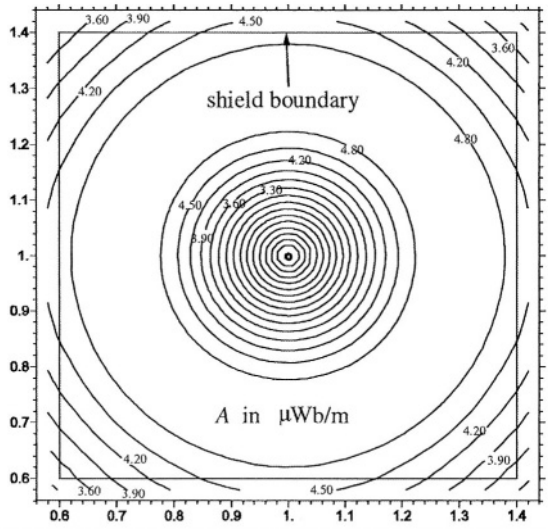


Fig. 2. Equipotential lines of a magnitude of the vector **A** 1 mm over the plate for $\mu_1 = 5$.

The pate thickness is 1mm and it is 800 × 800 mm wide. Total current flowing in the wire has value 200A, what gives current density $J = 200\text{A/cm}^2$. Relative permeabilities in y and z direction were in all simulation constant and had values $\mu_2 = 5$ and $\mu_3 = 1$, respectively. In order to explore an influence of anisotropy on magnetic field distribution over the plate only coefficient μ_1 was changed. All simulations were carried out for four μ_1 values: 5, 35, 70 and 105. In all cases magnetic induction over the shield was less then 60 μT, what means that only weak magnetostatic fields were considered

Numerical simulation was carried out in full three dimensions where equations (13) to (15) were solved. At distance 2000 mm from centre of the plate, potential **A** and its partial normal derivatives were assumed to be 0. Whole calculation domain was divided into 14280 tetrahedral finite elements with 67440 functional nodes, what gives altogether 134888 unknowns. These equations were solved by standard iterative method.

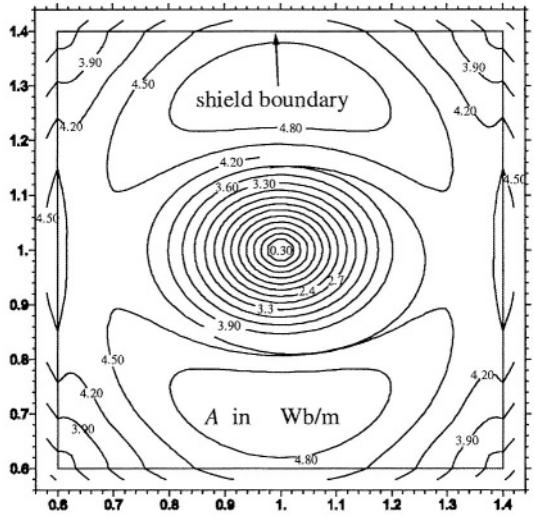


Fig. 3. Equipotential lines of a magnitude of the vector **A** 1 mm over the plate for $\mu_1 = 70$.

Results of computations are shown in subsequent figures. In figures 2 and 3 equipotential lines of a magnitude of the magnetic vector potential **A** are plotted. Fig.2 shows A for relative permeability in x direction $\mu_1 = 5$, that is when material is assumed to be isotropic and in Fig.3 for $\mu_1 = 70$. In both cases the plots are drawn over a plate at distance 1mm in z direction.

In Fig.2 the equipotantial lines have circular shape cantered at plate middle point while in Fig.3 one can see deformation of the magnetic field for the anisotropic case. The field is scratched substantially in x direction.

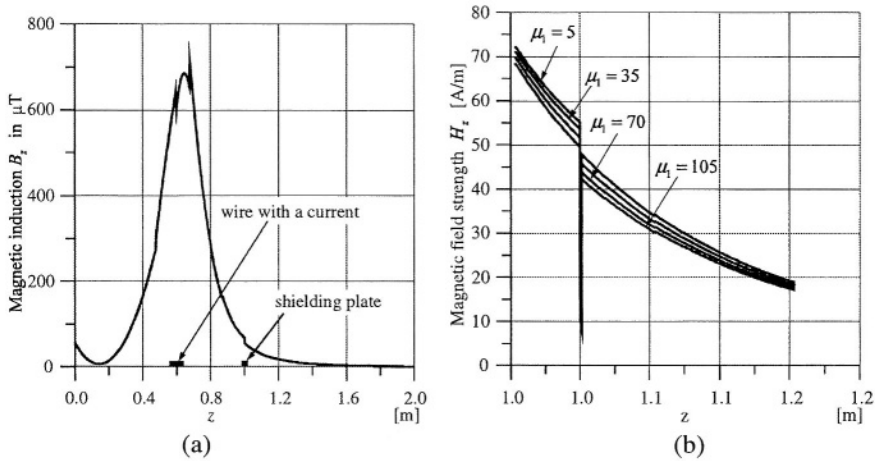


Fig. 4. *Z*-component of a magnetic induction **B** along z axis for relative permeability $\mu_1 = 70$ (a)and z-component of a magnetic field strength **H** along z axis for different values of relative permeability in *x* direction (b).

Fig.4a shows a plot of B_z along *z* axis for relative permeability $\mu_1 = 70$. On the parallel axis one can see symbolically depicted cross section of current carrying wire and shielding plate. The greatest value of B_z is attained in middle of the rectangular wire, as it is to expected. The value of B_z near shielding plate is about 100 lower as its maximal value. Over the plate in *z* direction it decreases substantially. One has to point out that on *z* axis there is only *z* component of the field because the *x* and *y* components due to problem symmetry are equal zero.

In Fig.4b magnetic field strength H_z over the plate in *z* direction is shown. One can see that for different permeability values the component H_z changes not substantially. This is caused by the fact that shield is distant 360 mm from plate and for this value of the strength, the field permeability is very low. Shielding effectiveness can be increased substantially by placing the shield, where induction is much greater or by increasing plate thickness.

Fig.5a shows plots of *z* component of a magnetic induction **B** over shielding plate along z axis for different values of relative permeability. One also can see that shielding effectiveness is low and increasing anisotropy does not cause substantial decrease in B_z.

Fig.5b shows plots of in B_z along shielding plate in *x* direction and the changes of its value for different permabilities. The sharp changes of curves are over the conducting wire. In the magnetostatic case the shielding is due only to flux shunting from the region where decreasing of the field is required. This situation one can observe in figures 2, 3, and 7. Increasing permeability both in *x* and *y* directions makes substantially shunting mechanism more effective.

The order of numerical errors can be assessed investigating, how chosen error indicator converges, when number of finite elements and also number of nodes increases. A good candidate for such error indicator can be deduced from Coulomb gauge (17).

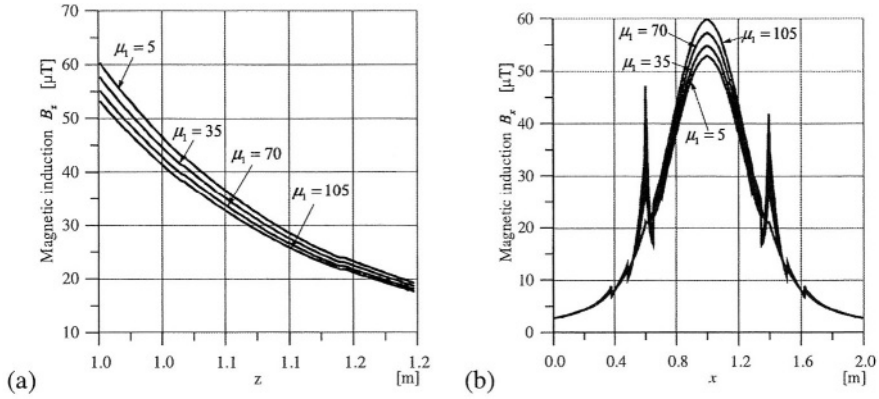


Fig. 5. *Z*-component of a magnetic induction **B** over shielding plate along z axis (a) and . *z*-component of a magnetic induction **B** along z axis (b) for different values of relative permeability.

It states that magnetic vector potential field **A** is conservative, what means that in computational domain there are not sources of this field. As consequence in ideal case the total flux of this field through any close surface placed in computational domain should be zero. Thus defining the error indicator in following way:

$$\text{Error} = \frac{\oint_S \mathbf{A} \cdot d\mathbf{s}}{\oint_S |\mathbf{A} \cdot \mathbf{n}| \cdot \mathrm{d}\, s} 100\% \tag{16}$$

where *S* is any closed surface and investigating how it changes with increasing value of number of nodes, we can asses the numerical stability of the computational process. Of course, such defined error indicator is only necessary condition for convergence and not sufficient.

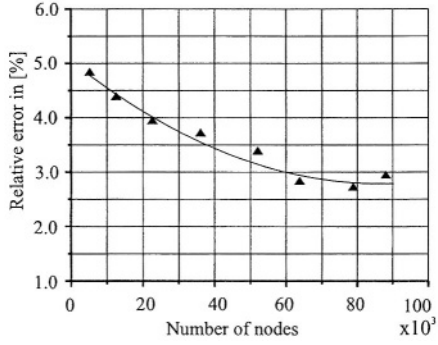


Fig. 6. Relative error indicator given by (16) in function of number of finite element nodes.

Because in shielding plate the filed changes most abruptly, as observation surface *S* boundary of this plate was chosen. Relative error defined as in (16) is shown in fig. 6.

We can see that it decreases as number of elements and nodes increases, what assure us that numerical process is convergent.

4 Conclusions

This paper gives descriptive methodology for anisotropic magnetic shielding that is based on solution Maxwell's equations for magnetostatic field in full three dimensions. First equations for vector magnetostatic potential **A** were formulated and subsequently gauge condition

$$\operatorname{div} \mathbf{A} = 0 \tag{17}$$

implemented. The method is quite general and powerful. It provides a tool for computing the effectiveness of shield design based on anisotropic material properties and geometric dimensions.

General conclusions from all calculation agree with that obtained experimentally in [6, 7]. The shielding effectiveness for thin shielding plate is rather low.

Acknowledgement. This work was supported by the AGH University of Science and Technology, under grant 11.11.120.183.

References

1. Magele C.A., Preis K., Renhart W.: Some improvements in non-linear 3D magnetostatics, IEEE Trans. on Magn., vol. 26, (1990) 375-378
2. Ayoub M., Roy F., Bouillault F., Razek A.: Numerical modelling of 3D magnetostatic saturated structures, IEEE Trans. on Magn., vol. 28, (1992)1052-1055
3. Kraehenbuehl, L., Muller D.: Thin layers in electrical engineering. Example of shell models in analysis eddy-currents by boundary and finite element methods, IEEE Trans. on Magn., vol. 29, (1993) 1450-1455
4. Kurgan E.: Magnetic analysis of inhomogeneous double-layer shields at low frequencies, Proc. of the 15th International Symposium on Electromagnetic Compatibility, Wrocław, (2000) 326 – 330
5. Silvester P., Ferrari R.L.: Finite elements for electrical engineers, Cambridge University Press, Cambridge, 1996.
6. Karwat, T.: Influence of the anisotropy on the shielding effectiveness of electromagnetic devices. (in Polish), Proc. of XXIV International Seminar on Fundamentals of Electrotechnics and Circuit Theory, Gliwice-Ustroń, (2001) 81 – 84
7. Kurgan E.: Magnetic Shielding of a PCB Stripline Structure, Proc. of Seminar on Electrical Engineering BSE'2001, vol. 13, Istebna, (2001) 106 – 111

Functionalization of Single-Wall Carbon Nanotubes: An Assessment of Computational Methods

Brahim Akdim[1], Tapas Kar[2], Xiaofeng Duan[1], and Ruth Pachter[1]

[1]Air Force Research Laboratory, Materials and Manufacturing Directorate
Wright-Patterson Air Force Base, OH, USA
{Brahim.Akdim,xiaofeng.Duan,Ruth.Pachter}@wpafb.af.mil
[2]Department of Chemistry and Biochemistry, Utah State University, Logan, UT, USA
tkar@wpafb.af.mil

Abstract. We summarize a theoretical study for modeling functionalization of single-wall carbon nanotubes (SWCNTs), specifically first principles density functional theory calculations, as compared to semi-empirical or simplified hierarchical methods. We focus on the assessment of the methods to be applied to obtain reliable results and gain a fundamental understanding of the diazotization and ozonolysis of SWCNTs. Computational challenges encountered are highlighted.

1 Introduction

Applications of SWCNTs are still limited by the inability to carefully control the behavior of these materials, for example, with respect to the separation of metallic vs. semiconducting tubes, or nanotubes with different diameters. Thus, a number of chemical functionalization and solubilization studies emerged, recently reviewed [1], including direct attachments of functional groups to the side-wall of a SWCNT using diazonium reagents [2,3,4], solvent free functionalization [5], fluorination and subsequent derivatization [6], or functionalization by strong oxidizing agents, exploring various oxidants [7], due to the inherent strain in SWCNTs, rationalized, in part, by the pyramidalization angle [8]. Most recently, structure-based sorting by sequence-dependent DNA assembly was reported [9]. In this study, we examined the diazotization [10], and ozonolysis, which was shown to enhance solubility [11], of SWCNTs.

In order to gain insight into the functionalization mechanisms of SWCNTs theoretically, a large number of atoms have to be included in quantum mechanical calculations, which could become infeasible [12]. Hence, mixed QM/MM methods, such as ONIOM, introduced by Morokuma et al. [13,14,15,16], were found appropriate [17] for modeling large molecular/nano systems. Within the ONIOM scheme, a first principles calculation is performed on a small part of the system, while the remaining atoms are treated at a lower level of theory, such as by semi-empirical or empirical methods.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 260–267, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In the two-layered scheme, E_{ONIOM} is given by:

$$E_{ONIOM} = E_3 + E_2 - E_1 \quad (1)$$

where E_3 and E_1 relate to the low-level theory of the complete and model systems, respectively, while E_2 is the energy of the model system computed at a high-level of theory. The multi-layer ONIOM method has been formulated [18]. However, for an accurate calculation, the system partitioning into subunits has to be applied with care. For example, in modeling nitrogen atoms interacting with carbon clusters, including a (9,0) nanotube [19], and using a range of sizes for the high-level portion, a large disparity in the binding energies was calculated, ranging from –16.5 to –78.9 kcal/mol, for the doublet state, depending on the model used. In our investigation, we examine the reliability of ONIOM for modeling SWCNT functionalization.

2 Computational Details

C(5,5) SWCNTs were used in all calculations, where 2-phenylimidazole (L) was modeled, functionalized at the SWCNT. The binding energy (BE) was calculated as follows:

$$BE = E(L-SWCNT) - [E(SWCNT) + E(L)] \quad (2)$$

A positive value of BE indicates an endothermic reaction, whereas the exothermic reaction is associated with a negative BE. Note that the adjacent carbon to the functional group is saturated with a hydrogen atom (cf. Figure 1a).

ONIOM calculations [20], applying B3LYP/6-31G*, were carried out with varying sizes of high-level SWCNT models, of 2, 12, and 16 carbon-atoms, while the full system was treated with a semi-empirical, or an empirical UFF (Universal Force Field) method [21]. The functionalized SWCNT was also modeled from first principles by using the B3LYP exchange-correlation functional with 3-21G and 6-31G* basis sets, and with varying tube lengths to assess system size effects, ranging from 5 (Figure 1b) to 11 unit cells. We note that our largest calculation (11 SWCNT unit cells, and the functional unit, at the B3LYP/3-21G level) consisted of 279 atoms. These calculations were carried out on a SGI Origin 3900, using 8 processors (300 MW in memory/CPU). A single SCF iteration's timing was ca. 5 minutes/CPU, and about 20 SCF iterations for an optimization cycle were required.

Simulations applying periodic boundary conditions were carried out using an all-electron linear combination of atomic orbitals DFT approach [22], previously shown to be appropriate for modeling nanotubes [23]. The Perdew-Burke-Ernzerhof (PBE) exchange-correlation functional [24], within the generalized gradient approximation, was applied, to reduce the over-binding effects caused by the local density approximation, and known to adequately predict the density of states as compared to experiment [25]. A double numerical d-functions basis set was used. To avoid intertube interactions, a unit cell with a distance of 30Å in the direction perpendicular to the

tube axis, separating a tube and its image, was employed. In the direction of the tube axis, 3 and 5 SWCNT units were studied (Figures 1c and 1d, respectively).

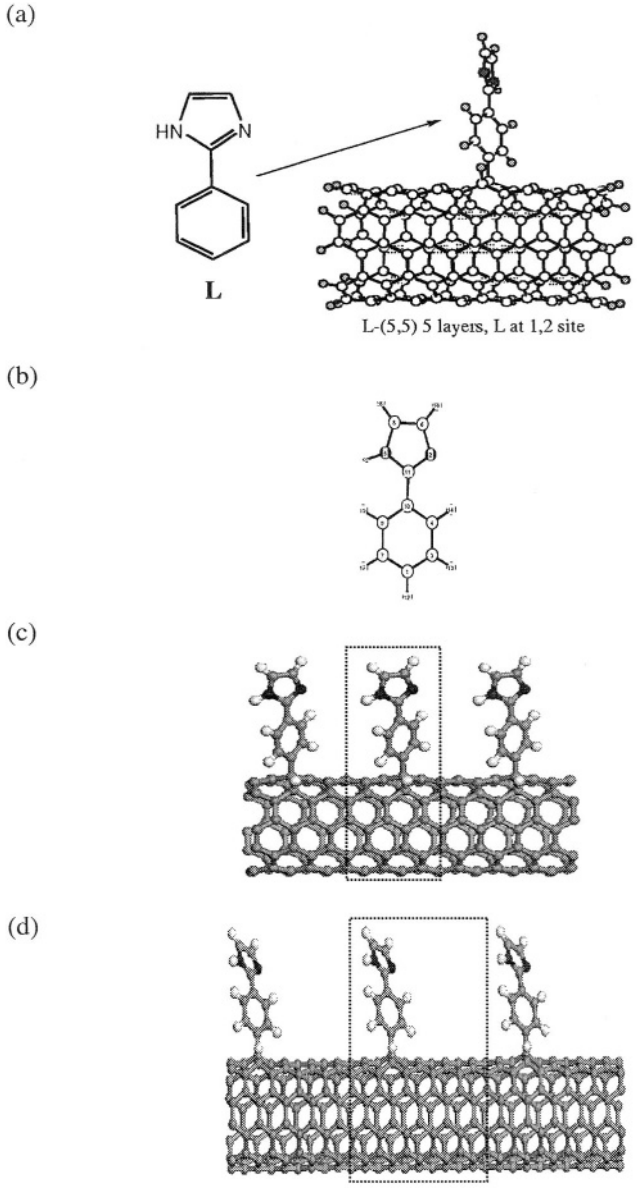


Fig. 1. Atomistic systems studied: (a) L-functionalized 5-unit cell SWCNT; (b) numbering for L; (c) and (d): 3- and 5- SWCNT units with periodic boundary conditions; the box delimits the super-cell used in the simulation

3 Results and Discussion

3.1 2-Phenylimidazole

The optimized 2-phenylimidazole structures show significant differences between the semi-empirical and DFT/B3LYP exchange-correlation functional results (e.g., bond lengths reported in Table 1; numbering in Fig. 1b). Moreover, notable differences were obtained for the inter-ring torsional angle, or co-planarity, namely, 11.2 deg, 30 deg, and 90 deg, when applying B3LYP/6-31+G**, PBE/DNP, or semi-empirical MNDO and PM3, respectively. Interestingly, the PBE/DNP result is in better agreement with previous Hartree-Force/6-31G** calculations for the ground state of 2-phenylimidazole of ca. 19 deg [26] than the B3LYP/6-31+G** result. A co-planar configuration was noted for the excited state [26].

Table 1. Optimized structure of 2-phenylimidazole; values in Å

Bonds	MNDO	AM1	PM3	PM5	3-21G[a]	6-31G*[a]	6-31+G**[a]
1-3	1.404	1.394	1.391	1.387	1.399	1.398	1.399
3-4	1.406	1.393	1.390	1.386	1.391	1.391	1.393
1-7	1.405	1.394	1.391	1.387	1.396	1.395	1.397
4-10	1.419	1.404	1.400	1.396	1.406	1.406	1.407
9-10	1.419	1.403	1.400	1.395	1.404	1.404	1.406
9-7	1.405	1.394	1.389	1.386	1.395	1.394	1.396
10-11	1.470	1.462	1.457	1.457	1.459	1.466	1.467
2-11	1.352	1.366	1.360	1.355	1.337	1.326	1.326
5-11	1.401	1.410	1.399	1.400	1.385	1.374	1.375
6-2	1.387	1.387	1.387	1.397	1.393	1.371	1.372
8-5	1.397	1.390	1.399	1.403	1.393	1.378	1.379
8-6	1.394	1.409	1.387	1.385	1.376	1.374	1.376

[a]B3LYP exchange-correlation functional.

3.2 Diazotization

Table 2 summarizes the binding energies for a C(5,5) functionalized SWCNT (Fig. 1a). In order to ascertain the applicability of our calculations to model functionalization of SWCNTS, which, as was pointed out, are known to require special conditions for reactions to take place, no recourse as to a suggested mechanism was taken at this stage. The results show the reaction to be endothermic, also when periodic boundary conditions were adopted, where BEs of 40 kcal/mol and 43 kcal/mol for the 3- and 5-unit cell SWCNT models, respectively, were calculated. We note that the closest distances between an atom in the functional group and its image in these models are in the order of 4.3Å and 8.9Å, respectively, as shown in Figs. 1c-d.

In probing the effects of different unit-cell sizes without periodic boundary conditions, we find, once again, the reaction to be endothermic, with a BE of ca.

40kcal/mol, for 5- and 7- unit cells, respectively. A lower value has been obtained for the 11-unit cell (Table 2), possibly due to the smaller basis set applied. These results further confirm that such reactions are difficult to occur.

Table 2. Binding energies (BE, kcal/mol) obtained with different theoretical methods

	Methods	BE(kcal/mol)
5- unit-cell	B3LYP/6-31G*	39.68
	AM1	17.02
	MNDO	7.83
	PM3	38.21
	PM5	3.10
	ONIOM(B3LYP/6-31G*:UFF)	-43.59[a]
	(2 Carbon model)	-20.78[b]
	ONIOM(B3LYP/6-31G*:UFF)	29.66[a]
	(12 Carbon model)	-97.63[b]
	ONIOM(B3LYP/6-31G*:UFF)	-7.90[a]
	(16 Carbon model)	35.85[b]
11- unit-cell	B3LYP/3-21G	19.01

binding energy using [a]ONIOM extrapolated energy;
[b]high level energy part of the pertinent model.

Calculations with ONIOM provided varying results, depending on the model size, emphasizing the importance of applying high-level first-principles methods (Table 2). These disparities may invalidate the application of this approach for modeling reliably the functionalization of SWCNTs. However, when modeling functionalization of SWCNTs with terminal carboxylic groups, obtained by oxidation [27, 28], where less subtle changes occur, the results are less sensitive to the use of ONIOM.

3.3 Ozonolysis

Ozonolysis was previously investigated by applying ONIOM [29], indicating that the 1,3 cycloaddition of O_3 onto the sidewall of a C(5,5) tube is possible and a binding energy of–38.7 kcal/mol has been estimated using B3LYP/6-31G*:AM1. In another study, Lu et al. [30] found a binding energy of–31.3 kcal/mol, considering a 6-layer tube, while first principles calculations were also performed [31]. Most recently, sidewall epoxidation of SWCNTs was studied with ONIOM [32]. In our study, 2- and 16-carbon atoms ONIOM calculations were performed within the two-layered scheme (Fig. 2). The (B3LYP/6-31G*:AM1) results are consistent with previous work. More accurate calculations, such as (B3LYP/6-31G*//B3LYP/6-31G*), result in different binding energies (Table 3).

The discrepancies when applying a 2- vs. 16-atom model within ONIOM emphasize, once again, the importance of an appropriate partitioning of the molecular

model. Furthermore, within the same model, different results were obtained when changing the low-level of theory (UFF or AM1), with an estimated difference of about 18 kcal/mol.

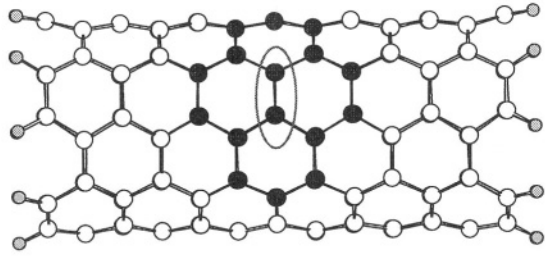


Fig. 2. ONIOM models for modeling ozonolysis: the oval circle points to the 2- carbon models, whereas the filled circles are the 16- carbon models.

Table 3. Ozonolysis of SWCNTs for different levels of theory

Methods	BE(kcal/mol)
AM1	-0.26
B3LYP/3-21G//B3LYP/3-21G	-48.70
B3LYP/6-31G*//B3LYP/3-21G	-23.51
B3LYP/6-31G*//B3LYP/6-31G*	-19.66
ONIOM B3LYP/6-31G*:UFF	-34.32[a]
(2-carbon model)	-59.6[b]
ONIOM B3LYP/6-31G*:AM1	-10.16[a]
(2-carbon model)	-72.91[b]
ONIOM B3LYP/6-31G*:UFF	-13.40[a]
(16-carbon model)	-7.17[b]
ONIOM B3LYP/6-31G*:AM1	-31.34[a]
(16-carbon model)	-15.64[b]

binding energy using [a]ONIOM extrapolated energy;
[b]high level part of the pertaining model.

4 Conclusions

Overall, as anticipated, our calculations show the diazotization of SWCNTs to be endothermic, while ozonolysis is exothermic. In assessing an appropriate level of theory to be applied in modeling functionalization of SWCNTs, we find that density functional theory calculations are preferred, while although the ONIOM model with a large number of atoms at a high-level of theory could provide reliable energetics, care

should be taken in defining a suitable model size within this framework. Indeed, to understand the proposed reaction mechanisms with SWCNTs, where water-soluble diazonium salts exhibit highly chemoselective reactions with metallic vs. semiconducting tubes, we currently apply DFT to calculate the electronic structures with respect to the reaction paths [33].

References

1. Sun, Y.-P., Fu, K., Lin, Y., Huang, W., Acc. Chem. Res. 35 (2002) 1096
2. Bahr, J. L., Tour, J. M., Chem. Mater. 13 (2001) 3823
3. Bahr, J. L., Yang, J., Kosynkin, D. V., Bronikowski, M. J., Smalley, R. E., Tour, J. M., J. Am. Chem. Soc. 123 (2001) 6536
4. Strano, M. S., Dyke, C. A., Usrey, M. L., Barone, P. W., Allen, M. J., Shan, H., Kittrell, C., Hauge, R. H., Tour, J. M., Smalley, R. E., Science 301 (2003) 1519
5. Dyke, A., Tour, J. M., J. Am. Chem. Soc. 125 (2003) 1156
6. Khabashesku, V. N., Billups, W. E., Margrave, J. L., Acc. Chem. Res. 35 (2002) 1087
7. Zhang, J., Zou, H., Qing, Q., Yang, Y., Li, Q., Liu, Z., Guo, X., Du, Z., J. Phys. Chem. B 107(2003)3712
8. Niyogi, S., Hamon, M. A., Hu, H., Zhao, B., Bhowmik, P., Sen, R., Itkis, M. E., Haddon, R. C., Acc. Chem. Res. 35 (2002) 1105
9. Zheng, M., Jagota, A., Strano, M, S., Santos, A. P., Barone, P., Chou, S. G., Diner, B. A., Dresselhaus, M, S., Mclean, R, S., Onoa, G. B., Samsonidze, G. G., Semke, E. D., Usrey, M., Walls, D. J., Science 302 (2003) 1545
10. Dang, T., Vaia, R. private communication.
11. Cai, L., Bahr, J. L., Yao, Y., Tour, J. M., Chem. Mater. 14 (2002) 4235
12. Schmidt, M. W., Baldridge, K. K, Boatz, J. A., Elbert, S. T., Gordon, M. S., Jensen, J. H., Koseki, S., Matsunaga, N., Nguyen, K. A., Su, S., Windus, T. L., Dupuis, M., Montgomery, J. A., J. Comput. Chem., 14 (1993) 1347
13. Maseras, F., Morokuma, K., J. Comput. Chem. 16 (1995) 1170
14. Humbel, S., Sieber, S. S., Morokuma, K., J. Chem. Phys. 105 (1996) 1959
15. Dapprich, S., Komáromi, I., Byun, K. S., Morokuma, K., Frisch, M. J., J. Mol. Struct. Theochem 462 (1999)1
16. Vreven, T., Morokuma, K., J. Comput. Chem. 21 (2000) 1419
17. Vreven T., Morokuma K., Farkas O., Schlegel H. B. Frisch M. J., Cherry L. J Comput. Chem. 24 (2003) 760
18. Tschumper, G. S., Morokuma, K., J. Mol. Struct. Theochem 592 (2002) 137
19. Walch, S. P., Chem. Phys. Lett. 374, (2003) 501
20. Gaussian2003,_http://www.gaussian.com/
21. Rappe, A. K., Casewit, S. J., Goddard, W. A., Skiff, W. M., J. Am. Chem. Soc., 114 (1992) 10024
22. Delley, B. J., Chem. Phys. 113 (2000) 7756; implemented by Accelyrs, Inc.
23. Akdim, B., Duan, X., Adams, W. W., Pachter, R., Phys. Rev. B. 67 (2003) 245404
24. Perdew, J. P., Burke, K., Ernzerhof, M., Phys. Rev. Lett. 77 (1996) 3865
25. Avramov, P. V., Kudin, K. N., Scuseria, G. E., Chem. Phys. Lett. 370 (2003) 597
26. Catalan, J., de Paz, J. L. G., del Valle, C., J., Kasha, M., J., Phys. Chem. A 101 5284 (1997)
27. Basiuk, V. A., Basiuk, E. V., Saniger-Blesa, J-M., Nano Lett., (2001) 657

28. Basiuk, V. A., Nano Lett., (2002) 835
29. Lu, X., Zhang, L., Xu, X., Wang, N., Zhang, Q., J. Phys. Chem. B 106 (2002) 2136
30. Lu, X., Tian, F., Xu, X., Wang, N., Zhang, Q., J. Am. Chem. Soc. 125 (2003) 7923
31. Duan, X., Akdim, B., Pachter, R., Dekker Encyclopedia of Nanoscience and Nanotechnology, in press.
32. Lu, X., Qinghong, Y., Zhang, Q., Org. Lett. 5 (2003) 3527
33. Duan, X., Akdim, B., Pachter, R., work in progress.

Improved Sampling for Biological Molecules Using Shadow Hybrid Monte Carlo

Scott S. Hampton and Jesús A. Izaguirre

University of Notre Dame, Notre Dame IN 46556, USA

Abstract. Shadow Hybrid Monte Carlo (SHMC) is a new method for sampling the phase space of large biological molecules. It improves sampling by allowing larger time steps and system sizes in the molecular dynamics (MD) step of Hybrid Monte Carlo (HMC). This is achieved by sampling from high order approximations to the modified Hamiltonian, which is exactly integrated by a symplectic MD integrator. SHMC requires extra storage, modest computational overhead, and a reweighting step to obtain averages from the canonical ensemble. Numerical experiments are performed on biological molecules, ranging from a small peptide with 66 atoms to a large solvated protein with 14281 atoms. Experimentally, SHMC achieves an order magnitude speedup in sampling efficiency for medium sized proteins.

1 Introduction

The sampling of the configuration space of complex biological molecules is an important and formidable problem. One major difficulty is the high dimensionality of this space, roughly $3N$, with the number of atoms N typically in the thousands. Other difficulties include the presence of multiple time and length scales, and the rugged energy hyper-surfaces that make trapping in local minima common, cf. [1]. This paper introduces Shadow Hybrid Monte Carlo (SHMC), a propagator through phase space that enhances the scaling of hybrid Monte Carlo (HMC) with space dimensionality.

The problem of sampling can be thought of as estimating expectation values for a function $A(\Gamma)$ with respect to a probability distribution function (p.d.f.) $\rho(\Gamma)$, where $\Gamma = [\mathbf{x}^{\mathrm{T}}, \mathbf{p}^{\mathrm{T}}]^{\mathrm{T}}$, and $\mathbf{x}^{\mathrm{T}}$ and $\mathbf{p}^{\mathrm{T}}$ are the vectors of collective positions and momenta. For the case of continuous components of Γ,

$$\langle A(\Gamma) \rangle_{\rho} = \int A(\Gamma)\rho(\Gamma)\mathrm{d}\Gamma \,. \tag{1}$$

Examples of observables A are potential energy, pressure, free energy, and distribution of solvent molecules in vacancies [2,3].

Sampling of configuration space can be done with Markov chain Monte Carlo methods (MC) or using molecular dynamics (MD). MC methods are rigorous sampling techniques. However, their application for sampling large biological molecules is limited because of the difficulty of specifying good moves for dense

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 268–275, 2004.
© Springer-Verlag Berlin Heidelberg 2004

systems [4] and the large cost of computing the long range electrostatic energy, cf. [3, p. 261]. MD, on the other hand, can be readily applied as long as one has a "force field" description of all the atoms and interactions among atoms in a molecule. Additionally, MD enables relatively large steps in phase space as well as global updates of all the positions and momenta in the system. MD finds changes over time in conformations of a molecule, where a conformation is defined to be a semi-stable geometric configuration. Nevertheless, the numerical implementation of MD introduces a bias due to finite step size in the numerical integrator of the equations of motion.

MD typically solves Newton's equations of motion, a Hamiltonian system of equations,

$$\dot{\Gamma}(t) = J\mathcal{H}_{\Gamma}(\Gamma(t)), \qquad J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}, \tag{2}$$

with a Hamiltonian

$$\mathcal{H}(\mathbf{x}, \mathbf{p}) = \frac{1}{2}\mathbf{p}^{\mathrm{T}}\mathrm{M}^{-1}\mathbf{p} + U(\mathbf{x}), \tag{3}$$

where M is a diagonal matrix of masses, $U(\mathbf{x})$ is the potential energy of the system, and $\mathbf{p} = M\dot{\mathbf{x}}$ are the momenta. Eq. (2) can also be written as

$$\dot{\mathbf{x}}(t) = \mathrm{M}^{-1}\mathbf{p}(t), \qquad \dot{\mathbf{p}}(t) = F(\mathbf{x}(t)), \tag{4}$$

where the conservative forces $F(\mathbf{x}(t)) = -\nabla U(\mathbf{x}(t))$.

Numerical integrators for MD generate a solution $\Gamma^n \approx \Gamma(n\delta t)$, where the step size or time step used in the discretization is δt. Typical integrators can be expressed as

$$\Gamma^{n+1} = \Psi(\Gamma^n), \tag{5}$$

where Ψ represents a propagator through phase space. Any time reversible and volume preserving integrator can be used for HMC. SHMC requires in addition that the integrator be symplectic (cf. [5, p. 69]). An integrator is symplectic if $\partial_{\Gamma}\Psi(\Gamma)^{\mathrm{T}} J \partial_{\Gamma}\Psi(\Gamma) \equiv J$. In this work, both implementations use the Verlet/Leapfrog discretization [6], which satisfies the constraints for both propagators.

HMC, introduced in [7], uses MD to generate a global MC move and then uses the Metropolis criterion to accept or reject the move. HMC rigorously samples the canonical distribution and eliminates the bias of MD due to finite step size. Unfortunately, the acceptance rate of HMC decreases exponentially with increasing system size N or time step δt. This is due to discretization errors introduced by the numerical integrator and cause an extremely high rejection rate. The cost of HMC as a function of system size N and time step δt has been investigated in [8,9]. These errors can be reduced by using higher order integrators for the MD step as in [10]. However, higher order integrators are not an efficient alternative for MD for two reasons. First, the evaluation of the force is very expensive, and these integrators typically require more than one force evaluation per step. Second, the higher accuracy in the trajectories is not needed in MD, where statistical errors and errors in the force evaluation are very large.

2 Shadow HMC

SHMC is a biased variation on HMC. It uses a smooth approximation to the modified Hamiltonian to sample more efficiently through phase space. The modified Hamiltonian is exactly conserved by the numerical integrator and a cheap, arbitrarily accurate, approximation called a shadow Hamiltonian has been proposed in [11]. SHMC samples a non-canonical distribution defined by high order approximations to the modified Hamiltonian, which greatly increases the acceptance rate of the method. A reweighting of the observables is performed in order to obtain proper canonical averages, thus eliminating the bias introduced by the shadow Hamiltonian. The overhead introduced by the method is modest in terms of time, involving only dot products of the history of positions and momenta generated by the integrator. There is moderate extra storage to keep this history. In this generalization of HMC, sampling is in all of phase space rather than configuration space alone.

Let $\tilde{\rho}(\mathbf{x}, \mathbf{p})$ be the target density of SHMC, where

$$\tilde{\rho}(\mathbf{x}, \mathbf{p}) \propto \exp\left(-\beta \tilde{\mathcal{H}}(\mathbf{x}, \mathbf{p})\right), \tag{6}$$

$$\tilde{\mathcal{H}}(\mathbf{x}, \mathbf{p}) = \max\left\{\mathcal{H}(\mathbf{x}, \mathbf{p}), \mathcal{H}_{[2k]}(\mathbf{x}, \mathbf{p}) - c\right\}. \tag{7}$$

Here, $\mathcal{H}_{[2k]}(\mathbf{x}, \mathbf{p})$ is the much smoother shadow Hamiltonian, defined in Section 3, and c is an arbitrary constant that limits the amount by which $\mathcal{H}_{[2k]}$ is allowed to depart from $\mathcal{H}(\mathbf{x}, \mathbf{p})$.

Algorithm 1 lists the steps for calculating SHMC. The first step is to generate a set of momenta, $\mathbf{p}'$, usually chosen proportional to a Gaussian distribution. $\mathbf{p}'$ is accepted based on a Metropolis criterion step proportional to the difference of the total and shadow energies. This step is repeated until a set of momenta are accepted. Next, the system is integrated using MD and accepted with probability proportional to Eq. (6). Finally, in order to calculate unbiased values, the observables are reweighted.

The purpose of the constant c is to minimize the difference in the energies so that the reweighted observables of $\mathcal{H}_{[2k]}$ are unbiased. Let $\Delta\mathcal{H} = \mathcal{H}_{[2k]} - \mathcal{H}$. Experiments suggest that $\Delta\mathcal{H}$ is predominantly positive in MD simulations. This is most likely due to the fact that the shadow Hamiltonian is designed to exactly conserve energy of the numerical solution of quadratic Hamiltonians such as those used in MD[11]. Currently, c is chosen proportional to the expected value of the discretization error, $\langle\Delta\mathcal{H}\rangle$. This value is obtained after running a sufficient number of steps and monitoring $\Delta\mathcal{H}$ at each step.

3 Shadow Hamiltonian

The modified equations of a system of differential equations are exactly satisfied by the approximate discrete solution of the numerical integrator used to solve them. These equations are usually defined as an asymptotic expansion in powers

Algorithm 1 Shadow Hybrid Monte Carlo (SHMC)

1. **MC Step:** Given $\mathbf{x}$, generate $\mathbf{p}'$ with p.d.f. $\tilde{\rho}(\mathbf{x}, \mathbf{p})$, using the acceptance-rejection method:
   a) Generate $\mathbf{p}'$ having p.d.f. $\rho_{\mathrm{p}}(\mathbf{p})$
   b) Accept with probability
   $$\min\left\{1, \frac{\exp(-\beta(\mathcal{H}_{[2k]}(\mathbf{x}, \mathbf{p}') - c))}{\exp(-\beta\mathcal{H}(\mathbf{x}, \mathbf{p}'))}\right\}$$
   c) Repeat (1a) - (1b) until P is accepted.
2. **MD Step:** Given Γ :
   a) $\Gamma' = R\Psi(\Gamma)$ (where Ψ nearly conserves $\mathcal{H}_{[2k]}$)
   b) Accept Γ' with probability
   $$\min\left\{1, \frac{\tilde{\rho}(\Gamma')}{\tilde{\rho}(\Gamma)}\right\}$$
   c) If rejected, choose Γ.
3. **Reweighting Step:** Given $\{A, \Gamma\}$, reweight observable A using $\rho(\Gamma)/\tilde{\rho}(\Gamma)$ before computing averages.

of the discretization time step. If the expansion is truncated, there is excellent agreement between the modified equations and the discrete solution [12].

In the case of a Hamiltonian system, Eq. (2), symplectic integrators conserve exactly (within roundoff errors) a modified Hamiltonian $\mathcal{H}^{\delta t}$. For short MD simulations (such as in HMC) $\mathcal{H}^{\delta t}$ stays close to the true Hamiltonian, cf. [5, p. 129–136]. Work by Skeel and Hardy [11] shows how to compute an arbitrarily accurate approximation to the modified Hamiltonian integrated by symplectic integrators based on splitting. The idea is to compute

$$\mathcal{H}_{[2k]}(x, p) = \mathcal{H}^{\delta t}(x, p) + O(\delta t^{2k}) . \tag{8}$$

$\mathcal{H}_{[2k]}$ is the shadow Hamiltonian of order $2k$. It follows from centered finite difference approximations to derivative terms in the expansion of $\mathcal{H}^{\delta t}$, and from interpolation to the evaluation points. It is a combination of trajectory information, that is, k copies of available positions and momenta generated by the MD integration, and an extra degree of freedom β that is propagated along with the momenta. By construction, $\mathcal{H}_{[2k]}$ is exact for quadratic Hamiltonians, which are very common in MD. Details can be found in the original reference.

A shadow Hamiltonian of order $2k$, k even, is constructed as a linear combination of centered differences of the position and momenta of the system. The formulae for the 4^{th} and 8^{th} order shadows, $k = 2$ and $k = 4$ respectively, follow:

$$\mathcal{H}_{[4]} = \frac{1}{2\delta t}\left(\mathrm{A}_{10} - \frac{1}{6}\mathrm{A}_{12}\right) , \tag{9}$$

$$\mathcal{H}_{[8]} = \frac{1}{2\delta t}\left(210\mathrm{A}_{10} - \frac{2}{7}\mathrm{A}_{12} - \frac{19}{210}\mathrm{A}_{14} + \frac{5}{42}\mathrm{A}_{30} + \frac{13}{105}\mathrm{A}_{32} - 315\mathrm{A}_{34}\right). \quad (10)$$

Define the i^{th} centered difference formula to be $\delta\omega^{[i]}$. So, for example, $\delta\mathbf{x}^{[2]}$ would represent the 2^{nd} centered difference of the positions:

$$\delta\mathbf{x}^{[2]} = \mathbf{x}^{n+1} - 2\mathbf{x}^{n} + \mathbf{x}^{n-1}$$

Now define A_{ij}:

$$\mathrm{A}_{ij} = \begin{cases} \delta\mathbf{x}^{[i]} \cdot \delta\mathbf{p}^{[j]}\mathrm{M} - \delta\mathbf{x}^{[j]} \cdot \delta\mathbf{p}^{[i]}\mathrm{M} - \delta\beta^{[i]} & : \quad j = 0 \\ \delta\mathbf{x}^{[i]} \cdot \delta\mathbf{p}^{[j]}\mathrm{M} - \delta\mathbf{x}^{[j]} \cdot \delta\mathbf{p}^{[i]}\mathrm{M} & : \quad j \neq 0 \end{cases} \quad (11)$$

Finally, the β term propagated by Leapfrog is:

$$\beta = -\delta t(\mathbf{x}^{n} \cdot F^{n} + 2U(\mathbf{x}^{n})), \quad (12)$$

where the forces **F**, the positions **x**, and the momenta **p**, are vectors of length $3N$, and N is the number of atoms in the system. M is a diagonal matrix containing the mass of each atom.

4 Numerical Tests

SHMC was tested with a 66 atom Decalanine, and a more complex solvated protein, BPTI, with 14281 atoms. The methods and example systems are available by obtaining PROTOMOL [13] from our website[1]. Simulations were run on a Linux cluster administered by the Department of Computer Science and Engineering at the University of Notre Dame. Each node contains 2, 2.4 GHz Xeon processors and 1 GB RDRAM.

The performance of HMC and SHMC is dependent upon the input parameters of time step δt and trajectory length L. Here, L is amount of simulated time for one MC step. L should be long enough so that the longest correlation times of interest are sampled during an MD step, thus avoiding the random walk behavior of MC. SHMC also needs a tuning parameter c to indicate allowed divergence between the shadow and total energy.

Several techniques have been used to compare SHMC and HMC. The efficiency of sampling is measured by computing the cost to generate a new geometric conformation. The statistical error is measured by computing the potential energy and its standard deviation.

Statistical Correctness. In order to test the statistical correctness of the reweighted values of SHMC, the potential energies (PE) and their standard deviations were computed. Table 1 shows the average potential energy (PE) for Decalanine. Looking through the values, there is little difference statistically speaking. All of the reweighted values are within at least one standard deviation of the unweighted HMC values. Additionally, the reweighted standard deviation is acceptable in all cases.

[1] http://protomol.sourceforge.net

Table 1. Average potential energy (kcal/mol) and standard deviation for Decalanine for HMC and SHMC using an 8^{th} order shadow Hamiltonian.

	Time step (fs)					
Method	0.5	0.75	1.0	1.25	1.5	2
HMC	97.5 ± 6.5	97.4 ± 6.9	100 ± 6.6	99.8 ± 6.7	98.1 ± 7.1	97.4 ± 9.1
SHMC	103 ± 6.7	102 ± 7	96.8 ± 7.2	98.9 ± 6.8	97.3 ± 8	99.7 ± 8.4
c	0.4	0.4	0.6	1.2	1.2	2.8

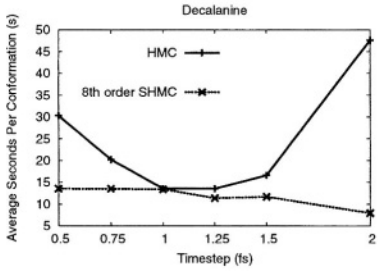


Fig. 1. Average computer time per discovered conformation for 66-atom Decalanine.

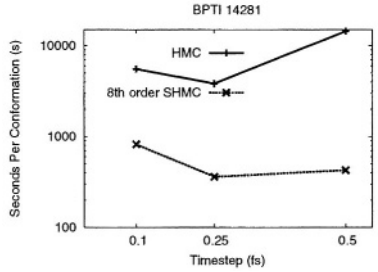


Fig. 2. Average computer time per discovered conformation for 14281-atom BPTI.

Sampling Efficiency. The number of molecular conformations visited by HMC and SHMC is determined using a method suggested in [14]. The sampling efficiency of HMC and SHMC is defined as the computational cost per new conformation. This value is calculated by dividing the running time of the simulation by the number of conformations discovered. This is a fair metric when comparing different sampling methods, since it takes care of the overhead of more sophisticated trial moves, and any other effects on the quality (or lack thereof, e.g., correlation) of samples produced by different sampling techniques.

Figure 1 shows the number of conformations per second as a function of the time step for Decalanine. At its best, HMC is only as good as SHMC for one time step, $\delta t = 1$. In terms of efficiency, SHMC shows a greater than two-fold speedup over HMC when the optimal values for both methods are used. Figure 2 shows even more dramatic results for BPTI with 14281 atoms. The speedup in this case is a factor of 10. This is expected, since the speedup increases asymptotically as $\mathcal{O}(N^{1/4})$[15].

The following graphs demonstrate how c affects simulations. Figure 3 shows a plot of the standard deviation of the potential energy as a function of the value chosen for c. The system is Decalanine, with a time step of 2 fs. Figure 4 shows that the probability of accepting the MD move also decreases as c increases. In the first case, a large c is desirable and in the second case a small c is best.

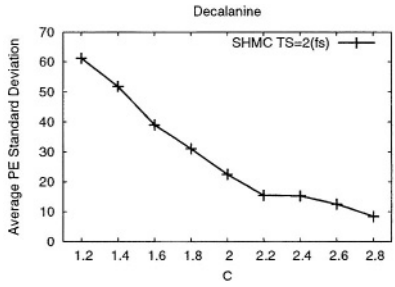


Fig. 3. The effect of c on the standard deviation of the potential energy.

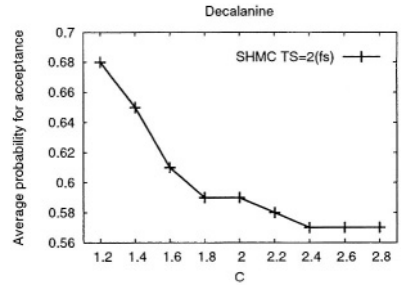


Fig. 4. The effect of c on the probability of accepting the MD step.

5 Discussion

SHMC is a rigorous sampling method [15] that samples a p.d.f. induced by a modified Hamiltonian. Because this modified Hamiltonian is more accurate than the true Hamiltonian, it is possible to increase the efficiency of sampling. Since the modified Hamiltonian is by construction close to the true Hamiltonian, the reweighting does not damage the variance. The additional parameter, c, of SHMC, measures the amount by which the modified and the true Hamiltonian can depart. Different regions of phase space may need different optimal parameters. Here, c is chosen to satisfy both bounds on the statistical error of sampling and an acceptable performance. A rule of thumb is that it should be close to the difference between the true and the modified Hamiltonian. Other criteria are possible, and it would be desirable to provide "optimal" choices.

The efficiency of Monte Carlo methods can be improved using other variance reduction techniques. For example, [16] improves the acceptance rate of HMC by using "reject" and "accept"windows. It accepts whether to move to the accept window or to remain in the reject window based on the ratio of the sum of the probabilities of the states in the accept and the reject windows. SHMC is akin to importance sampling using the modified Hamiltonian. The method of control variates [17] could also be used in SHMC.

Conformational dynamics [18,19] is an application that might benefit from SHMC. It performs many short HMC simulations in order to compute the stochastic matrix of a Markov Chain. Then it identifies almost invariant sets of configurations, thereby allowing a reduction of the number of degrees of freedom in the system.

Acknowledgments. This work was partially supported by an NSF Career Award ACI-0135195. Scott Hampton was supported through an Arthur J. Schmitt fellowship. The authors would like to thank Robert Skeel, David Hardy, Edward Maginn, Gary Huber and Hong Hu for helpful discussions.

References

1. Berne, B.J., Straub, J.E.: Novel methods of sampling phase space in the simulation of biological systems. Curr. Topics in Struct. Biol. **7** (1997) 181–189
2. Leach, A.R.: Molecular Modelling: Principles and Applications. Addison-Wesley, Reading, Massachusetts (1996)
3. Schlick, T.: Molecular Modeling and Simulation - An Interdisciplinary Guide. Springer-Verlag, New York, NY (2002)
4. Brass, A., Pendleton, B.J., Chen, Y., Robson, B.: Hybrid Monte Carlo simulations theory and initial comparison with molecular dynamics. Biopolymers **33** (1993) 1307–1315
5. Sanz-Serna, J.M., Calvo, M.P.: Numerical Hamiltonian Problems. Chapman and Hall, London (1994)
6. Hockney, R.W., Eastwood, J.W.: Computer Simulation Using Particles. McGraw-Hill, New York (1981)
7. Duane, S., Kennedy, A.D., Pendleton, B.J., Roweth, D.: Hybrid Monte Carlo. Phys. Lett. B **195** (1987) 216–222
8. Creutz, M.: Global Monte Carlo algorithms for many-fermion systems. Phys. Rev. D **38** (1988) 1228–1238
9. Mehlig, B., Heermann, D.W., Forrest, B.M.: Hybrid Monte Carlo method for condensed-matter systems. Phys. Rev. B **45** (1992) 679–685
10. Creutz, M., Gocksch, A.: Higher-order hybrid monte carlo algorithms. Phys. Rev. Lett. **63** (1989) 9–12
11. Skeel, R.D., Hardy, D.J.: Practical construction of modified Hamiltonians. SIAM J. Sci. Comput. **23** (2001) 1172–1188
12. Hairer, E., Lubich, C.: Asymptotic expansions and backward analysis for numerical integrators. In: Dynamics of Algorithms, New York, IMA Vol. Math. Appl 118, Springer-Verlag (2000) 91–106
13. Matthey, T., Cickovski, T., Hampton, S., Ko, A., Ma, Q., Slabach, T., Izaguirre, J.A.: PROTOMOL: an object-oriented framework for prototyping novel algorithms for molecular dynamics. Submitted to ACM Trans. Math. Softw. (2003)
14. Kirchhoff, P.D., Bass, M.B., Hanks, B.A., Briggs, J., Collet, A., McCammon, J.A.: Structural fluctuations of a cryptophane host: A molecular dynamics simulation. J. Am. Chem. Soc. **118** (1996) 3237–3246
15. Hampton, S.: Improved sampling of configuration space of biomolecules using shadow hybrid monte carlo. Master's thesis, University of Notre Dame, Notre Dame, Indiana, USA (2004)
16. Neal, R.M.: An improved acceptance procedure for the hybrid Monte Carlo algorithm. J. Comput. Phys. **111** (1994) 194–203
17. Lavenberg, S.S., Welch, P.D.: A perspective on the use of control variables to increase the efficiency of monte carlo simulations. Management Science **27** (1981) 322–335
18. Schüette, C., Fischer, A., Huisinga, W., Deuflhard, P.: A direct approach to conformational dynamics based on hybrid Monte Carlo. J. Comput. Phys **151** (1999) 146–168
19. Schüette, C.: Conformational dynamics: Modelling, theory, algorithm, and application to biomolecules. Technical report, Konrad-Zuse-Zentrum für Informationstechnik Berlin (1999) SC 99-18.

A New Monte Carlo Approach for Conservation Laws and Relaxation Systems

Lorenzo Pareschi[1] and Mohammed Seaïd[2]

[1] Department of Mathematics, University of Ferrara, 44100 Italy
pareschi@dm.unife.it
[2] Fachbereich Mathematik AG8, TU Darmstadt, 64289 Darmstadt, Germany
seaid@mathematik.tu-darmstadt.de

Abstract. We present a Monte Carlo method for approximating the solution of conservation laws. A relaxation method is used to transform the conservation law to a kinetic form that can be interpreted in a probabilistic manner. A Monte Carlo algorithm is then used to simulate the kinetic approximation. The method we present in this paper is simple to formulate and to implement, and can be straightforwardly extended to higher dimensional conservation laws. Numerical experiments are carried out using Burgers equation subject to both smooth and nonsmooth initial data.

1 Introduction

Monte Carlo methods have been always very popular in scientific computing. This is mainly due to the ability to deal efficiently with very large (multiscale) structures without many meshing problems and to their simplicity in keeping the fundamental physical properties of the problems. In particular Monte Carlo methods have been widely used for numerical simulations in rarefied gas dynamics described by the Boltzmann equation [1,5]. More recently these methods have been extended to treat regimes close to continuum situations described by the Euler or Navier-Stokes equations [7,8,9,2]. The common idea in these approximations is to take advantage of the knowledge of the equilibrium state of the equation to build a scheme with the correct behavior close to the fluid-limit. For example, for the Boltzmann equation close to fluid regimes particles are sampled directly from a Maxwellian distribution as in Pullin's method [10]. In this article, inspired by these methods, we use a relaxation approximation to transform a conservation law into a semilinear system which has the structure of a discrete velocity model of the Boltzmann equation. This kinetic form leads naturally to a probabilistic representation. Therefore, the main ideas used in [7,8,9] can be used to simulate the limiting conservation law. More precisely advection of particles is made according to the characteristic speeds of the relaxation system, and the projection into the equilibrium is performed with a suitable sampling strategy.

Let consider the scalar conservation law

$$\begin{aligned} &u_t + \varphi(u)_x = 0, \qquad (x,t) \in \mathbb{R} \times \mathbb{R}^+, \\ &u(x, t=0) = u_0(x), \qquad x \in \mathbb{R}, \end{aligned} \tag{1}$$

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 276–283, 2004.
© Springer-Verlag Berlin Heidelberg 2004

where $u \in \mathbb{R}$ and the flux function $\varphi(u) : \mathbb{R} \to \mathbb{R}$ is nonlinear. As in [3], we replace the scalar Cauchy problem (1) by the semilinear relaxation system

$$\begin{aligned} u_t + v_x &= 0, \\ v_t + a u_x &= -\frac{1}{\varepsilon}\left(v - \varphi(u)\right), \\ u(x,0) = u_0(x), \quad & v(x,0) = \varphi\left(u_0(x)\right), \end{aligned} \tag{2}$$

where $v \in \mathbb{R}$, a is positive constant, and ε is the relaxation rate. When $\varepsilon \longrightarrow 0$, solution of the relaxation system (2) approaches solution of the original equation (1) by the local equilibrium $v = \varphi(u)$. A necessary condition for such convergence is that the subcharacteristic condition [3,6]

$$-a \leq \varphi'(u) \leq a, \qquad \forall\, u, \tag{3}$$

is satisfied in (2). The main advantage in considering the relaxation method lies essentially on the semilinear structure of the system (2), which has two linear characteristic variables (given by $v \pm \sqrt{a}u$), and consequently it can be solved numerically without using Riemann solvers (see [3] and the references therein).

Our purpose in the present paper is to construct a Monte Carlo approach for the conservation law (1) using the fact that, for the relaxation system (2) a kinetic formulation can be easily derived. The organization of the paper is as follows. Section 2 is devoted to the probabilistic formulation for the relaxation system (2). In section 3 we discuss a Monte Carlo algorithm for the relaxation model. Section 4 illustrates the performance of the approach through experiments with the Burgers equation. In the last section some concluding remarks and future developments are listed.

2 Probabilistic Interpretation

In order to develop a probabilistic formulation for the relaxation system (2) we introduce the kinetic variables f and g as

$$u = f + g \qquad \text{and} \qquad v = a(f - g). \tag{4}$$

The relaxation system can be rewritten in a diagonal form as

$$\begin{aligned} f_t + \sqrt{a} f_x &= \frac{1}{\varepsilon}\left(\frac{g - f}{2} + \frac{\varphi(f+g)}{2\sqrt{a}}\right), \\ g_t - \sqrt{a} g_x &= \frac{1}{\varepsilon}\left(\frac{f - g}{2} - \frac{\varphi(f+g)}{2\sqrt{a}}\right). \end{aligned} \tag{5}$$

To solve numerically the equations (5) we split the problem into two steps:

(i) The transport stage

$$\begin{aligned} f_t + \sqrt{a} f_x &= 0, \\ g_t - \sqrt{a} g_x &= 0. \end{aligned} \tag{6}$$

(ii) The relaxation stage

$$f_t = \frac{1}{\varepsilon}\left(\frac{g-f}{2} + \frac{\varphi(f+g)}{2\sqrt{a}}\right),$$
$$g_t = \frac{1}{\varepsilon}\left(\frac{f-g}{2} - \frac{\varphi(f+g)}{2\sqrt{a}}\right). \tag{7}$$

Note that this splitting is first order accurate. A second order splitting for moderately stiff relaxation stages can be derived analogously using the Strang method [11]. For simplicity first we will describe the relaxation problem (7), and then we show how to combine it with the stage (6) for the full problem.

We assume, without loss of generality, that for a fixed x

$$f \geq 0, \qquad g \geq 0, \qquad u = f + g = 1.$$

Furthermore, we assume that the flux function in (1) satisfies

$$\varphi'(u) < 1, \qquad 0 \leq \varphi(u) \leq u. \tag{8}$$

Although problem (7) can be solved exactly we consider its time discretization

$$f^{n+1} = f^n + \frac{\Delta t}{\varepsilon}\left(\frac{g^{n+1}-f^{n+1}}{2} + \frac{\varphi(f^{n+1}+g^{n+1})}{2}\right),$$
$$g^{n+1} = g^n + \frac{\Delta t}{\varepsilon}\left(\frac{f^{n+1}-g^{n+1}}{2} - \frac{\varphi(f^{n+1}+g^{n+1})}{2}\right). \tag{9}$$

Since $f^{n+1} + g^{n+1} = f^n + g^n$ we can write

$$f^{n+1} = f^n + \frac{\Delta t}{\varepsilon}\left(\frac{\varepsilon}{\varepsilon+\Delta t}\frac{g^{n+1}-f^{n+1}}{2} + \frac{\varepsilon}{\varepsilon+\Delta t}\frac{\varphi(f^{n+1}+g^{n+1})}{2}\right),$$
$$g^{n+1} = g^n + \frac{\Delta t}{\varepsilon}\left(\frac{\varepsilon}{\varepsilon+\Delta t}\frac{f^{n+1}-g^{n+1}}{2} - \frac{\varepsilon}{\varepsilon+\Delta t}\frac{\varphi(f^{n+1}+g^{n+1})}{2}\right).$$

Or equivalently

$$f^{n+1} = (1-\lambda)f^n + \lambda\left(\frac{g^{n+1}-f^{n+1}}{2} + \frac{\varphi(f^{n+1}+g^{n+1})}{2}\right),$$
$$g^{n+1} = (1-\lambda)g^n + \lambda\left(\frac{f^{n+1}-g^{n+1}}{2} - \frac{\varphi(f^{n+1}+g^{n+1})}{2}\right), \tag{10}$$

with $\lambda = \dfrac{\varepsilon}{\varepsilon+\Delta t}$. Now let us define the probability density

$$P^n(\xi) = \begin{cases} f^n, & \text{if} \quad \xi = +\sqrt{a}, \\ g^n, & \text{if} \quad \xi = -\sqrt{a}, \\ 0, & \text{elsewhere.} \end{cases}$$

Note that $0 \leq P^n(\xi) \leq 1$ and $\sum_\xi P^n(\xi) = 1$. Moreover

$$\sum_\xi P^{n+1}(\xi) = \sum_\xi P^n(\xi) = 1.$$

The system (10) can be seen as the evolution of the probability function $P^n(\xi)$ according with

$$P^{n+1}(\xi) = \begin{cases} (1-\lambda) f^n + \lambda E_f(u^n), & \text{if} \quad \xi = +\sqrt{a}, \\ (1-\lambda) g^n + \lambda E_g(u^n), & \text{if} \quad \xi = -\sqrt{a}, \\ 0, & \text{elsewhere}, \end{cases} \tag{11}$$

where $E_f(u) = \dfrac{u + \varphi(u)}{2}$ and $E_g(u) = \dfrac{u - \varphi(u)}{2}$.

We remark that, since $\varepsilon > 0$, the condition $0 \leq \lambda \leq 1$ is satisfied. Obviously, (11) represents a convex combination of two probability densities. In fact, thanks to (8), we have $E_f(u) \geq 0$, $E_g(u) \geq 0$ and $E_f(u) + E_g(u) = 1$.

3 Monte Carlo Algorithm

In order to develop a Monte Carlo game to approach the solution to the kinetic system (5), in addition to a good random number generator, we need a way to sample particles from an initial data and some other basic tools which are described with details in the lecture notes [9]. Thus, we consider two families of particles that defines samples of our probability density $P^n(\xi)$.

Let us define with $\{\xi_1, \xi_2, \ldots, \xi_N\}$ the particle samples, we know that $\xi_j = +\sqrt{a}$ or $\xi_j = -\sqrt{a}$ with probability f^n or g^n respectively. We have the relation

$$P^{n+1}(\xi) = (1-\lambda) P^n(\xi) + \lambda E^n(\xi), \tag{12}$$

where $E^n(\xi)$ is defined as: $$E^n(\xi) = \begin{cases} \dfrac{u + \varphi(u)}{2}, & \text{if} \quad \xi = +\sqrt{a}, \\ \dfrac{u - \varphi(u)}{2}, & \text{if} \quad \xi = -\sqrt{a}, \\ 0, & \text{elsewhere}. \end{cases}$$

Hence, the relaxation stage (7) can be solved in the following way:

- Given a particle sample ξ the evolution of the sample during the time integration process is performed according to :
  - with probability $(1-\lambda)$ the sample is unchanged
  - with probability λ the sample is replaced with a sample from $E^n(\xi)$
- To sample a particle from $E^n(\xi)$ we proceed as follows:
  - with probability $\dfrac{u + \varphi(u)}{2}$ take $\xi = +\sqrt{a}$
  - with probability $\dfrac{u - \varphi(u)}{2}$ take $\xi = -\sqrt{a}$

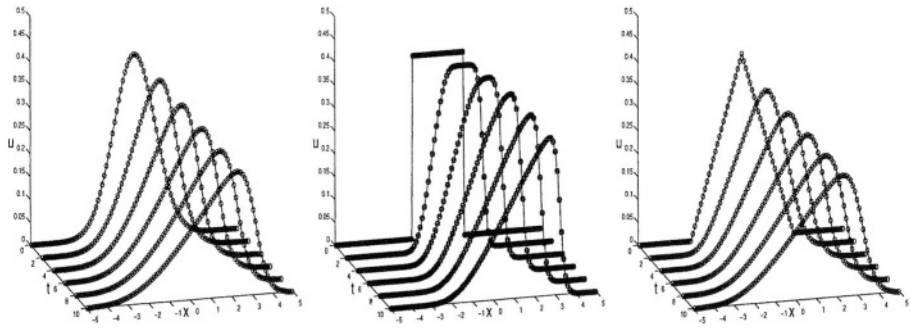


Fig. 1. Evolution in time-space of Gaussian distribution (right), box distribution (medium) and cone distribution (left) using the Monte Carlo approach.

Note that the relaxation stage is well defined for any value of ϵ. In particular as $\epsilon \to 0$ we have $\lambda \to 1$ and thus particles are all sampled from the local equilibrium $E^n(\xi)$. To generate particles the spatial domain is first divided into cells $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ with stepsize Δx and centered in x_i. Then particles are generated from a given piecewise initial data in each cell and are randomly distributed around the cell center x_i.

Once the particle distribution is updated by the above steps, the transport stage (6) of the splitting is realized by advecting the position of the particles according to their speeds. Thus, given a sample of N particles at positions $x_1, x_2, \cdots, x_N$ and speeds $\xi_1, \xi_2, \cdots, \xi_N$ (equal either to $+\sqrt{a}$ or $-\sqrt{a}$) the new position of the particle sample $\{x_i, \xi_i\}$ is simply

$$x_i^{\text{new}} = x_i^{\text{old}} + \Delta t \xi_i, \qquad i = 1, \ldots, N,$$

where x_i^{new} and x_i^{old} are respectively, the new and old positions of the sample ξ_i, and Δt is the time stepsize.

Remark 1 *The Monte Carlo method presented in this paper applies also if* $u > 0$ *and* $u \neq 1$. *For this case*

$$P^n(\xi) = \begin{cases} \dfrac{f^n}{u^n}, & \text{if} \quad \xi = +\sqrt{a}, \\ \dfrac{g^n}{u^n}, & \text{if} \quad \xi = -\sqrt{a}, \\ 0, & \text{elsewhere.} \end{cases} \quad \text{and} \quad E^n(\xi) = \begin{cases} \dfrac{u + \varphi(u)}{2u}, & \text{if} \quad \xi = +\sqrt{a}, \\ \dfrac{u - \varphi(u)}{2u}, & \text{if} \quad \xi = -\sqrt{a}, \\ 0, & \text{elsewhere.} \end{cases}$$

It is easy to verify that in this case, P^n *defines a probability (i.e.,* $0 \leq P^n(\xi) \leq 1$ *and* $\sum_\xi P^n(\xi) = 1$*).*

4 Numerical Results

In what follows we present numerical results obtained for some tests on Burgers equation given by the equation (1) with the flux function defined as $\varphi(u) = \frac{u^2}{2}$.

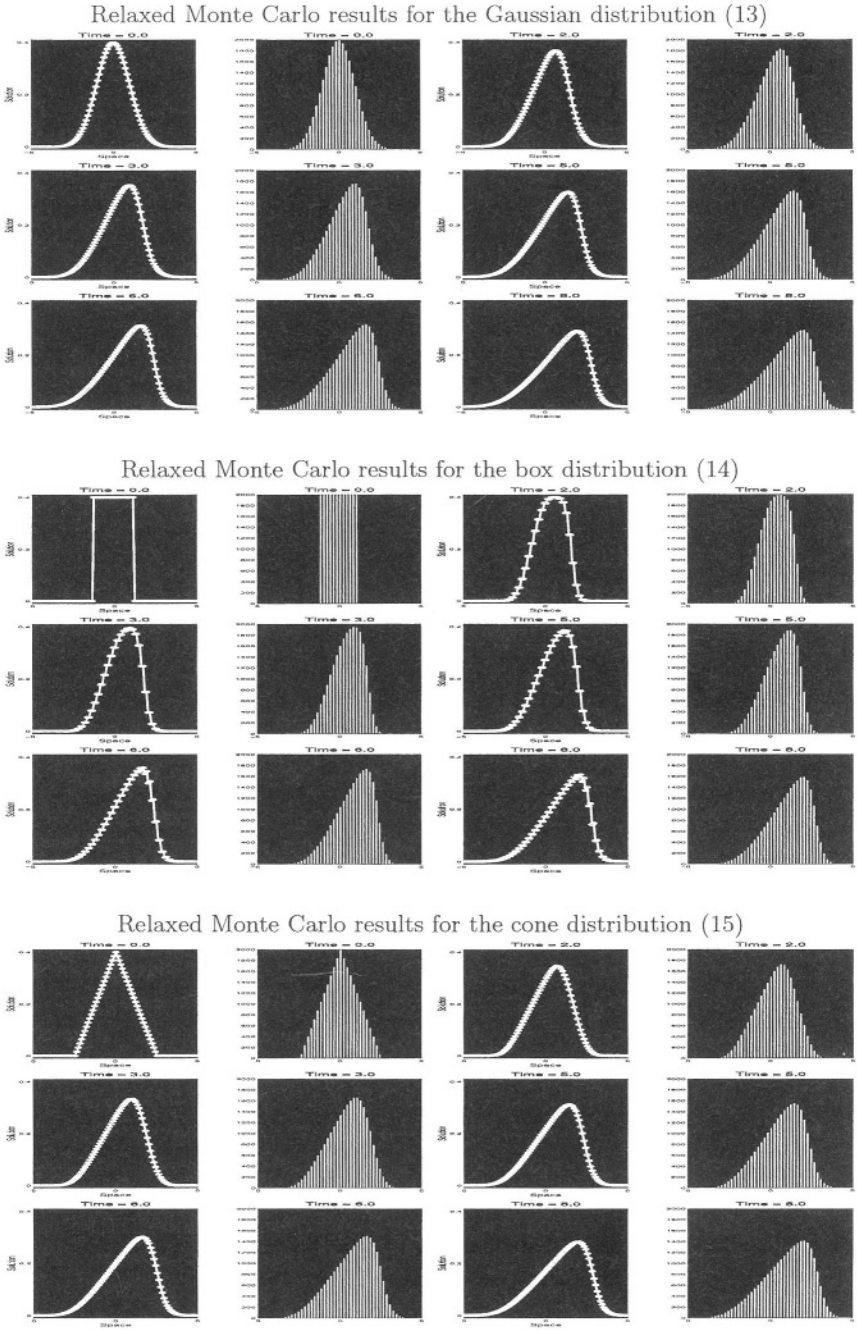


Fig. 2. Relaxed Monte Carlo results for the distributions (13)-(15).

Note that this flux function satisfies the assumptions (8). We solve the Burgers equation in the space interval $[-\delta, \delta]$ subject to periodic boundary conditions and using the following three different initial data:

1. Gaussian distribution

$$u_0(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}. \tag{13}$$

2. Box distribution

$$u_0(x) = \begin{cases} \dfrac{2}{\delta}, & \text{if} \quad |x| \leq \dfrac{\delta}{4}, \\ 0, & \text{elsewhere.} \end{cases} \tag{14}$$

3. Cone distribution

$$u_0(x) = \begin{cases} -\dfrac{4|x|}{\delta^2} + \dfrac{2}{\delta}, & \text{if} \quad |x| \leq \dfrac{\delta}{2}, \\ 0, & \text{elsewhere.} \end{cases} \tag{15}$$

Note that the total mass in these initial data is equal to unity, $\int_{\mathbb{R}} u_0(x)dx = 1$. In all our computations we used $\delta = 5$, the spatial interval is discretized into $M = 100$ gridpoints uniformly spaced. The number of particles N is set to 10^4 which is large enough to decrease the effect of the fluctuations in the computed solutions. Here we present only results for the relaxed case (*i.e.* $\varepsilon = 0$). Results for the relaxing case (*i.e.* $\varepsilon \neq 0$) can be done analogously. The time step Δt is chosen such a way the CFL = 1 and the computation have been performed with a single run for different time $t \in [0, 10]$.

In Fig. 1 we display the evolution in time of these initial distributions. For the sake of comparison, we have included in the figures the results obtained by the Lax-Friedrichs method [4], plotted by the solid lines. In the Monte Carlo approach the solution $u(x_i, t)$ at the gridpoint x_i and time t has been reconstructed by averaging the number of particles in each cell as

$$u(x_i, t) = \frac{n(x_i, t)}{N} \sum_{j=1}^{M} u(x_j, t = 0), \qquad i = 1, \ldots, M,$$

where $n(x_i, t)$ denotes the number of the particles localized in the cell $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ at time t. As can be seen the shock is well captured by the Monte Carlo method. Fig. 2 shows again the results for the initial data (13)-(15) along with the evolution of particle distribution in the space interval for six different times. The Monte Carlo approach preserves the positivity of the solution as well as the conservation of mass

$$\int_{\mathbb{R}} u(x, t)dx = \int_{\mathbb{R}} u_0(x)dx = 1, \qquad \forall\, t > 0.$$

We would like to mention that the new Monte Carlo approach can approximate conservation laws with diffusive source terms, for example viscous Burgers equations. The diffusion stage in the algorithm can be treated, for example, by the well-known Random walk method.

5 Concluding Remarks

We have presented a simple Monte Carlo algorithm for the numerical solution of conservation laws and relaxation systems. The algorithm takes advantage of the relaxation model associated to the equation under consideration which can be regarded as the evolution in time of a probability distribution. Although we have restricted our numerical computations to the case of one-dimensional scalar problems, the most important implication of our research concerns the use of effective Monte Carlo procedures for multi-dimensional systems of conservation laws with relaxation terms similarly to the Broadwell system and the BGK model in rarefied gas dynamics. Our current effort is therefore to extend this approach to systems of conservation laws in higher space dimensions. Another extension will be to couple the Monte Carlo method at the large ϵ scale with a deterministic method at the reduced small ϵ scale as in [2] for a general relaxation system. Finally we remark that the Monte Carlo approach proposed in this paper is restricted to first order accuracy. A second order method it is actually under study.

Acknowledgements. The work of the second author was done during a visit at Ferrara university. The author thanks the department of mathematics for the hospitality and for technical and financial support. Support by the European network HYKE, funded by the EC as contract HPRN-CT-2002-00282, is also acknowledged.

References

1. Bird G.A.: Molecular Gas Dynamics. Oxford University Press, London, (1976)
2. Caflisch R.E., Pareschi L.: An implicit Monte Carlo Method for Rarefied Gas Dynamics I: The Space Homogeneous Case. J. Comp. Physics **154** (1999) 90–116
3. Jin S., Xin, Z.: The Relaxation Schemes for Systems of Conservation Laws in Arbitrary Space Dimensions. Comm. Pure Appl. Math. **48** (1995) 235–276
4. LeVeque Randall J.: Numerical Methods for Conservation Laws. Lectures in Mathematics ETH Zürich, (1992)
5. Nanbu K.: Direct Simulation Scheme Derived from the Boltzmann Equation. J. Phys. Soc. Japan, **49** (1980) 2042–2049
6. Natalini, R.: Convergence to Equilibrium for Relaxation Approximations of Conservation Laws. Comm. Pure Appl. Math. **49** (1996) 795–823
7. Pareschi L., Wennberg B.: A Recursive Monte Carlo Algorithm for the Boltzmann Equation in the Maxwellian Case. Monte Carlo Methods and Applications **7** (2001) 349–357
8. Pareschi L., Russo G.: Time Relaxed Monte Carlo Methods for the Boltzmann Equation. SIAM J. Sci. Comput. **23** (2001) 1253–1273
9. Pareschi L., Russo G.: An Introduction to Monte Carlo Methods for the Boltzmann Equation. ESAIM: Proceedings **10** (2001) 35–75
10. Pullin D.I.: Generation of Normal Variates with Given Sample. J. Statist. Comput. Simulation **9** (1979) 303–309
11. Strang, G.: On the Construction and the Comparison of Difference Schemes. SIAM J. Numer. Anal. **5** (1968) 506–517

A Parallel Implementation of Gillespie's Direct Method

Azmi Mohamed Ridwan, Arun Krishnan*, and Pawan Dhar

Bioinformatics Institute, 30 Biopolis Street, #07-01 Matrix, Singapore 138671.
{azmi,arun,pk}@ bii.a-star.edu.sg

Abstract. Gillespie's Direct Method Algorithm (1977), is a well-known exact stochastic algorithm for simulating coupled reactions that requires the use of random numbers to calculate which reaction occurs next and when it occurs. However this algorithm is serial in design. For complex chemical systems, this will involve computationally intensive requirements with long simulation runs. This paper looks at decreasing execution times by attempting to parallelize this algorithm through splitting the computational domain into smaller units which will result in smaller computations and thus faster executions.

1 Introduction

Stochastic simulation has become an important tool for scientists in modeling complex chemical systems. Traditional methods of solving these systems usually involve expressing them mathematically through the use of ordinary differential equations which are notoriously difficult to solve. Gillespie's Direct Method [1] was a breakthrough in the sense that it could accurately and feasibly simulate these systems stochastically on what were then state-of-the-art computer systems. Since then, there have been improvements to the algorithm. One prominent recent modification is by Gibson[2].

The main disadvantage of Gillespie's algorithm is that it is essentially serial in nature. For complex chemical systems, this would result in computationally intensive executions and long simulation runs. The purpose of this paper is to study the feasibility of improving execution times through parallelization. The availability of compute clusters and parallel programming libraries (such as MPI, OpenMP and PVM) makes this possibility most attractive.

There are essentially two methodologies for achieving faster results. The first is known as **MRIP (Multiple Replication in Parallel)** [3]. The other method decomposes the problem domain into smaller sub-domains having fewer molecular species and having an instance of the Gillespie algorithm running. However there is a need to maintain the fundamental assumptions of the Gillespie algorithm while parallelizing in this manner. In this paper, we describe the procedure for using Domain Decomposition in order to parallelize Gillespie's Direct Method Algorithm. We will also show the application of the methods for a few chemical systems and the speedups obtained.

* To whom correspondence should be addressed

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 284–291, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Methodology

2.1 Gillespie's Direct Method Algorithm

This section will briefly highlight the important aspects of Gillespie's algorithm. The reader is encouraged to read [1] for a more detailed description and proofs. The Gillespie algorithm is impressive in its simplicity. The algorithm begins by initializing the stochastic rate constants for the reactions and the initial populations of the various species. A loop is then started with the following steps. First, the probability of each reaction occuring at the current time is calculated. Then, random numbers are used to determine which reaction should occur as well as to calculate the next time step. The time is then incremented and the species' population are adjusted according to the reaction selected. Finally, the loop repeats itself until stopping criterions are met.

2.2 Data Collection

One of the primary concerns in writing a parallel version of the Gillespie algorithm is the collation of the data. This occurs as a result of the use of random numbers in the algorithm. The implementation of random numbers in computer programs is almost always pseudo-random which requires an initial seed. Thus for proper solutions, each instance of the program must use a unique initial seed. However this will mean that each instance will have a unique time evolution. This then implies that in all probability, there will be no corresponding data points for any of the instances for a specific time. One simple solution, termed here as 'nearest point', would be to use the various species population at the point of the latest reaction before that collection point.

2.3 Domain Decomposition (DD) Method

The Domain Decomposition method involves dividing the entire species into smaller independent populations. The fundamental assumption for Gillespie's algorithm is that for a fixed container of volume V, the system should be in thermal equilibrium thus implying that the molecules will at all times be distributed randomly and uniformly. It remains to be seen whether the Domain Decomposition method would lead to incorrect results due to a violation of this fundamental assumption.

To develop the Domain Decomposition method, the Gillespie algorithm needs to be examined. Although for the most part the algorithm remains unchanged, there is a need to reexamine the rate constants. While the deterministic rate constants k_i are assumed to be constant, the stochastic rate constants are not necessarily so. To illustrate this, we list general relationships between the deterministic (k_i) and stochastic (c_i) rate constants. The relationships are obviously dependent on the type of reaction involved. When the molecular species are divided by the number of sub-domains, the respective stochastic rate constants must be adjusted accordingly to maintain constant deterministic rate constants.

$$\sum_{j=1}^{N} S_j \rightarrow \qquad NS_1 \rightarrow$$

$$k_i = V^{j-1} c_i \qquad k_i = \frac{V c_i}{N!}$$

2.4 Domain Decomposition Method with Synchronization (DDWS)

As stated previously, the fundamental assumption of the Gillespie algorithm is the fact that the system in the volume is supposed to be well-mixed. However the DD method could violate this if the sub-simulations produce large differences in the population of a species. Hence in order to improve the accuracy of the parallel Gillespie algorithm, there is a need to introduce some form of interaction between the sub-domains. Schwehm [4] implements this by randomly exchanging molecules between neighboring sub-domains. This form of diffusion is motivated by the way partial differential equations are solved numerically. This implementation however is very costly as large numbers of point-to-point messages must be used.

A simpler method of averaging out the species' populations at the appropriate step is used here. This would be more in line with the spirit of Gillespie's original algorithm. In the implementation of this algorithm, synchronizations are done at regular time intervals (in fact, in the same step when the population data are collected). This is easy to implement as the number of synchronizations done would be constant regardless of the total number of iterations for the Gillespie loop.

3 Chemical Reactions

To illustrate the parallel algorithms, we have chosen two types of chemical reactions: One whose simulations produces asymptotic results at steady states and the other that produces periodic results.

3.1 Michaelis-Menten Reactions

The Michaelis-Menten system is a set of well-known, enzyme catalyzed reactions that involves the binding of a substrate to an enzyme. These reactions are given below. The Michaelis-Menten system is an example of a 'deterministic' system.

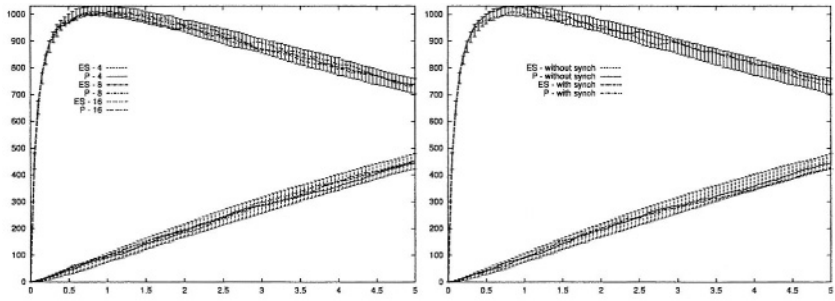
$$E + S \xrightarrow{k_1} ES \tag{1a}$$

$$ES \xrightarrow{k_2} E + S \tag{1b}$$

$$ES \xrightarrow{k_3} E + P \tag{1c}$$

For the implementation of the DD method, the rate constants must be modified as stated in Sect. 2.3. Looking at the MM equations (1b) and (1c), the

deterministic and stochastic reaction rate constants are directly related. Thus if the volume is divided into its sub-domains, the rate constants for these equations will remain unchanged. For (1a) the deterministic and stochastic rate constants are related by a volume factor. Therefore if the volume is divided into N sub-domains, then the stochastic rate constants must be increased by the same factor.



(a) Comparison of solutions from 3 DD plots with 4, 8 and 16 sub-domains for species ES and P.

(b) Comparison of a DD solution and a DDWS solution.

Fig. 1.

Figure 1a compares the results for a multiple replication (50 runs) simulation of the Serial algorithm (denoted by the verical error bars) and the DD method (with 4,8 and 16 sub-domains) for two of the molecular species. It can be seen that the DD method holds up well for these sets of reaction even for 16 sub-domains (with initial enzyme and substrate populations of 75 molecules each.) It would be difficult to distinguish the DD method results from those of the serial runs. Figure 1b shows a comparison between the serial solution , the DD method and the DDWS method (both of which uses the same initial random seeds). As can be seen, the addition of the synchronization does not lead to a qualitative difference in the results of the simulation.

3.2 Lotka Reactions

The Lotka Reactions[1], are an example of oscillatory reactions. It is a well-known system that has been adopted in many branches of science, most notably in ecology where it represents a crude predator-prey model.

$$\tilde{X} + Y_1 \xrightarrow{k_1} 2Y_1 \tag{2a}$$

$$Y_1 + Y_2 \xrightarrow{k_2} 2Y_2 \tag{2b}$$

$$Y_2 \xrightarrow{k_3} Z \tag{2c}$$

Note that in the first reaction, the bar over the X indicates that X is *open* i.e. the molecular population level of this species is assumed to be constant. Different instances of the Direct Method will yield similar frequencies but they will be out of phase with each other. Also the amplitude variations may differ significantly.

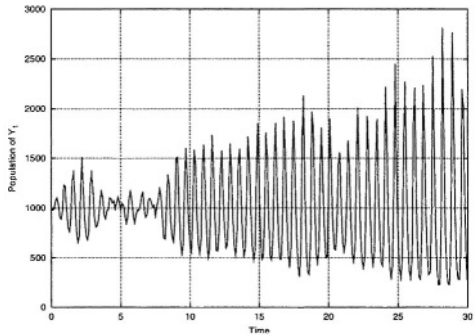
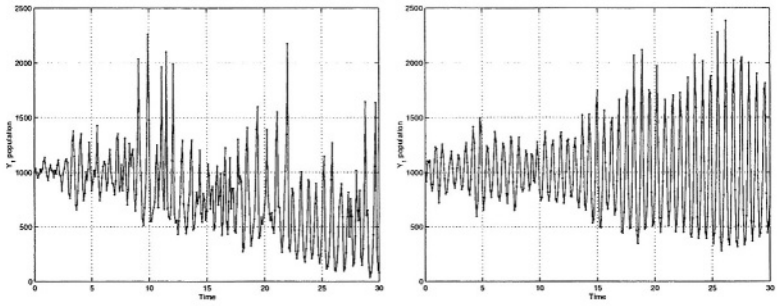


Fig. 2. Plot of species Y_1 for reaction (2) using the Serial implementation of the Direct method with $\bar{X} = 100, Y_1 = Y_2 = 1000, Z = 0, c_1X = 10, c_2 = 0.01, c_3 = 10$.

Figure 2 shows a plot of species Y_1 for a serial run . The Lotka reactions have steady-state solutions for Y_1, Y_2 at $Y_{1s} = c_3/c_2$ $Y_{2s} = c_1X/c_2$. Figure 2 demonstrates this with Y_1 oscillating around $Y_{1s} = c_3/c_2 = 1000$.

Figure 3a shows a plot of species Y_1 for the DD method applied to the Lotka reactions for 4 sub-domains. The solutions obtained are clearly incorrect as Y_1 does not oscillate around the steady-state solutions. To understand the reason for this, we must take note of what occurs in each sub-domain. As stated in Sect. 2.3, when the DD method is used, the stochastic rate constants must be modified appropriately. This results in a smaller steady-state solution for Y_1, Y_2 in each sub-domain. The oscillations will then occur around these values. If the value of Y_1 were to reach 0, only reaction (2c) is viable. Thus in the sub-domain method, imbalances may occur where some sub-domains may be void of any Y_1 and Y_2 species.

Figure 3b is an implementation of the Lotka reaction using the DDWS. The figure suggests that the correct solution has been derived as it resembles a serial solution (i.e. the solution oscillates around the steady-state value). This solution works because, synchronizations prevent the Y_1 species from being extinct in any one sub-domain (provided there is a nonzero population of Y_1 in any of the sub-domains).



(a) Plot of species $Y_1$for reaction (2) using the DD method.

(b) Plot of species Y_1 for reaction (2) using the DDWS

Fig. 3.

3.3 Brusselator Reactions

The Brusselator[1] is another set of well known reactions that represents oscillatory systems. Unlike the Lotka reactions previously, it is 'positively stable' and the amplitude of oscillations are more consistent with each other. The reactions can be expressed as:

$$\bar{X}_1 \xrightarrow{k_1} Y_1 \tag{3a}$$

$$\bar{X}_2 + Y_1 \xrightarrow{k_2} Y_2 + Z_1 \tag{3b}$$

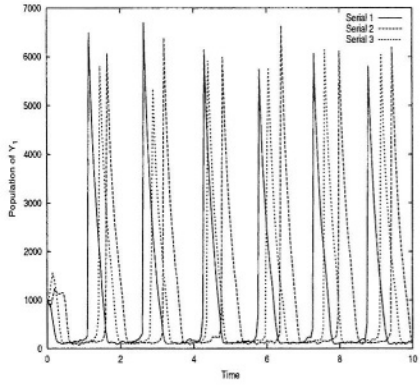
$$2Y_1 + Y_2 \xrightarrow{k_3} 3Y_1 \tag{3c}$$

$$Y_1 \xrightarrow{k_4} Z_2 \tag{3d}$$

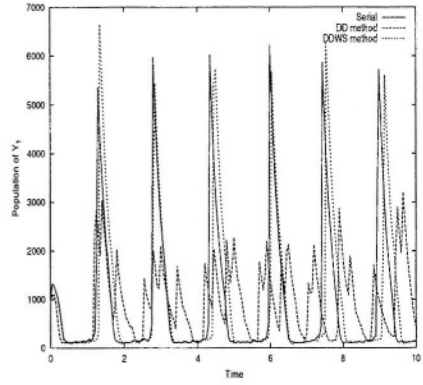
The serial plots (Fig. 4a) of the Brusselator reactions show that while the periods and amplitudes of the oscillations for the three plots are similar, the phases are not. As such when the DD method is used, the population of the species Y_1, Y_2) between the sub-domains are out of phase with each other, resulting in clearly inaccurate results (Fig. 4b - long dashed line). However once synchronizations are used, the solution obtained appears to be consistent with a serial solution of the reactions(Fig. 4b - short dashed line).

4 Performance Results

Figure 5 shows speedup graphs for the Michaelis-Menten reactions with two different sets of intial values together with the Brusselator reaction simulation using the values in Fig. 4. For the MM simulations, 'small' corresponds to: $E = 12000$, $S = 12000, ES = 0, P = 0, c_1 = 0.01, c_2 = 1, c_3 = 100$ while 'large' corresponds to: $E = 36000$, $S = 36000, ES = 0$, $P = 0, c_1 = 0.01, c_2 = 250$, $c_3 = 1$.



(a) The Brusselator plot of species Y_1 for 3 Serial runs with initial populations as $Y_1 = 1000, Y_2 = 2000, c_1X_1 = 5000, c_2X_2 = 50, c_3 = 0.00005, c_4 = 5$.



(b) The Brusselator plot of species Y_1 for Serial, DD method, DDWS

Fig. 4.

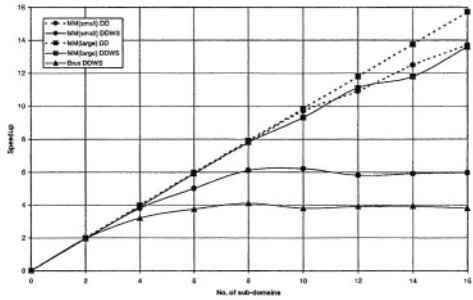


Fig. 5. Speedup for Michaelis Menten and Brusselator reactions.

It is quite apparent from the comparison of smaller and larger initial values for the MM reactions that, while the reactions remain the same, the speedup graphs differ significantly. The speedup for the MM reactions thus depend on the total number of iterations of Gillespie loop which in turn depends on the initial population values and rate constants used. The speedup for the DD method is much better than the DDWS method. However as the number of sub-domains are increased, the speedup will plateau and eventually decrease as the computation done in each sub-domain decreases.

This plateauing in the speedup is more apparent when synchronizations are introduced. This results in a constant number of synchronizations regardless of the initial population of the species and rate constants. As the number of sub-

domains increases, the number of iterations between synchronization decreases. Also the collective operation used (*MPI_Allreduce*), has to handle an increasing number of processes thus increasing the cost of using it.

As a comparison, the speedup for the Brusselator equations using the values used previously are also shown. As stated before, a correct solution is only derived when synchronizations are used. The speedup shows the inevitable plateauing as the number of sub-domains is increased.

5 Summary

In this paper we have presented an approach to parallelizing Gillespie's Direct Method algorithm keeping in mind the need to remain consistent with the fundamental assumptions of the algorithm.The basic premise of the DD method is to divide the molecular population into smaller sub-domains where computations can be completed faster.

For oscillatory chemical systems, such as the Lotka reactions and the Brusselator, periodic synchronizations are needed. This introduces diffusion which prevents buildup of any particular species in a sub-domain thus ensuring the well-mixed nature of the whole domain.

The speedups obtained by the parallel Gillespie Algorithm show a "plateauing" effect in the presence of synchronizations (DDWS method). The DD method, without synchronizations shows very good speedup; however, its use is restricted to non-oscillatory systems.

Despite the fact that the methodology works for the systems under study here, it is not possible to state categorically as to whether it would work for any arbitrary system. Work remains to be done to study the efficacy of this method on larger, more highly coupled systems.

References

1. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. **81** (1977) 2340–2361
2. Gibson, M.A.: Computational methods for stochastic biological systems. Ph.D. Thesis, Calif. Inst. Technology (2000)
3. Ewing, G., McNickle, D., Pawlikowski, K.: Multiple replications in parallel: Distributed generation of data for speeding up quantitative stochastic simulation. In: Proc. of IMACS'97, 15th Congress of Int. Association for Mathematics and Computers in Simulation, Berlin, Germany (1997) 397–402
4. Schwehm, M.: Parallel stochastic simulation of whole-cell models. In: ICSB 2001 proceedings. (2001)

Simulation of Deformable Objects Using Sliding Mode Control with Application to Cloth Animation

Farshad Rum and Brandon W. Gordon

Control and Information Systems (CIS) Laboratory, Department of Mechanical and Industrial Engineering, Concordia University, Montreal, Quebec, Canada,
bwgordon@me.concordia.ca

Abstract. A new method is presented for simulation of deformable objects that consist of rigid and flexible structural elements using the control based Singularly Perturbed Sliding Manifold (SPSM) approach. The problem is multi-scale due to its rigid and flexible components and forms a set of differential-algebraic equations. The problem is formulated as a set of ODEs with inequality constraints by allowing some deviations in the rigid links. The SPSM approach is particularly well suited for such problems. It is shown that this method can handle inconsistent initial conditions, and it allows the user to systematically approximate the equations due to its robustness properties. The inherent attractivity of the sliding dynamics enables the method to handle sudden changes in position, velocity or acceleration while satisfying geometrical constraints. The desired level of accuracy in constraint errors is achieved in a finite time and thereafter. Moreover, the new approach is explicit and capable of performing multi-rate and real-time simulations. Finally, it is shown that the SPSM approach to simulation requires the inversion of a smaller matrix than comparable implicit methods. The result is significantly improved performance for computationally expensive applications such as cloth animation.

1 Introduction

Animation of deformable structures such as hair, chain, cloth and jelly-type materials has imposed some challenging problems due to their multi-scale nature. Such problems have little resistance in bending/shear directions, but are often very stiff with hard constraints in elongation/shear directions. The resulting set of equations is therefore stiff and traditional explicit methods usually cannot handle them efficiently due to small time steps that they demand.

Implicit methods on the other hand can handle larger time steps; however, they have no built-in mechanism to deal with algebraic constraints. For a visually realistic animation of cloth a maximum deviation of 10% in the stiff direction (usually elongation) is recommended [1]; otherwise, the cloth will become like rubber. Traditionally it has been up to the user to select proper parameters in the stiff direction so that deviations do not exceed their limit. Such an approach might require a lot of trial and errors and if a large gain is necessary it can substantially limit the allowable time step, which was the reason for incorporating the method in the first place.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 292–299, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Another approach has been to first solve the equations by imposing and correcting the constraints [2]. Such a method requires additional overhead and can result in artifacts that may in turn require another algorithm to correct the momentum/energy alterations caused by enforcing the constraints [3]. The proposed SPSM approach, however, has a built-in mechanism [4] to handle the limits on the constraints while at the same time solving the set of ODEs. Another merit of this approach is that due to its *attractive boundary layer* [5] it can be combined with existing codes that require sudden corrections in position, velocity or acceleration of particles in order to satisfy various geometrical constraints.

As we show in Sect. 2 the governing equations of a flexible object can be easily written as a set of ordinary differential equations in bending and shear direction and constrained by a set of algebraic equations in elongation directions. This class of equations is commonly referred to as differential-algebraic (DAEs) equations. In general, one can write such equations as:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{z}) \,, \quad \mathbf{0} = \mathbf{g}(t, \mathbf{x}, \mathbf{z}) \tag{1}$$

with $\mathbf{x} \in \Re^n$, $\mathbf{z} \in \Re^m$, $\mathbf{f} : \Re \times \Re^n \times \Re^m \to \Re^n$, and $\mathbf{g} : \Re \times \Re^n \times \Re^m \to \Re^m$. Where nonlinear functions $\mathbf{f}$ and $\mathbf{g}$ respectively represent the ODE part and algebraic constraints of the DAE.

In most cases the accuracy level required for constraints allows us to permit a certain amount of error, ε. Therefore the above set of DAEs can be relaxed to the following set of ODEs with algebraic inequalities:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{z}) \,, \quad \left|\mathbf{g}_i(t, \mathbf{x}, \mathbf{z})\right| \le \varepsilon_i \,, \; i = 1, \ldots, m \tag{2}$$

where $\mathbf{g}_i(t, \mathbf{x}, \mathbf{z})$ represents each algebraic constraint and m is the total number of them. The common amount of ε used in simulation of cloth objects is 10% of the rest length in that link.

Singularly perturbed sliding manifold, SPSM, is a recent method developed to particularly attack problems of the type given by equations (2). SPSM equations can be efficiently solved by any explicit method and this allows us to perform a multi-rate simulation [6]; moreover, this approach is object-oriented thus any simulation code developed by this approach can be easily integrated with other simulation codes [6]. SPSM realization is a robust method that allows us to ignore various terms and make efficient simulations in a systematic manner. Reaching phase and locking properties of the sliding control [5] translate into two desirable properties in our application. Firstly the fact that inconsistent initial conditions that are an issue for BDF methods [7] are systematically dealt with; in fact it is possible to show that after a finite time the bounds on errors are satisfied [5] and secondly the locking property of the sliding control assures us that once these bounds are reached they will be satisfied for the rest of the simulation.

Finally as we will see in Sect. 3 SPSM method has to invert a smaller size matrix at each time step compared to BDF methods.

2 Problem Formulation

We model the flexible object as a collection of distributed masses connected to each other via rigid/flexible connections, which is also referred to as a particle system. These models have the ability to capture complex dynamical behaviors and are well suited to animation needs [3]. All the forces either internal or external simply depend on location and velocity of particles therefore in order to simulate such systems we only need to compute forces on each particle and two simple integrations will yield positions and velocities.

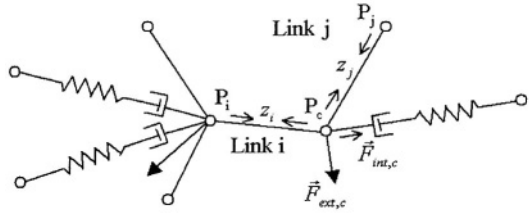


Fig. 1. A generic 3D section of a flexible object modelled as a particle system

As shown in Fig. 1 we can generally categorize the forces on particles as internal forces due to flexible connections, $\vec{\mathbf{F}}_{int}$, that are responsible for shear and bending behavior of the object, internal forces due to rigid connections, z, and finally the external forces, $\vec{\mathbf{F}}_{ext}$, that represent interaction forces between the object and the environment such as collisions, contacts, gravity, wind, *etc.* Take note that we are not modeling the rigid links as springs but rather keep their forces as unknowns for the controller to determine. As a result we have the following ODE:

$$\begin{cases} \dot{\vec{\mathbf{R}}}_k = \vec{\mathbf{V}}_k \\ \dot{\vec{\mathbf{V}}}_k = \dfrac{1}{m_k}\left(\sum_j z_j \hat{\mathbf{r}}_{kj} + \vec{\mathbf{F}}_{int,k} + \vec{\mathbf{F}}_{ext,k}\right), \quad k = 1,\ldots,n_p \end{cases} \tag{3}$$

with these constraints

$$0 = L_i - L_i^0, \quad i = 1,\ldots,n_l \ . \tag{4}$$

Here $\vec{\mathbf{R}}_k$ and $\vec{\mathbf{V}}_k$ represent the position and velocity of each particle, n_p is the number of particles, $\hat{\mathbf{r}}_{kj}$ is the unit vector from particle k to j that is at the other end of the rigid link connecting them. For constraints, L_i is the instantaneous length of the i^{th} link, L_i^0 is its desired length and n_l represents the total number of links.

By permitting the length of links to change as much as ε_i the constraints will be the following inequalities:

$$\left|L_i - L_i^0\right| \le \varepsilon_i, \quad i = 1,\ldots,n_l \ . \tag{5}$$

We start application of the SPSM method by introducing the following error variable:

$$w_i = L_i - L_i^0 \tag{6}$$

Differentiating w.r.t. time (see Fig. 1) one obtains:

$$\dot{w}_i = \hat{\mathbf{r}}_{ci} \cdot \left(\vec{\mathbf{V}}_i - \vec{\mathbf{V}}_c \right), \quad \ddot{w}_i = \frac{\left\| \vec{\mathbf{V}}_i - \vec{\mathbf{V}}_c \right\|^2 - \dot{w}_i^{\,2}}{L_i} + \hat{\mathbf{r}}_{ci} \cdot \left(\dot{\vec{\mathbf{V}}}_i - \dot{\vec{\mathbf{V}}}_c \right) . \tag{7}$$

Since

$$\dot{\vec{\mathbf{V}}}_c = \frac{1}{m_c} \left(\sum_j z_j \hat{\mathbf{r}}_{cj} + \vec{\mathbf{F}}_{int,c} + \vec{\mathbf{F}}_{ext,c} \right) \tag{8}$$

We can see that z terms appear in $\ddot{w}$. Therefore, according to definition of index of a DAE [7] our problem is of index three. The sliding surface designed by the SPSM method will then be:

$$s_i = \mu^2 \ddot{w}_i + 2\mu \dot{w}_i + w_i \tag{9}$$

where μ is a positive parameter that determines the dynamics of the fast motion. The SPSM method then designs a controller that forces the motion to the above desired dynamics. In order to see the effect of μ on error we recall the following result from [5]:

Lemma 1. If $|s_i(t)| \le \varepsilon_i$ for $t \ge t_r$ then the error and its derivatives are bounded by:

$$|w_i(t)| \le \varepsilon_i, \ |\dot{w}_i(t)| \le \frac{2\varepsilon_i}{\mu} \text{ and } |\ddot{w}_i(t)| \le \frac{4\varepsilon_i}{\mu^2} \text{ for } t \ge t_r \ . \tag{10}$$

Note that in the above lemma we did not assume that initial conditions are necessarily consistent. Sliding mode control guarantees that after a finite *reaching time,* t_r, the motion will be contained within the desired accuracy bound, $|s_i(t)| \le \varepsilon_i$, and the locking property guarantees that the motion will satisfy the required bounds ever after [5]. This in fact not only solves our ODE problem with inequality constraints, $|w_i(t)| \le \varepsilon_i$, but also keeps the error derivatives bounded after a finite reaching phase with no need for the initial conditions to be consistent.

In order to achieve the above goals we design a controller that determines the value of $\mathbf{v} = \dot{\mathbf{z}}$. The value of the link forces, $\mathbf{z}$, will then be obtained by integrating $\mathbf{v}$.

Differentiating $\ddot{w}_i$ w.r.t time and packing the $\ddot{\mathbf{w}}$ vector we can write:

$$\ddot{\mathbf{w}} = \mathbf{J}_\Omega \mathbf{v} + \boldsymbol{\beta}, \ \mathbf{J}_\Omega = \frac{\partial \ddot{\mathbf{w}}}{\partial \mathbf{z}} \tag{11}$$

Substituting in eq. (9) we obtain:

$$\dot{\mathbf{s}} = \mu^2\ddot{\mathbf{w}} + 2\mu\ddot{\mathbf{w}} + \mathbf{w} = \mu^2\mathbf{J}_\Omega\mathbf{v} + \mu^2\boldsymbol{\beta} + 2\mu\ddot{\mathbf{w}} + \mathbf{w} = \mathbf{J}_s\mathbf{v} + \boldsymbol{\alpha} \tag{12}$$

where

$$\mathbf{J}_s = \mu^2\mathbf{J}_\Omega\,,\ \boldsymbol{\alpha} = \mu^2\boldsymbol{\beta} + 2\mu\ddot{\mathbf{w}} + \mathbf{w}\,. \tag{13}$$

If we solve the above equation for $\mathbf{v}$ we can steer the sliding motion into the desired boundary layer. At this stage we incorporate ideas from sliding modes control and make use of its robustness properties. Since computation of exact $\boldsymbol{\alpha}$ can potentially be expensive we approximate it by $\hat{\boldsymbol{\alpha}}$. Moreover, it is possible to avoid inverting the original $\mathbf{J}_S$ that is potentially very large when we have a large number of rigid links in the object and use an approximate inverse denoted by $\hat{\mathbf{J}}_S^{-1}$. In [4] Gordon shows that if we compute $\mathbf{v}$ by the following controller:

$$\mathbf{v} = -\hat{\mathbf{J}}_s^{-1}\left(\hat{\boldsymbol{\alpha}} + \mathbf{K}\,\mathrm{diag}\left[sat\left(\frac{s_i}{\varepsilon_i}\right)\right]\right) \tag{14}$$

the motion will converge to its desired error bound after a short reaching phase and will stay there ever after, if the following conditions are satisfied:

$$\begin{aligned} &1\text{-}\ \mathbf{J}_s\hat{\mathbf{J}}_s^{-1}\mathbf{K} - \left|\, diag[\boldsymbol{\alpha} - \mathbf{J}_s\hat{\mathbf{J}}_s^{-1}\hat{\boldsymbol{\alpha}}]\,\right| \text{has to be positive diagonally dominant.} \\ &2\text{-}\ \mathbf{J}_s\hat{\mathbf{J}}_s^{-1}\mathbf{K} \text{ must be uniformly positive definite.} \end{aligned} \tag{15}$$

The sat(.) function used in eq. (14) is indeed the linear saturation function and is used to smooth the control and help us avoid the chattering phenomenon [5] common to sliding mode control methods.

In this work we simply invert the real jacobian matrix, $\hat{\mathbf{J}}_s^{-1} = \mathbf{J}_s^{-1}$, thus reducing (15) to:

$$\begin{aligned} &1\text{-}\ \mathbf{K} - \left|\, \mathrm{diag}[\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}]\,\right| \text{has to be positive diagonally dominant.} \\ &2\text{-}\ \mathbf{K} \text{ must be uniformly positive definite.} \end{aligned} \tag{16}$$

The above criteria can then be easily satisfied by a large enough $\mathbf{K}$. All we are left with is choosing $\hat{\boldsymbol{\alpha}}$ and computing the $\mathbf{J}_s$ matrix.

The exact expression for $\boldsymbol{\alpha}$ is given by eq. (13). One can notice that $\ddot{\mathbf{w}}$ and $\mathbf{w}$ have already been evaluated in the process of calculating $\mathbf{s}$ and choice of

$$\hat{\boldsymbol{\alpha}} = 2\mu\ddot{\mathbf{w}} + \mathbf{w} \tag{17}$$

does not involve much computational overhead. If we further take our gain matrix to be diagonal, $\mathbf{K} = \mathrm{diag}(k_i)$, the only sufficient condition we need to meet becomes:

$$k_i \geq \mu^2|\beta_i| \tag{18}$$

Given the fact that β_i mainly depends on the ODE (3) that is under control we do not have to retune the gains each time we try a new value for μ.

Consider the generic link i in Fig. 1 and the particle, P_c connecting it to link j. Using definition (11) with equations (7) and (8) yields:

$$\left[\mathbf{J}_\Omega\right]_{i,j} = \begin{cases} -\hat{\mathbf{r}}_{ci}.\hat{\mathbf{r}}_{cj} / m_c & \text{if } i \neq j \\ -1/m_{i1} - 1/m_{i2} & \text{if } i = j \text{ and none of the two end particles} \\ & \text{of link } i \text{ has an acceleration constraint} \\ -1/m_{i,\text{free}} & \text{if } i = j \text{ and only one end particle of link } i \\ & \text{does not have an acceleration constraint} \\ 0 & \text{if links } i \text{ and } j \text{ have no nodes in common} \end{cases} \tag{19}$$

In the above equation masses of the two end particles of link i are denoted by m_{i1} and m_{i2} and mass of the particle in link i that does not have an acceleration constraint is represented by $m_{i,\text{free}}$. Examples of an acceleration constraint could include when the particle is fixed at its place or when it is attached to another object, which is considerably more massive compared to the cloth. In the latter situation the acceleration of the attaching particle is mainly governed and constrained by the corresponding point in that object, for example consider the attachment points in animation of parachute or sail for cloth, or the connecting point of hair to an object.

Finally take note that the size of the jacobian matrix that has to be inverted is equal to the number of rigid links, n_l. We will make use of this fact in Sect. 3.

Remark 1. In simulations that we performed, the algorithm proved to be robust against programming errors that yielded a slightly wrong $\mathbf{J}_s$. Aside from the fact that an amount of error in permitted by eq. (15), if the user makes a mistake in recognizing if a particle's acceleration is or is not constrained, the constraint could be considered as a neglected external force on that particle. This simply induces an error in $\hat{\boldsymbol{\alpha}}$ and as demonstrated by (16) can be robustly cancelled by choosing a big $\mathbf{K}$, which apparently does not involve any additional overhead. This fact can be especially handy when our flexible object dynamically changes its connections with other objects, *e.g.* when the sails are torn and taken away by a strong wind! A more common case happens in interactive animations where some points of the flexible object are dynamically chosen and moved by the user. We successfully tested this idea in our simulations presented in Sect. 3.

3 Application to Cloth Animation

We adopt the structure proposed by Provot [2] that has proven effective in cloth animation [3]. The particles are rigidly linked to their adjacent horizontal and vertical particles. Shear spring-dampers attach immediate diagonally adjacent particles and finally bending characteristics are modeled by inserting spring and dampers between every other horizontally or vertically neighboring nodes.

In animations we study a n_r by n_c grid of particles and we set the maximum amount of deviation in rigid links to be less than 10% of their intended lengths. As mentioned in Sect. 2 in all simulations we simplified the programming task by ignoring constraints in particles in evaluation of $\mathbf{J}_s$.

In order to test the new algorithm we initially compressed the left side of the cloth and fixed it at its two left corners. This is a typical example of inconsistent initial conditions that commonly arise in cloth animation, *e.g.* to attach pieces of cloth to rigid bodies. It is obvious that finding a consistent set of initial conditions that locates particles in proper positions to give this simple geometric shape is potentially tedious. In our example we exerted a tiny impulse to one of the particles and the algorithm automatically located all particles in proper positions that satisfied all constraints (Figs. 2).

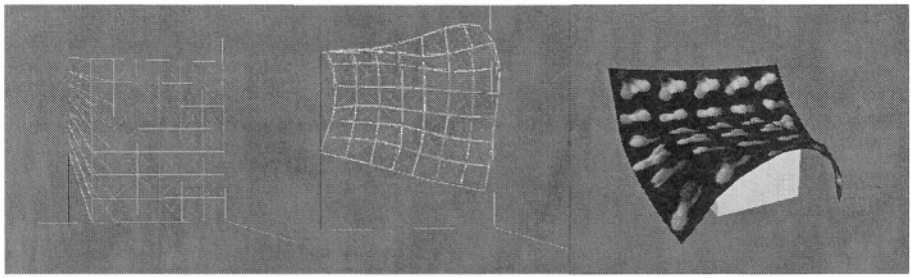


Fig. 2. Testing the reaching phase of the algorithm by inconsistent initial conditions (a) before and (b) after a finite reaching time and subject to wind and gravity (c) subject to an external geometrical constraint

In simulations the motion reaches the desired boundary layer in at most 2 seconds for the link with the most initial deviation. We also tested the algorithm simultaneously under gravity, wind and an external object that imposed geometrical perturbations as shown in Figs. 2.

In order to compare the SPSM approach with the well-established BDF method we run a number of simulations using the popular implicit method proposed in [8]. The results of simulations are presented in table (1). For simplicity we used a constant step size in all simulations. The set of equations in both cases are sparse and the CG method [9] is used in all simulations. Note that eq. (19) shows that $\mathbf{J}_s$ is symmetric and sparse with a maximum of seven non-zero elements on each row; because each link is at most connected to six other ones.

As shown in table (1) the SPSM method always has a smaller matrix size. Given a n_r by n_c grid of particles the number of links equals $n_l = n_r(n_c - 1) + n_c(n_r - 1)$ and the number of particles is $n_p = n_r n_c$. The size of the matrix in implicit method [8] is three times the number of particles, $3n_p = 3n_c n_r$, compared to the size of the $\mathbf{J}_s$, which is equal to the number of links or $n_l = 2n_c n_r - n_c - n_r$. Table 1 summarizes the results of simulations on an 8 × 8 grid of particles for 20 seconds. Simulations were run on a 2519 MHz processor. Note that in order to avoid instability in the implicit

method we had to reduce the amount of initial perturbations (left side compression) to one-third the amount shown in Figs. 2. It is evident from these results that the proposed approach is much more computationally efficient than standard implicit methods.

Table 1. Simulation results for SPSM and the implicit method [8]

Method	CPU time (sec)	Size of the sparse system	Max. stable cte. step size (sec)
Implicit	41.12	192	0.002
SPSM	5.47	112	0.019

4 Conclusions

In this paper we have studied the problem of simulating a deformable object consisting of rigid and flexible inner connections using sliding mode control. As our case study we developed a code that simulated a piece of cloth under initial and continuous disturbances and compared it to a popular implicit method. The new approach was seven times faster, more than three times more robust to disturbances, handled almost ten times larger time steps and finally led to a sparse system with nearly half the size of the implicit method.

References

1. House, D., Breen ,D. E., (eds.): Cloth Modeling and Animation. A.K. Peters, Natick Mass. (2000)
2. Provot, X.: Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behaviour. Proc. Graphics Interface 95 (1995) 147-154
3. Desbrun, M., Meyer, M., Barr, A.H.: Interactive Animation of Cloth-Like Objects for Virtual Reality. Journal of Vizualisation and Computer Animation (2000)
4. Gordon, B.W.: State Space Modelling of Differential-Algebraic Systems using Singularly Perturbed Sliding Manifolds. Ph.D Thesis, MIT, Mechanical Engineering Dept., August (1999)
5. Slotine, J.-J.E.: Sliding Controller Design for Nonlinear Systems. Int. J. Control. 40 (1984) 2
6. Gu, B., Asada, H.H.: Co-Simulation of Algebraically Coupled Dynamic Subsystems. ACC (2001) 2273-2278
7. Brenan, K., Campbell, S., Petzold, L.: Numerical Solution of Initial Value Problems in Differential-Algebraic Equations. Amsterdam, North-Holland (1989)
8. Baraff, D., Witkin, A..: Large Steps in Cloth Simulation. In: Cohen, M. (ed.): SIGGRAPH 98 Conference Proceedings. Annual Conference Series, P. Addison-Wesley, July (1998) 43-54
9. Shewchuk, J.: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Technical report CMU-CS-TR-94-125, Carnegie Mellon University (1994)

Constraint-Based Contact Analysis between Deformable Objects

Min Hong[1], Min-Hyung Choi[2], and Chris Lee[3]

[1] Bioinformatics, University of Colorado Health Sciences Center,
4200 E. 9th Avenue Campus Box C-245, Denver, CO 80262, USA
Min.Hong@UCHSC.edu

[2] Department of Computer Science and Engineering, University of Colorado at Denver, Campus Box 109, PO Box 173364, Denver, CO 80217, USA
minchoi@acm.org

[3] Center for Human Simulation, University of Colorado Health Sciences Center,
P.O. Box 6508, Mail Stop F-435, Aurora, CO 80045, USA
chris.lee@mesanetworks.net

Abstract. The key to the successful simulation of deformable objects is to model the realistic behavior of deformation when they are influenced by intricate contact conditions and geometric constraints. This paper describes constraint-based contact modeling between deformable objects using a nonlinear finite element method. In contrast to the penalty force based approaches, constraint-based enforcement of contact provide accuracy and freedom from finding proper penalty coefficients. This paper is focused on determining contact regions and calculating reaction forces at appropriate nodes and elements within the contact regions. The displacement and deformation of all nodes are dynamically updated based on the contact reaction forces. Our constraint based contact force computation method guarantees tight error bound at the contact regions and maintains hard constraints without overshoot or oscillation at the boundaries. In addition, the proposed method doesn't require us to choose proper penalty coefficients, thus greater numerical stability can be achieved and generally large integration steps can be utilized for the ODE solver. Contact conditions are formulated as nonlinear equality and inequality constraints and the force computation is cast into a nonlinear optimization problem. Our rigid-to-deformable and deformable-to-deformable contact simulation demonstrates that the non-penetration constraints are well maintained.

1 Introduction

With the increased demand for the visual realism in character animation, medical and scientific visualization, deformable object simulation is becoming a major issue. In medical visualization, for example, realistic deformation is quite complex to achieve where skin, muscle, ligaments and organs are all highly deformable and in constant contact. Thus far, deformable object simulation and animation have been addressed

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 300–308, 2004.
© Springer-Verlag Berlin Heidelberg 2004

from the modeling perspective where the main focus was to accurately and efficiently represent the deformation itself based on the given external forces [8]. Previous research demonstrated how deformable bodies are reacting and influenced by the known external forces, ignoring complex contact interplay between multiple deformable objects [1,11,17]. However, deformable object simulation should be addressed in a broad context of interaction with surrounding environments such as surface contacts and geometric constraints [14]. When two flexible objects collide, they exert reaction forces on each other resulting in the deformation of both objects. While many important breakthroughs have been made in modeling the deformation, the simulation of isolated deformable objects without an accurate contact model has few practical applications. Baraff et al. [2] have presented a flexible-body model that represents a compromise between the extremes of the nodal and rigid formulations and demonstrated the dynamic simulation of flexible bodies subject to non-penetration constraints. However, their flexible bodies are described in terms of global deformations of a rest shape and limited to specific geometric structures.

Also previously, a simplified particle system [11] and the depth field method [20] have been used to simulate the contact. Joukhadar et al. [19] demonstrated a fast contact localization method between deformable polyhedral. Hirota et al. [18] used penalty forces to prevent self-collision of FEM-based human model. Baraff [4] also implemented similar spring types of penalty force to prevent inter-penetration for cloth simulation. However, these approaches allow penetration upfront and measure the penetration depth to estimate the force to rectify the situation. These approaches are prone to the overshoot and oscillation problem if the stiffness coefficient is not correct, and the larger coefficients make the integration stepping suffer. However, an even more serious problem is that determining proper values for the stiffness of the spring is not trivial. Therefore, sliding contact between two deformable objects is very problematic while using a penalty-based method to maintain a tight error bound. Recent researches in cloth simulation using repulsions, penalty force and geometric fixes [5, 10, 15] also share similar problems.

This paper demonstrates a computational scheme for representing the geometry and physics of volumetric deformable objects and simulating their displacements when they are colliding and preserving accurate contact positions. The proposed technique focuses on determining contact regions of both objects and calculates accurate contact reaction forces at appropriate nodes and elements to maintain the resting contact relations. Our penetration-free deformation is based on the consolidation of colliding and resting contacts often used in rigid body simulation [3] and uses quadratic programming [21] to solve the non-penetration constraints.

2 Collision Detection and Contact Region Identification

Our deformation model is based on the standard nonlinear finite element method (FEM) analysis [6, 7] with Cauchy-Green deformation tensor and we have chosen to use a nearly incompressible Neo-Hookean material [9]. Since deformable object is discretized with a finite number of elements, the collision detection problem can be

interpreted as determining minimum distance between two non-convex boundary polyhedra. Collision detection has attracted considerable attention in geometric modeling and robotics [12]. A volumetric object is meshed with a set of tetrahedra and the surface is represented with a set of triangles. An entire object is approximated with a hierarchical tree of axis-aligned bounding boxes (AABB) to facilitate the quick collision rejection. Although an AABB quick reject test eliminates substantial computation time, actual costly intersection tests between geometric primitives at the leaves of the trees are inevitable. We are mainly interested in vertex-face collision and with an option to turn on and off edge-edge collision detection. This issue is detailed in the contact force computation section.

Once the penetrated objects and the intersections are found, the simulator must compute the exact collision time to include contact forces to prevent further penetration. Finding the exact collision time is time consuming [13], so often a numerical estimate is preferred within a certain tolerance. In addition, finding the exact collision time between deformable bodies using back-tracking and binary search is not practical because, unlike rigid objects which have relatively few nodes per object, soft objects contact may include numerous nodes that can penetrate the surfaces of other objects in a given time step. Thus potentially vast number of triangle-edge collisions can occur repeatedly during back-tracking and simulation can be slowed down substantially. Instead, when a penetration is detected between a node and a surface, the inter-penetrated nodes are moved back onto the surface. By doing so, we use a completely inelastic collision, similar to a zero restitution coefficient, from the colliding node's point of view. The artificial abrupt deformation is estimated and added to the node's force vector by interpolating internal forces of the colliding element based on the distance it is moved back. Then, the simulator sets up relative velocities of the node and surface contact points (computed with 3 nodes of the surface) to estimated velocities which are calculated by interpolation of relative velocities to ensure the legal condition of the collision for a non-penetration constraint. Changes in the internal force of the colliding node represent the compression and stored energy of the colliding elements and subsequently used for expansion and separation. This process simplifies the collision detection so that the actual contact force computation can be done more efficiently.

3 Contact Force Computation

Once the collisions are detected and the penetrated feature list is formed, the contact constraints are formulated and reaction forces are computed to prevent the penetration. Our collision and contact model is an extension from well-known rigid body polygonal objects contact scheme [3] to deal with multiple contact points within the contact region and the management of active constraint set. As fig. 1 illustrates, at a given time step, a point x on the contactor surface (Body I) can be either in contact with a point y on the target surface (Body J) or outside the surface. This can be verified by calculating the distance $g(x, y)$ from x to y . If $g(x,y) > 0$, there is no contact

between the two points and the normal contact force λ between these points is equal to zero. On the other hand, if $g(x,y) = 0$, the two points are in contact and the contact force has a non-zero value. These complimentary conditions can be shown as follows:

$$g(x,y) \geq 0, \quad \lambda \geq 0, \quad g\lambda = 0$$

To model the contact relation between deformable bodies, above conditions should be applied at each of the contact nodal points.

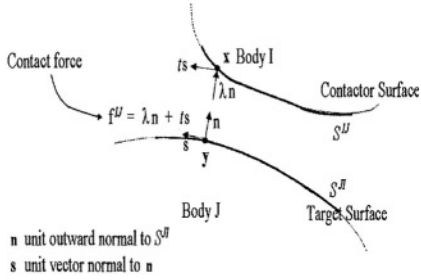


Fig. 1. Contact conditions

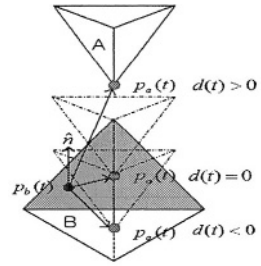


Fig. 2. Contact conditions between two tetrahedral

Our computation model for constraint-based contact analysis focuses on converting the contact cases into non-penetration constraints and formulates the contact forces to maintain the constraints at the contactor and target nodal points. Contact forces are applied exactly the same as external forces and generate deformation on both sides. Fig. 2 shows three possible contact scenarios, before, at contact, and after the penetration between two tetrahedra with a vertex-face contact. When the two bodies are disjoint at a given time t_0, the prospect contact points (closest point to the other object) on each side of object P_a and P_b are separated and the relative distance d is positive. However, when an actual penetration happens, the relative distance becomes negative as illustrated in fig. 2. $P_a(t_0) = P_b(t_0)$ denotes that two points are in contact at time t_0. The contact status of two points at time t_0 can be written in distance function in the normal direction as:

$$d_i(t_0) = \hat{n}_i(t_0) \cdot (P_a(t_0) - P_b(t_0))$$

where $\hat{n}_i$ is the outwards unit surface normal of a point i on the surface of object B. [0]Once the collision detection routine finds all colliding elements, non-penetration constraints are formulated to prevent further penetration in the next time step. Our simulator enforces this constraint by setting the relative velocity to zero and by maintaining the relative acceleration to be greater than or equal to zero. The relative velocity $\dot{d}_i(t)$,a time derivative of $d_i(t)$,can be represented as:

$$\dot{d}_i(t) = \dot{\hat{n}}_i(t) \cdot (P_a(t) - P_b(t)) + \hat{n}_i(t) \cdot (\dot{P}_a(t) - \dot{P}_b(t))$$

Starting from $d_i(t) \simeq 0$ and having $\dot{d}_i(t) = 0$ ensures that the two points are not moving toward penetration or separation at the current time step. Relative acceleration with respect to the nodal positional acceleration can be written as:

$$\ddot{d}_i(t) = \hat{n}_i(t) \cdot (\ddot{P}_a(t) - \ddot{P}_b(t)) + 2\dot{\hat{n}}_i(t) \cdot (\dot{P}_a(t) - \dot{P}_b(t))$$

To prevent inter-penetration, the condition $\ddot{d}_i(t) \geq 0$ must be maintained. Since $\ddot{d}_i(t)$ has nodal positional accelerations $\ddot{P}(t)$ term for both side of objects, and nodal accelerations can be rewritten with repulsive outward contact force $f_{ci}(t) \geq 0$ and mass.

Fig. 2 only illustrates the vertex-face case but edge-edge case is similar. Once the collision is detected, there are two contact points at each side of an edge and the normal direction can be computed with a cross product of two edges to represent the perpendicular direction. Computing the relative velocity and acceleration is similar to the vertex-face case. Since we're using a mesh structure to discretize a volume, the density of mesh determines the overall resolution and details of objects. If the average unit tetrahedron size is small enough, then the edge-edge case can be ignored since the edge-edge penetration can be detected by the vertex-face within a few times. However, if the size of average tetrahedron is relatively big, then the edge-edge penetration shows a significant visual artifacts and it must be removed. For deformable objects where each volumetric structure is meshed with fine triangles and tetrahedra, enforcing the edge-edge condition often causes additional computing load without substantial improvement in visual realism. In addition, sometimes enforcing edge-edge condition make the two objects remain in contact on an edge instead of a contact area, resulting local separation gaps within the contact area. Our system adopts an option to ignore the edge-edge cases when the mesh size is relatively small compare to the size of object volume.

4 Numerical Solution Method

To maintain the contact constraint, contact force must be positive to represent repulsive force toward the outside of the contact face $f_{ci}(t) \geq 0$ or the relative acceleration must be $\ddot{d}_i(t) \geq 0$ to guarantee the object separation. In addition, if any one of those conditions is met, then the other inequality must be zero. This last condition can be written as $f_{ci}\ddot{d}_i(t) = 0$. These complementary conditions can be arranged as a quadratic programming problem [21] with a general form as follows:

$$\begin{aligned} &\text{Minimize} \quad c^T x + \frac{1}{2} x^T Q x \\ &\text{Subject to} \quad Ax \geq b, \; x \geq 0 \end{aligned}$$

Since the function $f_{ci}\ddot{d}_i(t) = 0$ is monotonic, the minimization will have at least one solution and will converge to zero. Therefore, it can be used as an objective function and the $f_{ci}(t) \geq 0$ and $\ddot{d}_i(t) \geq 0$ conditions can be considered as inequality constraints in the QP system. If we rewrite the relative acceleration as a function of unknown contact forces as:

$$\ddot{d}_i(t) = a_{i1} f_{c1} + a_{i2} f_{c2} + \cdots + a_{in} f_{cn} + b_i$$

where a_{ij} is a n by n matrix for coefficients of unknown contact force, then we can get the linear and quadratic coefficients c^T and Q. Interior Point Method for QP often uses slack variables to make all inequality constraints into non-negatives. Non-negativity constraints are replaced with logarithmic barrier terms and added to the objective function. As the non-negativity constraints approach zero, the logarithmic barrier function becomes negative infinity, moving the objective function to the positive infinity. Therefore, the objective function will prevent them from becoming negative. The remaining linear constraints are all equalities and Lagrangian multipliers can be applied to get the local minima. The condition for the local minima is Karush Kuhn Tucker (KKT) condition. Incorporating the equality constraints into the objective function using Lagrange multipliers makes the system:

$$\text{Minimize } c^T x+\frac{1}{2}x^T Qx-\mu\sum_i \log x_i-\mu\sum_i \log w_i+\lambda^T(b-Ax+w)$$

where μ is a weight coefficient. If we take the derivative of the objective function and set it to zero for the local minima, we then have a resulting system of $2m + 2n$ linear equations with $2m + 2n$ unknowns, where m and n denote the number of nodes and constraints. Since this is sparse linear system, we can apply well-known sparse linear system solvers including conjugated gradient method or Cholesky factorization.

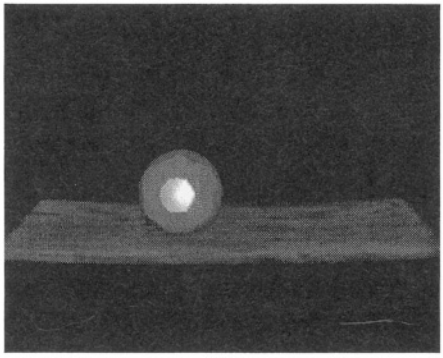


Fig. 3. Ball and board in contact

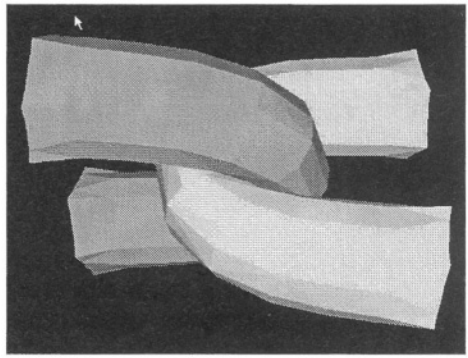


Fig. 4. Two deformable cylinders in contact

5 Experiments

We have described a FEM-based deformation simulator and implemented collision detection and contact resolution modules. Our simulator uses the 4^{th}-order Runge-Kutta ODE solver with adaptive step sizes. Fig. 3 shows a contact simulation between two deformable bodies, a stiff wood board and another stiff heavy metal ball. An animation [16] shows its accurate maintenance of the contact condition as well as realistic behaviors of the two stiff objects under contact. Since the models have relatively fine mesh structure, the simulation was performed and rendered off-line. Fig. 4 shows two deformable cylinders in contact. These cylinders are relatively coarsely

meshed and softer than the example shown in fig. 3. This example demonstrates that the contact and overall volumes of the objects are preserved even if they are under large deformation. An internal view from the animation shows the cylinders wrap around each other shows no penetrated nodes under the large deformation. The accuracy of our collision and contact model is quantitatively evaluated by measuring penetration and/or separation distance at contact regions.

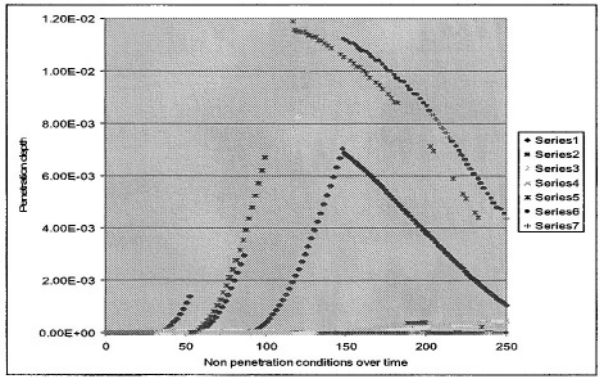


Fig. 5. Penetration depth

Fig. 5 shows an experimental penetration depth data from the simulation of two deformable bodies in contact shown in fig. 4. It runs in real-time and programmed to record all contacts error measures. It sampled 7 areas of contacts with the total of about 250 individual contact constraints enforced over time. The tolerance of penetration is set to 10E-2 and the average unit length of an element is 10, so it allows about 1/1000 of the size of a tetrahedral element for legal penetration. As shown in the graph, a majority of contact constraints are maintained at 10E-3 precision. Some contacts undergo penetrations up to the tolerance level of 10E-2 but nonetheless they are controlled within the user-defined error bound. Although the stability of simulation is heavily dependent to the material properties and ODE step sizes, our QP solver converges well within the pre-defined error bounds. For models with approximately 500 tetrahedral elements each side, our simulator performs at a visual interactive rate with an average of 25 frames per second on a Pentium 4 1.7GHz.

6 Conclusion and Future Works

This paper described a constraint-based collision and contact simulation between deformable bodies. The hierarchical collision detection and the initial conditioning process for non-penetration constraint enforcement simplifies the constraint formulation and accelerates the overall contact simulation. The nonlinear optimization based contact constraint enforcement demonstrates tight error bound at the contact region and numerical stability. The performance of the system is efficient enough to run medium to small scale FEM model in real-time. Our constraint-based contact elimi-

nates the needs for computing proper penalty force coefficients. Although some coarsely meshed objects can be simulated in real-time using a generic PC, deformation and contact between densely meshed structures still remains as a significant challenge. Proper parallelization of FEM deformation with respect to the collision detection, and contact force computation could be one of the natural extensions from this work. Adaptive re-meshing and simulation based on the amount of deformation and area of interest would also substantially increase the performance as well.

Acknowledgement. This research is partially supported by Colorado Advanced Software Institute (PO-P308432-0183-7) and NSF CAREER Award (ACI-0238521).

References

1. S. Cotin, H. Delingette, and N. Ayache. Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation, IEEE Tr. On Visualization and Computer Graphics, 1999.
2. D. Baraff and A. Witkin. Dynamic Simulation of Non-Penetrating Flexible Bodies, ACM Computer Graphics, Vol. 26, No. 2, 1992
3. A. Witkin and D. Barraf. Physically Based Modeling, SIGGRAPH 03' Course notes, 2003
4. D. Baraff and A. Witkin. Large Steps in Cloth Simulation, Proc. Computer Graphics, Annual Conference Series, ACM Press, 1998, pp. 43-54.
5. D. Baraff, M. Kass, and A. Witkin. Untangling Cloth, ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH 2003: Volume 22, Number 3, 862-870.
6. K. Bathe. Finite Element Procedures, Prentice Hall, Upper Saddle River, New Jersey 07458.
7. G. Beer, Finite Element, Boundary Element and Coupled Analysis of Unbounded Problems in Elastostatics, International Journal for Numerical Methods in Engineering, Vol 19. p567-580, 1980.
8. J. Berkley, S. Weghorst, H. Gladstone, G. Raugi, D. Berg, and M. Ganter. Fast Finite Element Modeling for Surgical Simulation, Proc. Medicine Meets Virtual Reality (MMVR'99), ISO Press, 1999, pp. 55-61.
9. J. Bonet, R. D. Wood, Nonlinear continuum mechanics for finite element analysis, Cambridge University press.
10. R. Bridson, R. Fedkiw, and J. Anderson, Robust treatment of collisions, contact and friction for cloth animation, Proc. SIGGRAPH 2002, ACM Press, Vol 21, pp. 594-603, 2002
11. M. Bro-Nielsen and S. Cotin. Real-Time Volumetric Deformable Models for Surgery Simulation Using Finite Elements and Condensation, Proc. Eurographics'96, Vol. 15, 1996.
12. D. Chen and D. Zeltzer. Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using Finite Element Method, Proc. SIGGRAPH 92, ACM Press, 1992.
13. M. Choi, James F. Cremer, Geometrically-Aware Interactive Object Manipulation, The Journal of Eurographics Computer Graphics Forum. Vol. 19, No. 1, 2000.
14. M. Hong, M. Choi, R. Yelluripati, Intuitive Control of Deformable Object Simulation using Geometric Constraints, Proc. The 2003 International Conference on Imaging Science, Systems, and Technology (CISST' 03), 2003
15. M. Desbrun, P. Schroder, Interactive Animation of Structured Deformable Objects, Graphics Interface, 1999

16. Computer Graphics Lab. University of Colorado at Denver, http://graphics.cudenver.edu/ICCS04.html
17. S. Gibson and B. Mirtich, A Survey of Deformable Modeling in Computer Graphics, Tech. Report No. TR-97-19, Mitsubishi Electric Research Lab., Cambridge, MA, Nov 1997
18. G. Hirota, S. Fisher, A. State, H. Fuchs, C. Lee, Simulation of Deforming Elastic Solids in Contact, Siggraph 2001 Conference Abstract and Applications
19. A. Joukhadar, A. Wabbi, and C. Laugier. Fast Contact Localization Between deformable Polyhedra in Motion, IEEE Computer Animation, June 1996.
20. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models, Int. J. Computer Vision, 1(4), 1987, pp. 321-332.
21. Y. Ye, Interior Point Algorithms: Theory and Analysis, Wiley-Interscience Series in Discrete Mathematics and Optimization John Wiley & Sons.

Prediction of Binding Sites in Protein-Nucleic Acid Complexes*

Namshik Han and Kyungsook Han**

School of Computer Science and Engineering, Inha University, Inchon 402-751, Korea
han_3567@hotmail.com, khan@inha.ac.kr

Abstract. Determining the binding sites in protein-nucleic acid complexes is essential to the complete understanding of protein-nucleic acid interactions and to the development of new drugs. We have developed a set of algorithms for analyzing protein-nucleic acid interactions and for predicting potential binding sites in protein-nucleic acid complexes. The algorithms were used to analyze the hydrogen-bonding interactions in protein-RNA and protein-DNA complexes. The analysis was done both at the atomic and residue level, and discovered several interesting interaction patterns and differences between the two types of nucleic acids. The interaction patterns were used for predicting potential binding sites in new protein-RNA complexes.

1 Introduction

A variety of problems concerned with protein-DNA interactions have been investigated for many years, but protein-RNA interactions have been much less studied despite their importance. One reason for this is that only a small number of protein-RNA structures were known. As a result, these structures were generally studied manually on a small-scale. The task of analyzing the protein-RNA binding structures manually becomes increasingly difficult as the complexity and number of protein-RNA binding structures increase. Now that an increasing number of protein-RNA structures are known, there is a need to automatically analyze the interactions involved and to compare them with protein-DNA interactions.

In contrast to the regular helical structure of DNA, RNA molecules form complex secondary and tertiary structures consisting of elements such as stems, loops, and pseudoknots. Generally only specific proteins recognize a given configuration of such structural elements in three-dimensional space. RNA forms hydrogen bonds and electrostatic interactions, and possess hydrophobic groups; it can therefore make specific contacts with small molecules. However, the basis of its interaction with proteins is unclear. In our previous study of protein-RNA complexes, we analyzed the interaction patterns between the protein and RNA at the level of residues and atoms [1]. As an extention of the previous study, we attempted to predict potential binding sites in protein-nucleic acid complexes by analyzing the hydrogen-bonding (H-

* This work was supported by the Ministry of Information and Communication of Korea under grant number 01-PJ11-PG9-01BT00B-0012.

** To whom correspondence should be addressed. email: khan@inha.ac.kr

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 309–316, 2004.
© Springer-Verlag Berlin Heidelberg 2004

bonding) interactions between the amino acids of proteins and the nucleotides of nucleic acids.

2 Types of Hydrogen Bonding Interactions

Hydrogen bonds were classified into 3 types: (1) ***single interactions*** in which one hydrogen bond is found between an amino acid and a nucleotide, (2) ***bidentate interactions*** where an amino acid forms two or more hydrogen bonds with a nucleotide or base-paired nucleotides, and (3) ***complex interactions*** where an amino acid binds to more than one base step simultaneously [1]. Our definition of hydrogen bond types is slightly different from that of Luscombe *et al.* [2]. The latter only analyzed hydrogen bonds between amino acids and bases, whereas we also consider hydrogen bonds with the RNA backbone. Therefore, our study can reveal differences in binding propensities between bases, sugar groups and phosphate groups.

3 Frameworks

3.1 Datasets

Protein-RNA complex structures were obtained from the PDB database [3]. Complexes solved by X-ray crystallography at a resolution ≤ 3.0Å were selected. As of September 2002, there were 188 protein-RNA complexes in PDB, and 139 of them were at a resolution ≤ 3.0 Å. We used PSI-BLAST [4] for similarity searches on each of the protein and RNA sequences in these 139 complexes in order to eliminate equivalent amino acids or nucleotides in homologous protein or RNA structures. 64 out of 139 protein-RNA complexes were left as the representative, non-homologous complexes after running the PSI-BLAST program with an E value of 0.001 and an identity value of 80% or below. We excluded 13 out of the 64 complexes that have no water molecules or are composed of artificial nucleotides. Table 1 lists the 51 protein-RNA complexes in the final data set. For the dataset of protein-DNA complexes, we used 129 protein-DNA complexes used in the study of Luscombe [2].

Table 1. Protein-RNA complexes in the data set.

Molecular	Num.	PDB ID
tRNA	15	1EFW, 1SER, 1C0A, 1QTQ, 1H4Q, 1H4S, 1FFY, 1GAX, 1B23, 1F7U, 1G59, 1IL2, 1QF6, 2FMT, 1K8W
mRNA	1	1B7F
Ribosome	6	1HC8, 1MMS, 1DFU, 1FEU, 1DK1, 1I6U
Ribozyme	4	1JBR, 1JBS, 1CX0, 1B2M
Virus	12	2BBV, 1F8V, 1KNZ, 1E7X, 1HE0, 1HE6, 1HDW, 1ZDH, 1ZDI, 5MSF, 6MSF, 7MSF
TRAP	3	1C9S, 1GTF, 1GTN
SRP	4	1JID, 1L9A, 1LNG, 1HQ1
Others	6	1FXL, 1G2E, 1DI2, 1EC6, 1KQ2, 1URN

3.2 Hydrogen Bonds

The number of hydrogen bonds between the amino acids and nucleotides in the protein-RNA complexes was calculated using CLEAN, a program for tidying Brookhaven files, and HBPLUS [5], a program to calculate the number of hydrogen bonds. The hydrogen bonds were identified by finding all proximal atom pairs between hydrogen bond donors (D) and acceptors (A) that satisfy the given geometric criteria. The positions of the hydrogen atoms (H) were theoretically inferred from the surrounding atoms, because hydrogen atoms are invisible in purely X-ray-derived structures. The criteria considered to form the hydrogen bonds for this study were: contacts with a maximum D-A distance of 3.9 Å, maximum H-A distance of 2.5 Å, and minimum D-H-A and H-A-AA angles set to 90°, where AA is an acceptor antecedent (see Fig. 1).

All protein-RNA bonds were extracted from the HBPLUS output files. There were 1,568 hydrogen bonds in the dataset. We conducted separate experiments in order to compare the properties of single interactions, bidentate interactions and complex interactions, and the results were analyzed for the three types of hydrogen bonds: (1) single interactions, (2) bidentate interactions, and (3) complex interactions.

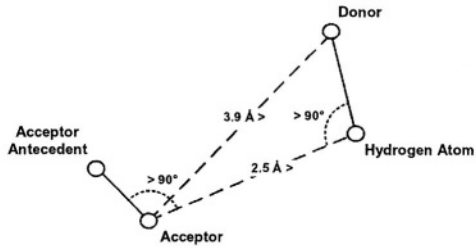


Fig. 1. Angles and distances used in the definition of the hydrogen bonds.

4 Algorithms

4.1 Analysis Algorithm

As shown in Fig. 2, the analysis algorithm is composed of 4 phases. In phase 1, the algorithm constructs arrays to store the amino acid and nucleic acid sequences, respectively, and classifies hydrogen bonds. These arrays and lists are used to determine interaction types. The algorithm also analyzes whether a nucleotide is paired with other nucleotides. It is essential to discriminate whether binary or multi bond is single interaction or not. So, it is used to classify the interaction types. These processes are the basis of phases 2–4 of the algorithm.

In phase 2, the algorithm investigates the internal hydrogen bond relations of the nucleic acid and records the result of the investigation in a linked-list. It also investigates the hydrogen bonds between the protein and nucleic acid and records this result in a linked-list. These processes are important groundwork for identifying

binding patterns as they represent the relation between pairs of residues in the form of linked-lists. These are then used in phase 4 to parse the classified interaction types.

In phase 3, the algorithm classifies the bonding type of each amino acid into unitary, double and multi-bond based on the number of hydrogen bonds between the amino acid and the nucleic acid. It inspects whether the amino acid forms two or more hydrogen bonds with the base or base pair. This is one of the most important processes because it can directly identify the double bond of the bidentate interaction. Since double bonds are abundant, it can eliminate many unnecessary operations.

In phase 4, the algorithm parses the outcomes of phase 3 to determine binding patterns and numbers of hydrogen bonds involving each region of nucleotides and amino acids. The analysis is done both at the atomic and residue level, and the results help us identify how proteins recognize binding targets, which nucleotides are favored by which amino acids, and their binding sites.

4.2 Prediction Algorithm

The prediction algorithm is composed of two phases. In phase 1, it splits unknown protein structure into dices and examines all dices to sort potential binding sites with high probability. Splitting the protein structure requires the coordinate values of all atoms and the center position of every residue. Every PDB file of a structure has the starting coordinate value, which is outside the structure. The algorithm selects the closest residue from the starting coordinates of the structure. It then finds neighbor residues of the closest residue and the residues within a dice.

In phase 2, the algorithm constructs the structure-based residue lists that contain structural information for each dice. It then compares the lists to the nucleic acid sequence to predict potential binding sites using the interaction propensities and patterns. All potential binding sites are examined to predict the best binding site candidate. The structure information was used to eliminate spurious candidates at the last step of prediction. For example, a potential binding site with interaction between sheets in proteins and stems in RNA or DNA is eliminated, since the sheet structure in proteins prefers the loop structure in RNA or DNA. More details are explained in section 5.3.

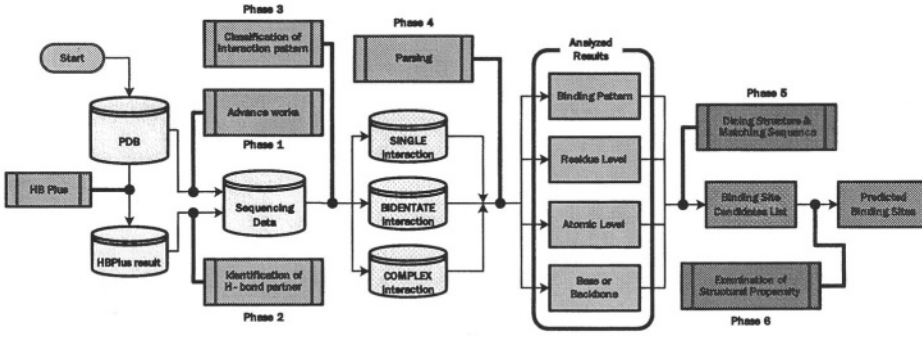


Fig. 2. Sequence for analyzing the protein-nucleic acid complexes and for predicting potential binding sites.

5 Results

5.1 Differences between DNA and RNA

In protein-DNA complexes, almost equal numbers of hydrogen bonds were involved in single, bidentate and complex interactions [2]. However, in protein-RNA complexes, 60% of the hydrogen bonds were found in single interactions.

RNA and DNA were different in their preference for backbone versus base contacts. 32% of the hydrogen bonds between protein and DNA involved base contacts, compared with 50% in protein-RNA. This can be explained by the structural difference between RNA and DNA. DNA is a double stranded molecule, and its bases are therefore already involved in hydrogen bonding. Hence, the base region is not as flexible as the backbone and is therefore less able to bind to amino acids. The bases in single-stranded regions of RNA, on the other hand, are quite flexible.

DNA and RNA were also different in their favored amino acids. GLU and ASP have acidic side chain groups, and more frequently hydrogen bind to RNA than to DNA. In protein-RNA complexes, these two amino acids are ranked 5th and 7th, respectively, but in protein-DNA complexes they are ranked 11th and 12th (Table 2). In particular, both GLU and ASP bind very frequently to guanine in the protein-RNA complexes (Table 3). The opposite is the case with GLY and ALA, which bind to DNA more often than to RNA. They rank 10th and 14th, respectively in protein-RNA complexes, but 5th and 9th in protein-DNA complexes. Both GLY and ALA have non-polar side chains, and residues with small side chains bind to double stranded DNA more easily than those with large side chains.

Table 2. Comparison of protein-DNA complexes with protein-RNA complexes in terms of the number of hydrogen bonds in amino acids.

Rank	DNA		RNA	
	Residue	H - bonds	Residue	H - Bonds
1	ARG	597	ARG	306
2	LYS	293	LYS	257
3	THR	292	SER	164
4	SER	207	THR	151
5	**GLY**	**168**	**GLU**	**136**
6	ASN	167	ASN	125
7	GLN	149	**ASP**	**116**
8	TYR	80	GLN	61
9	**ALA**	**71**	TYR	59
10	HIS	60	**GLY**	**40**
11	GLU	**53**	HIS	36
12	**ASP**	**19**	PHE	31
13	ILE	16	LEU	19
14	CYS	11	**ALA**	**17**
...	...	...	...	...
20	VAL	3	MET	1

5.2 Interaction Propensities and Patterns in Protein-RNA Complexes

In bidentate interactions, GLU and ASP mainly bind to guanine whereas THR and LYS generally bind to adenine. This binding preference results in characteristic patterns of binding between the amino acid and nucleotide pairs (Table 3 and Figure 3). For example, the binding pattern shown in the GLU–G pair is most common. An exception is LYS: there are 69 hydrogen bonds between LYS and adenine bases, but there is no prominent binding pattern.

In protein-RNA complexes, the side chain of an amino acid binds to the only one base rather than base pairs or base steps. In contrast, there are many hydrogen bonds between a side chain and a base pair or base step in protein-DNA complexes [3]. This difference can again be explained by the structural difference between RNA and DNA.

Table 3. Binding patterns involved in bidentate interactions. The hydrogen bonds donor (D) and hydrogen bond acceptor (A) are indicated in parentheses.

ARG						ASP		ASN		GLU		GLN						SER		THR	
NH1 (D)	NH2 (D)	NE (D)	NH2 (D)	NH1 (D)	NH2 (D)	OD1 (A)	OD1 (A)	OD1 (A)	ND2 (D)	OE1 (A)	OE2 (A)	OE1 (A)	NE2 (D)	OE1 (A)	NE2 (D)	OE1 (A)	NE2 (D)	OG (A)	OG (D)	OG1 (D)	OG1 (A)
C		U		G		G		A		G		U		G		G		A		A	
N3 (A)	O2 (A)	O2 (A)	O2 (A)	O6 (A)	O6 (A)	N1 (D)	N2 (D)	N6 (D)	N1 (A)	N1 (D)	N2 (D)	N3 (D)	O4 (A)	N2 (D)	O2 (A)	N2 (D)	N3 (A)	N6 (D)	N1 (A)	N7 (A)	N6 (D)
11		2		1		12		1		37		2		1		1		1		18	

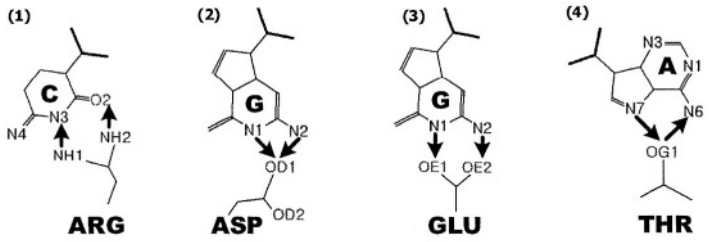


Fig. 3. Frequent binding patterns. Binding patterns (1), (2), (3), and (4) were observed in 11, 12, 37 and 18 complexes in the dataset, respectively.

5.3 Structural Propensities and Binding Sites

Protein helices bind equally to nucleotide pairs and non-pairs in H-bonding interactions. In contrast, sheets prefer non-pairs to pairs, and turns prefer pairs to non-pairs. Non-pairs have been considered to have high interaction propensity in general, but our study found this is not the case since turns prefer pairs and helices show no preference. In protein-RNA complexes, this implies that sheets prefer to bind to RNA loops and turns prefer to bind to RNA stems [6].

Fig. 4 shows both the known binding sites and the predicted binding sites of the NS5B part of Hepatitis C Virus (HCV) [7], *Thermus thermophilus* Valyl-tRNA

synthetase [8] and *Escherichia coli* Threonyl-tRNA synthetase [9]. Table 4 represents both known and predicted binding sites of the NS5B part of Hepatitis C Virus. The predicted binding sites do not exactly correspond to the known binding sites. However, all predicted binding sites are found near or within the known binding sites, and therefore can reduce the region of potential binding sites effectively.

Table 4. Sequences of the known and predicted binding sites of the NS5B part of Hepatitis C Virus. Residues in red color represent those common in the known and predicted binding sites. Residues in blue represent those in the known sites only.

| | | |

6 Discussion

We have developed a set of algorithms for analyzing H-bonding interactions and nucleic acids and for predicting potential binding sites between amino acids. This paper presents the results of such an analysis and compares the characteristics of RNA and DNA binding to proteins and prediction results.

The protein-RNA complexes display specific binding patterns. In bidentate interactions in protein-RNA complexes, GLU and ASP overwhelmingly bind to guanine while THR and LYS generally bind to adenine. DNA binds to GLY and ALA preferentially, whereas RNA usually does not binds to them but rather to GLU and ASP. This binding preference results in favored binding patterns. For example, the binding pattern of the GLU–G pair is the most common.

The binding patterns obtained from analyzing H-bonding interactions between amino acids and nucleotides were used to predict potential binding sites of protein-nucleic acid complexes. The binding sites predicted by our algorithm do not exactly correspond to the known binding sites, but it can reduce the region of potential binding sites and the unnecessary experiments. This indicates that prediction was performed in a conservative manner. However, a more rigorous study is required to improve the prediction results for various test cases.

References

1. Han, N., Kim, H., Han, K.,: Computational Approach to Structural Analysis of Protein-RNA Complexes. Lecture Notes in Computer Science, Vol. 2659 (2003) 140-150
2. Luscombe, N.M., Laskowski, R.A., Thornton, J.M.: Amino acid–base interactions: a three-dimensional analysis of protein–DNA interactions at an atomic level. Nucleic Acids Research 29 (2001) 2860-2874
3. Westbrook, J., Feng, Z., Chen, L., Yang, H., Berman, H.M.: The Protein Data Bank and structural genomics. Nucleic Acids Research 31 (2003) 489-491
4. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research 25 (1997) 3389-3402
5. McDonald, I.K., Thornton, J.M.: Satisfying Hydrogen Bonding Potential in Proteins. J. Mol.Biol. 238 (1994) 777-793
6. Kim, H., Jeong, E., Lee, S.-W., Han, K.: Computational analysis of hydrogen bonds in protein-RNA complexes for interaction patterns. FEBS Letters 552 (2003) 231-239
7. Bressanelli, S., Tomei, L., Roussel, A., Incitti, I., Vitale, R.L., Mathieu, M., De Francesco, R., Rey, F.A.: Crystal structure of the RNA-dependent RNA polymerase of hepatitis C virus. Proc. Natl. Acad. Sci. 96 (1999) 13034-13039
8. Fukai, S., Nureki, O., Sekine, S., Shimada, A., Tao, J., Vassylyev, D.G., Yokoyama, S.: Structural Basis for Double-Sieve Discrimination of L-Valine from L-Isoleucine and L-Threonine by the Complex of tRNAVal and Valyl-tRNA Synthetase. Cell 103 (2000) 793-803
9. Sankaranarayanan, R., Dock-Bregeon, A.-C., Romby, P., Caillet, J., Springer, M., Rees, B., Ehresmann, C., Ehresmann, B., Moras, D.: The Structure of Threonyl-tRNA Synthetase-tRNAThr Complex Enlightens Its Repressor Activity and Reveals an Essential Zinc Ion in the Active Site. Cell 97 (1999) 371-381

Prediction of Protein Functions Using Protein Interaction Data

Haemoon Jung and Kyungsook Han*

School of Computer Science and Engineering, Inha University, Inchon 402-751, Korea
idjhmoon@hotmail.com, khan@inha.ac.kr

Abstract. Information on protein-protein interactions provides valuable insight into the underlying mechanism of protein functions. However, the accuracy of protein-protein interaction data obtained by high-throughput experimental methods is low, and thus requires a rigorous assessment of their reliability. This paper proposes a computational method for predicting the unknown function of a protein interacting with a protein with known function, and presents the experimental results of applying the method to the protein-protein interactions in yeast and human. This method can also be used to assess the reliability of the protein-protein interaction data.

1 Introduction

High-throughput experimental techniques enable the study of protein-protein interactions at the proteome scale through systematic identification of physical interactions among all proteins in an organism [1]. The increasing volume of protein-protein interaction data is becoming the foundation for new biological discoveries. Protein-protein interactions play important roles in nearly all events that take place in a cell [1]. Particularly valuable will be analyses of proteins that play pivotal roles in biological phenomena in which the physiological interactions of many proteins are involved in the construction of biological pathways, such as metabolic and signal transduction pathways [2].

Here is a more elaborate example. All biochemical process is regulated through protein function. Therefore, diseases (hereditary/non-hereditary) are made manifest as a result of protein function. Also, finding protein function is necessary for the normal progression of medicinal development. Thus, with the completion of genome sequencing of several organisms, the functional annotation of the proteins is of most importance [3]. This function is considered as property of sequence or structure.

Several research groups have developed methods for functional annotation. The classical way is to find homologies between a protein and other proteins in databases using programs such as FASTA and PSI-BLAST, and then predict functions [3]. Another approach is called the *Rosetta stone method,* where two proteins are inferred to interact if they are together in another genome [3]. We propose a method for determining the reliability of the protein-protein interaction data obtained by high-

* To whom correspondence should be addressed. Email: khan@inha.ac.kr

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 317–324, 2004.
© Springer-Verlag Berlin Heidelberg 2004

throughput experimental methods. This method can also be used for predicting the unknown function of a protein interacting with a protein with known function.

2 Analysis of Yeast Protein Data

2.1 Data Sets

The data of yeast protein interactions were obtained from the Comprehensive Yeast Genome Database (CYGD) at MIPS [7]. After removing 1,150 redundant data, 9,490 non-redundant interactions were left. In addition to the interaction data, the information on the yeast protein catalogue such as class, enzyme code number, motif, function, complex and cell localization were extracted. Table 1 shows the number of yeast protein entries in each catalogue.

Table 1. Yeast protein catalogues

Protein catalogues	Number of protein entries
Classes	1,070
Complexes	1,321
EC number	1,152
Motifs	2,539
Phenotype	3,120
Function	11,506
Cell localization	2,407

The function catalogue in particular is important because the primary focus of this study is the prediction of protein functions. We shall now examine the function catalogue more closely.

Functions of yeast proteins were all arranged from FunCat (The Functional Catalogue) of MIPS. The FunCat is an annotation scheme for the functional description of proteins from prokaryotes, unicellular eukaryotes, plants and animals [7]. Taking into account the broad and highly diverse spectrum of known protein functions, the FunCat consists of 30 main functional categories (or branches) that cover general fields like cellular transport, metabolism and signal transduction [7]. The main branches exhibit a hierarchical, tree like structure with up to six levels of increasing specificity [7]. In total, the FunCat has 1,445 functional categories. Among of them, 215 functional categories were applied to yeast. Tables 2 and 3 show the number of yeast protein entries and yeast proteins, respectively, in each functional category of FunCat.

Table 2. The number of yeast proteins in each function category

Function category	Number of yeast proteins
Unclassified function	2,399
uncleus	764
cytoplasm	554
mitochondrion	366
transcriptional control	334
mitotic cell cycle and cell cycle control	326
C-compound and carbohydrate utilization	260
ribosome biogenesis	215
stress response	175
budding, cell polarity and filament formation	170
protein modification	167
pheromone response, mating-type determination, sex-specific proteins	160
endoplasmic reticulum	157
protein targeting, sorting and translocation	146
plasma membrane	145
vesicular transport (Golgi network, etc.)	129
regulation of C-compound and carbohydrate utilization	120
...	...
plasma membrane	1
tRNA transcription	1

2.2 Analysis Results

The protein interaction data in each catalogue of Table 1 were analyzed. As shown in Fig. 1, the proteins in a same complex interact with the highest probability (0.33), and the proteins with a same function interact with the second highest probability (0.28). The proteins in a same complex interact *and* with a same function interact with the probability of 0.39, which is higher than 0.33 and 0.28 but not much higher. This implies that a large portion of the proteins in the two catalogues overlap.

The ratio of the number of interactions between proteins in a same complex to the total number of interactions is 0.25. The ratio of the number interactions between proteins with a same function to the number of interactions between proteins in a same complex is 0.97. This indicates that interacting proteins in a same complex also have a same function with a high probability. The inverse is not necessarily true. Consequently, we discovered an association rule: If proteins A and B are in a same complex and interact each other, proteins A and B have a same function.

Table 3. Scheme of yeast function catalogue

Function category	Protein entry
	YLR410w
modification by acetylation,	YOR023c
deacetylation	YOR064c
	YPR131c
modification with sugar residues(e.g.	YNL048w
glycosylation)	YJL183w
	YDL230w
	YDR099w
unspecified signal transduction	YDR315c
	YER177w
	YOR206w
	YOR257w
	YMR131c
other control of cellular organization	YHR102w
	YDR060w
	YDR260c
	YGL029w
	YCR083w
other detoxification	YOR163w
other transposable elements, viral and	YDR289c
plasmid proteins	YGR275w
	YJL047c
	YDL235c
protein binding	YLR006c
	YIL147c
	YBR294w
homeostasis of sulfates	YPR003c
	YLR092w
vacuolar degradation	YGL227w
	YAL035w
initiation	YKL204w
mRNA processing	YNL317w
3'-end processing	YKL208w
	YBR059c
	YOR090c
protein modification by	YLL010c
phosphorylation, dephosphorylation	YIL113w
	YLR019w
	YDR499w

(Other categofies are not shown here)

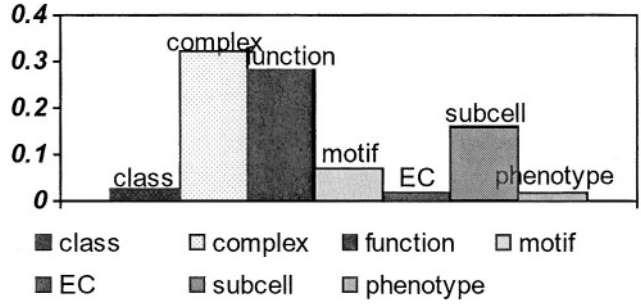


Fig. 1. The probability of interaction between proteins in each catalogue.

This rule is also confirmed in the protein-protein interaction networks of Fig. 2. Fig. 2A shows a network of interactions between proteins in a same complex, visualized by InterViewer [4, 5]. Interacting proteins with a same function are selected from this network and shown in green. Most nodes of Fig. 2A are selected as proteins involved with the interactions between proteins with a same function. Figure 2B shows a network of interactions between proteins with a same function. Interacting proteins in a same complex are selected from this network and shown in green, too. Only a small portion of the entire nodes is selected.

3 Prediction of Human Protein Function

Uetz *et al.* [8] and Ito *et al.* [9] show that the function of uncharacterized proteins can be predicted in the light of the interacting partner by using the principle of 'guilt by association', which maintains that two interacting proteins are likely to participate in the same cellular function [2, 6].

Since the function of many human proteins is unknown, we predicted the function of human proteins. Suppose that $\mathbf{SP}_j$ is a protein with unknown function, $\mathbf{P}_i$ is a protein interacting with $\mathbf{SP}_j$, AF is the total number of functions of interacting proteins, AP is the total number of interacting proteins, and that $\mathbf{DP}_i$ is the degree of a particular partner protein. Then the score function for assessing the probability that protein $\mathbf{SP}_j$ has the same function with protein $\mathbf{P}_i$ can be computed by equation 1.

$$Score(j,i) = DP_i \times \frac{1}{AF} \times \frac{1}{AP} \times CP_i \tag{1}$$

where $\mathbf{CP}_i$ is a complex constant. Algorithms 1-3 describe how to predict the function of a protein. In the algorithms, protein_num is the number of gathered proteins, and cnt represents the number of interacting partners with a same function. All interaction data are cleaned first by Algorithm 2, and the function of interacting partners is counted in Algorithm 3.

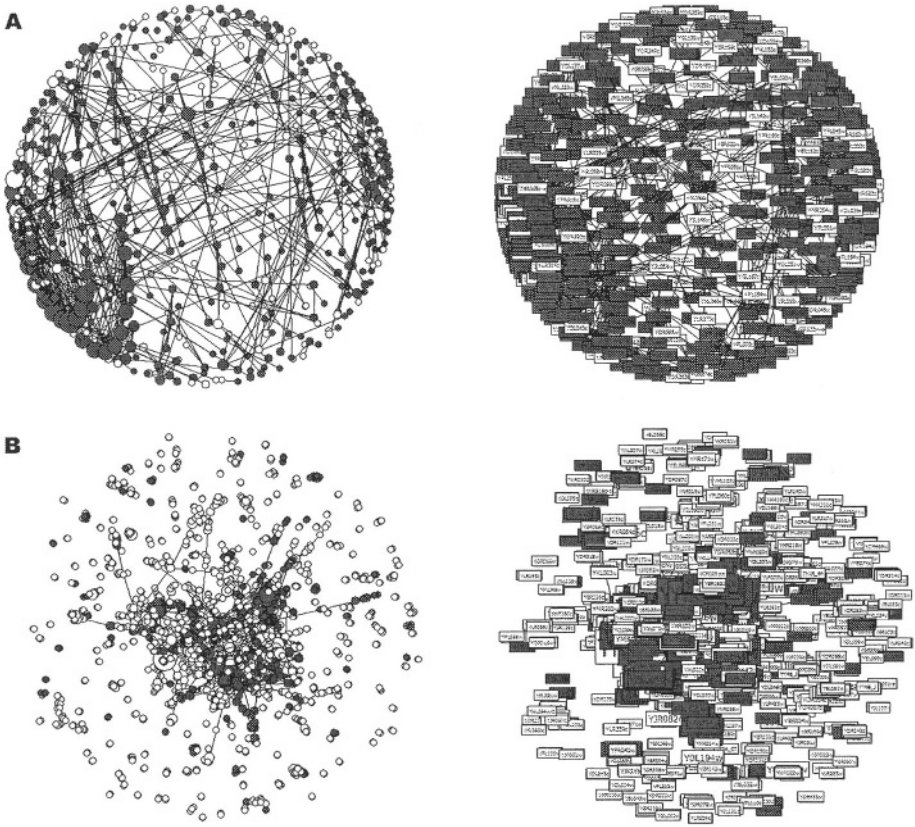


Fig. 2. (A) Left: A network of interactions between proteins in a same complex. Interacting proteins with a same function are shown in green. The network in the right shows the same network in the left but with node labels shown. (B) Left: A network of interactions between proteins with a same function. Interacting proteins in a same complex are shown in green. The network in the right shows the same network in the left but with node labels shown.

Algorithm 1 CreateNode ()

```
Extract structureID, structure from structure table
FOR k=0 protein_num-1 DO
    WRITE k, proteinID, structure IN node table
ENDFOR
```

Algorithm 2 ClearInteract ()

```
Extract source, target from interactions
FOR k=0 protein_num-1 DO
  IF (source==k) THEN
    WRITE source, target IN non-redundant_interaction
  ELSEIF (target==k) THEN
```

```
    Temp=target
    Target=source
    Source=temp
    WRITE target, source IN the non-redundant_interaction
  ENDIF
ENDFOR
```

Algorithm 3 CountPartner ()

```
Extract source, target from the non_redundant interaction table
Extract proteinID, structure from the node table
Cnt=0
FOR i=0 non_redundant interaction's row-1 DO
  IF (target==node.proteinID)
    WRITE source, target, node.structure IN temp
  ENDIF
ENDFOR
FOR j=0 temp's row-1 DO
  IF (temp.proteinID==j)
    compare_structure=temp.structure
   FOR k=0 temp's row-1 DO
    IF(compare_structure==temp.structure)
     cnt=cnt+1
    ENDIF
   ENDFOR
```

Table 4 summarizes the prediction results. The probability of interaction between proteins in each category is different from that in yeast proteins.

Table 4. The probability of interaction between human proteins

Items	Value
Ratio of the interactions between proteins in a same complex to the total interactions	0.26
Ration of the interactions between proteins with a same function to the total interactions	0.34
Ratio of interactions between proteins with a same function to the interactions between proteins in a same complex	0.3547
Ratio of interactions between proteins in a same complex to the interactions between proteins with a same function	0.53

4 Conclusion

Proteins have many different biological functions by interacting with other proteins, and two interacting proteins are likely to have same biological function. Therefore, if

a protein with known function is identified to interact with an uncharacterized protein, the function of the uncharacterized protein can be predicted. From the analysis of the experimental data of yeast protein interactions, we discovered a reliable association rule that "if proteins A and B exist in a same complex and interact each other, proteins A and B have a same function." We have developed an algorithm for predicting the function of proteins based on the association rule, and applied the algorithm to the human protein interaction data. Experimental results show that the algorithm is potentially useful for predicting the function of uncharacterized proteins.

Acknowledgements. This work was supported by the Ministry of Information and Communication of Korea under grant IMT2000-C3-4.

References

1. Chen, Y., Xu, D.: Computational analyses of high-throughput protein-protein interaction data. Current protein and peptide science 4 (2003) 159-180
2. Saito, R., Suzuki, H., Hayashizaki, Y.: Interaction generality, a measurement to assess the reliability of a protein-protein interaction. Nucleic Acids Research 30 (2002) 1163-1168
3. Deng, M., Zhang, K., Mehta, S., Chen, T., Sun, F.: Prediction of Protein Function Using Protein-Protein Interaction Data. IEEE Computer Society Bioinformatics Conference (2002) 197-206
4. Ju, B., Park, B., Park, J., Han, K.: Visualization and analysis of protein interactions. Bioinformatics 19 (2003) 317-318
5. Han, K., Ju, B.: A fast layout algorithm for protein interaction networks. Bioinformatics 19 (2003) 1882-1888
6. Oliver, S.: Guilt-by-association goes global. Nature 403 (2000) 601–603.
7. CYGD Home Page http://mips.gsf.de/genre/proj/yeast/index.jsp
8. Uetz, P., Giot, L., Cagney, G., Mansfield, T.A., Judson, R.S., Knight, J.R., Lockshon, D., Narayan, V., Srinivasan, M., Pochart, P. et al.: A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae. Nature (2000) 403 623–627.
9. Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., Sakaki, Y.: A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proc Natl Acad Sci USA (2001) 98 4569–4574

Interactions of Magainin-2 Amide with Membrane Lipids

Krzysztof Murzyn, Tomasz Róg, and Marta Pasenkiewicz-Gierula

Molecular Modelling Group, Faculty of Biotechnology, Jagiellonian University, ul. Gronostajowa 7, Kraków, Poland
{murzyn, tomekr, mpg}@mol.uj.edu.pl

Abstract. Magainin-2 is a natural peptide that kills bacteria at concentrations that are harmless to animal cells. Molecular modelling methods were applied to investigate basic mechanisms of magainin-2 amide (M2a) membrane selectivity. Three computer models of a lipid matrix of either animal or bacterial membrane containing M2a were constructed and simulated. Specific interactions between membrane lipids and M2a peptides, responsible for M2a selectivity are elucidated.

1 Introduction

Magainin-2 (M2, GIGKFLHSAKKFGKAFVGEIMNS) is a natural, 23-amino acid cationic peptide expressed in the skin of an African frog *Xenopus leavis*. M2 selectively kills bacteria at concentrations that are harmless to animal cells. In organic solvents and in the vicinity of a lipid bilayer, M2 forms an -helix of a distinct hydrophobic moment [1]. Such a helix possesses a polar and non-polar face. At the physiological conditions, the total electrostatic charge of M2 amide (M2a), which was used in this study, is +4 e. It results from the positively charged N-terminus, four positively charged Lys and one negatively charged Glu residues. By physical interactions with the membrane lipids, M2a disturbs the lamellar structure of the lipid matrix of biomembranes. The extent of this disturbance depends on the lipid composition of the membrane, particularly on the content of anionic lipids, as well as the peptide-to-lipid ratio.

The lipid matrix of the animal plasma membrane consists mainly of neutral phosphatidylcholine (PC) and cholesterol (Chol) molecules. In contrast, the main lipid components of the bacterial membrane are neutral phosphatidylethanolamine (PE) and negatively charged phosphatidylglycerol (PG) molecules. Due to a positive charge, the effect of M2a molecules on the bacterial membrane is stronger than on the animal membrane. In the initial stage of interaction with the membrane, M2a molecules locate on the outer leaflet of the cell membrane. In the animal membrane, interactions between M2a, PC, and Chol are such that push peptide molecules away from the surface. In the bacterial membrane, when the peptide-to-lipid (P/L) ratio is above 1:40 [2], M2a together with PG form large openings in the membrane [2, 3]. The openings (toroidal pores) consist of 4 - 7 M2a and several PG molecules.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 325–331, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In this study, molecular modelling methods were applied to investigate basic mechanisms of M2a membrane selectivity. This selectivity results from specific interactions between membrane lipids and M2a peptides. Three computer models of membrane systems were constructed and simulated. The first and second systems constituted, respectively, model animal and bacterial membranes each containing two M2a molecules located horizontally on the membrane surface (carpet models, EUCARPET and PROCARPET, respectively). The third system consisted of a model bacterial membrane containing five M2a molecules that together with PG molecules formed a toroidal pore in the membrane centre (PORE). Molecular dynamics (MD) simulations of the carpet models were carried out for 12 ns and of PORE for 5 ns.

2 Methods

2.1 Simulation Systems

As a model of the animal membrane, a 1-palmitoyl-2-oleoyl-phosphatidylcholine (POPC) bilayer containing ~23 mol% cholesterol (Cho1) was used. Both POPC and Cho1 are major constituents of the animal plasma membrane. Details concerning the POPC-Chol bilayer construction are given in [4]. The EUCARPET system contained 68 POPC, twenty Cho1, two M2a, eight Cl^-, and 2533 water molecules (12647 atoms in total). Chlorine ions were added to neutralise +8 e charge on two M2a molecules. As a model of the bacterial membrane, a bilayer made of 1-palmitoyl-2-oleoyl-phosphatidylethanolamine (POPE) and 1-palmitoyl-2-oleoyl-phosphatidylglycerol (POPG) in ratio 3:1 was used. Such a lipid composition is typical for the inner bacterial membrane. The PROCARPET system contained 66 POPE, twenty two POPG, two M2a, fourteen Na^+, and 2614 water molecules (12414 atoms in total). Fourteen sodium ions together with two M2a molecules neutralised –22 e charge on POPG molecules. Details concerning the POPE-POPG bilayer construction are given in [5]. The PORE system contained 138 POPE, 46 POPG (25 mol% POPG), five M2a, twenty six Na^+, and 5909 water molecules (28457 atoms in total). The toroidal pore was built according to the model proposed by Ludtke et al. [2] of five M2a and twenty POPG molecules (Fig. 1). In the pore, lipids interpose magainin helices oriented perpendicular to the membrane surface such that the polar faces of the amphiphilic helices and the polar heads of the lipids constitute the pore lining. As a result, both membrane leaflets form a continuous surface, which allows for free diffusion of lipids between the outer and inner membrane layers. Details concerning the construction of PORE are given in [6].

2.2 Simulation Parameters

For POPC, POPE, POPG, M2a, and Chol, optimised potentials for liquid simulations (OPLS) [7], for water, TIP3P [8], and for sodium and chlorine ions, Aqvist's parameters were used. Phospholipid, peptide, Na^+, and Cl^- molecules were treated as the solute molecules and water was the solvent. The united-atom approximation was applied to the CH, CH_2, and CH_3 groups of the peptide and lipid molecules. All polar groups of the solute and solvent molecules were treated in full atomic detail. The nu-

merical values for the atomic charges of POPE and POPG are given in [5] and of POPC follow those in Charifson et al. [9]. To retain the double bond in the oleoyl chain of phospholipids in the *cis* conformation, restraints on the double bond dihedral were imposed with a force constant of $100\,kcal{\cdot}mol^{-1}{\cdot}rad^{-2}$. Restraints acted whenever the deviation from an ideal conformation of 0° exceeded ±30°. Procedures for supplementing the original OPLS base with the missing parameters for the lipid headgroup were described by Pasenkiewicz-Gierula et al. [10], and those for the β-chain sp2 carbon atoms by Murzyn et al. [11]. Chiral centres in POPE, POPG, and POPC molecules were kept in a chosen configuration by defining relevant improper torsions. Improper torsions were parameterised in the OPLS forcefield with the half of the potential barrier equal to $14.0\,kcal{\cdot}mol^{-1}{\cdot}rad^{-2}$, periodicity of 3 and the energy maximum at 0°.

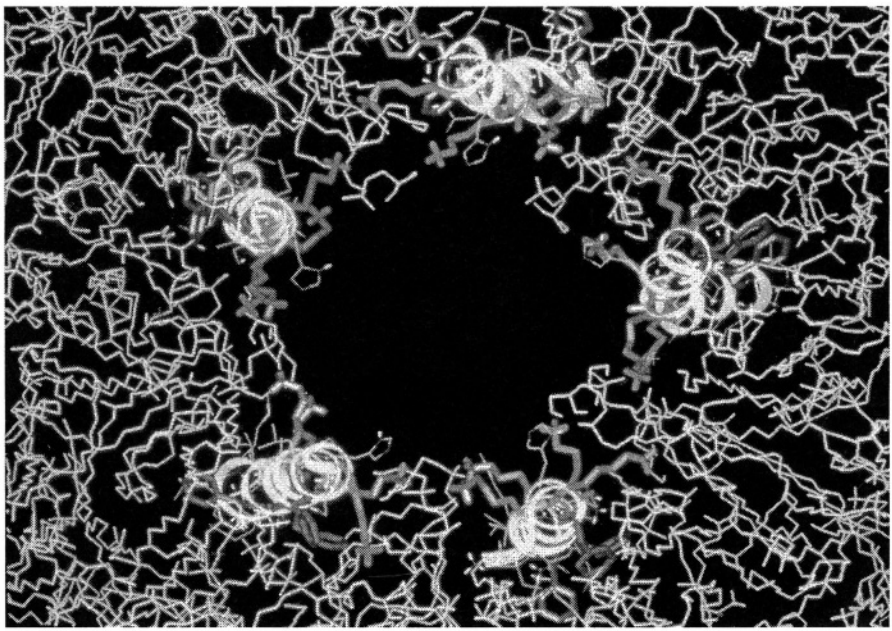


Fig. 1. Top view of M2a-POPG toroidal pore in the bacterial membrane. M2a molecules are shown as light ribbons, the Lys and Phe residues are shown as sticks, and phospholipid molecules (mainly POPG) are shown in white as lines. For clarity, water molecules were removed

2.3 Simulation Conditions

All three bilayer systems contain charged molecules and ions, therefore, in the MD simulations, long-range electrostatic and van der Waals interactions were evaluated by means of the Particle-Mesh-Ewald (PME) summation method [12]. A real cutoff of 12 Å, with a β-spline interpolation order of 5, and a direct sum tolerance of 10^{-6} were used. Three-dimensional periodic boundary conditions were employed. The

simulations were carried out using AMBER 5.0 [13]. The SHAKE algorithm [14] was used to preserve the bond lengths of the OH, NH, and NH_2 groups of water and peptide molecules. The list of nonbonded pairs was updated every 25 steps.

The MD simulations were carried out at a constant pressure (1 atm) and at a temperature of 310 K (37 °C). The temperatures of the solute and solvent were controlled independently by the Berendsen method [15]. The applied pressure was controlled anisotropically, each direction being treated independently with the trace of the pressure tensor kept constant for 1 atm, again using the Berendsen method [15]. The relaxation times for temperatures and pressure were set at 0.4 and 0.6 ps, respectively. The bilayers were simulated for 12 ns (EUCARPET and PROCARPET) and 5 ns (PORE). For analyses, last 10-ns fragments of EUCARPET and PROCARPET trajectories and the last 3-ns fragment of PORE trajectory were used. The average values given below are ensemble and time averages obtained from the block averaging procedure. Errors in the derived average values are standard error estimates.

2.4 Analyses

The instantaneous orientation of the peptide molecules in the bilayers was monitored using a rigid body quaternion fit as implemented in MMTK package [16]. Details are given in [6]. The reference helical structure of the M2a molecule was the experimentally determined peptide structure [1].

3 Results

3.1 Short Distance Lipid-Peptide Interactions

3.1.1 Animal Membrane

In EUCARPET, M2a molecules are positioned in the interfacial region of the membrane. Such localisation allows them to interact both with the polar and non-polar fragments of POPC and Chol molecules. The side-chains of polar and charged amino acids interact with the phosphate and carbonyl groups of POPC and the hydroxyl group of Chol via direct hydrogen bonds (H-bond) and water bridges (WB). The largest contribution to M2a-POPC polar interactions have two Lys residues (Lys10 and 14), and to M2a-Chol polar interactions has Glu19 (Fig. 2). The side-chains of non-polar residues and non-polar fragments of POPC and Chol interact via van der Waals interactions. The largest contribution to M2a-POPC non-polar interactions have mainly aromatic residues (Phe7, 12, and 16), and to M2a-Chol non-polar interactions mainly aliphatic residues (Ile2, Leu6, and Val17). Numbers of M2a-lipid interactions are given in Table 1.

3.1.2 Bacterial Membrane

In PROCARPET, localisation of M2a molecules is similar to that in EUCARPET. Nevertheless, non-polar interactions between M2a and phospholipids are significantly stronger than in EUCARPET, particularly those of Phe12 and Phe16. The number of direct M2a-lipid H-bonds is similar in both membranes, whereas the number of M2a-

lipid water bridges is much less than in EUCARPET. The largest contribution to M2a-lipid polar interactions have Lys4, 10, and 11, and Gly19. Numbers of M2a-lipid interactions are given in Table 1.

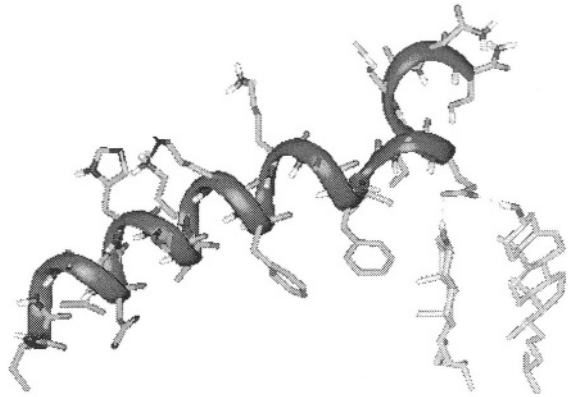


Fig. 2. One of two M2a molecules in the animal membrane together with two Chol molecules that are simultaneously H-bonded to Glu19. M2a is shown as dark ribbon, the residues and Chol molecules are shown as sticks

Table 1. Average numbers of hydrogen bonds (H-bonds) and water bridges (WB) between a M2a and surrounding lipids, as well as carbon atoms in phospholipid or Chol molecules that are within 7.5Å from any atom of M2a non-polar residues (non-polar interactions, NP) in EUCARPET, PROCARPET and PORE. Errors are standard error estimates

	EUCARPET		PROCARPET	PORE
	POPC	CHOL		
H-bonds	6.3 ± 0.4	1.1 ± 0.1	7.0 ± 0.5	7.2 ± 0.6
WB	11.4 ± 0.8	1.8 ± 0.2	7.1 ± 0.4	7.9 ± 0.5
NP	1138 ± 45	93 ± 4	1301 ± 50	1155 ± 40

3.1.3 Toroidal Pore

In PORE, M2a molecules together with POPG headgroups form a toroidal pore. Even though M2a are located perpendicular to the bilayer surface, their local environment is similar to the interfacial region of PROCARPET. Consequently, the pattern of polar and non-polar interactions between peptide and lipid molecules are to a large extent similar to those in PROCARPET. Numbers of M2a-lipid interactions are given in Table 1. The strongest contribution to polar interactions arises from all four Lys residues, whereas to non-polar interactions arises form all three Phe residues.

3.2 Orientation and Conformation of Peptide Molecules in the Membrane

3.2.1 Animal Membrane

The M2a molecules in EUCARPET remain nearly parallel to the bilayer surface and their orientation is similar to the initial one. Strong H-bonding between two Chol molecules and Glu 19 of one M2a molecule (Fig. 2) results in a local loss of helicity of this molecule.

3.2.2 Bacterial Membrane

The M2a molecules in PROCARPET also remain nearly parallel to the bilayer surface nevertheless, their orientation differs from the initial one. The peptides rotated about their long axes by an average angle of ~15°.

3.2.3 Toroidal Pore

The M2a molecules in PORE fluctuate about the orientation perpendicular to the membrane surface. Like in PROCARPET, the peptide molecules underwent limited rotation about their long axes.

4 Conclusions

1. Interactions between Chol and M2a in EUCARPET impeded favourable interaction of the peptide with membrane phospholipids. On the longer time scale, this may lead to desorption of the peptide from the membrane surface.
2. M2a molecules in PROCARPET, by rotating about their long axes, assumed orientation that maximises their interaction with both polar and non-polar groups of phospholipids. These interactions affect the organisation of both interfacial and hydrophobic membrane region. On the longer time scale, this may lead to local destruction of the membrane lamellar structure.
3. M2a-lipd interactions in PORE are similar to those in PROCARPET. These interactions stabilize the pore structure.

Acknowledgements. This work was supported by grants 6 P04A 041 16, 6P04A 031 21 and KBN/SGI ORIGIN 2000/UJ/048/1999 from the Committee for Scientific Research and partially by European Union (contract no. BIER ICA1-CT-2000-70012). KM and TR acknowledge a fellowship award from the Polish Foundation for Science.

References

1. Gesell, J., Zasloff, M., Opella, S.J.: Two-dimensional H-1 NMR experiments show that the 23-residue magainin antibiotic peptide is an alpha-helix in dodecylphosphocholine micelles, sodium dodecylsulfate micelles, and trifluoroethanol/water solution. J. Biomol. NMR. 9 (1997) 127-135

2. Ludtke, S.J., He, K., Wu, Y., Huang, H.: Cooperative membrane insertion of magainin correlated with its cytolytic activity. Biochim. Biophys. Acta 1190 (1994) 181-184
3. Matsuzaki, K., Murase, O., Furii, N., Miyajima, M.: An antimicrobial peptide, magainin 2, induced rapid flip-flop of phospholipids coupled with pore formation and peptide translocation. Biochemistry 35 (1996) 11361-11368
4. Róg, T.: Effects of cholesterol on the structure and dynamics of phospholipid bilayers: a molecular dynamics simulation studies. Ph.D. thesis. Jagiellonian University, Poland (2000) 1-147
5. Murzyn, K., Pasenkiewicz-Gierula, M.: Construction and optimisation of a computer model for a bacterial membrane. Acta Biochim. Pol. 46 (1999) 631-639
6. Murzyn, K., Pasenkiewicz-Gierula, M.: Construction of a toroidal model for the magainin pore. J. Mol. Mod. 9 (2003) 217-224
7. Jorgensen, W.L., Tirado-Rives, J.: The OPLS potential functions for proteins. Energy minimizations for crystals of cyclic peptides and crambin. J. Am. Chem. Soc. 110 (1988) 1657-1666
8. Jorgensen, W.L., Chandrasekhar, J., Madura, J.D., Impey, R.W., Klein M.L.: Comparison of simple potential functions for simulating liquid water. J. Chem. Phys. 79 (1983) 926-935
9. Charifson, P.S., Hiskey, R.G., Pedersen, L.G.: Construction and molecular modeling of phospholipid surfaces. J. Comp. Chem. 11 (1990) 1181-1186
10. Pasenkiewicz-Gierula, M., Takaoka, Y., Miyagawa, H., Kitamura, K., Kusymi, A.: Charge pairing of headgroups in phosphatidylcholine membranes: A molecular dynamics simulation study. Biophys. J. 76 (1999) 1228-1240
11. Murzyn, K., Róg, T., Jezierski, G., Takaoka, Y., Pasenkiewicz-Gierula, M.: Effects of phospholipid unsaturation on the membrane/water interface: a molecular simulation study. Biophys. J. 81 (2001) 170-183
12. Essmann, U., Perera, L., Berkowitz, M.L., Darden, T., Lee, H., Pedersen, L.G.: A smooth particle mesh Ewald method. J. Chem. Phys. 103 (1995) 8577-93.
13. Case, D.A., Pearlman, D.A., Caldwell, J.W., Cheatham III, T.E., Ross, W.S., Simmerling, C., Darden, T.A., Merz, K.M., Stanton, R.V., Cheng, A.L., Vincent, J.J., Crowley, M., Ferguson, D.M., Radmer, R.J., Seibel, G.L., Singh, U.C., Weiner, P.K., Kollman, P.A.: AMBER 5.0. University of California, San Francisco (1997)
14. Ryckaert, J. P., Cicotti, G., Berendsen, H.J.C.: Numerical integration of the Cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. J. Comp. Phys. 22 (1977) 327-341
15. Berendsen, H.J.C., Postma, J.P.M., van Gunsteren, W.F., DiNola, A., Haak, J.R.: Molecular dynamics with coupling to an external bath. J. Chem. Phys. 81 (1984) 3684-3690
16. Hinsen, K.: The molecular modeling toolkit: A new approach to molecular simulations. J. Comp. Chem. 21 (2000) 79-85

Dynamics of Granular Heaplets: A Phenomenological Model

Yong Kheng Goh[1,2] and R.L. Jacobs[2]

[1] Department of Mathematical Sciences, University of Essex, Colchester CO4 3SQ, United Kingdom
kygoh@essex.ac.uk
[2] Department of Mathematics, Imperial College, London SW7 2AZ, United Kingdom
r.jacobs@imperial.ac.uk

Abstract. When a discrete granular layer on a uniform substrate is tapped from beneath the material piles up into discrete heaps which gradually coarsen. We investigate the relaxation dynamics of the heaping process. We present a non-linear phenomenological partial differential equation to describe the formation of the heaplets. This equation is derived from the continuity equation for a diffusive powder system and from a constitutive equation giving the current as the sum of three terms: the first proportional to the gradient of the height profile with a limiting factor, the second related to the average curvature of the heap surface and the third related to the Gaussian curvature.

1 Introduction

When we perturb a layer of granular material a rich variety of interesting phenomena can occur depending on the nature of the perturbation. Examples are subharmonic wave patterns and oscillons in vertically vibrated granular layers [1, 2]; compaction and memory effects in tapped tall granular columns [3,4]; and stratification of a granular mixture flowing down an inclined plane [5,6]. Even simple experiments such as tapping a thin granular layer, result in interesting phenomena such as the formation of isolated granular "droplets" (*heaplets*) [7].

In this article we are particularly interested in a simple system that consists of a *thin* layer of granular material subjected to a series of discrete taps from beneath. We are interested in constructing a computer model of the dynamics of formation of heaplets in the tapped granular layer. The first section of the article describes an experimental setup in which the phenomena can be observed and also shows typical morphologies that develop. Then we introduce our phenomenological model by constructing a surface free-energy functional. The dynamics of the system is then derived by setting the time-derivative of a density equal to the functional derivative of the free-energy and the result is cast into the form of an equation of continuity. The equation is solved numerically and results are presented. Finally we summarise our results.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 332–339, 2004.
© Springer-Verlag Berlin Heidelberg 2004

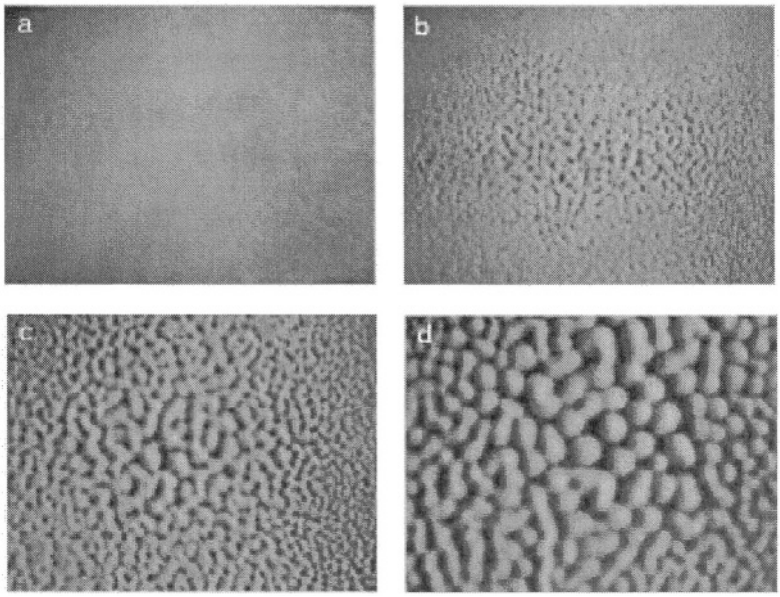


Fig. 1. A layer of silica beads subjected to gentle taps. The first picture (a) is the initial flat layer with pictures (b)–(d) showing the morphology of the layer after $k = 20, 40,$ and 100 taps.

2 A Simple Experiment

A homogeneous and *thin* (1–3 particles depth) layer of silica beads (approximately 30μm in diameter) is prepared on a thin glass plate. The layer is then tapped from below at the center. After each tap we wait for a long enough period until all activity on the surface of the layer has ceased before tapping again. Figure 1 shows a series of photographs obtained from an experiment on a tapped silica layer with nearly constant tapping intensity.

The layer is initially flat as shown in Figure 1a. After several taps, the flat layer of silica beads becomes unstable and starts to corrugate (Figure 1b). As the number of taps increases, corrugations coarsen and the layer now forms a landscape of ridges (Figure 1c). The pattern finally develops into more isolated heaplets (Figure 1d). The experiment is fairly robust in the sense that the frequency and intensity of the taps do not have to be carefully controlled for the pattern to develop. When the pattern is fully developed, the heaplets become nearly stable against any further taps, with the characteristic size and the rate of the formation of heaplets being proportional to the intensity of the taps.

Duran [8] made a close analogy between the heaplet formation process and the de-wetting process of a layer of water on a glass plate. The value of this analogy can be seen from the above figures where clusters of silica beads slowly coalesce to form small heaplets as if under the influence of surface tension. Of

course there is no true surface tension since the silica beads interact via hard core repulsions but an *effective* surface tension can come arise from convective drags [7] or the inelasticity of the particles [9].

3 Equation of Continuity and Current Equation

It is useful to have quasi-hydrodynamic equations to describe the dynamics of heaplet formation. However there is no obvious time variable in the problem and we need to take care in the continuum modelling. In the tapped layer there are two distinct phases of activity with different time scales in the dynamics. The first is an excitation phase when the layer is tapped. This is followed by a relaxation phase during which the perturbed layer relaxes towards a meta-stable state. Ordinary clock time cannot be used as a proper time variable for the purpose of continuum modelling, since the actual duration between two consecutive taps is unimportant (provided the system relaxes completely.) What is important is the number of tap cycles that the layer has undergone. Therefore a more useful "time" variable is the number of taps rather than the clock time. In this paper time t refers to a *continuum* version of the number of taps.

We want to set-up an equation of mass continuity with a constitutive current equation to model heaplet formation on a tapped granular layer. The current equation must describe lateral diffusion of the granular particles to regions of higher density. This leads to an effective attraction between particles and encourages the formation of heaps. However the slope of the side of a static heap cannot exceed a value determined by the angle of repose, θ_r of the material. This suggests that the effective diffusion is limited by a slope-dependent factor and goes to zero at a critical slope equal to $\tan\theta_r$. Further terms are needed in the current equation and are related to the curvatures of the surface of the layer. These terms should describe the situation when the system is tapped and protrusions are eroded away so that the sides of the heaps become as smooth as possible i. e. the curvature of the surface becomes as small as possible.

Gathering these requirements we introduce a surface free energy functional

$$\mathcal{F}[\rho] = \int \mathrm{d}^2\boldsymbol{x} \left\{ -\frac{1}{2}D(\gamma)\left(|\nabla\rho|^2 - \frac{1}{2}B|\nabla\rho|^4\right) + \kappa(\gamma)[\mathrm{tr}(\rho_{ij})]^2 + \kappa'(\gamma)[\det(\rho_{ij})]^2 \right\} \tag{1}$$

where the density field $\rho(\boldsymbol{x}, t)$ i. e. the mass per unit area is proportional to the height of the layer assuming no compaction during tapping. D, κ and κ' are positive parameters that depend on the tapping intensity γ, and $1/\sqrt{B} = \tan\theta_r$ is the critical slope of the material. Here ρ_{ij} is the Hessian matrix $\rho_{ij} = \partial^2\rho/\partial x_i \partial x_j$. The first two terms mimic the effective attractive and give rise later to a diffusion equation with negative time i. e. one that favours the accumulation rather than the dissipation of heaps. The trace and determinant in the last two terms are the two rotational invariants of the Hessian matrix.

Now we assume that $\rho(\boldsymbol{x}, t)$ evolves so as to minimise the free energy according to model A Langevin dynamics, in Gunton's terminology [10], and this gives

an equation of motion of form

$$\frac{\partial \rho}{\partial t} = -\zeta \frac{\delta \mathcal{F}}{\delta \rho(\boldsymbol{x}, t)}, \tag{2}$$

where ζ is a rate constant that sets the time scale of the growth of the pattern and is set to unity by choice of units. $\mathcal{F}$ is defined as in (1). Taking the functional derivative of the free energy and substituting into (2) we get

$$\frac{\partial \rho}{\partial t} = -\boldsymbol{\nabla} \cdot \left\{ D(\gamma)(1 - B|\nabla \rho|^2)\boldsymbol{\nabla}\rho + \kappa(\gamma)\boldsymbol{\nabla}(\nabla^2 \rho) + \kappa'(\gamma)\boldsymbol{g} \right\}. \tag{3}$$

The components of the two-dimensional vector $\boldsymbol{g}$ in the last equation are

$$g_x = \frac{\partial \mathcal{K}}{\partial x}\frac{\partial^2 \rho}{\partial y^2} - \frac{\partial \mathcal{K}}{\partial y}\frac{\partial^2 \rho}{\partial x \partial y} \tag{4}$$

and

$$g_y = \frac{\partial \mathcal{K}}{\partial y}\frac{\partial^2 \rho}{\partial x^2} - \frac{\partial \mathcal{K}}{\partial x}\frac{\partial^2 \rho}{\partial x \partial y}, \tag{5}$$

where $\mathcal{K} = \det(\rho_{ij})$. This is just the equation of continuity with a constitutive equation for the mass current

$$\boldsymbol{J} = D(\gamma)(1 - B|\nabla \rho|^2)\boldsymbol{\nabla}\rho + \kappa(\gamma)\boldsymbol{\nabla}(\nabla^2 \rho) + \kappa'(\gamma)\boldsymbol{g}. \tag{6}$$

However $\boldsymbol{g}$ given by (4) and (5) is not unique and $\boldsymbol{g}$ is chosen in the above form to avoid calculating fifth derivatives numerically later in the simulation. Alternative forms of $\boldsymbol{g}$ can give rise to the same equation for ρ on substituting into (3). These forms of $\boldsymbol{g}$ differ from the form given above by an additional gauge term which is the curl of an arbitrary vector. One example of these alternative forms is found by interchanging $\mathcal{K}$ and ρ in (4) and (5).

We now give a physical interpretation of the terms in the free energy functional. The first term and the second term give the diffusion term proportional to $\boldsymbol{\nabla}\rho$ with limiting factor $(1 - B|\nabla \rho|^2)$ in (3). The diffusion term must have the opposite sign to the usual (if $|\nabla \rho|$ is small and the limiting factor is positive) in order to mimic the short-range attractive force between particles. Thus the coefficient D must be positive so that grains diffuse toward each other to form clusters. The limiting factor $(1 - B|\nabla \rho|^2)$ can be understood as the result of anisotropy in the system due to gravity and ensures that the slope of the heaps cannot exceed the critical slope. Because of the opposite sign of the diffusion term the equation can be understood as a negative time diffusion equation.

The last two terms in $\mathcal{F}$ come from the squares of the two rotational invariants of the Hessian matrix ρ_{ij}. These invariants are the trace and the determinant of ρ_{ij}. They are lowest order approximations to the average and Gaussian curvatures of the surface and we now refer to them as such. We can understand these curvature terms intuitively thus. As the system is tapped protrusions are eroded so that the sides of the heaps becomes as smooth as possible i. e. the curvatures of the surface become as small as possible. Equation (2) ensures the

system evolves to minimise the curvatures of the surface. There are no linear terms in $\mathcal{F}$ proportional to the trace and determinant because as we can show, with periodic (or free) boundaries, these give no contribution to the equation of motion of ρ.

4 Calculations

We now solve (3) with (4) and (5) numerically by discretising the spatial variables on an $N \times N$ square lattice using a simple mid-point algorithm to deal with the time variable. We can use either periodic boundary conditions or hard boundary conditions. However we are only interested here in the pattern formation process, not the interaction of the system with the container wall so here we use only periodic boundary conditions. Various initial conditions are possible but here we start off with a level granular layer with random height fluctuations of amplitude 10^{-5} superimposed. This amplitude is small compared to the final average height of the heaps which is determined by the value of the parameter B and the length of time for which the calculation is run.

Stability problems arise if κ is negative. This can be easily seen if we take the Fourier transform of the linearised equation which is

$$\frac{\partial \rho_k}{\partial t} = (D\,k^2 - \kappa\,k^4)\rho_k. \tag{7}$$

If κ is positive small fluctuations of long-wave-length in the initial state are unstable to growth until limited by the non-linear term proportional to B in $\boldsymbol{J}$. This is in accordance with observation. If κ is negative small fluctuations of long-wave-lengths in the initial state grow and are not limited by the non-linear term because the term proportional to κ dominates. Negative values of κ and κ' are thus quite unphysical because they imply that protrusions are not eroded away but grow indefinitely as they accrete grains in a situation where there is no real attractive force between grains.

As time increases instabilities develop from the density fluctuations and grow into heaps. While the heaps grow grains from the surrounding areas are depleted. This creates discontinuities in the layer and introduces extra complications when calculating derivatives of the density field. To avoid this we assume in the simulation that the surface of the layer never touches the base of the container. The assumption enables us to avoid these problems but it has a draw-back since the heaps will continue growing until the whole system unrealistically becomes a single big heap. In reality the surface of the layer eventually touches the base of the container and the heaps become very nearly stable and cease to grow.

In the following we take a box size $L = 1$ divided into 10^4 cells so that $N = 100$. We take the following values for the parameters: $D = 1$, $B = 10^3$, $\kappa = 10^{-4}$, and we vary κ' so that we can see the balance between the terms affecting the occurrence of heaps and ridges. We use periodic boundary conditions throughout.

Figure 2 shows late-time plots of the surface for values of the parameters where the Gaussian curvature term is unimportant ($\kappa' = 10^{-6}$). The appearance

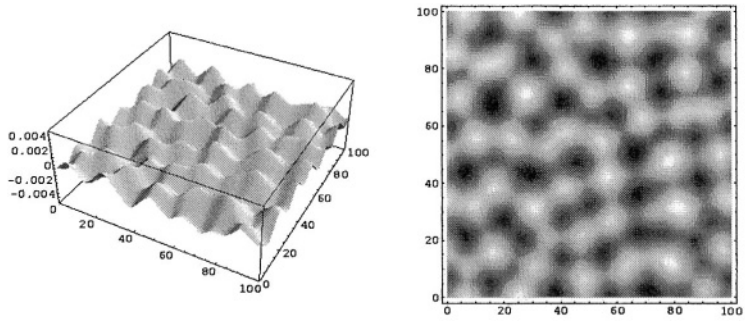


Fig. 2. A three-dimensional plot and a grey-scale contour plot of the surface produced by our differential equation after a long time. The parameters used to produce the plot favour the production of heaps i. e. $\kappa' \ll \kappa$.

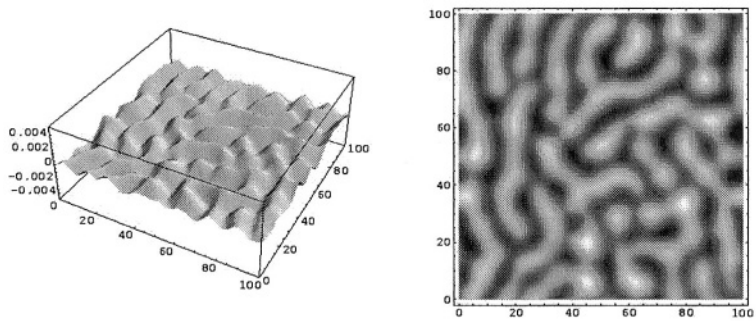


Fig. 3. A three-dimensional plot and a grey-scale contour plot of the surface produced by our differential equation after a long time. The parameters used to produce the plot favour the production of ridges i.e. when κ' is comparable to κ (in this case $\kappa' = \kappa$).

of the heaps is similar to those found experimentally except that our calculations display structures that are more homogeneous which is to be expected for a surface with no edges and periodic boundary conditions. Figure 3 shows late-time plots of the surface for values of the parameters where the Gaussian curvature term is important ($\kappa' = 10^{-4}$). The resulting structure is a pattern of labyrinthine ridges. The pattern is similar to the patterns that have been found in the early stage of the experiments performed (c.f. Figure 1c). An interesting feature of these last figures from a theoretical point of view is that they show long-range correlations which arise from a purely local theory. The term discouraging saddles leads to long-range correlations along the ridges.

The differences in the morphologies of the results can be understood by looking at the surface free energy in (1). Equation (3) is derived by minimising the

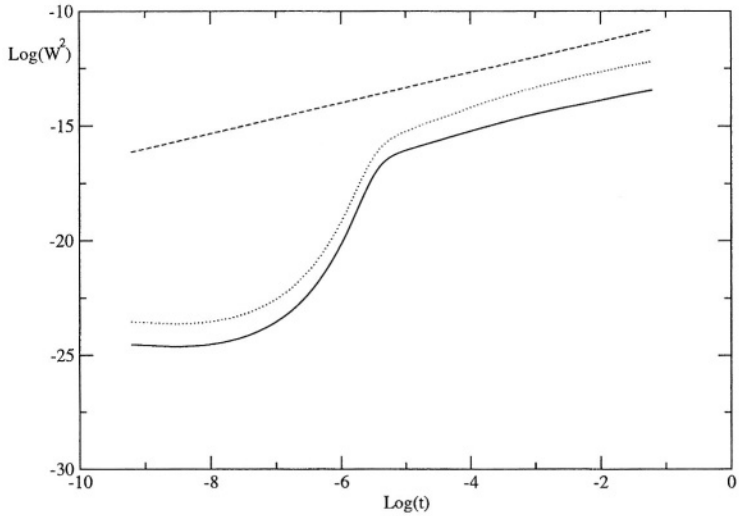


Fig. 4. Log-log plot for $\mathcal{W}^2$ versus t. The solid line corresponds to the parameter $\kappa' = 10^{-6}$, the dotted line corresponds to $\kappa' = 10^{-4}$ and they are offset vertically by unity for clarity. A reference line (dashed) with slope 2/3 is plotted to aid comparison with the late-time behaviour for both small and large κ' curves.

functional $\mathcal{F}[\rho]$, and therefore the dynamic of the system is towards minimal surface curvature. For small values of κ', the trace term is important and $[\mathrm{tr}(\rho_{ij})]^2$ is small at locations where two local perpendicular curvatures curve in opposite directions $\partial^2\rho/\partial y^2 = -\partial^2\rho/\partial x^2$, i. e. at saddle points or where the surface is smooth. Thus the equation produces a profusion of saddle points. Thus the fact that two heaps (or two valleys) join at a saddle point suggests that minimising the trace term encourages the formation of as many heaps as possible. On the other hand, for slightly larger values of κ', the determinant term encourages the formation of ridges. This can be seen from a similar argument to that used for the trace term. We can always choose a local reference frame (x', y') such that $\det(\rho_{ij}) = \partial^2\rho/\partial x'^2\, \partial^2\rho/\partial y'^2$. Minimising the determinant term in (1) requires $\partial^2\rho/\partial x'^2\, \partial^2\rho/\partial y'^2 = 0$ which means the surface is either flat in one direction but is curved in the perpendicular direction (which is true for ridges); or the surface is flat in both directions (but unstable due to the diffusion term).

Figure 4 shows a log-log plot of the roughness $\mathcal{W}^2$ of the pattern for both small $(\kappa' = 10^{-6})$ and large $(\kappa' = 10^{-4})$ values of κ'. Here $\mathcal{W}^2$ is defined as

$$\mathcal{W}^2 = \left\langle (\rho(\boldsymbol{x}, t) - \bar{\rho})^2 \right\rangle, \tag{8}$$

where the average is over all lattice sites. $\mathcal{W}^2$ measures the mean square deviation of the height of the surface from its mean and is zero for a level surface and grows as the heaplets grow. $\mathcal{W}^2$ is proportional to the square of the characteristic length scale of the pattern in the horizontal direction because the slope of the surface is almost everywhere equal to the critical slope $1/\sqrt{B}$. As can be seen from

Figure 4, the dynamics of (3) can be divided into at least 3 regimes. For $t < 10^{-7}$, the system tends to become smoother because slopes greater than the critical slope decrease. For $10^{-7} < t < 10^{-5}$ there is a rapid formation of heaps due to the growth of fluctuations. For $t > 10^{-5}$ the heaps have reached their critical slope and then grow coarser but with the fixed critical slope almost everywhere. In the late-time regime, $\mathcal{W}^2 \sim t^{2/3}$. This scaling law is consistent with Siegert and Plishke's model [11] of molecular-beam epitaxy deposition. Despite an extra deposition term and the absence of a Gaussian term in their model, they have observed the same scaling law for the characteristic size of the mound formed in the deposition process.

5 Conclusion

We have studied a phenomenological model of a tapped granular layer. The model is derived by minimising a surface free energy consisting of four terms: two terms which give rise to the slope-limited negative-time diffusion, the average curvature term, and the Gaussian curvature term. The negative-time diffusion terms mimic the clustering effects described also by an effective surface tension. The curvature terms are needed because the system must evolve to a configuration of less protrusion. They also control the pattern formed: a striped pattern if the Gaussian curvature term is important; discrete heaplets if the average curvature term is dominant. In both cases the pattern coarsens with the number of taps and the coarsening of the width of the pattern is described by $\mathcal{W} \sim t^{1/3}$.

References

1. Melo, F., Umbanhowar, P.B., Swinney, H.L.: Hexagons, kinks, and disorder in oscillated granular layer. Phys. Rev. Lett. **75** (1995) 3838
2. Umbanhowar, P.B., Melo, F., Swinney, H.L.: Localised excitation in a vertically vibrated granular layer. Nature **382** (1996) 793
3. Knight, J.B., Fandrich, C.G., Lau, C.N., Jaegar, H.M., Nagel, S.R.: Density relaxation in a vibrated granular material. Phys. Rev. E **51** (1995) 3957
4. Nowak, E.R., Knight, J.B., Ben-Naim, E., Jaeger, H.J., Nagel, S.R.: Density fluctuations in vibrated granular materials. Phys. Rev. E **57** (1998) 1971
5. Makse, H.A., Havlin, S., King, P.R., Stanley, H.E.: Spontaneos stratification in granular mixtures. Nature **386** (1997) 379
6. Ristow, G.H., Riguidel, F.X., Bideau, D.: Different characteristics of the motion of a single particle on a bumpy inclined line. J. Phys. II France **4** (1994) 1161
7. Duran, J.: Rayleigh-Taylor instabilities in thin films of tapped powder. Phys. Rev. Lett. **87** (2001) 254301
8. Duran, J.: Ripples in tapped or blown powder. Phys. Rev. Lett. **84** (2000) 5126
9. Goh, Y.K., Jacobs, R.L.: Coarsening dynamics of granular heaplets in tapped granular layers. New J. Phys. **4** (2002) 81
10. Gunton, J.D., Droz, M.: Introduction to the theory of metastable and unstable states. Volume 183 of Lecture notes in physics. Springer-Verlag, Berlin (1983)
11. Siegert, M., Plischke, M.: Formation of pyramids and mounds in molecular beam epitaxy. Phys. Rev. E **53** (1996) 307

Modelling of Shear Zones in Granular Materials within Hypoplasticity

Jacek Tejchman

Civil Engineering Department, Gdansk University of Technology,
80-952 Gdansk, Poland
tejchmk@pg.gda.pl

Abstract. This paper presents a FE-analysis of shear localization in granular bodies with a finite element method based on a hypoplastic constitutive law. The law can reproduce essential features of granular bodies in dependence on the void ratio, pressure level and deformation direction. To simulate the formation of a spontaneous shear zone inside of cohesionless sand during plane strain compression, a hypoplastic law was extended by polar, non-local and gradient terms. The effects of 3 different models on the thickness of a shear zone was investigated.

1 Introduction

Localization of deformation in the form of narrow zones of intense shearing can develop in granular bodies during processes of granular flow or shift of objects with sharp edges against granular materials. Shear localization can occur spontaneously as a single zone, in several zones or in a regular pattern. They can also be induced in granular bodies along walls of stiff structures at granular bodies. An understanding of the mechanism of the formation of shear zones is important since they act as a precursor to ultimate failure.

Classical FE-analyses of shear zones are not able to describe properly both the thickness of localization zones and distance between them since they suffer from a spurious mesh sensitivity (to mesh size and alignment). The rate boundary value problem becomes ill-posed. (i.e. the governing differential equations of equilibrium or motion change the type by losing ellipticity for static and hiperbolicity for dynamic problems) [1]. Thus, the localization is reduced to a zero-volume zone. To overcome this drawback, classical constitutive models require an extension in the form of a characteristic length to regularize the rate boundary value problem and to take into account the microstructure of materials (e.g. size and spacing of micro-defects, grain size, fiber spacing). Different strategies can be used to include a characteristic length and to capture properly the post-peak regime (in quasi-static problems): polar models [2], non-local models [3] and gradient models [4].

In this paper, a spontaneous shear localization in granular bodies was investigated with a finite element method based on a hypoplastic constitutive law extended by polar, non-local and gradient terms.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 340–347, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Hypoplasticity

Hypoplastic constitutive models [5], [6] are an alternative to elasto-plastic ones for continuum modelling of granular materials. In contrast to elasto-plastic models, a decomposition of deformation components into elastic and plastic parts, yield surface, plastic potential, flow rule and hardening rule are not needed. The hypoplastic law includes barotropy (dependence on pressure level), pycnotropy (dependence on density), dependence on the direction of deformation rate, dilatancy and contractancy during shearing with constant pressure, increase and release of pressure during shearing with constant volume, and material softening during shearing of a dense material. The feature of the model is a simple formulation and procedure for determination of material parameters with standard laboratory experiments [7]. Owing to that one set of material parameters is valid within a large range of pressures and densities. The constitutive law can be summarised as follows:

$$\overset{o}{\sigma}_{ij} = f_s [L_{ij}(\hat{\sigma}_{kl}, d_{kl}) + f_d N_{ij}(\hat{\sigma}_{ij})\sqrt{d_{kl} d_{kl}}], \tag{1}$$

$$L_{ij} = a_1^2 d_{ij} + \hat{\sigma}_{ij}^{*} \hat{\sigma}_{kl}^{*} d_{kl}, \qquad N_{ij} = a_1 (\hat{\sigma}_{ij} + \hat{\sigma}_{ij}^{*}), \tag{2}$$

$$\overset{o}{\sigma}_{ij} = \overset{\bullet}{\sigma}_{ij} - w_{ik}\sigma_{kj} + \sigma_{ik} w_{kj}, \tag{3}$$

$$d_{ij} = 0.5(v_{i,j} + v_{j,i}), \qquad w_{ij} = 0.5(v_{i,j} - v_{j,i}), \tag{4}$$

$$\hat{\sigma}_{ij} = \frac{\sigma_{ij}}{\sigma_{kk}}, \qquad \hat{\sigma}_{ij}^{*} = \hat{\sigma}_{ij} - \frac{1}{3}\delta_{ij}, \tag{5}$$

$$f_s = \frac{h_s}{n h_i}\left(\frac{1+e_i}{e}\right)\left(-\frac{\sigma_{kk}}{h_s}\right)^{1-n}, \qquad h_i = \frac{1}{c_1^2} + \frac{1}{3} - \left(\frac{e_{i0} - e_{d0}}{e_{c0} - e_{d0}}\right)^{\alpha} \frac{1}{c_1\sqrt{3}}, \tag{6}$$

$$\overset{\bullet}{e} = (1+e) d_{kk}, \qquad f_d = \left(\frac{e - e_d}{e_c - e_d}\right)^{\alpha}, \tag{7}$$

$$e_d = e_{d0} \exp[-(-\sigma_{kk}/h_s)^n], \qquad e_i = e_{i0} \exp[-(-\sigma_{kk}/h_s)^n], \tag{8}$$

$$e_c = e_{c0} \exp[-(-\sigma_{kk}/h_s)^n], \tag{9}$$

$$a_1^{-1} = c_1 + c_2\sqrt{\hat{\sigma}_{kl}^{*} \hat{\sigma}_{lk}^{*}}\,[1 + \cos(3\theta)], \qquad \cos(3\theta) = -\frac{\sqrt{6}}{[\hat{\sigma}_{kl}^{*} \hat{\sigma}_{kl}^{*}]^{1.5}} (\hat{\sigma}_{kl}^{*} \hat{\sigma}_{lm}^{*} \hat{\sigma}_{mk}^{*}), \tag{10}$$

$$c_1 = \sqrt{\frac{3}{8}}\frac{(3 - \sin\phi_c)}{\sin\phi_c}, \qquad c_2 = \frac{3}{8}\frac{(3 + \sin\phi_c)}{\sin\phi_c}, \tag{11}$$

wherein: σ_{ij} - Cauchy stress tensor, e – current void ratio, $\overset{o}{\sigma}_{ij}$ - Jaumann stress rate tensor, $\hat{\sigma}_{ij}$ - normalised stress tensor, $\hat{\sigma}_{ij}^{*}$ - deviatoric part of the normalised stress tensor, d_{ij}- rate of deformation tensor, w_{ij}- spin tensor, $v_{i,j}$- gradient of velocity, f_s -

stiffness factor, f_d – density factor, h_s - granular hardness, θ - Lode angle, e_c - critical void ratio, e_d – minimum void ratio, e_i - maximum void ratio, e_{i0} - maximum void ratio at pressure equal to zero, e_{d0} - minimum void ratio at pressure equal to zero, e_{c0} - critical void ratio at pressure equal to zero, ϕ_c - critical angle of internal friction during stationary flow, n – compression coefficient, α - pycnotropy coefficient, a_1 – coefficient determining the shape of the stationary stress surface.

The constitutive relationship requires 7 material constants: e_{i0}, e_{d0}, e_{c0}, ϕ_c, h_s, n and α. The FE-analyses were carried out with the following material constants (for so-called Karlsruhe sand): e_{i0}=1.3, e_{d0}=0.51, e_{c0}=0.82, ϕ_c=30°, h_s=190 MPa, n=0.5 and α=0.3 [6].

A hypoplastic constitutive law cannot describe realistically shear localization since it does not include a characteristic length. A characteristic length was taken into account by means of a polar, non-local and gradient theory.

3 Enhanced Hypoplasticity

3.1 Polar Hypoplasticity

The polar terms were introduced in a hypoplastic law (Eqs.1-11) with the aid of a polar (Cosserat) continuum [2]. Each material point has for the case of plane strain three degrees of freedom: two translational degrees of freedom and one independent rotational degree of freedom. The gradients of the rotation are connected to curvatures which are associated with couple stresses. It leads to a non-symmetry of the stress tensor and a presence of a characteristic length.

The constitutive law can be summarised for plane strain as follows [9], [10] (Eqs.3-11 and Eqs.12-17):

$$\overset{o}{\sigma}_{ij} = f_s [L_{ij}(\hat{\sigma}_{kl}, \hat{m}_k, d^c_{kl}, k_k d_{50}) + f_d N_{ij}(\hat{\sigma}_{ij}) \sqrt{d^c_{kl} d^c_{kl} + k_k k_k d^2_{50}}\,], \tag{12}$$

$$\overset{o}{m}_i / d_{50} = f_s [L^c_i(\hat{\sigma}_{kl}, \hat{m}_k, d^c_{kl}, k_k d_{50}) + f_d N^c_i(\hat{m}_i) \sqrt{d^c_{kl} d^c_{kl} + k_k k_k d^2_{50}}\,], \tag{13}$$

$$L_{ij} = a_1^2 d^c_{ij} + \hat{\sigma}_{ij}(\hat{\sigma}_{kl} d^c_{kl} + \hat{m}_k k_k d_{50}), \quad L^c_i = a_1^2 k_i d_{50} + a_1^2 \hat{m}_i(\hat{\sigma}_{kl} d^c_{kl} + \hat{m}_k k_k d_{50}), \tag{14}$$

$$N_{ij} = a_1(\hat{\sigma}_{ij} + \hat{\sigma}^*_{ij}), \qquad N^c_i = a_1^2 a_c \hat{m}_i, \tag{15}$$

$$\overset{o}{m}_i = \overset{\bullet}{m}_i - 0.5 w_{ik} m_k + 0.5 m_k w_{ki}, \tag{16}$$

$$d^c_{ij} = d_{ij} + w_{ij} - w^c_{ij}, \qquad k_i = w^c_{,i}, \qquad w^c_{kk} = 0, \qquad w^c_{21} = -w^c_{12} = w^c, \tag{17}$$

wherein m_i – Cauchy couple stress vector, $\overset{o}{m}_i$ – Jaumann couple stress rate vector, d^c_{ij} – polar rate of deformation tensor, k_i – rate of curvature vector, w^c – rate of Cosserat rotation, d_{50} – mean grain diameter, a_c – micro-polar constant (a_c=a_1^{-1}) [10].

3.2 Nonlocal Hypoplasticity

A non-local approach is based on spatial averaging of tensor or scalar state variables in a certain neighbourhood of a given point (i.e. material response at a point depends both on the state of its neighbourhood and the state of the point itself). To obtain a regularisation effect for both the mesh size and mesh inclination, it is sufficient to treat non-locally only one internal constitutive variable (e.g. equivalent plastic strain in an elasto-plastic formulation [4] or measure of the deformation rate in a hypoplastic approach [11]) whereas the others can retain their local definitions. In the hypoplastic calculations, the non-local measure of the deformation rate $d = \sqrt{d_{kl} d_{kl}}$ in Eq.1 was treated non-locally:

$$d^*(x) = \frac{1}{A} \int_{-\infty}^{\infty} w(r)\, d(x+r)\, dV\,, \qquad w(r) = \frac{1}{l\sqrt{\pi}} e^{-(\frac{r}{l})^2}\,, \tag{18}$$

where r is the distance from the material point considered to other integration points of the entire material body, w is the weighting function (error density function) and A is the weighted volume. The parameter l denotes a characteristic length (it determines the size of the neighbourhood influencing the state at a given point).

3.3 Gradient Hypoplasticity

The gradient approach is based on the introduction of a characteristic length by incorporating higher order gradients of strain or state variables into the constitutive law [4]. By expanding e.g. the non-local measure of the deformation rate $d(x+r)$ in Eq. 18 into a Taylor series around the point r=0, choosing the error function w as the weighting function (Eq.19), cancelling odd derivative terms and neglecting the terms higher than the second order one can obtain the following expression (2D-problems):

$$d^*(x,y) = d + \frac{l^2}{4}\left(\frac{\partial^2 d}{\partial x^2} + \frac{\partial^2 d}{\partial y^2} + 2\frac{\partial^2 d}{\partial x \partial y}\right), \tag{19}$$

where l is a characteristic length. To evaluate the gradient term of the measure of the deformation rate d and to consider the effect of adjacent elements, a standard central difference scheme was used [12]:

$$(\nabla^2)^I d = \sum_n \left(\frac{\Delta^2}{\Delta x_n^2}\right)^I d = \sum_n \frac{\dfrac{d^{I+1} - d^I}{x_n^{I+1} - x_n^I} - \dfrac{d^I - d^{I-1}}{x_n^I - x_n^{I-1}}}{x_n^{i+1} - x_n^i}\,, \qquad n = x, y, z\,, \tag{20}$$

where "i" denotes a grid point and "I" a grid element.

4 FE-Results

The FE-calculations of plane strain compression tests were performed with a sand specimen which was h_o=10 cm high and b=2 cm wide (length l=1.0 m). As the initial stress state, the state with $\sigma_{22}=\sigma_c+\gamma_d x_2$ and $\sigma_{11}=\sigma_c$ was assumed in the sand specimen where σ_c denotes the confining pressure (σ_c=0.2 MPa), x_2 is the vertical coordinate measured from the top of the specimen, γ_d denotes the initial volume weight (σ_{11} - horizontal normal stress, σ_{22} - vertical normal stress).

A quasi-static deformation in sand was initiated through a constant vertical displacement increment prescribed at nodes along the upper edge of the specimen.. To preserve the stability of the specimen against the sliding along the bottom boundary, the node in the middle of the bottom was kept fixed.

To numerically obtain a shear zone inside of the specimen, a weaker element with a higher initial void ratio, e_0=0.90, was inserted in the middle of the left side.

4.1 Polar Hypoplasticity

Figs. 1 and 2 present the results of plane strain compression within polar continuum compared t. The normalized load-displacement curves with a different mean grain diameter (d_{50}=0 mm, 0.5 mm and 1.0 mm) in dense specimen (e_o=0.60, σ_c=0.2 MPa) are depicted in Fig.1. Fig.2 shows the deformed FE-meshes with the distribution of void ratio (the darker the region, the higher the void ratio).

The FE-results demonstrate that the larger the mean grain diameter, the higher the maximum vertical force on the top. The lower mean grain diameter, the larger the material softening (the behaviour of the material is more brittle). At the beginning, two shear zones are created expanding outward from the weakest element. Afterwards, and up to the end, only one shear zone dominates. The complete shear zone is already noticeable shortly after the peak. It is characterised both by a concentration of shear deformation and Cosserat rotation, and an increase of the void ratio. The thickness is about $t_{sz}\cong$6 mm=12×d_{50} (d_{50}=0.5 mm) and $t_{sz}\cong$10 mm=10×d_{50} (d_{50}=1.0 mm). An increase of the thickness of the shear zone with increasing d_{50} corresponds to a decrease of the rate of softening. The material becomes softer, and thus a larger deformation can develop. The calculated thickness of the shear zone in Karlsruhe sand (d_{50}=0.5 mm) is in accordance with experiments: t_{sz}=13×d_{50} [13] and 9×d_{50} [14].

4.2 Nonlocal Hypoplasticity

The results with a non-local measure of the deformation rate d^* using a different characteristic length l of Eq. 18 (l=0 mm, 0.5 mm, 1.0 mm and 2.0 mm) for dense sand (e_o=0.60, σ_c=0.2 MPa) are shown in Fig.3.

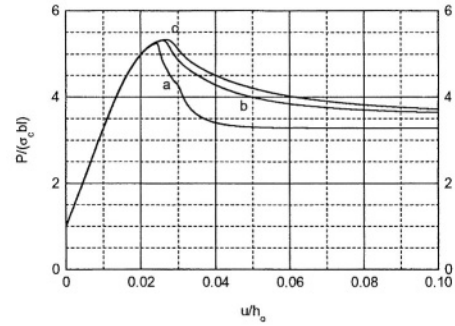


Fig. 1. Load-displacement curves (polar continuum): a) d_{50}=0.0 mm, b) d_{50}=0.5 mm, c) d_{50}=1.0 mm

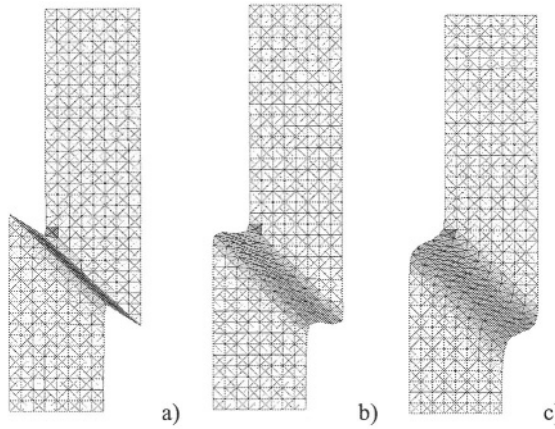


Fig. 2. Deformed FE-meshes with the distribution of void ratio in the residual state (polar continuum): a) d_{50}=0.0 mm, b) d_{50}=0.5 mm, c) d_{50}=1.0 mm

Similarly as in a polar continuum, the larger the characteristic length, the larger the maximum vertical force on the top and the smaller the material softening (the behaviour is more ductile). The vertical forces are almost the same as within a polar continuum. If the characteristic length is larger (l=2.0 mm), the shear zone does not appear. The thickness of the shear zone t_{sz} with l=0.5 mm is smaller than this with d_{50}=0.5 mm within a polar continuum. However, the thickness of the shear zone with l=1 mm is close to that within a polar continuum: $t_{sz} \cong 7$ mm=14×l=14×d_{50}. In general, the relationship between the non-local and polar characteristic length is $l \cong 2 \times d_{50}$ on the basis of the shear zone thickness.

4.3 Gradient Hypoplasticity

The results with a gradient measure of the deformation rate for dense sand (e_o=0.60, σ_c=0.2 MPa) are shown in Fig.4.

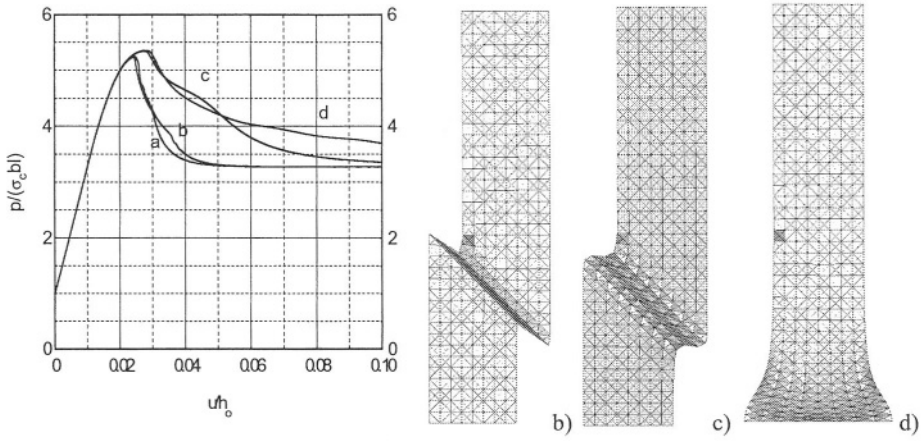


Fig. 3. Load-displacement curves and deformed FE-meshes with the distribution of void ratio in the residual state (non-local continuum): a) l=0 mm, b) l=0.5 mm, c) l=1.0 mm, d) l=2 mm

The evolution of the vertical force on the top is qualitatively similar as in the polar and non-local continuum. The thickness of the shear zone $t_{sz} \cong 7.3\ \text{mm} \cong 7 \times l$ (l=1.0 mm) is slightly larger than within a non-local continuum (l=1.0 mm) and a polar continuum (d_{50}=0.5 mm).

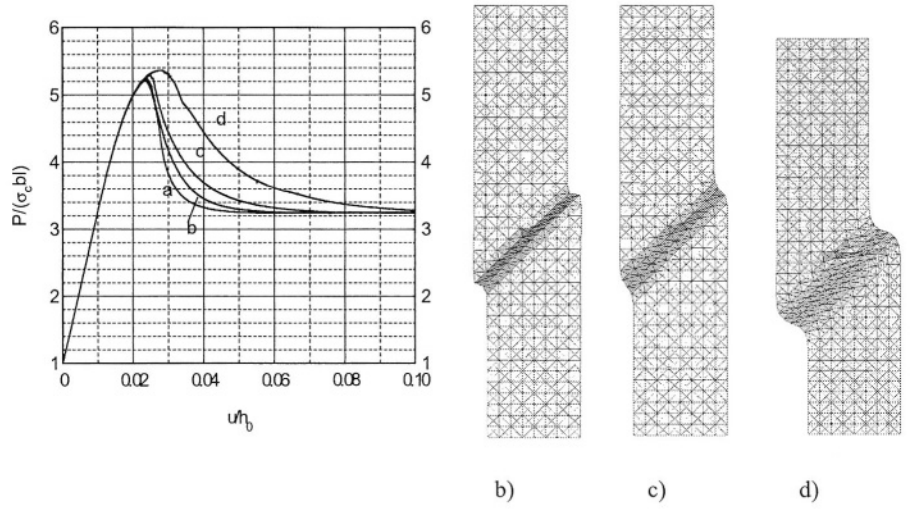


Fig. 4. Load-displacement curve and deformed FE-mesh with the distribution of void ratio in the residual state (gradient continuum): a) l=0 mm, b) l=0.5 mm, c) l=1 mm, d) l=2 mm

5 Conclusions

The results with a conventional hypoplastic constitutive model suffer from a mesh-dependency. The thickness of shear zones is severely mesh-dependent.

A polar, non-local and gradient hypoplastic model provide a full regularisation of the boundary value problem during plane strain compression. Numerical solutions converge to a finite size of the localization zone upon mesh refinement.

The thickness of the localized shear zone and the bearing capacity of the granular specimen increase with increasing characteristic length.

The characteristic length within a non-local and gradient theory can be related to the mean grain diameter on the basis of a basis of a back analysis of experiments.

References

1. de Borst, R., Mühlhaus, H.-B., Pamin, J., Sluys, L.: Computational modelling of localization of deformation. In: D. R. J. Owen, H. Onate, E. Hinton, editors, Proc. of the 3rd Int. Conf. Comp. Plasticity, Swansea, Pineridge Press (1992) 483-508
2. Tejchman, J., Wu, W.: Numerical study on shear band patterning in a Cosserat continuum. Acta Mechanica 99 (1993) 61-74
3. Bazant, Z., Lin, F., Pijaudier-Cabot, G.: Yield limit degradation: non-local continuum model with local strain, Proc. Int. Conf. Computational Plasticity, Barcelona. In: Owen, editor, (1987) 1757-1780
4. Zbib, H. M., Aifantis, E. C.: On the localisation and postlocalisation behaviour of plastic deformation. Res Mechanica 23 (1988) 261-277
5. Gudehus, G.: Comprehensive equation of state of granular materials. Soils and Foundations 36, 1 (1996) 1-12
6. Bauer, E.: Calibration of a comprehensive hypoplastic model for granular materials. Soils and Foundations 36, 1 (1996) 13-26
7. Herle, I., Gudehus, G.: Determination of parameters of a hypoplastic constitutive model from properties of grain assemblies, Mechanics of Cohesive-Frictional Materials 4, 5 (1999) 461-486
8. Oda, M.: Micro-fabric and couple stress in shear bands of granular materials. Powders and Grains. In: C. Thornton, editor, Rotterdam, Balkema (1993) 161-167.
9. Tejchman, J., Herle, I., Wehr, J.: FE-studies on the influence of initial void ratio, pressure level and mean grain diameter on shear localisation. Int. J. Num. Anal. Meth. Geomech. 23 (1999) 2045-2074
10. Tejchman, J.: Patterns of shear zones in granular materials within a polar hypoplastic continuum. Acta Mechanica 155,1-2 (2002) 71-95
11. Tejchman, J.: Comparative FE-studies of shear localizations in granular bodies within a polar and non-local hypoplasticity. Mechanics Research Communications 2004 (in print)
12. Alehossein, H., Korinets, A: Gradient dependent plasticity and the finite difference method. Bifurcation and Localisation Theory in Geomechanics. In: H.-B. Mühlhaus et all, editors, (2001) 117-125
13. Vardoulakis, I.: Scherfugenbildung in Sandkörpern als Verzweigungsproblem. Dissertation, Institute for Soil and Rock Mechanics, University of Karlsruhe 70 (1977).
14. Yoshida, Y., Tatsuoka, T., Siddiquee, M.: Shear banding in sands observed in plane strain compression. Localisation and Bifurcation Theory for Soils and Rocks, eds.: R. Chambon, J. Desrues and I. Vardoulakis, Balkema, Rotterdam (1994) 165-181

Effective Algorithm for Detection of a Collision between Spherical Particles

Jacek S. Leszczynski and Mariusz Ciesielski

Czestochowa University of Technology, Institute of Mathematics & Computer Science, ul. Dabrowskiego 73, 42-200 Czestochowa, Poland
{jale,cmariusz}@k2.pcz.czest.pl

Abstract. In this work we present a novel algorithm which detects contacts between spherical particles in 2D and 3D. We estimate efficiency of this algorithm throughout analysis of the execution time. We also compare our results with the Linked Cell Method.

1 Introduction

The dynamics of granular materials is characterised by particles which move under arbitrary extortion and interact with each other. Many properties of granular materials are still under investigations, especially convection, segregation, granular flows, ability to clusterisation, etc. Therefore computer simulations become an interesting tool which can develop physics and engineering groups. In discrete approaches, such as the molecular dynamics method and the event-driven method we need to detect particle collisions in order to add additional conditions which arise during the collision. Moreover collision detection has also many practical applications, e.g. in modelling of physical objects, in computer animations and in robotics.

The mechanism of collision detection involves time of calculations and mutual locations of contacting objects. Especially we need to detect the begin time of a contact which issues from the precise detection of two contacting points in the colliding objects. Each object is characterised by some shape and this is the main reason that to apply a convenient algorithm for the collision detection.

The general way of collision detection includes an information whichever the geometrical contact occurred. Above problem in the objective of study in papers [4,7,8] where different mechanisms of collision detection are investigated. The key aspect in the existing algorithms is how to detect the collisions in the fast way that to reduce the computational time.

In this paper we will focus on the two algorithms applied for the collision detection in 2D and 3D where particles have simple circular or spherical forms. The first algorithm called the Linked Cell Method [1,2,3,4,7,8] assumes space division into a regular lattice. Within the lattice one try to find particle collisions. We will present here a novel algorithm in order to reduce the computational time. The novel algorithm involves another way of collision detection in comparison

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 348–355, 2004.
© Springer-Verlag Berlin Heidelberg 2004

to the Linked Cell Method. In the next sections we will explain some details of the algorithm especially for spherical particles.

The spherical shape of the particles makes only the mathematical description easier but does not limit collision detection to be in any way less meaningful. The reader may find more informations in [4,7] concerning collision detection adapted to an arbitrary form of particle shapes. In other words, the detection of collision for arbitrary shapes of particles may decompose into the collision detection of particles having spherical shapes (the spherical shape is generated on the particle having the arbitrary shape) and within the spherical shapes one try to find collision for arbitrary shapes of the particles.

In this paper we will focus on the molecular dynamics method [5] where during motion of particles the contacts may eventually take place. This method takes into consideration an expression for the repulsive force acting between a pair of contacting particles. Within the molecular dynamics particles virtually overlap when a collision occurs. The overlap reflects the quantity important deformations of particle surfaces. Before the application of the repulsive force in motion equation we need to detect a contact between two particles. Therefore we introduce particles numbers by the index $i = 1, ..., np$ where np is assumed as the total number of considered particles. We also introduce the natural function $j(i) \neq i$, of a particle i in order to find the particle index in a set of particles. According to [5] we define the overlap of two spherical particles being in a contact as

$$\left\|\zeta_{j(i)}\right\| = r_{j(i)} + r_i - \left\|\mathbf{x}_{j(i)} - \mathbf{x}_i\right\| \tag{1}$$

where $r_i, r_{j(i)}$ are particle radiuses, $\|\cdot\|$ is the norm representing relative distance between the mass centre of particles. When a contact happens over time of calculations we need to consider $\left\|\zeta_{j(i)}\right\| = 0$.On the other hand, when $\left\|\zeta_{j(i)}\right\| \geq 0$ we detect also a contact between two particles.

Considering a system of particles np we take into account many arithmetic operations in order to check particle contacts. However, in order to check all possible contacts between a pair of particles one has to calculate $np(np - 1)/2$ arithmetic operations. This is very simple algorithm which is time consuming, especially when total number of considered particles np is large. Unfortunately, above algorithm is practically unacceptable. In computer simulations one uses algorithms which analyse lists of neighbouring particles or the lists of particle groups.

In this paper we will consider the Linked Cell Method as well as our novel algorithm for the collision detection. We shall present simulation results taking into account efficiencies of two methods.

2 Algorithms for Detection of Collisions

2.1 Linked Cell Method

The Linked Cell Method is very efficient [1,7,8] in practical simulations where the total number of particles is large. Within this method one divides the considered

area of space into a regular lattice where n_{cell} represents the total number of cells. For 2D problems we have $n_{cell} = m_1 \cdot m_2$ but for 3D problems we consider $n_{cell} = m_1 \cdot m_2 \cdot m_3$ where m_k is the division number of space in the direction k. Moreover we can calculate dimensions of an individual cell as $a_k = L_k/m_k$ where L_k are global dimensions of the considered space. We also take into consideration a relationship between the cell dimensions and particle dimensions as

$$\min_k(a_k) \geq d_{\max} \tag{2}$$

where $d_{\max}$ is maximal diameter of a particle in a set of considered particles *np*. The averaged particle number in each cell is $p_{avg} = np/n_{cell}$.

Implementation of the cell structures algorithm discuses in details in [1,8]. The first step of this implementation includes searching and sorting procedures of particle positions in order to find place in the appropriate cell. In the procedures the mass centre of particle defines the particle position in the cell. We call this step of calculations as grouping of particles into the cells structures. In this step two matrices are used. The first header matrix (which has the total number of elements n_{cell}) and the one-dimensional linked-list matrix (the dimension of this matrix is *np*) are necessary for storage of particle indexes in the appropriate cells. This way of storage is very convenient and optimises computer memory. Even we predict time t_{cell} in which one particle finds a place in the cell, the total time associated with all considered particles occuping place in cells is $np \cdot t_{cell}$. The second step of the Linked Cell Method involves detection of contacts for particles occuping space in neighbouring cells connecting with the considering cell. In this step we check all possible contacts in a simple way - every particle with every other ones. The number of possible contacts checken between several pairs of particles is determined as $4.5 \cdot np \cdot p_{avg}$ in 2D and $13.5 \cdot np \cdot p_{avg}$ in 3D - instead of the brute-force approach where we have $0.5 \cdot np \cdot (np - 1)$. Assuming time t_{test} used to check one pair of contacting particles we calculate time necessary for execution of the algorithm as $t_{sim} = np \cdot t_{cell} + 4.5 \cdot np \cdot p_{avg} \cdot t_{test}$ in 2D or $t_{sim} = np \cdot t_{cell} + 13.5 \cdot np \cdot p_{avg} \cdot t_{test}$ in 3D. The shortest value of execution time t_{sim} issues from the smallest value of the mean number of particles occuping space in one cell p_{avg}. However in prediction of the parameter p_{avg} we need to take into account expression (2) which has direct influence on the total number of cells n_{cell}.

2.2 New Method which Detects Particle Contacts

In this section we will propose a new algorithm which indicates particle contacts. We assume that particle shape has the spherical form. Fig. 1 shows same details of the algorithm with useful notations. For the contact calculations we need to input the following data: the total number of particles *np,* particle radiuses r_i and particle positions $\mathbf{x}_i$ of the mass centre, for $i = 1, \ldots, np$. First step of the algorithm basis on calculations of distances l_i between the begin of system coordinates $\mathbf{x}_0$ and the point of a particle which lies on the sphere at the particle. In solving this problem generally, the following algorithm is presented:

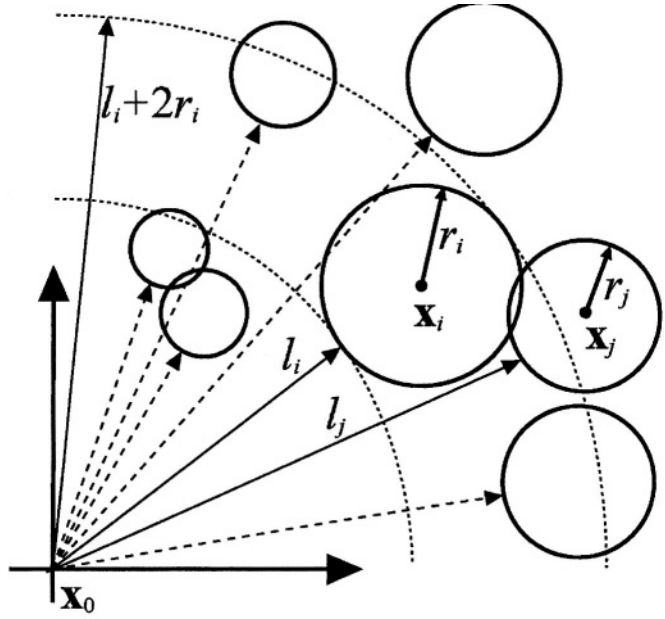


Fig. 1. Scheme illustrates our algorithm with useful notations.

Algorithm 1

Step 1. Introductory calculations:

- right choose of point $\mathbf{x}_0$ which establishes the begin of system coordinates (It means that all particle positions should have the same sign, positive or negative, in the system of coordinates.),
- calculations of the distances l_i in the form

$$l_i = \left\|\mathbf{x}_i - \mathbf{x}_0\right\| - r_i. \tag{3}$$

Step 2. The distances l_i are sorted by an arbitrary sorting algorithm, e.g. QuickSort [6]. (As a direct result of this sort we obtain a matrix *nl* in which several particle indexes are included.),

Step 3. Searching and detection of contacts:

```
for i = 1 to np − 1
  for j = i + 1 to np
    dist = l_{nl_j} − l_{nl_i}
    if 2 · r_{nl_i} < dist then
      break
    else
      ||ζ|| = r_{nl_i} + r_{nl_j} − ||x_{nl_j} − x_{nl_i}|| //overlap caclulations
      if ||ζ|| ≥ 0 then
```

```
        detect_a_contact(nl_i, nl_j) // contact between particles nl_i and nl_j
      endif
    endif
  end
end.
```

Presented algorithm requires to reserve memory in order to store two one-dimensional matrices where the each dimension equals to *np*. In this case we have a matrix of indexes *nl* and a matrix with the particle distances *l*. We estimate time t_l in which the one distance l_i is calculated and time t_{sort} necessary for the sorting procedure, and time t_{test} is a time where a pair of particle distances is checken. In t_{test} we have to calculate $\|\zeta_{j(i)}\|$ by formula (1). This is the arithmetical operation which is time consuming. Sumarising all times we obtain time t_{sim} which reflects same cost of the algorithm $t_{sim} = np \cdot t_l + t_{sort} + nt \cdot t_{test}$. In this expression *nt* indicates particle pairs which are eventually in the contact. The value *nt,* being depended on values r_i and L_k, decreases when particle concentration decreases in the considered volume. This happen when we consider the large volume and small diameters of particles.

The algorithm has the following advantages:

- it uses small amount of computer memory in order to storage data,
- it checks contacts locally being dependent on choose of the begin point $\mathbf{x}_0$ of system coordinates,
- structure of the computational code is very simple,
- it is independent on space dimension. (It means that we find contacts in the same way for both 2D and 3D problems. Note that at the preparation stage the norm of relative distance between the mass centre of particles differs in 2D in comparison to 3D.)

However, we need to take into account that efficiency of the algorithm represented by t_{sim}, decreases for dense packing of particles. This is disadvantage at the algorithm.

3 Simulation Results

On the base of previous section we perform computer simulations for detection of particle contacts. In this section we will try to compare results of simulation obtained form the Linked Cell Method and of course from our algorithm. The basic indicator for such a comparison is time necessary for detection of all contacts t_{sim} which is registered for above algorithms operating in the same initial date.

To prepare the initial data we generate particle diameter and particle position randomly. Nevertheless we assume a range of variation of the particle diameter. Therefore we prepare three the following test:

$$\begin{aligned} &A. \quad d \in < 0.05, 0.1 >, \\ &B. \quad d \in < 0.02, 0.05 >, \\ &C. \quad d \in < 0.005, 0.02 > . \end{aligned} \tag{4}$$

With regard to particle location we assume a rectangular box in 2D and a cubic box in 3D where the box dimension is nx. We generated randomly particle positions $\mathbf{x}_i$ within the box. The parameter nx is assumed to be $np/nx^2 = 100$ for 2D problems and $np/nx^3 = 1000$ for 3D problems. For presented initial data we calculate a ratio which is the number of contacts eventually happen to the number of particles occuping space in the box. Taking into account three assumptions presented by formula (4) we obtain several results of the ratio as A – 87%, B – 19%, C – 2% for 2D and A – 90%, B – 10%, C – 4% for 3D.

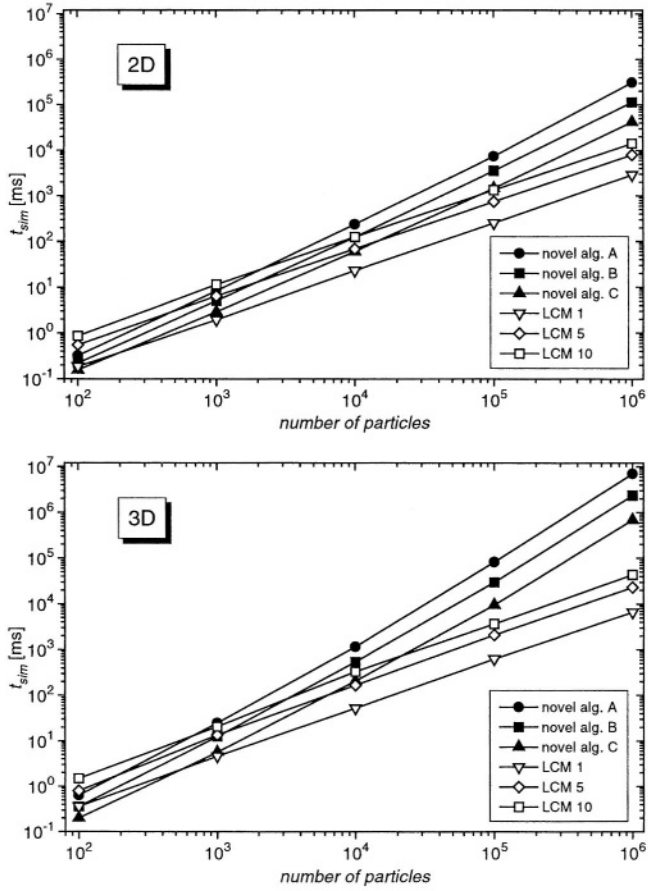


Fig. 2. Simulation results of the execution time t_{sim} over the number of considered particles for both 2D and 3D.

Fig. 2 shows the execution time t_{sim} consumed by the algorithms over the number of considered particles in the box. We taken into account this time

Table 1. Comparison of the execution time for 10000 particles and $d \in \langle 0.02, 1.0 \rangle$

	novel alg.	LCM100	LCM204	LCM625	LCM1111
t_{sim} [s]	1.627	1.141	2.296	8.828	12.332

Table 2. Comparison of the execution time for 100 particles and $d \in \langle 0.1, 2.0 \rangle$

	novel alg.	LCM11	LCM25	LCM100
t_{sim} [s]	0.37	1.01	1.4	1.59

Table 3. Comparison of the execution time for 100 particles with dimensions $d \in \langle 0.1, 0.3 \rangle$ and one big particle $d = 5.64$

	novel alg.	LCM101
t_{sim} [s]	0.197	1.38

calculated for both 2D and 3D problems. Open symbol connected with a line represent results obtained by our algorithm. For the Linked Cell Method we performed simulations being dependent on the averaged number of particles p_{avg} in one cell. In this case we calculated the number of lattice cells n_{cell} that to obtain the averaged number of particles in one cell as 1, 5 or 10 particles per one cell. Fig. 2 shows above results are representing by indicators LCM1, LCM5 and LCM10. We can observe that the number of particles occupying space in one cell is significant for the execution time in LCM. Nevertheless, we cannot observe this in our algorithm. The number of particles occupying space in one cell can increase in calculations when we consider a situation for particles differing more by their dimensions. This could happen for one big particle and a lot of small particles. To show this disadvantage in LCM we perform the following tests in 2D. Table 1 shows the execution time for 10000 particles differing by sizes $d \in \langle 0.02, 1.0 \rangle$. When the number of particles increases in one cell (LCM100, LCM204, LCM625, LCM1111) the execution time also increases in comparison to our algorithm, where this time does not change. Table 2 presents similar case to the previous one but particle dispersions is larger. It should be noted that greater dispersion in particle sizes influences significally to extension of the execution time calculated by LCM. The interesting case is also presented by Table 3. In this case we considered one cell with 100 small particles differing by sizes in the range of $d \in \langle 0.1, 0.3 \rangle$ and one big particle with dimension $d = 5.64$. We can observe that the execution time increases up to $1.38\,s$ for LCM in comparison to our algorithm where $t_{sim} = 0.197\,s$.

4 Conclusions

When we consider results presented by Fig.2, we can notice that the averaged number of particles p_{avg} in the Linked Cell Method has direct influence on the

execution time t_{sim}. For this method we decrease the value of the parameter p_{avg} the execution time t_{sim} is decreased too. For our algorithm we confirm that the execution time t_{sim} decreases when the averaged diameter of a particle in the particle distribution decreases too. Moreover in our algorithm we do not observe any influence of particle dispersions in the particle distribution on the execution time t_{sim}. However, some influence of this dispersion are noted in the Linked Cell Method. In direct comparison of the execution time being a result of above methods we state that the Linked Cell Method presents better results for dense number of particles in considered space. However the particles should not have large dispersion of diameters in the particle distribution. In opposite to previous method, our algorithm is more suitable and flexible (it generates smaller values of the execution time in comparison to the Linked Cell Method) when particle concentration in the considered volume is small and the dispersion of particle diameters in the distribution function may differ much more than in the Linked Cell Method.

Acknowledgment. This work was supported by the State Committee for Scientific Research (KBN) under the grant 4 T10B 049 25.

References

1. Allen M.P. and Tildesly D.J.: Computer Simulation of Liquids, Clarendon, Oxford, (1987)
2. Hockney R.W., Eastwood J.W.: Computer Simulation Using Particles, McGraw-Hill, New York (1981)
3. Iwai T., Hong C.W., Greil P.: Past particle pair detection algorithms for particle simulations, International Journal of Modern Physics C **10** (1999), pp. 823-837
4. Muth B., Müller M.-K., Eberhard P., and Luding S.: Collision detection and administration for many colliding bodies, submitted (2002)
5. Pourin L., Liebling Th.M.: Molecular-dynamics force models for better control of energy dissipation in numerical simulations of dense granular media, Physical Review E, **65** (2001) 011032 pp. 1-7
6. Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P.: Numerical Recipes in C/Fortran, Cambridge University Press, (1992)
7. Schinner A.: Fast algorithms for the simulation of polygonal particles, Granular Matter **2** (1999) 1, pp. 35-43
8. Tijskens E., Ramon H., de Baerdemaekerm J.: Discrete element modelling for process simulation in agriculture, Journal of Sound and Vibration **266** (2003), pp. 493-514

Vorticity Particle Method for Simulation of 3D Flow

Henryk Kudela and Paweł Regucki

Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27,
50-370 Wrocław, Poland
{henryk.kudela, pawel.regucki}@pwr.wroc.pl

Abstract. The vortex–in–cell method for three-dimensional, viscous flow was presented. A viscous splitting algorithm was used. Initially the Euler inviscid equation was solved. Following that, the viscous effect was taken into account by the solution of the diffusion equation. The diffusion equation was then solved by the particle strength exchange (PSE) method. Validation of the method was tested by simulation of the leap-frogging phenomenon for two vortex rings moving along a common axis of symmetry and the reconnection phenomenon of two colliding vortex rings for viscous flow.

1 Introduction

Interest in computational vortex methods stems from the fact that vorticity plays a fundamental role in all real fluid dynamics phenomena. Vortex particles introduced into the computation permit direct tracking of the vorticity and, additionally, allow for an analysis of the flow phenomena in terms of this vorticity. One can distinguish two different types of vortex methods, the direct method based on the Biot-Savart law where the velocity of each vortex particle is calculated by summing up the contribution of all particles in the domain, and the vortex–in–cell (VIC) method where the velocity is obtained on grid nodes by solving Poisson equations for a vector potential. After that, we differentiate it using the finite difference method, and interpolate the value of the velocity to the position of the vortex particles. Despite the development of fast summation algorithms, VIC methods are still several orders faster than direct methods [1, 4]. In the literature one finds that VIC calculations relate mainly to 2D flow, whereas the extension to 3D flow still requires further investigation.

In this work we validate a 3D VIC method using examples of vortex ring dynamics: the leap-frogging of two vortex rings, and the reconnection of two colliding vortex rings. The vortex rings are the simplest 3D vortex structures that can be utilized easily in the laboratory. They are observable in the real turbulent flow. The interaction of two vortex rings gives an interesting and good example of non-linear interaction of the regions with concentrated vorticity and it may serve as a clue to understanding the nature of turbulence.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 356–363, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Equations of Motion and Description of the Vortex–in–Cell Method

The equations that describe the evolution of the vorticity field in the three dimensional, incompressible, viscous flow are [2]:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\,\boldsymbol{\omega} \;=\; (\boldsymbol{\omega} \cdot \nabla)\,\boldsymbol{u} + \nu\,\Delta\boldsymbol{\omega} \;, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} \;=\; 0 \tag{2}$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ is the vorticity vector, $\boldsymbol{u} = (u_1, u_2, u_3)$ is the velocity and ν – kinematic viscosity of the fluid. The condition of incompressibility (2) assures the existence of vector potential $\boldsymbol{A}$ [8]:

$$\boldsymbol{u} \;=\; \nabla \times \boldsymbol{A} \tag{3}$$

where the components of vector potential $\boldsymbol{A}$ are obtained by the solution of the Poisson equations (it is assumed additionally that $\nabla \cdot \boldsymbol{A} = 0$):

$$\Delta A_i \;=\; -\,\omega_i \;, \quad i = 1, 2, 3 \;. \tag{4}$$

In the vortex–in–cell method, (actually we should speak about the vorticity–in–cell method) the continuous field of vorticity is replaced by a discrete distribution of Dirac delta measures [2,10]:

$$\boldsymbol{\omega}(\boldsymbol{x}) \;=\; \sum_{p=1}^{N} \boldsymbol{\alpha}_p(\boldsymbol{x}_p)\;\delta(\boldsymbol{x} - \boldsymbol{x}_p) \tag{5}$$

where $\boldsymbol{\alpha}_p$ means vorticity particle $\boldsymbol{\alpha}_p = (\alpha_{p1}, \alpha_{p2}, \alpha_{p3})$ at position $\boldsymbol{x}_p = (x_{p1}, x_{p2}, x_{p3})$. The domain of the flow is covered by the numerical mesh $(N_x \times N_y \times N_z)$ with equidistant spacing h, and the i-component of the vector particle $\boldsymbol{\alpha}_p$ is defined by the expression:

$$\alpha_i \;=\; \int\limits_{V_p} \omega_i(x_1, x_2, x_3)\,d\boldsymbol{x} \approx h^3\,\omega_i(\boldsymbol{x}_p) \;, \quad \boldsymbol{x}_p \in V_p \;, \quad |V_p| = h^3 \;. \tag{6}$$

In our method, the modelling of the viscous, incompressible flow is based on the viscous splitting algorithm:

- At first the Euler equations of motion for the inviscid flow are solved. From the Helmholtz theorem we know that the vorticity is carried on by the fluid:

$$\frac{d\,\boldsymbol{x}_p}{d\,t} = \boldsymbol{u}(\boldsymbol{x}_p,\, t) \;, \tag{7}$$

$$\frac{d\,\boldsymbol{\alpha}_p}{d\,t} = [\nabla\,\boldsymbol{u}(\boldsymbol{x}_p,\, t)] \cdot \boldsymbol{\alpha}_p \;. \tag{8}$$

The right side of (8) may be expressed by virtue of the vector identity as $(\boldsymbol{\omega}\cdot\nabla)\boldsymbol{u} = [\nabla\boldsymbol{u}]\cdot\boldsymbol{\omega} = [\nabla\boldsymbol{u}]^T\cdot\boldsymbol{\omega} = 0.5([\nabla\boldsymbol{u}]+[\nabla\boldsymbol{u}]^T)\cdot\boldsymbol{\omega}$ [2]. We use the term $([\nabla\boldsymbol{u}]^T\cdot\boldsymbol{\omega})$ because it better preserves the invariants of the motion for the inviscid flow [2]. Velocity and stretching are calculated on the grid nodes by the finite difference method and after that they are interpolated to the position of the particles.

- Next, the strength of the vector particles $\boldsymbol{\alpha}_p$, due to the viscosity, is updated:

$$\frac{d\boldsymbol{\alpha}_p}{dt} = \nu\,\Delta\boldsymbol{\alpha}(\boldsymbol{x}_p)\ . \tag{9}$$

The Laplacian on the right side of (9) according to the PSE method is replaced by the integral operator, and the equation takes the form:

$$\frac{d\boldsymbol{\alpha}_p}{dt} = \nu\,\sigma^{-2}\sum_{q=1}^{N}(\boldsymbol{\alpha}_q-\boldsymbol{\alpha}_p)\,\eta\left(\frac{\boldsymbol{x}_q-\boldsymbol{x}_p}{\sigma}\right)\ . \tag{10}$$

The kernel η must satisfy the moment properties [2]:

$$\int x_i\,x_j\,\eta(\boldsymbol{x})\,d\boldsymbol{x} = 2\,\delta_{ij} \quad \text{for } i,j = 1,2,3\ , \tag{11}$$

$$\int x_i^{i_1}\,x_j^{i_2}\,\eta(\boldsymbol{x})\,d\boldsymbol{x} = 0 \quad \text{if } i_1+i_2 = 1 \text{ or } 3 \le i_1+i_2 \le r+1\ , \tag{12}$$

$$\int |\boldsymbol{x}|^{r+2}\,|\eta(\boldsymbol{x})|\,d\boldsymbol{x} < \infty\ . \tag{13}$$

As a kernel $\eta(\boldsymbol{x})$ we took the function [3]:

$$\eta(\boldsymbol{x}) = \begin{cases} \dfrac{\mathcal{C}}{1+|\boldsymbol{x}|^2} & \text{for } |\boldsymbol{x}| \le 2\ , \\ 0 & \text{for } |\boldsymbol{x}| > 2 \end{cases} \tag{14}$$

where $\mathcal{C} = 0.269$ was calculated by us in order to satisfy the condition (11). That guarantees that the PSE method is second order. In calculating we took $\sigma = 1.5\,h$. Such a choice of σ we made on the basis of a numerical study concerning the dissipation rate of kinetic energy. Equation (10) is solved using the second order Euler scheme.

To solve (4) on the numerical mesh the strength of particles $\boldsymbol{\alpha}_p$ has to be redistributed on the mesh nodes (l,m,n):

$$\omega_i(x_l,x_m,x_n) = \frac{1}{h^3}\sum_p \alpha_i(\boldsymbol{x}_p)\,\varphi_{lmn}(\boldsymbol{x}_p)\ , \quad i = 1,2,3 \tag{15}$$

where for φ we used the B-spline of the third order $\varphi_{lmn}(\boldsymbol{x}) = \varphi_l(x)\,\varphi_m(y)\,\varphi_n(z)$ and the one-dimensional B-spline has the form:

$$\varphi(x) = \begin{cases} \frac{1}{2}|x|^3 - x^2 + \frac{2}{3} & \text{for } |x| < 1 \ , \\ -\frac{1}{6}|x|^3 + x^2 - 2|x| + \frac{4}{3} & \text{for } 1 \le |x| \le 2 \ , \\ 0 & \text{for } |x| > 2 \ . \end{cases} \tag{16}$$

The equations (4) are solved by the fast Poisson solver with the periodic boundary conditions.

Summarizing, calculation at one time step $t^n = n\,\Delta t$ goes as follows:
1) redistribution of the particle masses on the grid nodes (15),
2) solution of the Poisson equations (4) and calculations of the velocity on the grid nodes by virtue of (3),
3) interpolation of the velocities from the grid nodes to the particle positions by the second order Lagrange interpolation, advancing in time the positions of particles using the fourth order Runge–Kutta method (7) and updating the strength of the particles (8),
4) in the new positions, the strength of the particles is updated due to the viscosity (10).
This completes the one time step.

3 Numerical Results

As a computational domain for our experiments we chose a cube $10 \times 10 \times 10$ covered by a rectangular grid with equidistant spacing $h = 0.1$ in each direction (Fig. 1A). The time step was $\Delta t = 0.02$. The single vortex ring was divided into 100 slices (Fig. 1B) and in each slice the vorticity was redistributed on 121 particles (Fig. 1C). Finally, one ring was approximated by the set of 12,100 vector vorticity particles.

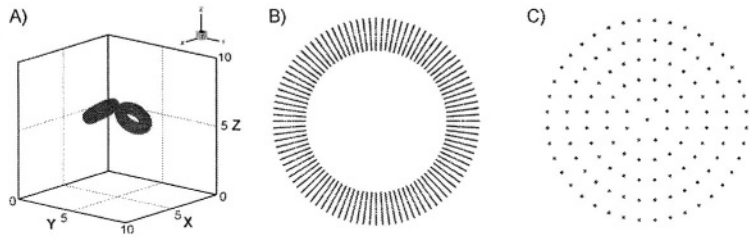


Fig. 1. (*A*) Computational domain with two vortex rings; (*B*) division of the vortex ring into 100 slices; (*C*) initial position of the 121 particles at a single slice

At first we tried to reproduce the leap-frogging ("vortex game") phenomenon [7,11,14]. When two co-axial vortex rings are made to travel in the same direction the velocity field induced by the second ring will cause the first ring to contract and accelerate. At the same time the velocity field induced by the first ring causes an expansion in diameter and a slowing down of the second ring. In effect the first ring is drawn through the center of the second ring and emerges in front of it. When this occurs the roles of the rings are reversed and the process may repeat itself. The numerical results of the simulation of the "vortex game" is presented in Fig. 2. We used two identical vortex rings with a uniform vorticity distribution inside the cores. Their parameters were: radius of the rings $R = 1.5$, radius of the cores $\varepsilon = 0.15$, circulation $\Gamma = 1.0$ and the positions of the ring centers: (5.0,3.4,5.0), (5.0,4.3,5.0).

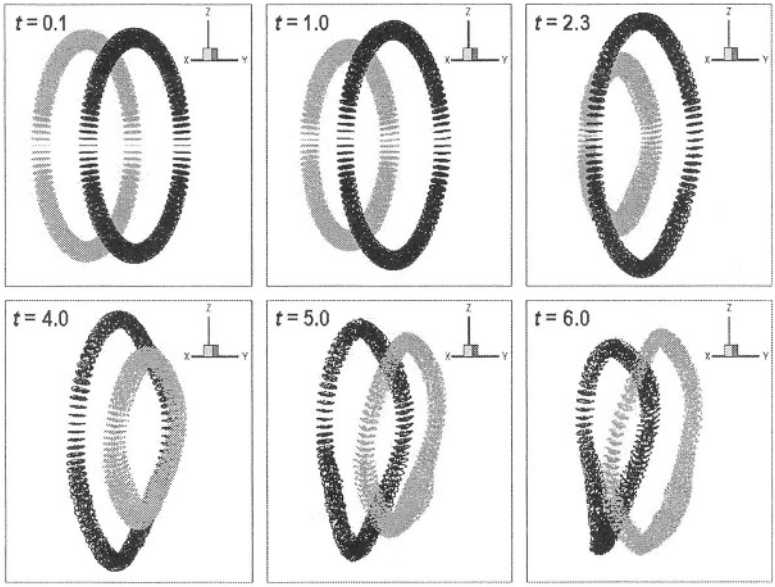


Fig. 2. The sequence of the time position of the vortex particles for the leap-frogging phenomenon in the inviscid flow

In this case we assume that the flow is inviscid. It is known that the "vortex game" is relatively difficult to repeat experimentally [7]. This is due to the fact that it is very sensitive in regard to the initial positions of the rings and their parameters. In Fig. 3 we presented the "vortex game" starting from different initial parameters: radiuses of the rings $R_1 = 1.0$, $R_2 = 1.5$; radiuses of the cores $\varepsilon_1 = 0.3$, $\varepsilon_2 = 0.25$; circulations $\Gamma_1 = 1.5$, $\Gamma_2 = 1.0$; the positions: (5.0,3.5,5.0), (5.0,4.0,5.0). In effect during their evolution the tail structure emerged. It is the most typical situation observed in the experiments [14].

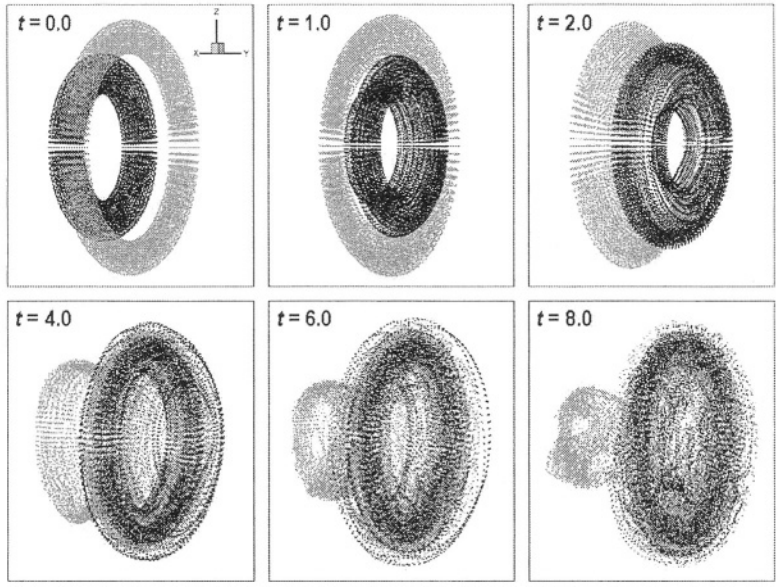


Fig. 3. The sequence of the time position of the vortex particles for the motion of the two rings the "tail structure" in the inviscid flow

Next we studied the reconnection of two vortex rings in viscous flow. It is an intriguing phenomenon (also called cut-and-connect or crosslinking) involving collision of two vortex rings that leads to the changes in connectivity and topology of the vortex rings. It has been extensively studied numerically and experimentally [5,6,13]. We used two identical vortex rings with a uniform vorticity distribution inside the cores. Their parameters were: radius of the rings $R = 1.0$, radius of the cores $\varepsilon = 0.4$, circulations $|\Gamma| = 1.0$ and the positions: (5.0, 3.5, 6.0), (5.0, 6.5, 6.0). The initial inclination of the rings to the vertical axis was 54 °. Kinematic viscosity was $\nu = 0.001$. The sequence of the time position of the vortex particles for the reconnection phenomenon was presented at Fig. 4. In the top diagrams, two rings collided and, in the effect of the first reconnection, formed one elongated ring ($t = 15.0$). Further evolution of this ring led to a second reconnection and in the end there were again two rings connected by the thin filaments (bottom diagrams of Fig. 4). The time evolution of $|\omega|$ iso-surfaces during the reconnection phenomenon was shown at Fig. 5. For better viewing, the bottom diagrams show the vorticity field from a different point of view than the top ones do. In the two top diagrams ($t = 4.0$, $t = 10.0$), a part of vorticity was removed from one ring in order to better show the contact zone. The final diagram clearly shows two vortex rings connected by two thin vortex structures, in literature called "threads" [6,9]. The presented sequence of the vortex rings reconnection process is in good qualitative agreement with the experiment [13].

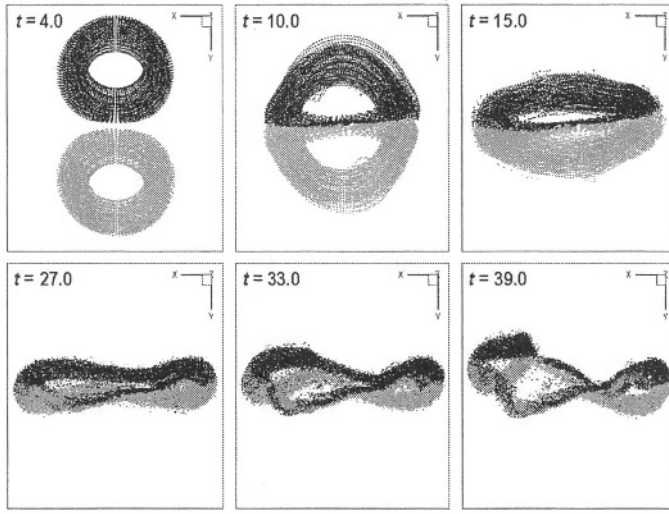


Fig. 4. The sequence of the time position of the vortex particles for the reconnection phenomenon (*view from the top*)

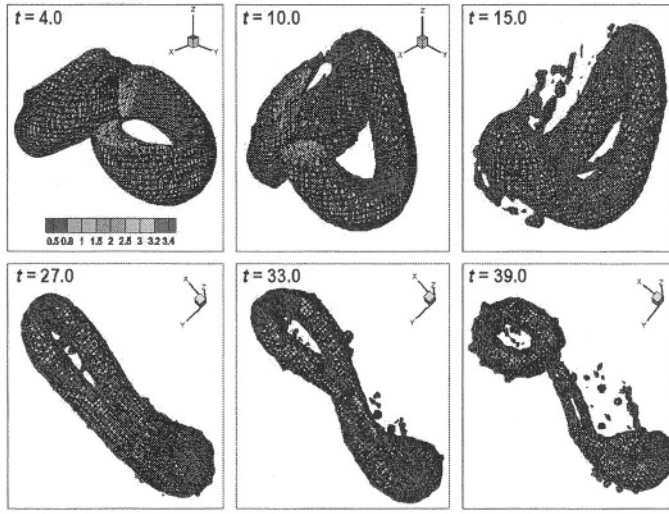


Fig. 5. Time evolution of $|\omega|$ surface during the reconnection phenomenon of two vortex rings. For better viewing, the bottom diagrams show the vorticity field from a different direction than the top ones do

4 Closing Remarks

The presented results indicate that the vorticity particle method is very attractive for studying vortex dynamics phenomena. Compared to the direct vortex method, the vorticity particle method is several orders faster [1,4]. In the near future we intend to include in the algorithm a solid boundary with a no-slip condition, and create a general-purpose program for simulating viscous flow in 3D.

Acknowledgment. This work was supported by The State Committee for Scientific Research under KBN Grant No. 4 T10B 050 25.

References

1. Cottet, G.-H.: 3D Vortex Methods: Achievements and Challenges. In: Vortex Methods, Selected Papers of the First International Conference on Vortex Methods Kobe Japan 1999, ed. K. Kamemoto and M. Tsutahara, World Scientific, (2000) 123–134
2. Cottet, G.-H., Koumoutsakos, P.: Vortex Methods: Theory and Practice. Cambridge University Press, New York (2000)
3. Cottet, G.-H., Michaux, B., Ossia, S., VanderLinden, G.: Comparision of Spectral and Vortex Methods in Three-Dimensional Incompressible Flows. J. Comput. Phys. **175** (2002) 702–712
4. Cottet, G.-H., Poncet, Ph.: Particle Methods for Direct Numerical Simulations of Three-Dimensional Wakes. J. Turbulence. **3**(38) (2002) 1–9
5. Kida, S., Takaoka, M.: Vortex Reconnection. Annu. Rev. Fluid Mech. **26** (1994) 169–189
6. Kida, S., Takaoka, M., Hussain, F.: Collision of Two Vortex Rings. J. Fluid Mech. **230** (1991) 583–646
7. Lim, T.T., Nickels, T.B.: Vortex Rings. In: Fluid Vortices, ed. Sh.I. Green, Kluwer Academic Publishers, Dordrecht (1996) 95–153
8. Marshall, J.S.: Inviscid Incompressible Flow. John Wiley & Sons, Inc., New York (2001)
9. Melander, M.V., Hussain, F.: Reconnection of Two Antiparallel Vortex Tubes: a New Cascade Mechanism. In: Turbulent shear flows 7, Springer–Verlag, Berlin (1991) 9–16
10. Kudela, H., Regucki, P.: The Vortex–in–Cell Method for the Study of Three-Dinemsional Vortex Structures. In: Tubes, Sheets and Singularities in Fluid Dynamics, Vol. 7 of Fluid Mechanics and Its Applications, Kluwer Academic Publisher, Dordrecht (2002) 49–54
11. Oshima, Y., Kambe, T., Asaka, S.: Interaction of Two Vortex Rings Moving along a Common Axis of Symmetry. J. Phys. Soc. Japan. **38**(4) (1975) 1159–1166
12. Regucki, P.: Modelling of Three Dimensional Flows by Vortex Methods. Phd Thesis (in polish). Wrocław University of Technology, Wrocław, Poland (2003)
13. Schatzle, P.: An Experimental Study of Fusion of Vortex Rings. Phd Thesis. California Institute of Technology, USA (1987)
14. Yamada, H., Matsui, T.: Mutual Slip-Through of a Pair of Vortex Rings. Phys. Fluids. **22**(7) (1979) 1245–1249

Crack Analysis in Single Plate Stressing of Particle Compounds

Manoj Khanal, Wolfgang Schubert, and Jürgen Tomas

Mechanical Process Engineering, Process Engineering Department,
Otto-von-Guericke-University of Magdeburg, D-39106 Magdeburg, Germany
{Manoj.Khanal, Wolfgang.Schubert, Jürgen.Tomas} @vst.uni-magdeburg.de

Abstract. Particle compound material is the composition of different particles with inhomogeneous and non-uniform properties. Particle compound material is the most complicated engineering material whose properties vary according to the applications, method of manufacturing and ratio of its ingredients. The quasi-homogeneous materials like building materials and constituents of tablets, pellets etc are some of the examples of particle compounds.
The 2 Dimensional Finite Element Analysis was done with single plate compressive stressing on the particle compound material to have the idea of the stress distributions during stressing. Then the method was followed by the Discrete Element Method (DEM) for further analysis.
The study of crack propagating mechanism in particle compound was represented by a model material which resembles the high strength materials, pressed agglomerates and more. The paper analyses the cracking of the particle compounds, here concrete ball, with respect to continuum and discrete approach.

1 Introduction

Particle compound material [1] is the composition of different particles with inhomogeneous and non-uniform properties. The research with the spherical shaped particle compound, here concrete sphere, was to observe the cracking mechanisms while subjected to single plate stressing along the diameter during diametrical compression test. The aggregate material as the value component is fixedly embedded in concrete so that cracking can only occur by forced crushing. During this process the bonds between aggregate and binding material, which is the second but valueless component, have to be burst. The compression stressing test, termed as indirect tensile test, can be considered as one of the method to evaluate the fracture strength of particle compounds.

There are number of references by Schubert [2], Kiss and Schönert [3], Arbiter et al. [4], Tomas et al. [5], Potapov and Campbell [6], Mishra et al. [7], Salman et al. [8], Moreno et al. [9] and Khanal et al. [10] which have shown the study of impact and compression stressing for the fracturing of materials and cracking of material during normal and oblique impact conditions. Khanal et al. [10] have studied the central and oblique impact of particle compound materials with finite and discrete element methods and compared with experiments of central impact condition. The discrete

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 364–371, 2004.
© Springer-Verlag Berlin Heidelberg 2004

element simulations of diametrical compression tests were carried out [11,12,13]. Thornton et al. [11] have studied the diametrical compression simulation on dense and loosely packed agglomerates and showed that the loose agglomerates failed by disintegration and dense agglomerate by fractured. For the low strength materials like tablets, the simulations/measurements are generally performed with comparatively low velocity for slow deformation, quasi-static, where the test specimen fails by propagation of one or few cracks at a failure stress that is relatively insensitive to the loading rate [13] (see[11]). The experiment and analysis of fragmentation of brittle spheres under static and dynamic compressions were dealt by Chau et al. [14]. The finite element method has also been adopted to study the crushing of spherical materials [15,16].

The cracking method in spherical model is different than the regular structures when subjected to impact or diametrical stressing loading. To study the cracking mechanism in the concrete ball, the two-dimensional Finite Element compressive analyses were carried out for single plate diametrical stressing condition to understand the stress pattern distributions before cracking. In reality, the discrete element approach is preferable for the analysis of the particle compound, particularly to understand the fragmentation behaviour during fracturing, which treats the particle compound as the constituents of different individual small balls as the particles. The compression stressing on concrete sphere is done with relatively higher speed with higher deformation rate as compared to the agglomerates.

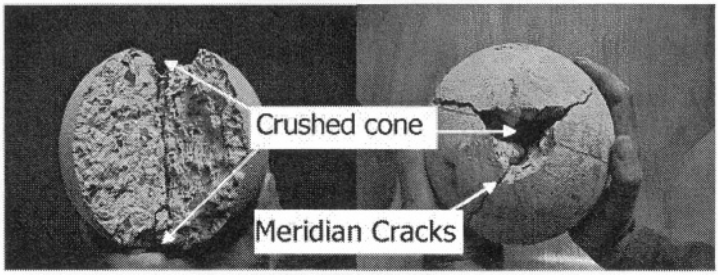


Fig. 1. Cracks observed during single plate stressing of concrete sphere.

2 Results and Discussions

2.1 Experimental

Spherical shaped concrete spheres of 150 mm in diameter of B35 (35 N/mm^2 compressive strength) strength category as common material of civil engineering were chosen for the single plate compression stressing. The compression stressing experiments were done with universal testing machine. The concrete sphere was clamped on the lower base plate and the upper plate was moving with the defined velocity to generate the stress inside the sphere. The concrete sphere was coloured with different colours before crushing. After crushing, the generated fragments were collected, assembled according to the colour and analyzed to observe the crack

patterns. The fractured ball obtained from experiment is shown in Fig.1 with meridian cracks and crushed cone.

2.2 Finite Element Simulation of Compression Test by Single Plate Stressing

To study the dynamic stress patterns generated inside the concrete ball during single plate diametrical stressing, the finite element simulation was carried out with commercially available software called ANSYS [17]. Though the continuum analysis does not allow to visualize the high velocity multiple crack propagation and fragmentation behaviour during simulation, but it provides the idea of stress distributions inside the sphere during stressing. It is known that crack generation and propagation are functions of stress [1]. Hence, the finite element analysis can be considered as an essential investigation. The finite element model description with input parameters is dealt in another paper [1]. An ANSYS model was considered for a two-dimensional sphere of 150 mm in diameter, stressing at the velocity of 20 m/s. Actually, different velocities were tried and it was noted that the change in velocities causes the change in intensity of stresses. Hence, for the clarity here the simulation of only one velocity is presented.

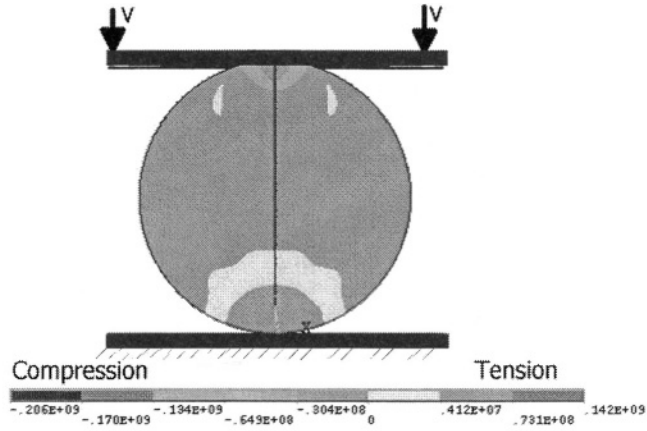


Fig. 2. Stress generated in stressing direction during compression test.

Fig 2. shows the stress generated during single plate diametrical stressing. The upper plate is stressing the ball and lower plate is constraint from any movement. When the plate stresses the ball, the stress wave is initiated and propagated from the contact zone and moves towards the other low stressed zone (opposite side). These different stressed zones show distinct region nearer to the contact zone and is similar to the cone in three dimension or wedge (triangle) in two dimension. At the contact area the compressive zones can be seen as a triangular shape. The tension waves are generated from the boundary of the contact zone and can be seen in figure as two dynamic tension spots nearer to the contact zone. The propagated tension waves can also be seen on the fixed plate side as an arch shape. The boundary of the conical or wedge shape has the maximum absolute change in values from compression to tension. This transition from the compressive to tensile region takes place in the

periphery of the compressed zone. Hence, this region has a dominant effect on the crack generation, and so, the boundary of this region initiates the crack with a similar shape during stressing. In single plate stressing, though the stressing is done in one direction, the other direction has also the similar disturbed region like the shape of the wedge. This is because of the propagation and reflection of the waves in to and fro direction. This implies even in single plate compression, there should be the formation of the another wedge in opposite side (i.e. in the stationary plate side). The dimension of this second wedge depends upon the stressing condition and velocity. The generation of two wedges can be further verified by discrete element simulations. The propagating tension waves from stressing to non-stressing sides should also generate the cracks in the similar propagating manner from one side to another. These types of cracks are called meridian cracks in experiments and diametrical cracks in 2 dimensional simulations.

2.3 Discrete Element Simulation of Compression Test by Single Plate Stressing

The concrete sphere is considered as the mixtures of different sized particles having random properties with porosities. Though the material seems complicated, it can be easily modelled with the principle of discrete element method. The discrete element solution scheme assumes each of its constituents as the separate entity and applies the Newton's second law of motion and force displacement law at contacts. This individual particles allow to delete the contact when they experiences the higher force than their material strength. This process shows the fragmentation behaviour of the material, like in reality. The two dimensional concrete ball was modelled with randomly arranged 1000 particles as distinct elements with the PFC [18] which solves with distinct element method. The detailed modelling of concrete ball and input parameters are dealt in [1]. Left picture of Fig 3. shows the assembly of gravels (aggregates) and hardened cement paste to represent the concrete ball. Thus modelled concrete ball was diametrically compressed with single plate stressing. In single plate stressing, the upper plate was allowed to move against the ball and the lower plate was fixed.

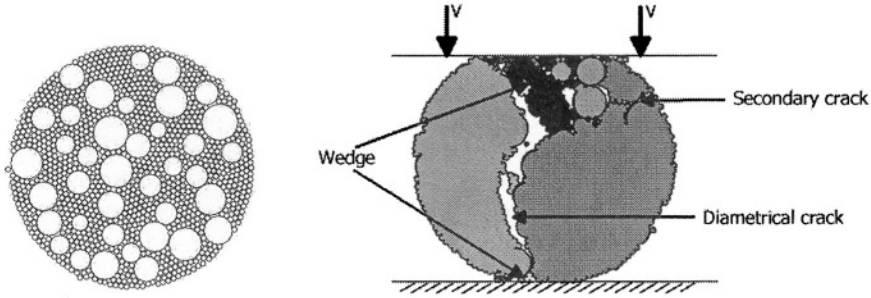


Fig. 3. Left – modelled concrete ball with DEM (bigger balls – aggregates, smaller balls – hardened cement paste), Right – Fractured ball during compression test by single plate stressing.

Right picture of Fig. 3 shows the fractured ball during single plate compression at the plate velocity of 1 m/s. The different gray tone pieces show the fragments

generated after crushing. Here, two disturbed regions are generated in two sides of the ball – one on the stressing side and another on the fixed plate side. In the stressing side, the crushed region has the similar wedge shape as predicted by continuum analysis. The crushed wedge in the stationary plate side is small as compared to the case of stressing side. The stress waves generated during stressing are somewhat damped on the boundary of the particles and on the discontinuous region of the material and only the less amount of the stress waves propagate and reach in the lower side of the ball. Hence, this causes the less disturbances as compared to the stressing side, as a result, the smaller wedge is produced in the opposite side of the stressing. The figure shows the different types of cracks obtained during simulation. The diametrical cracks are propagated from stressed side to the low stressed side. The secondary cracks are linking the diametrical cracks. The bigger balls on the crushed zone show the liberated aggregates during stressing.

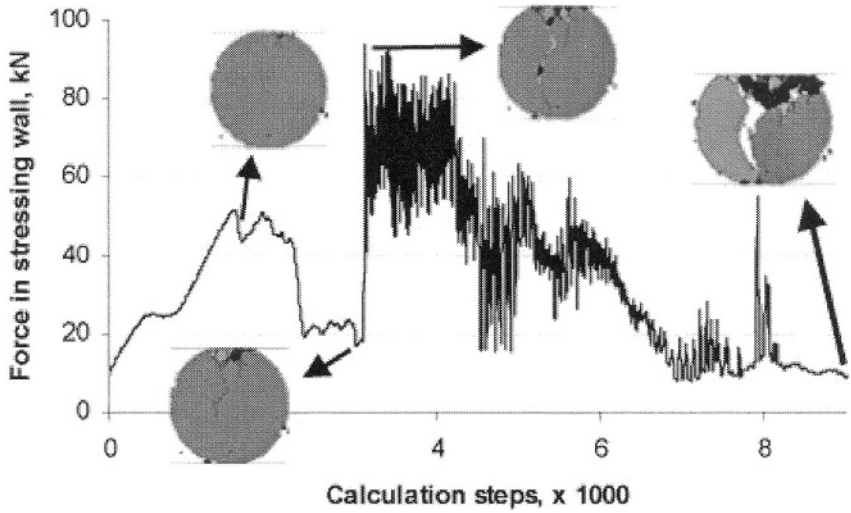


Fig. 4. Different stages of fractured ball during single plate compression stressing, v=1m/s.

The different stages during compression test by single plate stressing of two dimensional concrete sphere with calculation steps are shown in Fig. 4. The first peak was observed when there was a starting of the crack. The fall in force curve after the first crack shows that the energy was utilized to generate the crack. The generation of crack means the generation of new surfaces and consumes energy. It is clear from the figure that before the generation of cracks, the force curves are increasing and after propagation of the cracks these force curves are decreasing. This shows that energy is required to generate and propagate cracks. In other words, crack propagation is the generation of new surfaces. It is seen from the figure that even after failure of the ball the force curves attain the higher values. The reason for this is, when wall touches the aggregates, the wall experiences the higher amount of resistance from the aggregates (bigger balls). In other cases, the disturbances caused by the stressing of wall will be adjusted by the discontinuous nature of the material. Which means, the stress waves

propagating inside the ball from the stressing condition will be damped by the boundaries of the particles and the pores present in-between the constituents of the concrete spheres.

2.4 Comparison of Experiments and Simulations

Fractured ball from experiment and simulations showed the crushed cone (wedge in 2D simulation) during stressing. In the experimentally fractured ball, the inelastic contact deformation produces the cone of fines [5]. The contact area was noted very smooth, all the micro irregularities were smoothened. There were no secondary cracks generated during experiments. This is because it was not allowed to crush further after the breaking of the ball. This provides the evidence that the secondary cracks are generated only after forming the meridian cracks. The experiments were conducted with the velocity of 1e-6 m/s, which is obviously very less stressing velocity to generate the secondary cracks. This experiment was done with Universal Testing Machine (UTM), where it was impossible to provide higher velocities. Even if the stressing plate was allowed to stress further, the plate motion would not be sufficient to hold the ball from falling down from the lower plate to crush it because immediately after fracturing the ball, the ball crushed into pieces and fell out from the experimental rig due to gravity. But in discrete element simulation it was possible to farther the stressing wall against the ball. Hence, we can see the secondary cracks in the fractured ball. This also adds to the proof that the secondary cracks are generated only if there is sufficient energy remaining after forming the meridian cracks. Here, it has to be noted that the experiments are 3 dimensional whereas the simulations are 2 dimensional, that is why it is not possible to relate point to point between them.

2.5 New Surface Generation

It is known that during stressing of the sphere, the fracturing process produces the fragments and the quantity of fragments depends upon the stressing condition. Here, the stressing condition imply the stressing velocity and the material strength of the model. In discrete element method all the constituents are glued together with the bonds existing between the particles. Therefore, when these bonds break the particles are separated out. In other words, the breaking of bonds mean the generation of the new surfaces. Fig.5 shows how the bonds are breaking with the increasing velocity. From figure, it can be observed that with the increasing velocity the bond breakage between the particles are also increasing. Actually, the broken bonds are nothing it is similar to producing the cracks. When the material is not able to sustain the strength generated by stressing, the material deletes the contact with each other and propagates the crack. For the less stressing energy less bonds between the particles are broken and with higher velocities more bonds are broken and after certain limit of the stressing velocity no further more bonds are broken. Hence, higher than this limiting velocity there is no production of broken bonds even with the increasing stressing velocities. Here, in this case, the effective number of bonds are breaking at the range of 5 - 10 m/s. After 10 m/s the process becomes inefficient because there will not be any significant number of bonds breaking even with increasing input energy. The application of the input energy after this limiting velocity is a wastage of the energy.

This can be validated by the new surface generation curve (dotted curve) as shown in figure, which shows that the new surface generating process is efficient till 10 m/s.

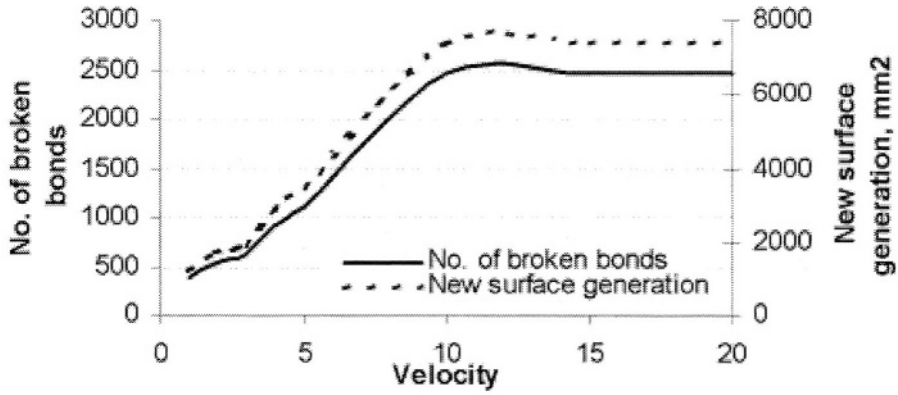


Fig. 5. Broken bonds and new surface generation versus velocity.

3 Conclusions

The stress pattern distributions obtained from the finite element simulation during stressing was the criterion to predict the crack initiation and propagation in the spherical particle compound materials. The formation of the wedge predicted by finite element method was further verified from the discrete element method. Discrete element analysis showed the crack initiation and propagation in the similar way as predicted by finite element simulation along with generation of different cracks and thus forming different fragments. The secondary cracks were clearly seen and analyzed with discrete element analysis.

The discrete element analysis was done with 2 dimensional software. It has presented the basics of crushing and fragmentation phenomena with less time. But there were some limitations in 2 D simulation, as the crushed cone could not be seen as cone instead it can be seen as wedge (triangle). Hence, during simulation it was realized that 3 dimensional software could provide realistic inside to the crushing system to analyse the cracking mechanism.

Acknowledgement. The authors would like to acknowledge German Research Foundation (DFG) for the financial support.

References

1. M. Khanal, W. Schubert, J. Tomas, Ball Impact and Crack Propagation - Simulations of Particle Compound Material, Granul. Matter, 5 (2004) 177-184.
2. H. Schubert, Zur Energieausnutzung bei Zerkleinerungsprozessen, Aufbereit. – Tech. 34 (10) (1993) 495-505
3. L. Kiss, K. Schönrt, Aufschlusszerkleinerung eines zweikomponentigen Modellstoffes unter einzelkornbeanspruchjung durch druck-rund prallbeanspruchjung [Liberation of two component materail by single particle compression and Impact crushing]. Aufbereit.-Tech.30(5)(1980)223-230
4. N. Arbiter, C. C. Harris, G. A. Stamboltzis, Single fracture of brittle spheres, Soc. Min. Eng. AIME, Trans. 244 (1969) 118-133
5. J. Tomas, M. Schreier, T. Gröger, S. Ehlers, Impact crushing of concrete for liberation and recycling, Powder Technol 105 (1999) 39-51
6. Alexander V Potapov, Charles S. Campbell, Computer simulation of impact-induced particle breakage, Power Techno, 82 (1994) 207-216
7. B. K. Mishra, C. Thornton, Impact breakage of a particle agglomerates, Int. J. Miner. Process. 61 (2001) 225-239
8. A. D. Salman, D. A. Gorham, A. Verba, A study of solid particle failure under normal and oblique impact, Wear 186-187 (1995) 92-98
9. R. Moreno, M. Gadhiri, S. J. Antony, Effect of impact angle on the breakage of agglomerates: a numerical study using DEM, Powder Techno, 130 (2003) 132-137
10. M. Khanal, W. Schubert, J. Tomas, Crack initiation and propagation – Central and oblique impacts, Comp. Mater. Sci. (submitted)
11. Thornton, M. T. Ciomocos, M. J. Adams, Numerical simulations of diametrical compression tests on agglomerates, (obtained per mail)
12. A. Lavrov, A. Vervoort, M. Wevers, J A L Napier, Experimental and numerical study of the Kaiser effect in cyclic Brazilian tests with disk rotation, Int. J. Rock Mech. Min., 39 (2002) 287-302
13. Malcom Mellor, Ivor Hawkes, Measurement of tensile strength by diametrical compression of discs and annuli, Eng. Geol., Vol, 5, Issue 3 (1971) 173-225
14. K. T. Chau, X. X. Wei, R. H. C. Wong, T. X. Yu, Fragmentation of brittle spheres under static and dynamic compressions: experiments and analyses, Mech. Mater., 32(2000) 543-554
15. S. Lee, G. Ravichandran, Crack initiation in brittle solids under multiaxial compression, Eng. Eng. Fract. Mech., 10 (2003) 1645-1658
16. Oliver Tsoungui, Denis Vallet, Jean Claude Charmet, Numerical model of crushing of grains inside two dimensional granular materials, Powder technology, 105 (1999) 190-1998
17. FEM Manual ANSYS 6.1, ANSYS Inc., Southpointe, 275 Technology Drive, Canonsburg, PA 15317.
18. Particle Flow Code in 2 Dimensions Manual, Vers. 3.0, Itasca Consulting Group Inc., Minneapolis, Minnesota, US (2002).

A Uniform and Reduced Mathematical Model for Sucker Rod Pumping

Leiming Liu[1], Chaonan Tong[1], Jianqin Wang[2], and Ranbing Liu[3]

[1]University of Science and Technology in Beijing, 100083, China
leiming.liu@163.com
[2]Laboratory of Remote Sensing Information Sciences, Institute of Remote Sensing Applications, Chinese Academy of Sciences, P. Box 9718, Beijing 100101, China
[3]University of Petroleum in Beijing, 102249, China

Abstract. The dynamic mathematical models were greatly developed to describe the motion of the sucker-rod pumping system since 1960s and various mathematic models based on different assumptions were presented. This paper does a study in three sides. First, a mathematical model is presented to describe the sucker-rod string with longitudinal and transverse vibrations, and coupled with longitudinal vibration of the tubing and fluid columns in a deviated well. Second, the relations of several kinds of mathematical models in sucker-rod pumping systems are discussed, and the model made in this paper, when based on different assumptions, can become different models made by people these years, which are presented in important references. Third, a method of characteristics is used to transform the set of partial differential equations which describe the vibration of the sucker-rod string, and coupled with vibrations of tubing and liquid columns in the sucker-rod pumping system. Through the transformation, a diagonal partial differential equations set is obtained. Hence a relatively complex model is transformed into a reduced model which is easy to solve. This model has basic meaning for using the technique of pattern recognition to make automatic diagnosis of sucker rod pumping system.

1 Introduction

Rod pumping is still the most widely used means for artificial lift in oil wells, so ever since 1960s, people lay great emphasis on the mathematical methods of predicting, designing and diagnosis of the sucker-rod pumping systems.

Gibbs(1963)[1] made a 1D mathematical model for the vibration of the sucker rod, which is comprised with a second-order partial differential equation including boundary conditions. Doty and Schmidt(1983)[2] presented a composite model in which both rod string dynamics and fluid dynamics are coupled to account for viscous damping, which was comprised of four first-order equations with four unknown variables, boundary conditions and initial conditions. In the paper of Wang et al. (1992)[3], a set of six equations governing the vibration of the sucker-rod string, tubing and liquid columns in the sucker-rod pumping system are presented.

All these three models stated above are for vertical wells. At present many wells are designed as deviated wells. Lukasiewicz(1991)[4] presented a model of sucker-

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 372–379, 2004.
© Springer-Verlag Berlin Heidelberg 2004

rod string with longitudinal and transverse displacements in a inclined well. Gibbs (1992)[5] studied the situation that sucker rod is crooked, but he assumed the transverse displacement was ignored, for the transverse displacement of the rod was constrained by the tubing. The wave equation about the rod with longitudinal displacement that Gibbs presented took the crooked wells into consideration. Xu and Mo(1993, 1995)[6,7] made a 3D model of vibrations of sucker-rod string. Li et al. (1999)[8] made a set of equations describing the 3D vibration of the sucker rod beam pumping system.

The models stated above didn't take the movement of the fluid column into consideration. On the basis of the work stated above, this paper presents a set of first-order partial differential equations and the concerning boundary and initial conditions. The equations describe the 2D vibration of rod string, the longitudinal vibration of tubing column and coupled with the displacement of the fluid column in the directional and horizontal wells. In these equations, three of them are geometry equations can be solved through the method dealing with non-dynamic problems, the other six first-order partial differential equations are a set of quasi-linear hyperbolic equations, they are dynamic equations, but can be reduced to diagonal equations, that is to make each equation be a partial deviation equation containing only one unknown variable.

2 Analysis of the Geometry Relations and Forces Acting on the Spatially Curved Rod

2.1 Geometry Equations of Spatial Curve

We build a spatial rectangular coordinates END (see Fig.1)for the spatial crooked sucker-rod at the well bore, take a rod element of length Δs and build a natural coordinates $t_1 t_2 t_3$ at the center of Δs. t_1 is the tangent unit vectors of the element Δs, t_2 is the principal unit normal vectors and t_3 is the subnormal unit normal vectors. The Frenet equations is:

$$\begin{cases} \boldsymbol{t}_1 = \mathrm{d}\boldsymbol{r}/\mathrm{d}s \\ \mathrm{d}\boldsymbol{t}_1/\mathrm{d}s = k\boldsymbol{t}_2 \\ \mathrm{d}\boldsymbol{t}_2/\mathrm{d}s = -k\boldsymbol{t}_1 + \tau\boldsymbol{t}_3 \\ \mathrm{d}\boldsymbol{t}_3/\mathrm{d}s = -\tau\boldsymbol{t}_2 \end{cases} \quad (1)$$

where k, τ are the curvature and buckling of the well axis curve and $\boldsymbol{r}$ the position vector of the point on the curve .

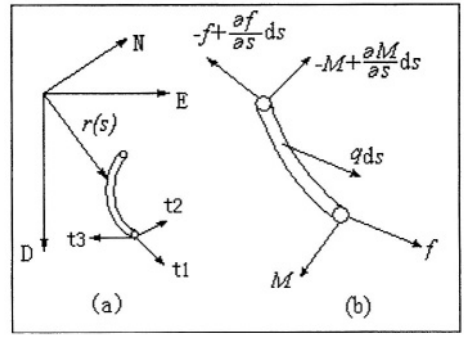


Fig. 1. Forces on rod element

2.2 The Analysis of Forces Acting on the Element of Curved Rod

Let the internal forces, force moments and external forces acting on element of the curved rod and the displacement of the centroid of the element in the natural coordinates respectively be presented as follows:

$$\begin{cases} \boldsymbol{f} = (f_1, f_2, f_3)^T, \boldsymbol{M} = (M_1, M_2, M_3)^T \\ \boldsymbol{q} = (q_1, q_2, q_3)^T, \boldsymbol{u} = (u_1, u_2, u_3)^T \end{cases} \tag{2}$$

We can obtain the following force equilibrium equations and moment equilibrium equations:

$$\begin{cases} \partial \boldsymbol{f} / \partial s + \boldsymbol{q} = m \partial^2 \boldsymbol{u} / \partial t^2 \\ \partial \boldsymbol{M} / \partial s = \boldsymbol{f} \times \boldsymbol{t}_1 \end{cases} \tag{3}$$

where m is the mass of unit length rod (kg), and $m = A\rho$, where A is the rod cross-sectional area (m^2), ρ is the rod material density ($\mathrm{kg/m}^3$). Let E_A be dragging stiffness of the rod ($\mathrm{N \cdot m}^2$), E_I be bending stiffness of the rod (N). If the effects of twisting moment are ignored, the constitutive equations describing the relations between the internal forces acting on the rod element of length and deformation are obtained:

$$\begin{cases} f = E_A(\dfrac{\partial u_1}{\partial s} - k u_2) \\ M_2 = E_I(-\dfrac{\partial^2 u_3}{\partial s^2} - \tau \dfrac{\partial u_2}{\partial s}) \\ M_3 = E_I(\dfrac{\partial^2 u_2}{\partial s^2} - \tau \dfrac{\partial u_3}{\partial s} + k) \end{cases} \tag{4}$$

3 Partial Differential Equations Describing the Movement of Suck-Rod String and Tubing and Fluid Columns in the Directional and Horizontal Wells

3.1 Fundamental Assumptions

(1) The well bore is considered as planar curve.
(2) The sucker-rod only has longitudinal and transverse vibrations in a plane, i.e.u_3=0.
(3) The cross-section of the sucker rod is round.
(4) The sucker rod string is concentric with the well bore.

According to these assumptions, from Eq. (4), we get M_2 =0, from the second equation of (3), we get $\dfrac{\partial M_1}{\partial s} = 0$.

3.2 Partial Differential Equations Describing Motions of Rod String and Tubing Column

Based on fundamental assumptions, we expand Eq.(3) and take the derivative of the second equation of Eq.(3) with respect to s, then apply the principal of vector derivation and Eq.(4), we obtain:

$$\begin{cases} \dfrac{\partial f_{r1}}{\partial s} + q_{r1} = m_r \dfrac{\partial v_{r1}}{\partial t}, (\dfrac{\partial u_{r1}}{\partial t} = v_{r1}) \\ \dfrac{\partial f_{r1}}{\partial t} = E_{Ar}(\dfrac{\partial v_{r1}}{\partial s} - k_r v_{r2}) \\ \dfrac{\partial f_{r2}}{\partial s} + q_{r2} = m_r \dfrac{\partial v_{r2}}{\partial t}, (\dfrac{\partial u_{r2}}{\partial t} = v_{r2}) \\ k f_{r1} - \dfrac{\partial f_{r2}}{\partial s} = E_{Ir} \dfrac{\partial^2}{\partial s^2}(\dfrac{\partial^2 u_2}{\partial s^2} + k) \end{cases} \tag{5}$$

where k_r is the plane curvature of the rod (1 /m), m_r is mass of the unit length of the sucker rod (kg), E_{Ar} the dragging stiffness (N), E_{Ir} the bending stiffness ($\mathrm{N \cdot m^2}$).

The curvature causes the lateral displacements of the rods between the two guides, so the relations between the transverse displacement and internal force of the sucker rod can be described by the last two equations of Eq.(5). As for the tubing, it has the same expressions as Eq.(5).

Considering the computing precision of the moment of the tubing, we ignore the transverse displacement of the tubing, the last two equations of Eq.(5) can become a geometry equation. In the following discussion the subscribe t refer to tubing properly, then we can obtain the differential equations describing vibrations of the tubing:

$$\begin{cases} \dfrac{\partial f_t}{\partial s} + q_{t1} = m_t \dfrac{\partial v_t}{\partial t}, (\dfrac{\partial u_t}{\partial t} = v_t) \\ \dfrac{\partial f_t}{\partial t} = E_{At} \dfrac{\partial v_t}{\partial s} \\ k_t E_{At} \dfrac{\partial u_t}{\partial s} + q_{t2} = E_{It} \dfrac{\mathrm{d}^2 k_t}{\mathrm{d}s^2} \end{cases} \tag{6}$$

3.3 1D Flow Equation of the Fluid

As for the flowing fluid of the tubing, we assume fluid column contains no gas, though the well bore is a curve, the movement of the fluid is still 1D flow. Let ρ_f be the density of the fluid ($\mathrm{kg/m^3}$) at the point of s and at the time of t ; and the functions of velocity (m/s) and pressure ($\mathrm{N/m^2}$) of the fluid are respectively represented by the variables of v_f and p_f. So the equation of continuity of the fluid, Euler motion equation and the state equation are presented as follows:

$$\begin{cases} \dfrac{\partial \rho_f}{\partial t} + \rho_f \dfrac{\partial v_f}{\partial s} + v_f \dfrac{\partial \rho_f}{\partial s} = 0 \\ \dfrac{\partial v_f}{\partial t} + v_f \dfrac{\partial v_f}{\partial s} + \dfrac{1}{\rho_f} \dfrac{\partial p_f}{\partial s} = \dfrac{q_f}{\rho_f (A_t - A_r)} \\ p_f = p_f(\rho_f) \end{cases} \tag{7}$$

where q_f is the external forces acting on the fluid column of per unit length (including forces by the sucker rod string , rod coupling and tubing), A_r is the cross-sectional area of the sucker rod and A_t the inner cross-sectional area of the tubing (m^2). The equation of fluid state shows that the pressure p_f is only related with density ρ_f , and according to the physical meaning, $\frac{dp}{d\rho} > 0$, so define

$$\frac{d\rho_f}{dp_f} = a(\rho_f)^2 \tag{8}$$

ρ_f has the representation:

$$\rho_f = \rho_0^{C_\rho (p_f - p_0)} \tag{9}$$

or the approximate expression of Eq. (9):

$$\rho_f = \rho_0 [1 + C_\rho (p_f - p_0)] \tag{10}$$

where ρ_0 , p_0 are respectively the density and pressure of the fluid in the standard state, C_ρ is the relative increment of the density when the pressure increases one unit, i.e. elastic condensation coefficient of the density of the fluid. Let

$$Q_f = q_f / \rho_f (A_t - A_r) \tag{11}$$

Using Eqs. (8) and (11), Eq. (7) becomes the following equations:

$$\begin{cases} \dfrac{\partial \rho_f}{\partial t} + v_f \dfrac{\partial \rho_f}{\partial s} + \rho_f \dfrac{\partial v_f}{\partial s} = 0 \\ \dfrac{\partial v_f}{\partial s} + v_f \dfrac{\partial v_f}{\partial s} + \dfrac{a^2(\rho_f)}{\rho_f} \dfrac{\partial \rho_f}{\partial s} = Q_f \end{cases} \tag{12}$$

4 Analysis of External Forces Acting on the Rod String and Tubing and Fluid Columns

Through the effects of the fluid viscosity resistance, lateral extrusion stress, coulomb friction and gravity caused by the periodic motion of the sucker rod, the sucker rod

string, tubing column and fluid column comprises a system of motion. These forces are expressed as follows:

$$q_{r1} = -\eta v_{r1} - \lambda |q_N| sign(v_{r1}) + mg\cos\alpha \tag{13}$$

$$q_{r2} = -\eta v_{r2} + q_N \cos\varphi - mg\sin\alpha \tag{14}$$

where η is viscous damping coefficient of the fluid, m_r is the mass of the unit length rod, φ is the angle between lateral extrusion stress and the principal normal vector, λ is friction factor, q_N is lateral extrusion stress between the rod and the tubing[7], α is the angle between the tangent of the element of the curved rod and vertical. Here the viscosity resistance is comprised of two parts, one is the viscosity resistance between fluid and the sucker rod, the other between the fluid and the coupling of the rod, and also:

$$q_{t1} = -\eta v_t - \lambda |q_N| sign(v_t) + mg\cos\alpha \tag{15}$$

$$q_{t2} = q_N \cos\varphi - m_t g\sin\alpha \tag{16}$$

where η, λ, q_N is the same as which is stated above, but the difference is there is no term of viscous resistance in Eq. (16), and m_t is the mass of the unit length rod.
The force q_f in Eq. (7) has the representation as follows:

$$q_f = -\rho_f (A_t - A_r)g + \eta v_{r1} + \eta v_t \tag{17}$$

ηv_{r1}, ηv_t of Eq. (17) has the contrary sign with the corresponding term of Eqs. (13) and (15), this shows that the viscous resistance acting on the rod string and tubing column has the reaction to the fluid.

5 Boundary and Initial Conditions

The boundary and initial conditions are very important, the former determines if the solution can correctly describe the motion stated, and the correct choice of the latter can make the computing program converges to its periodic numeric solution of stationary state as fast as possible. Papers [1], [2], [3] and [7] made deep study about the boundary and initial conditions and presented useful conclusions. This paper applies those results.
As for the diagnosis, the surface boundary conditions are

$$\begin{cases} f_{r1}(0,t) = f_1(t), v_{r1}(0,t) = v_1(t), v_{r2}(0,t) = 0, f_{r2}(0,t) = 0 \\ v_t(0,t) = 0, f_t(0,t) = 0, v_{r2}(s_i,t) = 0, p_f(0,t) = p_0(t) \end{cases} \tag{18}$$

where $f_1(t)$ is the measured loading function of the polished rod, $v_1(t)$ is the velocity of the polished rod , $s_i\,(i=1,2,\cdots,N)$ represents placing a rod guide where $s = s_i$, and

p_0 represents tubing head pressure and is constant in most of the cases. As for the predicted model, we should consider the boundary conditions in the place of oil-well pump, see paper [3].

The correct choice of the initial conditions can make the computing program converges to its periodic numeric solution of stationary state as fast as possible, so refer to papers [2] and [3], we make discrete transform along the axes of the well for the initial conditions given in the papers, and obtain the initial conditions for this model.

6 Characteristic Transform of the Partial Differential Equations Set

The first two equations of Eq. (5), Eq. (6) and Eq.(12) comprise a equations set about the unknown variables of $v_{r1}, f_{r1}, v_t, f_t, v_f$ and ρ_f. We apply the theory of characteristic for the quasi-linear hyperbolic partial differential equations set to transform the six equations into a diagonal equations set about the variables of w_1, w_2, w_3, w_4, w_5 and w_6.

$$\begin{cases} \dfrac{\partial w_1}{\partial t} - \sqrt{\dfrac{E_{Ar}}{\rho_r}}\dfrac{\partial w_1}{\partial s} = \dfrac{q_{r1}}{\rho_r A_r}, \dfrac{\partial w_2}{\partial t} + \sqrt{\dfrac{E_{Ar}}{\rho_r}}\dfrac{\partial w_2}{\partial s} = \dfrac{q_{r1}}{\rho_r A_r} \\ \dfrac{\partial w_3}{\partial t} - \sqrt{\dfrac{\rho_t}{E_{At}}}\dfrac{\partial w_3}{\partial s} = \dfrac{q_t}{\rho_t(A_h - A_t)}, \dfrac{\partial w_4}{\partial t} + \sqrt{\dfrac{\rho_t}{E_{At}}}\dfrac{\partial w_4}{\partial s} = \dfrac{q_t}{\rho_t(A_h - A_t)} \\ \dfrac{\partial w_5}{\partial t} + (v_f + a)\dfrac{\partial w_5}{\partial s} = Q_f, \dfrac{\partial w_6}{\partial t} + (v_f - a)\dfrac{\partial w_6}{\partial s} = -Q_f \end{cases} \tag{19}$$

where $w_1 = v_{r1} + \dfrac{1}{A_r\sqrt{\rho_r E_{Ar}}} f_{r1}, w_2 = v_{r1} - \dfrac{1}{A_r\sqrt{\rho_r E_{Ar}}} f_{r1}, w_3 = v_t + \dfrac{1}{\sqrt{\rho_t E_{At}(A_h - A_t)}} f_t$,

$$w_4 = v_t - \frac{1}{\sqrt{\rho_t E_{At}(A_h - A_t)}} f_t,\ w_5 = v_f + \int_{\rho_{f0}}^{\rho_f} \frac{a(z)}{z}\mathrm{d}z,\ w_6 = -v_f + \int_{\rho_{f0}}^{\rho_f} \frac{a(z)}{z}\mathrm{d}z.$$

where ρ_{f0} is the density of the fluid when $p = p_0$ and p_0 is the tubing head pressure. The Eq. (19) is easy to solve applying the difference method.

7 Conclusions

The model presented in this paper is the basic model for the directional and horizontal wells, it takes the longitudinal and transverse vibrations of the sucker rod into consideration, and also considers the coupled motions of rod string and tubing and fluid columns. So it is a relatively accurate model simulating the dynamic behavior of the sucker-rod system.

This is a basic model and concentrates the characters of some principal mathematical models of the sucker-rod system. When the curvature of the well bore $k = 0$, the mathematical model becomes the model for the coupled vibrations of the rod string and tubing and fluid column in the vertical well[3]. If we further consider the tubing is anchored, and assume the tubing is rigid, the model describing the coupled vibrations of the rod string and fluid column in the vertical well is obtained[2]. If the fluid is assumed as non-condensable, the wave equation of the vibration of the sucker-rod in the vertical well is obtained[1].

As for the mathematical model about the directional and horizontal well, if we assume the fluid is non-condensable, and assume the tubing is anchored and rigid, we can obtain the mathematical model about the 2D vibration of the sucker rod in the directional and horizontal well[4,5].

Finally, according to the characteristic deformation we have done, as the Eq. (19) shows, it is also easily-solving model.

Acknowledgement. Jianqin Wang thanks Dr. Yong Xue, his PhD supervisor, for his support from the "CAS Hundred Talents Program".

References

1. Gibbs, S.G., 1963. Predicting the behavior of sucker-rod systems. J. Pet. Technol. (Jul.), pp. 769-778.
2. Doty, D.R. and Schmidt, Z., 1983. An improved model for sucker rod pumping. J. Soc. Pet. Eng. (Feb.), pp. 33-41.
3. Wang, G.W., Rahman,S.S. and Wang, G.Y.,1992.An improved model for sucker rod pumping systems. Proc. 11th. Australas Fluid Mech. Conf., Tasmania, 14-18 Dec., 1992, 2:1137-1140.
4. Lukasiewicz, S.A., 1991.Dynamic behavior of the sucker rod string in the inclined well.SPE 21665,pp.313- 321.
5. Gibbs, S.G., 1992.Design and diagnosis of deviated rod-pumped wells.J. Pet. Technol. (Jul.), pp.774-781.
6. Xu, J., Hu, Y. and U.T.,1993.A method for designing and predicting the sucker rod string in deviated pumping wells.SPE 26929 ,pp.383-384.
7. Xu, J.and Mo, Y.,1995.Longitudinal and transverse vibration model of the sucker rod string in directional wells and its application in diagnosis.J.Tongji.University.(Feb.), pp.26-30.
8. Li, Z. et al., 1999.Fundamental equations and its applications for dynamical analysis of rod and pipe string in oil and gas wells.ACTA.20(3),pp.87-90.

Distributed Computation of Optical Flow

Antonio G. Dopico[1], Miguel V. Correia[2], Jorge A. Santos[3], and Luis M. Nunes[4]

[1] Fac. de Informatica, U. Politecnica de Madrid, Madrid. dopico@fi.upm.es
[2] Inst. de Engenharia Biomédica, U. do Porto, Fac. de Engenharia. mcorreia@fe.up.pt
[3] Inst. de Educacao e Psicologia, U. do Minho, Braga. jas@iep.uminho.pt
[4] Dirección General de Tráfico, Madrid. argos@dgt.es

Abstract. This paper describes a new parallel algorithm to compute the optical flow of a video sequence. A previous sequential algorithm has been distributed over a cluster. It has been implemented in a cluster with 8 nodes connected by means of a Gigabit Ethernet. On this architecture, the algorithm, that computes the optical flow of every image on the sequence, is able of processing 10 images of 720x576 pixels per second.

Keywords: Optical Flow, Distributed Computing

1 Introduction

There is a wide variety of areas of interest and application fields (visual perception studies, scene interpretation, motion detection, filter for in-vehicle inteligent systems etc.) that can benefit from optical flow computing. The concept of optical flow derives form a visual system concept analogue to human retina, in which a 3d world is represented in a 2d surface by means of an optical projection. In the present case we will use a simplified 2d representation consisting in a matrix of n pixels in which only grey values of image are considered. Spatial motion and velocity is then represented as a 2d vector field showing the distribution of velocities of apparent motion of the brightness pattern of a dynamic image.

The optical flow computation of a moving sequence is an intensive demanding application both in memory and computational terms. As the computers performance improves the users expectations raises too: higher resolution video recording systems allow to reduce the negative effects of spatial and temporal motion aliasing. In [1] synthetic images with 1312x2000 pixels at 120 Hz are used. Given the growing need of computer performance the parallelization of the optical flow computation appears to be the only alternative to achieve a massive processing of long video sequences.

This idea of parallelization was proposed some years ago, [2], with four processors, obtained very modest results: processing up to 7-8 images of 64x64 pixels per second, too small resolution to be useful.

More recently [3] proposes the decomposition of the optical flow computation in small tasks: by dividing the image in independent parts the parallelization

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 380–387, 2004.
© Springer-Verlag Berlin Heidelberg 2004

becomes easier to approach, although with the drawback of the overheads associated with dividing the images and grouping the obtained results. As this has not been yet implemented no results are available.

A possible alternative to parallellization could be to simplify drastically the optical flow algorithm. [4] Presents an alternative based on additions-subtractions that needs much less computational resources but, according to the authors, with the pay-off of incorrect results.

In the present work the parallelization of the optical flow computation is approached with the objective of maximizing performance with no lost of the quality of the results, allowing massive computing of long sequences using standard image resolutions. The gain due to parallelization will be referred to a sequential version of an equivalent algorithm.

2 Optical Flow Computation Sequential Algorithm

Following the survey of Barron et al., [5], the method of Lucas, [6,7], has been chosen to compute optical flow. This method seems to provide the best estimate with less computational effort. Fleet's method, [8], would possibly provide an even better estimate, but the computational cost would be higher due to the use of several Gabor spatio-temporal filters [5].

2.1 Lucas Optical Flow Algorithm

In Lucas's method, optical flow is computed by a gradient based approach.

It follows the common assumption that image brightness remains constant between time frames:

$$I(x, y, t) = I(x + u\delta t, y + v\delta t, t + \delta t) \tag{1}$$

which, by also assuming differentiability and using a Taylor series expansion, can be expressed by the motion constraint equation:

$$I_x u\delta t + I_y v\delta t + I_t \delta t = \mathcal{O}\left(u^2\delta t^2, v^2\delta t^2\right) \tag{2}$$

or, in more compact form (considering δt as the temporal unity):

$$\nabla I \cdot \mathbf{v} + I_t = \mathcal{O}\left(\mathbf{v}^2\right) \tag{3}$$

where $\mathcal{O}\left(\mathbf{v}^2\right)$ represents second order and above terms.

In this method, the image sequence is first convolved with a spatio-temporal Gaussian to smooth noise and very high contrasts that could lead to poor estimates of image derivatives. Then, according to Barron et al. implementation, the spatio-temporal derivatives I_x, I_y and I_t are computed with a four-point central difference.

Finally, the two components of velocity $\mathbf{v} = (u, v)$ are obtained by a weighted least-squares fit of local first-order constraints, assuming a constant model for $\mathbf{v}$ in each small spatial neighborhood $\mathcal{N}$, by minimizing:

$$\sum_{x \in \mathcal{N}} \mathbf{W}^2(\mathbf{x}) \left[\nabla I(\mathbf{x}, t) \cdot \mathbf{v} + I_t(\mathbf{x}, t)\right]^2 \tag{4}$$

where $\mathbf{W}(\mathbf{x})$ denotes a window function that weights more heavily at the centre. The solution results from:

$$\mathbf{v} = (\mathbf{A}^T\mathbf{W}^2\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}^2\mathbf{b} \tag{5}$$

where, for n points $\mathbf{x}_i \in \mathcal{N}$ at a single time t,

- $\mathbf{A} = [\nabla I(\mathbf{x}_1), ..., \nabla I(\mathbf{x}_n)]^T$,
- $\mathbf{W} = diag[\mathbf{W}(\mathbf{x}_1), ..., \mathbf{W}(\mathbf{x}_n)]$ and
- $\mathbf{b} = -(I_t(\mathbf{x}_1), ..., I_t(\mathbf{x}_n))^T$.

The product $\mathbf{A}^T\mathbf{W}^2\mathbf{A}$ is a 2×2 matrix given by:

$$\mathbf{A}^T\mathbf{W}^2\mathbf{A} = \begin{bmatrix} \sum \mathbf{W}^2(\mathbf{x}) I_x^2(\mathbf{x}) & \sum \mathbf{W}^2(\mathbf{x}) I_x(\mathbf{x}) I_y(\mathbf{x}) \\ \sum \mathbf{W}^2(\mathbf{x}) I_y(\mathbf{x}) I_x(\mathbf{x}) & \sum \mathbf{W}^2(\mathbf{x}) I_y^2(\mathbf{x}) \end{bmatrix} \tag{6}$$

where all sums are taken over points in the neighborhood $\mathcal{N}$.

Simoncelli [9,10] present a Bayesian perspective of equation 4. They model the gradient constraint using Gaussian distributions. This modification allows to identify unreliable estimates using the eigenvalues of A^TW^2A.

2.2 Implementation

Now the sequential implementation of the Lucas-Kanade algorithm is explained [11,12]:

- The implementation first smoothes the image sequence with a spatiotemporal Gaussian filter to attenuate temporal and spatial noise as do Barron et al. [5]:
  - Temporal smoothing Gaussian filter with $\sigma = 3.2$, requiring $6\sigma + 1$ (21) frames, the current frame, 3σ (10) past frames and 3σ (10) future frames.
  - Spatial smoothing Gaussian filter with $\sigma = 3.2$, requiring $6\sigma + 1$ (21) pixels, the central pixel and 3σ (10) pixels for each side relative to this central pixel. This symmetric Gaussian filter in one dimension is applied twice, first in the X dimension and then in the Y dimension.
- After the smoothing, spatiotemporal derivatives (I_t, I_x, I_y) are computed with 4-point central differences with mask coefficients:

$$\frac{1}{12}(-1, 8, 0, -8, 1) \tag{7}$$

- Finally, the velocity is computed from the spatiotemporal derivate:
  - A spatial neighborhood of 5x5 pixels is used for the velocity calculations.
  - A weight matrix identical to Barron [5], i.e., with 1-D weights of (0.0625, 0.25, 0.375, 0.25, 0.0625) is also used for the velocity calculations.
  - The noise parameters used are $\sigma_1 = 0.08$, $\sigma_2 = 1.0$, $\sigma_p = 2.0$ ([9]).
  - Velocity estimates where the highest eigenvalue of A^TW^2A is less than 0.05 is considered unreliable and removed from the results ([5]).

2.3 Sequential Algorithm Results

Figure 1.a shows an image of a interlaced video sequence with 720x576 pixels, that have been processed with the described algorithm. The optical flow obtained is shown in figure 1.b. The car on the left is going faster than the car on the center and the car on the right is going slower than the car on the center.

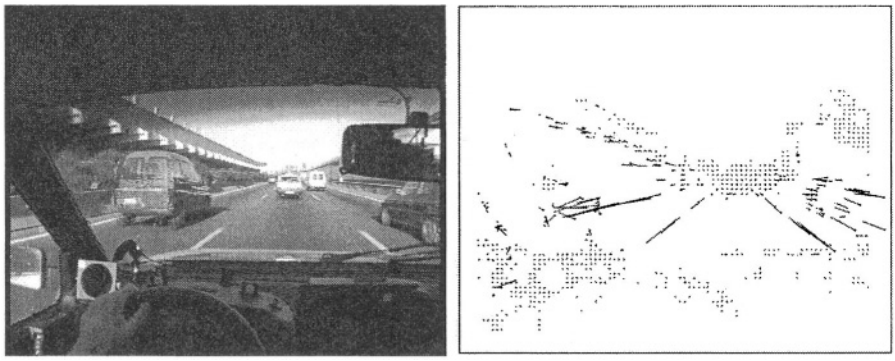


Fig. 1. Video Sequence: Frames 19 and 29.

3 Parallelization of the Optical Flow Computing

The parallelization of the sequential algorithm is explained in this section.

3.1 Parallel Algorithm

The execution time of the different tasks of the sequential algorithm have been measured to obtain an estimation of its weights. The measures have been obtained using a workstation with an Intel Xeon 2.4 GHz and 1GB of main memory, though the important data are not the absolute times but the relationship among the different tasks.

- The temporal smooth, in T, is slower than the others because it works with a high number of images. Moreover it has to read them from the disk (12 ms).
- The spatial smooth in X employs 8 ms.
- The spatial smooth in Y employs 7 ms. Probably the difference is because now the image is in the cache memory.
- Computation of the partial derivatives, (It,Ix,Iy), 10 ms.
- Computation of the velocity of each pixel and writing the results to disk, 130 ms. This is more than triple the time spent by the rest of the tasks.

These times are spent with each image in the video sequence.

Unlike [3] the images have not been divided to avoid the introduction of unnecessary overheads, because in that case they had to be divided, then processed and finally group the results. Moreover, the possible boundary effects should be taken into account. Anyway, this option could be useful in some cases.

To structure the parallelization, the existing tasks have been taken into account. The first four tasks are connected as a pipeline because they need the data of several images to work properly. The last one only needs a single image and it is actually independent. The fourth task will send derivatives from complete images to different copies of task five in a rotative way.

Although a 8 nodes cluster has been used for the implementation, the followed schema is flexible enough to be adapted to different situations:

- Four nodes. The first one executes all the tasks except computing the velocity of the pixels (37 ms). The rest of the nodes compute the velocities, when they finish with an image, they start with the next one (130/3 = 43 ms per node). One image would be processed every 43 ms (maximum of 37 and 43).
- Eight nodes. The first node computes the temporal smooth and the spatial smooth for the X co-ordinate (12+8=20ms). The second one computes the spatial smooth for the Y co-ordinate and the partial derivatives (7+10=17ms). The rest of the nodes compute the velocities (130/6=21ms). An image is processed every 21 ms (maximum of 20, 17 and 21).
- Sixteen nodes. The first four nodes are dedicated to the first four tasks (12, 8, 7 and 10 ms respectively). The rest of the nodes compute the velocities (130/12=11ms). An image would be processed every 12ms (maximum of 12, 8,7,10,11).

In the three cases, the communication time has to be added. This time would depend on the net (Gigabit, Myrinet, etc.) but in every case it has to be taken into account and it will employ several milliseconds.

With this scheme, even if a cluster is not used it would not be a problem. For example, a shared memory tetraprocessor could be used and the tasks could be distributed in the same way than with a four nodes cluster.

With more than 16 nodes, there are not enough tasks to distribute. To obtain a higher degree of parallelism the images would be divided as [3] proposes. Each subimage would be independent of the rest if some boundary pixels are added. That is, as the spatial smooth uses 25 pixels, the central one and 12 on each side, each subimage would need 12 pixels more per boundary. So, to divide images of 1280x1024 pixels in 4 subimages (2x2) they should be divided in regions of 652x524 pixels, with overlapped boundaries. In this way each image would be totally independent.

3.2 Cluster Architecture

A cluster with 8 biprocessor nodes (2.4 GHz, 1GB RAM) running Linux (Debian with kernel 2.4.21) and openMosix has been used. The nodes are connected using a Gigabit Ethernet switch. This distributed memory architecture was chosen because it is not expensive, it is easy to configure and it is broadly extended.

3.3 Implementation

The tasks of the previously described algorithm have been assigned to the different nodes of the cluster. For communications, the message passing standard MPI, in short, the open source implementation LAM/MPI version 6.5.8. of the University of Indiana, has been used.

For the mentioned communications, non blocking messages have been used, in such a way that the computation and the communications are overlapped. Moreover, the use of persistent messages avoids the continuous creation and destruction of the data structures used by the messages. This has been possible because the communication scheme is always the same. The information that travels between two given nodes has always the same structure and the same size so, the message backbone can be reused.

About the non blocking messages, a node, while processing the image `i`, has already started a non blocking sending to transfer the results of processing the previous image `i-1` and has also started a non blocking reception to simultaneously gather the next image `i+1`. This allows simultaneously send, receive and compute in each node.

About the task distribution among the nodes, the scheme has been the following:

- **Node 1**. Executes the following tasks:
  - Reads the images of the video sequence from the disk.
  - Executes the temporal smooth. To do that, the current image and the twelve previous ones are used
  - Executes the spatial smooth for the `x` co-ordinate.
  - Sends to the node 2 the image smoothed in `t` and `x`.
- **Node 2**. Executes the following tasks:
  - Receives the image from node 1.
  - Executes the spatial smooth for the `y` co-ordinate.
  - Computes the partial derivative in `t` of the image. To do that five images are used, the current one, the two previous and the two next. So, if the image `i` is received, the derivative in `t` of the image `i-2` is computed.
  - Computes the partial derivatives in `x` and `y` of the image.
  - Sends the computed derivatives `It`, `Ix` and `Iy` to the next nodes (from 3 to 8) in a cyclic mode. When the node 8 is reached, it starts again in the node 3.
- **Rest of the nodes.** They execute the following tasks:
  - Receive the partial derivatives in `t`, `x` and `y` of the image, `It`, `Ix` and `Iy`.
  - Using the derivatives, computes the velocity of each pixel as (vx, vy).
  - Write in the disk the computed velocities.

Figure 2 shows the distribution of the tasks among the nodes.

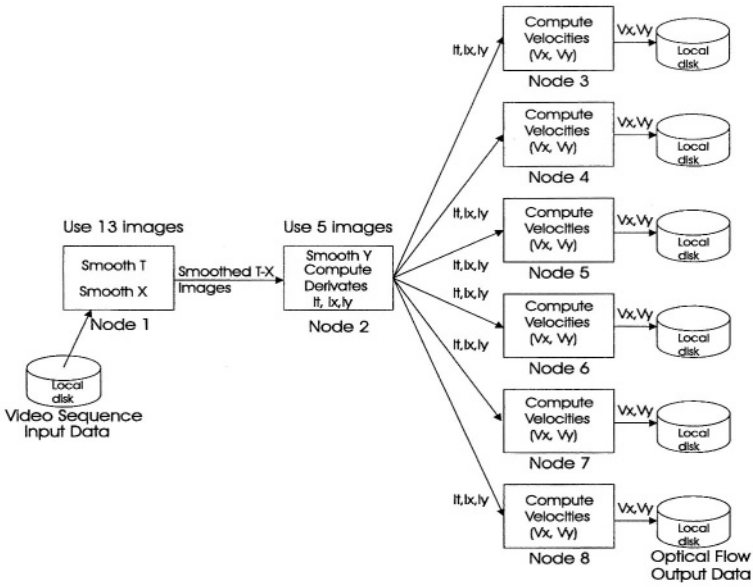


Fig. 2. Tasks Distribution.

3.4 Results

With this parallelization scheme and using the above described cluster, the computation of the optical flow is achieved at 30 images per second with images of 502x288 pixels. For images of 720x576 pixels the obtained speed is 10 images per second. Note that the optical flow, in both cases, is computed for every image in the video sequence without skipping any one. This performance means a speedup of 6 over the sequential version employing 8 nodes.

4 Conclusions and Future Works

This paper presents a new distributed algorithm for computing the optical flow of a video sequence. This algorithm is based on the balanced distribution of its tasks among the nodes of a cluster of computers. The distribution done is flexible and can be adapted to several environments, with shared memory as well as with distributed memory. Moreover, it is easily adaptable to a wide range of nodes number: 4, 8, 16, 32 or more.

The algorithm has been implemented on a cluster with 8 nodes and a gigabit Ethernet, where 30 images per second can be processed with images of 502x288 pixels, or 10 images per second if the images are of 720x576 pixels. With respect to the sequential version the resulting speedup is 6 times faster.

Taking into account the modest performance obtained in [2] with four processors (6-7 images per second with images of 64x64 pixels), or the inconvenients of the simplified algorithms [4] the results obtained with the algorithm proposed

here are very hopeful. The interesting parallelization [3] cannot be compared because it is not yet implemented.

The obtained performance brings important advantages. Working with longer sequences, larger images (1280x1024 pixels or even larger) and higher frequencies is now feasible. Increased temporal resolution is particularly beneficial in complex scenarios with high speed motion. In this line, the particular motion aliasing pattern of the current interlaced cameras can be reduced by an additional algorithm that duplicates the frequency and may be helpful prior the optic flow computation: the video sequence can be rebuilt by combining each half frame both with the prior and with the next half-frame, (hf1+hf2); (hf2+hf3); (hf3+hf4); (hf4+hf5), and so on. The result is an upgraded sequence with less motion aliasing and double temporal frequency.

Regarding real time applications, by connecting the video signal directly to one of the nodes of the cluster and digitizing the video sequence on the fly, the current implementation of the algorithm allows on line optical flow calculation of images of 502x288 pixels at 25 to 30 Hz.

References

1. Lim, S., Gamal, A.: Optical flow estimation using high frame rate sequences. In: Proceedings of the International Conference on Image Processing (ICIP). Volume 2. (2001) 925–928
2. Valentinotti, F., Di Caro, G., Crespi, B.: Real-time parallel computation of disparity and optical flow using phase difference. Machine Vision and Applications **9** (1996) 87–96
3. Kohlberger, T., Schnörr, C., Bruhn, A., Weickert, J.: Domain decomposition for parallel variational optical flow computation. In: Proceedings of the 25th German Conference on Pattern Recognition, Springer LNCS. Volume 2781. (2003) 196–202
4. Zelek, J.: Bayesian real-time optical flow. In: Proceedings of the 15th International Conference on Vision Interface. (2002) 266–273
5. Barron, J., Fleet, D., Beauchemin: Performance of optical flow techniques. International Journal of Computer Vision **12** (1994) 43–77
6. Lucas, B.: Generalized Image Matching by Method of Differences. PhD thesis, Department of Computer Science, Carnegie-Mellon University (1984)
7. Lucas, B., T., K.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI). (1981) 674–679
8. Fleet, D., Langley, K.: Recursive filters for optical flow. IEEE Transactions on Pattern Analysis and Machine Intelligence **17** (1995) 61–67
9. Simoncelli, E., Adelson, E., Heeger, D.: Probability distributions of optical flow. In: IEEE Conference on Computer Vision and Pattern Recognition. (1991) 310–315
10. Simoncelli, E.: Distributed Representation and Analysis of Visual Motion. PhD thesis, Massachusetts Institute of Technology (1993)
11. Correia, M., Campilho, A., Santos, J., Nunes, L.: Optical flow techniques applied to the calibration of visual perception experiments. In: Proceedings of the Int. Conference on Pattern Recognition, 13 ICPR. Volume 1. (1996) 498–502
12. Correia, M., Campilho, A.: Real-time implementation of an optical flow algorithm. In: Proceedings of the Int. Conference on Pattern Recognition, 16th ICPR, Volume IV. (2002) 247–250

Analytical Test on Effectiveness of MCDF Operations

Jun Kong[1,2], Baoxue Zhang[3], and Wanwu Guo[1]

[1] School of Computer and Information Science, Edith Cowan University
2 Bradford Street, Mount Lawley, Western Australia 6050, Australia
w.guo@ecu.edu.au
[2] Department of Computer Science, Northeast Normal University
138 Renmin Street, Changchun, Jilin, China
kongjun@nenu.edu.cn
[3] School of Mathematics and Statistics, Northeast Normal University
138 Renmin Street, Changchun, Jilin, China
zhangbx@nenu.edu.cn

Abstract. Modified conjugate directional filtering (MCDF) is a method proposed by Guo and Watson recently for digital data and image processing. By using MCDF, directionally filtered results in conjugate directions can be not only merged into one image that shows the maximum linear features in the two conjugate directions, but also further manipulated by a number of predefined generic MCDF operations for different purposes. Although a number of cases have been used to test the usefulness of several proposed MCDF operations, and the results are 'visually' better than some conventional methods, however, no quantified analytical results on its effectiveness have been obtained. This has been the major obstacle on the decision whether it is worth developing a usable MCDF system. This paper firstly outlines a FFT-based analytical design for conducting the tests, and then presents the results of applying this analytical design to the analysis of MCDF(add1) operation for an image of digital terrain model in central Australia. The test verifies that the MCDF(add1) operation indeed overcomes the two weaknesses of using the conventional directional filtering in image processing, i.e., separation in presentation of processed results in different directions, and significant loss in low-frequency components. Therefore, the MCDF method is worth for further development.

1 Introduction

Guo and Watson [1] recently reported the trial work on using a method called the modified conjugate directional filtering (MCDF) for digital image processing. By using MCDF, directionally filtered results in conjugate directions can be not only merged into one image that shows the maximum linear features in the two conjugate directions, but also further manipulated by a number of predefined MCDF operations for different purposes. MCDF is modified from the previous proposal named conjugate directional filtering (CDF) [2], because further study reveals that the CDF has two weaknesses, i.e., a weighting system for further data manipulation during the operation was not considered; CDF-processed images often lack contrast depth because most background information is removed as a result of applying directional filtering.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 388–395, 2004.
© Springer-Verlag Berlin Heidelberg 2004

MCDF overcomes these problems by superimposing the weighted CDF data onto the original data. In this way, not only further enhanced by a weighting factor are the conjugated features, but also retained is all the information on the original image. By introducing these two considerations into the CDF, MCDF becomes much more powerful [1][3].

Although a number of cases have been used to test the usefulness of several proposed MCDF operations, and the results are 'visually' better than some conventional methods [4][5], however, no quantified analytical results on its effectiveness have been obtained. This has been the major obstacle on the decision whether it is worth the development of a usable MCDF system. Our recent study on analytical test design and experiments on MCDF operations have led to the acquisition of some positive and encouraging results. In this paper, we first briefly present the concepts of the MCDF operations, and then outline the analytical design for the tests. Due to the restriction on the paper length, we only apply this analytical design to the analysis of MCDF(add1) operation in this paper. An image of digital terrain model in central Australia is used to analyze the effectiveness of the MCDF(add1) operation.

2 MCDF Operations and Analytical Test Design

Directional filtering is used to enhance linear features in a specific direction [4][5][6]. In some cases, identifying conjugate linear information on an image is particularly concerned. Directional filtering can be made in two specific conjugate directions to enhance these conjugate features. Normally the filtered results from the two conjugate directions are shown on two separate images. This is inconvenient for revealing the relationships between linear features in the two conjugate directions. The linear enhancement using directional filtering is achieved by constraining or removing the textural features or low-frequency components from the original image to outline the structural features or high-frequency components contained in the original image. Thus, directionally filtered image often lacks contrast depth because most background information is removed.

These two weaknesses of using the conventional directional filtering are overcome by MDCF method, which firstly combines two (or more) directional-filtered results in conjugate directions into one image that exhibits the maximum linear features in the two conjugate directions, and secondly retains the background information by superimposing the directionally filtered data onto the original data. Therefore, the analytical tests should be designed in a way through which these two improvements can be clearly revealed.

Assuming f_0 to be the original data file, f_1 and f_2 to be the directional-filtered data files in the two conjugate directions, the general operation of the MCDF can be expressed as [1]

$$\text{MCDF} = W_0 \cdot f_0 + F_2[W_1 \cdot F_1(f_1), W_2 \cdot F_1(f_2)]; \tag{1}$$

where W_0, W_1 and W_2 are selective constants; F_0, F_1 and F_2 are pre-defined generic functions. Consequently, some MCDF operations are defined using formula (1) as

$$\text{MCDF(add1)} = W_0 \cdot f_0 + W_1 \cdot f_1 + W_2 \cdot f_2; \tag{2}$$

$$\text{MCDF(add2)} = W_0 \cdot f_0 + \text{abs}(W_1 \cdot f_1 + W_2 \cdot f_2); \tag{3}$$

$$\text{MCDF(add3)} = W_0 \cdot f_0 + W_1 \cdot \text{abs}(f_1) + W_2 \cdot \text{abs}(f_2); \tag{4}$$

$$\text{MCDF(max1)} = F_0(W_0 \cdot f_0) + \max(W_1 \cdot f_1, W_2 \cdot f_2); \tag{5}$$

$$\text{MCDF(max2)} = F_0(W_0 \cdot f_0) + \max[W_1 \cdot \text{abs}(f_1), W_2 \cdot \text{abs}(f_2)]; \tag{6}$$

$$\text{MCDF(ampl)} = W_0 \cdot f_0 + \text{sqrt}(W_1 \cdot f_1 \cdot f_1 + W_2 \cdot f_2 \cdot f_2). \tag{7}$$

We propose a design for taking the analytical tests shown in Figure 1. Firstly, original image and each of its MCDF-processed images are input individually to a processing unit for analysis using fast Fourier transform (FFT). The output from this FFT analysis includes the 2D Cartesian spectrum, and radial spectrum of the corresponding input image [7]. Compared with the outcomes of the original and MCDF(add1) images, the 2D Cartesian spectrum is used to directly identify whether the MCDF operations indeed have brought enhanced information in the conjugate directions into MCDF-processed images; the radial spectrum is used to quantify whether the MCDF-processed images have retained the background information or low-frequency components of the original image while the structural features or high-frequency components are enhanced.

To make the analytical results acceptable as widely as possible, the FFT analysis of our tests are carried out using FFT functions provided by Matlab [8] [9]. Next section reports the test results of MCDF(add1) operation using this test design on an image of digital terrain model.

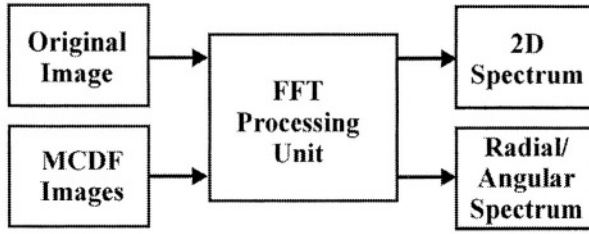


Fig. 1. Schematic diagrams of the design for analytical test of MCDF operations

3 Test Results of Image of Digital Terrain Model (DTM)

Figure 2a is the original image of digital terrain model (DTM) of a desert in central Australia. This region has a relatively low topographic relief (< 200 m). The dark colors indicate the desert whereas the light colors indicate the highlands or hills in the desert. NW-trending features are prominent whereas some subtle NE-trending structures also exist. However, with the dominance of dark colors in the desert, detailed features in the desert are hardly seen on the original image. Figure 2b shows the 2D

Cartesian spectrum of this image. The conjugated NW-and-NE trending features are reflected as alignments mixed within the elongated frequency zone in Figure 2b. Figure 2c shows that the intensity of different frequency components decreases dramatically with the increase in frequency, with the intensity of high-frequency components (>400 Rad/s) being less than 1% of the maximum intensity.

Figure 3a shows the image processed with MCDF(add1). NE and NW directions are selected as the two conjugate directions for directional filtering. The combination of $W_0 = 1$, $W_{NW} = W_{NE} = 2$, $F_1 = F_2 = 1$ generates an image on which many NE and NW linear features are outlined in the 'dark' desert. The conjugated NW-and-NE trending features are clearly reflected as alignments which separate the original elongated frequency zone into two fan-shaped sub-zones in Figure 2b. This indicates that this MCDF operation indeed enhanced the features in these conjugate directions. The total area of these 2 sub-zones is larger than that of the elongated zone in the Figure 1b, which implies the enhancement of high-frequency components contained in the original image. Figure 3c further shows that the high-frequency components have been intensified to 12% of the maximum intensity. The intensity of medium-frequency components is also increased whereas the expected low-frequency components are retained with the same intensity.

4 Discussions and Conclusion

To verify that the MCDF(add1) operation indeed enhances the conjugated features in both NE and NW directions in the image, comparison can be made between the two spectra of the original and MCDF(add1) images (Fig. 2b & Fig. 3b). In the spectrum of the original image, NE and NW trending information is mixed with other components in an elongated frequency zone, distinguished by locally 'light-colored' alignments in these two conjugated directions from other components. However, in the spectrum of the MCDF(add1) image, NE and NW trending information is distinguished by clear 'light-dark' margins along these two conjugated directions, which indicates the significant separation between the enhanced high-frequency NE and NW trending features from their surrounding low-frequency components. As expected, the MCDF(add1) image (Fig. 3a) indeed has shown more NE and NW trending information than that in the original image (Fig. 2a).

To verify that the MCDF(add1) operation not only enhances the conjugated features in both NE and NW directions in the image, but also retains the low-frequency information in the original image, we use the statistical results of radial spectra of both the original and MCDF(add1) images (Table 1) to outline the facts. It is evident that the MCDF(add1) operation has enhanced the highest-frequency component by 9 times from its relative intensity of 0.5% in the original image to 4.5% in the MCDF(add1) image. This is achieved by keeping almost no change in the maximum intensity and standard deviation in both images, which means that there is almost no loss in low-frequency components in the MCDF(add1) image. The medium-frequency components are also intensified from 6.3% in the original image to 16.9% in the MCDF(add1) image, an increase of 2.7 times. By keeping the same low-frequency components, bringing a moderate increase in medium-frequency components, and elevating high-frequency components by at least 9 times, all together the MCDF(add1) operation makes not only features in the NE and NW directions in the

MCDF(add1) image look more prominent, but also the whole image appear richer in contrast depth and thus more smooth.

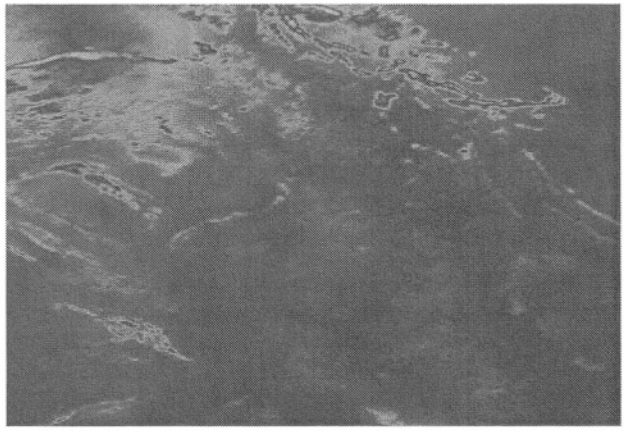
Table 1. Statistics of radial spectra of the original and MCDF(add1) images

Statistics	Original image		MCDF(add1) image	
	Absolute	Relative (x/Max)	Absolute intensity	Relative (x/Range)
Min (high-frequency components)	826	0.5%	7446	4.5%
Max (low-frequency components)	164359	100%	164345	100%
Median (medium-frequency components)	10372	6.3%	27810	16.9%
Mean (average frequency components)	22267	13.5%	34883	21.2%
Std	26299	16%	25492	15.5%

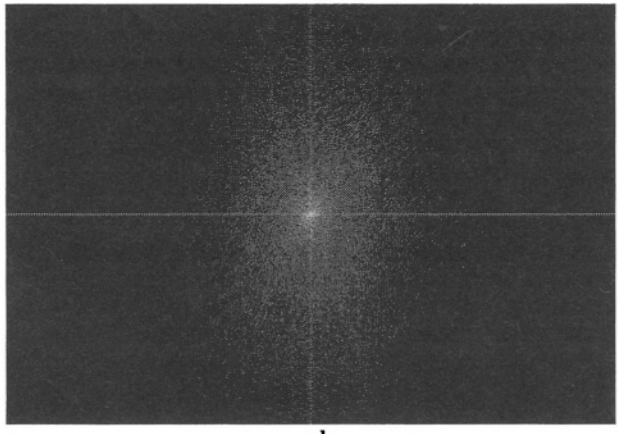
Our FFT analysis on the DTM image proves that the MCDF(add1) operation indeed overcomes the two weaknesses of using the conventional directional filtering in image processing, i.e., separation in presentation of processed results in different directions, and significant loss in low-frequency components. Although the results of using MCDF(add1) are presented here only, tests over other MCDF operations also reveal the similar results (Table 2). Therefore, the MCDF method is worth for further development.

Table 2. Statistics of radial spectra of an aerial photograph and its MCDF(add2) image

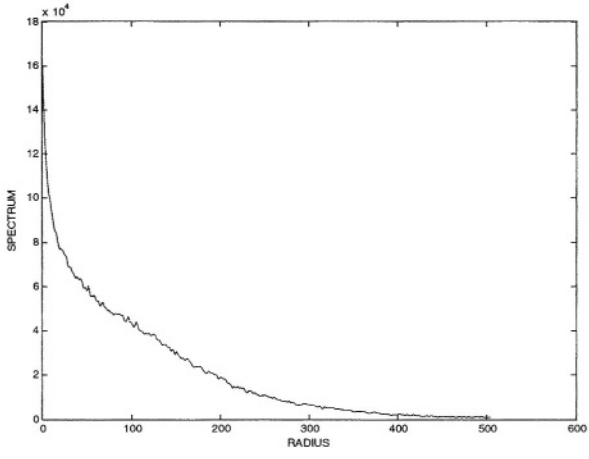
Statistics	Original image		MCDF(add1) image	
	Absolute	Relative (x/Max)	Absolute intensity	Relative (x/Range)
Min (high-frequency components)	12297	7.8%	32974	20.9%
Max (low-frequency components)	158111	100%	158105	100%
Median (medium-frequency components)	36711	23.2%	53042	33.5%
Mean (average frequency components)	43556	27.5%	57933	36.6%
Std	28270	17.9%	22712	14.4%



a

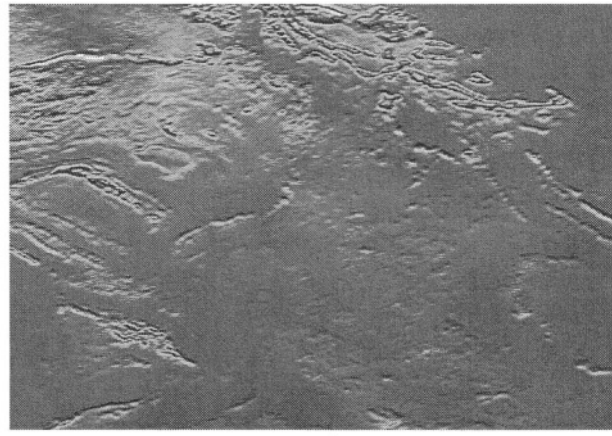


b

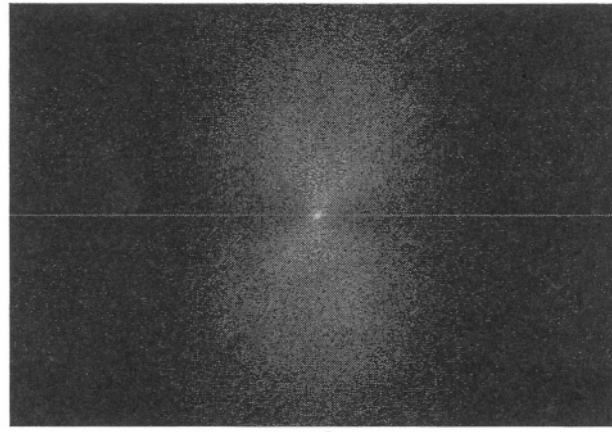


c

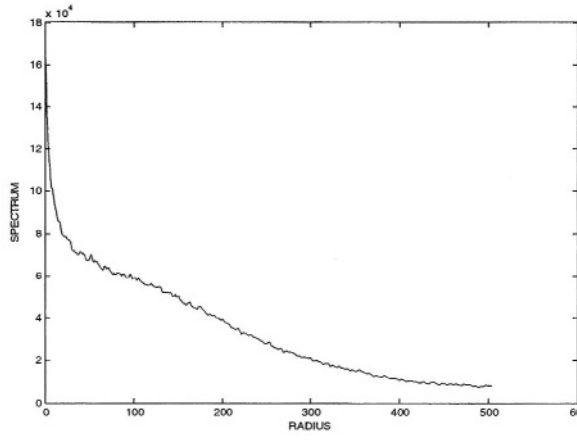
Fig. 2. Original image (a), 2D spectrum (b), and radial spectrum (c)



a



b



c

Fig. 3. MCDF(add1) image (a), 2D spectrum (b), and radial spectrum (c)

Acknowledgement. We are grateful to the Northern Territory Geological Survey of Department of Mines and Energy of Australia for providing us with the DTM data. The Faculty of Communication, Health and Science of the Edith Cowan University is thanked for supporting this research project. The constructive comments made by the anonymous referees are acknowledged.

References

1. Guo, W., Watson, A.: Modification of Conjugate Directional Filtering: from CDF to MCDF. Proceedings of IASTED Conference on Signal Processing, Pattern Recognition, and Applications. Crete, Greece (2002) 331-334.
2. Guo, W., Watson, A.: Conjugated Linear Feature Enhancement by Conjugate Directional Filtering. Proceedings of IASTED Conference on Visualization, Imaging and Image Processing. Marbella, Spain (2001) 583-586.
3. Watson. A., Guo, W.: Application of Modified Conjugated Directional Filtering in Image Processing. Proceedings of IASTED Conference on Signal Processing, Pattern Recognition, and Applications. Crete, Greece (2002) 335-338.
4. Jahne, B.: Digital Image Processing: Concepts, Algorithms and Scientific Applications. Springer-Verlag, Berlin Heidelberg (1997).
5. Proakis, J.G., Manolakis, D.G.: Digital Signal Processing: Principles, Algorithms and Applications. Prentice-Hall, Upper Saddle River New York (1996).
6. Richards, J.A.: Remote Sensing Digital Image Analysis. Springer-Verlag, Berlin Heidelberg (1993).
7. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, Prentice Hall (2002).
8. Hanselman, D., Littlefield, B.R.: Mastering MATLAB 6. Prentice Hall (2001).
9. Phillips, C.L., Parr, J.M., Riskin, E.A.: Signals, Systems, and Transforms. Prentice Hall (2003).

An Efficient Perspective Projection Using VolumePro™

Sukhyun Lim and Byeong-Seok Shin

Inha University, Department of Computer Science and Engineering
253 Yonghyeon-Dong, Nam-Gu, Inchon, 402-751, Korea
g2011498@inhavision.inha.ac.kr, bsshin@inha.ac.kr

Abstract. VolumePro is a real-time volume rendering hardware for consumer PCs. It cannot be used for the applications requiring perspective projection such as virtual endoscopy since it provides only orthographic projection. Several methods have been proposed to approximate perspective projection by decomposing a volume into slabs and applying successive orthographic projection to them. However it takes a lot of time since entire region of each slab should be processed, even though some of the region does not contribute to final image. In this paper, we propose an efficient perspective projection method that exploits several subvolumes with cropping feature of VolumePro. It reduces the rendering time in comparison to slab-based method without image quality deterioration since it processes only the parts contained in the view frustum.

1 Introduction

Volume rendering is a visualization method of displaying volumetric data as a two-dimensional image [1]. However it is hard to achieve interactive speed since it requires large amount of computation. For this reason VolumePro hardware was released in 1997 by Misitubishi Electric [2]. It provides real-time rendering on standard PC platform. One drawback of VolumePro is that it does not produce perspective projection images. Although an algorithm to simulate perspective projection using parallelly projected slabs in VolumePro API was presented [3,4], it takes long time since some of the entire region that does not belong to the view frustum should be processed. In this paper, we propose an efficient method to approximate perspective projection using subvolume feature in VolumePro. VolumePro can subdivide a volume data into subvolumes with a size less than or equal to the original volume. Our method renders subvolumes located in view frustum instead of the entire volume.

Direct volume rendering produces an image directly from the volumetric data [5]. Several optimization methods are devised for reduce rendering time, and they can be classified into software-based and hardware-based approaches. The software accelerated techniques usually require additional storage and preprocessing [6,7]. Hardware accelerated techniques achieve interactive speed on a specified workstation, but it is difficult to incorporate those techniques into a standard PC platform [8,9,10].

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 396–403, 2004.
© Springer-Verlag Berlin Heidelberg 2004

VolumePro is a hardware implementation of raycasting using shear-warp factorization. It provides real-time rendering (up to 30 fps) with compositing, classification and shading. One drawback of this hardware is that it does not produce perspective projection images. For this reason, some methods to approximate perspective projection from several parallel projected slabs are presented. These methods generate a series of slabs, which are parts of the volume data in-between two cutting planes orthogonal to the viewing direction. Then intermediate images for those slabs are generated, scaled and clipped with regard to FOV(field of view) angle. All of these images are blended to make a final image. This method is efficient when the thickness of slab is properly determined. However processing time may increase since the outside region of view frustum should be rendered for each slab.

In section 2, we briefly review the slab-based method. In section 3, we explain our method that uses subvolume in detail and describe advantages of subvolume method over previously-proposed slab-based method. Also we describe the analysis of our algorithm in comparison to conventional method. Experimental results are presented in section 4. Finally we conclude our work in the last section.

2 Projected Slab-Based Algorithm

VolumePro provides two features to generate partial volumes : a slab and a subvolume. A slab is a part of volume data defined in between two orthogonal planes and a subvolume is a partial volume against entire volume. Figure 1 shows an example of a slab and a subvolume respectively.

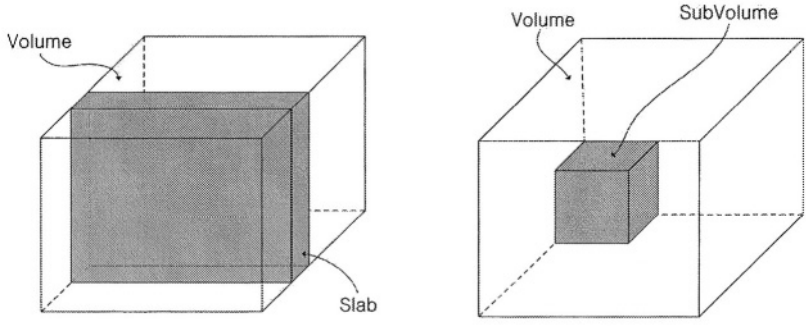


Fig. 1. An example of a slab (left) and a subvolume (right)

Figure 2 depicts the rendering process using slabs for approximating perspective projection. At first, we divide an entire volume into consecutive slabs according to viewing condition (figure 2(a)). We have to specify the plane normal, distance from camera position and thickness to define a slab as shown in figure 3.

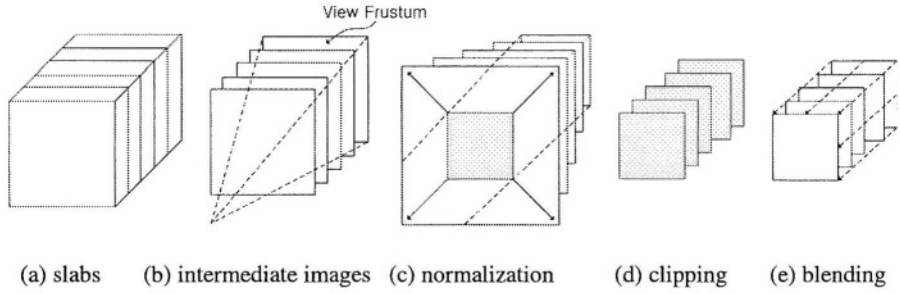


Fig. 2. Process of slab-based rendering for approximating perspective projection : (a) subdivide a volume into several slabs (b) make intermediate images for all slabs (c) normalize intermediate images according to distance from view point (d) clip those images against parallel view volume (e) blend the images to generate final image

Create(double *A, B, C, D*, double *Thickness*);
(*A, B, C*) is a normal vector
D is a distance from origin
Thickness is the distance between the two parallel planes

Fig. 3. VolumePro API used to set a slab

Next, intermediate images are generated by orthographic projection in each slab (figure 2(b)). The images are normalized (scaled) considering the distance from camera to corresponding slab (figure 2(c)). The normalized images are clipped against parallel view volume (figure 2(d)) and blended them to make final image using texture blend function in graphics hardware (figure 2(e)).

Figure 4 shows how to operate the slab-based projection algorithm. Let a thickness of *n*-th slab be Z_n and a distance from camera position to the first slab be D_0.

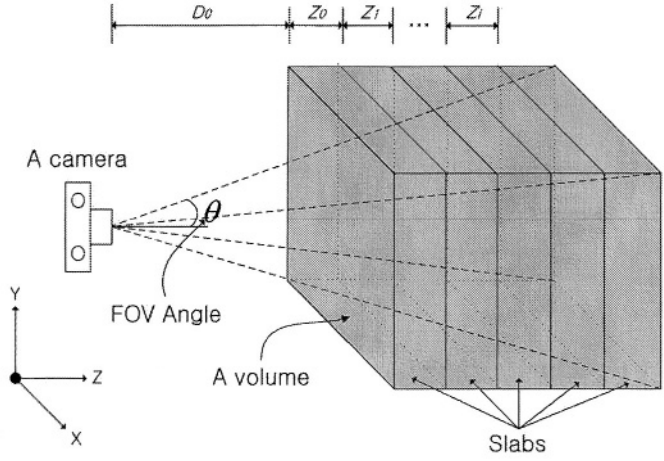


Fig. 4. Slab-based rendering using VolumePro API

3 Projected Subvolume-Based Algorithm

It is possible to set the VOI (volume of interest) region using subvolume feature in VolumePro. Figure 5 illustrates the subvolumes contained in current view frustum where Z_n is a thickness of n-th subvolume.

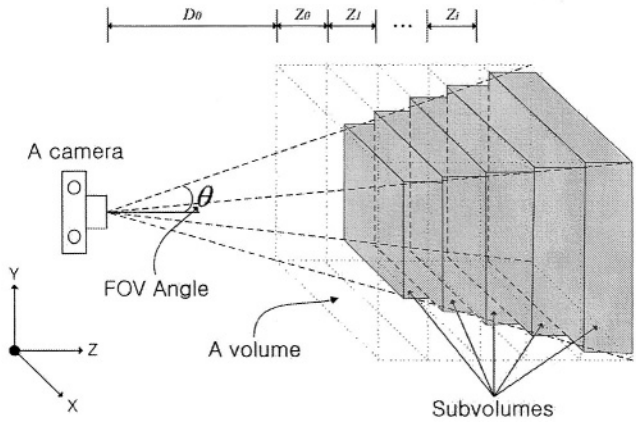


Fig. 5. Subvolume-based rendering using VOI feature in VolumePro

Gray boxes indicate regions of which the voxels are processed and transparent regions are discarded from rendering in figure 5. Therefore rendering time is much shorter than that of slab-based method and the memory for storing intermediate image can be reduced.

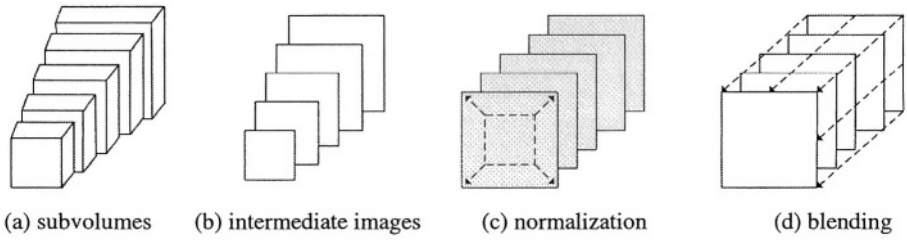


Fig. 6. Process of subvolume-based rendering : (a) subdivide a volume into several subvolumes (b) make intermediate images for all subvolumes (c) normalize intermediate images to fit into parallel view volume (d) blend the images to generate final image

Figure 6 shows the rendering steps using subvolume feature. After deciding the camera position and orientation, we make several subvolumes contained in view frustum (figure 6(a)). We can define a subvolume using VLICrop() API of VolumePro API as shown in figure 7.

VLICrop(double *Xmin, Xmax, Ymin, Ymax, Zmin, Zmax*);
(*Xmin, Xmax*), (*Ymin, Ymax*), (*Zmin, Zmax*) are the minimum and maximum values of a subvolume in *x*-, *y*- and *z*-coordinates

Fig. 7. VolumePro API used to set a subvolume

To specify the minimum and maximum values in principal axes, the distance from camera to *i*-th subvolume (D_i), width and height of the subvolume ($Width_i$ and $Height_i$) should be calculated as shown in Equation (1). ΔZ_i means a thickness of *i*-th subvolume.

$$\begin{aligned} D_i &= D_0 + \sum_{n=0}^{i} \Delta Z_n \\ Width_i &= 2\tan\theta(D_{i-1} + \frac{\Delta Z_i}{2}) \\ Height_i &= 2\tan\gamma(D_{i-1} + \frac{\Delta Z_i}{2}) \end{aligned} \tag{1}$$

In second step, intermediate images are generated by rendering of subvolumes (figure 6(b)) as shown in Figure 8.

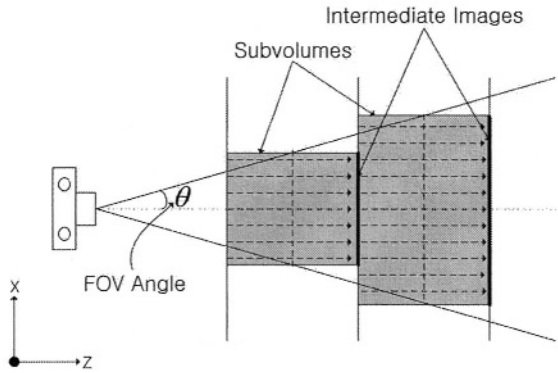


Fig. 8. How to generate intermediate image for each subvolume

Next, the intermediate image is normalized according to distance from view point (figure 6(c)). While slab-based method should clip intermediate images against parallel view volume after scaling them, all the pixels in an intermediate image are contributed to final image in our subvolume-based method, so clipping is not necessary. Final image is generated by blend function in graphics hardware just as in slab-based method (figure 6(d)).

When we use slab-based method, rendering time is composed of five components as follows :

$$T_{slab} = t_{sg}^{sl} + t_{im}^{sl} + t_{no}^{sl} + t_{cl}^{sl} + t_{bl}^{sl}\,. \tag{2}$$

where t_{sg}^{sl}, t_{im}^{sl}, t_{no}^{sl}, t_{cl}^{sl}, and t_{bl}^{sl} are the time for slab generation, intermediate image generation, normalization, clipping time, and blending.
In subvolume-based method, rendering time can be defined as follows :

$$T_{sub} = t_{sg}^{sv} + t_{im}^{sv} + t_{no}^{sv} + t_{bl}^{sv} . \tag{3}$$

where t_{sg}^{sv}, t_{im}^{sv}, t_{no}^{sv}, and t_{bl}^{sv} are the time for subvolume generation, intermediate image generation, normalization, and blending.

Slab generation cost is almost the same as subvolume generation cost since two functions are performed on VolumePro hardware ($t_{sg}^{sl} \bullet t_{sg}^{sv}$). Intermediate image generation time of slab-based method is longer than that of subvolume-based method since intermediate image using slab-based method is generated not for partial volume but for entire volume ($t_{im}^{sl} > t_{im}^{sv}$). Normalization cost of slab-based method is more expensive than that of subvolume-based method since normalization step in slab-based method generates image generated for entire volume ($t_{no}^{sl} > t_{no}^{sv}$). Blending time of slab and subvolume method is the same ($t_{bl}^{sl} = t_{bl}^{sv}$). Consequentially, in general, total rendering time of subvolume-based method is less costly than that of slab-based method ($T_{slab} > T_{sub}$).

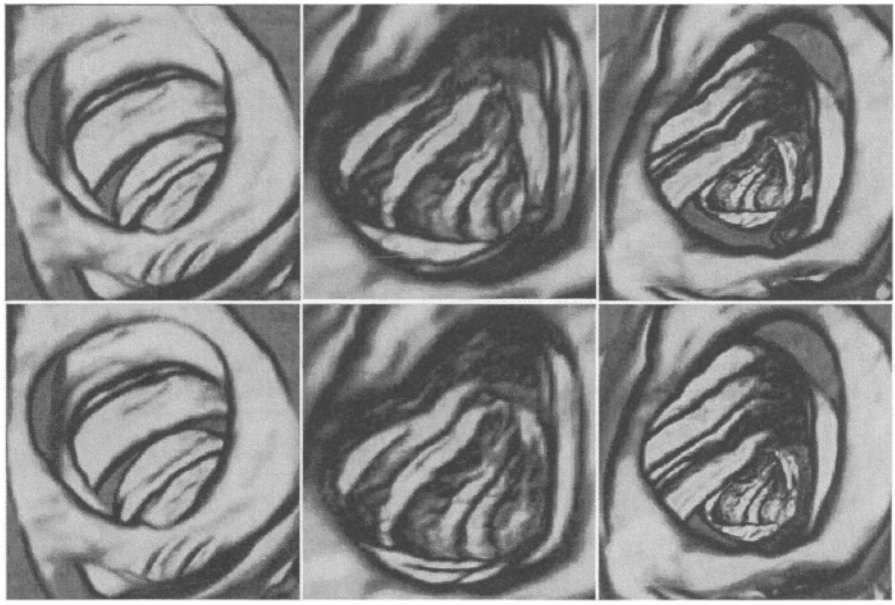


Fig. 9. Comparison of the quality of images rendered by the slab-based method (upper row) and our method (bottom row) in several positions in human colon.

4 Experimental Results

We compare the rendering time and image quality of the conventional slab-based method and our method. Both methods were implemented on a 1.7GHz Pentium IV PC with 1GB of main memory. We use VolumePro 500 model with 256MB voxel memory. The volume data used for the experiment is a clinical volume obtained by scanning a human abdomen with resolution 256 × 256 × 256.

Figure 9 shows a comparison of the quality of image produced by both methods. The thickness of a slab and a subvolume (ΔZ_i) is 32 voxels, and FOV angles (θ and γ) are 30 degrees. Image resolution is 400×400 and it performs two-times super-sampling in the z-direction. Comparing the images generated by two methods, it is not easy to distinguish image quality.

Table 1 denotes the rendering time to get a final image in both methods. Sub-volume-based method is faster than slab-based approach. When we approximated perspective projection by using slabs or subvolumes, the smaller the thickness of slab or subvolume, the more realistic results we can get. However the rendering time is inversely proportional to the thickness. Table 1 shows that the processing time of subvolume-based method is shorter than that of slab-based method in any cases. However the difference of rendering time in-between two method becomes smaller as the thickness decreases. Therefore we have to choose appropriate value of thickness. According to experimental results, the thickness of 20~90 voxels is a proper selection to approximated perspective projection.

Table 1. Comparison of rendering time (FOV angle = 20°).

Thickness(voxels) / Method	256	120	90	70	60	40
A. Slab-based (msec)	117	156	167	197	234	313
B. Subvolume-based (msec)	47	78	117	153	198	280
B/A (%)	**59.9**	**50.1**	**29.8**	**22.3**	**15.4**	**10.5**

Thickness(voxels) / Method	30	20	10	5	2	1
A. Slab-based (msec)	392	553	1063	2086	5128	10124
B. Subvolume-based (msec)	353	514	1020	2042	5081	10063
B/A (%)	**10.0**	**7.2**	**4.0**	**2.1**	**0.9**	**0.6**

5 Conclusion

Since VolumePro hardware provides only orthographic projection, we cannot apply it to applications demanding perspective projection. Although some approaches to approximate perspective projection using parallelly projected slabs have been presented, it takes a lot of time since the entire region that does not belong to view frustum should be processed. In this paper we present an efficient method to approximate perspective projection using subvolume. To approximate perspective projection, we make several subvolumes with cropping feature of VolumePro. We conclude that our method is faster than slab-based method when we set the thickness as 20~90 voxels.

Acknowledgment. This work was supported by grant No. R05-2002-000-00512-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

References

1. Yagel, R.: Volume Viewing: State of the Art Survey, SIGGRAPH 97 Course Note 31, (1997)
2. Pfister, H., Hardenbergh, J., Knittel, J., Lauer, H. and Seiler, L.: The VolumePro Real-Time Ray-Casting System, Proceedings of SIGGRAPH 99, Los Angeles, CA, (1999) 251-260
3. Vilanova, A., Wegenkittl, R., Konig, A. and Groller, E. : Mastering Perspective Projection through Parallelly Projected Slabs for Virtual Endoscopy, SCCG'01-Spring Conference on Computer Graphics, (2001) 287-295
4. K. Kreeger, W. Li, S. Lakare, and A. Kaufman: Perspective Virtual Endoscopy with VolumePro Parallel Rendering, http://www.cs.sunysb.edu/~vislab/, (2000)
5. Levoy, M.: Display of Surfaces from Volume Data, IEEE Computer Graphics and Applications, Vol. 8, No. 3, (1988) 29-37
6. Yagel, R. and Kaufman, A.: Template-based volume viewing, Computer Graphics Forum (Eurographics 92 Proceedings), Cambridge, UK, (1992) 153-167
7. Lacroute, P. and Levoy, M.: Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation, Computer Graphics (SIGGRAPH 94 Proceedings), Orlando, Florida, (1994) 451-458
8. Westermann, R. and Eart, T.: Efficiently Using Graphics Hardware in Volume Rendering Applications, Computer Graphics, (1998) 167-177
9. Yagel, R., Kaufman, A., Cabral, B., Cam, N. and Foran, J.: Acclerated volume rendering and tomographic reconstruction using texture mapping hardware, Symposium on Volume Visualization, (1994) 91-97
10. Ma, K., Painter, J., Hansen, C. and Krogh, M.: A data distributed, parallel algorithm for ray-traced volume rendering, Proceedings of the 1993 Parallel Rendering Symposium, San Jose, CA, (1993) 15-22

Reconstruction of 3D Curvilinear Wireframe Model from 2D Orthographic Views

Aijun Zhang[1], Yong Xue[1,2*], Xiaosong Sun[1], Yincui Hu[1], Ying Luo[1], Yanguang Wang[1], Shaobo Zhong[1], Jianqin Wang[1], Jiakui Tang[1], and Guoyin Cai[1]

[1]Laboratory of Remote Sensing Information Sciences, Institute of Remote Sensing Applications, Chinese Academy of Sciences, P. Box 9718, Beijing 100101, China,
aj.zhang@263.net
[2]Department of Computing, London Metropolitan University, 166-220 Holloway Road, London N7 8DB, UK
y.xue@londonmet.ac.uk

Abstract. An approach for reconstructing wireframe models of curvilinear objects from three orthographic views is discussed. Our main stress is on the method of generating three-dimensional (3D) conic edges from two-dimensional (2D) projection conic curves, which is the pivotal work for reconstructing curvilinear objects from three orthographic views. In order to generate 3D conic edges, a five-point method is firstly utilized to obtain the algebraic representations of all 2D projection curves in each view, and then all algebraic forms are converted to the corresponding geometric forms analytically. Thus the locus of a 3D conic edge can be derived from the geometric forms of the relevant conic curves in three views. Finally, the wireframe model is created after eliminating all redundant elements generated in previous reconstruction process. The approach extends the range of objects to be reconstructed and imposes no restriction on the axis of the quadric surface.

1 Introduction

Automatic conversion from 2D engineering drawings to solid models which allows existing engineering drawings to be fully used for newer designs is an important research topic in computer graphics and CAD. Much work has already been done on automatically reconstructing solid models from orthographic views to date[1-6]. However, the existing approaches have some limitations, which hinder the work from developing further. One of the major limitations is the narrow range of solid objects which can be generated from 2D views. The earlier work was able to generate planar polyhedral objects whose projections contain only lines[1,2]. Most of the later research has concerned with extending the range of objects to be reconstructed. Sakurai[3], Gu[4], Lequette[5], and Kuo[6] extended the earlier method to deal with curved surfaces. However, they restricted the orientation of the quadric surface to be parallel to one of the coordinate axes or projection planes.

In this paper, we propose a wireframe-oriented approach that can handle a wider variety of manifold objects with curved surfaces and conic edges than existing

* Corresponding author

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 404–411, 2004.
© Springer-Verlag Berlin Heidelberg 2004

methods, and impose no restrictions on the orientation of the quadric surface. A key idea of our work is to utilize five-point method to generate 3D conic edges, which was employed by Kuo. Nevertheless, our method differs from that of Kuo's somewhat. The difference will be discussed in section 3.2.

2 Preprocessing of the Input Drawing

The input drawing consists of three orthographic views of the objects. i.e., front view, top view, and side view. Only the necessary geometric elements in the drawing are taken into consideration, and three views have been separated and identified[6]. In addition, auxiliary lines must be added to the views as the corresponding projections of tangent edges and/or silhouette edges for curved objects.

2D points and segments in each view are stored in *P_list*(*view*) and *S_list*(*view*), respectively, where *view* is front, top, or side. Each item of *P_list*(*view*) holds coordinate value and type. Each item of *S_list*(*view*) holds indices of its two endpoints and type.

3 Generation of Wireframe

Generating wireframe is the early stage of reconstruction using wireframe-oriented approach, during which all the possible 3D elements constituting the wireframe are recovered from the 2D elements in three views. We call the 3D elements candidate vertices, candidate edges, and candidate faces during the reconstruction, written as c-vertices, c-edges, and c-faces, respectively.

3.1 Generation of 3D Candidate Vertices

We use the method to generate 3D c-vertices which has been detailed in previous work [1-6]. The c-vertices are stored in *V_list*(), each item of which holds coordinate values, corresponding points in each views, associated 3D edges, and type.

3.2 Generation of 3D Candidate Edges

In this section, we focus on the method for generating 3D conic edges, which is based on some important properties of the conic section under orthographic projection.

3.2.1 Construction and Representation of Projection Conics

Conic curves are defined in two ways in general: implicitly and parametrically. In solid modeling, a common way of representing a conic is by its more intuitive geometric parameters[7], so this representation form, unlike the implicit one, requires a different definition for each type of conic. In most cases, however, the type of a projection curve is not directly available from the views, for there are no explicit

notations for it. Hence it is much difficult to derive the straightforward geometric representation for each type of conic curve. In our approach, therefore, the five-point method is employed to obtain the algebraic equation of a 2D projection conic curve firstly, and then the algebraic form is converted to the geometric representation analytically, seeing that it is more easy to obtain the algebraic equation of a conic curve and this form can be conveniently converted to the geometric representation. While in Kuo's approach, the geometric representation of 2D projection conic curves are obtained directly, which needs more complex geometric computation because no more definite conic types are available from the views.

We begin with describing the general equation for a 2D conic curve in the projection plane coordinates. Basically, the 2D coordinate systems associated with the three projection planes are denoted as *x-y* for top view, *x-z* for front view, and *y-z* for side view, respectively. Without loss of generality, suppose $c_t(p_1,p_2)$ is a curve in top view, in which $p_1(x_1,y_1)$ and $p_2(x_2,y_2)$ are the endpoints of the curve. We regard it as a conic on the basis of the theorems of the conics under orthographic projection [8].

The conic curve c_t in top view can be described by an algebraic expression:

$$a_{11}x^2 + 2a_{12}xy + a_{22}y^2 + 2a_{13}x + 2a_{23}y + a_{33} = 0 \tag{1}$$

In order to construct this conic curve, we choose arbitrary three points excluding two endpoints p_1 and p_2, say $p_i(x_i,y_i)$, i=3,4,5. The five points, $p_i(x_i,y_i)$, i=1,2,...,5, in which no three points are collinear, uniquely determine the projection conic. The conic coefficients in Eq.(1) are the solution to

$$\mathbf{P}\begin{bmatrix} a_{11} & a_{12} & a_{22} & a_{13} & a_{23} & a_{33} \end{bmatrix}^{\mathrm{T}} = 0$$

where

$$\mathbf{P} = \begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 & 2x_1 & 2y_1 & 1 \\ x_2^2 & 2x_2y_2 & y_2^2 & 2x_2 & 2y_2 & 1 \\ x_3^2 & 2x_3y_3 & y_3^2 & 2x_3 & 2y_3 & 1 \\ x_4^2 & 2x_4y_4 & y_4^2 & 2x_4 & 2y_4 & 1 \\ x_5^2 & 2x_5y_5 & y_5^2 & 2x_5 & 2y_5 & 1 \end{bmatrix}$$

3.2.2 Classification of Projection Conics

The type of a conic expressed in the form of Eq.(1) is determined easily by being converted to the geometric representation according to analytic geometry [7]. The geometric parameters for the conic are derived as follows:

(1) The orientation θ of conics

$$\theta = \frac{1}{2}ctg^{-1}\frac{a_{11}-a_{22}}{2a_{12}}$$

(2) The geometric parameters for central conics

$$x_0 = \frac{a_{12}a_{23}-a_{22}a_{13}}{a_{11}a_{22}-a_{12}^2} \qquad y_0 = \frac{a_{12}a_{13}-a_{11}a_{23}}{a_{11}a_{22}-a_{12}^2}$$

$$\{a,b\}=\begin{cases}\left\{\sqrt{-a'_{33}/a'_{11}}\,,\sqrt{-a'_{33}/a'_{22}}\right\} & \text{for an ellipse}\\ \left\{\sqrt{-a'_{33}/a'_{11}}\,,\sqrt{a'_{33}/a'_{22}}\right\} & \text{for a hyperbola}\end{cases}$$

where

$$a'_{11}=\frac{a_{11}+a_{22}}{2}+\frac{a_{11}-a_{22}}{2}\cos 2\theta+a_{12}\sin 2\theta$$

$$a'_{22}=\frac{a_{11}+a_{22}}{2}-\frac{a_{11}-a_{22}}{2}\cos 2\theta-a_{12}\sin 2\theta$$

$$a'_{33}=(a_{13}x_0+a_{23}y_0)+a_{33}$$

(3) The geometric parameters for a parabola

$$p=\sqrt{-I_3/I_1^3}$$

$$x_0=\frac{c^2-2a_{23}c+a_{22}a_{33}}{2(a_{12}a_{23}-a_{13}a_{22})}$$

$$y_0=\frac{a_{11}c^2-2a_{12}a_{13}c+a_{12}^2a_{33}}{2a_{12}(a_{13}a_{22}-a_{12}a_{23})}$$

where

$$I_1=a_{11}+a_{22}\qquad I_3=\begin{vmatrix}a_{11}&a_{12}&a_{13}\\a_{12}&a_{22}&a_{23}\\a_{13}&a_{23}&a_{33}\end{vmatrix}\qquad c=\frac{a_{12}a_{13}+a_{22}a_{23}}{a_{11}+a_{22}}$$

From the above derivations, we could obtain the parameter equations of commonly utilized conic curves, e.g., elliptical curves, hyperbolic curves, and parabolic curves, easily [7].

3.2.3 Reconstruction of 3D Conics

We now consider a point matching method to determine the relationship between the three projection conic curves of a 3D conic. Let v_1 and v_2 be two different vertices and their corresponding points in three views be (pt_1, pf_1, ps_1) and (pt_2, pf_2, ps_2), respectively. A conic edge, $e(v_1, v_2)$, was found when there existed a conic connection between the two corresponding points of v_1 and v_2 in each view, viz. $c_t(pt_1, pt_2)$ between pt_1 and pt_2 in top view, $c_f(pf_1, pf_2)$ between pf_1 and pf_2 in front view, $c_s(ps_1, ps_2)$ between ps_1 and ps_2 in side view, and it satisfied Eq.(2), providing there was no internal vertices on the edge, in between v_1 and

$$|pt_i(x)\text{-}pf_i(x)|\le\varepsilon \text{ and } |pt_i(y)\text{-}ps_i(y)|\le\varepsilon \text{ and } |pf_i(z)\text{-}ps_i(z)|\le\varepsilon \qquad (2)$$

where $pt_i \in c_t(pt_1, pt_2)$, $pf_i \in c_f(pf_1, pf_2)$, $ps_i \in c_s(ps_1, ps_2)$, and i=3,4,5, ε is a tolerance introduced to allow for an inexact matching problem, considering the fact that the input data may not give an exact alignment of coordinates between each view. According to the point matching method, when the expression of each corresponding projection conic curve in three views is generated, it is easy to derive the 3D conic.

Without loss of generality, let

$$\begin{cases} x = \varphi_1(\alpha) \\ z = \varphi_2(\alpha) \end{cases} \qquad \alpha_1 \le \alpha \le \alpha_2 \tag{3}$$

and

$$\begin{cases} x = \phi_1(\beta) \\ y = \phi_2(\beta) \end{cases} \qquad \beta_1 \le \beta \le \beta_2 \tag{4}$$

be two corresponding projections of a 3D conic edge, in the front and top views, respectively. Eq.(3) and (4) share the x-axis and have the same x-coordinate. Therefore we can derive that the combination of Eq.(3) and (4) associates with the locus of a 3D conic edge.

Observe Eq.(3) and (4), in order to eliminate one of the two variants(e.g.), it is necessary to review the algebraic form of the projection conic in front view

$$f(x,z) = a_{11}x^2 + 2a_{12}xz + a_{22}z^2 + 2a_{13}x + 2a_{23}z + a_{33} = 0 \tag{5}$$

solving the z-coordinate value, then the explicit form of Eq.(5) is expressed as

$$z = \frac{-(a_{12}x + a_{23}) \pm \sqrt{(a_{12}x + a_{23})^2 - a_{22}(a_{11}x^2 + 2a_{13}x + a_{33})}}{a_{22}} \tag{6}$$

We can rewrite Eq.(6), substituting x value derived from Eq.(4) for the original x value in Eq.(3), as the following simplified form:

$$z = \phi_3(\beta) \qquad \beta_1 \le \beta \le \beta_2 \tag{7}$$

It follows that the corresponding 3D conic is the locus which satisfies

$$\begin{cases} x = \phi_1(\beta) \\ y = \phi_2(\beta) \\ z = \phi_3(\beta) \end{cases} \qquad \beta_1 \le \beta \le \beta_2 \tag{8}$$

Thus, in accordance with the types of the relevant 2D projection conic curves in three views, we can get the corresponding 3D conic depicted in the form of Eq.(8).

3.2.4 3D c-Edges Reconstruction Algorithm

In previous algorithms, the process to generate c-edges is obviously time consuming[1-6]. In this section, an accelerated method is introduced to decrease the processing time. The major steps of 3D c-edges generation procedure are as follows:

Step1. Select a 2D projection conic segment arbitrarily in one view, e.g., c_t (pt_1, pt_2) in top view, where $pt_1, pt_2 \in P_list(Top)$, $c_t \in S_list(Top)$;

Step2. Get two 3D vertices, v_1 and v_2, from the *V_list*(), whose corresponding projections in top view are pt_1 and pt_2, respectively. Then search for the corresponding 2D projections of v_1 and v_2 in another two views, i.e., the front and side views, accordingly $pf_1, pf_2 \in P_list(Front)$, $ps_1, ps_2 \in P_list(Side)$, respectively;

Step3. Examine each pair of 2D points, (pf_1, pf_2) and (ps_1, ps_2), to determine whether or not there exists a 2D curve segment connecting the two points of each pair. If it is true, it is certain that at least one 3D edge between v_1 and v_2;

Step4. In accordance with the type of c_t (pt_1, pt_2) in top view, choose the corresponding 2D curve segments, c_f (pf_1, pf_2) and c_s (ps_1, ps_2), respectively, from the remaining two views. A 3D c-edge between v_1 and v_2 then can be generated by applying the method for constructing a 3D conic discussed in previous sections.

During the above steps, each 2D segment is labeled as *examined* after being examined. In this case, the performance of constructing 3D c-edges is finished until all segments in three views are labeled as *examined.*

All 3D c-edges are stored in *E_list*(), each item of which holds two endpoints, corresponding conic segments in each of three views, parametric equation of the 3D conic containing the edge, and type.

3.3 Construction of Wireframe

There may exist some redundant elements in *V_list*() and *E_list*(), on account of the fact that the reconstruction is a process of recovering information from low dimensions to high dimensions. In the stage of wireframe generation, redundant elements generally involve overlapping edges and pathological elements[2],which may not only increase the complexity of the computation, but also introduce ambiguities in the wireframe as well as solid model generation process. Therefore, they must be eliminated from *V_list*() and *E_list*(), respectively.

Thus, we can establish the wireframe with the *V_list*()and *E_list*() containing the information of 3D vertices and edges, whose reprojections are identical with the input drawings, and that each element, i.e., vertex or edge, in the wireframe satisfies the following topological conditions:

(1) $\forall v_j \in V_list(), \exists \rho(v_j) \geq 3$;

(2) $\forall e_j \in E_list(), \exists\ v_j, v_{j+1} \in V_list()$, where v_j, v_{j+1} are two endpoints of e_j;

(3) $\forall e_i, e_j \in E_list(), \exists e_i \cap e_j = \{ v_k \}$ and $v_k \in V_list()$, or $e_i \cap e_j = \phi$.

Where $\rho(v)$ denotes the edge connectivity of the c-vertex v.

4 Implementation

Based on the method described above, the prototype implementation of reconstruction is realized in C++ . Figure 1 demonstrates the case that can be handled by our method, the implementation is restricted to three orthographic views and curvilinear objects. Figure 1(b) shows the wireframe reconstructed from a three-view engineering drawing with straight lines, circular and elliptical arcs in Figure 1(a).

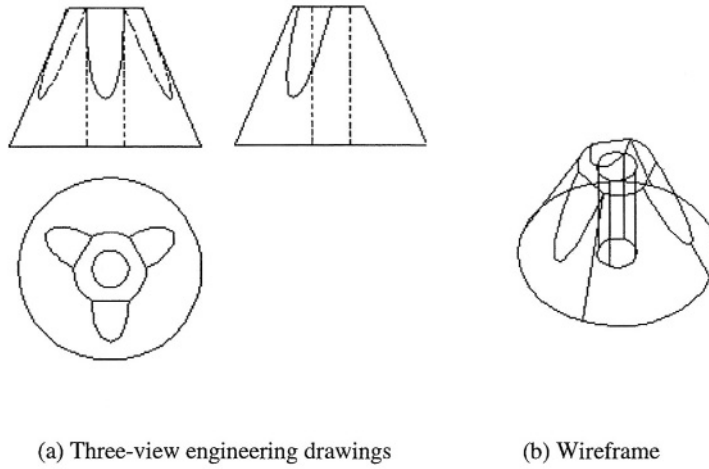


Fig. 1. The wireframe of an object reconstructed from a three-view engineering drawing with straight lines, circular and elliptical arcs

5 Conclusion

A wireframe-oriented approach to reconstruct 3D wirframes of curvilinear solids from three orthographic views is presented, in which the method to generate conic edges is emphasized. In order to obtain 3D conic edges, the five-point method is firstly applied to obtain the geometric representations of 2D projection conic curves in each view in two steps, and then 3D conic edges described in the form of coordinate locus are accordingly derived from corresponding 2D projections afterwards by using point matching method. In addition, an accelerated algorithm to generate 3D edges is introduced to decrease the processing time. At last, the wireframe is established when the redundant elements are removed from candidate vertices and edges. Our approach extends the range of objects to be reconstructed, i.e., an object may include straight lines, circular arcs, elliptical arcs, parabolic arcs, and hyperbolic arcs, and imposes no restriction on the axis of the quadric surface.

Acknowledgement. This publication is an output from the research projects "CAS Hundred Talents Program", "Digital Earth" (KZCX2-312) funded by Chinese Academy of Sciences and "Dynamic Monitoring of Beijing Olympic Environment Using Remote Sensing" (2002BA904B07-2) funded by the Ministry of Science and Technology, China.

References

1. Wesley MA, Markowsky G. Fleshing out projection. IBM Journal of Research and Development 1981; 25(6): 934-954.
2. Yan QW, Philip CL, Tang Z. Efficient algorithm for the reconstruction of 3-D objects from orthographic projections. Computer-aided Design 1994; 26(9): 699-717.
3. Sakurai H, Gossard DC. Solid model input through orthographic views. Computer Graphics 1983; 17(3): 243-25.
4. Gu K, Tang Z, Sun J. Reconstruction of 3D solid objects from orthographic projections. Computer Graphics Forum 1986; 5(4): 317-324.
5. Remi Lequette. Automatic construction of curvilinear solids from wireframe views. Computer-aided Design 1988; 20(4): 171-179.
6. Kuo MH. Reconstruction of quadric surface solids from three-view engineering drawings. Computer-aided Design 1998; 30(7): 517-527.
7. Wilson PR. Conic representations for shape description. IEEE Computer Graphics and Applications 1987; 7(4): 23-30.
8. Nalwa VS. Line-drawing interpretation: straight lines and conic sections. IEEE Transactions on Pattern Analysis and Machine Intelligence 1988; 10(4): 514-529.

Surface Curvature Estimation for Edge Spinning Algorithm*

Martin Cermak and Vaclav Skala

University of West Bohemia in Pilsen
Department of Computer Science and Engineering
Czech Republic
{cermakm|skala}@kiv.zcu.cz

Abstract. This paper presents an adaptive modification of the Edge spinning method for polygonization of implicit surfaces. The method insists on the shape of triangles and the accuracy of resulting approximation as well. The main advantages of the triangulation presented are simplicity and the stable features that can be used for next expanding. The implementation is not complicated and only the standard data structures are used. The presented algorithm is based on the surface tracking scheme and it is compared with the other algorithms which are based on the similar principle, such as the Marching cubes and the Marching triangles algorithms.

1 Introduction

Implicit surfaces seem to be one of the most appealing concepts for building complex shapes and surfaces. They have become widely used in several applications in computer graphics and visualization.

An implicit surface is mathematically defined as a set of points in space $\mathbf{x}$ that satisfy the equation $f(\mathbf{x}) = 0$. Thus, visualizing implicit surfaces typically consists in finding the zero set of *f,* which may be performed either by polygonizing the surface or by direct ray tracing. There are two different definitions of implicit surfaces. The first one [2], [3] defines an implicit object as $f(\mathbf{x}) < 0$ and the second one, F-rep [8], [10], [11], [12], defines it as $f(\mathbf{x}) \geq 0$. In our implementation, we use the F-rep definition of implicit objects.

Existing polygonization techniques may be classified into three categories. Spatial sampling techniques that regularly or adaptively sample the space to find the cells that straddle the implicit surface [2], [4]. Surface tracking approaches (also known as continuation methods) iteratively create a triangulation from a seed element by marching along the surface [1], [2], [5], [7], [9], [15]. Surface fitting techniques progressively adapt and deform an initial mesh to converge to the implicit surface, [10].

* This work was supported by the Ministry of Education of the Czech Republic - project MSM 235200002

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 412–418, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Principle of the Edge Spinning Algorithm

Our algorithm is based on the surface tracking scheme (also known as the continuation scheme, see Fig. 1) and therefore, there are several limitations. A starting point must be determined and only one separated implicit surface is polygonized for such point. Several disjoint surfaces can be polygonized from a starting point for each of them.

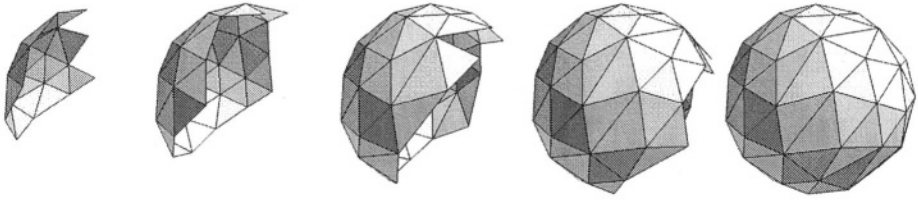


Fig. 1. Continuation scheme, new triangles are directly generated on an implicit surface.

The algorithm uses only the standard data structures used in computer graphics. The main data structure is an edge that is used as a basic building block for polygonization. If a triangle's edge lies on the triangulation border, it is contained in the *active edges list* (*AEL*) and it is called as an *active edge*. Each point, which is contained in an active edge, contains two pointers to its left and right active edge (left and right directions are in active edges' orientation).

The whole algorithm consists of the following steps (more detailed description in [5]):

1. Initialize the polygonization:
   a. Find the starting point $\mathbf{p}_0$ and create the first triangle $\mathbf{T}_0$.
   b. Include the edges $(\mathbf{e}_0,\mathbf{e}_1,\mathbf{e}_2,)$ of the first triangle $\mathbf{T}_0$ into the active edges list.
2. Polygonize the first active edge *e* from the active edges list.
3. Update the *AEL;* delete the currently polygonized active edge *e* and include the new generated active edge/s at the end of the list.
4. If the active edges list is not empty return to step 2.

3 Edge Spinning

The main goal of this work is a numerical stability of a surface point coordinates' computation for objects defined by implicit functions. In general, a surface vertex position is searched in direction of a gradient vector $\nabla\mathbf{f}$ of an implicit function *f*, e.g. as in [7]. In many cases, the computation of gradient of the function *f* is influenced by a major error that depends on modeling techniques used [8], [9], [10], [11], [13], [14]. Because of these reasons, in our approach, we have defined these restrictions for finding a new surface point $\mathbf{p}_{new}$:

- The new point $\mathbf{p}_{new}$ is sought on a circle; therefore, each new generated triangle preserves the desired accuracy of polygonization. The circle radius is proportional to the estimated surface curvature.
- The circle lies in the plane that is defined by the normal vector of triangle T_{old} and axis *o* of the current edge *e,* see Fig. 3; this guarantees that the new generated triangle is well shaped (isosceles).

3.1 Determination of the Circle Radius

The circle radius is proportional to the estimated surface curvature. The surface curvature in front of current active edge is determined in according to angle α between the surface normals $\mathbf{n}_1$, $\mathbf{n}_2$, see Fig. 2. The normal vector $\mathbf{n}_1$ is computed at point **S** that lies in the middle of the current active edge *e* and the vector $\mathbf{n}_2$ is taken at initial point $\mathbf{p}_{init}$ that is a point of intersection of the circle c_1 with the plane defined by the triangle T_{old}.

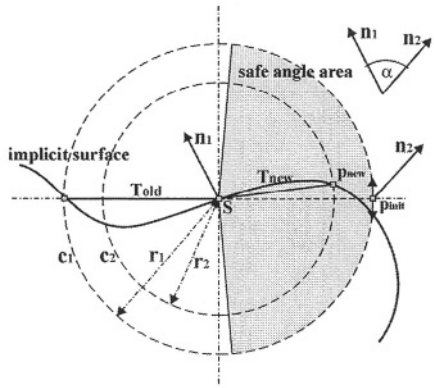


Fig. 2. The circle radius estimation.

Note that the initial radius r_1 of the circle c_1 is always the same and it is set at beginning of polygonization as the lowest desired level of detail (LOD).
The new circle radius r_2 is computed as follows.

$$r_2 = r_1 \cdot k, k \in \langle 0,1 \rangle;$$
$$k = \left(\frac{\alpha_{\lim} - \alpha \cdot c}{\alpha_{\lim}} \right), \tag{1}$$

where α_{lim} a limit angle and the constant *c* represents a speed of "shrinking" of the radius according to the angle α. To preserve well shaped triangles, we use a constant k_{min} that represents a minimal multiplier. In our implementation we used $\alpha_{min} = \pi/2$, $k_{min} = 0.2$ and $c = 1.2$.

Correction notes:

if $(\alpha > \alpha_{min})$ then $k = k_{min}$
if $(k < k_{min})$ then $k - k_{min}$

These parameters affect a shape of triangles of the polygonal mesh generated.

3.2 Root Finding

If the algorithm knows the circle radius, the process continues as follows.

1. Set the point $\mathbf{p}_{new}$ to its initial position; the initial position is on the triangle's $\mathbf{T}_{old}$ plane on the other side of the edge *e,* see Fig. 3. Let the angle of the initial position be $\alpha=0$.

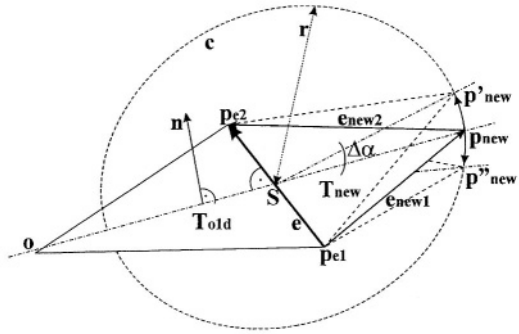


Fig. 3. The principle of root finding algorithm.

2. Compute the function values $f(\mathbf{p}_{new}) = f(\alpha)$,
   $f(\mathbf{p}'_{new}) = f(\alpha + \Delta\alpha)$ – initial position rotated by the angle $+\Delta\alpha$,
   $f(\mathbf{p}''_{new}) = f(\alpha - \Delta\alpha)$ - initial position rotated by the angle $-\Delta\alpha$; Note that the rotation axis is the edge *e.*
3. Determine the right direction of rotation; if $|f(\alpha + \Delta\alpha)| < |f(\alpha)|$ then $+\Delta\alpha$ else $-\Delta\alpha$.
4. Let the function values $f_1 = f(\alpha)$ and $f_2 = f(\alpha \pm \Delta\alpha)$; update the angle $\alpha = \alpha \pm \Delta\alpha$.
5. Check which of following case appeared:
   a) If $(f_1 \cdot f_2)<0$ then compute the accurate coordinates of the new point $\mathbf{p}_{new}$ by the binary subdivision between the last two points which correspond to the function values f_1 and f_2;
   b) If the angle $|\alpha|$ is less than α_{safe} (see *safe angle area* in Fig. 2) return to step 4.
   c) If the angle $|\alpha|$ is greater than α_{safe} then there is a possibility that both triangles $\mathbf{T}_{old}$ and $\mathbf{T}_{new}$ could cross each other; the point $\mathbf{p}_{new}$ is rejected and it is marked as *not found.*

4 Experimental Results

The Edge spinning algorithm (ES) is based on the surface tracking scheme and therefore, we have compared it with other methods based on the same principle – the Marching triangles algorithm (MTR, introduced in [7]) and the Marching cubes method (MC, Bloomenthal's polygonizer, introduced in [2]).
As a testing function, we have chosen the implicit object Genus 3 that is defined as follows.

$$f(\mathbf{x}) = r_z^4 \cdot z^2 - \left[1 - (x/r_x)^2 - (y/r_y)^2\right] \cdot \left[(x - x_1)^2 + y^2 - r_1^2\right] \cdot \left[(x + x_1)^2 + y^2 - r_1^2\right] = 0$$

where the parameters are: $\mathbf{x} = [x,y,z]^T$, r_x=6, r_y=3.5, r_z=4, r_1=1.2, x_1=3.9.

The values in Table 1 have been achieved with the desired lowest level of detail (LOD) equal 0.8. It means that maximal length of triangles' edges is 0.8. Note that there is not defined a unit of length, so that number could be for example in centimeters as well as the parameters of the function Genus 3 described above.

Table 1. Values of the object Genus 3 with the lowest level of detail LOD = 0.8

	ES	MTR	MC
# Triangles	4886	947	1056
# Vertices	2439	473	516
Avg dev.	10,99	56,80	73,28
Angle crit.	0,65	0,67	0,38
Elength crit.	0,77	0,78	0,54

The table contains the number of triangles and vertices generated. The value *Avg dev.* means the average deviation of each triangle from the real implicit surface. It is measured as an algebraic distance of a gravity centre of a triangle from an implicit surface, i.e. the function value at the centre of gravity of the triangle. Note that the algebraic distance strongly depends on the concrete implicit function; in our test, the Genus 3 object is used for all methods, so the value has its usefulness. The value *Angle crit.* means the criterion of the ratio of the smallest angle to the largest angle in a triangle and the value *Elength crit.* means the criterion of the ratio of the shortest edge to the longest edge of a triangle. The value *Avg dev.* shows the accuracy of an implicit object approximation and the adaptive ES algorithm is logically the best of tested methods. The criterions of angles and length of edges in triangles are similar for the ES and the MTR algorithms, so the both approaches generate well-shaped triangular meshes.

For visual comparison, the resulting pictures of the Genus 3 object generated in the test are in figures below. Fig. 4a shows the object generated by the adaptive algorithm, so the number of triangles generated is higher in dependence on the surface curvature. In Fig. 4b, some parts of the object are lost because the algorithm just connect nearest parts by large triangles depending of the lowest level of detail. The resulting image generated by the Marching cubes algorithm is shown in Fig. 4c. This algorithm produces badly-shaped triangles but it is fast and also stable for complex implicit surfaces with C^0 continuity, only.

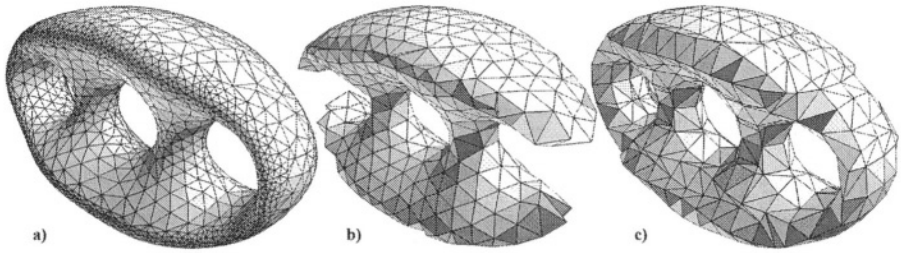


Fig. 4. The Genus 3 object generated by the a) Adaptive Edge spinning algorithm; b) Marching triangles algorithm; c) Marching cubes algorithm.

Fig. 5 shows the object modeled as intersection of two spheres. The left picture is polygonized without detection of sharp edges, and the right picture is polygonized with the edge detection principle applied to the ES method, see [6]. This object complies only the C^0 continuity and it is correctly polygonized by our method.

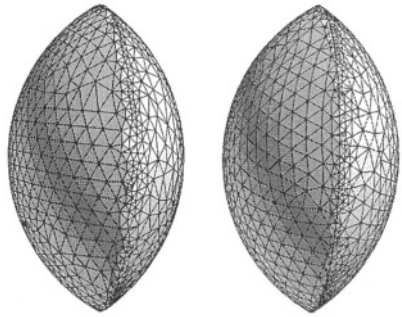


Fig. 5. Intersection of two spheres generated by the Adaptive Edge spinning algorithm

5 Conclusion

This paper presents the new adaptive approach for polygonization of implicit surfaces. The algorithm marches over the object's surface and computes the accurate coordinates of new points by spinning the edges of already generated triangles. Coordinates of the new points depend on surface curvature estimation. We used the estimation by deviation of angles of adjacent points because it is simple and fast for computation. The similar measurement has been used as curvature estimation in [16] as well. Our experiments also proved its functionality.

The algorithm can polygonize implicit surfaces which comply C^1 continuity, thin objects and some non-complex objects of C^0 continuity (an object should have only sharp edges, no sharp corners or more complex shapes). In future work, we want to modify the current algorithm for more complex implicit functions of the C^0 continuity, only.

Acknowledgement. The authors of this paper would like to thank all those who contributed to development of this new approach, especially to colleagues MSc. and PhD. students at the University of West Bohemia in Plzen.

References

1. Akkouche, S., Galin, E.: Adaptive Implicit Surface Polygonization using Marching Triangles, Computer Graphic Forum, 20(2): 67-80, 2001.
2. Bloomenthal, J.: Graphics Gems IV, Academic Press, 1994.
3. Bloomenthal, J.: Skeletal Design of Natural Forms, Ph.D. Thesis, 1995.
4. Bloomenthal,J., Bajaj, Ch., Blinn, J., Cani-Gascuel, M-P., Rockwood, A., Wyvill, B., Wyvill, G.: Introduction to implicit surfaces, Morgan Kaufmann, 1997.
5. Čermák, M., Skala, V.: Polygonization by the Edge Spinning, 16th Conference on Scientific Computing, Algoritmy 2002, Slovakia, ISBN 80-227-1750-9, September 8-13.
6. Čermák, M., Skala, V.: Detection of Sharp Edges during Polygonization of Implicit Surfaces by the Edge Spinning. Geometry and graphics in teaching contemporary engineer, Szczyrk 2003, Poland, June 12-14.
7. Hartmann, E.: A Marching Method for the Triangulation of Surfaces, The Visual Computer (14), pp.95-108, 1998.
8. "Hyperfun: Language for F-Rep Geometric Modeling", http://cis.k.hosei.ac.jp/~F-rep/
9. Karkanis, T., Stewart, A.J.: Curvature-Dependent Triangulation of Implicit Surfaces, IEEE Computer Graphics and Applications, Volume 21, Issue 2, March 2001.
10. Ohtake, Y., Belyaev, A., Pasko, A.: Dynamic Mesh Optimization for Polygonized Implicit Surfaces with Sharp Features, The Visual Computer, 2002.
11. Pasko, A., Adzhiev, V., Karakov, M., Savchenko,V.: Hybrid system architecture for volume modeling, Computer & Graphics 24 (67-68), 2000.
12. Rvachov, A.M.: Definition of R-functions, http://www.mit.edu/~maratr/rvachev/p1.htm
13. Shapiro, V., Tsukanov, I.: Implicit Functions with Guaranteed Differential Properties, Solid Modeling, Ann Arbor, Michigan, 1999.
14. Taubin, G.: Distance Approximations for Rasterizing Implicit Curves, ACM Transactions on Graphics, January 1994.
15. Triquet, F., Meseure, F., Chaillou, Ch.: Fast Polygonization of Implicit Surfaces, WSCG'2001 Int.Conf., pp. 162, University of West Bohemia in Pilsen, 2001.
16. Velho,L.: Simple and Efficient Polygonization of Implicit Surfaces, Journal of Graphics Tools, 1(2):5-25,1996.

Visualization of Very Large Oceanography Time-Varying Volume Datasets

Sanghun Park[1], Chandrajit Bajaj[2], and Insung Ihm[3]

[1] School of Comp. & Info. Comm. Engineering, Catholic University of Daegu
Gyungbuk 712-702, Korea
mshpark@cu.ac.kr
http://viscg.cu.ac.kr/~mshpark
[2] Department of Computer Sciences, University of Texas at Austin
TX 78731, USA
bajaj@cs.utexas.edu
http://www.cs.utexas.edu/~bajaj
[3]Department Computer Science, Sogang University
Seoul 121-742, Korea
ihm@sogang.ac.kr
http://grmanet.sogang.ac.kr/~ihm

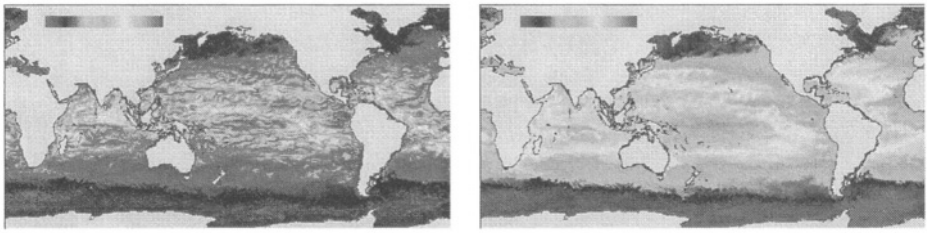
Abstract. This paper presents two visualization techniques suitable for huge oceanography time-varying volume datasets on high-performance graphics workstations. We first propose an off-line parallel rendering algorithm that merges volume ray-casting and on-the-fly isocontouring. This hybrid technique is quite effective in producing fly-through movies of high resolution. We also describe an interactive rendering algorithm that exploits multi-piped graphics hardware. Through this technique, it is possible to achieve interactive-time frame rates for huge time-varying volume data streams. While both techniques have been originally developed on an SGI visualization system, they can be also ported to commodity PC cluster environments with great ease.

1 Introduction

Understanding the general circulation of oceans in the global climate system is critical to our ability to diagnose and predict climate changes and their effects. Recently, very high quality time-varying volume data, made of a sequence of 3D volume data, were generated in the field of oceanography. The model has a resolution of 1/6 degree (2160 by 960 points) in latitude and longitude and carries information at 30 depth levels. It includes several scalar and vector field data sets at each time step: temperature, salinity, velocity, ocean surface height, and ocean depth. The datasets are from a 121 day oceanographic simulation. The time step interval is 300 seconds beginning on Feb-16-1991 at 12:00:00. Each scalar value of voxel is stored in four bytes, and the total size of the data is about 134 GB (refer to Table 1).

Usually, oceanographers have used pre-defined color-mapped images to visualize and analyze changes in an ocean. Because there is a one-to-one correspondence between colors in color maps and voxel values, these images are quite intuitive. However, they appear as a simple 2D plane, which leads to difficulties in understanding dynamic changes between the time steps. On the other hand, images created by 3D rendering techniques,

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 419–426, 2004.
© Springer-Verlag Berlin Heidelberg 2004



(a) Volume rendering scheme combined with on-the-fly isocontouring

(b) Color mapping scheme

Fig. 1. Comparison of two visualization schemes on T data

such as ray-casting and splatting, may be less intuitive in case illumination models are applied carelessly. However, SD-rendered images of high quality have an advantage that detailed changes between time steps are described effectively. Figure 1 compares two visualized images produced using the volume rendering/isocontouring method and the color-mapping method, respectively, for the temperature data.

In this paper, we discuss two different approaches designed for efficient rendering of the huge time-varying oceanography datasets on high performance graphics architectures. First, we present our off-line parallel rendering method to produce high-resolution fly-through videos. In particular, we propose a hybrid scheme that combines both volume rendering and isocontouring. Through parallel rendering, it effectively produces a sequence of high quality images according to a storyboard.

Secondly, we explain an interactive multi-piped 4D volume rendering method designed for the same datasets. Although this technique is reliant on the color-mapping method and does not create 3D rendered images of high quality, it is possible to freely control camera and rendering parameters, select a specific time step, and choose a color map in real-time through graphical user interfaces.

Table 1. Statistics on the oceanography datasets used

Name	Dataset	Attribute	Dimension	Time Step	Size (GB)
T	Temperature	Scalar	$2160 \times 960 \times 30$	121	28.0
S	Salinity	Scalar	$2160 \times 960 \times 30$	115	26.7
Vel	Velocity	Vector (U, V, W)	$2160 \times 960 \times 30$	113	78.6
PS	Surface Height	Scalar	2160×960	114	0.9

2 Offline Parallel Rendering

2.1 Basic Algorithm

Our off-line parallel volume rendering of a data stream was designed to quickly produce high-resolution fly-through movies using parallel processors. It is based on our parallel

rendering algorithm that tried to achieve high performance by minimizing, through data compression, communications between processors during rendering [1]. The algorithm was also extended to uncompressed data streams as input data and implemented on a Cray T3E, SGI Onyx2, and a PC cluster. In our parallel scheme, the image screen is partitioned into small, regularly spaced pixel tiles, which form a pool of tasks. During run time, processors are assigned to tiles from the pool of tasks waiting to be executed. The processors perform a hybrid rendering technique, combining volume rendering and isocontouring as explained below, repeatedly on tiles until the task pool is emptied. Load balancing is carried out dynamically during rendering.

2.2 Combining Volume Rendering and Isocontouring

It is very useful to extract isosurfaces at chosen data values to analyze information in volume data, but isocontouring of large volume data generates a large number of isocontour polygon sets. We designed a new rendering algorithm, merging volume ray-casting and isocontouring. In this scheme, isocontour extraction during the rendering process does not require large sets of polygons to be created and stored. When cast rays intersect volumes and isosurfaces, shading and composing are properly applied on the fly. The algorithm takes advantage of visualizing interesting materials and isosurfaces simultaneously, and using higher order interpolation and approximation. In our implementation, the ocean floors and continents were rendered by isocontouring at a selected function value, and the other parts were visualized by volume ray-casting (see Figure 1 (a)).

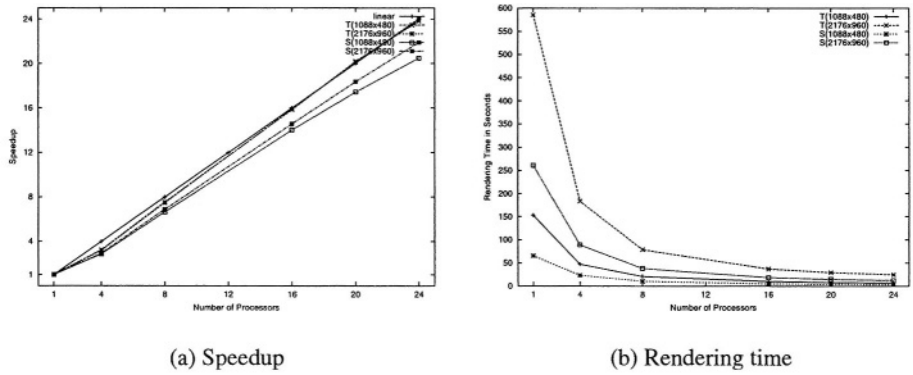
To create meaningful, high quality 3D images that match pre-defined color maps, we modified Phong's illumination model, which determines the final colors from the sum of three components, ambient color, diffuse, and specular: $I = k_a I_a + \sum_{i=1}^{n} I_{l_i}\{k_d(N \cdot L_i) + k_s(N \cdot H_i)^{n_s}\}$. In this basic illumination formula, ambient color $k_a I_a$ is ignored and diffuse and specular coefficients k_d and k_s are substituted for a color k_c from a pre-defined color map according to the data values at the sampling points. The effects showing 3D appearance and dynamic changes between time steps are demonstrated well through the calculation of the gradient N used in diffuse and specular terms: $\overline{I} = \sum_{i=1}^{n} I_{l_i} k_c\{N \cdot L_i + (N \cdot H_i)^{n_s}\}$.

In traditional volume rendering, it is important to use appropriate transfer functions based on gradient magnitude and functional values, as materials are visualized with specific ranges. In the visualization of oceanography data, defining the transfer functions is not a critical issue because all voxels are rendered regardless of their values. We used uniform transfer functions defined over whole values. To minimize mismatches between colors in the final images and the pre-defined color maps, caused by improper composition, our algorithm maintains very dense sampling intervals and uses the 'over' operator for color composition.

Figures 1 (a) and (b) were created by the proposed rendering method and the color-mapping method, respectively, where the first time step volume of temperature T data at a depth of 100 meters was visualized. Although there are some mismatches between colors of the ocean and the color map in Figure 1 (a), the figure clearly shows dynamic changes in temperature when a movie for all time steps is played.

2.3 Results of Offline Parallel Rendering

Here, we only present results for uncompressed data to avoid the degradation resulting from using compressed volume data. The MPI (Message Passing Interface) toolkit was



(a) Speedup (b) Rendering time

Fig. 2. Results on speedup and rendering time

used as an interprocessor communication library, and timings were taken in seconds for generating 2176 × 960 and 1088 × 480 perspective images using 32 × 32 tiles on a SGI Onyx2 with 24 processors and 25 GB of main memory. Figure 2 shows the performances of the average speedup and rendering times on T and S data for one time step. The data loading times from secondary disks to main memory were not considered in these results. When 24 processors were used, it took 3.19 seconds to render a 1088 × 480 image and 6.42 seconds to render a 2160 × 960 image, both of S data. On the other hand, it took 65.33 and 260.96 seconds to generate the same images on a uniprocessor. The primary reason for the increased speed is that our scheme minimizes the data communication overhead during rendering. Only communication for task assignments and image segment collection is necessary.

3 Interactive Multi-pipe 4D Rendering

3.1 Implementation Details

We have also developed an effective multi-pipe rendering scheme for the visualization of time-varying oceanography data on the SGI Onyx2 system that has six InfiniteReality2 graphics pipes with multiple 64MB RM9 Raster Managers. The graphics system can be tasked to focus all pipelines on a single rendering window, resulting in near-perfect linear scalability of visualization performance [2]. It is optimized for rendering polygon-based geometric models, not for visualizing volume data. Most large volume datasets contain much more voxels than can be stored in texture mapping hardware. To take advantage of graphics hardware acceleration, volume datasets are partitioned into sub-volumes called bricks. A brick is a subset of voxels that can fit into a machine's texture mapping hardware. In general, the optimal size of bricks is determined by various factors such as texture memory size, system bandwidth, and volume data resolution. In our application, the size of the bricks is dynamically determined to minimize texture loading.

Texture-based volume rendering includes a setup phase and a rendering phase (see Figure 3 (a)). The setup phase consists of volume loading and data bricking whose computational costs depend on disk and memory bandwidth. This process does not

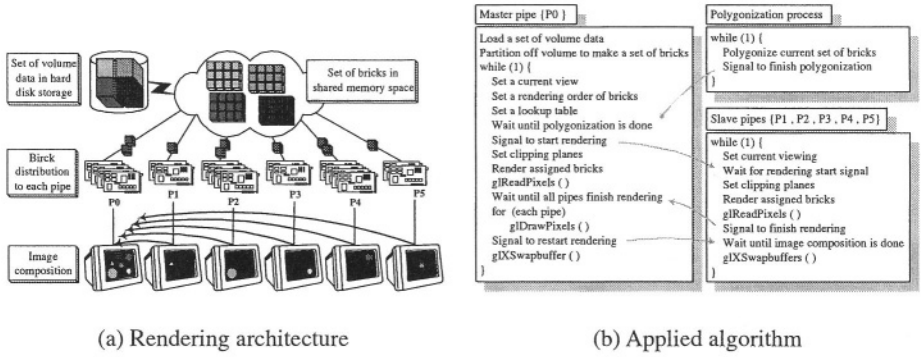


Fig. 3. Our multi-pipe visualization scheme

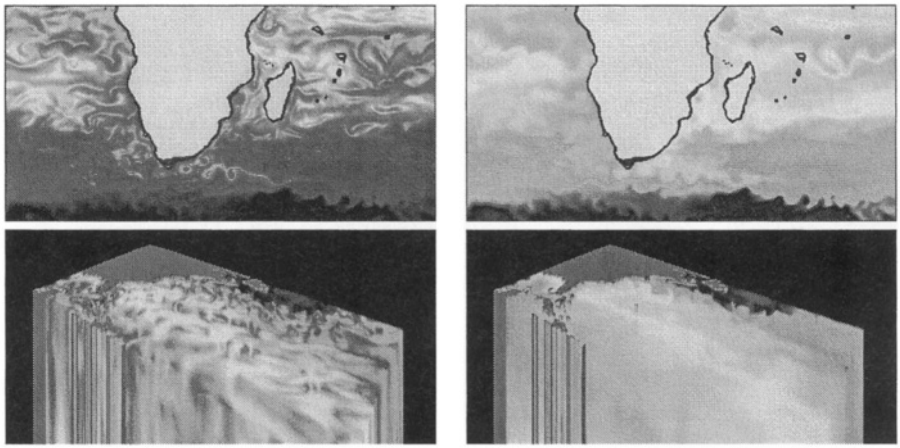
affect the actual frame rates in run-time rendering because the entire voxels of each test set are loaded into shared memory.

The rendering phase involves texture loading from shared memory space to texture memory, 3D texture mapping in Geometry Engines and Raster Managers, and image composition. It is important to optimize each step in the rendering phase to achieve interactive-time frame rates. Because the maximum texture download rate is 330 MB/second from host memory, it takes at least 0.19 seconds $(\frac{64}{330})$ to load one 64 MB brick. Despite the ability that downloads and draws textures simultaneously, allowing textures to be updated on the fly, the cost is so expensive that real-time frame rates are hard to achieve. When volume datasets, much larger than the amount of texture memory on the graphics platform, are visualized, the inevitable texture swapping hinders real-time rendering [5], hence should be minimized. Actually, without texture swapping, it was possible to create over 30 frames per second using the Onyx2. Multiple graphics pipes are involved in our scheme. As each pipe loads different bricks, the amount of texture swapping can be minimized.

As mentioned, partitioning large volume data into bricks in hardware accelerated rendering is necessary. It is important to assign them to the respective pipes carefully because the number of Raster Managers of our system varies pipe-by-pipe (see Figure 3 (a) again). It takes 0.08 seconds (12.24 frames per second) to render a volume dataset of 64 MB using a pipe in $\{P_0, P_2, P_4\}$ with four Raster Managers. On the contrary, if a pipe in $\{P_1, P_3, P_5\}$ with two Raster Managers is used to render the same data, the rendering time is almost doubled (0.15 seconds: 6.81 frames per second). We were able to improve the speedup by assigning additional bricks to pipes with more Raster Managers.

Figure 3 (b) gives an overview of our algorithm based upon the object-space division method to minimize texture swapping. The master pipe $\{P_0\}$ plays an important role in controlling the slave pipes $\{P_1, P_2, P_3, P_4, P_5\}$ and composing sub-images. The slave pipes render assigned bricks and write sub-images to shared memory space under the control of the master pipe. The polygonization process as a separate thread process continues to generate sequences of polygons perpendicular to the viewing direction until the program is finished.

Once the current view is set, a rendering order of bricks is determined. Each pipe starts creating sub-images for the assigned bricks and then the master pipe composes the sub-



(a) Hybrid method (b) Color mapping method

Fig. 4. Visualization results on T data

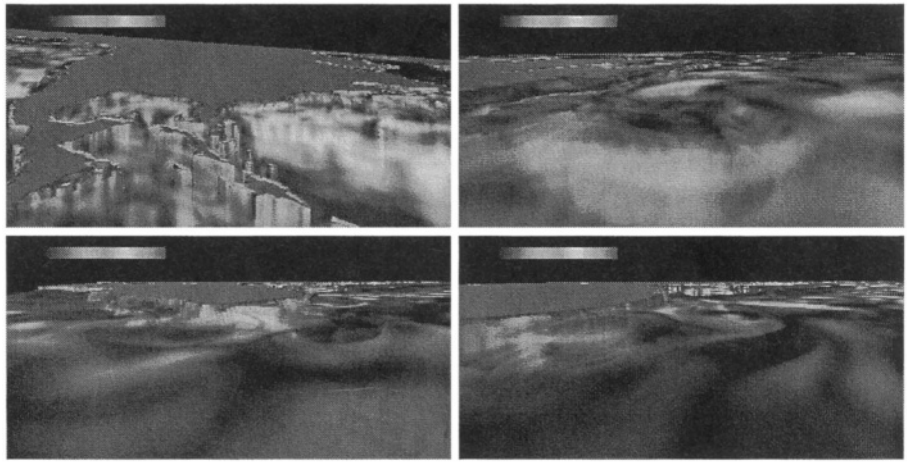


Fig. 5. Images of T data created by combining volume rendering and isocontouring

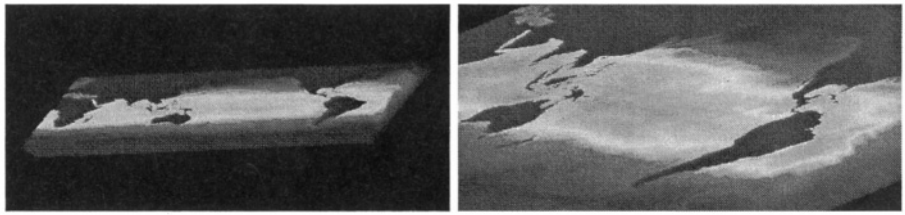


Fig. 6. Interactive multi-pipe time-varying volume visualization of T data

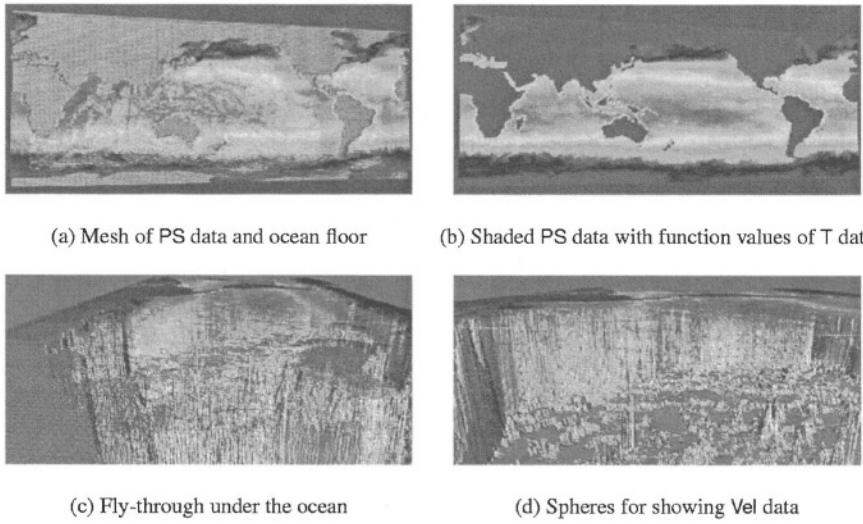


Fig. 7. A fly-through using the SGI Performer

images according to the order. Our algorithm requires synchronization for every frame between the master pipe and the slave pipes. Because the master pipe must wait until all slave pipes have written sub-images in shared memory space, the actual frame rate is dictated by the slowest pipe. We tried to solve this problem by proportionally assigning bricks. It is inevitable that the rendering of time-varying data requires additional texture swapping. To reduce the swapping cost, the algorithm routinely checks to make sure that the next brick to be rendered is already stored in texture memory.

3.2 Performance of Multi-pipe Rendering

We have used the OpenGL and OpenGL Volumizer [4] to measure the timing performance in rendering images of size 640 × 640. As hardware-supported texture memory requires the dimensions of the bricks to be a power of two, each original time step of volume data was reduced to 2048 × 1024 × 32 volume with 1-byte voxels for this experiment. The best rendering speed was achieved when three pipes $\{P_0, P_2, P_4\}$ were used. The oceanography data was rendered at an interactive frame rate of 12.3 frames per second with varying time steps.

4 Experimental Results and Concluding Remarks

When two videos, one created by our rendering scheme and the other by the color mapping method, were displayed at the same time in a single window, we found that it enables an oceanographer to easily analyze the information contained in the time-varying volume data. In Figure 4, interesting local regions are visualized to see the difference of the two rendering methods. The oblique images clearly show that our hybrid technique represents ocean changes better than the color mapping method does.

Figure 5 demonstrates animation frames taken from high density movies made by our off-line parallel scheme.

The images of Figure 6 are snapshots created by our multi-pipe technique at an interactive frame rate. As it creates images by rendering a sequence of resampling planes intersected with bricks, the volume data is also considered a geometric object. Therefore, it is easy to visualize volume data and other geometric objects simultaneously. It is also trivial to dynamically set cutting planes that allow us to investigate the inside of an opaque volume. Our multi-pipe rendering scheme can be easily transformed as a core rendering module for stereoscopic multi-screen display systems.

Time-varying oceanography data could be visualized using the OpenGL Performer [3]. Because the OpenGL Performer uses only polygonal models as input data and exploits optimized multi-pipe rendering, we had to extract geometric information from the time-varying volume dataset in a pre-processing stage. In Figure 7, the dynamic ocean surface polygons and the colors of the polygon vertices defined as functional values were generated from the PS data and the T data, respectively. The mesh dataset for the ocean floor was created by a simple polygonization algorithm. To visualize the velocity vector field, we made spheres from Vel. The radius of the spheres represents the magnitude of velocity at the sampling point, and the line segments attached to the spheres indicate the direction. During a fly-through at an interactive frame rate, we were able to effectively investigate the relationship between datasets in the dynamically changing ocean field.

In this paper, we have presented two visualization techniques to render huge time-varying oceanography data. We focused not only on creating high quality images using an off-line parallel rendering algorithm, but also on an interactive rendering algorithm that takes advantage of the multi-pipe feature of an SGI's high performance graphics system. Currently, our visualization techniques are planned to be ported to commodity PC cluster environments.

Acknowledgements. We wish to thank Prof. Detlef Stammer and Dr. Arne Biastoch of Scripps Institution of Oceanography for allowing access to the oceanography data. This work was supported by Korea Research Foundation Grant (KRF-2003-003-D00387).

References

1. C. Bajaj, I. Ihm, G. Koo, and S. Park. Parallel ray casting of visible human on distributed memory architectures. In *Proceedings of VisSym '99 (Joint EUROGRAPHICS-IEEE TCVG Symposium on Visualization),* pages 269–276, Vienna, Austria, May 1999.
2. G. Eckel. Onyx2 Reality, Onyx2 InfiniteReality and Onyx2 InfiniteReality2 technical report. Technical report, Silicon Graphics, Inc., 1998.
3. G. Eckel and K. Jones. OpenGL Performer programmer's guide. Technical report, Silicon Graphics, Inc., 2000.
4. Silicon Graphics Inc. OpenGL Volumizer programmer's guide. Technical report, Silicon Graphics, Inc., 1998.
5. W. R. Volz. Gigabyte volume viewing using split software/hardware interpolation. In *Proceedings of Volume Visualization and Graphics Symposium 2000,* pages 15–22, 2000.

Sphere-Spin-Image: A Viewpoint-Invariant Surface Representation for 3D Face Recognition*

Yueming Wang, Gang Pan, Zhaohui Wu, and Shi Han

Department of Computer Science and Engineering
Zhejiang University, Hangzhou, 310027, P. R. China
{ymingwang, gpan}@zju.edu.cn

Abstract. This paper presents a new free-form surface representation scheme, which we call *Sphere-Spin-Image* (SSI), and its application to 3D face recognition. An SSI, associated with a point on the surface, is a 2D histogram constructed from a neighborhood surface of the point using position information, which captures the characteristic of local shape. Thus, a free-form surface can be represented by a series of SSIs. Correlation coefficient is used as similarity metric for comparing SSIs. During face recognition, the SSIs of points on face surface are computed, and recognition task is achieved by SSI-comparison-based voting method. To reduce computational cost, only some particular points of face surface are involved in voting. With face database consisting of 31 different pose models, experimental result of equal error rate 8.32% demonstrates performance of the proposed method.

1 Introduction

Free-form surface representation schemes are widely used both in computer graphics and computer vision, and have been studied extensively recently. For the purpose of registration and recognition, the representation scheme should be (1)viewpoint-independent, (2)general enough to describe the sculpted objects, (3)as compact and expressive as possible. Several representation schemes have been presented during the last few years. Dorai et al. [12] used shape index($S=1/2 - \frac{1}{\pi}\arctan\frac{K_1+K_2}{K_1-K_2}$) along with a spectral extension of the shape measure to build a view-dependent representation of free-form surface named *COSMOS*. Johnson and Hebert [9,10] introduced *spin image* which comprises descriptive images associated with oriented points on the surface. Using a single point basis, the positions of the other points on the surface are described by two parameters. The accumulation of these parameters for many points on the surface of the object results in an image at each oriented point. *Point signature* proposed by Chua and Jarvis [2] serves to describe the structural neighborhood of a point by encoding the minimum distances of points on a 3D contour to

* This work is in part supported by NSF of China under grant No.60273059, National 863 High-Tech Programme under No.2001AA4180, Zhejiang NSF for Outstanding Young Scientist under No.RC01058 and Doctoral Fund under No.20020335025.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 427–434, 2004.
© Springer-Verlag Berlin Heidelberg 2004

a reference plane. Yamany and Farag [11] generated a *signature image* where the image axes are the distance between a point basis and other points on the surface and the angle between the normal at the point basis and the vector from the point to each other point. The signature image encodes the mean curvature.

Automatic face recognition has been studied extensively in computer vision area over past decade. However, most approaches of face recognition are based on intensity images of faces, only a few exploited three dimensional information [1]. Face recognition paradigms based on range data images have more potential than intensity images based approaches because the former is invariant under the changes of lighting conditions, color and reflectance properties of face while the latter is sensitive to above changes. Previous work of face recognition based on range data is briefly reviewed in the following. Gordon [6] employed depth and curvature features from face range data and performed face recognition. It is done by computing curvatures of face surface to find fiducial regions such as eye corner cavities to calculate the face-specific descriptors(eg. eye separation etc.). Tanaka et al. [5] presented a correlation-based face recognition approach based on the analysis of maximum and minimum principal curvatures and their directions. Chua and Han [3] treated face recognition problem as a non-rigid object recognition problem. The rigid parts of the face were extracted after registering the range data of faces having different facial expressions using point signature. The most appropriate models from the database constructed with the rigid parts of face surfaces are ranked according to their similarity with the test face.

Motivated by *spin image* [10] and *point signature* [3], we present a new approach of 3D local shape representation named *Sphere-Spin-Image*(SSI) . The SSI of a point is constructed by mapping 3D coordinates of those surface's points within a sphere centered at this point into 2D space. Since SSI represents the local shape of a point, it can be utilized as a intermediate representation in 3D face recognition.

This paper is organized as follows: The SSI representation is described in Section 2. Section 3 discusses the special points selection process and the voting method based on SSI comparison for face recognition. The experimental results and conclusion are given in section 4 and section 5 respectively.

2 Sphere-Spin-Image(SSI) Representation Scheme

2.1 Definition of Sphere-Spin-Image

Given a point $\boldsymbol{p}$ with unit normal $\boldsymbol{n}$ and tangent plane TP_p through $\boldsymbol{p}$ on the surface S, each other point $\boldsymbol{p_k}$ on the surface S can be related to $\boldsymbol{p}$ by two parameters: 1) the distance $\rho_k=\|\boldsymbol{p}-\boldsymbol{p_k}\|$; 2) the distance from $\boldsymbol{p_k}$ to the tangent plane TP_p, $d_k=\boldsymbol{n}\cdot(\boldsymbol{p_k}-\boldsymbol{p})$, as shown in Fig.1 (a). Apparently, d_k could be either positive or negative while ρ_k could not be negative.

The Sphere-Spin-Image(SSI) is defined as follows: for each point $\boldsymbol{p}$ on the surface S, shown in Fig. 1 (b), a sphere with radius r can be placed centrally at

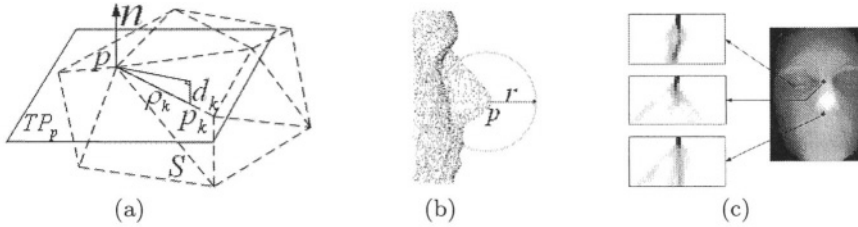


Fig. 1. (a) Relative position of point $\boldsymbol{p_k}$ to point $\boldsymbol{p}$. (b) A surface patch surrounded by a sphere centered at $\boldsymbol{p}$. (c) Examples of SSIs taken at different point location on face surface. Notice the difference between the SSI corresponding to a location and SSI of other location.

$\boldsymbol{p}$. The SSI of $\boldsymbol{p}$ is a 2D histogram of ρ_k and d_k formed by those points both on the surface S and in the sphere. It can be formalized as:

$$\boldsymbol{SSI}\colon S^r(\boldsymbol{p}) = (S^r_{ij}(\boldsymbol{p}))_{m\times n, 1\le i\le m, 1\le j\le n} \tag{1}$$

$$S^r_{ij}(\boldsymbol{p}) = \sum_k s^k_{ij} \tag{2}$$

where

$$s^k_{ij} = \begin{cases} 1, \text{ if } \boldsymbol{p_k} \in S \text{ and } \|\boldsymbol{p_k} - \boldsymbol{p}\| \le r \\ \quad \text{and } i * \Delta b_1 \le \rho_k < (i+1) * \Delta b_1 \\ \quad \text{and } j * \Delta b_2 \le d_k + r < (j+1) * \Delta b_2 \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

In Eq. 3, Δb_1 and Δb_2 are the sizes of bins in the 2D histogram. Although we can notice that there is a missing degree of freedom which is the cylindrical angular parameter around unit normal with SSI representation scheme of free-form surface, each SSI uniquely defines the location of corresponding point on the surface using small sizes of bins, as shown in Fig. 1 (c). Obviously, the SSI is insensitive to pose variance.

The SSI representation scheme is inspired by *spin-image* [9,10] and *point signature* [2]. There are two main differences between SSI and spin-image. Firstly, an SSI of a point $\boldsymbol{p}$ uses points within the sphere centered at $\boldsymbol{p}$ on the surface to construct 2D histogram, while a spin-image of a certain point on the surface uses all points on the surface to construct 2D histogram. Generally, most 3D scanning methods can only capture part of the objects at a time [8]. This could be due to limitation in the physical design and the technology the scanner is built upon. Using these scanning methods parts of an object's surface may not be recoverable because the object itself shadows the structured light from the detector in the sensor. As shown in Fig. 2, each face surface contains regions that are invisible in the other. Thus, spin image constructed by all surface's points, including points in those invisible regions, will bring ambiguity during matching. Contrastively, SSIs of two corresponding points of different views will cover the same regions of the surface by using the sphere of some appropriate radius. Secondly, SSI employs

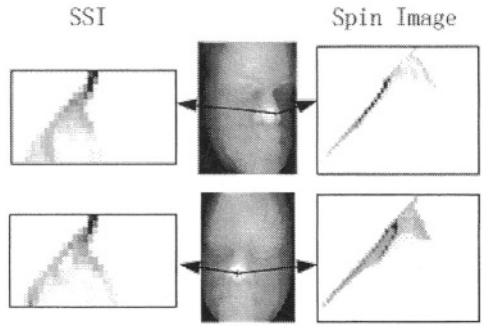


Fig. 2. Two depth images in different pose for the same person and their SSIs and spin images at the same location

the distance from other points to $\boldsymbol{p}$ as a dimension, instead of the distance from other points to the normal of point $\boldsymbol{p}$ as a dimension in spin-image. Hence, it is consistent with the use of sphere to define point's neighborhood.

2.2 SSI Generation

Prior to SSI comparison and face recognition with SSI, we present a scheme to derive the SSI representation from object models represented by triangular meshes.

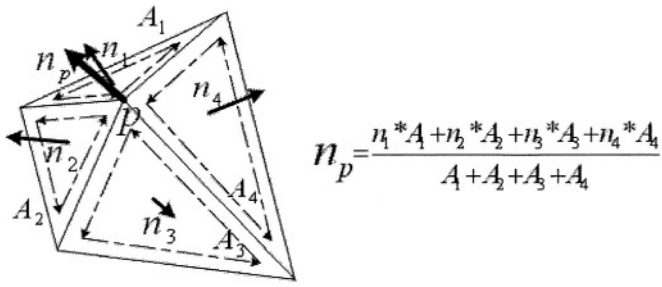


Fig. 3. Normal of vertex p calculation using the weighted means of triangles' normal adjacent to p. n_p is the normal of vertex p, n_i $(i=1,2,3,4)$ is triangle's normal, A_i $(i=1,2,3,4)$ is triangle's area.

Given a free-form surface consisting of triangular meshes, the normal of a vertex on the surface can be determined by the triangular patches adjacent to it. For SSIs of points on different models, the normals of vertices on different

models should be consistent, that is, all normals of vertices on different models must be oriented to the outside of the object. First of all, we sort edges of all triangular meshes of surface to be counter-clockwise viewed from one side of object. The consistent triangle's normal can be calculated based on the sorted edges, as shown in Fig. 3. Then normals of vertices whose orientation consistently extend to one side of the object is obtained by the weighted mean of triangle's normals adjacent to it. Finally, using the method proposed by Johnson [9], the orientation of all normals on the surface vertices is determined by calculating the scalar products of surface normal at each vertex and the vector from the centroid of the object to the vertex.

Before the construction of SSI, the radius r of sphere and Δb_1 and Δb_2 must be determined. In our experiments, the r is set as fifty times the mesh resolution and Δb_1 and Δb_2 are predetermined to be two to four times the mesh resolution which meets enough precision requirements. Then, the size of SSI, m and n defined in Eq.1 , can be easily computed by the equations:

$$m = \lfloor \frac{r}{\Delta b_1} \rfloor + 1, n = \lfloor 2 * \frac{r}{\Delta b_2} \rfloor + 1 \tag{4}$$

where $\lfloor\ \rfloor$ is the floor operator.

Additionally, to deal with the different resolution of range images, we resample the data of all models with same method to construct triangular meshes and normalizing the SSI after the SSI construction. For normalization, we simply let the 2D histogram divided by the total number of those points in the sphere.

2.3 SSI Comparison

Since correlation coefficient between images is constantly used to compare the similarity of images, it is also used to SSIs matching in our experiments. The correlation coefficient between a given SSI S_r^1 and a candidate SSI S_r^2 is:

$$CorCoe(S_r^1, S_r^2) = \frac{Cov(S_r^1, S_r^2)}{\sqrt{D(S_r^1) * D(S_r^2)}} \tag{5}$$

where

$$Cov(S_r^1, S_r^2) = \frac{1}{N} \sum_i S_r^1(i) S_r^2(i) - \frac{1}{N^2} \sum_i S_r^1(i) \sum_i S_r^2(i) \tag{6}$$

and

$$D(S_r) = \frac{1}{N} \sum_i S_r(i)^2 - (\frac{1}{N} \sum_i S_r(i))^2 \tag{7}$$

3 Face Recognition

Before face recognition with SSI, the model database must be built up. For each model, computing SSIs of every point on the surface has a very high time complexity. Selecting some special points which serve as landmarks of the object to compute SSIs can considerably reduce the cost while not block the performance of face recognition markedly. We select these points through two steps:

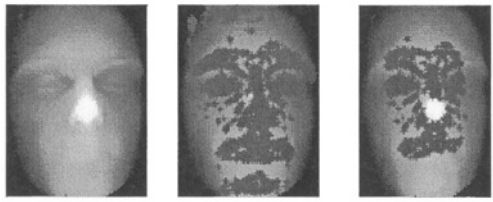


Fig. 4. (a) range image of a model, (b) Special points selection after step(1). (c) Special points selection after step(2). In (b) and (c), red parts denote the selected points.

- Firstly, ridge lines of a face surface are extracted by thresholding the minimum principal curvature of surface points, $k_{min} \leq threshold_c$. It is shown in Fig. 4(b).
- Secondly, from Fig. 4(b), it can be seen that ridge lines of a face surface contains many marginal points which are little help for face recognition but increase the computing complexity. Thus, we choose points both on ridge lines and in middle area of face surface as special point set P_{sp} of which SSIs are computed to construct model database. Marginal points are eliminated from ridge lines by counting the numbers of points within the point's related sphere and discarding the point whose related sphere contains points less than $threshold_s$. Fig. 4 (c) shows the set P_{sp}.

By the construction of SSIs of points in set P_{sp} for each model, we complete a model database in which each model has a SSI set S_{sp}. Face recognition is then implemented by voting method similar to[3]. For a given scene face, we want to select the most similar model in the database. Set P_{sp} and S_{sp} of scene face can be generated using the same construction method as models. For each SSI in S_{sp} of a scene face, correlation coefficients with every SSI in a model's S_{sp} in database are calculated. If the maximum of those correlation coefficients is larger than a $threshold_{CorCoe}$, the model receives a vote. Suppose the SSI number in set S_{sp} of the scene face is N_c and the model receives N_m votes, a voting rate associated to the model is N_m/N_c. The voting rate represents the likelihood of each model being correctly matched with the scene face. Using the voting method, the candidate models are ordered according to similarity.

4 Experimental Results

In order to test the effectiveness of the presented approach, we perform several experiments with models obtained by Delaunay triangulation of range images downloaded from SAMPL(Signal Analysis and Machine Perception Laboratory) [7] at Ohio State University. The difference between the models from the range images of SAMPL and general model represented by polygon mesh is that the former have no points information in the regions occluded by itself when the

Fig. 5. Sample images from SAMPL. Six range images in model database. Notice the variance of the model pose.

model keeps certain special pose. Thus, the range images contain less 3D information. Our experiments involve 31 models of six human subjects and two human subjects have several views with significantly different poses. Fig. 5 shows six samples of the range images.

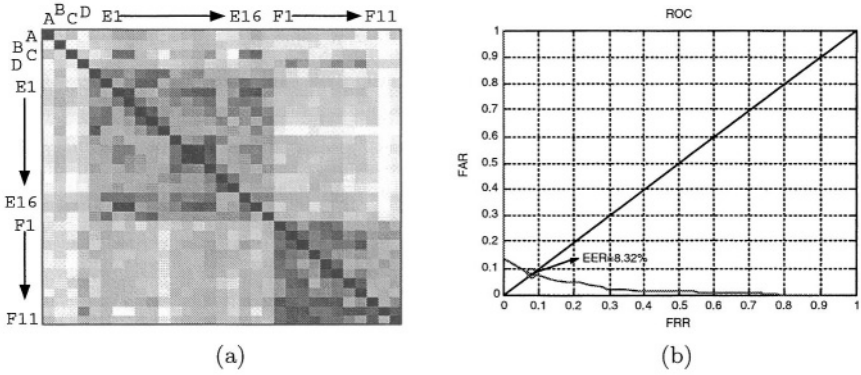


Fig. 6. (a) Similarity matrix for 31 models. Lightness indicates the dissimilarity between models. (b) ROC(Receiver Operator Characteristic) curve of recognition results. EER=8.32%.

In our experiments, we investigated the ability of SSI representation scheme and the voting method to recognition among models in the constructed model database. We calculated voting rates for all pairs of 31 models and the resulting voting rates are shown as a matrix in Fig. 6 (a). The lightness of each element(i,j) is proportional to the magnitude of the dissimilarity between models i and j. The darker represent better matches, while the lighter indicate worse matches. The symbols at the left and top side in Fig. 6 (a) denote different models where the same letter with different number means the models are of same human subject with different pose.

False reject rate(FRR) and false acceptance rate are quoted in many cases to evaluate recognition performance. A false acceptance rate(FAR) is the percentage of imposters wrongly matched, and a false rejection rate(FRR) is the

percentage of valid users wrongly rejected. Given a threshold, a FRR and a FAR can be calculated from the matrix. With different thresholds, a plot of numerous FAR and FRR combination, named receiver operator characteristic(ROC) curve, is obtained, shown in Fig. 6 (b). Equal Error Rate(EER) is the error rate when FRR equals FAR which can evaluate the performance of our recognition algorithm. With the significantly different pose models, EER in our experiment still maintains 8.32% which is a promising result.

5 Conclusion

This paper introduced a new free-form surface representation named SSI. An SSI of a given surface's point encodes neighborhood shape of the point through constructing a 2D histogram. It is done by mapping 3D coordinates of surface's points within a sphere centered at the given point into 2D space. The key feature of this approach is that it is viewpoint-invariant. Based on SSIs comparison of special points on models, face recognition experiments were performed by voting method. Although our model database contains relatively less face subjects due to the difficulty of obtaining the facial range images, significant pose variance in our 3D face database make our experimental results convincing.

References

1. W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld. Face recognition: a literature survey. ACM Computing Survey, 2003, 35(4): 399-458
2. C. S. Chua, R. Jarvis. Point signatures: A new representation for 3D object recognition. *IJCV,* 1997, 25(1): 63-85
3. C. S. Chua, F. Han, Y. K. Ho. 3D Human Face Recognition Using Point Signature. *Proc. of Int'l Conf. on Automatic Face and Gesture Recognition,* 2000: 233-238
4. R. Osada, T. Funkhouser, B. chazelle et al. Shape distributions. *A CM Transactions on Graphics,* 2002, 21(4): 807-832.
5. H. T. Tanaka, M. Ikeda, H. Chiaki. Curvature-based Face Surface Recognition Using Spherical Correlation-Principal Directions for Curved Object Recognition-. *Proc. of Int'l Conf. on Automatic Face and Gesture Recognition,* 1998: 372-377
6. G. G. Gordon. Face recognition from depth maps and surface curvature . *Proc. of SPIE Conf. on Geometric Methods in Computer Vision,* 1991, 1570: 234-247.
7. Range Imagery, SAMPL (Signal Analysis and Machine perception Laboratory at the Ohio State University). [Online]. Available: http://sample.eng.ohio-state.edu
8. R. J. Campbell, P.J.Flynn. A Survey Of Free-Form Object Representation and Recognition Techniques. *CVIU,* 2001, 81(2): 166-210
9. A. E. Johnson, M.Hebert. Surface matching for object recognition in complex three-dimensional scenes. *Image Vision Computing,* 1998, 16: 635-651.
10. A. E. Johnson, M.Hebert. Using spin images for efficient object recognition in cluttered 3D scenes.*IEEE Trans. PAMI.*1999, 21(5): 433-449
11. S. M. Yamany, A.A.Farag. Surface signatures: an orientation independent free-form surface representation scheme for the purpose of objects registration and matching. *IEEE Trans. PAMI* 2002, 24(8): 1105-1120
12. C. Dorai, A.K.Jain. COSMOS-A representation scheme for 3D free-form objects. *IEEE Trans. PAMI,* 1997, 19: 1115-1130

Design and Implementation of Integrated Assembly Object Model for Intelligent Virtual Assembly Planning*

Jing Fan, Yang Ye, and Jia-Mei Cai

College of Information Engineering, Zhejiang University of Technology
Zhaohui Liuqu, Hangzhou, 310014, P.R. China
fanjing@zjut.edu.cn

Abstract. Based on the analysis of the virtual assembly planning, this paper presents a new assembly planning method, which is named Intelligent Virtual Assembly Planning. The architecture of the IVAP system consists of four main components: Intelligent Assembly Planning Environment, Virtual Assembly Planning Environment, Data/Knowledge Acquisition Interface, and Inner Interactive Interface. The integrated assembly object model used in the system is described in detail in the paper.

1 Introduction

The rapid development of Virtual Environment provides a new tool to assembly planning of products. After getting data of the virtual components, and applying Virtual Environment as interactive interface to help to complete the task of assembly planning, virtual prototype of the product can be obtained. With the support of Virtual Environment and computer system, the engineer can manipulate the virtual components effectively and easily to create, simulate and evaluate assembly sequence. The way working in Virtual Environment is similar to the way of working in the real environment, and is easily accepted by the user.

The research on virtual assembly has developed quickly in the world. But it still has many limitations. For example, the user in Virtual Environment can not work for a long time, because his eyes wearing HMD easily feel tired; the delay between assembly operation and display result affects the precision of assembly; and the technique of collision detection on line still need to be improved. On the analysis of the technology of Computer Aided Assembly Planning, a new method of virtual assembly planning is proposed, which applies the formalized assembly knowledge of expert that is obtained using the technology of artificial intelligence in the virtual assembly, and creates Integrated Assembly Model to represent different kinds of assembly data and knowledge, together with reasoning strategy, to implement the intelligent virtual assembly planning.

* The research work in this paper is supported by the project of Zhejiang Natural Science Foundation of China (No. 602092).

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 435–442, 2004.
© Springer-Verlag Berlin Heidelberg 2004

This paper introduces the main idea of intelligent virtual assembly planning, and focuses on the Integrated Assembly Object Model used in intelligent virtual assembly planning.

2 Related Research

The first research of assembly model was done by Lieberman and Welsey in 1977. They have used graph structure to describe the information of assembly, in which node represents the part, and arc represents the relation between parts [1]. After that there are a lot of research work have been done in this area, such as the liaison graph proposed by Bourjault [2], the AND/OR graph used by Homem De Mello and Sanderson [3], the assembly constraints graph described by J. Wolter [4], and the virtual link designed by Lee and Gossard [5], etc. These assembly models are used to generate the assembly sequence automatically. But the common limitation of these models is that they can not express the assembly knowledge efficiently.

The assembly models used in virtual environment are usually based on the geometry model. They can express the geometry features of the parts effectively, but ignore other important non-geometry information like the physical properties and assembly semantics [6],[7]. Using this kind of assembly models, the assembly knowledge and experiences from the planning experts is difficult to be represented and utilized in a formalized way.

3 Intelligent Virtual Assembly Planning

The main idea of the intelligent virtual assembly planning is as follows: using the new interactive tool provided by virtual environment in the assembly planning, and applying the techniques of AI to support the virtual assembly planning, to improve the suitability and efficiency of the assembly planning to the complex assembly objects.

The architecture of intelligent virtual assembly planning system is shown in Fig. 1. There are four main parts in the system:

- Intelligent Assembly Planning Environment. It contains the Integrated Data/Knowledge Model, Intelligent Planner and Evaluator. Integrated Data/Knowledge Model consists of integrated assembly object model, scene model and interaction model. The integrated assembly object model is the emphasis of this paper, which is used to express the assembly objects.
- Virtual Assembly Planning Environment. It contains Simulation Module, State Control Module, and Input/Output Module, etc.
- Data and Knowledge Acquisition Interface. It provides the interface to input data from other systems and get assembly knowledge from the assembly planning experts. The assembly data and knowledge are transformed to construct the Integrated Assembly Model, and define assembly rules.

– Inner Interactive Interface. It is used to implement the communication and coordination between Intelligent Assembly Planning Environment and Virtual Assembly Planning Environment. The assembly data and knowledge are exchanged and shared through the Inner Interactive Interface.

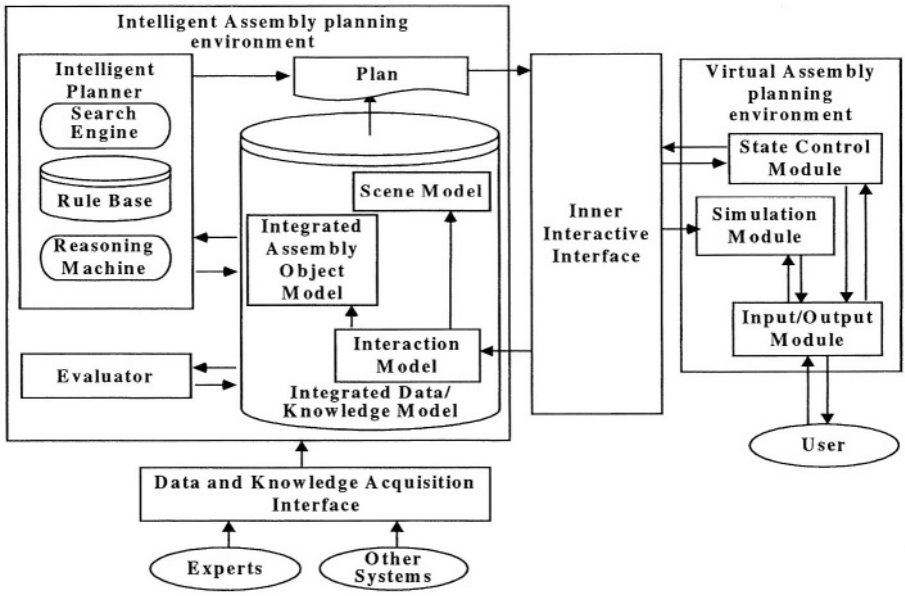


Fig. 1. Architecture of Intelligent Virtual Assembly Planning System

4 Integrated Assembly Object Model

Since the assembly models that have been proposed in the past time can not efficiently express the complex assembly objects, especially in the virtual assembly environment for the virtual assembly planning, we present the integrated assembly object model to express the assembly data and knowledge of the virtual components.

The integrated assembly object model contains data model, constraint model, and semantic model.

– Data model expresses the basic information of virtual assembly objects such as geometry features and physical properties in a unique form.
– Constraint model describes the constraints of virtual assembly objects, which mean the relations between the objects that cannot be expressed in the data model. All the assembly constraints are expressed by predicate.
– Semantic model describes the semantic of a group of assembly objects, including the information of assembly process and the operations of the assembly objects.

4.1 Data Model

Virtual assembly objects is clustered and described in a unique form, which is called Virtual Assembly Concept (VAC). Formalized expression of VAC is as the follows:

```
<VAC>::=
(<Name><Superclass><Subclass><Components><Attributes>
<Geometry-Features><Physical-Features><Join-Methods>
<Mating-Parts><Tools>)
  <Name>::=<string>
  <Superclass>::=<VAC>|NULL
  <Subclass>::=<VAC>|NULL
  <Components>::=<VAC>|NULL
  <Attributes>::=(<Material><Color><Function><Motion>)
  <Geometry-Features>::=
    (<Shape><Size><Position><Orientation>)
  <Physical-Features>::=
    (<Weight><Gravity-Center><Stiffness><Sticky>
     <Asperity><Vulnerability>)
  <Join-Methods>::=
    (<Snap><Screwing><Peg-hole><Pinning><Riveting>)
  <Mating-Parts>::=<VAC>
  <Tools>::=<string>
```

4.2 Constraint Model

The constraints in the constraint model are classified as two kinds, Low-level constraints and High-level constraints. Low-level constraints are established on the basis of data model while High-level constraints are based on the Low-level constraints.

Low-level constraints are composed of Low-level geometry constraints and Low-level physical constraints. Low-level geometry constraints represent geometry of the assembly objects, such as position, orientation, mate, co-axis, insertion, etc. Low-level physical constraints represent the force effects between the objects.

High-level constraints represent the abstract information of the relations between the objects. It includes: High-level geometry constraints, High-level physical constraints, topology constraints, and precedence constraints, etc. User defined constraints, like cost constraint and time constraint, which are named soft constraints, are also included in the High-level constraints.

4.3 Integrated Semantic Model

The assembly semantic of the virtual assembly objects is expressed by the integrated semantic model (ISM), which is established in a higher level to describe the dynamic information between objects, such as assembly operations, sequence and process.

Assembly process information describes the assembly process of the group of virtual object according to the semantic. It includes transformation from initial state to

final state. The changes of the states are made by the operation steps. Each step is corresponding to a list of assembly operations.

Assembly sequence information describes the precedence of the assembly operations to the objects.

Assembly operation describes the assembly operations on the objects, which consists of basic meta operations, as translation and rotation.

Formalized expression of the ISM is a quintuple.

$$ISM = <E, O, P_{info}, S_{info}, O_{info}>$$

Where E is the set of the assembly objects, O is the set of the assembly operations, P_{info} stores the process information, S_{info} presents the assembly sequence information and O_{info} represents the information of assembly operation.

$$E= \{ P_i | i = 1,\cdots,n\}$$

There are n assembly objects P_i.

$$O= \{ O_j (P_i) | i= 1,\cdots n; j = 1,2\}$$

There are n assembly objects, each have two meta-operations, translation O_1 and rotation O_2. There are $2n$ operations totally.

$$P_{info}= \{S_{p-1}, STEP_p | p=1,\cdots,t\}$$

There are $t+1$ states and t steps. S_o is the initial state and S_t is the final state. S_{p-1} is transformed to state S_p with Step $STEP_p$. Each step is composed of a list of assembly operations.

$$S_{info}=\{SE, SA\}$$

SE is the sequence between the assembly objects.
SA is the sequence between the operations.

$$O_{info}=\{A_k | k=1,\cdots,m\}$$
$$A_k \in \{O_j (P_i) | i=1,\cdots,n; j=1,\cdots,2\}$$

There are m operations used in the assembly process.

5 The Implementation of the Model

5.1 Data Model

The virtual components are described in VRML2.0.

The description of the virtual components is constructed with two parts: nodes and features. Each node represents one kind of virtual components.

```
PROTO Name [
Node (){
#<Join_Methods>, <Tool>, Weight, …
```

```
Group {
Children_1 [
Translation Gravity-Center
Shape {
          Material {color {}}
          Geometry {}…}
…]
Children_2 []
…
Children_n []
 }
} ]
Name {};
```

The features of the virtual component used in the assembly are extracted from node. Different nodes have different features. The basic geometry feature is represented in the following structure:

```
Feature parameter(shape,keysize,position,Orientation);
{
ExposedField MFNode shape
ExposedField MFFloat keysize
ExposedField MFVec3f position x,y,z
ExposedField MFRotation Rotation
}
```

For example, the basic geometry features of the part bolt are showed as follows:

```
Bolt (Cylinder, <radius, height>, <x, y, z>, rotation)
```

5.2 Constraint Model

Feature parameter can be transformed to lower geometry constraints automatically. After all the components are created in the VRML, the data of the components are transformed to constraints and stored in the Constraint Model.

The following are some Low-level constraints used popularly.

```
Position (Ni, x, y, z)
Orientation(Ni, α , β , ϕ )
Mate (N1.face, N2.face)
Co-axis (N1, N2.hole)
Insert (N1, N2.hole)
Align (N1.face, N2.face)
Parallel (N1, N2)
Vertical (N1, N2)
Tangent (N1.face, N2.face)
Loose (N1, N2)
Close (N1, N2)
```

High-level constraints are established on the Low-level constraints. Here are some High-level constraints.

```
Interfering (N1, N2)
Connect (N1, N2)
Proceed (N1, N2)
```

5.3 Semantic Base

The components that are contributed to one specialized assembly function are collected together to construct the assembly semantic.

The Semantic of "Part1 and part2 are jointed by bolt" is:

```
E= {part1, part2, Bolt}
O= {T(part1),T(part2),T(Bolt),R(part1),R(part2),R(Bolt)}
Initial state S0 :{part1,part2,Bolt} ⊕ STEP1=>
State S1 :{[part1,part2],Bolt} ⊕ STEP2=>
Final state S2: {[part1, part2, Bolt]}
STEP1: {A1, A2}
STEP2: {A3, A4};
Sequence: part1->part2,part2->part1;[part1,part2]->Bolt
A1->A2->A3->A4
A1: Place (part1, part2)
A2: Align (part1, part2)
A3: Align (Bolt, [part1, part2])
A4: Insert (Bolt, [part1, part2])
```

5.4 Example

In the example there are eight kinds of components, Axle, Base, Bolt, Bushing, L_Bracket, Pulley, Washer and Key (see Figure2), which are expressed by eight nodes in VRML And their geometry features are gotten by function call through the VRML In/Out event interface.

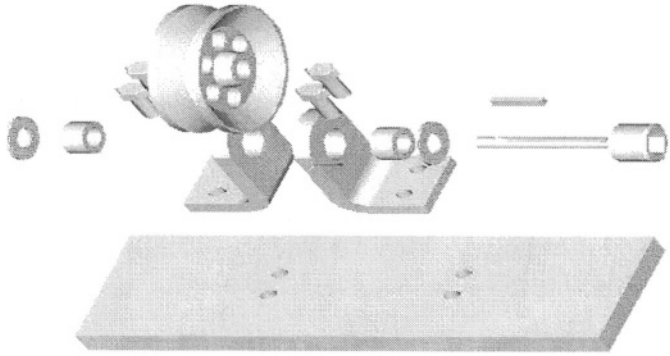


Fig. 2. Example

There are 15 constraints and three semantics totally obtained from the data model. And here are the three assembly semantics.
1."Part Base and Part L_Bracket are jointed by Part Bolt"
2."Part L_Bracket, Part Pulley and Part Axle are jointed"
3."Part Key is placed in the slot on Part Axle"

6 Summary

Although the mechanical parts can be created in the 3D model easily, the relations between parts cannot be described effectively. The integrated assembly object model can be used to express the data and knowledge of the virtual assembly objects in different level for assembly planning. The establishment of the powerful assembly model will help to simplify and improve virtual assembly planning.

References

1. Lieberman L. I., Wesley M. A.: AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly, IBM Journal of Research and Development, 21(4), (1977) 321-333
2. Bourjault A.: Contributionune approche methodologique de l'assemblage automatise: elaboration automatique des sequences operatiores, Thesis d'Etat Universite de Franche-Comte, Besancon, France (1984)
3. Homem de Mello L., Sanderson A.: And/Or Graph Representation of Assembly Plans, IEEE Transactions on Robotics and Automation, 6(2), (1990) 188-199
4. Wolter J.: On The Automatic Generation Of Assembly Plans, Computer-Aided Mechanical Assembly Planning, Kluwer Academic Publishers (1991)
5. Lee K., Gossard D. C: A Hierarchical Data Structures for Representing Assemblies, part I., Computer Aided Design, 17(1), (1985) 15-19
6. Bullinger H.J., Richter M. and Seidel K.A.: Virtual Assembly Planning, Human Factors and Ergonomics in Manufacturing, 10(3), (2000) 331-341
7. Dewar R., Carpenter I. D., Ritchie J. M., and Simmons J. E. L: Assembly Planning in a Virtual Environment, Proceedings of PICMET '97 Portland USA (1997) 664-667

Adaptive Model Based Parameter Estimation, Based on Sparse Data and Frequency Derivatives

Dirk Deschrijver, Tom Dhaene, and Jan Broeckhove

Group Computational Modeling and Programming (CoMP)
University of Antwerp, Middelheimlaan 1, 2020 Antwerp, Belgium
{dirk.deschrijver, tom.dhaene, jan.broeckhove}@ua.ac.be

Abstract. Rational functions are often used to model and to interpolate frequency-domain data. The data samples, required to obtain an accurate model, can be computationally expensive to simulate. Using the frequency derivatives of the data significantly reduces the number of support samples, since they provide additional information during the modeling process. They are particularly useful when the data is sparse and the samples are selected adaptively.

1 Introduction

The simulation of complex structures can be computationally very expensive and resource-demanding. One often wants to minimize the number of costly data samples, in order to obtain an accurate model in an acceptable amount of time [1]. The computational cost of higher-order derivatives is sometimes only a fraction of the cost of simulating additional samples. Some Finite Element Method (FEM) simulators, such as the High Frequency Structure Simulator (HFSS) [2] can take advantage of this property.

The Model Based Parameter Estimation (MBPE) [3] [4] is an efficient technique to model Linear Time Invariant systems (LTI) in the frequency domain using a rational pole-zero function. By expanding the numerator and denominator in a Forsythe polynomial basis [5], some numerical instabilities are improved, since this makes the normal equations best conditioned [6]. Furthermore, by using multiple spline interpolation with adaptive knot placement [7], the numerical stability issues of highly complex multi-pole systems can be circumvented.

In this paper, we extend MBPE so that the frequency derivatives of the data can be taken into account.

2 Model Based Parameter Estimation

2.1 Rational Model

Using the Model Based Parameter Estimation (MBPE) technique, the frequency domain data can be modeled by a rational transfer function :

$$H(f_s) = \frac{\sum_{n=0}^{N} N_n.(j2\pi f_s)^n}{\sum_{d=0}^{D} D_d.(j2\pi f_s)^d} \tag{1}$$

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 443–450, 2004.
© Springer-Verlag Berlin Heidelberg 2004

or equivalently :

$$\left(\sum_{d=0}^{D} D_d.(j2\pi f_s)^d\right).H(f_s) = \sum_{n=0}^{N} N_n.(j2\pi f_s)^n \tag{2}$$

where the $H(f_s)$ represent the data samples simulated at the discrete frequency f_s, $\forall s = 0, ..., S$. N and D are the order of numerator and denominator respectively. Coefficient D_0 can be set to 1 without loss of generality, since it is always possible to convert to this form by dividing numerator and denominator by D_0.

The calculation of numerator and denominator coefficients in this polynomial basis (the power series) leads to an ill-conditioned Vandermonde system. For highly dynamic systems, severe numerical problems will make the results inaccurate and completely useless. Therefore, the numerator and denominator of the rational function will be decomposed in a separate basis of Forsythe polynomials. This approach guarantees that the set of normal equations are best conditioned [8].

2.2 Forsythe Polynomials

The Forsythe polynomials are derived from the following three-term recurrence relation :

$$\begin{aligned} p_{-1}(f) &= 0 \\ P_0(f) &= 1 \\ P_1(f) &= j2\pi f.p_0(f) \\ &\cdots \\ P_i(f) &= j2\pi f.p_{i-1}(f) + \beta_{i-1}.p_{i-2}(f) \end{aligned} \tag{3}$$

The coefficients β are calculated by summation over all negative and positive sample frequencies

$$\beta_i = \sqrt{\sum_{s=0}^{S}[P_i(f_s)][P_i(f_s)]^*} \tag{4}$$

which leads to the following orthonormal Forsythe polynomials

$$p_i(f) = \frac{P_i(f)}{\beta_i} \tag{5}$$

The polynomials are orthonormal over the discrete sample set f_s, such that

$$\sum_{s=0}^{S}[p_m(f_s)].[p_n(f_s)]^* = \delta_{mn} \quad \forall 0 \leq m, n < S \tag{6}$$

After performing the basis transformation, the coefficients of numerator and denominator can be calculated by solving the following set of equations for all frequencies f_s with a Singular Value Decomposition (SVD) :

$$A_s x = b_s \tag{7}$$

where

$$A_s = \begin{bmatrix} \Re[-p_0(f_s)] \ldots \Re[-p_N(f_s)] \; \Re[p_1(f_s)H(f_s)] \ldots \Re[p_D(f_s)H(f_s)] \\ \Im[-p_0(f_s)] \ldots \Im[-p_N(f_s)] \; \Im[p_1(f_s)H(f_s)] \ldots \Im[p_D(f_s)H(f_s)] \end{bmatrix} \tag{8}$$

$$x = \begin{bmatrix} N_0 \ldots N_N \; D_1 \ldots D_d \end{bmatrix}^T \tag{9}$$

$$b_s = \begin{bmatrix} \Re[-p_0(f_s)H(f_s)] \\ \Im[-p_0(f_s)H(f_s)] \end{bmatrix} \tag{10}$$

Note that the system can be overdetermined, such that the number of rows in matrix A exceeds the number of columns (or the number of unknown coefficients).

3 Derivatives

Now assume that frequency derivatives of the data are available at the discrete frequencies f_s.

If $p_i^{(t)}(f_s)$ represents the t^{th} derivative of the i^{th} Forsythe polynomial, evaluated in frequency f_s, it can be expressed as :

$$p_i^{(t)}(f_s) = \frac{t.p_{i-1}^{(t-1)}(f_s) + j2\pi f.p_{i-1}^{t}(f_s) + \beta_{i-1}.p_{i-2}^{(t)}(f_s)}{\beta_i} \tag{11}$$

To avoid a breakdown of the orthogonality, virtual samples are introduced when frequency derivatives are used.

Equation (2) can be generalized by taking the frequency derivatives into account. The coefficients N_n and D_n of the rational fitting model now satisfy :

$$H^{(t)}(f_s).\sum_{d=0}^{D} D_d.p_d^{(0)}(f_s) = \sum_{n=t}^{N} N_n.p_n^{(t)}(f_s) - \sum_{k=1}^{t}\sum_{d=k}^{D} D_d.p_d^{(k)}(f_s).H^{(t-k)}(f_s).\frac{t!}{k!(t-k)!} \tag{12}$$

where $p_n^{(t)}$ is the t^{th} order derivative of the n^{th} order numerator Forsythe polynomial, $p_d^{(t)}$ is the t^{th} order derivative of the d^{th} order denominator Forsythe polynomial, and $H^{(t)}$ is the t^{th} order derivative of the frequency domain data. All derivatives are relative to $j2\pi f$. The set of equations at all frequencies f and for all derivatives t, can be written in a matrix form, similar to equation (7).

In some cases, the magnitude of the derivatives can be extremely small or large. Calculating the coefficients using all the information can make the corresponding least-squares system ill-conditioned. Therefore it is recommended to check the magnitude of the derivatives before they are used. A trade-off needs to be made between the loss of information and the loss of numerical stability.

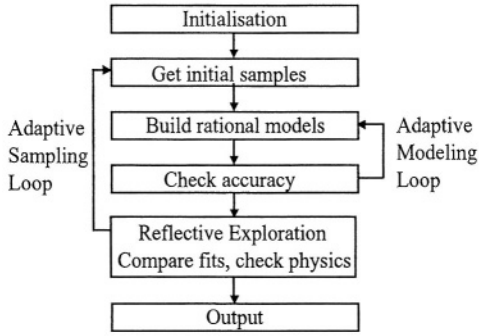


Fig. 1. Flowchart of the FFS algorithm

4 Fast Frequency Sweep

To minimize the cost of simulating computationally expensive data samples, an adaptive algorithm is used that automatically selects a minimal sample distribution and model complexity [1].

The flow chart of the algorithm is shown in Figure 1. It consists of an adaptive modeling loop, and an adaptive sample selection loop.

The algorithm starts with 4 samples equidistantly spaced over a certain frequency range of interest. Depending on the number of available data samples, multiple rational models are built with different order of numerator and denominator, exploiting all degrees of freedom.

The rational fitting models are evaluated in the data points, and compared against one another. If the error between the rational model (and its derivatives) evaluated in the selected sample points and the simulated data samples (and its derivatives), exceeds a certain threshold, the model is rejected, and the model's complexity is increased. All models with different order of numerator and denominator are ranked, and the 2 best models (i.e. with lowest overall error) are retained.

The difference between the two models is called the estimated fitting error, and new samples should be chosen in such way, that the maximum estimated fitting error is minimized. A reliable way to estimate the fitting error, and select new samples is given in [1]. Optionally, the algorithm can be extended with an adaptive spline selection loop, as introduced in [7].

5 Example: Ring Resonator

The new proposed technique is used to model the reflection coefficients S_{11} of a microwave RingResonator over the frequency range [1 GHz - 1.2 GHz]. The desired accuracy level is -60 dB, which corresponds to 3 significant digits. A highly accurate full-wave electro-magnetic simulator Momentum [9] is used to simulate the sparse data samples.

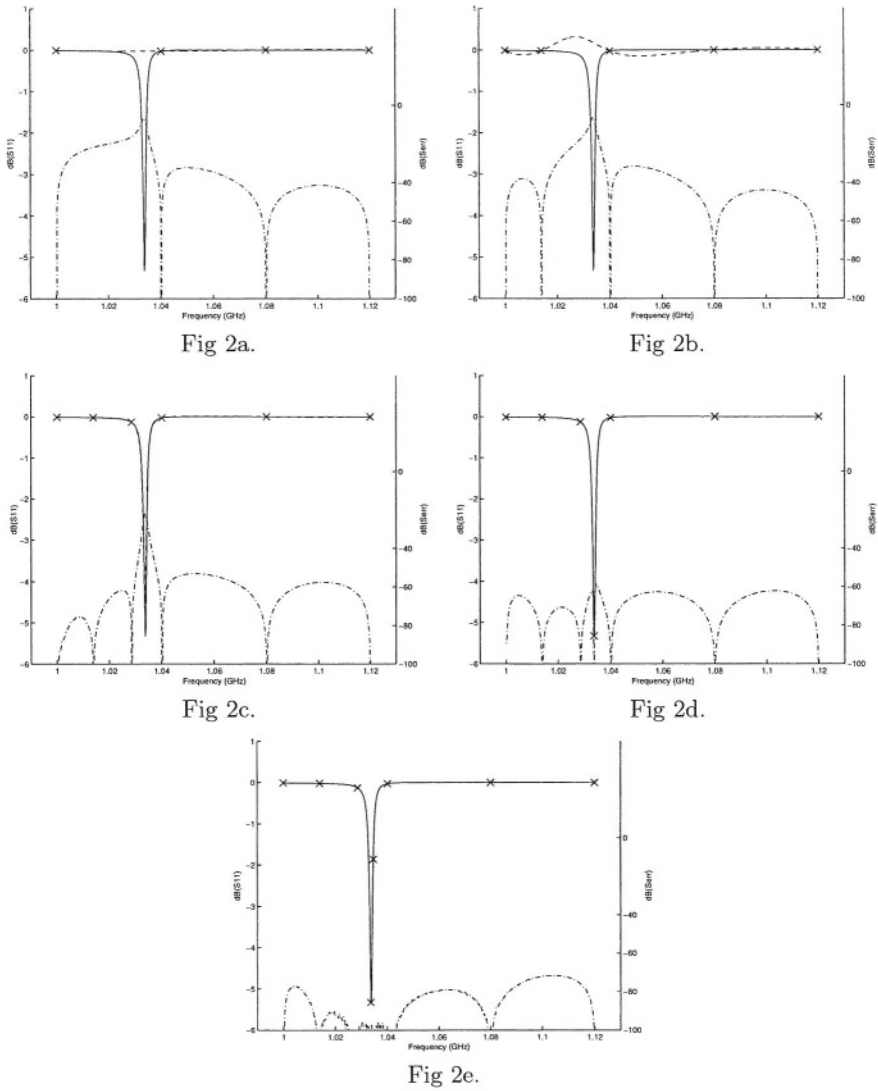


Fig. 2. Multiple steps of adaptive modeling process of a RingResonator, based on a sparse set of data samples (no frequency derivatives). The desired reference model is represented by a solid line. The selected samples are marked with a cross, and the "best" intermediate model is plotted as a dashed line. The real modeling error is plotted as a dash-dotted line (right axis).

In Figure 2a-e, the component is modeled using the default MBPE, i.e. without making use of frequency derivatives. The algorithm initially starts with 4 samples, equidistantly spaced over the frequency range of interest, and builds several interpolation models. The "best" model is shown as a dashed line. In

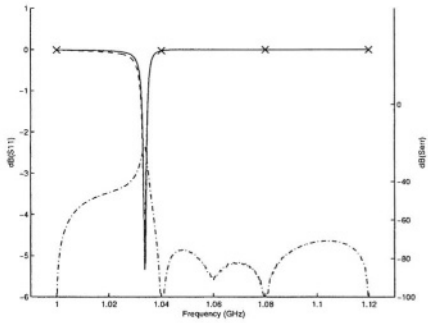


Fig 3a.

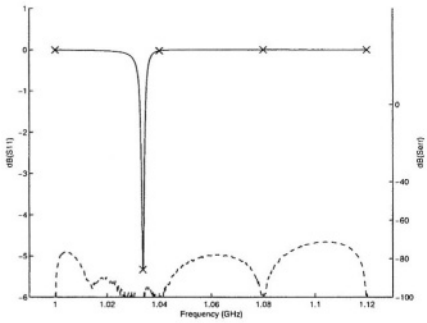


Fig 3b.

Fig. 3. Multiple steps of adaptive modeling process of a RingResonator, based on a sparse set of data samples + first frequency derivatives. The desired reference model is represented by a solid line. The selected samples are marked with a cross, and the "best" intermediate model is plotted as a dashed line. The real modeling error is plotted as a dash-dotted line (right axis).

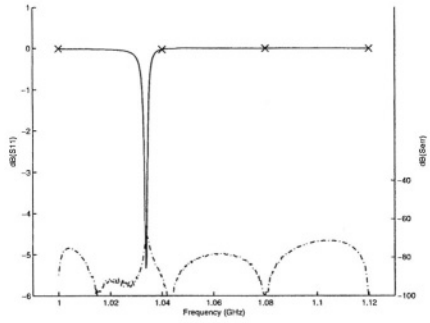


Fig. 4. Multiple steps of adaptive modeling process of a RingResonator, based on a sparse set of data samples + first and second frequency derivatives. The desired reference model is represented by a solid line. The selected samples are marked with a cross, and the "best" intermediate model is plotted as a dashed line. The real modeling error is plotted as a dash-dotted line (right axis).

each iteration of the algorithm, new samples are selected where the estimated error (the difference between 2 rational approximants, based on the same set of support samples) is maximal. Based on the extra data points, new rational models are built and evaluated, and the estimated error function is updated. The iterative process is repeated until the error is below a predefined accuracy threshold. In this example, 8 support samples are automatically selected, to obtain the required precision.

In Figure 3a-b, the first frequency derivatives are also used in the modeling process. This additional information is exploited by the algorithm, and now it

only needs 5 samples to find an accurate model. In this example, the extra sample was chosen very efficiently in the dip of the resonance.

In Figure 4, the second frequency derivatives are also used in the modeling process. The initial set of equally distributed samples already guarantee sufficient accuracy.

Of course, it can be useful to take also higher-order derivatives into account, especially if the behaviour of the system is highly dynamic. Typically, if more derivatives are used, fewer samples will be needed to get an accurate model. However, the value of the higher order derivatives in determining the model parameters is usually less than the value of lower order derivatives or samples, partially due to numerical issues.

6 Conclusion

In this paper, we extend the MBPE technique with frequency derivatives. During the adaptive modeling process, this reduces the number of required support samples significantly. This approach is particularly useful if the computational cost of simulating the required support samples is very high, and when the samples are selected adaptively. And so, an accurate model can be obtained, using a minimal set of data samples and their frequency derivatives. The stability of the algorithm is improved by a transformation of the rational model into a generalized Forsythe orthonormal basis, which makes the normal equations of the least squares system best conditioned.

Acknowledgement. This work was supported by the Fund for Scientific Research-Flanders.

References

1. Dhaene, T.: Automated Fitting and Rational Modeling Algorithm for EM-based S-parameter data. LNCS **2367,** Springer-Verlag, PARA 2002, Espoo (Finland) (2002) 99-105
2. Ansoft: Ansoft HFSS, Ansoft Corporation, Pittsburgh, PA
3. Miller, E. K.: Model Based Parameter Estimation in Electromagnetics: Part I. Background and Theoretical Development. IEEE Antennas and Propagation Magazine **40(1)** (1998) 42-51
4. Miller, E. K.: Model Based Parameter Estimation in Electromagnetics: Part II. Application to EM observables. Applied Computational Electromagnetics Society Newsletter, **11(1)** (1996) 35-56
5. Forsythe, G. E.: Generation and use of Orthogonal Polynomials for Data-Fitting with a Digital Computer. Journal of SIAM **5(2)** (1957) 74-88
6. Vandersteen, G.: Curve Fitting using Splines, Polynomials and Rational approximations : a comparative study. NORSIG 96, IEEE Nordic Signal Processing Symposium, Espoo (Finland) (1996) 41-44

7. Deschrijver, D., Dhaene, T.: Adaptive Knot Placement for Rational Spline Interpolation of Sparse EM-Based data. ICECOM 03, 17th International Conference on Applied Electromagnetics and Communications, Dubrovnik (Croatia) (2003) 433-436
8. Forsythe, G. E., Straus, E.G.: On Best Conditioned Matrices. Proceedings of the American Mathematical Society, **6** (1955) 340-345
9. Agilent EEsof Comms EDA: Momentum Software, Agilent Technologies, Santa Rosa, CA

Towards Efficient Parallel Image Processing on Cluster Grids Using GIMP*

Paweł Czarnul, Andrzej Ciereszko, and Marcin Frączak

Faculty of Electronics, Telecommunications and Informatics
Gdansk University of Technology, Poland
pczarnul@eti.pg.gda.pl, {cierech,marcin.f}@wp.pl
http://fox.eti.pg.gda.pl/~pczarnul

Abstract. As it is not realistic to expect that all users, especially specialists in the graphic business, use complex low-level parallel programs to speed up image processing, we have developed a plugin for the highly acclaimed GIMP which enables to invoke a series of filter operations in a pipeline in parallel on a set of images loaded by the plugin. We present the software developments, test scenarios and experimental results on cluster grid systems possibly featuring single-processors and SMP nodes and being used by other users at the same time. Behind the GUI, the plugin invokes a smart DAMPVM cluster grid shell which spawns processes on the best nodes in the cluster, taking into account their loads including other user processes. This enables to select the fastest nodes for the stages in the pipeline. We show by experiment that the approach prevents scenarios in which other user processes or even slightly more loaded processors become the bottlenecks of the whole pipeline. The parallel mapping is completely transparent to the end user who interacts only with the GUI. We present the results achieved with the GIMP plugin using the smart cluster grid shell as well as a simple round robin scheduling and prove the former solution to be superior.

1 Introduction

While both the cluster and grid architectures ([1]) and real grid software based on Globus ([2]) as well as image processing tools become mature and available for a wider range of systems and network topologies, it is still a difficult task to merge the two worlds in open NOWs. We investigate available solutions and make an attempt to process in parallel both loosely coupled and pipelined images on NOW systems. In the context of parallel image processing there is a need for an easy-to-use graphical user interface of a familiar application like Adobe Photoshop or the GIMP and an efficient but not too complex a tool for selection of best resources. Ideally, as we managed to achieve, the second stage is completely hidden from the user provided the prior configuration of the parallel environment. Following the Sun Grid Engine terminology ([3]) we define the cluster grid as a set of distributed resources providing a single point of entry (running GIMP in our approach) to users in an institution.

* Work partially sponsored by the Polish National Grant KBN No. 4 T11C 005 25

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 451–458, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Related Work

System or application level solutions which could assist in running either sequential or parallel image processing codes in parallel, range from system-level Mosix ([4]) which features process migration for load balancing for Linux boxes, cluster management systems like LoadLeveler, PBS, LSF ([5]) Condor ([6]) for exploiting idle cycles in shared networks up to Sun Grid Engine ([3]) which enables both queueing submitted HPC tasks and launching interactive processes on least loaded nodes allowing grid computing. There are libraries and environments like MPI, PVM ([7]) available for programmers. MatlabMPI ([8]) implements the MPI-like programming API for Matlab scripts to be run on parallel systems using the file system for communication. Although the latter three can assist in writing parallel image processing, none of them is by design coupled to specific parallel applications, including graphic design.

In regard to support for multithreaded image processing, in applications like Adobe Photoshop on SMP machines some filters like Gaussian Blur, Radial Blur, Image Rotate, Unsharp Mask can potentially benefit from many processors ([9]). However, some filters which can be accomplished fast can run even slower on SMP boxes than on a single processor due to the large synchronization overhead compared to the processing time. It has been confirmed that the multithreaded approach can speed up processing on the latest Intel HyperThreading processors ([10]) when running filters in Photoshop 7.0. [11] presents threaded GIMP plugins implementing Gaussian Blur.

[12] assumes, similarly to our approach, that it is unrealistic to expect the knowledge on parallelization issues from graphic specialists and provides the programmer with the architecture in which data parallel image processing applications can be coded sequentially, automatically parallelized and run on a homogeneous cluster of machines.

[13] presents a skeleton based approach for data parallel image processing in which only algorithmic skeletons coded in C/MPI need to be chosen for particular given low-level image operators to produce a parallel version of the code.

Finally the Parallel Image Processing Toolkit (PIPT, [14]) provides an extendable framework assisting image processing offering low-level API which can be incorporated into parallel MPI programs. This allows operations on chunks of images in parallel. Similarly, [15] describes a library of functions SPMDlib meant to help the development of image processing algorithms and a set of directives SPMDdir which are mapped by a parser to the library which provides an easy-to-use and high-level API.

3 Our Approach for Cluster Parallel Image Processing

In order to show the benefits of cluster computing in an open multi-user NOW, we have implemented an extension to the DAMPVM environment ([16], [17]) which provides the end user with a smart cluster shell which starts a shell command or a set of commands on best nodes in the network. Compared to Sun Grid Engine, it supports any platform PVM can run on, not only Solaris Operating Environments (on SPARC processors) and Linux x86 for Sun Grid Engine 5.3. Moreover, the DAMPVM sources can be modified to use a cluster even more efficiently. In view of this, the presented solution can easily use the monitoring/filtering infrastructure ([18]) and the divide-and-conquer features ([17]) of

DAMPVM. Thus it has been naturally extended to enable the GIMP plugin to launch processes on least loaded processors in a parallel environment incorporating run-time changes into decision making. The plugin can make use of the GIMP support for reading and writing various graphic formats like TIFF, PNG, JPEG etc. and leave the computing part for the efficient C++ code using PVM or other means of communication like MPI.

To prove the usefulness of this approach we have tested the following scenarios:

1. Starting command-line multi process conversion (using ImageMagick's `convert` utility) of large images in a network with varying loads. The cluster shell selects the least loaded nodes for `convert` to be run and thus optimizes the wall time of the simulation.
2. A GIMP plugin which invokes parallel pipelined processing using PVM applying a sequence of filters on images read by GIMP, all implemented by us within this work[1]. In corresponding external conditions, we prove that the allocation of pipeline processes to processors using the DAMPVM cluster shell can lead to a noticeable reduction of the execution time compared to round-robin/random allocation. This is visible even in lightly loaded networks where the DAMPVM shell chooses the least loaded nodes. In the pipeline, even small, difficult to notice processor loads appear to be a bottleneck for the pipeline which justifies our approach.

3.1 Speed Measurement and Task Queueing

In a heterogeneous network different but comparable processors like AMD Athlon XPs, MPs, Intel Pentium 4 (HT) can process different codes at different speeds. Secondly, the load of other users must be measured and filtered in order to hide high-frequency peaks corresponding to short-lived but CPU-bound actions like starting Mozilla etc. The latter was implemented in the DAMPVM runtime before ([18]). Within the scope of this work it was extended with precise speed measurement for specific code operations to reflect the real CPU-bound long application which follows the measurement. This can be the same processing command used on smaller images.

We applied the algorithm shown in Figure 1 to the DAMPVM cluster shell. When there are idle processors in the mixed single-processor, SMP system (as indicated by DAMPVM schedulers) the relevant information is stored in an array and the idle processors are assigned tasks successively with no delays. After the procedure has been completed, a 10-second delay is introduced to allow for the increased load being detected by the DAMPVM runtime. When all processors are still busy, every 2-second time slot a load check is performed. The values result from arbitrary delays to monitor load coded in the DAMPVM runtime. This scheme allows to use idle processors at once and queue pending supposedly processor and disk-bound image processing tasks in order not to overload the processor with too many processes resulting in process switching overhead. The load is monitored on every host by DAMPVM schedulers ([18]) and then collected asynchronously by a cluster manager using ring fashion communication. The load information includes: machine speed, the number of CPUs per node, idle percentages of the processors, CPU load by other users and the system, the number of processes, file storage available on the node, link start-up times and bandwidth.

[1] Download software from `http://fox.eti.pg.gda.pl/~pczarnul`

```
for(int o=0;o<iterations;o++) {
  if (nHowManyProcessorsAvailable<=0) // a trick -- if some processors were
    // available from the previous check then
    while ((nHowManyProcessorsAvailable=HowManyProcessorsAvailable())==0)
      sleep(2); // do not invoke HowManyProcessorsAvailable() again
  // now there is an idle processor out there so let's start the application
  // find new processors if all idle ones previously have been allocated tasks
  if ((nHowManyProcessorsAvailable<=0) || (firsttime)/* for the first time */) {
    MYIdleProcessors=GetIdleProcessors(MYhostnames,MYavailableprocessorspeeds);
    // now mark processors as idle
  } // find the best node among the previously selected processors ON WHICH NO
  // TASKS HAVE BEEN SPAWNED; spawn the task on the best node via a remote
  // launcher which sets the work requirements for this task on that node
  PC_Spawn("dampvmtasklauncher1",NULL,MYhostnames,1,&ptid,NULL);
  // mark processor "busy"; send the system command to the spawned launcher
  nHowManyProcessorsAvailable--;
  if (nHowManyProcessorsAvailable<=0) // after all the tasks have been spawned
    sleep(10); // - wait until the system becomes aware of the increased load
} PC_Recv();    // wait for confirmation from the tasks
```

Fig. 1. Task Queueing in DAMPVM Remote Shell

3.2 A GIMP Plugin for Pipelined Operations on Clusters

The main idea of the plug-in is to apply a set of up to ten filters to a large number of images. The images should be placed on one machine with the GIMP, connected via network to a cluster of computers running PVM, a cluster grid. The architecture of the plug-in consists of three layers (Figure 2):

- a Script-Fu wrapper,
- a supervisory GIMP-deployed plug-in,
- slave node modules in the proposed parallel pipelined architecture.

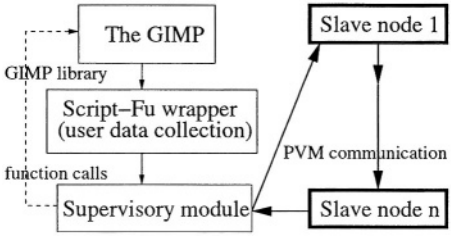


Fig. 2. GIMP Plugin's Architecture

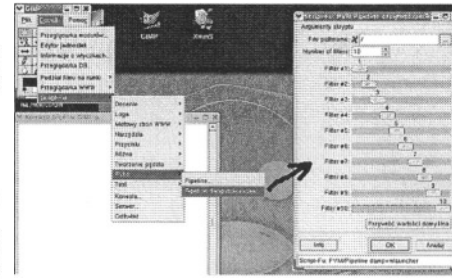


Fig. 3. GIMP's Pipelined Plugin Interface

The GIMP provides developers with an easy to use scripting language called Script-Fu, that is based on a scheme programming language. From the Script-Fu level you can run any of the GIMP library function including GIMP plug-ins properly registered in the PDB (GIMP's Procedural DataBase). Script-Fu also aids developers in creating of plug-in interfaces. The wrapper's function is to gather data from the user and relay them to the supervisory GIMP-deployed plug-in.

Both, the supervisory module and the slave module, are programs written in C using PVM for communication. The supervisory module calls the GIMP's library functions to open each image for processing, acquires raw picture data from the GIMP's structures, and passes it to the first of the pipeline nodes. This allows to combine the easy-to-use interface with the underlying parallel architecture. The images are then processed in the created slave node pipeline. The pipeline can be created in two ways: either by using PVM (Pipeline plug-in) in accordance with a static list of hosts created by the user, or dynamically, with assistance of PVM based DAMPVM remote shell (Pipeline dampvmlauncher plug-in) thus taking advantage of the speed and load information of individual nodes in creating the slave node pipeline.

The slave node module implements a series of image filters (3x3, 5x5 matrix area filters and simple non-context filters). Each slave applies a filter to the image and passes it on through the pipeline.

The interfaces for both plug-ins are identical, the difference lies in the code that is implemented in the supervisory module. The user invokes the plugin from the GIMP's context menu and specifies a path name for the files intended for filtering and the number of filters that will be applied to all the images. Figure 3 shows the plugin window superimposed on the context menu invoking it. Slide bars let the user specify the types of filters at the stages of the pipeline.

4 Experimental Results

We have performed lab experiments to prove that image processing on cluster grids can be successfully assisted by the load-aware remote DAMPVM shell. We used a cluster of 16 Intel Celeron 2GHz, 512MB machines interconnected with 100Mbps Ethernet.

4.1 Parallel Image Conversion with a Cluster Shell

In this case, we tested the ability of the DAMPVM runtime and the remote shell to detect least loaded nodes, spawn tasks remotely, queue pending tasks as described above and submit them when processors become idle. These features were tested in the scenario with one node loaded with other CPU-bound processes for which the results are shown in Figure 4. We ran ImageMagick's convert utility to convert a 4000 × 4000 48MB TIFF image to Postscript. On 1 processor, we ran the command without the remote shell. On $P > 1$ processors in Figure 4 there were $P + 1$ processors available, one of which was overloaded. The remote shell omitted the overloaded node. The scaled speed-up was computed as the single-processor execution time multiplied by P divided

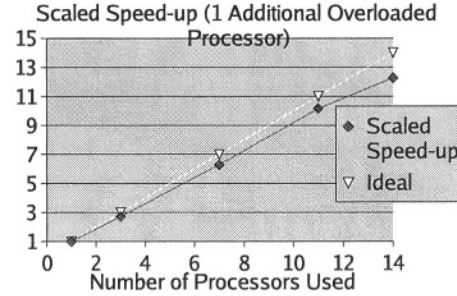


Fig. 4. Scaled Speed-up for Remote Shell `convert` Runs

by the time achieved on P processors with the load on the P + 1-th processor. The latter processor was chosen randomly. Less than ideal values but showing small overhead result from additional load monitoring, spawn and the queueing procedure.

4.2 Pipelined Image Filtering as a GIMP Plugin Using a Cluster

In this case we compared launching pipelined computations using a static allocation of pipeline stages to processors and the dynamic allocation using the remote shell.

In the case that work is distributed in an unloaded and homogeneous network the efficiency of both plugins should be similar. However, in a normal network environment such conditions are hard to achieve and thus it is expected that total work times will be considerably shorter for the DAMPVM module. It is not only in academic examples in which some node(s) is overloaded but also in a seemingly idle network. In the latter case, some system activities or a user browsing the Internet contribute to the processor usage and such a node becomes effectively the pipeline bottleneck if selected as a stage of the pipeline. The remote shell enabled the plugin to avoid placing pipeline nodes on machines that are already loaded with work (e.g. the computer running the GIMP).

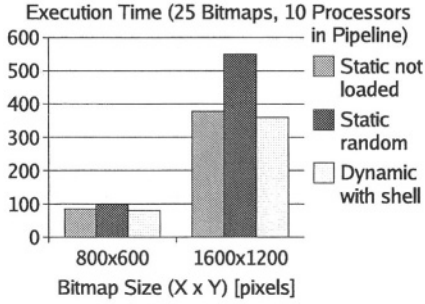


Fig. 5. Execution Time for Various Pipeline Allocation Methods by Image Size

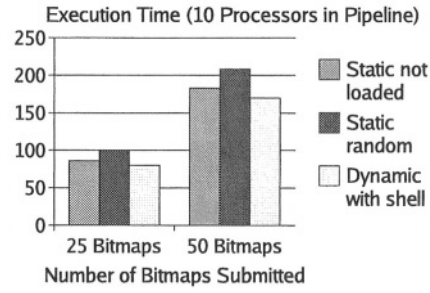


Fig. 6. Execution Time for Various Pipeline Allocation Methods by Number of Images

The variables in the pipelined simulations are as follows:

- the number of stages/processors in the pipeline *(P),*
- the number of images to process *(N),*
- the size of the images: may be of similar or different sizes,
- the type of filters at the stages: may be uniform or different in regard to the processing time.

Figure 5 presents results for 3 × 3 matrix filters (taking same time to complete) on P = 10 processors for N = 25 800 × 600 and 1600 × 1200 bitmaps while Figure 6 shows results for for 5 × 5 matrix filters (taking same time to complete) on P = 10 processors for N = 25 and N = 50 800 × 600 bitmaps. In all cases there were 14 idle processors available. On one of them there was the GIMP running, acting as a master host, on another one a user performed simple editing in Emacs. In the "Static not loaded"

case, the allocation was done statically by listing the available hosts in a file, from which 10 successive processors were chosen. The master host at the end of 14-processor list was thus omitted. In the second test ("Static random") the master host was acting as the first stage in the pipeline, both reading images through GIMP, processing it and passing to the second stage. Especially for large images this becomes a bottleneck since the master host also saves the results to the disk. Finally, in the "Dynamic with shell" example, the remote shell was launched to start 10 slave node processes on least loaded nodes. It automatically omitted both the master host and the processor busy with text editing although seemingly idle. Compared to the "Static not loaded" case it is visible that even small additional loads in the static allocation scenario slow down the pipeline, justifying the use of the remote shell.

The best theoretical speed-up of the pipeline is strictly bounded and, assuming no overhead for communication which is more apparent for larger bitmaps, can be estimated as $\frac{NP}{N-1+P}$ ($N = 25$, $P = 10$) which approximates to $\frac{250}{34} \approx 7.35$. The time obtained for 1 processor and 25 800 × 600 bitmaps and the 10-stage pipeline was 439s giving speed-up 5.49. This is due to costly communication and synchronization. It must be noted also that on 1 processor all the processes run concurrently which means that there are costly context switches and slow downs due to the GIMP's disk operations.

5 Summary and Future Work

We have presented software developments combining advanced pipelined filter processing of images selected in GIMP with parallel clusters showing improvement in the execution time when used with a load-aware remote shell rather than static process assignment. It is easy to select the proposed as well as add new filters to customize the graphic flow, for example, to perform the common sequence of operations on images transformed to thumbnails for WWW use, usually transformed from TIFF: adjust levels (can be pipelined itself), convert to the 16-bit format, adjust contrast, brightness, possibly saturation, scale the image, apply the unsharp mask, convert to JPEG.

As it is a more practical pipeline flow, we are planning to implement such a pipeline and execute it on a cluster of 128 2-processor machines at the TASK center in Gdansk, Poland as well. Note that the proposed approach can be used widely for pipelined image conversion for WWW gallery creation or assist in pipelined sequences of advanced graphic designers working with the GIMP. The implementation could also be extended to other popular applications like Adobe Photoshop.

References

1. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications **15** (2001) 200–222 http://www.globus.org/research/papers/anatomy.pdf.
2. Globus: Fundamental Technologies Needed to Build Computational Grids (2003) http://www.globus.org.
3. Sun Microsystems Inc.: Sun Grid Engine 5.3. Administration and User's Guide. (2002) http://wwws.sun.com/software/gridware/faq.html.

4. Barak, A., La'adan, O.: The MOSIX Multicomputer Operating System for High Performance Cluster Computing. Journal of Future Generation Computer Systems **13** (1998) 361–372
5. Platform Computing Inc.: PLATFORM LSF, Intelligent, policy-driven batch application workload processing (2003) http://www.platform.com/products/LSF/.
6. Bricker, A., Litzkow, M., Livny, M.: Condor Technical Summary. Technical report, Computer Sciences Department, University of Wisconsin-Madison (10/9/91)
7. Wilkinson, B., Allen, M.: Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers. Prentice Hall (1999)
8. Kepner, J.: Parallel Programming with MatlabMPI. MIT, Lexington, MA, U.S.A. (2003) http://www.ll.mit.edu/MatlabMPI/.
9. Marc Pawliger: Multithreading Photoshop (1997) http://www.reed.edu/~cosmo/pt/tips/Multi.html.
10. Mainelli, T.: Two cpus in one? the latest pentium 4 chip reaches 3 ghz and promises you a virtual second processor via intel's hyperthreading technology. PC World Magazine (Jan 2003)
11. Briggs, E.: Threaded Gimp Plugins (2003) http://nemo.physics.ncsu.edu/~briggs/gimp/.
12. Seinstra, F., Koelma, D., Geusebroek, J., Verster, F., Smeulders, A.: Efficient Applications in User Transparent Parallel Image Processing. In: Proceeding of International Parallel and Distributed Processing Symposium: IPDPS 2002 Workshop on Parallel and Distributed Computing in Image Processing, Video Processing, and Multimedia (PDIVM'2002), Fort Lauderdale, Florida, U.S.A. (2002) citeseer.nj.nec.com/552453.html.
13. Nicolescu, C., Jonker, P.: EASY-PIPE - An "EASY to Use" Parallel Image Processing Environment Based on Algorithmic Skeletons. In: Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS'01), Workshop on Parallel and Distributed Image Processing, Video Processing, and Multimedia (PDIVM'2001), San Francisco, California, USA (2001) http://csdl.computer.org/comp/proceedings/ipdps/2001/0990/03/099030114aabs.htm.
14. Squyres, J.M., Lumsdaine, A., Stevenson, R.L.: A Toolkit for Parallel Image Processing. In: Proceedings of SPIE Annual Meeting Vol. 3452, Parallel and Distributed Methods for Image Processing II, San Diego (1998)
15. Oliveira, P., du Buf, H.: SPMD Image Processing on Beowulf Clusters: Directives and Libraries. In: Proceedings of International Parallel and Distributed Processing Symposium (IPDPS'03), Workshop on Parallel and Distributed Image Processing, Video Processing, and Multimedia (PDIVM'2003), Nice, France (2003) http://csdl.computer.org/comp/proceedings/ipdps/2003/1926/00/19260230aabs.htm.
16. Czarnul, P.: Programming, Tuning and Automatic Parallelization of Irregular Divide-and-Conquer Applications in DAMPVM/DAC. International Journal of High Performance Computing Applications **17** (2003) 77–93
17. Czarnul, P.: Development and Tuning of Irregular Divide-and-Conquer Applications in DAMPVM/DAC. In: Recent Advances in Parallel Virtual Machine and Message Passing Interface. Number 2474 in Lecture Notes in Computer Science, Springer-Verlag (2002) 208–216 9th European PVM/MPI Users' Group Meeting, Linz, Austria, September/October 2002, Proceedings.
18. Czarnul, P., Krawczyk, H.: Parallel Program Execution with Process Migration. In: International Conference on Parallel Computing in Electrical Engineering (PARELEC'00), Proceedings, Quebec, Canada (2000)

Benchmarking Parallel Three Dimensional FFT Kernels with ZENTURIO*

Radu Prodan[1], Andreas Bonelli[2], Andreas Adelmann[3], Thomas Fahringer[4], and Christoph Überhuber[2]

[1] Institute for Software Science, University of Vienna, Liechtensteinstrasse 22, A-1090 Vienna, Austria
[2] Institute for Applied Mathematics and Numerical Analysis, Vienna University of Technology, Wiedner Hauptstrasse 8-10/1152, A-1040 Vienna, Austria
[3] Paul Scherrer Institut, CH-5232 Villigen, Switzerland
[4] Institute for Computer Science, University of Innsbruck, Technikerstrasse 25/7, A-6020 Innsbruck, Austria

Abstract. Performance of parallel scientific applications is often heavily influenced by various mathematical kernels like linear algebra software that needs to be highly optimised for each particular platform. Parallel multi-dimensional Fast Fourier Transforms (FFT) fall into this category too. In this paper we describe a systematic methodology for benchmarking parallel application kernels using the ZENTURIO experiment management tool. We report comparative results on benchmarking two three dimensional FFT kernels on a Beowulf cluster.

1 Introduction

The performance of parallel scientific applications is often influenced by various mathematical kernels like linear algebra software that needs to be optimised for high performance on each individual platform. Advanced parallel multi-dimensional Fast Fourier Transform (FFT) algorithms make use of such linar algebra software. While tools like the Automatically Tuned Linear Algebra Software (ATLAS) [1] are designed to automatically perform such optimisations, they usually have a strong narrow focus with limited hard-coded parametrisation and performance measurement options.

The ultimate goal of the work described by this paper is to evaluate several parallel FFT kernels for various configuration parameters like problem size, machine size, interconnection network, communication library, and target machine architecture. Our results will serve as input to a group of physicists at the Paul Scherrer Institut that have initiated and stimulated this work within the context of solving large scale partial differential equations. In this context, we formulated a generic methodology for benchmarking arbitrary software kernels for arbitrary configuration and run-time parameters using the ZENTURIO experiment management tool [2].

* This research is supported by the Austrian Science Fund as part of the Aurora project under contract SFBF1104.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 459–466, 2004.
© Springer-Verlag Berlin Heidelberg 2004

This paper is organised as follows. Section 2 describes the ZENTURIO experiment management tool in brief. Section 3 is the main section of the paper devoted to our FFT benchmark experiments conducted with ZENTURIO. Section 4 concludes the paper.

2 ZENTURIO Experiment Management Tool

ZENTURIO is a semi-automatic tool for multi-experimental performance and parameter studies of parallel applications on cluster an Grid architectures. ZENTURIO has been designed as a distributed Grid service-based architecture, presented in detail in [2]. In addition to the Grid-enabled architecture, ZENTURIO can be run in light-weight cluster mode too, where all Grid services are replaced by simple Java objects that run on the cluster front-end. This mode was used for conducting the FFT benchmarks described in this paper and will be outlined in the remainder of this section.

Existing conventional parameter study tools [3] restrict parameterisation to input files only. In contrast, ZENTURIO uses a directive-based language called ZEN [4] to annotate any application files and specify arbitrary application parameters (e.g. program variables, file names, compiler options, target machines, machine sizes, scheduling strategies, data distributions, software libraries), as well as a wide range of performance metrics (e.g. cache misses, load imbalance, execution, communication, or synchronisation time). Additionally, constraint directives are introduced to filter erroneous experiments with no semantical meaning (see Example 1).

Through a graphical *User Portal,* the user inputs the application files, together with the compilation and execution commands. Based on the ZEN directives, an *Experiment Generator* module automatically instruments the application and generates the corresponding set of experiments. The SCALEA [5] instrumentation engine, based on the Vienna Fortran Compiler front-end that supports the MPI, OpenMP, and HPF programming paradigms, is used to instrument the application for performance metrics. After each experiment has been generated, an *Experiment Executor* module is responsible for compiling, executing, and managing its execution. The Experiment Executor interacts at the back-end with a batch job scheduler like fork, LSF, LoadLeveler, PBS, Sun Grid Engine, Condor, GRAM, or DUROC, as supported by our implementation. After each experiment has completed, the application output results and performance data are automatically stored into a PostgreSQL-based relational *Experiment Data Repository.* High level performance overheads related to the MPI, OpenMP, or HPF programming paradigms are computed using a post-mortem performance analysis component of SCALEA. An *Application Data Visualiser* module of the User Portal, based on the Askalon Visualisation Diagram package [6], provides a graphical interface to automatically query the repository and generate customisable visualisation diagrams that show the variation of any output parameters or performance metrics as a function of arbitrary input parameters (expressed through ZEN annotations).

3 Three Dimensional FFT Benchmarks

Our goal for this paper is to show a generic methodology for benchmarking and comparative analysis of parallel application kernels using ZENTURIO. We exemplify our techniques in the context of two three dimensional FFT algorithms which we briefly describe in the following.

FFTW [7] is a portable subroutine library for computing the Discrete Fourier Transform (DFT) in one or more dimensions of arbitrary input size, and of both real and complex data. Existing benchmarks [8] performed on a variety of platforms show that FFTW's performance is typically superiour to that of other publicly available FFT software, and is even competitive with non-portable, highly optimised vendor-tuned codes. The power of FFTW is the ability to optimise itself on the machine it executes through some pre-defined codelets run by a planner function before calling the real FFT.

wpp3DFFT developed by Wes Petersen at ETH Zurich uses a generic implementation of Temperton's in-place algorithm [9] for an $n = 2^m$ problem size, with the particular focus of making the transpose faster. The optimised algorithm pays a flexibility price, which restricts both the problem matrix size and the machine size to powers of two.

All experiments have been conducted on a single Intel Pentium III Beowulf cluster at ETH Zurich, which comprises 192 dual CPU Pentium III nodes running at 500 MHz with 1GB RAM, interconnected through 100 MBit per second Fast Ethernet switches. The nodes are organised into 24 node frames interconnected through 1 GBit per second optical links. Future work will repeat the benchmarks on various other parallel platforms, including IBM SP.

3.1 ZEN Parameter Annotations

There are three problem parameters which we vary in our benchmarks:

Problem size ranging from 2^3 to 2^8, expressed through source file annotations. Large problem sizes could not be run due to the limited amount of memory available on one node.

Communication library expressed by the `MPIHOME` ZEN variable in the application Makefile (see Example 1). The communication libraries under comparative study are LAM and MPICH-P4 MPI implementations. Shared memory has been used for communication within each SMP node. The constraint directive insures the correct association between the MPI library location and the `mpirun` command script defined in the PBS script used to submit the application on the cluster.

Machine size ranging from 2^1 to 2^6 dual nodes, each node running two MPI processes, expressed through PBS script annotations.

The total execution time, the transpose time, and MPI performance overheads have been measured using the performance behaviour directives based on the SCALEA performance library. Since small FFT problems have extremely

short execution times (order of milliseconds), they are prone to perturbations from the operating system or other background processes with low nice priority. To avoid such consequences, we repeat each experiment for a long enough amount of time (five minutes) and compute the mean of all measurements.

Example 1 (Sample Annotated Makefile).

```
MPIHOME = /usr/local/apli/lam
#ZEN$ ASSIGN MPIHOME ={/usr/local/apli/lam, /usr/local/apli/mpich}
#ZEN$ CONSTRAINT INDEX MPIHOME == run.pbs:MPIRUN
. . .
$(EXEC): $(OBJS)
  $(MPI_HOME)/bin/mpicc -o $(EXEC) $(OBJS) $(LIBS)
```

3.2 Benchmark Results

The annotations described in Section 3.1 specify a total of 72 experiments, which were automatically generated and conducted by ZENTURIO separately for each FFT algorithm. After each experiment has finished, the performance data collected is automatically stored into the Experiment Data Repository. The Application Data Visualiser module of the User Portal is used to formulate SQL query against the data repository and generate customisable diagrams that display the evolution of arbitrary performance metrics as a function of application parameters, mapped to arbitrary visualisation axes.

Figures 1(a) and 1(b) display the speedup curves of the two FFT algorithms, normalised against the lowest machine size executed (2 dual nodes), as a sequential experiment was not available. The speedup is bad for small problem sizes for which large parallelisation deteriorates performance. Large problem sizes offer some speedup until a certain critical machine size.

The explanation for the poor speedup curves is given by the large fraction used by the transpose (region 2) and MPI overheads (i.e. `MPI_Sendrecv_replace` routine to interchange elements in transpose) from the overall execution time, as displayed in Figure 1 (c) (FFTW shows similar overhead curves). It is interesting to notice that both algorithms scale quite well until 16 dual nodes for a 2^8 problem size, after which the performance significantly degrades. The reason is the fact that larger machine sizes spawn across multiple cluster frames which communicate through 3 PCI switches, 2 Ethernet, and 2 Fast-Ethernet wires that significantly affect the transpose communication time. For small problem sizes, the execution time is basically determined by the transpose overhead that naturally increases proportional with the machine size (see Figures 1(d) and 1(e)). In contrast to wpp3dFFT, FFTW shows an interesting behaviour of keeping the transpose and the total execution time constant even for large machine sizes. The explanation is given by the load balancing analysis from Figure 2(a).

ZENTURIO offers a series of data aggregation functions, comprising maximum, minimum, average, or sum, for metrics measured across all parallel (MPI) processes or (OpenMP) threads of an application. Let $\mathcal{M}$ denote a performance

metric and $\mathcal{M}_i$ its measured instantiations across all n parallel processes or threads of a parallel application, $\forall\ i \in [1, n]$. We define the *load balance* aggregation function for the metric $\mathcal{M}$ as the ratio between the average and sum aggregation values: $\frac{\sum_{i=1}^{n} \mathcal{M}_i}{\sum_{i=1}^{n} \mathcal{M}_i}$.

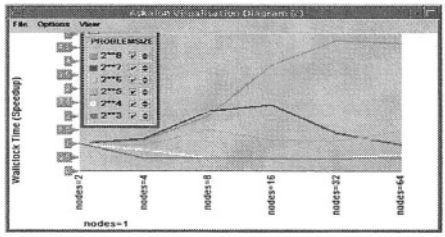
wpp3dFFT shows a good load balance close to 1 for all problem and machine sizes (see Figure 2(b)), while FFTW exhibits a severe load imbalance behaviour, the smaller problems are and the larger the machine sizes get (see Figure 2(a)). The explanation is the fact that FFTW in its planner function (that chooses optimised codelets for a certain platform) also detects that a machine size is too large for a rather small problem size to be solved. As a consequence, it decides to use only a subset of the processors for doing useful computation and transpose, while the remaining MPI processes simply exit with an `MPI_Finalize`. This explains the even execution time for small problem sizes shown in Figure 1(d).

Figure 1(f) shows a better performance of the LAM MPI implementation compared to MPICH for small problems and large machine sizes. Such experiments are bound to exchanging large number of small messages dominated by latencies, for which the LAM implementation seems to perform better. Large problem sizes shift the focus from message latency to network bandwidth, in which case both implementation perform equally well (see Figure 1(g)). Another suite of experiments currently under way on a local cluster at the University of Vienna shows that high speed interconnection networks (not available on the ETH cluster) like Myrinet give an approximate two fold improve in performance (see Figure 1(h)).

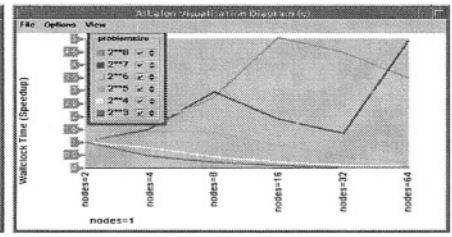
A comparative analysis of the two FFT parallel algorithms shows, as expected, a better performance of wpp3DFFT compared to FFTW for large problem sizes, which is due to the highly optimised wpp3DFFT transpose implementation (see Figure 2(c)). For small problem sizes, FFTW performs much better due to its intelligent run-time adjustment of machine size in the planning phase (see Figure 2(d)). The metric in which the ETH physicists are particularly interested is the ratio between the transpose and computation ratio, the latter being defined as the difference between the overall execution time and the transpose. This metric is comparatively displayed in Figures 2(e) and 2(f).

4 Conclusions

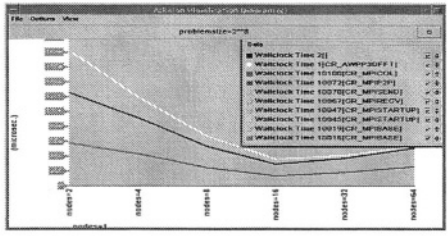
We have described a general methodology for benchmarking parallel application kernels using the ZENTURIO experiment management tool. We have applied the methodology for semi-automatic comparative benchmarking of two parallel FFT algorithms on a single Beowulf cluster. Parallel three dimensional FFT algorithms suffer from a severe communication bottleneck due to the highly expensive transpose (data communication) operation which increases proportional with the problem size. High performance networks like Myrinet are crucial for improving performance of such algorithms. LAM exhibits smaller latencies compared to MPICH in exchanging small messages, and similar bandwidth for large



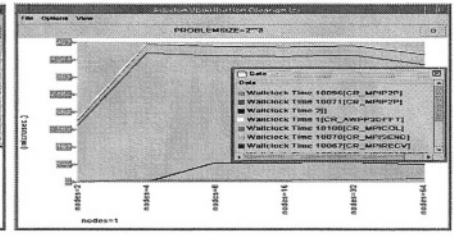
(a) FFTW Speedup.



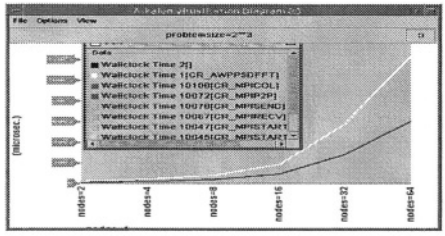
(b) wpp3DFFT Speedup.



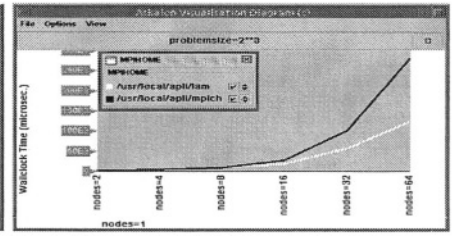
(c) wpp3DFFT Overheads (2^8 size).



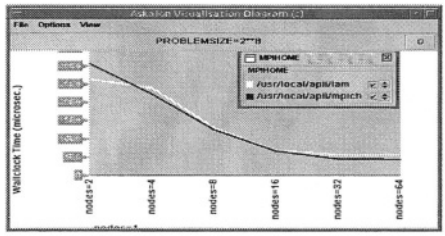
(d) FFTW Overheads (2^3 size).



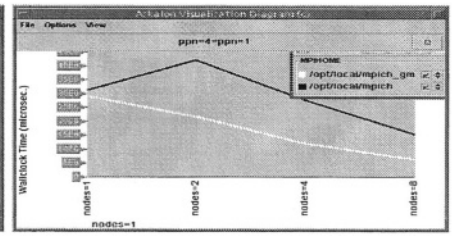
(e) wpp3DFFT Overheads (2^3 size).



(f) wpp3DFFT LAM versus MPICH.

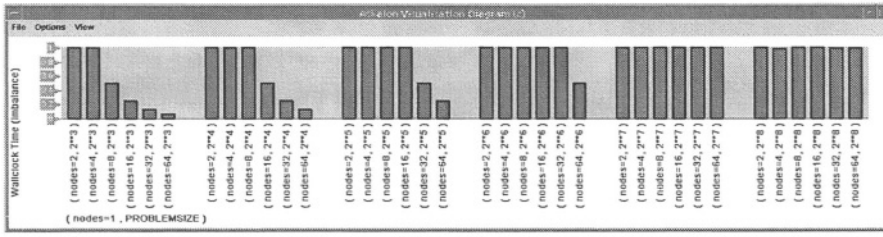


(g) FFTW LAM versus MPICH.

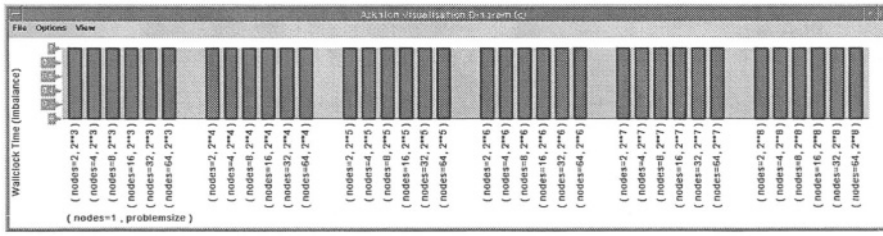


(h) Fast Ethernet versus Myrinet.

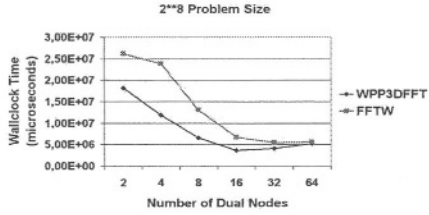
Fig. 1. Three Dimensional FFT Benchmark Results.



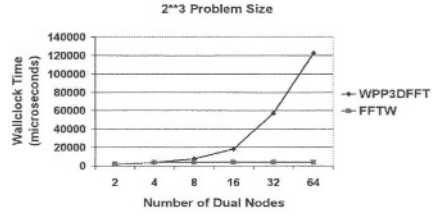
(a) FFTW Load Balance.



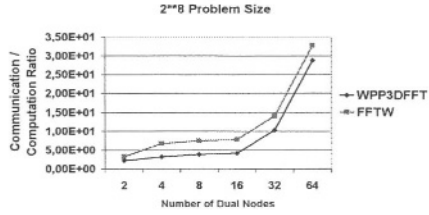
(b) wpp3DFFT Load Balance.



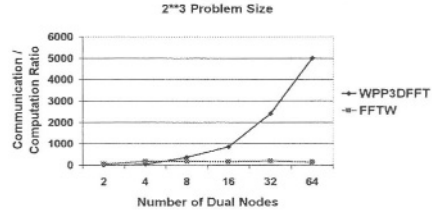
(c) FFTW versus wpp3DFFT.



(d) FFTW versus wpp3DFFT.



(e) FFTW versus wpp3DFFT.



(f) FFTW versus wpp3DFFT.

Fig. 2. Three Dimensional FFT Benchmark Results.

messages. wpp3DFFT algorithm performs better than FFTW in solving large FFTs due to an optimised transpose implementation. However, wpp3DFFT pays a flexibility price that restricts the problem and machine sizes to powers of 2, while FFTW can solve any problem size over arbitrary machine sizes. Smaller problem sizes are solved more efficiently by FFTW due to an intelligent run-time adjustment of machine size in the planning stage before calling the real FFT.

Future work will enhance ZENTURIO with a generic framework that employs standard heuristics like genetic algorithms and simulated annealing for solving a variety of NP complete optimisation problems, such as scheduling of single (workflow, MPI) Grid applications, as well as of large sets of applications for high performance throughput (complementary to [3]). The work will rely on existing Grid monitoring and benchmarking infrastructures like [10] and [11].

References

1. R. Clint Whaley and Jack J. Dongarra. Automatically Tuned Linear Algebra Software (ATLAS). In *Proceedings of the High Performance Networking and Computing Conference,* Orlando, Florida, 1998. ACM Press and IEEE Computer Society Press.
2. Radu Prodan and Thomas Fahringer. ZENTURIO: A Grid Middleware-based Tool for Experiment Management of Parallel and Distributed Applications. *Journal of Parallel and Distributed Computing,* 2003. To appear.
3. D. Abramson, R. Sosic, R. Giddy, and B. Hall. Nimrod: A tool for performing parameterised simulations using distributed workstations high performance parametric modeling with nimrod/G: Killer application for the global grid? In *Proceedings of the 4th IEEE Symposium on High Performance Distributed Computing (HPDC-95),* pages 520–528, Virginia, August 1995. IEEE Computer Society Press.
4. Radu Prodan and Thomas Fahringer. ZEN: A Directive-based Language for Automatic Experiment Management of Parallel and Distributed Programs. In *Proceedings of the 31st International Conference on Parallel Proces sing (ICPP 2002).* IEEE Computer Society Press, August 2002.
5. Hong-Linh Truong and Thomas Fahringer. SCALEA: A Performance Analysis Tool for Parallel Programs. *Concurrency and Computation: Practice and Experience,* 15(11-12):1001–1025, 2003.
6. T. Fahringer, A. Jugravu, S. Pllana, R. Prodan, C. Seragiotto, and H.-L. Truong. ASKALON - A Programming Environment and Tool Set for Cluster and Grid Computing. www.par.univie.ac.at/project/askalon, Institute for Software Science, University of Vienna.
7. Matteo Frigo and Steven G. Johnson. FFTW: An adaptive software architecture for the FFT. In *Proc. 1998 IEEE Intl. Conf. Acoustics Speech and Signal Processing,* volume 3, pages 1381–1384. IEEE, 1998.
8. Matteo Frigo and Steven G. Johnson. benchFFT. http://www.fftw.org/benchfft/.
9. Clive Temperton. Self-sorting in-place fast Fourier transforms. *SIAM Journal on Scientific and Statistical Computing,* 12(4):808–823, July 1991.
10. The CrossGrid Workpackage 2. Grid Application Programming Environment. http://grid.fzk.de/CrossGrid-WP2/.
11. Hong-Linh Truong and Thomas Fahringer. SCALEA-G: a Unified Monitoring and Performance Analysis System for the Grid. In *2nd European Across Grid Conference (AxGrids 2004),* Nicosia, Cyprus, Jan 28-30 2004.

The Proof and Illustration of the Central Limit Theorem by Brownian Numerical Experiments in Real Time within the Java Applet

Monika Gall, Ryszard Kutner, and Wojciech Wesela

Institute of Experimental Physics, Department of Physics, Warsaw University, Smyczkowa 5/7, PL-02678 Warsaw, Poland

Abstract. We developed the interactive Java applet[1] which made it possible to prove and illustrate by Brownian numerical experiments the Central Limit Theorem (CLT); the theorem which yet constitutes the basis for the Gaussian stochastic processes and the reference case for the non-Gaussian ones. Our approach emphasizes the contrast between theoretical complexity and simplicity provided by our probabilistic simulations. We argue that the present approach should be the preliminary stage for the advanced educational process before the analytical stage is developed. We stress that the Gaussian probability distribution function (PDF) is a stable as distinguished, e.g., from the delta-Dirac, Poisson and t-student ones which were also considered for comparison. The latter distribution was chosen so as to have all the moments higher than the second order diverging so as to verify the validity of CLT. As our experiments were performed in real time we were able to visualize the convergence of the processes (versus the size of the statistical ensemble of Brownian experiments) both to the variance of the sum of independent identically distributed random variables (which linearly increases with the number of the latter) and to the Gaussian PDF (which is the asymptotic distribution of this sum independently of the single-step PDF used here). We hope that our experimental approach will inspire students to undertake their own studies, e.g., to consider the non-Gaussian processes where CLT is violated which is the modern trend in statistical physics and its applications.

1 Motivation

It is commonly known that the limit theorems constitute one of the main trends in the theory of stochastic processes and their applications. Among them the Central Limit Theorem (CLT) plays a crucial role in broad applications covering many fields ranging from statistical physics across stochastic simulations to signal processing [1].

[1] This applet is available under the internet address: http://primus.okwf.fuw.edu.pl /erka/DIDACT/CTG BROWN/ .

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 467–474, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Roughly speaking, the CLT is one of the main statistical properties responsible for the main question, namely: what is the probability distribution function (PDF) of the sum of an unlimited number of (independent identically distributed) random variables? This problem was intensively and extensively studied in the last century by a huge number of researchers; these studies provided many handbooks, monographs, review papers and scientific works in mathematics and in natural, social and economical sciences. However, it is really surprising that up to now no Java applets were published as a modern educational tool directly verifying CLT and supporting, e.g., the distant learning of the above mentioned sciences.

In this paper we study the (unbiased) Brownian motion or random walk (hence, throughout this article we shall use the physical notation where random variables can be treated as displacements or increments of the Brownian particle). By using our Brownian numerical experiments we verify (with good approximation) the weakest version of CLT which should be adventageous both for the random walk theories and for various applications of CLT [2].

2 CLT and Possibilities of the Advanced Java Applet

There are many mathematical formulations of various versions (weaker or stronger) of the Central Limit Theorem. In the present work we consider the one, often used in natural, social and economical sciences.

General description. The user of the applet can choose a concrete distribution of single displacements of the walker from among the typical ones: (i) the delta-Dirac, (ii) the Gaussian, (iii) the Poisson and (iv) the Student's distribution offerred by the option *Distribution of single diplacements* (see e.g., Fig.1).

The Student's PDF was defined by three degrees of freedom so as to have all moments higher than the second order diverging (opposite to the first three distributions having all moments finite). The latter case is particularly interesting as the speed of convergence to the limit Gaussian distribution cannot be covered in this case neither by the well-known Chebyshev-Gnedenko-Kolmogorov expansion nor by the Berry-Esséen theorems 1 and 2 [3,4]. Thus the decisive role of the finite variance is emphasized and the validity of CLT is verified. Other PDFs (for example, so popular as the uniform, log-normal, and Gamma ones) can be also attached to the applet by the user himself as the Java is an object, platform independent programming and the source code of our applet is available (our applet is working under the Java 2 version 1.4.1.1 or higher).

The fundamental step of each experiment is to draw the length of a single, independent displacement of the walker from the chosen PDF common for all steps and all experiments. This is the initial assumption of the CLT. For example, in this work we present the results for Gauss and Poisson PDFs. Since in all our cases time is homogeneous, the most general version of the CLT, which assumes various distributions for different steps of the walker, cannot be considered here.

The asimuth of the displacement is always drawn from the uniform probability distribution function as we consider unbiased (isotropic) random walks (i.e. the unbiased version of the CLT). Hence, as the number of steps n is again introduced by the user (by applying the option *Number of particle steps,* see e.g., Fig.1), the characteristic zig-zag trajectory can be designed in each experiment together with the total displacement $\mathbf{R}(\mathbf{n})$, marked by the interval which connects the beginning (empty circle) and the end (full circle) of the trajectory.

In this way we illustrate the Brownian motion i.e. an incessant and irregular motion of the particle immersed in an ambient medium (which is invisible here), e.g. suspended in a fluid. Each single step of the Brownian particle (defined by consequent turning points) is characterized by the abruptness, hence the velocity of the particle is a quantity practicaly unmeasurable so a less definite description was used. The key point of the theory of Brownian motion was given by Smoluchowski who found that sufficiently large fluctuations, although relatively rare, are still common enough to explain the phenomenon [2].

Detailed description. The total displacement $\mathbf{R}(n)$ is composed of n single, elementary displacements (increments) $\mathbf{r}_j$, $j = 1 \ldots, n$,

$$\mathbf{R}(n) = \sum_{j=1}^{n} \mathbf{r}_j; \tag{1}$$

of course, the CLT involves a fundamental constraint

$$\sigma^2 = \langle (\mathbf{r}_j)^2 \rangle_{pdf} = \langle \mathbf{r}^2 \rangle_{pdf} < \infty, \; j = 1, 2, 3, \ldots, \tag{2}$$

obeyed for any step j and for each distribution mentioned above (here $\langle \ldots \rangle_{pdf}$ means the average over the single-step PDF).

The basic, global aim of the applet is to construct a statistical ensemble (series) of the mentioned above independent experiments within which the particle performs a two-dimensional random walk. For example, in the frame of the left window in all the enclosed figures the zig-zag trajectories of the walker are shown. Note that by using the zoom, the trajectory can be enlarged 2 or 4 times to allow observation of its structure (e.g., some elements of self-similarity in the stochastic sense). The number of such experiments as well as the number of particle steps common for all experiments is declared by the user at the beginnig. Usually, the applet is able to conclude series of experiments (consisting of hundred thousands) sufficient for a good statistics within a few minutes while their visualization is performed (during the simulation) as often as it is set by the user (by using the option *Snapshot picture after each ... experiments,* cf. Fig.1). Moreover, the option *Delay* (again see Fig.1) makes it possible to execute a single experiment sufficiently slowly to observe, e.g., elementary displacements of the Brownian particle. By setting this statistical ensemble of experiments the required averages (just over the statistical ensemble) and required statistics can be easily constructed (cf. the final stages presented in Figs.1 - 3).

In fact, the applet calculates two time-dependent quantities: (i) the mean-square displacement (MSD) of the walker $\langle [\mathbf{R}(t)]^2 \rangle_{se}$ (here $\langle \ldots \rangle_{se}$ is the *L*-

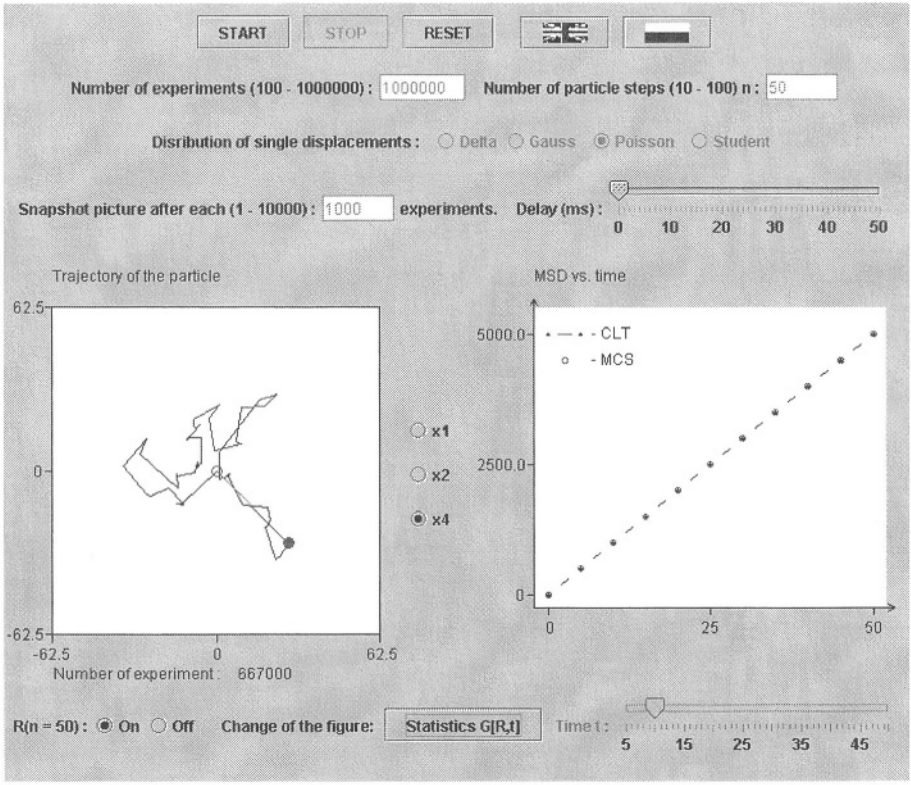


Fig. 1. Snapshot of the screen's picture of a single experiment of the Brownian motion together with the MSD vs. time of the Brownian particle almost for the final stage (after 667000 experiments).

dependent average over the statistical ensemble where L is the size of this ensemble, cf. Sec.3) and (ii) the statistics (in the form of the histogram) $G(\mathbf{R};t)$ of the total displacement passed within time t[2]. These two quantities are the main ones considered by the CLT.

The right window of the applet is working in real time in two modes, which are selected by the user: the first, which visualizes the preparation of the MSD vs. time t and the second one, which analogously shows the statistics at time t which is (almost) smoothly fixed by using the applet's slider. In both cases the applet shows the movie, which is the more stable the greater is the size of the statistical ensemble (L); this is a reminiscence of the Law of Large Numbers (LLN) [5].

[2] Here time $t = \tau n$, where τ is the time needed by the walker to perform a single step, common for all steps (in each experiment we put $\tau = 1$); this defines, the so called 'discrete-time' random walks.

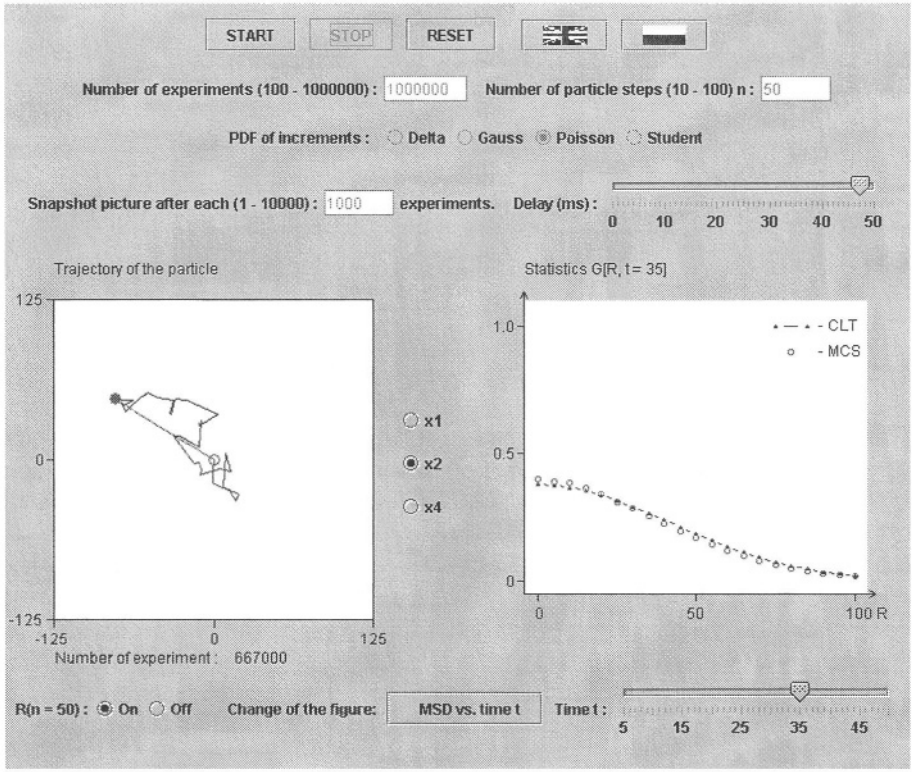


Fig. 2. Snapshot of screen's picture of a single experiment of the Brownian motion together with the statistics $G(R,t)$ of the Brownian particle displacements for long time $(t = 35)$ almost the final stage (after 667000 experiments).

By using our applet we exploit conditions sufficient for the Central Limit Theorem to occur.

3 The Algorithm and Results

The applet constructs a statistical ensemble of independent (similar) experiments within which the walker performs two-dimensional random walks; all necessary details of these walks are recorded.

Preparation of the MSD. By using the statistical ensemble, the mean-square displacement is easily calculated

$$\langle[\mathbf{R}(m)]^2\rangle_{se} = \frac{1}{L}\sum_{l=1}^{L}[\mathbf{R}^l(m)]^2,\ m = 1, 2, 3, \ldots, n, \tag{3}$$

where m is the current number of the walker steps common for all experiments, which increases up to its maximal value n set before the applet starts, and

analogously L is the current number of experiments which increases up to its maximal value set by the user at the beginning. The MSD is currently presented in the right window of the applet within the MSD mode. As the parameters L and m increase up to their maximal values we can observe convergence of the MSD to its stable limit which is just a reminiscence of the LLN.

Preparation of statistics. As the real space is isotropic we can use the followin formula to simplify the relation which constitutes the basis for the histogram and to remarkably improve the statistics

$$\frac{L(R, R+\Delta R; t)}{L} = \int_0^{2\pi} d\phi \int_R^{R+\Delta R} dR' R' \cdot PDF(R'; t)$$
$$\approx 2\pi \Delta R \cdot (R + \frac{\Delta R}{2}) \cdot PDF(R_\xi; t), \qquad (4)$$

where quantity $L(R,\ R+\Delta R; t)$ is the number of experiments (from the statistical ensemble) in which we found the particle within the ring of inner radius R and thickness ΔR after t steps. Hence, $L(R,\ R+\Delta R; t)/L$ estimates the (related) probability of finding the walker within the ring of inner radius R and thickness ΔR at time t; here $R = \mid \mathbf{R} \mid$ is the length of the position vector $\mathbf{R}$ and the thickness $\Delta R \ll \sigma$. The radius R_ξ obeys inequalities $R \leq R_\xi \leq R + \Delta R$ and was chosen so as the experimental statistics fulfilled the normalization condition. It is seen from relation (4) that we calculate the searched one-sided probability distribution function, $PDF(R(\approx R_\xi); t)$, as a function of $R(\geq 0)$ for various times t; this is visualized in the right window by using the applet slider given by option *Time t*.

Results. The final (in practice) situation after 667000 experiments (for the Poisson PDF of single-step increments) is shown in Figs.1 and 2. In these figures the final stage of convergence is observed; a good agreement between experimental results and theoretical predictions is well seen, which proves experimentally the Central Limit Theorem. The discrepancy can be observed only for the short time range ($t \approx 10$). Althoug the data shown in these figures concern the Poisson PDF for a single walker step, the results for other PDFs are very similar; they differ only by the speed of convergence to the asymptotic MSD and the Gaussian PDF. As for the Gaussian PDF of single-step increments the convergence is quickest (cf. Fig.3 where time $t = 5$ was sufficient) a comment should be made.

There is a characteristic difference between the Gaussian and other PDFs, namely, the Gaussian one is a stable distribution in the sense that PDF of $\mathbf{R}(n)$ is again a Gaussian probability distribution function for any $n \geq 1$. This effect can be observed by using the applet where the results (for the Gaussian PDF of single-step increments) can be shown, for example, for $n = 5,\ 10$ and 40.

For any case, the final stage of statistics requires a much larger size of the statistical ensemble than that required by the time-dependent mean-square displacement; this can be easily observed within the applet where after about $L \approx 10000$ the MSD vs. time well agrees with the theoretical prediction but not the statistics which still exhibits too large scattering of the data points. However, the speed of convergence to the Gaussian PDF for increasing L is the greatest for the case of the Gaussian PDF of single-step increments.

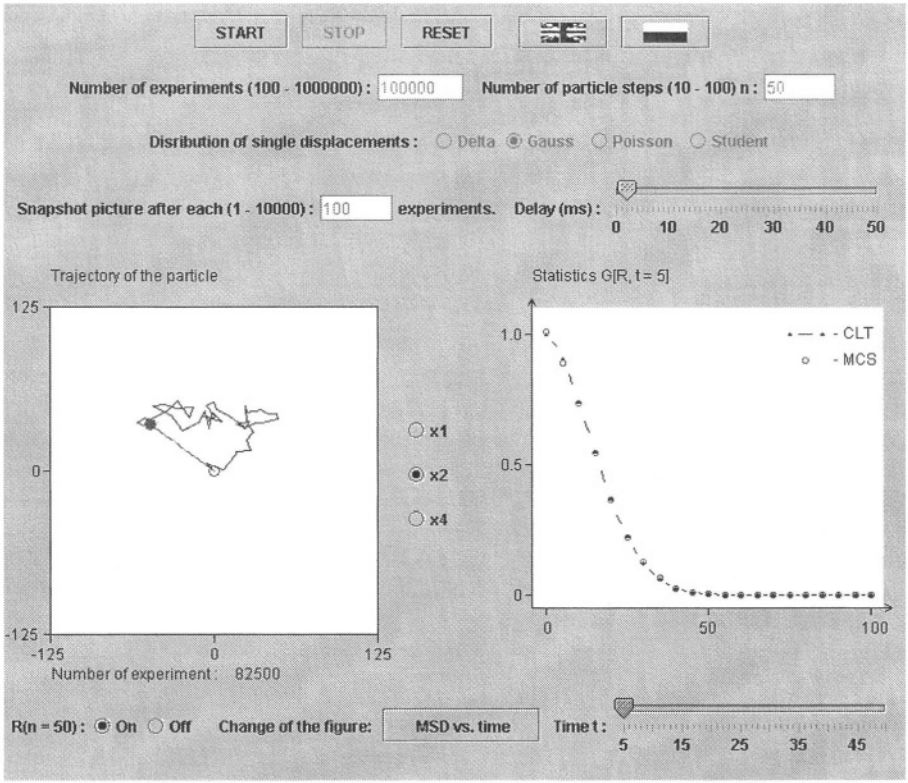


Fig. 3. Snapshot of the screen's picture of a single experiment of the Brownian motion together with the statistics $G(R, t = 5)$ of the Brownian particle displacements for the final stage (after 82500 experiments).

4 Concluding Remarks

It is remarkable that so rich numerical results were obtained by using a so simple probabilistic algorithm [6]. Our algorithm obeys two constraints fundamental for our considerations. The first one

$$\lim_{n\to\infty} \sum_{j=1}^{n} \frac{\langle [\mathbf{r}_j]^2 \rangle_{pdf}}{j^2} = \zeta(2) \cdot \langle \mathbf{r}^2 \rangle_{pdf} < \infty, \tag{5}$$

(where ζ (...) is the Riemann ζ-function) is sufficient for the strong Kolmogorov Law of Large Numbers to occur as we took into account only the PDFs having a finite variance of single-step displacements; the variance is identical for all successive steps (as a consequence of time homogeneity).

The second constraint

$$\lim_{n\to\infty} \frac{\langle [\mathbf{R}(n)]^2 \rangle_{se}}{n^2} = \lim_{n\to\infty} \frac{\langle \mathbf{r}^2 \rangle_{pdf}}{n} = 0 \tag{6}$$

is sufficient for both the Central Limit Theorem anf the Law of Large Numbers to occur, where the former quantitatively and the latter qualitatively were verified by our numerical experiments.

Our main result clearly illustrates the concept of the **basin of attraction** in the functional space of probability distribution functions. To describe this concept we focus our attention on delta-Dirac, Gaussian, Poisson and Student distributions which convergence to the Gaussian PDF, which is the attractor of all these distributions. Their convergence is practically quite rapid as need only dozen single steps of the walker. Note that only the Gaussian PDF remains the Gaussian along the trajectory in the functional space, which illustrates the concept of the **stable distribution** (i.e., its shape is preserved all the time); in this sense other distributions considered by us are unstable. Though the Gaussian attractor is, maybe, the most important one in functional space of PDFs, other (stable and unstable) attractors also exist and play an increasing role in probabilistic theories and their applications [7,8,9].

References

1. Sornette D.: Critical Phenomena in Natural Sciences. Chaos, Fractals, Selforganization and Disorder: Concepts and Tools. Springer-Verlag, Berlin 2000
2. Mazo R.M.: Brownian Motion. Fluctuations, Dynamics and Applications. Clarendon Press, Oxford 2002
3. Mantegna R.N., Stanley H.E.: An Introduction to Econophysics. Correlations and Complexity in Finance. Cambridge Univ. Press, Cambridge 2000
4. Bouchaud J.-P., Potters M.: Theory of Financial Risks. From Statistical Physics to Risk Management. Cambridge Univ. Press, Cambridge 2001
5. Feller W.: An introduction to probability theory and its applications. Vol.I. J. Wiley & Sons, New York 1961
6. Landau D.P., Binder K.: Monte Carlo Simulations in Statistical Physics. Cambridge Univ. Press, Cambridge 2000
7. Bouchaud J.-P., Georges A., Anomalous Diffusion in Disordered Media: Statistical Mechanisms, Models and Physical Applications. Phys. Rep., Vol.195 (1990) 127-293
8. Shlesinger M.F., Zaslavsky G.M., Frisch U., (Eds.): Lévy Flights and Related Topics in Physics. LNP, Vol.450. Springer-Verlag, Berlin 1995
9. Pęka1ski A., Kutner R., Sznajd-Weron K., (Eds.): Anomalous Diffusion. From Basics to Applications. LNP Vol.519. Springer-Verlag, Berlin 1999

An Extended Coherence Protocol for Recoverable DSM Systems with Causal Consistency*

Jerzy Brzeziński and Michał Szychowiak

Institute of Computing Science
Poznań University of Technology
Piotrowo 3a, 60-965 Poznań, POLAND
phone: +48 61 665 28 09, fax: +48 61 877 15 25
{jbrzezinski,mszychowiak}@cs.put.poznan.pl

Abstract. This paper presents a new checkpoint recovery protocol for Distributed Shared Memory (DSM) systems with read-write objects. It is based on independent checkpointing integrated with a coherence protocol for causal consistency model. That integration results in high availability of shared objects and ensures fast restoration of consistent state of the DSM in spite of multiple node failures, introducing little overhead. Moreover, in case of network partitioning, the extended protocol ensures that all the processes in majority partition of the DSM system can continuously access all the objects.

1 Introduction

One of the most important issues in designing modern Distributed Shared Memory (DSM) systems is fault tolerance, namely recovery, aimed at guaranteeing continuous availability of shared data even in case of failures of some DSM nodes. The recovery techniques developed for general distributed systems suffer from significant overhead when imposed on DSM systems (e.g. [3]). This motivates investigations for new recovery protocols dedicated for the DSM. Our research aims at constructing a new solution for the DSM recovery problem which would tolerate concurrent failures of multiple nodes or network partitioning. In [2] we have proposed the concept of a coherence protocol for *causal consistency model* [1] extended for low cost checkpointing which ensures fast recovery. To the best of our knowledge it is the first checkpoint-recovery protocol for this consistency model. In this paper we present a formal description of the protocol as well as the proof of its correctness.

This paper is organized as follows. In section 1 we define the system model. Section 3 details a new coherence protocol extended with checkpointing in order to offer high availability and fast recovery of shared data. The protocol is proven correct in section 4. Concluding remarks are given in section 5.

* This work has been partially supported by the State Committee for Scientific Research grant no. 7T11C 036 21

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 475–482, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Basic Definitions and Problem Formulation

2.1 System Model

The DSM system is a distributed system composed of a finite set P of sequential processes P1, P2, ..., Pn that can access a finite set O of shared objects. Each shared object consists of its current state (object value) and object methods which read and modify the object state. We distinguish two operations on shared objects: read access and write access. The read access ri(x) to object x is issued when process Pi invokes a read-only method of object x. The write access wi(x) to object x is issued when process Pi invokes any other method of *x*. Each write access results in a new object value of *x*. By $r_i(x)v$ we denote that the read operation returns value v of x, and by $w_i(x)v$ that the write operation stores value v to x.

The replication mechanism is used to increase the efficiency of the DSM object access, by allowing each process to locally access a replica of the object. However, concurrent access to different replicas of the same shared object requires consistency management. The coherence protocol synchronizes each access to replicas, accordingly to the appropriate consistency model. This protocol performs all communication necessary for the interprocess synchronization via message-passing.

2.2 Causal Memory

Let *local history* H_i denote the set of all access operations issued by P_i, *history* H – the set of all operations issued by the system and HW – the set of all write operations.

Definition 1. *The* ***causal-order*** *relation in H, denoted by* $\rightarrow$, *is the transitive closure of the local order relation* $\rightarrow_i$ *and a write-before relation that holds between a write operation and a read operation returning the written value:*

(i) $\forall_{o^1, o^2 \in H} \left(\left(\exists_{i=1..n} o^1 \rightarrow_i o^2 \right) \Rightarrow o^1 \rightarrow o^2 \right)$,

(ii) $\forall_{x \in O} w(x)v \rightarrow r(x)v$,

(iii) $\forall_{o^1, o^2, o \in H} \left((o^1 \rightarrow o \ \wedge \ o \rightarrow o^2) \Rightarrow o^1 \rightarrow o^2 \right)$.

As P_i is sequential, it observes the operations on shared objects in a sequence which determines a local serialization $\mapsto_i$ of the set $H_i \cup HW$.

Definition 2. *Execution of access operations is* ***causally consistent*** *if serialization* $\mapsto_i$ *satisfies the following condition:*

$$\forall_{o^1, o^2 \in H_i \cup HW} \left(o^1 \rightarrow o^2 \Rightarrow o^1 \mapsto_i o^2 \right).$$

The causal consistency model guarantees that all processes accessing a set of shared objects will perceive the same order of causally related operations on those objects.

3 The Integrated Coherence-Checkpointing Protocol

We describe now the integrated coherence-checkpointing protocol, CAUSp, which is an extension of a basic coherence protocol originally proposed in [1]. The basic protocol ensures that all local reads reflect the causal order of object modifications, by *invalidating* all potentially outdated replicas. If at any time, process P_i updates an object x, it determines all locally stored replicas of objects that could have possibly been modified before x, and invalidates them, preventing from reading inconsistent values. Any access request issued to an invalid replica of x requires fetching the up-to-date value from a *master replica* of x. The process holding a master replica of x is called x's *owner*. We assume the existence of reliable *directory services* which can provide a process with the identity of the current owner of any required object.

3.1 The CAUSp Protocol

The CAUSp protocol distinguishes 3 ordinary states of an object replica: *writable* (indicated by the WR status of the replica), *read-only* (RO status), and *invalid* (INV status). Only the WR status enables to perform instantaneously any write access to the replica. However, every process is allowed to instantaneously read the value of a local replica in either RO or WR state. Meanwhile, the INV status indicates that the object is not available locally for any access. Thus, the read or write access to the INV replica, and the write access to the RO replica require the coherence protocol to fetch the value of the master replica of the accessed object.

The causal relationship of the memory accesses is reflected in the vector timestamps associated with each shared object. Each process P_i manages a vector clock VT_i. The value of i-th component of the VT_i counts writes performed by P_i. More precisely, only intervals of write operations not interlaced with communication with other processes are counted, as it is sufficient to track the causal dependency between operations issued by distinct processes.

There are three operations performed on VT_i:

- $inc(VT_i)$ – increments a i-th component of the VT_i; this operation is performed on write-faults and read requests from other processes;
- $update(VT_i,VT_j)$ – returns the component wise maximum of the two vectors; this operation is performed on updating a local replica with some value received from another process;
- $VT_i<VT_j$ – true iff $\forall k$: $VT_i[k]\leqslant VT_j[k]$ and $\exists k$: $VT_i[k]\neq VT_j[k]$

The replica of object x stored at P_i has been assigned a vector timestamp VT_i^x. The $local_invalidate_i(VT)$ operation ensures the correctness of the protocol, by setting to INV the status of all RO replicas x held by P_i, for which $VT_i^x<VT$ is true.

Checkpoints are stored in DSM as *checkpoint replicas* denoted C (checkpoint) and ROC (read-only checkpoint). The identities of DSM nodes holding checkpoint replicas are maintained in CCS (*checkpoint copyset*). CCS(x) is initiated at the creation of x, then maintained by the object owner, but never includes the owner. The content of CCS(x) can change accordingly to further access requests or failure pattern, or any load balancing mechanisms, but the number of checkpoint replicas should always en-

sure the desired failure resilience [2]. Checkpoint replica is updated on checkpoint operations and never becomes invalidated. Checkpoint operation *ckpt(x)v* consists in atomically updating all checkpoint replicas held by processes included in CCS(x) with value v, carried in CKPT(x,v,VT^x) message, and setting their status to C. After that moment, any C replica can be switched to ROC on the next local read access to x. Until the next checkpoint, the ROC replica serves read accesses as RO replicas do (as the checkpoint replica holds a prefetched value of x).

Actions of the extended protocol are presented in Fig. 1.

3.2 Delayed Checkpointing and Burst Checkpointing

The checkpointing is performed independently for each process. Moreover, several modifications of objects can be performed without the need for taking checkpoints. Indeed, when P_i issues several subsequent writes to x (which is generally a typical behavior resulting from program locality), the checkpoint is not necessary as far as this sequence is not interrupted by any read access from another process. However, it is necessary to remark, that at the moment of checkpointing x, P_i can also own some other object y, which has been modified before the last modification of x. Then, if P_i fails after checkpointing x but before checkpointing y, the consistency of the memory could be violated on recovery, since the formerly checkpointed value of y will be inconsistent with the checkpointed value of x. Therefore, on each checkpoint of a dirty object, P_i is required to atomically checkpoint all dirty replicas *(burst checkpointing)*.

3.3 Recovery

Before any failure occurs there are at least $nr_{min}+1$ replicas of x, thus in case of a failure of $f \leqslant nr_{min}$ processes (at most f processes crashes or become separated from the majority partition) there will be at least one non-faulty replica of x available. If the directory manager discovers the unavailability of x's owner, it sequentially contacts processes included in CCS(x) and elects the first available as new owner. In case of the network partitioning, there is at least one such a process in the primary partition.

In order to protect the causal consistency of further access operations, the local invalidation operation must be processed on recovering an object from its checkpoint. The elected owner will perform $local_invalidate_i(x.VT)$ for each x recovered from a C state replica. This operation is presented in Fig. 2.

4 Correctness of the Protocol

Due to space limitation we will here provide the proof of the safety property only. The safety property asserts that the CAUSp protocol correctly maintains the coherency of shared data, accordingly to the causal consistency model, besides any allowable failures. For the sake of simplicity of the presentation, the recovery operation $recover_i(x)v$ is considered as a read access operation $r_i(x)v$ from the safety viewpoint.

```
1.  on r_i(x)v :          if x.state = C
2.                            local_invalidate_i(x.VT)
3.                            VT_i := update(VT_i,x.VT)
4.                            x.state := ROC
5.                        if x.state = INV
6.                            send R_REQ(x.id) to x.owner
7.                            wait for UPD(x,VT_k)
8.                            local_invalidate_i(VT_k)
9.                            VT_i := update(VT_i,VT_k)
10.                           x.VT := VT_k
11.                           x.state := RO
12.                       v := x.value

13. on w_i(x)v :          if x.state ≠ WR
14.                           send W_REQ(x.id) to x.owner
15.                           wait for ACK(x.id)
16.                           inc(VT_i)
17.                           x.VT := VT_i
18.                           x.state := WR
19.                           x.owner := P_i
20.                       x.value := v
21.                       x.dirty := true

22. on R_REQ(id) from P_j:    inc(VT_i)
23.                           find x :: x.id = id
24.                               ckpt_i(x)
25.                               send UPD(x, VT_i) to P_j

26. on W_REQ(id) from P_j:    find x :: x.id = id
27.                               ckpt_i(x)
28.                               x.state := RO
29.                           send ACK(x.id) to P_j

30. on CKPT(x,VT) from P_j:   find y :: y.id = x.id
31.                               y.value := x.value
32.                               y.state := C
33.                               y.VT := VT
34.                           send ACK(x.id) to P_j

35. on ckpt_i(x) :        if x.dirty=true
36.                         for each y :: y.dirty=true
37.                           atomically update checkpoint replicas of y with
                              sending CKPT(y,VT_i) to all P_j∈CCS(y) and
                              collecting ACK(y.id) from all correct of them
38.                         y.dirty := false
```

Fig. 1. Actions of the CAUSp protocol for process P_i

4.1 Proof of the Safety Property

First, let us introduce another vector clock comparison operation:

- $VT_i \leqslant VT$ – which is true iff $\forall k$: $VT_i[k] \leqslant VT[k]$

```
1  on recover_i(x)v :  if x.state = C
2                          local_invalidate_i(x.VT)
3                          VT_i := update(VT_i,x.VT)
4                      x.state := WR
5                      x.dirty := false
6                      x.owner := P_i
```

Fig. 2. Recovery procedure performed by the new owner of x

Lemma 1. *For any P_i, vector clock VT_i is monotonically increasing.*

Proof. There are only 2 possible modification operations performed by the protocol on the vector clock:
1) $inc(VT_i)$– incrementing the i-th position $VT_i[i]$ of the vector clock;
2) $VT_i := update(VT_i,VT)$ – which performes $VT_i[k] := max(VT_i[k],VT[k])$ for each k-th position of the vector clock. Therefore, the vector clock is never decremented and for any two subsequent vector values VT_i^1 and VT_i^2; $VT_i^1 \leqslant VT_i^2$ is true. □

Lemma 2. *For any operations $w_i(x)v$, $w_j(x)u$, such that $w_i(x)v \rightarrow_i w_i(x)u$ holds, and $r_j(x)v$ $(i \neq j)$, there is $VT^1 \leqslant VT^2$, where VT^1 is the value of $x.VT$ when operation $r_j(x)v$ ends and VT^2 is the value of VT_i when the second write operation $w_i(x)u$ ends.*

Proof. First, note that value v of x has been made visible to P_j only if, after $w_i(x)v$ (i=*x. owner*), P_i had received R_REQ sent by P_j on $r_j(x)$. The first action performed by P_i on reception of that R_REQ was $inc(VT_i)$. Let VT^u be the incremented value of VT_i. Then, update message $UPD(x,VT^u)$ was sent from P_i to P_j, making $x.VT$ equal to VT^u. Now, from Lemma 1, $VT^u \leqslant VT^2$ must be true, so we get $x.VT \leqslant VT^2$. □

Lemma 3. *For any operations $w_i(x)v$ and $r_j(x)v$, $i \neq j$, there is $VT^1 < VT^2$, where VT^1 is the value of VT_i at the moment when operation $w_i(x)v$ ends and VT^2 is the timestamp value received in $UPD(x,VT^2)$ message during operation $r_j(x)v$.*

Proof. Again, note that value v of x has been made visible to P_j if, after $w_i(x)v$ time-stamped with value VT^1, P_i had received R_REQ sent by P_j on $r_j(x)$. So, P_i has performed $inc(VT_i)$ on reception of that R_REQ. Then, $UPD(x,VT^2)$ message was sent with the incremented value of VT_i, which is VT^2. Thus we have $VT^1 < VT^2$. □

Lemma 4. *For any operations o_i^1 and o_j^2, such that $i \neq j$ and $o_i^1 \rightarrow o_j^2$ holds, there is $VT^1 < VT^2$, where VT^1 is the value of VT_i when operation o_i^1 ends and VT^2 is the value of VT_j when operation o_j^2 ends.*

Proof. The case $o_i^1 = w_i(x)v$, $o_j^2 = r_j(x)v$ is subject of Lemma 3, thus it has been proven already. In case of $o_i^1 = w_i(x)v$ such that $w_i(x)v \rightarrow r_j(x)v \rightarrow_j o_j^2$, let VT^r be the value of $x.VT$ when operation $r_j(x)v$ ends. From Lemma 3 $VT^1 < VT^r$ and from Lemma 1 $VT^r \leqslant VT^2$. Thus, $VT^1 < VT^2$ is true.

For a general case: $o_i^1 \rightarrow w_k(x)v \rightarrow r_j(x)v \rightarrow_j o_j^2$, we apply the following induction:

step 1) if $o_i^1 \rightarrow_i w_i(x)v \rightarrow r_j(x)v \rightarrow_j o_j^2$, let VT^w be the value of VT_i when operation $w_i(x)v$ ends and VT^r be the value of $x.VT$ when operation $r_j(x)v$ ends. From Lemma 1 we have both $VT^1 \leqslant VT^w$ and $VT^r \leqslant VT^2$. On the other hand, from Lemma 3, $VT^w < VT^r$ is true. Thus, $VT^1 < VT^2$ is also true.

step 2) if $o_i^1 \rightarrow w_k(x)v \rightarrow r_j(x)v \rightarrow_j o_j^2$, where $k \neq i$, we can follow the induction recursively assuming $o^2 = w_k(x)v$, and apply step 1 or step 2 for $o_i^1 \rightarrow o^2$. □

In the following analysis we shall consider a value returned by a read access performed by the protocol, as the causal consistency violation may result only from reading an inconsistent value (there is no notion of "causally inconsistent writing").

Definition 3. *Read operation r(x)v performed by the CAUSp protocol is* ***legal*** *iff value v has not been overwritten by any causally preceding write operation, i.e.:*

$$\nexists_{w(x)u \in HW} \quad u \neq v \ \wedge \ w(x)v \rightarrow w(x)u \ \wedge \ w(x)u \rightarrow r(x)v$$

The value that can be returned by a legal read is consistent in terms of Definition 2. A value which is not consistent will hereafter be called *outdated.*

Now let us note that as long as P_i does not interact with other processes, all local access operations are performed on consistent values. This is because the local order preserves causal dependency. Value v of x held by P_i may actually become outdated only if P_i receives an UPD(y,VT") message from another process with some value $y.value=a$ that causally depends on $w(x)u$ which, in turn, overwrites v with u, i.e.: $w(x)v \rightarrow w(x)u \wedge w(x)u \rightarrow w(y)a \wedge r_i(y)a \rightarrow_i r_i(x)$. However, this chain of dependency will be reflected in value VT" of the vector clock received with this UPD(y, VT") message. This vector clock value is used in local invalidation which succeeds the reception of the UPD message.

Lemma 5. *All outdated values of locally available replicas are invalidated during the local invalidation operation.*

Proof. By contradiction, let us assume that P_i holds a replica of x with outdated value v timestamped $x.VT$, thus $r_i(x)v$ is not legal, i.e.:

$$\exists_{w(x)u \in HW} \quad u \neq v \ \wedge \ w(x)v \rightarrow w(x)u \ \wedge \ w(x)u \rightarrow r_i(x)v$$

The following relations are possible between $w(x)v$ and $w(x)u$:

1) both operations $w(x)v$, $w(x)u$ have been issued by the same process, say P_j, i.e.: $w_j(x)v \rightarrow_j w_j(x)u$. The causal dependency $w_j(x)u \rightarrow r_i(x)$ is only due to some operation $r_i(y)a$ performed by P_i before $r_i(x)$. Let us assign value VT^1 of VT_j to $w_j(x)u$, VT^2 to $w_j(y)a$, VT" to UPD(x VT") message sent by P_j after $w_j(x)v$, and, finally, VT" to UPD(y,VT") message sent after $w_j(y)a$. In this case $x.VT=VT'$ holds for replica of x at P_i and the following predicates are fulfilled: $VT' \leqslant VT^1$ from Lemma 1, $VT^1 \leqslant VT^2$ from Lemma 1 and $VT^2 < VT$" from Lemma 3. Provided the above, $x.VT < VT$" is true, and it makes x invalidated during $local_in\text{-}validation_i(VT$") performed on reception of the second update message UPD(y,VT"). The value returned then by subsequent $r_i(x)$ cannot be v anymore.

2) operations $w(x)v$, $w(x)u$ have been issued by distinct processes, i.e.: $w_k(x)v \rightarrow w_j(x)u$. Let us assign value VT^0 of VT_k to operation $w_k(x)v$ and then values VT^1, VT^2,

VT' , VT'' to $w_j(x)u$, $w_j(y)a$, UPD(x, VT') and UPD(y, VT''), respectively, as in point 1 of this proof. In this case, still $x.VT=VT'$ holds for replica of x at P_i, $VT^1 \leqslant VT^2$ and $VT^2<VT''$ also hold as above and, moreover, $VT'<VT^1$ is true from Lemma 4. Therefore, $x.VT<VT''$ is also true in this case, and x will be invalidated during *local_invalidation*$_i$(VT'') on reception of UPD(y, VT''). Again, the value returned by subsequent $r_i(x)$cannot be v anymore. □

Theorem 1. (safety) *Every read access to object x performed by any process is legal.*

Proof. Let us consider $r_i(x)v$ issued to the replica of x in one of the following states:

1) WR – this case is trivial as this state designates the master replica of x which is never outdated since only the owner can modify x;

2) INV – also trivial as no access is ever possible to a replica in the INV state;

3) RO – no replica in that state can contain an outdated value v because any modification causally following the write of v can be made visible to P_i only after reception of some UPD message which triggers local invalidation. From Lemma 5 all outdated values have been invalidated then;

4) C – similar to INV, except that no update is necessary; only the local invalidation is processed on the first read access issued to a C replica and the replica state is switched into ROC before value v is read;

5) ROC – the local invalidation was performed before the replica state has been switched to ROC, so all outdated values were invalidated (from Lemma 5). □

5 Conclusions

The coherence-recovery protocol proposed in this paper uses a novel burst checkpointing approach. Checkpointing is performed independently for each process, thus does not incur high interprocess synchronization overhead. The protocol enables fast recovery on the assumption of the reliability of the directory services, i.e. directory information for each object must be correctly provided to the requested processes at any moment in the majority partition. Fortunately, this assumption can be fulfilled with directory checkpointing and replication for each object, similarly to application data objects.

References

1. Ahamad M., Hutto P.W. and John R.: Implementing and Programming Causal Distributed Shared Memory. 11th Int.Conf. on Distributed Computing (1991) 274-281
2. Brzeziński J. and Szychowiak M.: An Extended Coherence Protocol for Recoverable DSM Systems with Causal Consistency. Int. Conf. on Parallel and Distributed Processing Techniques and Applications PDPTA'03, Las Vegas (2003)
3. Kongmunvattana A., Tanchatchawal S. and Tzeng N.-F.: Coherence-Based Coordinated Checkpointing for Software Distributed Shared Memory Systems. 20th Conf. on Distributed Computing Systems (2000) 556-563

2D and 3D Representations of Solution Spaces for CO Problems

Eugeniusz Nowicki and Czesław Smutnicki

Wrocław University of Technology,
Institute of Engineering Cybernetics,
Janiszewskiego 11-17, 50-372 Wrocław, Poland
enowicki@ict.pwr.wroc.pl, czeslaw.smutnicki@pwr.wroc.pl

Abstract. This paper presents various graphical techniques developed by us to support research of some very hard combinatorial optimization (CO) problems. Thanks to increased efficiency of algorithms and high power of modern PC, we are able to make visualization of the solution space and search trajectories, which in order, allows us to analyze and design modern approximate approaches quickly running towards good solution. We provide unique graphic representations of solution spaces and search trajectories of modern approximate algorithms dedicated for the job-shop problem, the one of the hardest problem in CO.

1 Introduction

Among variety of combinatorial optimization (CO) problems, the *scheduling theory* provides many hard cases, among which the *job shop scheduling problem* is known as a particularly hard one. This multidimensional, discrete CO problem follows from Operations Research (OR) practice, has relatively simple formulation, excellent industrial applications, finite but huge number of solutions and unfortunately is strongly NP-hard. That's why it is also considered as a permanent indicator of practical efficiency of advanced scheduling algorithms, manifesting indirectly the power of OR theory and algorithmic skillfulness. During last fifty years, a lot of various methods have been designed to solve the problem in a quick time, see reviews in [1,8].

Modern approximate approaches, as an example tabu search (TS), simulated annealing (SA) and genetic search (GS), dominate in the job-shop research carried out in the last decade. Although made works improve significantly algorithms efficacy (measured by the accuracy as opposing to the running time) observed phenomena of surprisingly good accuracy and capricious convergence depending on instances remain unexplained to the end. One supposes that the answer lies in the chaotic structure of the solution space, characteristic for CO problems. In particular, amazing TS algorithm, very fast and of excellent accuracy below 4%, designed by us few years ago for the job-shop problem [4], inspires us to explore these phenomena using advanced graphical tools.

We used for visualizations selected public benchmarks with the number of operations varying from 225 to 2,000, [7]. Research was supported by Grant T11A 01624 of the State Committee for Scientific Research.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 483–490, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Problem, Models, Solutions

There are a set of jobs $N = \{1,\ldots,n\}$, a set of machines $M = \{1,\ldots,m\}$ and a set of operations $O = \{1,\ldots,o\}$. Set O is decomposed into subsets corresponding to the jobs. Job j consists of a sequence of o_j operations indexed by $(l_{j-1}+1,\ldots,l_{j-1}+o_j)$ which should be processed in that order, where $l_j = \sum_{i=1}^{j} o_i$, $j = 1,\ldots,n$ $(l_0 = 0)$, and $\sum_{i=1}^{n} o_i = o$. Operation i must be processed on machine $\mu_i \in M$ during the time $p_i > 0$, $i \in O$. Each machine can process at most one operation at a time. A feasible schedule is defined by start times $S_i \geq 0$, $i \in O$, such that the above constraints are satisfied. We look for a feasible schedule that minimizes the makespan $\max_{i\in O}(S_i + p_i)$.

The choice of the model (MILP, disjunctive graphs) determines the form of the solution and thus character, structure, dimensionality and cardinality of the space. We operate on the following permutation-and-graph model, [4], which has the smallest cardinality with minimal fraction of infeasibility.

Set O can be decomposed into subsets $M_k = \{i \in O : \mu_i = k\}$, $k \in M$. The sequence of operations from M_k is defined by $\pi_k = (\pi_k(1),\ldots,\pi_k(|M_k|))$, permutation on M_k, $k \in M$. Let Π_k be the set of all permutations on M_k. The processing order of operations is defined by m-tuple $\pi = (\pi_1,\ldots,\pi_m)$, where $\pi \in \Pi = \Pi_1 \times \Pi_2 \times \ldots \times \Pi_m$. It is linked with schedule S_i, $i \in O$ by the digraph $G(\pi) = (O, R \cup E(\pi))$ with a set of nodes O and a set of arcs $R \cup E(\pi)$, where $R = \bigcup_{j=1}^{n} \bigcup_{i=1}^{o_j-1} \{(l_{j-1}+i, l_{j-1}+i+1)\}$ and $E(\pi) = \bigcup_{k=1}^{m} \bigcup_{i=1}^{|\pi_k|-1} \{(\pi_k(i), \pi_k(i+1))\}$. Node $i \in O$ in $G(\pi)$ has weight p_i, arcs have weight zero. Processing order π is feasible if $G(\pi)$ does not contain a cycle. Then S_i equals the length of the longest path going to the vertex i (but without p_i) in $G(\pi)$, $i \in O$. Makespan $C_{\max}(\pi)$ for feasible π equals the length of the longest path (critical path) in $G(\pi)$. Thus, Π is the solution space, where a processing order $\pi \in \Pi$ is its element.

3 Solution Space

Space Π is discrete, has dimension o, is finite, but its cardinality is of astronomical size. For the small historical instance FT10 $(n = 10, m = 10, o = nm = 100)$, we have $|\Pi| = \prod_{k=1}^{m}(|M_k|!) = (n!)^m \approx 4 \cdot 10^{65}$, however only (!) approximately $4 \cdot 10^{48}$ solutions are feasible, see [5]. The projection of feasible solutions of FT10 on 2D plane printed in high resolution 2400 dpi covers area of $4 \cdot 10^{32}$ km^2, whereas the surface of Jupiter has only $6.4 \cdot 10^{10}$ km^2. "Typical" searching procedure is able to check at most 10^9 solutions, which corresponds to only 0.1 m^2 of the area and can be shown as thread of 0.01 mm width and 10 km length.

The space Π can be perceived as hyper-graph, where nodes represent solutions and each arc links two solutions, so that the former can be obtained from the later by using a *move*. For the given solution π, the subset of solutions (nodes) linked by single arc with π is called the *hyper-neighborhood* of π, and can be generated by various types of moves. The move changes sequence of operations processed on a common machine as follows: (A) A-move swaps a pair $v = (a, b)$ of two adjacent operations a, b, (S) S-move swaps a pair $v = (a, b)$ of

two any operations a, b, (I) I-move for a pair $v = (a, b)$ of operations a, b, cuts out operation a and then inserts it immediately after operation b (if a precedes b), or immediately before operation b (otherwise). Let π_v denote the processing order obtained after the move v. The set of all moves of type Z performed from πwill be denoted by $V^Z(\pi)$,whereas the obtained hyper-neighborhood by $\mathcal{N}^Z(\pi) = \{\pi_v : v \in V^Z(\pi)\}$, $Z \in \{A, S, I\}$. The sequence of successive solutions $\pi^0, \pi^1, \ldots, \pi^r$, where $\pi^{i+1} \in \mathcal{N}^Z(\pi^i)$, $i = 0, 1, \ldots, r-1$ constitutes the *trajectory* (path in the hyper-graph). Hyper-neighborhoods of type A, S, I have strong connectivity property, i.e. for each pair of solutions $\alpha, \beta \in \Pi$ exists trajectory so that $\alpha = \pi^0, \pi^1, \ldots, \pi^r = \beta$ and r is finite.

Because of hyper-neighborhood size, its selected subset has been used in local search and called simply *neighborhood.* There are a few attributes for neighborhoods generated by moves: size, feasibility, usefulness and connectivity, [5], which influence on trajectories going through the space. The most popular neighborhoods are the following two based on A-moves. The first is generated by move set $V^{A1}(\pi) \subseteq V^A(\pi)$,where $v = (a, b) \in V^{A1}(\pi)$ if only a and b belongs to a critical path in $G(\pi)$, [3]. All these moves are feasible (leads from feasible to feasible solution), but some of them are useless (leads to essentially worse solutions). Neighborhood has only weak connectivity property (β is not any but one of optimal solutions). The second is generated by the move set $V^{A2}(\pi) \subseteq V^{A1}(\pi)$, obtained from $V^{A1}(\pi)$ by elimination of all useless moves, with the help of block property; all moves are feasible, [4]. Although $\mathcal{N}^{A2}(\pi)$ does not have the connectivity property, in experiments it behaves better than other neighborhoods.

4 Distances in the Space

We define the distance $D^Z(\pi, \sigma)$ between processing orders $\pi, \sigma \in \Pi$ as the minimal number of moves $Z \in \{A, S, I\}$ necessary to go from π to σ (i.e. the shortest path in hyper-graph). Clearly, we have $D^Z(\pi, \sigma) = 1$ for $\sigma \in \mathcal{N}^Z(\pi)$. It has been shown, [5], that because move is being performed inside single machine, then permutation π_k can be mapped onto permutation on set $N^k = \{1, 2, \ldots, |M_k|\}$, $k \in M$. Thus, the measure $D^Z(\pi, \sigma) = \sum_{k \in M} D_k^Z(\pi_k, \sigma_k)$ and its properties can be derived from properties for measure $D_k^Z(\pi_k, \sigma_k)$ between permutations π_k and σ_k on the set N^k. Follows results from [2], there have been collected in [5] basic properties of measures for all types of moves (A,N,I): recipes for calculation, mean value, variance and computational complexity. Here only notice that the measure $D_k^A(\pi_k, \sigma_k)$ equals the number of inversions in permutation $\pi_k^{-1}\sigma_k$, where π_k^{-1} is the inverse permutation π_k, $k \in M$. Hereinafter we analyze the use of A-moves only, thus upper index A will be skipped in proper notions.

For the sake of huge size of the space, its character can be analyzed statistically by random sampling. In Fig. 1 (left) one can find, for typical benchmark ta04 ($n = 15$, $m = 15$, $o = 225$), the distribution of $DIST(\pi) = 100 * D(\pi, \pi^{REF})/D_{\max}$, where $D_{max} = \max_{\alpha,\beta} D(\alpha, \beta) = mn(n-1)/2$, taking into account the random sample of ALL solutions and the random sample of FEASible solutions, where π^{REF} is a fixed reference solution (optimal solution).

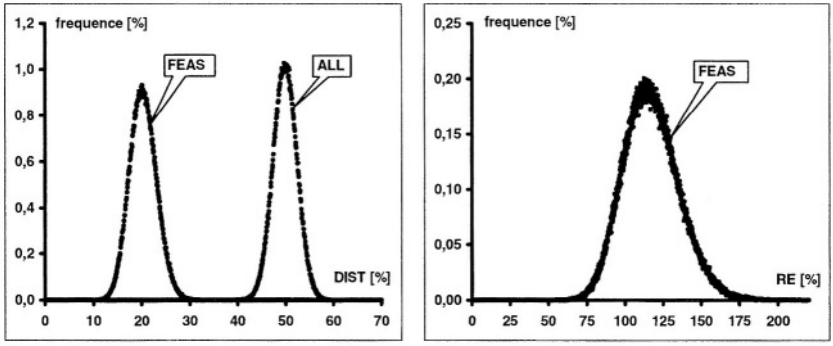


Fig. 1. Instance ta04: Distribution of $DIST(\pi)$ *(left)* and $RE(\pi)$ *(right)* for 500,000 random solutions; ALL - for any solutions, FEAS - for feasible solutions

Curve ALL fits ideally to the theoretical distribution of $D(\alpha, \beta)$, with the mean $0.5 * D_{\max}$. From curve FEAS one can read the mean value of *DIST* distribution for feasible solutions equal approximately $0.25 * D_{\max}$; this holds for almost all benchmarks ta–. Both distributions do not depend on the choice of π^{REF}.

The distribution of $RE(\pi) = 100 * (C_{\max}(\pi) - C_{\max}(\pi^{REF})/C_{\max}(\pi^{REF})$, which is the relative error, for the same sample of FEASible solutions is shown in Fig. 1 (right). The distribution is close to gaussian. RANdomly found solution has RE≈120% (standard deviation 15%), whereas we know that the best constructive algorithm called INSA offers RE≈20% and the best improvement algorithm TS has RE≈4%. Note also, that finding feasible solution with RE below 50% is statistically infinitesimal. Results remain valid for all ta– benchmarks.

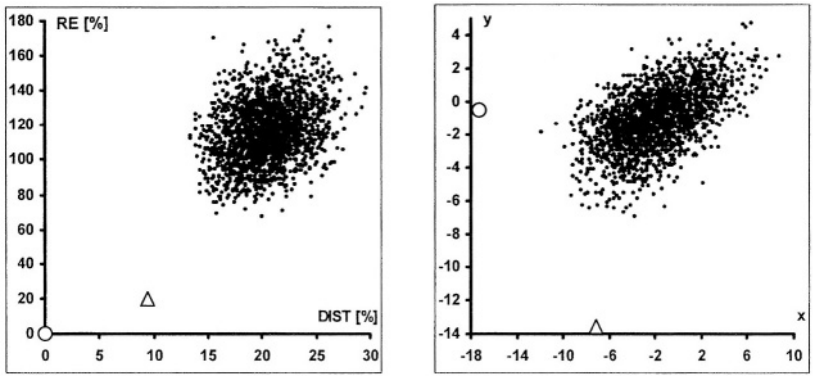


Fig. 2. Instance ta04: 2,000 random feasible solutions in coordinates DIST/RE *(left)* and X/Y *(right);* △ marks starting solution found by INSA, ○ - optimal solution

The main goal of the space analysis is the identification of the *landscape* and its typical element called the *big valley,* which plays role of an *attractor* in the space. Its presence means that distances between good locally optimal solutions and the global one are significantly positively correlated with goal function values of these solutions. It also means that majority of very good local optima concentrates in a relatively small area of the solution space.

In Fig. 2 one can find plot of random feasible solutions showing the correlation 0.3 between DIST and RE (statistically significant on the level 0.1%), both calculated with respect to the optimal solution π^{REF}. This fact confirms the existence of "big valley", not only for good locally optimal solutions (as it showed in [5]) but also for *all* feasible solutions.

5 2D Transformations

Our aim is to transform solutions $\pi^j \in \Pi$, $j = 1, \ldots, t$ into points on 2D plane "rationally" preserving the distance from Π. Problem of transformation continuous n-dimensional space R^n into space R^2 is known for years, see the review in [6]. However, some troubles appear in the application to CO problems. First, the source space is discrete. Second, metric in Π is not Euclidian, takes only discrete values, and can influence on the final result of the transformation. Third, and the most important in our opinion, the calculation process should be cheap since in practice we need to transform 1,000 ... 10,000 solutions from the space with dimension 200...2,000. That's why we focus chiefly on *reference points* approaches. Among a few proposals, we describe below only one, fast and of satisfactory *mapping accuracy* (accuracy is measured by the correlation between distances in space Π and Euclidian distances of 2D images). Let σ^i, $i = 1, \ldots, r$ be a sequence of reference solutions. Each π is transformed into point (x, y) on 2D plane according to the following formula

$$\begin{bmatrix} x \\ y \end{bmatrix} = \sum_{i=1}^{r} D(\pi, \sigma^i) \begin{bmatrix} \sin(\frac{(i-1)\pi}{r}) \\ \cos(\frac{(i-1)\pi}{r}) \end{bmatrix} \quad (1)$$

We performed tests on most of 80 ta- benchmarks, with $r = 3$ reference solutions: $\sigma^1 = \pi^{REF}$, σ^2 – solution generated by algorithm INSA used as the starting point for our method TS in [4], and σ^3 – solution generated by SPT rule, one of the most commonly applied in constructive algorithms. In the series of graphical runs for a few known algorithms, we found similar results for all benchmarks. Then, we discuss here only one, typical. In Fig. 3 one can find search trajectories (in various axes) performed by algorithms TS, SA, RANdom; all work on neighborhood $\mathcal{N}^{A2}$. Compare chaotic track of algorithm RANdom trapped in a local sub-area of the solution space with directional track of TS; in these conditions the behavior of SA is similar to RANdom. (From unpublished here results we found that RANdom and SA behaves even worse if we replace $\mathcal{N}^{A2}$ by greater neighborhood e.g. $\mathcal{N}^{A1}$ or $\mathcal{N}^{A}$.) This static figure does not reflect the evolution of the trajectory in time, therefore in Fig. 4 we have provided speed of convergence of tested methods in iterations (in time) – TS is the best.

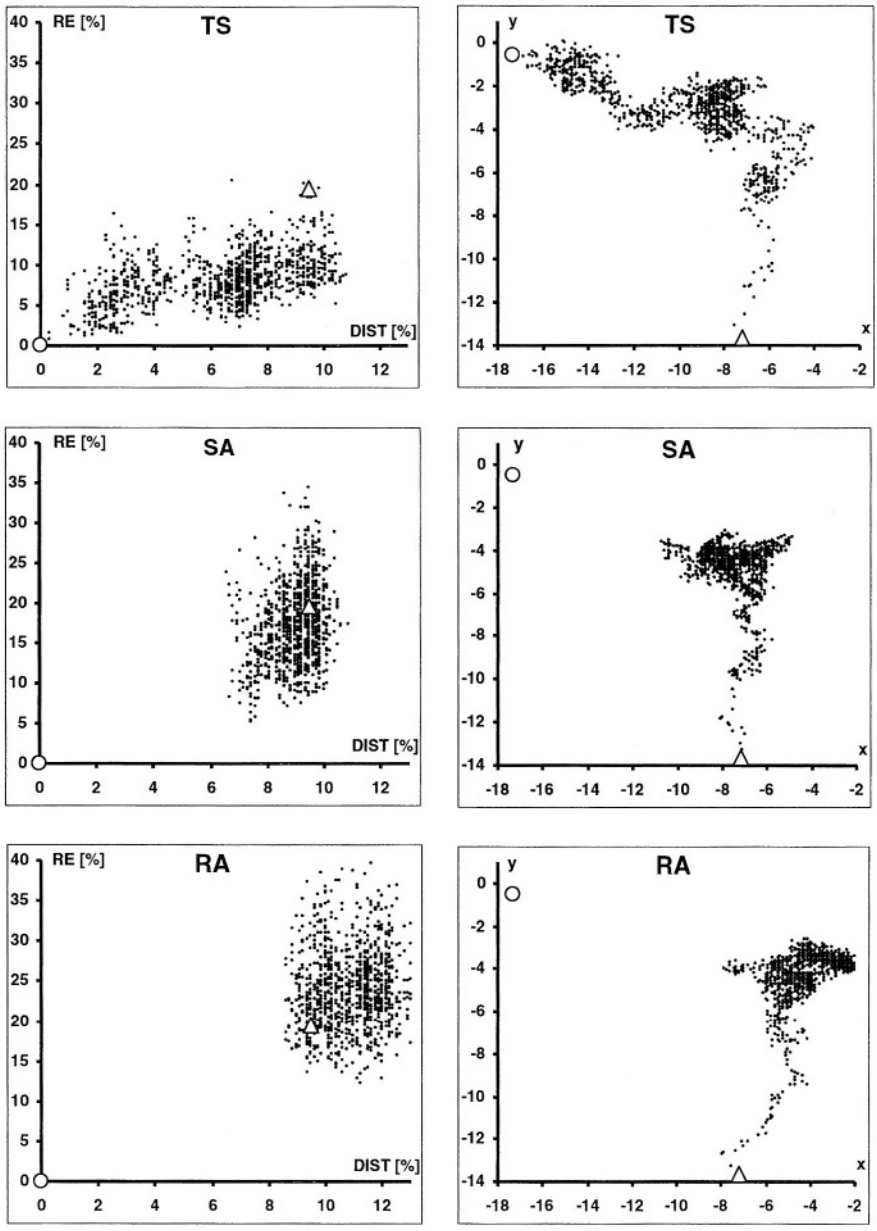


Fig. 3. Instance ta04: 10,000 iterations of algorithms TS, SA, RANdom (ploted every 10 iterations) in coordinates DIST/RE *(left)* and X/Y (*right*); △ marks starting solution found by INSA, ○ - optimal solution

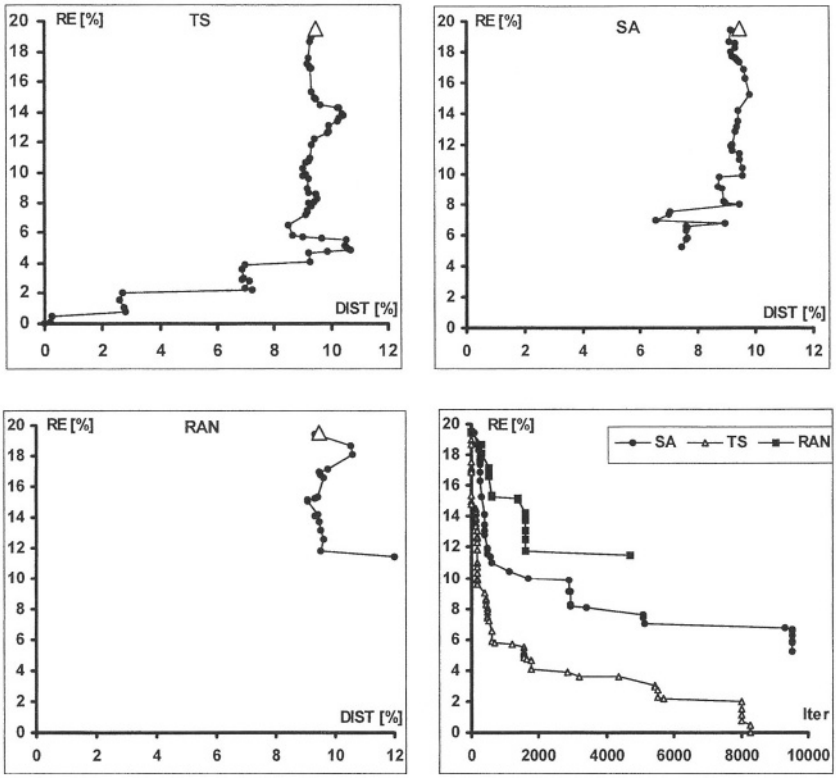


Fig. 4. Speed of convergence for the instance and algorithms from Fig. 3; there are marked only solutions reducing the makespan

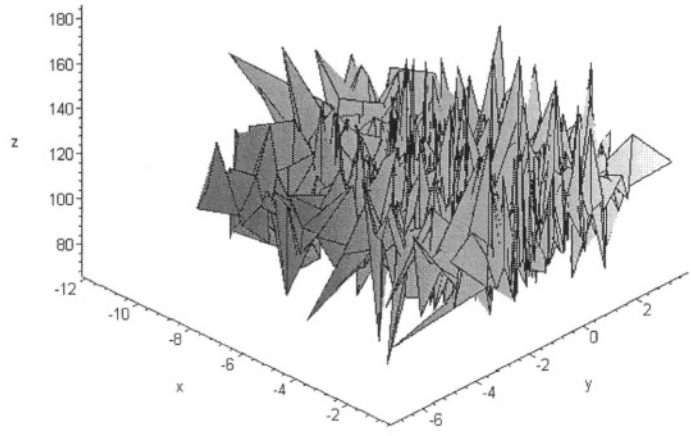


Fig. 5. Instance ta04: Optimization surface spanned on solutions from Fig. 2 right

Clearly, the choice of r and the location of reference solution influences on the image – this subject is currently being studied. Primal tests suggest that reference solutions should be spread widely, whereas the increase of r does not influence on mapping accuracy significantly.

6 Optimization Surfaces

Using certain transformation of solutions into 2D points, we can try to draw in 3D the optimization surface, spanned on points (x_i, y_i, z_i), $i = 1, \ldots, t$, where (x_i, y_i) is the image of solution π^i on 2D, $z_i = RE(\pi^i)$, and $\pi^1, \ldots, \pi^t$ is the sample of feasible solutions from Π. An example of such surface spanned on a part of points from Fig. 2 is shown in Fig. 5. It destroys our classical imagination about optimization process – the surface is sharp without any regularity, like the brush, fur, high mountain or stone forest. Local extremes are distributed densely, chaotically, and act as deception points. The primal two natural associations are more suitable since through the increase of the sample cardinality one obtains higher density of tops and wells on almost the same area. The figure indicates fundamental troubles in the research of CO problems, formulated as the rhetorical question "How to find the proper path on this surface being a small ant with limited visibility?" This pessimistic attitude is overcomed by existing algorithms – TS from [4] finds such path intentionally and consciously, tending to RE=0% for ta04 (RE$\approx$4% for all ta– instances). Other methods, like SA or GS search space through the sampling points on the surface - properties of such methods are completely predicable. That's why frequently SA and GS need support by complete/partial neighborhood search procedures in order to reach satisfactory quality.

References

1. Błażewicz, J., Domschke, W., Pesh, E.: The job shop scheduling problem. Conventional and new solution techniques. European Journal of Operational Research **93** (1996) 1–33
2. Knuth, D.E.: The art of computer programming. Addison Wesley Longman (1997)
3. Van Laarhoven, P., Aarts, E., Lenstra, J. K.: Job-shop scheduling by simulated annealing. Operations Research **40** (1992) 113–125
4. Nowicki, E., Smutnicki C.: A fast tabu search algorithm for the job-shop problem. Management Science **42** (1996) 797–813
5. Nowicki, E., Smutnicki, C.: Some new ideas in TS for job-shop scheduling. Technical Report 50/2001. In print in: Rego C., Alidaee B. (eds.): Adaptive Memory and Evolution: Tabu Search and Scatter Search. Kluwer Academic Publisher (2004)
6. Siedlecki, W., Siedlecka, K., Sklansky, J.: An overview of mapping techniques for exloratory pattern analysis. Pattern Recognition **25** (1988) 411–429
7. Home page of E. Taillard: http://www.eivd.ch/ina/collaborateurs/etd/default.htm
8. Vaessens, R., Aarts, E., Lenstra, J.K.: Job shop scheduling by local search. INFORMS Journal of Computing **8** (1996) 303–317

Effective Detector Set Generation and Evolution for Artificial Immune System

Cholmin Kim[1], Wonil Kim[2], and Manpyo Hong[1]

[1] Internet Immune System Laboratory
Graduate School of Information and Communication
Ajou University
{ily, mphong}@ajou.ac.kr
[2] College of Electronics and Information Engineering
Sejong University
wikim@sejong.ac.kr

Abstract. Human bodies defend themselves against harmful invaders by the Natural Immune System (NIS). The NIS possesses a huge set of detectors against many different invaders. Based on the essential principles of the NIS, the idea of Artificial Immune System (AIS) can be used for protecting computer resources. The AIS can detect the nonself-activity using a self-adopted detector set. The detector set is generated randomly and evolved by a Genetic Algorithm (GA). However, the random generation and the blind GA adoption decrease the performance of detector set and generate unnecessary redundancy. To overcome this problem, we propose a more efficient detector generation and evolution scheme. Using the proposed idea, the AIS can produce more effective detector set with time advantages.

1 Introduction

Detecting intrusions and anomalies in a complex computer system have the same nature as the 'Achilles and Turtle' problem. Many researchers try to find a good protection mechanism for a secure computer system, but the new intrusion skill or tool that can circumnavigate the protection mechanism is also developed immediately [1]. Nowadays, many researchers have realized that a set of specific methods to counteract a specific type of intrusion is no longer the permanent solution. They found that a complete understanding of the behavior of the system to be protected is the best way to treat the intrusion [2]. One kind of technique in this paradigm is an Artificial Immune System (AIS) [3]. The main idea of AIS comes from the human body. The human body has a fundamental protection mechanism. The Natural Immune System (NIS) is the essential part of the human self-protection mechanism [2]. The NIS can distinguish self from nonself. Self comprises all components that are necessary for normal functioning of the human body whereas nonself is foreign material that could harm normal functioning of the human body.

The basic concept of AIS is the same as NIS. As in NIS, any kind of program that tries to accesses the computer system can be categorized into self and nonself [3]. In this case, nonself is a program which has harmful function to computer system. Thus

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 491–498, 2004.
© Springer-Verlag Berlin Heidelberg 2004

a nonself detector is required to protect a system from illegal programs. These detectors are generated randomly and evolved adaptively. The GA (Genetic Algorithm) is the most frequently used technique to evolve these detectors. As evolution stages process, more accurate detector set for nonselves can be generated.

The current AIS, however, has some problems in randomness and blind GA adoption. By the random generation of detector, we can obtain larger detector set but it may not guarantee that every element of this set will be certainly being useful. In fact, large portions of generated detectors are not used. The blind crossover and mutation also have the similar problems. The current AIS randomly chooses crossover point and mutation value. Thus some meaningless gene relocation and mutation can occur. Due to these problems, the performance of detectors decreases and hence relatively large storages are needed. To overcome this, we propose improved techniques for initial detector generation and evolution.

There is a difference between the previous AIS and the proposed idea. The previous AIS protect the self-file by searching some nonself-file signatures. Thus, the detector is a binary string for the file. However, the detection target of the proposed system is a system call sequence. The proposed idea will generate meaningful system call sequences and evolve to accurate one.

The rest of this paper is organized as follows: In Section 2, we introduce the brief overview of NIS, AIS and GA method used in immune systems. In Section 3, we propose the idea to improve the performance of detector set and discuss experimental results. Section 4 concludes.

2 Ideas from Immune Systems and Natural Evolution

2.1 The Natural Immune System (NIS)

The NIS consists of a multitude of cells and molecules that interact in a variety of ways to detect and eliminate infectious agents (pathogens) [2, 3]. Detection and elimination of pathogens is the consequence of trillions of cells interacting through simple and localized rules. Consequently the NIS is very robust to failure of individual components and attacks on the NIS itself [2].

The problem of detecting pathogens is often described as distinguishing self from nonself as in previous section. However, many pathogens are not harmful, and an immune response to eliminate them may damage the body. In these cases it would be healthier not to respond. Therefore it would be more accurate to say that the problem faced by the NIS is distinguishing between harmful nonself and everything else [3].

2.2 The Artificial Immune System

Using the main idea of NIS and D'haeseleer, Forrest, and Helman suggested AIS [6]. They concentrated their interest on the Negative Selection (NS). NS is a kind of NIS behaviors. It searches for the protein that binds with detectors. Detectors are generated for a protein that does not belong to self and will detect all nonself objects. The matching algorithm detects sequence of contiguous bits containing particular position and length.

The detection range of AIS can be described as follows. The string space is a set of every string which can be occurred in the system. For host-based AIS, it can be every combination of system call sequences in certain length. The string space partitioned into self strings *S* and nonself strings *N*. Nonself strings indicate some intrusion or abuse. The detector set *R* is a subset of all possible candidate detectors and is the actual set the AIS can generate. Nonself strings can be further divided into detectable nonself strings (*N'*) and holes *(H, H = N - N'*). Limitation of memory size leads to detect not every nonself strings but some portions of them. Detectable nonself strings can be obtained depending on the choice of *R*.

2.3 Protein and Computer System Call

Detector set of NIS is collection of protein strings, whereas the set of proposed AIS is system call sequences [3, 7]. There are at least 10^{16} nonself proteins for human body. The NIS will store a portion of those. On the other hands, current Linux system has about 200 system calls. Thus, if we consider the detector as size 3 string the number of detector combinations will be 8 million (200 * 200 * 200). The details of detector size selection will be described in section 3. Detector of size 7 is also frequently used. In this case the number will reach to $1.28 * 10^{16}$. If we just compare the numbers, we can find that the negative selection problem for the NIS and AIS has the similar complexity. However, because of the parallelism, the NIS is more efficient. Each lymphocyte of NIS moves totally independently and carries an individual detector. In the contrast, since the number of processors of a computer is restricted, the detector set of AIS is activated in some serial manners and needs more operations than single lymphocyte of NIS. Consequently, we need more optimized generation and evolution mechanism for the AIS detector set.

2.4 The Genetic Algorithm

The NIS appears to be able to recognize at least 10^{16} nonselves. The human genome contains about 10^5 genes. From an information-processing perspective, recognizing an almost limitless number of foreign cells and molecules, and distinguishing these from self molecules are formidable tasks. Thus, the individual cells that comprise the immune system encode and operate the control mechanism in parallel. The genetic mechanism involving the combinatorial association of a number of gene segments underlies the construction of the receptors, so that an NIS has the genetic capability of expressing over 10^{16} different receptors [4, 5]. To implement this in the AIS, the AIS model uses GA which is an idealized computational model of Darwinian evolution based on the principles of genetic variation and natural selection.

In order to use the GA to evolve a set of bit strings that have the property of target, the GA must concern the characteristics of the target gene. In our AIS, the target is a system call sequences.

GA is typically used to evolve a population in which each member specifies one candidate solution and the individual solution competes with each other. The majority of GA analysis has focused on these optimization applications. From a pattern-recognition perspective, the nonself recognition problem can be viewed as a string-

matching problem in which the task is to discover sets of common substrings that collectively the population of nonself strings. The learning task is thus to evolve a set of nonselves that meets the coverage requirement.

3 The Proposed Detector Set Generation and Implementation

We briefly introduce the AIS and the detectors used in AIS. Since the random generation does not guarantee the usefulness of detector, some unnecessary detectors could be generated. The detector generation system, however, could not decide the usefulness of the newly generated strings. By this reason, the previous AIS maintain garbage detectors and need relatively large storage space. In addition to these problems, the number of detectors determines the time that is needed to compare a detector with a target. Consequently, excessive number of detectors also decreases the performance of matching operation.

On the other hand, the GA in AIS has problems too. In GA, evolution is done by the crossover and the mutation. In the crossover shuffle process, input genes are divided into several sub-genes and these sub-genes change their positions each other. Since the detector strings in our system are consisting of system call sequences, they have some contextual meanings. Thus, swapping of sub genes can make brand new object gene and produce a useless detector. With the same reason the mutation has a problem too. A blind mutation will generate garbage too.

In order to overcome these problems we propose three techniques, filtered initial generation of detector, filtered detector crossover and biased detector mutation.

It is observed that 3 and 7 is the best size for identifying the system call sequences [7]. We use size 3 sequences in first heuristic and 7 in the others.

3.1 Class Based Detector Generation

To improve the initial detector generation phase, we uses some strategic selection of detector. Since the detector strings consist of system call sequences, they have contextual meanings and contain similar string with both normal and abnormal system call sequences. Thus we compare newly generated one with exist strings and if the new one is not similar with any of exist strings, we discard it.

The method we propose for initial filtering constructs the detector set based on a context. In this method we first generate a random system call number, and decide a detector class which the first system call number is belonging. Figure 1 describes the mechanism

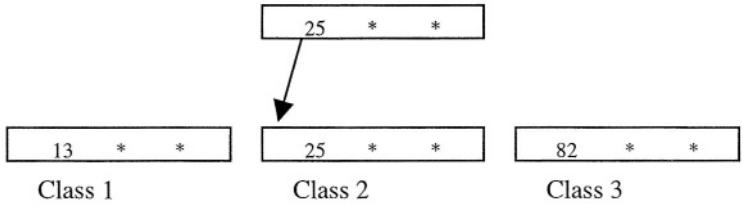


Fig. 1. Initial detector generation and classification

Since we know the belonging class of current detector after the generation of the first system call number, we can remove the system call numbers which are quite different from other members in the class during the remaining system call generation phase. Table 1 shows the system call sequences of length 3 during an intrusion. The sequences are sorted and duplications are eliminated. It is extracted from sunsendmailcp (SSCP) intrusion log files [8].

SSCP script uses a special command line option to cause sendmail to append an email message to a file. By using this script on a file such as /.rhosts, a local user may obtain root access.

Table 1. System call sequences of length 3 during an intrusion

Class 1	2	5	23	**Class 9**	19	128	3
	2	11	17		19	128	23
	2	50	27	**Class 10**	23	4	50
Class 2	3	3	5		23	45	3
	3	3	112		23	50	27
	3	4	14	**Class 11**	27	2	50
	3	19	32		27	4	27
	3	93	88		27	50	3
	3	112	19	**Class 12**	45	3	112
Class 3	4	3	3	**Class 13**	50	3	93
	4	14	112		50	27	4
	4	18	50		50	27	50
Class 4	5	4	18		50	50	27
	5	23	4	**Class 14**	88	167	17
	5	23	50	**Class 15**	112	3	19
Class 5	9	3	5		112	50	50
	9	9	3	**Class 16**	128	3	3
Class 6	11	17	4		128	3	4
Class 7	14	3	112		128	3	112
	14	112	19		128	4	18
Class 8	17	5	4		128	23	45
	17	112	19		128	112	3
Class 9	19	112	4	**Class 17**	167	17	5

Since these sequences are not found in normal system call sequences of sendmail daemon, they can be considered as nonself. As we can see in this case there are only 17 classes if we classify the sets by the first system call number. Logically there can be about 200 classes, but almost every class except these 17 classes are useless.

This classification can be adapted to more than one layer and will provide more hierarchical class architecture of detector.

3.2 Filtered Detector Crossover

In order to avoid generating unnecessary detectors in crossover, we also use context information of system call sequence. When we perform crossover operation, we divide the genes into substrings. By the consequence of shuffle substrings, the resulting string will have a quite different system call combination. If we compare the adjacent system calls with normal or abnormal system call sequences, we can

conclude whether the generated detector is a brand new one or not. Figure 2 shows the selected position for comparison.

In the system call sequence size 7, we can divide the sequence into two sub-gene of size 3 and 4. Thus the comparison locations are the 4^{th} and the 5^{th}. We will show the experimental result in section 4.

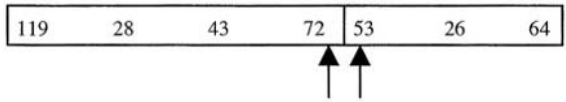


Fig. 2. The comparison location of system call

3.3 Biased Detector Mutation

There are about 200 different system call services. These system calls have different usage frequency. However, previous works performed uniform mutation. Due to the different frequencies, the random number selection must be biased. If it is not biased, the generated number can not represent the real system call number distribution.

We have used biased roulette for the random number. First we gather the frequencies of the system call number from intrusion log files. Figure 3 shows the frequencies of the system calls. As seen in the graph, intrusion log files have different distribution to the normal one.

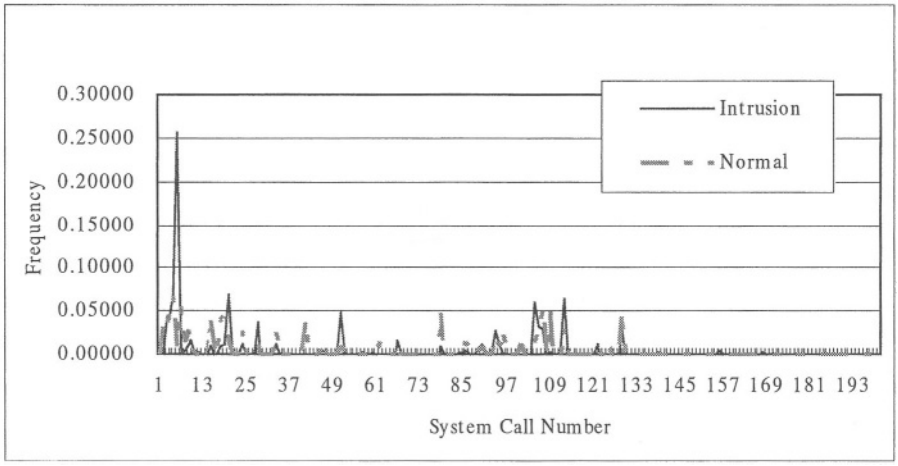


Fig. 3. System calls usage frequency distributions

3.4 Experiment and Evaluation

We experimented our ideas using the logged data of Immune System Laboratory in New Mexico University by the following steps [8]. First, we generated our detector set by the proposed ideas using single attack log file. Then we matched the generated set with another attack log file and normal log files to evaluate true positive (TP) and

false positive (FP) cases. We performed the same experiment with the blind detector generation scheme. Table 2 shows the result of the two experiments.

Table 2. The results of experiment 1 and 2.

	File size	# of matched detector in ex1 (Proposed Method)	# of matched detector in ex1 (Blind Method)	# of matched detector in ex2 (Proposed Method)	# of matched detector in ex2 (Blind Method)
Intrusion log files	1.1K	36/40	35/374	107/503	108/3013
	2.0K(1)	36/40	36/374	130/503	131/3013
	2.0K(2)	37/40	36/374	131/503	131/3013
Normal log files	1K	0/40	15/374	2/503	16/3013
	312K	1/40	30/374	4/503	35/3013
	5818K	1/40	55/374	7/503	56/3013

The first experiment shows the exactness of class based detector generation. We have generated detector set of size 40 with 20 classes. Then we search the generated detector set from three different sunsendmailcp intrusion log files that are not used in detector set generation. The log file sizes are 1.1K, 2.0K and another 2.0K bytes respectively. The intrusion log file format is a simple ASCII text file. At least 36 of 40 detectors were found in every intrusion log files. Three detectors are not founded in any intrusion log files. One detector is found in only the second log file of size 2.0K. We also generated the detector set in random manner without any class. It generates 374 detectors. But it found the same number of strings with the proposed method.

We matched the same detector sets with the three normal log files. The file sizes are 1.0K, 312.0K and 5818.0K bytes respectively. Normal log files have relatively large sizes than attack log files because it can be gathered in long period. Only 1 of our detectors is found in 2 log files. However, many detectors in blind scheme has found in each log files. It means our scheme decrees FP rate.

The second experiment show how effectively the proposed idea generates an accurate detector set by filtering the unnecessary detector in crossover and mutation steps. It was done for the system call sequences of size 7. We divided the original system call sequence of size 7 into two sub-genes. Sizes of sub-genes are 4 and 3. Thus the crossover point is offset 4 of the sequence. We have generated 503 of evolved detectors with our scheme. These detectors are evolved by following steps. First, we crossover genes then leave detectors that have frequently appeared system calls in position 4 and 5. Then mutate the detector with biased policy. The mutation rate is 1%. 3013 detectors are generated without our method. These detectors are crossover without filtering and mutated in random manner. As seen in Table 2, False Positive (FP) rates of the proposed method in both experiments are very low comparing with the blind method.

4 Conclusion

GA is one of popular methods used to detect nonselves in Artificial Immune System (AIS). However, unnecessary detectors are still generated due to the random initialization and blind GA adoption. We proposed a novel approach to solve these problems in this paper. The first one is classification of detector, the second one is crossover point comparison, and the third one is biased mutation. We used a system call sequence data provided by Immune System Laboratory in New Mexico University. By the experiments, we showed that the detector set that is generated and evolved by the proposed idea maintain high detection rate in spite of using small space. Our idea can be easily extended to other intrusion detection areas.

References

1. A. Acharya and M. Raje.: MAPbox: Using Parametererized Behavior Classes to Confine Aplications. In Computer Science Technical Report TRCS99-15, Department of Computer Science University of California, Santa Barbara (1999).
2. S. Forrest and S. Hofmeyr.: Engineering an immune system. Graft Vol. 4:5 (2001) 5-9.
3. S. Hofmeyr and S. Forrest.: Architecture for an Artificial Immune System. Evolutionary Computation 7(1), Morgan-Kaufmann, San Francisco, CA (2000) 1289-1296.
4. S. Forrest.: Genetic Algorithms. Computing Surveys Vol. 28:1 (1996) 77-80.
5. S. Forrest, B. Javornik, R.E. Smith, and A.S. Perelson.: Using Genetic Algorithms to Explore Pattern Recognition in the Immune System. Evolutionary Computation, Vol. 1, No. 3(1993)191-211.
6. P. D'haeseleer, S. Forrest, and P. Helman.: A Distributed Approach to Anomaly Detection (1997).
7. R. Sekar, M. Bender, D. Dhurjati and P. Bollineni.: A Fast Automaton-Based Method for Detecting Anomalous Program Behaviors, Proceedings of 2001 IEEE Symposium on Security and Privacy (2000) 144-155.
8. Computer Immune System Data Sets, Computer Immune System Laboratory, New Mexico University, http://www.cs.unm.edu/~immsec/data-sets.htm.
9. R. Biischkes, M. Borning and D. Kesdogan.: Transaction-based Anomaly Detection. In Proceedings of the Workshop on Intrusion Detection and Network Monitoring (1999).
10. J. Fritzinger and M. Mueller.: Java security. Technical report, Sun Microsystems, Inc (1996).
11. M. Opera, S. Forrest.: How the immune system generates diversity: Pathogen space coverage with random and evolved antibody libraries. University of New Mexico (1998).
12. S. A. Hofmeyr, S. Forrest.: Immunity by Design. University of New Mexico (1999).

Artificial Immune System against Viral Attack

Hyungjoon Lee[1], Wonil Kim[2*], and Manpyo Hong[1]

[1] Digital Vaccine Lab, G,raduated School of Information and Communication
Ajou University, Suwon, Republic of Korea
{prime, mphong}@ajou.ac.kr
[2] College of Electronics and Information Engineering,
Sejong University, Seoul, Republic of Korea
wikim@sejong.ac.kr

Abstract. Since the first computer virus has been found, scanning detection has been used as a primarily method in virus detection systems. As computer viruses and worms become more complex and sophisticated, the scanning detection method is no longer able to detect various forms of viruses and worms effectively. Many anti-virus researchers proposed various detection methods including artificial immune system to cope with these characteristics of viruses and worms. This paper discusses some principle of artificial immune system and proposes artificial immune based virus detection system that can detect unknown viruses.

1 Introduction

Since the computer virus first appeared in 1981, it has been evolved as computer environments such as operating system, network, and application program have changed. The frequent evolution of computer virus has forced anti-virus researchers continuous development of new detection method. However most anti-virus systems are still based on scanning detection using signature, since other mechanisms have false positive or detection speed problem in real situations. In this paper, we propose a new virus detection approach that employs the artificial immune system. Artificial immune system is based on human immune system. The human immune system recognizes, memorizes, and responses to virus adaptively. Anomaly detection and adaptability are important properties of artificial immune system. The proposed artificial immune based system applies this property and detects unknown computer viruses.

In the next section, Artificial Immune system will be discussed in detail. In Section 3, the proposed artificial immune based virus detection system will be described. Simulation results and the direction of future research will be discussed in Section 4 and Section 5 respectively.

* Author for correspondence +82-2-3408-3795

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 499–506, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Artificial Immune System

Immunologists have traditionally described the human immune system as the problem of distinguishing "self" from dangerous "non-self" and eliminating dangerous non-self. Similarly, the computer security can be viewed as the problem of distinguishing benign program from malicious program such as computer virus or worm. Spafford described a computer virus from a biological point of view by analyzing several characteristic of virus as in artificial life [1]. Since then, many anti-virus researchers have researched on artificial immune system to protect a system from the intrusion including viruses and worms.

Artificial immune system should have anomaly detection capability to defend unknown intrusion. Adaptability is also a necessary property of artificial immune system to learn unknown intrusion and to response learned intrusions quickly. Other properties such as distributability, autonomy, diversity and disposability are required for the flexibility and stability of artificial immune system [14].

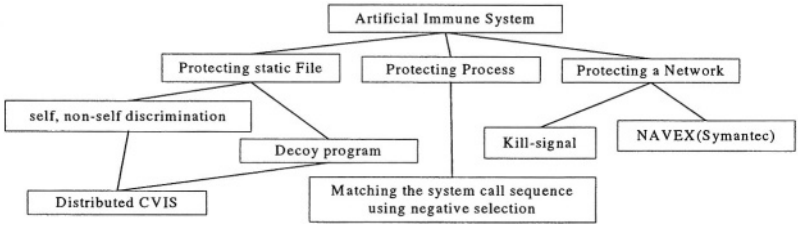


Fig. 1. Taxonomy of artificial immune system

Artificial immune system can be divided into three categories according to which computer component will be identified as a protected cell. Figure 1 shows this classification.

First, protecting static file category assumes a file including program code as a cell. Forrest proposed a method for change detection using negative selection algorithm [6]. Forrest's artificial immune system randomly generates check code called detector, and eliminates ones that can recognize self (benign program). This process is called negative selection algorithm. All programs or data matched with detectors are indicated as non-self (malicious code). Kephart proposed an artificial immune model that consists of monitoring, scanning and biological immunology [11]. In his model, known viruses are detected by scanner based on weak signature, and unknown viruses are collected and analyzed using decoy program. The decoy program is a dummy program that is not changeable. If decoy program is modified, it is virus infection. Distributed Computer Virus Immune System (CVIS) combined negative selection and decoy program. It uses decoy program to detect the unknown virus and employs negative selection to identify detected unknown virus [16].

Second, protecting process category assumes a process as a cell. Matching the system call sequence is proposed by Forrest [10]. The mechanism of this system is the same as self-nonself discrimination using negative selection. It is different from the first category that this mechanism matches detector with system call sequences, not with program codes.

Third, protecting a network category assumes a host as a cell. Kill-signal mechanism is proposed by Kephart, and a immune system called NAVEX that is proposed by Symantec [11,17]. Both immune systems propagate signatures of detected viruses to protect network connected hosts, as does biological immune system. In biological immune system, when an immune cell detects a virus, the cell spread chemical signal in order to duplicate the same immune cells.

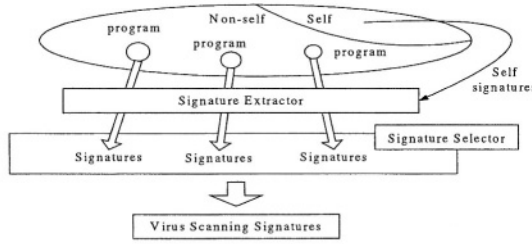


Fig. 2. Structure of proposed virus detection system

3 Proposed Virus Detection System

Negative selection algorithm and decoy programs are important methods in artificial immune system. Negative selection algorithm is inspired by genetic random-process. But this mechanism has a weakness that come from the difference between human body and computer system. State of human body is stable, whereas that of computer system is not. Negative selection algorithm is suitable in stable system. Idea of decoy program comes from the distributability and diversity of immune system. In the case of decoy program, there is no guarantee that a virus will attack a decoy program. The proposed artificial immune based Virus Detection System (VDS) compromises this weakness, and is suitable for dynamically changing environment such as computer system.

VDS is a signature learning system to detect unknown viruses. VDS classify incoming or changed (patched) programs into legitimate programs and viruses. The process of VDS is consists of the following three steps. In the first step, VDS assumes that all existing programs are legitimate. In the next step, all incoming and changed programs are classified into suspicious programs. Finally, VDS selects virus programs from these suspicious programs using detection method based on virus behavior. Hereafter, we refer to legitimate program as self and suspicious program as non-self.

VDS consists of signature extractor and signature selector as in Figure 2. Signature extractor produces signatures of non-self. The main operation of signature extractor is selecting bit strings that are not matched with any self code at the same position. Therefore, signature extractor produces signature of non-self that never matched with any self. Produced non-self signatures are collected and analyzed by signature selector.

Signature selector compares the similarity of non-self signature with each others. Since viruses tend to infect other programs, if the same non-self signatures are appeared frequently, signature selector classifies those into signatures of viruses, and these signatures are used for virus scanner. Notice that these non-self signatures never

match with any self program. In the case of less frequent signatures, it is assumed as signature of self. We classify the program that contains any one of these self signatures as self. This process provides the adaptable capacity to the proposed VDS. The detailed process of each steps are discussed in the following 3 sections.

3.1 Signature Representation

The VDS generates signatures to recognize self and non-self program. The size of signature extraction region, from which we analyze for signature extraction, is always constant. Since most infected file executes virus code at first, most entry pointer of infected program indicates virus code. Therefore, we decide that starting point of signature extraction region is entry point of program. Signature consists of several pairs of offset and bit string. Number of pairs, offset and bit string are selected by signature extractor.

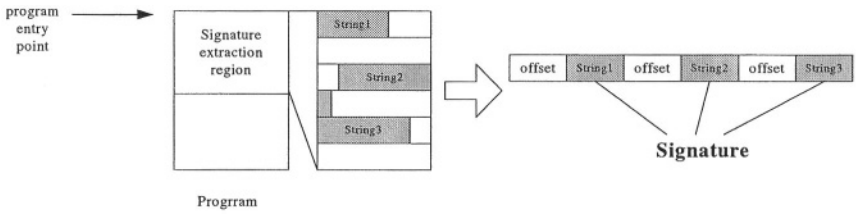


Fig. 3. Signature representation

3.2 Signature Extractor

Signature extractor produces non-self program signature that never matches with any self-program signature. This process consists of the following steps.

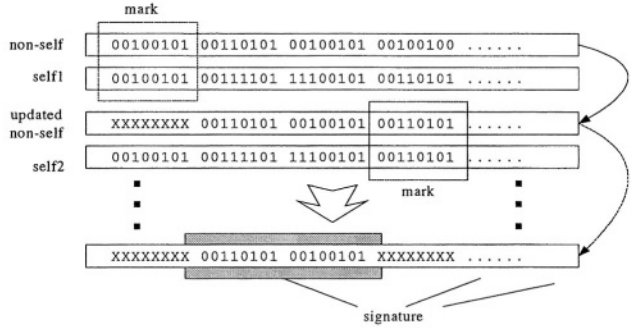


Fig. 4. Generating signature of non-self program

- Step 1: Divide a signature extraction region into several same sized comparison units.
- Step 2: Compare a signature extraction region of non-self program with each one of all self-programs.

- Step 3: If two comparison units on same position are same, mark the comparison unit in signature extraction region of non-self program. Continue this process on each signature extraction region of all self-programs.

In Figure 4, comparison unit is 1 byte. Non-self is compared with self1 and self2. When two comparisons are finished, first unit and fourth unit are marked. After all the comparisons with other self-programs are done, the signature of non-self consists of remained unmatched units. Since all of the comparison units that are the same with self's are marked as useless, the generated signature never matches with any self-programs.

3.3 Signature Selector

After we obtain signatures of non-self programs, we need to classify non-self program into virus or normal program. If virus infects other programs, the signature extractor generates signatures from same virus code because infected program is changed by one virus. Therefore, checking frequency of occurrence of the same signature is the same as checking the spread of the same virus code.

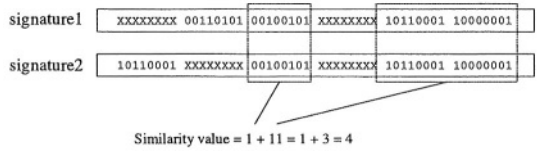


Fig. 5. Similarity between two signatures

Signature selector calculates the similarity values of non-self signatures, as shown in Figure 5. Comparison factors are bit sequence and offset of comparison unit in signature extraction region. If two factors of comparison units are equal, similarity function adds one to similarity value. When consecutive comparison units are equal, similarity value increases. For example, if two compare units are equal and adjacent, similarity value is 3 (11_2). In other words, when n consecutive comparison units are equal, similarity value is 2^n-1.

Note that signatures of the same programs are more similar than signatures of different programs. In other words, similarity values between signatures of the same program codes are higher than the others. Therefore, VDS can distinguish signatures of the same program codes from signatures of other distinct program codes. The threshold values for classifying signatures of the same and different programs are determined by analyzing similarity values of the entire non-self programs. Determining the threshold values for classifying is discussed in Section 4.

4 Simulation

VDS extracts signatures of non-self programs based on self-programs. Signature size and similarity value are important factors in VDS. As the number of self-programs that has the same comparison units at the same position is increased, the size of a

signature of the non-self program is decreased. In the worst case, VDS cannot generate the signature of particular program, because every comparison unit is marked as useless. The similarity value is used for classifying virus detector from suspicious signatures. In this section, we will discuss several simulation results from various signature extraction region size and comparison unit. The simulation is processed on several parameters as in Table. 1.

Table 1. Simulation parameters

Parameters	Variables
The number of self-programs	1385 execution files
The number of non-self programs	160 execution files (3 virus infected files)
Signature extraction region size	500byte, 1Kbyte, 5Kbyte, 10Kbyte
Comparison unit size	1byte, 2byte, 3byte

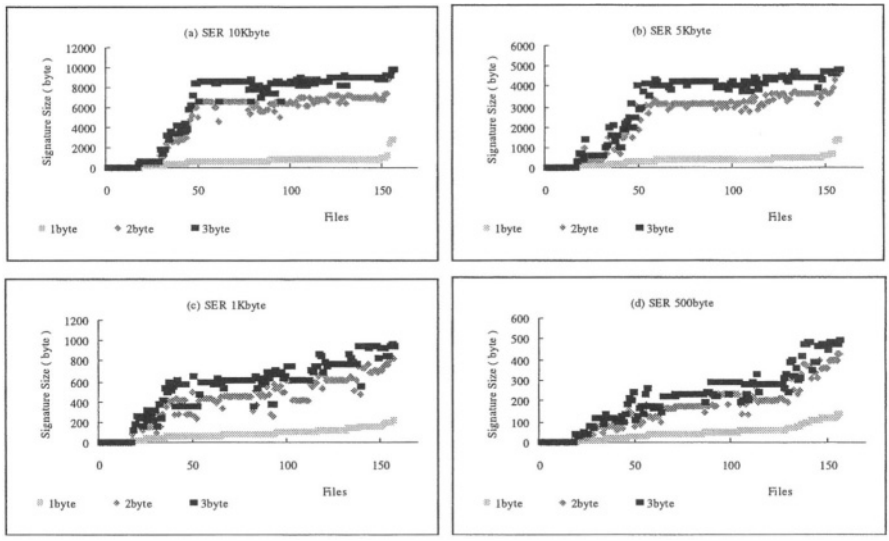


Fig. 6. Simulating using various sizes of SER: Graphs show signature size of different Signature Extraction Region (SER) 10Kbyte; 5Kbyte, 1 Kbyte, and 500byte. Each graph has three parameters of comparison unit 1byte, 2byte, and 3byte.

The graphs in Figure 6 show the relation between signature size and two parameters; signature extraction region and comparison unit. The more signature extraction region and comparison unit increases, the more signature size increases. Larger signature includes richer information about related non-self program. But, when signature is used for scanning viruses, small signature is effective. Signatures that are larger than 1KB are not feasible for a virus-scanning signature. Moreover, the percentage of files that has zero signature is independent with size of signature extraction region and comparison unit. The number of files with zero signature size is almost 14 (8.75%) in all cases. We chose two effective sizes of signature extraction region; 1Kbyte and 500 byte. We simulated comparison for signature selection about the extracted signatures of non-self programs on these two parameters.

Similarity values between each signature of non-self programs are shown in Figure 7. When signature extraction region is 1Kbyte and comparison unit is 1byte, 88% of signatures of non-self programs have similarity value zero. When signature extraction region is 500byte and comparison unit is 1byte, signatures with zero similarity value are 92% in entire signatures of non-self programs. Since the percentage of actual virus infected file is 1.875%, ideal percentage of signatures that similarity value is zero should be 98.125%. Then, we can classify non-self signatures into virus signatures, whose similarity value is greater than zero. But, the percentage of non-zero signatures is 6% even though the extraction region size is 1Kbyte and comparison unit size is 3 byte. We need to determine threshold value to classify non-self programs into normal programs and viruses.

When SER is fixed, increasing the size of comparison unit results in rapid increase of its similarity value, since consecutive extracted signature is larger. When larger comparison unit is used, the gap of similarity value is larger. When signature extraction region is 1Kbyte and comparison unit is 3byte, we can find easily threshold of similarity value 1.E+08 to classify signatures of non-self programs into virus, because ideal percentage of signatures of normal programs is 98.125%. When threshold of similarity value is 1.E+08, three signatures are selected by signature selector. Virus scanner using these signatures can detect virus-infected files.

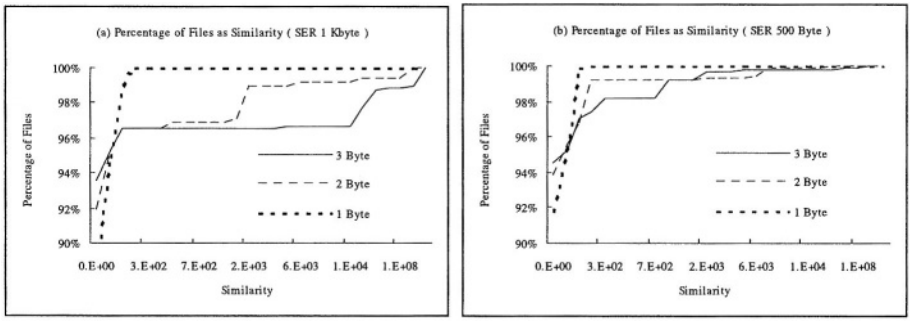


Fig. 7. Percentage of files as similarity: In the case of SER 1Kbyte and 500byte, Percentage of files is showed. Each graph has three parameters of comparison unit 1byte, 2byte, and 3byte.

5 Conclusion

Artificial immune system is based on distinguish “self” and “non-self” like biological immune system. But previous artificial immune system is not feasible for dynamic computer system, especially for negative selection algorithm. In this paper, we proposed the artificial immune based Virus Detection System (VDS) that is feasible for dynamic computer system. VDS is a signature learning system to detect unknown virus. VDS produces signatures of non-self from suspicious program, and classify them into self-program and virus. Since virus tends to infect other programs, if similar non-self signatures are appeared frequently, VDS classifies those into signature of viruses for virus scanning. Therefore, VDS detect adaptively unknown viruses

In the simulation of VDS with 1Kbyte of the Signature Extraction Region (SER) and 3byte of the comparison unit, 94% of extracted signatures were completely

different. In other words, their similarity values are zero. The remaining 6% signatures including virus signatures had distinguished similarity values. Especially, the 2% virus signatures had relatively high similarity values. Therefore, proposed VDS can classify suspicious non-self programs into normal programs and viral programs. Using threshold of similarity value, VDS can select virus signatures. Virus scanner using these signatures can detect virus-infected files correctly.

References

[1] Eugene H. Spafford, Computer Viruses as Artificial Life, in Artificial Life II, Addison-Wesley (1992).
[2] Richard Marko, Heuristics: Retrospective and Future, Virus Bulletin Conference (2002)
[3] Taras Malivanchuk, The WIN32 worms: Classification and Possibility of Heuristic detection, Virus Bulletin Conference (2002)
[4] Vesselin Bonchev, Macro and Script Virus Polymorphism, Virus Bulletin Conference, (2002)
[5] Gabor Szappanos, Are There Any Polymorphic Macro Viruses at All?, Virus Bulletin Conference, (2002)
[6] Stephanie Forrest, Self-Nonself Discrimination in a Computer, IEEE Symposium on Research in Security and Privacy (1994)
[7] J. P. Anderson, Computer Security threat monitoring and surveillance, Technical report, (1980)
[8] Dorothy E. Denning , An Intrusion-Detection Model, IEEE (1986)
[9] David Wagner and Drew Dean, Intrusion Detection via Static Analysis, IEEE (2001)
[10] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff, A Sense of Self for Unix Processes, IEEE (1996)
[11] Jeffrey O. Kephart, A Biologically Inspired Immune System for Computers, High Integrity Computing Laboratory IBM Thomas J. Watson. Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems (1994)
[12] Kim, J., Bentley, P., (1999), The Human Immune System and Network Intrusion Detection, submitted to EUFIT'99
[13] Digging For Worms, Fishing For Answers, (2002)
[14] Dipankar Dasgupta and Nii Attoh-Okine, Immunity-Based Systems : A survey, IEEE International Conference (1997)
[15] Anil Somayaji, Steven Hofmeyr and Stephanie Forrest, Principles of a Computer Immune System, (1997)
[16] Robert E. Marmelstein, David A. Van Veldhuizen and Gray B. Lamont, A Distributed Architecture for an Adaptive CVIS
[17] The Digital Immune System, Symantec technical report
[18] Stephanie Forrest, Anil Somayaji and David H. Ackley, Building Diverse Computer Systems, IEEE (1997)
[19] Dipankar Dasgupta and Stephanie Forrest, Novelty Detection in Time Series Data Using Ideas from Immunology, International Conference on Intelligent System (1999)
[20] Dipankar Dasgupta, Artificial Neural Networks and Artificial Immune System : Similarities and Differences, IEEE (1997)
[21] Okamoto, T. and Ishida, Y., Multiagent Approach Computer virus: An Immunity-Based System, AROB (1999)

Proposal of the Programming Rules for VHDL Designs

Jan Borgosz and Bogusław Cyganek

AGH – University of Science and Technology
30-059 Krakow, Poland
{borgosz, cyganek}@agh.edu.pl

Abstract. This paper presents novel developments of a programming methodology – in a form of the programming paradigms – for the group of hardware description languages (HDL). On the one hand, this group of languages is very similar to other high-level programming languages. From the other hand a description, which then must fit specific hardware arrays, needs to be coded in a special way with particular attention devoted to the code structure. In this paper it will be shown that the specific programming rules can make this process easier and more efficient. Also clarity of the code can be greatly improved, what is very important in the case of a team work. The complete set of the VHDL programming rules is presented, as well as examples of increased coding efficiency. This approach is much more general and stands in opposition to the other literature sources in which only selective programming advices are provided. Presented ideas were verified in many large telecommunication projects developed on the Xilinx' ISE Foundation platform.

1 Introduction

This paper is devoted to the programming rules for efficient programming in the VHDL language, which belongs to the group of the hardware description languages (HDL) for very high speed integrated circuits (VHSIC) [6]. Coding with clearly set rules is very important part of the properly set team oriented programmers' work. The main results of such approach to the problems are:

1) Shorter time needed for the different people to understand code;
2) Easier maintenance of the code;
3) Shorter time of the code development;
4) Decreased probability of the occurrence of the functional bugs;
5) Faster debugging;
6) An ability to use object oriented and template techniques.

Many effort was put to set programming rules for the high level languages like C, C++. There are many publications and researches about this subject, like [7][8][9][10]. The authors of this work, motivated by benefits listed above, have decided to find similar paradigms for VHDL. The main intention of this document is to show new set of the rules for all the people who are interested in improving coding

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 507–514, 2004.
© Springer-Verlag Berlin Heidelberg 2004

techniques. This document should be help for all advanced VHDL team members, as well.

2 Review of the Proposals of the VHDL Coding Rules

The programming process is not unique, therefore there are possible many approaches to do the same programming task – some of them are very specific, some provide more benefits than other. In this chapter we have focused on the proposals given by the other programmers and software vendors.

Xilinx is one from the most important software vendors of the modern VHDL tools' market. However, in Xilinx documentation only one appendix treats about this subject. In few words it is possible to say, that Xilinx recommends to keep as close as possible to hardware. Presented rules impose names coming from the FPGA structure. This approach is not very close to programming rules known for high level languages. As a result, projects written in this way can not be easily ported between platforms and scaled. Advices given by Xilinx may be explained by fact, that Xilinx takes care about successful implementation.

It is hard to find any document of these two vendors that would be entirely devoted to the VHDL programming conventions. Of course there are available online helps and tutorials, but after lecture the problem of the naming and programming rules still remains open.

Very interesting proposal of the naming convention may be found in the VFIR project [1]. Authors of this project have made special document with set of rules for VHDL programmers, with simplified ideas close to the ones presented in this paper. They have proposed simplified suffix system to distinguish between objects in VHDL language (strongly hardware dependent). In other work author points a need to use special rules and tries to propose them even for the clock signals edges [2]. There is also an another, and even more general, approach to the programming convention problem [5]. The problem of naming conventions is often omitted or simplified, even in the case of the VHDL handbooks like the VHDL CookBook [2].

3 New VHDL Programming Paradigms

The presented hereafter approach was created and tested in many successfully finalized VHDL projects. Finally, a well balanced solution was achieved that is as general as possible and as close to hardware implementation as needed.

In addition to the naming conventions, some additional rules have been imposed on projects, that brought the whole process closer to the object-oriented approach [4][9][10]. These rules are listed below:

1) Design is divided into building blocks;
2) Each building block is completely independent from the other blocks;
3) Parameters of each block are given with `generic` construction;
4) An interface to each block is constrained with global constants;

5) Blocks are connected in queue fashion, not in the star fashion;
6) Signals which are used in different blocks are passed from block to block;
7) Each block may be replaced or removed without any changes to the neighbor blocks;
8) All synchronous logic in blocks works with the some clock edge.

Usage of keyword `generic` allows to make code closer to the template idea. It is worth to say that a properly set VHDL design is object oriented in its nature.

3.1 General Naming Conventions

In VHDL designs we can distinguish the two most important types of objects:

1) Connections;
2) Building blocks.

Of course, these objects may be divided to subtypes due to its hardware nature. Connections may be signals between building blocks or external input / outputs. Building blocks may be low level elements of the FPGA structure or high level structures composed by the programmer.

General rule for the naming these two kinds of objects is that the instance name describes functionality as shortly as possible. When the direction of signal flow is important, its name should reflect this fact too. Created in this way part of the name will be called a *functionality name.*

So a name of each object in VHDL may be created due to the following concatenation rule:

$$FullName = PrefixName + FunctionalityName + SuffixName(optional)$$

Where *prefix name* comes from type of a VHDL object and *suffix name* is an optional information. Suffix may reflect additional information about associated type of the FPGA's building block hardware.

Authors propose two different rules of combining words into names. In the first rule, words in a name are separated by starting each new word with a capital letter. In the second rule, words are separated with the underscore "_" character. A choice of used convention should be done due to the software environment. For example, for Xilinx software the second convention will be more appropriate, due to the fact that Xilinx software always convert characters to lowercase.

Tables 1-3 contain proposal of the prefixes and suffixes for different objects which can be applied in Xilinx ISE environment (first rule).

Table 1. Set of the rules for design objects – building blocks, words separated with a capital letter

Design element	Rule	Example
Entity	Prefix "E"	EAdder
Architecture	Prefix "A"	ABehavioralAdder
Instantiation	Prefix "I"	IAdder
Package	Prefix "P"	PMyPackage

Table 2. Set of the rules for design objects – signals and miscellaneous, words separated with a capital letter

Design element	Rule	Example
Constant	Prefix "c"	cBitSize
Generic – scalar	Prefix "gs"	gsGenericValue
Generic – vector	Prefix "gv"	GvGenericValue (cBitSize downto 0);
Variable – scalar	Prefix "vs"	vsValue
Variable – vector	Prefix "vv"	vvValue (cBitSize downto 0);
Signal – scalar	Prefix "ss"	ssAlarm
Signal – vector	Prefix "sv"	svAlarm (cBitSize downto 0);
Input port – scalar	Prefix "ips"	ipsStrobe
Output port – scalar	Prefix "ops"	opsStrobe
Input port – vector	Prefix "ipv"	ipvData (cBitSize downto 0);
Output port – vector	Prefix "opv"	opvData (cBitSize downto 0);
Bidirectional port – vector	Prefix "bpv"	bpvData (cBitSize downto 0);
Bidirectional port – vector	Prefix "bpv"	bpvData (cBitSize downto 0);
Array	Prefix "a'	aMemory
Type	Prefix "t"	tDoubleArray
Subtype	Prefix "s" + Type Name	sCollectiontDoubleArray

Table 3. Examples of the rules for design objects – suffixes connected to hardware, words separated with a capital letter

Exemplary hardware element	Rule	Example
BUFGP	Sufix "BUFGP"	ipsClockBUFGP
IBUF	Sufix "IBUF"	ipsResetIBUF
FD	Suffix "FD"	ILatchFD
RAM16x1	Suffix "RAM16x1"	IMemoryRAM16x1

4 Adapting UML for the Illustration of the VHDL Design

It seems to be very interesting to adopt UML convention for the VHDL projects. As mentioned before, a nature of VHDL and hardware projects is close to an object oriented way of thinking. Thanks to this observations it was possible to use modified UML diagrams for illustration of VHDL projects. This idea is depicted in Fig. 1.

As can be seen, a VHDL entity with its name may be treated as a class name, ports may be resources and finally methods may be replaced with functionality. Additionally, a new part was added, called Generics. This new element of the block presents a possibility to parameterize an entity. This is an analogy to the templates in C++ [9]. Due to this assumptions a form of the UML cell for the class (Fig. 1a) may be replaced with a new form for the entity in VHDL (Fig. 1b).

Of course owning relation may be expressed as well. In the next chapter we present simple example of these ideas.

Fig. 1. a) Standard UML scheme used for the class notation. b) Adopted notation for the VHDL entity

5 Example of the Design with Proposed Programming Rules

As an example we present design with two independent PRBS (Pseudo Random Bit Sequence) generators. Each generator is built with the same parametrical code – a practical demonstration of the template technique. Fig. 2 depicts hardware structure of this design. As may be seen, inputs to the block are: `ipsClock`, `ipsReset`, `ipsPRBSSelect`, and an output constitutes `opsPRBSOutput`. Inputs `ipsClock`, `ipsReset` are common for both blocks: PRBS9 and PRBS11 generators. Outputs from generators `opsPRBSOutput` are multiplexed and finally output of the selected generator is an output from the whole design, also named `opsPRBSOutput`. It is important to see, that high clarity of the descriptions allows to analyze design quickly. Also information about kind of object is coded into its label. Detailed description of the PRBS generator will be presented in the next section.

5.1 Template Implementation of the PRBS Generator

The PRBS generator has architecture declared as follows:

```
entity EPRBS is
            Generic(
              gsFirstXORInput      : integer      := 9;
              gsSecondXORInput     : integer      := 5;
              gsActiveClockEdge    : std_logic    := '0'
                  );
   Port       (
              ipsClock   :  in std_logic;
              ipsReset   :  in std_logic;
              opsPRBSOut :  out std_logic
                   );
end EPRBS;
```

There are three parameters needed to define the PRBS generator structure. Two of them (`gsFirstXORInput`, `gsSecondXORInput`) come from definition of the PRBS polynomial and the third (`gsActiveClockEdge`) adopts clock edge of the module to rest of the system. A definition of the PRBS generator entity is presented below.

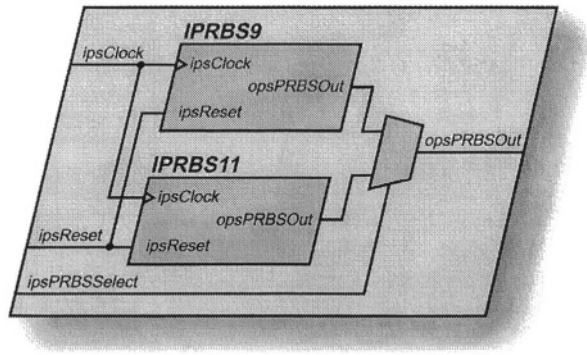


Fig. 2. Hardware structure of the presented example

Please note, that used programming rules immediately allows to identify type of an object. Moreover, any mistakes may be found and corrected faster. Also structure of the generator is more clear than with an ordinary naming convention.

```
architecture EPRBSBehavioral of EPRBS is
  signal svBuffer : std_logic_vector ((gsFirstXORInput) downto 0);

begin
  -- PRBS calculation
  process (ipsReset, ipsClock)
  begin
    if (ipsClock'event and ipsClock = gsActiveClockEdge) then
      svBuffer (0) <= svBuffer (gsFirstXORInput) XNOR svBuffer
                                                 (gsSecondXORInput);
    end if;
  end process;

  -- PRBS output generator
  process (ipsReset, ipsClock)
    begin
      if (ipsReset = '1') then
        opsPRBSOut <= '0';
      elsif (ipsClock'event and ipsClock = gsActiveClockEdge) then
        opsPRBSOut <= svBuffer (gsFirstXORInput);
      end if;
  end process;

  -- Buffer shifter
  process (ipsReset, ipsClock)
    begin
      if (ipsReset = '1') then
        svBuffer (gsFirstXORInput downto 1) <= (others => '0');
      elsif (ipsClock'event and ipsClock = gsActiveClockEdge) then
        svBuffer (gsFirstXORInput downto 1) <=
                          svBuffer ( (gsFirstXORInput -1) downto 0 );
      end if;
  end process;

end EPRBSBehavioral;
```

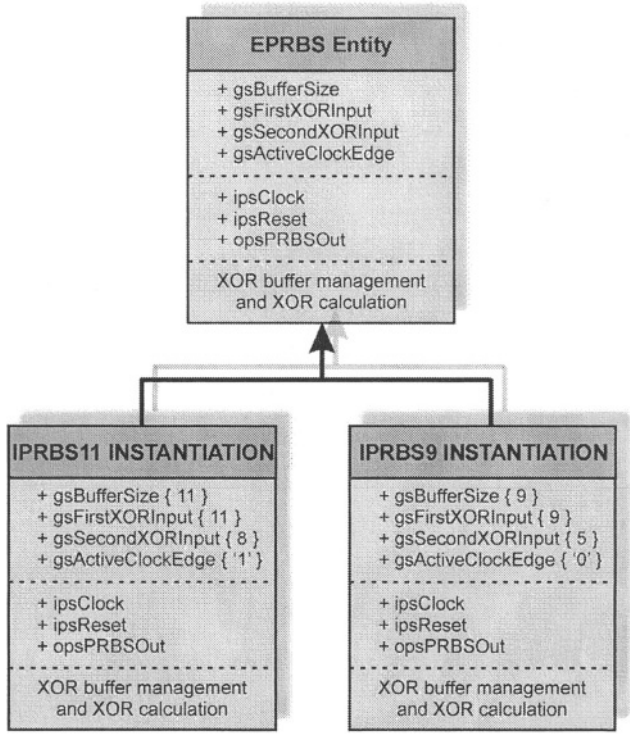


Fig. 3. Illustration of the template instantiation with the modified UML diagram

5.2 Instantiation of the PRBS Generator's Template

Instantiations of the created template are shown below:

```
-- PRBS 9
IPRBS9: ePRBS
Generic map (
  gsBufferSize          => 9,
  gsFirstXORInput       => 9,
  gsSecondXORInput      => 5,
  gsActiveClockEdge     => '0'
    )
PORT MAP(
  ipsClock => ipsClock,
  ipsReset => ipsReset,
  opsPRBSOut => ssPRBSOut1
);

-- PRBS 11
IPRBS11: ePRBS
Generic map (
  gsBufferSize          => 11,
  gsFirstXORInput       => 11,
  gsSecondXORInput      => 8,
  gsActiveClockEdge     => '1'
    )
```

```
            PORT MAP(
              ipsClock => ipsClock,
              ipsReset => ipsReset,
              opsPRBSOut => ssPRBSOut2
            );

opsPRBSOut <=  ssPRBSOut1 when ipsPRBSSelect = '0' else
               ssPRBSOut2;
```

Fig. 3 depicts relation between template and PRBS generators, instantiated to the design. Values of all parameters, basic functionality and input / output ports may be easily seen. The proposed programming rules allow to distinguish objects without any troubles.

6 Conclusions

This paper presents a proposal of the new programming rules for the VHDL designs. In opposition to other sources, a detailed description is presented. Also UML diagrams were adopted to make code more clear. Presented ideas were successfully tested with many projects, like STM-1,4,16 framers and mappers (clock speed 155 MHz with 16 bit wide data bus). Thanks to a well balanced and universal approach, the obtained code may be easily resized and reused. For example, an implementation of the error counter and synchronization tracking module for STM – 16 and STM – 4 took 20% of the time needed for STM – 1 (the first module).

References

1. Ameseder T., Pühringer T., Wieland G., Zeinzinger M.:VFIR Coding Conventions 1.0., WWW (2003)
2. Ashenden P.: VHDL Cookbook , White Paper available through WWW.
3. Cohen B.: Coding HDL for Reviewability, MAPLD 2002
4. Cyganek B.:“Programming Paradigms”, White Paper, Krakow (1999)
5. Gord A.: HDL Coding Rules and Guidelines, WWW (2003)
6. IEEE Standard, VHDL, 1987.
7. Meyer S.: Effective C++. Second Edition. Addison-Weseley (1998)
8. Meyer S.: More Effective C++. Addison-Weseley (1995)
9. Stroustrup B.: The C++ Programming Language. Third Edition. Addison-Weseley (1997)
10. Taligent: Taligent Guide to Designing Programs. Addison-Weseley (1995)
11. Xilinx: XST User Guide, Xilinx, USA (2003)

A Weight Adaptation Method for Fuzzy Cognitive Maps to a Process Control Problem

Elpiniki Papageorgiou and Peter Groumpos

Laboratory for Automation and Robotics
Department of Electrical and Computer Engineering
University of Patras, Rion 26500, GREECE
Tel. +30 2610 997293, Fax. +30 26120 997309
{epapageo,groumpos}@ee.upatras.gr

Abstract. Fuzzy Cognitive Maps constitute an attractive modeling approach that encompasses advantageous features. The most pronounces are the flexibility in system design, model and control, the comprehensive operation and the abstractive representation of complex systems. The main deficiencies of FCMs are the critical dependence on the expert's opinion and the potential convergence to undesired steady states. In order to overcome these deficiencies a possible solution is the utilization of learning methods to adapt the cause-effect relationships of the FCM model. This research work examines the formulation of a weight adaptation method suitable for FCMs based on nonlinear Hebbian-type learning algorithm. A process control problem is presented and its control is investigated using the adaptation technique.

Keywords: Learning algorithms, Nonlinear Hebbian learning, fuzzy logic, fuzzy cognitive maps, process control

1 Introduction

Conventional methods were used, successfully, to model and control systems, in the past years, but their contribution is limited in the representation, analysis and solution of complex systems. In such systems, the inspection of their operation depends on human knowledge. Generally, there is a great demand for the developing of autonomous complex systems that can be achieved taking advantage of human like reasoning. Fuzzy Cognitive Maps (FCMs) seem to be capable to deal with situations where the human reasoning process for any procedure includes uncertain descriptions.

FCMs are important modeling means for describing particular domains showing the concepts (variables) and the relationships between them. FCMs can contribute to the construction of more intelligent systems since a more intelligent a system becomes, the more symbolic and fuzzy representations it utilizes.

Over the last 10 years, a variety of FCMs have been used for representing knowledge and artificial intelligence in engineering applications, for instance, fault detection [1], geographical information systems [2] and modelling the supervision of distributed systems [3]. FCMs have been also used for several tasks like web data mining [4], and medical decision-making [5].

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 515–522, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In this research work we propose a learning methodology for fine-tuning FCM causal links in order to update the initial experts' knowledge and to control a process problem efficiently. In general, experts in a certain domain make the task of creating Fuzzy Cognitive Maps (FCMs), but it is very promising the learning of FCMs from the initial experts' knowledge and experience. Till today, very few research efforts have been made to investigate and propose a learning algorithm suitable for FCMs [6,7,8,9]. The introduction of FCM weight adaptation technique eliminates the deficiencies in the usage of FCM and enhances the dynamical behavior and flexibility of the FCM model.

2 Fuzzy Cognitive Maps Background

FCMs are special kind of cognitive maps and have been introduced by Kosko [9] to represent fuzzy cause effect relationships instead of the crisp cause-effect relations represented in the original cognitive maps. FCMs are fuzzy signed digraphs with feedback. Nodes in the graph are fuzzy sets and represent concepts. Directed edges represent causal effect relationships between the concepts as the case of generic cognitive maps.

The FCM structure is similar to a recurrent artificial neural network, where concepts are represented by neurons and causal relationships by weighted links connecting the neurons. Concepts reflect characteristics, of the system. Interconnections among concepts of FCM signify the cause and effect relationship that a concept has on the others. These weighted interconnections represent the direction and degree with which concepts influence the value of the interconnected concepts. Figure 1 illustrates the graphical representation of a Fuzzy Cognitive Map. The interconnection strength between two nodes C_j and C_i is w_{ji}, with w_{ji} taking on any value in the range –1 to 1.

There are three possible types of weights among concepts: $w_{ji} > 0$ which indicates positive causality between concepts C_j and C_i, $w_{ji} < 0$ and $w_{ji} = 0$.

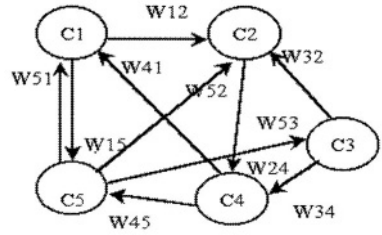


Fig. 1. A simple Fuzzy Cognitive Map

The value of each concept is calculated, computing the influence of other concepts to the specific concept, by applying the following calculation rule:

$$\mathbf{A}^{(k)} = f(\mathbf{A}^{(k-1)} + \sum \mathbf{A}^{(k-1)} \cdot \mathbf{w}) \quad (1)$$

where $\mathbf{A}^{(k)}$ is the vector of values of concepts C_i at iteration step k, and f is the sigmoid threshold function $f(x) = 1/(1+e^{-x})$.

The output vector $\mathbf{A}$ after the implementation of the threshold function is used as the input vector for the $(k+1)^{th}$ stage. This new input vector is again operated on the weight matrix. The following algorithm expresses the FCM interactions.

Algorithm 1: FCM model implications.

Begin

Step 1: Read the input vector $\mathbf{A}^0$

Step 2: Give the connection matrix $\mathbf{w}$

Step 3: Calculate the output vector $\mathbf{O}^{(k)} = \mathbf{A}^{(k-1)} + \sum \mathbf{A}^{(k-1)} \cdot \mathbf{w}$

Step 4: Apply threshold to output vector $\mathbf{A}^{(k)} = f(\mathbf{O}^{(k)})$

Step 5: If ($\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)}$ *), stop*
Else GO TO Step 1.
End

By studying the final state of the iterations, we infer for the model. Equilibrium in the system is attained when we have a set of repeated patterns. Repeated patterns can be fixed points or limit cycles or a chaotic attractor. The "chaotic attractor" is a limit cycle consisting of repeating patterns of differing lengths.

The methodology for developing FCMs is based on a group of experts who are asked to define concepts and describe relationships among concepts and use IF-THEN rules to justify their cause and effect suggestions among concepts and infer a linguistic weight for each interconnection. A more detailed description about the development of FCMs is given in [10].

The most significant weaknesses of the FCMs are their dependence on the expert's opinion and the uncontrollable convergence to undesired states. Learning procedures is a mean to increase the efficiency of FCMs, by modifying the FCM weight matrix.

3 Learning Methods for Fuzzy Cognitive Maps

Fuzzy Cognitive Map learning involves updating the strengths of causal links so that FCM converge in a desired region. An advanced learning strategy is to modify FCMs by fine-tuning its initial causal link or edge strengths through training algorithms similar to that in artificial neural networks.

Up-to-date, there have been proposed just a few unsupervised learning algorithms for FCMs [6,7,8] and they are mostly based on ideas coming from the field of artificial neural networks training. Kosko has proposed the first algorithm, named Differential Hebbian Learning, as a form of unsupervised learning, but without any mathematical formulation and implementation in real problems [6]. There is no guarantee that DHL will encode the sequence into the FCM and till today no concrete procedures exist for applying DHL in FCMs. Another method is the Adaptive Random Fuzzy Cognitive Maps (ARFCMs), which are trained with a reinforcement

learning law. They are based on the theoretical aspects of Random Neural Networks and can be used like associative memories [7]. Despite the recent fusion of FCMs and Random Neural Networks, the resulted Adaptive Random FCM has not any significant results in real dynamic system modeling.

Also, a different approach has been proposed for FCM learning, which is based on evolutionary computation [9]. This technique is exactly the same used in neural networks training. It doesn't take into consideration the initial structure and experts' knowledge for the FCM model and the calculated weights appear large deviations from the actual FCM weights without accepted physical meaning. So, a formal learning methodology suitable for FCMs and for practical applications is still needed.

4 Weight Adaptation Method Based on Nonlinear Hebbian Learning

The proposed weight adaptation procedure is based on the Nonlinear Hebbian Learning rule. The proposed nonlinear Hebbian-type rule for ANNs learning [10] have been adapted and modified for the FCM case, determining the Nonlinear Hebbian Learning (NHL) algorithm.

This learning algorithm is based on the premise that all the concepts in FCM model are triggering at each iteration step and change their values. During this triggering process the weight w_{ji} of the causal interconnection of the related concepts is updated and the modified weight $w_{ji}^{(k)}$ is derived for iteration k.

The value $A_i^{(k+1)}$ of C_i, concept at iteration $k+1$, is calculated, computing the influence of interconnected concepts with values A_j to the specific concept C_i due to modified weights $w_{ji}^{(k)}$ at iteration k, through the following equation:

$$A_i^{(k+1)} = f(A_i^{(k)} + \sum_{\substack{j=1 \\ j \neq i}}^{N} A_j^{(k)} \cdot w_{ji}^{(k)}) \tag{2}$$

Furthermore, some of concepts are defined as Desired Output Concepts (DOCs). These concepts stand for the factors and characteristics of the system that interest us, and we want to estimate their values, which represent the final state of the system. The distinction of FCM concepts as inputs or outputs is determined by the group of experts for each specific problem. experts select the output concepts and they consider the rest as initial stimulators of the system. The learning algorithm that extracts hidden and valuable knowledge of experts can increase the effectiveness of FCMs.

Taking the advantage of the general nonlinear Hebbian-type learning rule for NNs, [10], we introduce the mathematical formalism. The proposed rule has the general mathematical form:

$$\Delta w_{ji} = \eta_k A_i (A_j^{(k)} - w_{ji}^{(k-1)} A_i) \tag{3}$$

where the coefficient η is a very small positive scalar factor called learning parameter and is determined using experimental trial and error method in order to

optimize the final solution. We propose the following form of nonlinear weight adaptation algorithm:

$$w_{ji}^{(k)} = \gamma \cdot w_{ji}^{(k-1)} + \eta^{(k)} A_i \left(A_j^{(k)} - w_{ji}^{(k-1)} A_i\right) \quad (4)$$

where the η is the learning rate parameter and γ is the weight decay parameter.

The value of each concept of FCM is updated, through the eq. (2) where the value of weight $w_{ji}^{(k)}$ is calculated using eq. (4). Only the initially non-zero weights suggested by experts are updated for each iteration step. All the other weights remain equal to zero.

Also, we use two criteria functions for the termination of the proposed algorithm. One criterion is the maximization of a function J, which has been defined by Hebb's rule in [10]. The J have been proposed for the NHL examining the values of DOCs.

The J is defined as: $J = \sum_{i=1}^{l} (DOC_i)^2$ (5)

where l is the number of DOCs.

The second criterion is the minimization of the variation of two subsequent values of DOCs, represented in equation:

$$\left|DOC_j^{(k+1)} - DOC_j^{(k)}\right| < e \quad (6)$$

in order to terminate the iterative process of the learning algorithm. The term e is a tolerance level keeping the variation of values of DOC(s) as low as possible and it is proposed equal to $e = 0.001$, satisfying the termination of iterative process.

Algorithm 2: "Nonlinear Hebbian Learning"
Step 1: Read input concept state A^0 and initial weight matrix w^0
Step 2: Repeat for each iteration k.
Step 3: Calculate $A_i^{(k)}$ according to the eq. (2)
Step 4: Update the weights:
$w_{ji}^{(k+1)} = \gamma \cdot w_{ji}^{(k)} + \eta^{(k)} A_i \left(A_j^{(k)} - w_{ji}^{(k)} A_i\right)$
Step 5: Calculate the two criterion functions
Step 6: Until the termination conditions are met.
Step 7: Return the final weights w_{NHL}.

A generic description of the proposed NHL Algorithm for FCMs is given in Algorithm 2. In NHL algorithm, upper and lower bounds for the learning rate parameters γ and η are determined using trial and error experiments, $0 < \eta < 0.1$ and $0.9 < \gamma < 1$, respectively, and then constant values for each one are proposed for each case-study problem.

5 An Industrial Process Control Problem

A process control problem encountered in chemical industry is selected to illustrate the workings of the proposed weight adaptation algorithm. The process control

problem is a part of a chemical plant consisting of two tanks, three valves, one heating element and two thermometers for each tank, as depicted in Figure 2.

For this system a FCM model has been constructed in [3]. Variables and states of the system, such as the height of the liquid in each tank or the temperature, are the concepts of an FCM, which describes the system. For this simple system eight concepts are proposed and they give a good model of the system:

Concept 1: The height of the liquid in tank 1. The height of liquid is dependent on state of valve 1 and valve 2.
Concept 2: The height of the liquid in tank 2. The height of liquid is dependent on state of valve 2 and valve 3.
Concept 3: The state of the valve 1. The valve is open, closed or partially open.
Concept 4: The state of the valve 2. The valve is open, closed or partially open.
Concept 5: The state of the valve 3. The valve is open, closed or partially open.
Concept 6: The temperature of the liquid in tank 1.
Concept 7: The temperature of the liquid in tank 2.
Concept 8: Describes the operation of the heating element, which has different levels of operation and which increases the temperature of the liquid in tank 1.

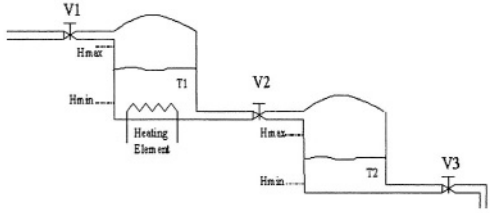


Fig. 2. The illustration for the process control example

These concepts are connected with each other. The signs and the weight of each connection have been determined in [3]. The interconnections among concepts can easily be changed and some new can be added or others can be removed if the human operator decides so, in order to have a better model and operation of the system. Three experts constructed the FCM. The experts determine the cause-effect relationship among concepts. Each connection between concepts has a weight, which ranges between [-1,1] and was determined by a group of experts. Each concept of the FCM takes a value, which ranges in the interval [0,1] and it is obtained after threshold the real measurement of the variable or state, which each concept represent.

The control objective is to keep values of these variables in the following range of values:

$$0.5 \le C1 \le 0.75, 0.75 \le C2 \le 0.8, 0.75 \le C6 \le 0.80, 0.65 \le C7 \le 0.70 \quad (7)$$

The values of the weights will be determined after training the FCM using NHL algorithm. The DOCs in this problem, as experts propose them, are the concepts: C_1 of "Tank 1", C_2 of "Tank 2", C_6 of "Temperature 1" and C_7 of "Temperature 2".

Experts have suggested the appropriate initial weights of the FCM model that are shown in the following weight matrix in eq. (8). The desired regions for the DOCs, which reflect the proper operation of the modeled system, have been prespecified in eq. (7).

$$\mathbf{w}^{initial} = \begin{bmatrix} 0 & 0 & 0.21 & 0.38 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.70 & 0.60 & 0 & 0 & 0 \\ 0.76 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.80 & 0.70 & 0 & 0 & 0 & 0 & 0.09 & 0 \\ 0 & -0.42 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.40 & 0 & 0 & 0 & 0 & 0.50 \\ 0 & 0 & 0 & 0.30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.40 & 0 \end{bmatrix} \quad (8)$$

6 Simulation Results

First, before the implementation and testing of the proposed NHL algorithm, we apply the typical Algorithm 1 to find the final state of the iterations in the previously described chemical process control problem. The initial values of concepts given in vector $\mathbf{A}^0$=[0.48 0.57 0.58 0.68 0.59 0.59 0.52 0.58] and the initial weight matrix $\mathbf{w}^{initial}$ are used in Algorithm 1 to calculate the final state of the process. After 10 iterations the equilibrium state is reached $\mathbf{A}^{equil}$=[0.6256 0.733 0.7675 0.860 0.7704 0.739 0.681 0.7548]. It is observed that the values for $\mathbf{C}_2$ and $\mathbf{C}_6$, in final state, are out of the suggested desired regions in eq. (7).

Next, the NHL algorithm is applied to train the FCM and modify the weights. This experiment concerns the weight adaptation of the FCM model for the specific case problem to converge in the desired convergence region, determining the appropriate values of cause-effect relationships among concepts (weights). The process starts by applying the same initial vector $\mathbf{A}^0$ and weights $\mathbf{w}^{initial}$. Notably, only the initially non-zero weights that connect the triggering concepts are updated at each iteration step The suggested exponential attenuation value of learning parameter η is 0.04 and the associate value for weight decay parameter γ is 0.98, after trial and error experiments, for fast system convergence in desired regions. If the learning parameters η, γ, take different values than the proposed ones, but within the initially suggested limits, then the system will converge in undesired regions.

The proposed algorithmic procedure continues, until the two termination criteria are met. The result of training the FCM is a set of connection weights w_{ji} that satisfies the termination criteria. After 12 iteration cycles, the system converges and the updated weight matrix is the following:

$$\mathbf{w}_{NHL} = \begin{bmatrix} 0 & 0 & 0.294 & 0.432 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.659 & 0.526 & 0 & 0 & 0 \\ 0.573 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.668 & 0.616 & 0 & 0 & 0 & 0 & 0.217 & 0 \\ 0 & -0.171 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.441 & 0 & 0 & 0 & 0 & 0.43 \\ 0 & 0 & 0 & 0.385 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.475 & 0 \end{bmatrix} \quad (9)$$

which leads the FCM to the steady state: C_1 =0.6197, C_2 =0.7632, C_3 =0.7850, C_4 =0.8715, C_5 =0.7620, C_6 =0.7512, C_7 =0.7094, C_8 =0.7440

These new values for weights describe new relationships among the concepts of FCM and the DOCs are within the desired regions. Thus the weight adaptation method adjusts the FCM cause-effect relationships and controls the system convergence in accepted steady states.

7 Conclusions

In this paper, the unsupervised weight learning method, namely Nonlinear Hebbian Learning, is introduced to adapt the cause-effect relationships between concepts and to eliminate the deficiencies that appear in operation of FCMs. In this way, we take into account the dynamic characteristics of the learning process and the environment.

The proposed method for weight adaptation of FCM is presented and accompanied with the good knowledge of a given system or process can contribute towards the establishment of FCM as a robust technique. Also a formal methodology suitable for process control problems has been developed and desired convergence regions for problem have been produced. In future work, further developments of learning algorithms will be examined.

References

1. Pelaez C.E. and J.B. Bowles. Using Fuzzy Cognitive Maps as a System Model for Failure Modes and Effects Analysis, Information Sciences, Vol. 88, (1996), 177-199.
2. Liu, Z.Q. & Satur, R.: Contexual fuzzy cognitive map for decision support in geographical information systems. In: J. of IEEE Trans. On Fuzzy Systems, 7, (1999), 495-507.
3. Stylios, C.D., Groumpos, P.P.: Fuzzy Cognitive Maps in Modeling Supervisory Control Systems, Journal of Intelligent & Fuzzy Systems, vol. 8, (2000), 83-98.
4. Lee, K.C., Kin, J.S., Chung, N.H. & Kwon, S.J.: Fuzzy cognitive map approach to web-mining inference amplification, In: J. of Expert Systems with Applic., 22, (2002), 197-211.
5. Papageorgiou, E.I., Stylios, C.D., Groumpos, P.P.: An Integrated Two-Level Hierarchical Decision Making System based on Fuzzy Cognitive Maps (FCMs), IEEE Transactions on Biomedical Engineering, Vol. 50, No 12, pp. 1326-1339, December 2003.
6. Kosko, B.: Fuzzy Cognitive Maps, Int. J. Man-Machine Studies, Vol. 24 (1986), 65-75.
7. Aguilar, J.: Adaptive Random Fuzzy Cognitive Maps. In: Garijio, F.J., Riquelme, J.C., Toro, M. (eds.): IBERAMIA 2002, Lecture Notes in Artificial Intelligence 2527, Springer-Verlag Berlin Heidelberg (2002), 402-410.
8. Papageorgiou, E.I., Stylios, C.D., Groumpos, P.P.: Activation Hebbian Learning Rule for for Fuzzy Cognitive Map Learning, *Proc. of 15th IFAC World Congress of Inter. Federation of Automatic Control,* Barcelona, Spain, July 21-26,CD-ROM, 2002.
9. Koulouriotis, D.E., Diakoulakis, I.E., Emiris, D.M.: Learning Fuzzy Cognitive Maps using evolution strategies: A novel schema for modeling a simulating high-level behavior. *Proc. 2001 IEEE Congress on Evolutionary Compu.,* Seoul, Korea, 2001, vol. 1, pp. 364-371.
10. Oja, E., Ogawa, H., Wangviwattana, J.: Learning in nonlinear constrained Hebbian networks. In. T. Kohonen et al. (eds.): Artificial Neural Networks (1991), Amsterdam: North-Holland, 385-390.

A Method Based on Fuzzy Logic Technique for Smoothing in 2D

Ahmet Çinar

Fırat University, Faculty of Engineering, Computer Engineering, 23119 Elazig, Turkey
acinar@firat.edu.tr

Abstract. This paper describes a novel approach based on fuzzy logic technique for smoothing process on the unstructured meshes in 2D. The proposed method works local on triangulation. Therefore, remeshing does not require. It is known that other smoothing operations such as laplacian smoothing , optimization based smoothing, angle based smoothing, and hybrid smoothing operations work global. That is, in these methods, meshing is achieved, afterwards remeshing is done for smoothing. Whereas, the goal of presented method is to find the best fit location of node while meshing is done. However, remeshing process is avoided.

1 Introduction

The basic element in unstructured mesh generation in 2D is triangle and quadrangle. Triangulation is known as creating mesh by using triangles [1]. For optimum triangulation, elements of the created unstructured mesh must be equal size. The often equilateral triangle or isosceles triangle are used. This process is called as smoothing[2],[3]. The goal of smoothing is to obtain regular elements in mesh. However, in the solving domain such as finite element method, the more corrected results can be achieved.

There are many kinds of smoothing in the literature. These are; laplacian smoothing [4],[5],[6], optimization based smoothing, hybrid smoothing which is combining of laplacian and optimization based smoothing,, angle based smoothing, variations of laplacian smoothing[7], winslow smoothing,..., etc. Although laplacian smoothing need computational cost less than other methods, invalid elements can be generated in mesh. On the other hand, while optimization]based smoothing is more likely to avoid invalid elements, the computational costs is higher than laplacian smoothing. Hybrid methods eliminate generating of invalid elements of laplacian smoothing and is decreased computational cost of optimization based smoothing. But, mesh quality of hybrid methods is not as good as pure optimization based smoothing. Angle based smoothing provides good mesh quality, but it needs computational cost less than optimization based smoothing.

The common property of all methods is the working as global smoothing. That is, firstly mesh is created and then remeshing is done for smoothing. This situation causes more time and more computational cost.

In this study, while creating a mesh, the smoothed mesh is produced in 2D. So, remeshing process does not need. Fuzzy logic technique based a method is applied to

M. Bubaket al. (Eds.): ICCS 2004, LNCS 3037, pp. 523–531, 2004.
© Springer-Verlag Berlin Heidelberg 2004

smoothing process. It has known that, fuzzy logic technique has used firstly by [8]. First application areas are control applications. And then, different areas are applied to such as speech and pattern recognition, image processing,..., etc. Applications about computer graphics are curve modeling [9], curve smoothing and blending[10],...,etc.

The proposed method works more effective and more speed according to aforementioned methods. Due to the fact that fuzzy logic technique is used, the method does not need more computational cost. Also, quality of mesh according to discrete points on the boundary curve is attractivable. The only effected factor of mesh quality is the size of triangle which is dependent on distance of discrete points and angle of discrete points. Presented method is avoided invalid elements which is drawback of laplacian smoothing method. Due to the fact that smoothed mesh process is done and as possible as equilateral triangle or isosceles triangle is used, the method has some advantages. Method is detailed in section two and three.

The organization of paper is as follows; section two describes a presented method which is the based on fuzzy logic. In Section 3, fuzzy membership functions, rule base table and a simple application are given. At the finally conclusion is offered.

2 Fuzzy Logic Based Smoothing

In this section, fuzzy based smoothing method (FBSM) is described. The basis of presented method is to create equilateral triangle or isosceles triangle. The problem we have interested is to generate smoothed triangulation in 2D. The goal is to form equilateral triangle or isosceles triangle as possible as. It is known that, a good triangulation algorithm should be provided following criteria [11];

- The proposed must be general. That is, it must be independent on positions of triangles.
- After smoothing process has finished, unsmoothed element should not be remain of.
- It must be effective as computational cost and time requirements
- It must be easy-to-implement.

The presenting method provides above conditions. In the following section, the application method is gives detailed.

2.1 Discretisation of Boundary Curve

For this process, geometric support construction method is used. The set of polygonal segments of boundary curve is located in the any list. And then, the construction of the polygonal support is such that the distance any segment and portion of the curve it represent must be bounded. Hence, refinement of curve will be more important and correct in the any highly curved regions on the boundary curve.

Let be a distance between a point and the supporting edge, and let b be the length of the edge. If ration of a/b is less than ε we will determine, the segment we have selected is correct or wrong. If it is wrong, then segment is subdivided into sub-

segments, each of them being analyzed recursively. As a result, we can make it geometric support. The ration a/b is user-defined.

Fig. 1. a. Approximation using polygon segment b. Discretized boundary values of circle

2.2 Locating of First Triangle

The presenting method is similar to advancing front in way of starting process. So, first triangle is located between two nodes selected randomly from list.

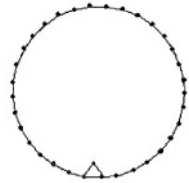


Fig. 2. Location of first triangle

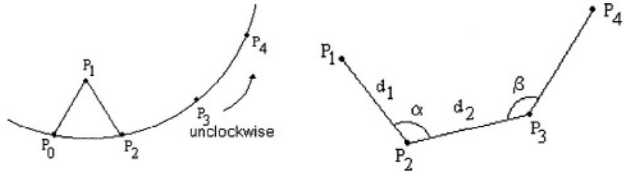


Fig. 3. a. Determining of list direction and node location for triangulation; b. Angles α and β, distances d_1 and d_2.

2.3 Fuzzy Smoothed Triangulation Method in 2D

Let points P_0,P_2,P_3,P_4 be discrete points and point P_1 be a vertex of first triangle. Firstly, triangle $P_0P_1P_2$ is located as shown figure 3.a. Afterwards, the operation that will be done is that triangles are to produce by selecting node position in way of interior of closed region in unclockwise direction. For this process, the following steps are applied;

Step 1.

The first triangle is that nodes are $P_0P_1P_2$. The purpose is to generate a node in such a way that, based edge of triangle is P_2P_3 and such that triangle to be created is equilateral or isosceles triangle. Let us draw a line dividing into two equal segment the angle α. The goal is to find the best location on the line to be drawn form point P_2. It must be found so point that, it will be proportional to distances d_1 and d_2. However,

triangle to be created will be equilateral or isosceles triangle. Firstly, selection of P_y seems to be reasonable at distance of d1 figure 4.b). The dark line depicts edges of a new triangle.

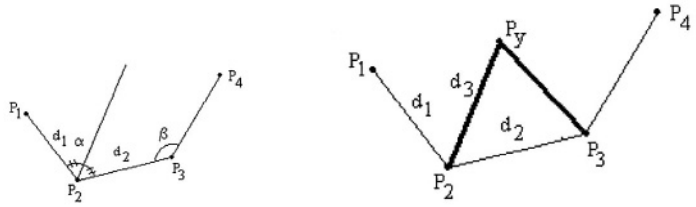


Fig. 4. a. Line drawing to divide into two equals segment the angle b. α.The situation of d_3 equal d_2

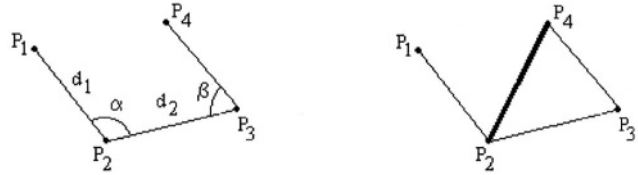


Fig. 5. a. The proposed method for β<90; b. Triangle producing by connecting P_2P_4.

In this way, triangle to be produced is the isosceles at the worst case. But, distance d1 is not proportional d_3. Therefore, unsmoothed triangles will occur when triangulation process is proceeded. To avoid this situation, distances d3 can be selected as half of summation of d_1 and d_2(equation1).

$$d_3=(d_1+d_2)/2 \quad (1)$$

Also, a different case may be position of P_4 (figure 5.b).At this case, instead of finding a new P_y, directly triangulation is done.

In this case, triangle $P_2P_3P_4$ is produced by connecting P_2P_4.

When the list has proceeded, same situation may appear while inner points are computed. Therefore, the position of $P_1P_2P_3$ is important.

At this case, triangulation is done without computing a new point. The vertex of new triangle to be created is $P_1P_2P_3$. (Fig. 6.b8).

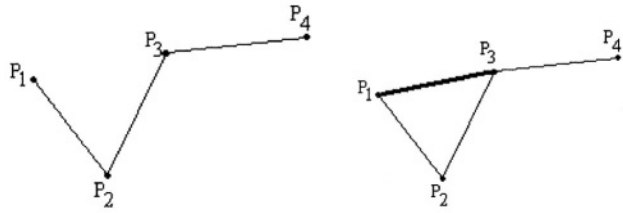


Fig. 6. a. Different $P_1P_2P_3$ points; . b.A new triangle to be created

Noticed that, the basic parameters in the triangulation process are angle α,β and distances d_1, d_2. Therefore, a new method that will be developed by using angles α

and β, distances d_1 and d_2 will provide fast and smoothing. For this process, fuzzy logic based a method is proposed in section3.

The following steps are shown the proposed method according to angles of α and β.

1. For α<90; triangulation is performed without finding a new point(figure 7.a)
2. For 90= α <180; It is important the value of angle β.
   a. For β <=90; two triangle is produced. The one of them is triangle P_1P_2 and other is $P_2P_3P_4$ (figure 7.b)

b. For β >90; A new point is computed by means of fuzzy logic technique. If it is noticed to figure 11, the position of P_4 is not so important. The positions of P_4 are shown as P_4',P_4'',P_4'''.

Another situations of node positions may be find out similarly.

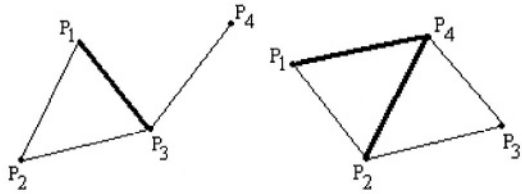


Fig. 7. a. Triangle for α<90. b.Triangle for β<=90.

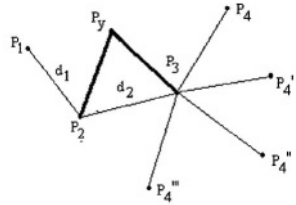


Fig. 8. A new triangle is triangle of $P_2P_3P_y$

3 Membership Functions and Rule Base Table

In this section, the membership functions and rule base table for fuzzy logic technique are given. Angle α and β and distances d_1 and d_2 are used for optimal triangulation. Angles α and β change in the interval [90,360] according to boundary of object. Also, distances d_1 and d_2 can change random according to discretisation of boundary. Therefore, proportional of them is used (equation 2).

$$k = d_1/d_2, \quad for\ d_1>d_2$$
$$k = d_2/d_1, \quad for\ d_1>d_1 \qquad (2)$$

If it is noticed to figure 9, the constant k can seem as if it were changed in the large interval. Whereas, due to the fact that almost equal discretization is made, the changing of k will be in certain interval.

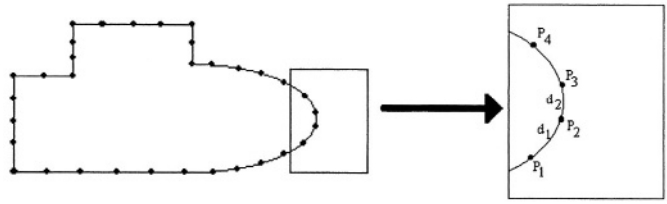


Fig. 9. The values of d1 and d2 .

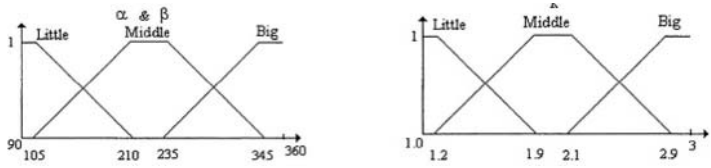


Fig. 10. Input membership functions

At this way, the tracing boundary and continuity of derivative would be provided. In this paper, the changing of k will be considered as [0.01,3]. At the other words, for $d_1 > d_2$, d_1 will be three times of d_2. It is known that, firstly, the determination of membership functions and rule base table is required for fuzzy logic technique. Angles α and β and constant k are used as inputs of fuzzy logic. Figure 10 shows input membership functions.

Some outlines on developing the membership function for fuzzy sets are listed as follows;

1. Membership function as a rational function of polynominals,
2. Membership function as a model for human concepts,
3. Heuristically based membership functions,
4. Membership functions based on statistics or probability density functions [12].

In this paper, we concern with human concept based membership function.

Figure 11 shows output parameters of fuzzy logic. These are distance d_3 and angle of γ.

Figure 12 depicts output membership function.

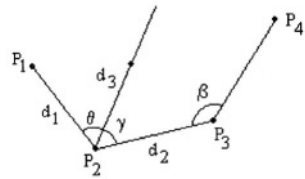


Fig. 11. Parameters using for output.

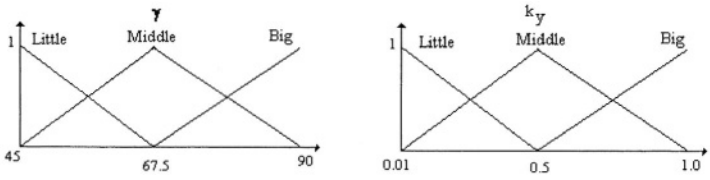


Fig. 12. Output membership function.

Distance d_3 is proportional k_y. Equation 3 depicts this relation.

$$d_3 = k_y * d_1, \quad for\ d_1 > d_2$$
$$d_3 = k_y * d_2, \quad for\ d_2 > d_1 \tag{3}$$

As defuzzyfication, middle of maximums are used in this study (Equation 4)

u_i: i^{th} value of membership function u.

$\mu(u_i)$: The maximum value of membership function u_i.

$$u^* = \frac{\sum_{i=1}^{l} .\mu(u_i).u_i}{\sum_{i=1}^{l} \mu(u_i)} \tag{4}$$

Table 1 shows rule base table for fuzzy logic.

Table 1.

Inputs			Outputs		Inputs			Outputs	
α	β	k	γ	k_y	α	β	k	γ	k_y
1	1	1	1	3	2	2	1	1	2
1	1	2	1	1	2	2	2	1	1
1	1	3	1	1	2	2	3	1	3
1	2	1	1	3	3	3	1	1	2
1	2	2	1	2	3	3	2	1	1
1	2	3	1	1	3	3	3	1	2
1	3	1	1	3	3	1	1	1	3
1	3	2	1	1	3	1	2	1	2
1	3	3	2	3	3	1	3	1	1
2	1	1	2	2	3	2	1	1	3
2	1	2	2	1	3	2	2	2	2
2	1	3	1	3	3	2	3	1	1

For α, β and k; 1: Little, 2: Middle and 3: Big. For γ and k_y ;1: Little, 2: Middle and 3: Big.

After computing angle of γ and k_y, location of py is computed in simple analytical manner.

The following section describes how to apply of fuzzy logic and how to find of Py

3.1 Application

In the presented application, how fuzzy logic technique was applied to smoothing is described.

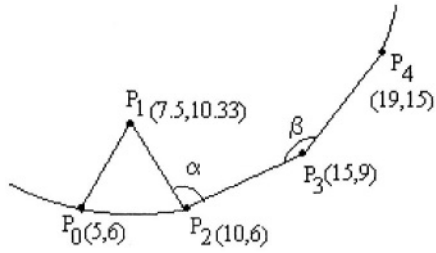


Fig. 13. An object which boundary was discreted

The first step of proposed method is to locate first triangle. For this process, the first triangle is constructed as equilateral triangle. The next step is to compute angles α, β and distances d_1, d_2 These values are computed as α=98.42° ,β=171.9° ,d_1= 4.89 and d_2=5.83.

γ and k_y values are computed by means of α=98.42° ve β=171.9° ve d_1= 4.89 ve d_2=5.83, which is inputs of fuzzy logic technique. These values are γ=53.1 and k_y=0.736 and d_3=0.735*5.83=4.29. Figure 14 depicts position of P_y.
The method is iterated according to object boundary. Figure 15.a shows unsmoothed object and figure 15.b shows smoothed object.

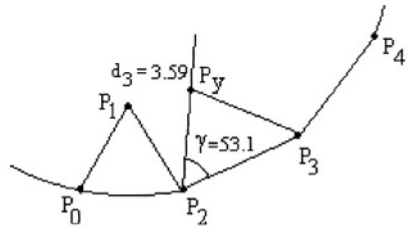


Fig. 14. The Computing value of P_y.

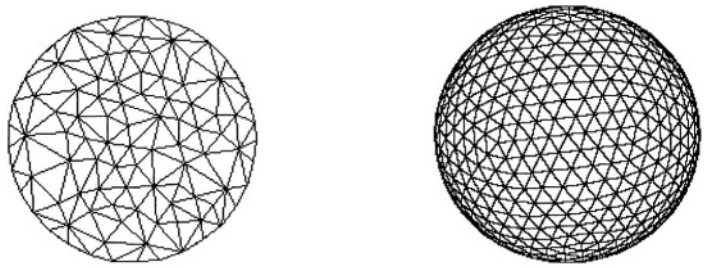


Fig. 15. a. Unsmoothed object **b.** Smoothed object.

4 Conclusion

In this study, a novel method based on fuzzy logic technique for smoothing in 2D is proposed. The use of fuzzy logic technique has provided only remeshing process for smoothing. Therefore, the proposed method is differ from other method in the literature. Due to the fact that smoothing process is achieved, remeshing process doesn't require. So, computational cost is decreased.

The proposed method has a little time requirements for fuzzy computation. This time is very little. In the proposed method, the changing of angle is not considered. As future work, the proposed method will be tried to be applied 3D object.

References

1. Parhasarthy V., S. Kodiyalam, "A Constrained optimization approach to finite element mesh smoothing", Finite Elemnts in anlysis and Design, Vol: 9. Pp:309-320, 1991
2. Field D.A., "Laplacian Smoothing and Delaunay Triangulation", Communications in Applied Numerical Methods, Vol: 4, pp:709,712, 1988
3. Freitag L.A., "On Combining Laplacian and Optimization based Mesh Smoothing Techniques,"AMD Trends in Unstructured Mesh Generation, Vol:220, pp:37,43, 1997
4. Hansbo P, "Generalized Laplacian Smoothing of Unstructured Grids" Communication in Numerical Methods in Engineering , Vol: 11:pp:454-464,1995
5. Zhao T., Shimada Kenji, "An Angle-based approach to two dimensional mesh smoothing" , 9th Int. Meshing Roundtable, pp:373-384, 2000

6.. Canann S., J.R. Tristano, M.L.Staten, "An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral and Quad-Dominant Meshes", 7th Int. Meshing Roundtable, pp:479-494. 1998

7 Saga S. Makino H., "A Method for Modeling Freehand Curves-the Fuzzy Spline Interpolation", Systems and Computers in Japan , Vol: 26, No:19, pp:1610-1619, 1995.

8 Zadeh L. "Fuzzy sets ", Inf. Control, Vol:8, pp:338-353, 1965.

9 Gao Lun Shan, Kawarada Hideo, "Applications of fuzzy average to curve and surfaces fitting", 2nd Fuzzy Eng. Sym.IEEE Inter. Con. on Fuzzy systems, Vol. 2, p 971-978 , 1995.

10 Ahmet Çınar, Ahmet Arslan, "Blending of 2D Objects Using Fuzzy Logic", Lecture Notes in Computer Science, Vol: 2206, pp: 968-976, 2001

11 Owen S.."A Survey of Unstructured Mesh Generation Technology" 7h Int. Meshing Roundtable, pp:239-268, 1998

12 Parhasarthy V., S. Kodiyalam, "A Constrained optimization approach to finite element mesh smoothing", Finite Elemnts in anlysis and Design, Vol: 9. Pp:309-320, 1991

Proportional-Integral-Derivative Controllers Tuning for Unstable and Integral Processes Using Genetic Algorithms

Marco Antonio Paz-Ramos[1,2], Jose Torres-Jimenez[3], and Enrique Quintero-Marmol-Marquez[2]

[1] Instituto Tecnologico y de Estudios Superiores de Monterrey Campus Ciudad de Mexico Calle del Puente 222 Col. Ejidos de Huipulco Tlalpan, 14380, Mexico, D.F., Mexico marco.paz@itesm.mx

[2] Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) Interior Internado Palmira S/N Cuernavaca, Morelos, Mexico {pazramos,eqm}@cenidet.edu.mx

[3] Instituto Tecnologico y de Estudios Superiores de Monterrey Campus Cuernavaca, Computer Science Department. Av. Paseo de la Reforma 182-A. Lomas de Cuernavaca 62589 Temixco Morelos, Mexico jtj@itesm.mx Tel. +52-777-3297169, Fax +52-777-3297168

Abstract. During the last years the use of intelligent strategies for tuning Proportional-Integral-Derivative (PID) controllers has been growing. The evolutionary strategies have won an important place thanks to their flexibility. In this paper, the automatic tuning of systems with scarce initial information and integrative and unstable dynamics, through a genetic approach is presented. The advantages of the proposed approach were highlighted through the comparison with the Ziegler-Nichols modified closed loop method, and the Visioli genetic approach. The proposed methodology goal is to expand the intelligent tuning application to a wider range of processes (covering systems with oscillatory or unstable modes).

Keywords: PID Controllers, Genetic Algorithms.

The first attempts to automate the tuning of Proportional-Integral-Derivative (PID) controllers were based on the time response of a process, but this approach had the drawback of requiring a lot of user interaction. A very important advance was made when it was decided to use the frequency response of a process instead of its time response, in this way a bigger degree of automation was attained. The first industrial PID controller based on processes' frequency response, was the Novatune [1], which initialized a PID controller automatically.

Applications based on PID controllers using frequency response were very successful [1] [2] [3], however the range of their applicability is restricted to asymptotically stable systems, due to the use of the relay feedback. By using

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 532–539, 2004.
© Springer-Verlag Berlin Heidelberg 2004

evolutionary meta heuristics, like Genetic Algorithms (GA), it is possible to expand the application range of PID controllers to systems with oscillatory or unstable modes. The application of GA to control applications has been increasing steadily and recently GA were applied: a) to tune Multiple-Input-Multiple-Output (MIMO) systems [4] [5]; and b) to improve significantly the performance of plants [6].

In this paper, a GA was used to automate the tuning of systems with scarce initial information and integrative and unstable dynamics, in order to get insight of the advantages of using a GA, the results were compared against the Ziegler-Nichols modified closed loop method. The rest of the paper is organized in four more sections: a) in section two the Ziegler-Nichols closed loop control method is presented; b) in section 3 the Visioli genetic method for integral and unstable processes was highlighted; c) in section 4 the Direct Genetic PID proposed method, that is able to complement the Ziegler-Nichols and Visioli methods is presented; and d) finally in section 5 main conclusions will be presented.

1 Ziegler-Nichols Closed Loop Method

With the microprocessor arrival, the industrial field controllers' horizons were expanded, thanks to the capacity to carry out elaborated algorithms. This situation encouraged some controller makers to develop auto tuning PID controllers. One example of this controllers is the Foxboro EXACT [7], whose operation was based on the transient response analysis. An alternative was the use of Ziegler and Nichols closed-loop method [7]. The original method consists of varying the proportional gain until a sustained oscillation is reached. Maintaining the system in this state, the amplitude and frequency of the oscillations are determined by measuring the system output. A PID controller has the standard form depicted in equation (1), where $u(t)$ defines the controller output (also called process output), $e(t)$ is the system error (difference among the desired set point for the variable under control, and the actual process variable value), K_p indicates the proportional gain, T_d is the derivative time constant, and T_i is the integral time constant.

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right) \tag{1}$$

The PID controller parameters for equation (1) are calculated and fixed according to the table 1, where K_u is the ultimate gain, and T_u is the ultimate period. The Ziegler and Nichols method for tuning an ISA PID controller, is a thumb rule based on the Nyquist curve. According to the Nyquist theorem a process is stable if the Nyquist curve does not encircle the point (-1,0).

1.1 Using the Relay Feedback System

Hägglund and Åström [1] [9] introduced an alternative to the increment of the proportional gain to achieve the oscillations inside a bigger margin of security

Table 1. Regulator parameters obtained by the Ziegler-Nichols closed-loop method

Controller	K_p	T_i	T_d
P	$0.5K_u$		
PI	$0.4K_u$	$0.8T_u$	
PID	$0.6K_u$	$0.5T_u$	$0.12T_u$

that the original free oscillations system. A non linear circuit was introduced. The non linear function has a relay characteristic. To obtain the value of the proportional gain according to the table 1 is necessary to calculate the equation (2) where d is the output relay amplitude and a is the oscillation amplitude, while T_u (necessary to obtain T_i and T_d) is measured directly of the oscillations.

$$K_u = \frac{4d}{a\pi} \tag{2}$$

1.2 Stability of Limit Cycle

The relay feedback method is successfully used in the industry [1], however to measure K_u and T_u is fundamental that the limit cycle were stable and symmetric. Åström[8] developed an analysis consisting in obtaining the Jacobian of the Poincaré map (W) which is related to the relay oscillations; the limit cycle is locally stable if and only if W has all its eigenvalues inside the unit disk. The phase portrait is a practical way to know if a process oscillation is stable or not, when the relay feedback is applied. Suppose the plant described in equation (3).The phase portrait of equation (3) obtained during the relay experiment is shown in the figure 1(a) . We can observe in the phase portrait that exists a stable limit cycle.

$$G(s)_1 = \frac{1}{(s+1)^3} \tag{3}$$

Now, if we take the phase portrait of the process represented in equation (4), the limit cycle is unstable, and it can be observed in figure 1(b). It is possible to obtain similar results for processes described in equations (5) and (6), given that, their limit cycles are stable but no asymptotically stable [8] (also these processes are unstable when the feedback relay is used).

$$G(s)_2 = \frac{1}{s^3 + s^2 + s + 1} \tag{4}$$

$$G(s)_3 = \frac{1}{s^2} \tag{5}$$

$$G(s)_4 = \frac{1}{s^2 + 1} \tag{6}$$

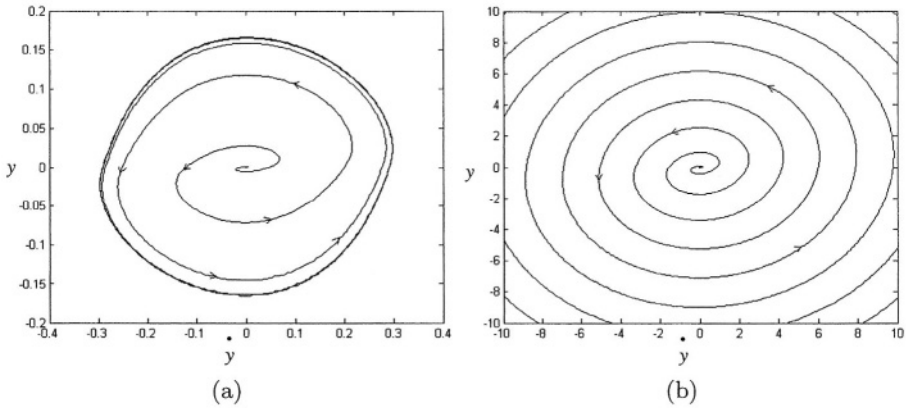


Fig. 1. (a) Phase portrait during the relay experiment for equation (3), (b) Phase portrait during the relay experiment for equation (4).

Generally the integrative and unstable processes do not allow reaching a controlled oscillation. If do not exist a stable oscillation it is not possible to apply the Ziegler and Nichols methods, one of the alternatives to tuning PID controllers for integral and unstable processes is the use of GA. The Visioli method [12] presented in next section uses internally a GA.

2 Visioli Genetic Method for Integral and Unstable Processes

In this section, a brief introduction to GA concepts will be presented first, and then the Visioli method (based on GA) is detailed. The original idea of GA was to achieve that the computing processes could have an evolutionary profile. In the last years there has been an important increase of genetic algorithms applications to the automatic control field. A genetic algorithm emulates an evolutionary process [11], thanks to a generation of virtual individuals population. The virtual population has a reproduction and death process, which allow space for a new generation. According to the schemata theorem [11] if reproduction possibilities for the better adapted individual are increased, the future generations will be more capable on the average than their predecessors. If we take the population individuals as potential solutions of a problem, and we build the algorithm so that the most capable individuals are the nearest to the solution, then we can hope after a certain number of generations, the solution to our problem.

Visioli presents in [12] a methodology for tuning PID controllers for integral and unstable processes. The method consists on a set of rules for integral and unstable processes, which are similar to the Ziegler and Nichols rules. These rules were obtained using a Genetic Algorithm optimization approach. The method contemplates the structure described in equation (7) for integrative processes

(K is the transfer function gain, and L is the system delay), and the structure described in equation (8) for unstable processes (T is a time response rate).

$$G(s) = \frac{K}{s} e^{-Ls} \tag{7}$$

$$G(s) = \frac{K}{Ts - 1} e^{-Ls} \tag{8}$$

In a Visioli paper [12] tables for tuning rules for optimal set-point response for unstable processes can be found. The rules are optimized to minimize the integral of the quadratic error (ISE).

Suppose an unstable plant whose transfer function is given by the equation (9). In accordance with Visioli rules [12], the parameter values will be: $K_p = 5.8$, $T_i = 1.88$, and $T_d = 0.11$. In the figure 4(a) the response of the process described in equation (9) when is controlled by the equation (1) is shown.

$$G(s)_5 = \frac{1}{s - 1} e^{-0.2s} \tag{9}$$

The main virtue of the Visioli method is its simplicity of use; however it has three important disadvantages: a)an identified model of the process is needed; b) the identified model has to be of first order approach; and c) If $L = 0$ (L is the system delay), the method can not be used. In the next section we propose a methodology for tuning a PID control for integral and unstable process which is independent to the transfer function order. The method is a direct optimization of the controller parameters across a genetic algorithm, as it will be shown this method is superior to the Visioli approach.

3 Direct Genetic PID

The use of genetic algorithms for tuning PID controllers has been growing, because is very easy to connect the structure of the controller with the optimization strategy. The genetic algorithm task is to optimize the controller parameters, by means of an evolutionary process. Suppose a chromosome coded as is indicated in equation (10), where C_k is the gain chromosome. C_k represents a simple individual whose genes are the PID controller parameters ($C_k = K_p T_i T_d$, $K_p = p_1, p_2, ..., p_n$, $T_i = i_1, i_2, ..., i_n$, and $T_d = d_1, d_2, ..., d_n$) . Every chromosome is tested and evaluated. The individuals with fitness higher than the average shall have more reproduction opportunities. The gains that allow a better control performance give more chance to inherit the chromosome profile. To optimize the search, the decoding indicated in equation (11) is carried out. In equation (11), A_i is a gene of the chromosome C_k, which is weighted by a decoder weight i, fixed in a heuristic way (see [4]), and K_i is a gain used by the equation (1). In the figure 2 is shown the optimization process, and it can be observed an on line identification process which use recursive least squares (RLS). The intern

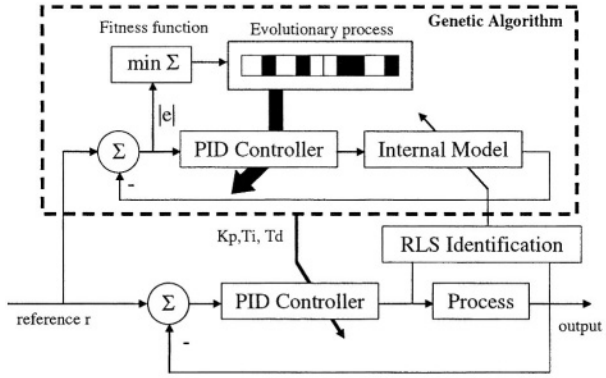


Fig. 2. Optimization structure

model is used in the fitness function described in equation (12), where r_i is the reference point, $\widehat{y}$ is the estimated process and n is a value that allows the output process establishment, when it is stable otherwise n will have a default value. In the equation (12), the minimization criterion is the integral of absolute error (IAE).

$$C_k = p_1, p_2, ..., p_n, i_1, i_2, ..., i_n, d_1, d_2, ..., d_n \tag{10}$$

$$K_i = \Delta_i A_i \tag{11}$$

$$FF = \sum_{i=0}^{n} |r_i - \widehat{y}| \tag{12}$$

The higher qualified individuals are those that minimize the equation (12). The controller parameters are modified using a commutation only when an important parameter variation happens.

The genetic algorithm for the optimization has a population of 100 members. In every reproduction process a tournament selection is performed involving 10% of population. The maximum allele value is 9.0072×10^{15}. For maximum focus in the search, decoding considers the value dividing the allele between ten times its maximum value. A mutation process which involves 10% of the population is applied.

Previously, it was stated that the process described in equation (4) does not have a stable limit cycle; therefore it is not possible to use the Ziegler and Nichols methods. Applying the Direct Genetic PID, it is possible to control a marginally stable process. In the figure 3(a), it is shown the plant controlled output and the evolutionary process evolution. The resultants gains for the process defined in equation (4) are $K_p = 0.156$, $T_i = 0.312$ and $T_d = 9.96$.

The genetic optimization to control the double integrative process described in equation (5) is presented in figure 3(b). The resultants gains for the process described in equation (5) are $K_p = 4.251$, $T_i = 99.16$ and $T_d = 0.781$. Also is shown the control for the unstable delayed process described by equation (9) in figure 4(b). The resultants gains for the process described in equation (9) are $K_p = 2.910$, $T_i = 1.367$ and $T_d = 0.135$.

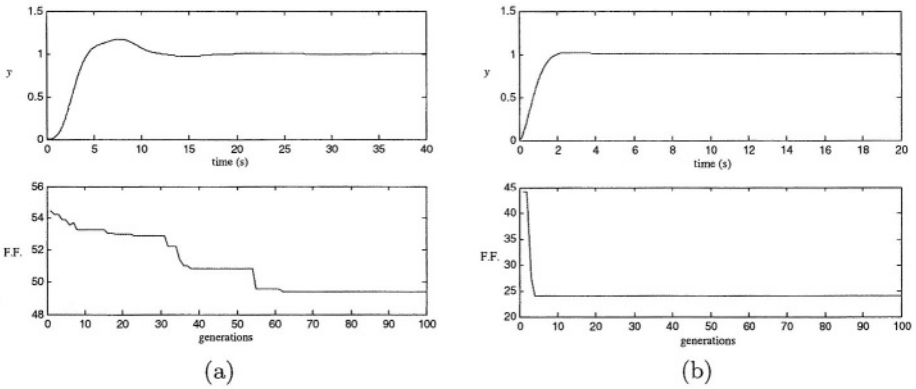


Fig. 3. (a) Genetic PID control of the process described in equation (4), (b) Genetic PID control of the process described in equation (5).

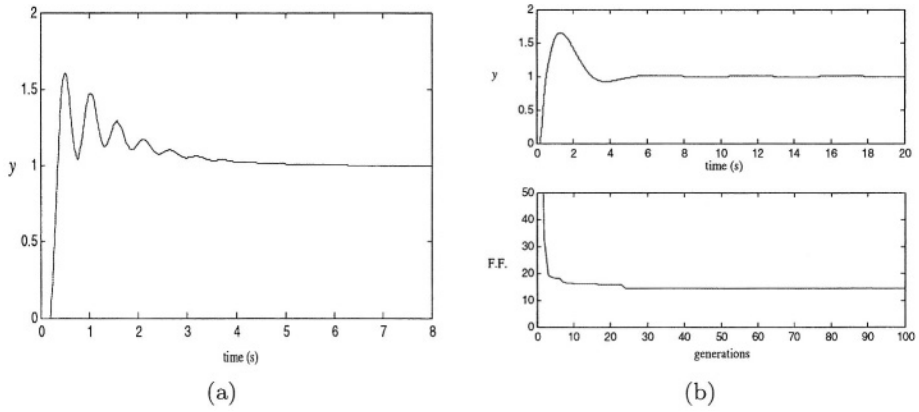


Fig. 4. (a) PID Control of the process described in equation (9) using Visioli method, (b) PID control of the process described in equation (9) using the proposed method.

4 Conclusions

In this paper a new Genetic PID approach was presented to make control of systems with oscillatory or unstable modes. The main advantages of the presented method relate to the possibility of make PID tuning when other methods have failed (the Visioli method is not able to handle the case when $L = 0$ meanwhile the proposed method is capable to handle this case). Satisfactory results were observed through the exhibited tests in the paper. It is intended that the Genetic PID approach can be connected easily as a backup controller for other adaptive controllers. It is remarkable that the Genetic PID approach proposed does not require to know the process features, because an estimation stage (RLS) can be used.

References

1. Häglund T., and Åström K. J., *Industrial Adaptive Controllers Based on Frequency Response Techniques,* Automatica, Vol. 27, No.4, (1991) 594-609.
2. Hägglund T, and. Åström K.J., *Supervision of Adaptive Control Algorithms,* Automatica. Vol. 36, (2000) 1171-1180.
3. Isermann R., and Lachmann K.-H., *Parameter-adaptive Control with Configuration Aids and Supervision Functions,* Automatica, Vol. 21, No. 6, (1984) 625-638.
4. Vlachos C., Williams D., and Gomm J. B., *Genetic Approach to Decentralized PID Controller Tuning for Multivariable Processes,* IEE Proc. Control theory Appl. Vol 146, No. 1, (1999) 58-64.
5. Paz R. M.A., García B. C. D., and Torres J. J., *Fuzzy-Genetic Controller for a Coupled Drives System,* IEEE International Symposium on Industrial Electronics, Puebla, Mexico (2000) 741-746.
6. Krohling R. A., and Rey UJ. P., *Design of Optimal Disturbance Rejection PID Controllers Using Genetic Algorithms,* IEEE Transactions on evolutionary computation. Vol 5, No. 1, (2001) 78-82.
7. Åström K. J., and Hägglund T., *PID Controllers: Theory, Design, and Tuning* ISA organization 1995, ISBN: 1556175167
8. Åström K. J. et al (ed.), *Adaptive Control, Filtering, and Signal Processing,* Springer-Verlag 1995,ISBN: 0387979883
9. Hägglund T., and Åström K. J., *Method and Apparatus in Tuning a PID-Regulator,* U.S. Patent. Number 4549123, 1995
10. Holmberg V. Relay, *Feedback of Simple Systems,* Ph.D. Thesis, August 1991, Department of Automatic Control Lund Intitute of Technology
11. Goldberg D., *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison Wesley 1989,ISBN: 0201157675
12. Visioli A., *Optimal Tuning of PID Controllers for Integral and Unstable Processes,* IEEE Proc. –Control Theory Appl., Vol. 148, No. 2. (2001) 180-184.

Enabling Systems Biology: A Scientific Problem-Solving Environment

M. Singhal, E.G. Stephan, K.R. Klicker, L.L. Trease, G. Chin Jr., D.K. Gracio, and D.A. Payne

Computational Sciences and Mathematics,
Pacific Northwest National Laboratory, Richland, Washington
{Mudita.Singhal, Eric.Stephan, Kyle.Klicker, Lynn.Trease, George.Chin, Debbie.Gracio, Debbie.Payne}@pnl.gov

Abstract. Biologists today are striving to solve multidisciplinary, complex systems biology questions. To successfully address these questions, software tools must be created to allow scientists to capture data and information, to share this information, and to analyze the data as elements of a complete system. At Pacific Northwest National Laboratory, we are creating the Computational Cell Environment, a biology-centered collaborative problem-solving environment with the goal of providing data retrieval, management, and analysis through all aspects of biological study. A horizontal prototype called SysBioPSE, demonstrates this vision. Our initial work is centered on developing the Distributed Data Management and Analysis subsystem, which is a specific tool for retrieving data from multiple heterogeneous data stores, providing storage facilities that support pedigree tracking and data and information analysis under a common user interface. With time, many such individual subsystems will be developed and integrated to fulfill the Computational Cell Environment vision.

1 Introduction

The genomics revolution is beginning to produce data at a rate that will soon rival that of high-energy physics. Biologists and informaticists are faced with critical issues of how to best organize this information and share it so that it will provide the greatest value to the greatest number of researchers. Major advances are being realized in the biological sciences by applying a systems approach. However, there are many obstacles that prevent scientists from using the vast amounts of data that are generated and made available on a daily basis. Systems biologists need access to distributed data repositories. Once the data are collected, tools are needed to integrate the heterogeneous data sets. Informatics-based analysis tools are then needed to analyze the data and help draw conclusions. Next, the researcher needs access to tools to easily share the data and knowledge with other researchers. The traditional approaches to answering biological questions are focused on the examination of a single gene or protein, whereas systems biology focuses on studying the complex interactions of biological systems integrating across multiple streams of data from various sources. The Computational Cell Environment (CCE) addresses each of these

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 540–547, 2004.
© Springer-Verlag Berlin Heidelberg 2004

needs, using advanced and emerging technologies to develop a flexible, problem-solving environment (PSE) for systems biology.

CCE is a scientific PSE. Scientific PSEs are complex computing systems that integrate the activities necessary to accomplish high-level domain tasks [1] [2]. Just like most other PSEs, the goal of CCE is to provide access to data stores, data pedigree tracking, analysis tools, domain-specific workflow support, a user environment, experiment management, and data transformation and filtering capabilities. CCE is complex because of the goal to access data from multiple distributed heterogeneous data stores while analyzing those data via multiple distributed heterogeneous applications. CCE is being developed as a laboratory-directed research and development project for the Biomolecular Sciences Initiative at the Pacific Northwest National Laboratory (PNNL).

2 SysBioPSE Prototype

The spectrum of biological study covers DNA, genes, RNA, proteins, phosphorylation, intra-cellular and extra-cellular networks, tissue, organs, and community interaction. Any individual researcher will likely focus on one of these areas but must draw on research or experimental results from a different area. For instance, a researcher studying intracellular networks must draw on the experimental results from gene expression and/or protein abundance. Once the network is defined, the researcher will likely bring in published research on the proteins of interest. Cross-cutting software analysis tools that can link multiple domains of biological research exist only on a limited scale. CCE has been developed to help link the multiple disciplines within the field of biology.

One of CCE's initial goals was to derive a vision of how biologists and bioinformaticists would conduct computational and collaborative research through a scientific PSE. A horizontal user-interface prototype called SysBioPSE (Biology PSE) was developed to convey the user-centered features and capabilities of an ideal PSE for biological sciences. SysBioPSE was driven directly from the requirements and design criteria emerging from interactions with biologists and bioinformaticists.

User requirements were gathered using a human computer interaction technique known as participatory analysis [3] [4]. Participatory analysis strives to understand the tasks and activities of a user by including the user in the process. A series of interviews with different classes of biologists, including biochemists, molecular biologists, cell biologists, and bioinformaticists, were conducted. We worked with the scientists to understand their work processes, methodology of performing research, the resources that they use, the mechanisms in which they use data and analysis tools to hypothesize, and the way in which they collaborate with each other. Requirements for an overall system were developed from this process.

Based on user requirements, SysBioPSE (Fig. 1) was developed, using the software design tool Director from Macromedia, which was then evaluated and refined across various biological domains. The key feature described in SysBioPSE is the capability to access and integrate various kinds of computational resources such as data sets, applications, and people. An important objective of SysBioPSE is to allow biologists to represent and organize their computational resources in ways that are logical and meaningful to them. To this end, SysBioPSE provides three different

mechanisms or paradigms for working with computational resources. First, a project view allows biologists to define related sets of applications, datasets, and team members and to organize these within specific workspaces called "workbenches." Second, a concept view allows biologists to relate resources to specific theoretical or experimental

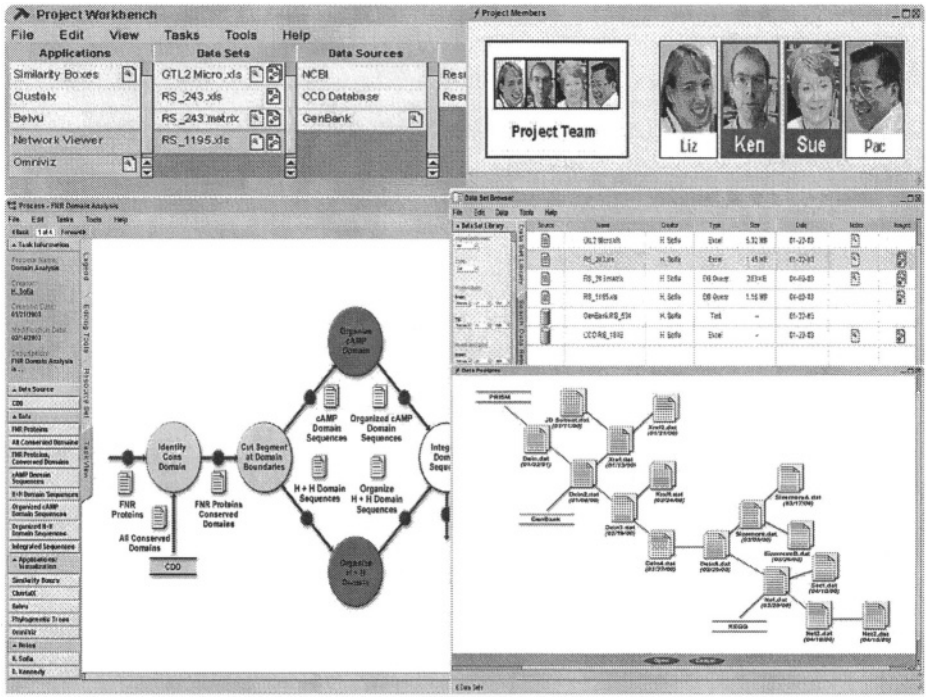


Fig. 1. SysBioPSE Prototype

concepts. In this view, biologists sketch concept diagrams in a free-form manner and link resources directly to visual concept representations. Third, a process view allows biologists to attach resources to specific steps of an experiment or research process. The three paradigms enforce the idea that computational resources are applied by biologists, not in an isolated manner, but in very specific scientific contexts. SysBioPSE captures these scientific contexts such that biologists may manage and apply computational resources in ways that are consistent with their scientific conceptions and perceptions.

3 Distributed Data Management and Analysis Subsystem

In addition to SysBioPSE, the CCE project also developed a functional subsystem aimed at implementing a narrow set of capabilities from the much larger SysBioPSE vision that may be immediately applied by biologists in their research endeavors. The

Distributed Data Management and Analysis (DDMA) system was designed to allow biologists to extract data from various external scientific databases and combine them into an integrated dataset. The development of this system required the CCE project to focus heavily on system architecture and implementation strategies and issues.

3.1 System Architecture

DDMA was developed as a client-server application based on the CCE architecture. The CCE architecture supports independently designed and developed components combined to deliver scalable solutions and reach a large community of researchers from many biology-related scientific domains. Accomplishing this goal requires that the components be designed for loose coupling and operation in a heterogeneous and rapidly evolving service-based environment. To achieve this, an open data-management architecture is devised.

An overall extensible environment is facilitated by defining a multi-tiered, structured architecture that provides a component-based plug-in capability for data resources, collaborative technologies, and analytic tools. Figure 2 shows the CCE architecture. This architecture is designed such that a few basic components are built and deployed initially to a user base, while new functionality is continually added through the development of new components.

The current system provides access to the proteomic data produced at PNNL stored in a system called the Proteomics Research Information Storage and Management (PRISM) System [5] as well as data from the National Center for Biotechnology Information (NCBI)[6], the Kyoto Encyclopedia of Genes and Genomes (KEGG)[7], microarray data, and delimited data. CCE provides easy access to several external data-analysis tools, including PubMed[8], the Conserved Domain Architecture Retrieval Tool (CDart)[9], and the Basic Local Alignment Search Tool (BLAST)[10], all from NCBI; and Cytoscape[11] from the Institute for Systems Biology, as well as analysis tools that are being developed internally at PNNL. Work is in progress to provide customized access to data sources and applications enabling data access from any data source and data analysis from any application.

CCE is implemented as a three-tier architecture system and is based on the Java 2 Enterprise Edition (J2EE) architecture [12]. J2EE is a platform that manages infrastructure and supports web services to enable the development of secure, robust, and interoperable applications. Several of the reasons for adopting a J2EE architecture include fast searching and querying, a better component architecture implemented through JavaBeans (an industry standard), better and more robust security features, transaction management, and most notably, a production quality system.

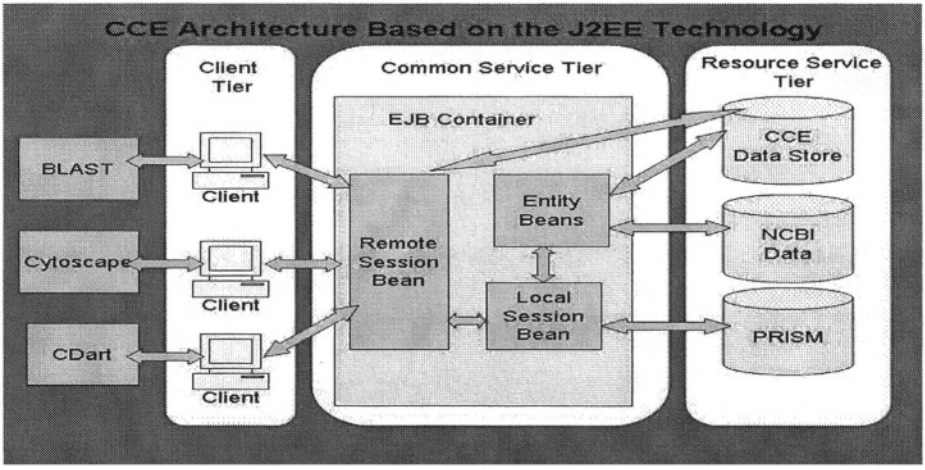


Fig. 2. CCE Architecture

3.2 User Interface

The DDMA user interface (Fig. 3) is centered on a project hierarchy containing projects and the datasets based on authentication. Navigation functions are provided to help users traverse large hierarchies. Once a user has identified the dataset of interest, it can be selected for viewing and analysis. When creating a new dataset, the user has the option of selecting an interface to one of the CCE-supported data sources, such PRISM[5], KEGG[7], and NCBI[6], or creating his/her own custom connection to a data source. Micro-array data as well as data from delimited files can be imported. The connection to the data source is seamless. The user can export the data in spreadsheet format. The system has the capability to handle very large datasets without long delays through a paging technique that is also seamless to the user. Also, large datasets can be navigated seamlessly through a visual panning technique. DDMA also incorporates a flexible, user-driven merge technology to integrate data from different data sources.

4 Summary and Future Work

SysBioPSE and the DDMA system comprise an effective, two-prong strategy for researching and developing PSEs for biological sciences. SysBioPSE provides an overall vision and high-level design of a powerful scientific user environment, while DDMA provides a functional tool that biologists may immediately apply to satisfy the critical data-integration needs of their current research. Through these two systems, CCE addresses four key user requirements:

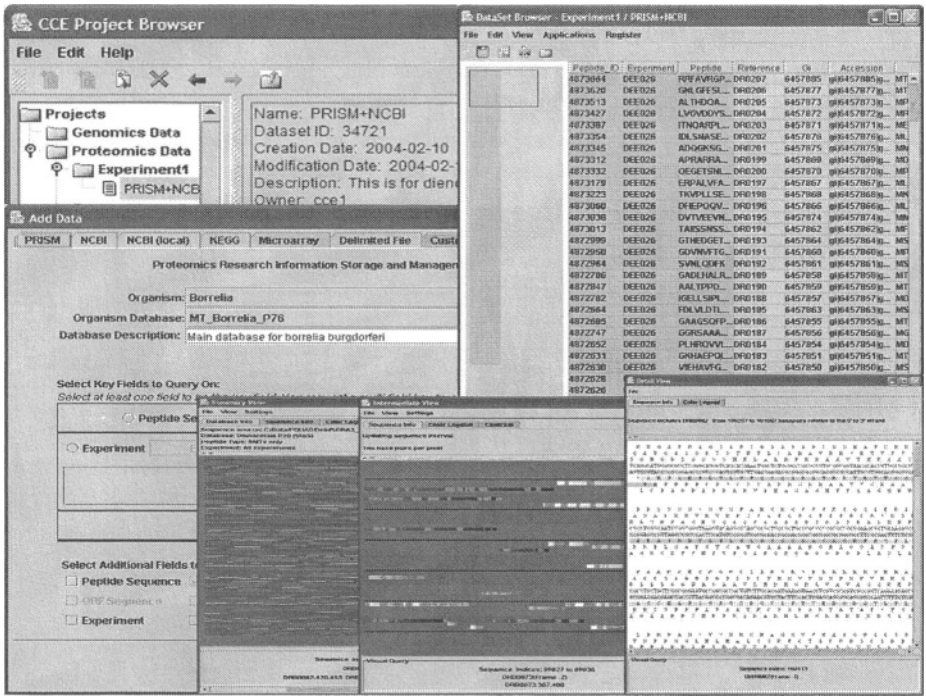


Fig. 3. DDMA User Interface

1) Data Storage and Retrieval: This supports access to terabytes of genomic and proteomic data produced by experimental and computational biologists. The linking and management of distributed, heterogeneous data sources is transparent to users. The data-management system works with and conforms to evolving and *de facto* standards within the biology community. In addition, the data-management system is responsible for storing and retrieving analysis data, images, annotations, tools, and reference information. Subscription capabilities allow users to securely place and retrieve data into and out of the data store. Software security mechanisms are deployed to verify the authentication of users. In the future, CCE will be extended to manage permissions and privileges to the data of users and groups. Encryption capabilities will be added to the system to ensure the secure transmission of data. A mechanism to plug-in data-conversion capabilities and tools, which allow researchers to apply and exchange data in standard scientific data formats, will be provided.

2) Data and Information Analysis: These capabilities include access to a suite of standard data-analysis and visualization tools. Access to these analytical tools allows researchers to examine, investigate, and navigate data to identify and analyze portions that are relevant to their specific research needs. Data-integration capabilities allow researchers to compare and merge different kinds of data, such as proteomic and micro-array data. A flexible environment allows researchers to work with and link combinations of analysis and visualization tools in useful and novel ways.

3) *Common User Interface:* This integrates and delivers data and information management, analysis, and sharing capabilities. The community data portal provides user-interaction layers on top of the various data-management capabilities, providing, at the highest level, interaction with the data-management capabilities through a common user interface. Through the use of profiles, the user interface and its access to data-management functions are customizable to address the specific demands of individuals.

4) Collaborative Data and Information Sharing: This is largely supported in CCE by common access to data, tools, and other resources. However, in the future, CCE will provide facilities beyond shared access to share data in the context of collaborative experimentation and analysis. Through real-time collaboration tools, such as audio/video conferencing, text chat, electronic whiteboards, and application sharing tools, remote researchers may spontaneously collaborate on the application and analysis of data. Through asynchronous, records-based mechanisms, such as mailing lists, bulletin boards, and electronic laboratory notebooks, researchers may collaborate on data over extended periods of time, involve larger numbers of researchers, and maintain a running history of the collaboration and analysis.

5 Conclusions

Today's scientific problems are complex and can benefit from a multidisciplinary environment that can assimilate remarkably diverse data repositories and computational bioinformatics tools that integrate across multiple domains. CCE has proven the possibility of accessing multiple, distributed heterogeneous data stores through one unifying interface. This foundation for a biological PSE will become more comprehensive with time. Moreover, the CCE architecture could easily be adopted in new PSEs supporting other domains.

The goal of CCE is very broad and shall realistically take many years to accomplish. The system has been designed initially to access a subset of the data and tools of interest and is targeted to a small user group. Over time, CCE will expand to include new user groups that will define the additional sets of data and analysis tools to be added to the system. Today, CCE, through DDMA, provides a single visual environment, portions of the data pedigree tracking, tools for data collection and basic data fusion, and several bioinformatics tools for data analysis, generally focused in the area of protein family and protein complex research.

The CCE is a PSE that will not only provide biologists and bioinformaticists greater ability and increased flexibility in manipulating, collecting and analyzing their data, and solving their problems, but will also transform the way this research is carried out and in turn enhance the science of systems biology.

References

1. Gallopoulos SE, Houstis, and JR Rice. 1994. "Problem-Solving Environments for Computational Science."*IEEE Computational Science and Engineering,* Summer: 11-23.
2. Rice JR, and RF Boisvert. 1996. "From Scientific Software Libraries to Problem-Solving Environments."*IEEE Computational Science & Engineering,* Fall: 44-53.
3. Chin G Jr., KL Schuchardt, JD Myers, and DK Gracio. 2000. "Participatory Workflow Analysis: Unveiling Scientific Research Processes with Physical Scientists." In Proceedings of Participatory Design Conference (PDC 2000), pp 30-39. Nov. 28-Dec. 1, New York, NY. CPSR, Palo Alto, CA.
4. Schuler D, and Namioka (eds). 1993. *Participatory Design: Principles and Practices.* Erlbaum Associates, Hillsdale, NJ.
5. PRISM: `http://www.sysbio.org/technologies/comp.stm`
6. NCBI: `http://www.ncbi.nlm.nih.gov/`
7. KEGG: `http://www.genome.ad.jp/kegg/`
8. PubMed: `http://www.ncbi.nlm.nih.gov/PubMed/`
9. `http://www.ncbi.nlm.nih.gov/Structure/lexington.cgi?amd=cdd&uid=16610`
10. BLAST:`http://www.ncbi.nlm.nih.gov/BLAST/`
11. Cytoscape: `http://www.cytoscape.org/`
12. Shannon BM, V Hapner, E Matena, Pelegri-Llopart, JS Davidson, and L Cable. 2000. *Java™ 2 Platform, Enterprise Edition: Platform and Component Specifications.* Addison-Wesley Professional. Available URL: http://java.sun.com/j2se

Depth Recovery with an Area Based Version of the Stereo Matching Method with Scale-Space Tensor Representation of Local Neighborhoods

Bogusław Cyganek

University of Science and Technology
Department of Electronics
Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@uci.agh.edu.pl

Abstract. Depth recovery is one of the classical problems of computer vision. Many methods have been developed that address this issue. However, each of them has specific behavior depending on the image contents, acquisition system, computation time, etc. The variety of methods is due to the non uniqueness of the matching process involved in depth recovery from a pair of images of a given scene. The presented method relies on tensor representation of local structures in images. This representation allows for extraction of the local phase, signal intensity and coherence measure of a local neighborhood in different scales. This operator showed to be very useful in detection of locally oriented structures and corners in single digital images – presented at the previous ICCS. In this paper, the tensor operator is used for area-based matching of stereo images. Due to tensor transformation of the input images the more reliable matching was achieved. The method was tested with many different classes of stereo images. In this paper we present and discuss the experimental results and also details of implementation.

1 Introduction

The presented stereo method is an area-based version (dense disparity maps) of the feature-based method presented in [4]. The method relies on tensor representation of local structures in the input images. The tensor operator was already used for detection of local structures and corners in *single* images [3]. Such tensor representation provides the local phase, signal intensity and coherence measure of a local neighborhood in different scales. Thus, this representation allows for more reliable matching than other simple area-based methods. Furthermore, with the tensor method the first step of segmentation into feature and featureless areas is obtained automatically. In effect, based on at least two images of a scene, the method makes possible segmentation of the nearest objects from the background. The presented method was tested in the navigation system for the vision impaired persons [5]. It allowed for direct localization of the nearest objects from the observed scene. The implementation with the novel winner-update matching [2] allowed for speed improvement of the version from [5].

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 548–551, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Tensor Detector of Local Structures in Pixel Neighborhoods

A detailed derivation of the tensor detector of local structures, as well as its practical implementation, were already published at the previous ICCS [3]. In this paper then we present only the final formulas. The detector of local structures in images takes the following vector form:

$$\mathbf{s} = \begin{bmatrix} T_{xx} + T_{yy} \\ \angle \mathbf{w} \\ c \end{bmatrix}, \quad \text{where} \quad \angle w = \begin{cases} \arctan\left(\dfrac{2T_{xy}}{T_{xx} - T_{yy}}\right) & , \; T_{xx} \neq T_{yy} \\ \dfrac{\pi}{2} & , \; T_{xx} = T_{yy} \;\wedge\; T_{xy} \geq 0 \\ -\dfrac{\pi}{2} & , \; T_{xx} = T_{yy} \;\wedge\; T_{xy} < 0 \end{cases} \tag{1}$$

The third component c of (1) constitutes a coherency measure [10]:

$$c = \begin{cases} \left(\dfrac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2 & , \; \mathrm{Tr}(\mathbf{T}) \neq 0 \\ 0 & , \; \mathrm{Tr}(\mathbf{T}) = 0 \end{cases}. \tag{2}$$

The structural tensor T, its eigenvalues $\lambda_{1,2}$, as well as its components $\mathbf{T}_{xx}$, $\mathbf{T}_{xy}$, and $\mathbf{T}_{yy}$ are defined in [3].

3 Dense Stereo with the Fast Block Matching

Based on the properties of the structural tensor operator, the evidence measure for our stereo method was defined as follows:

$$E(x,y,d) = \begin{cases} \alpha\left|I_l(x,y) - I_r(x+d_x, y+d_y)\right| + & \\ +\beta\left|s_{1l}(x,y) - s_{1r}(x+d_x, y+d_y)\right| + & \Psi(x,y,d) \neq 0 \\ +\gamma\left|s_{2l}(x,y) - s_{2l}(x+d_x, y+d_y)\right| & \\ \kappa & \Psi(x,y,d) = 0 \end{cases} \tag{3}$$

where α, β, γ are weight coefficients, $d=[d_x,d_y]$ denotes disparity (for canonical stereo setup $d_y=0$), $I_i(x,y)$ stands for intensity value in the *i-th* image at point (x,y), $mag_i(x,y)$, $angle_i(x,y)$ and $coh_i(x,y)$ are coefficients of the structural tensor, $\Psi(x,y,d)$ is a binary operator classifying pixels as belonging to an area with local structures (value *1*) and vice versa (value 0); k is a special value to denote those areas in images. Details of the function $\Psi(x,y,d)$ are provided in [4].

The canonical setup was assumed, thus the epipolar lines were known beforehand. Averaging of the evidence measure was done with a moving average filter in a rectangular window – usually a square from 3×3 to 17×17 pixels wide.

4 Experimental Results

The method was implemented in C++ and all the experiments were performed on the PC with Pentium l.8GHz. The winner-update strategy [2] was used for matching.

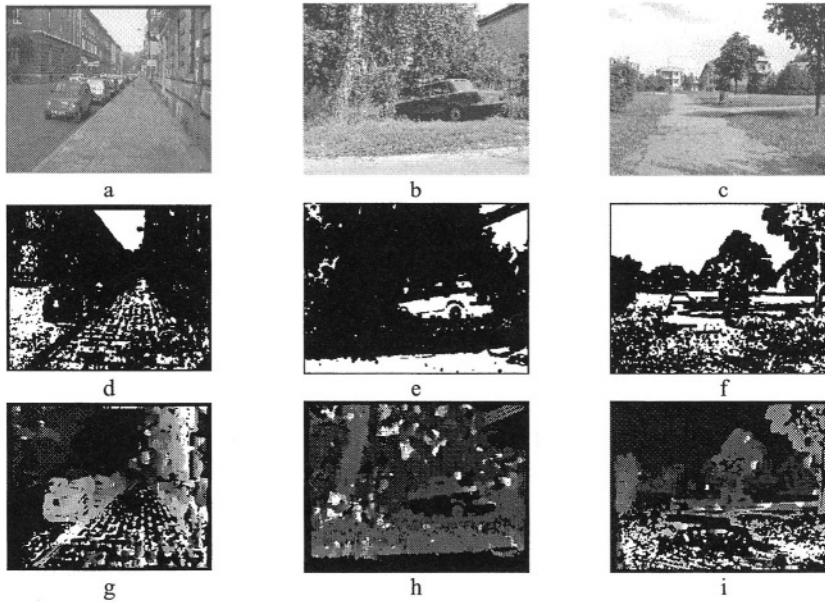


Fig. 1. Stereo matching of real images: the original left images (a,b,c), regions classified as textureless (in white) (d,e,f), disparity maps (g,h,i)

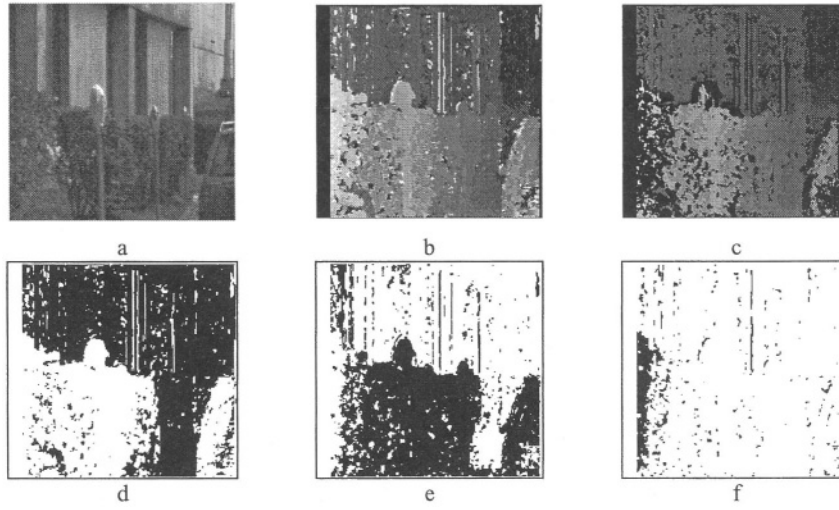


Fig. 2. Depth recovery and segmentation: left image (a), the disparity map with 3×3 matching windows (b), the disparity map after cross-checking (c), segmentation from disparity – background objects (in black) (d), middle objects (e), the closest objects (f)

Fig. 1(a,b,c) presents three left images of the real stereo pairs. The detected textureless areas – in white – are presented in Fig. 1(d,e,f). This level of segmentation is essential to obtain the disparity maps with a good quality – Fig. 1(g,h,i).

Fig. 2 presents experiments with the "Parkmeter" (a). The two disparity maps, obtained with the 3×3 matching windows, are presented in Fig. 2b and Fig. 2c. The latter is validated by a cross-checking. Fig 2.(d,e,f) contain objects (in black) detected by segmentation: background objects (d), middle range objects (e), the closest (f). Original image size is 256×240 pixels. Execution times for disparity maps are as follows: 1.54 s for Fig. 2b, and 3.18 for Fig. 2c.

5 Conclusions

In this paper the depth recovery method from multiple views of the same scene was presented. The dense disparity map was computed by the stereo method that relies on the tensor representation of local neighborhoods of pixels. This representation has the plausible feature of partitioning the input images into areas with sufficiently strong structures, at the same time rejecting featureless areas that cannot be reliably matched. Stereo matching is performed in the domain of the structural tensors by means of the L_1 measure which is computationally efficient. Computation time was also reduced at this stage by application of the winner-update technique for matching. The experimental results performed with different real images showed their robustness, especially in avoiding textureless areas.

References

1. Brown, M.Z., Burschka D., Hager G.D.: Advances in Computational Stereo. IEEE Trans. PAMI, Vol.25, No. 8, (2003) 993-1008
2. Chen, Y-S., Hung, Y-P., Fuh, C-S.: Fast Block Matching Algorithm Based on the Winner-Update Strategy. IEEE Trans. On Image Processing, Vol. 10, No. 8, (2001) 1212-1222
3. Cyganek, B.: Combined Detector of Locally-Oriented Structures and Corners in Images Based on a Scale-Space Tensor Representation of Local Neighborhoods of Pixels, in LNCS 2658, Proceedings of the ICCS 2003 (2003) 721-730
4. Cyganek, B.: Novel Stereo Matching Method That Employs Tensor Representation of Local Neighborhood in Images, Machine Graphics & Vision, 10/3 (2001) 289-316
5. Cyganek, B., Borgosz, J.: Computer Platform for Transformation of Visual Information into Sound Sensations for Vision Impaired Persons, in LNCS 2626 (2003) 182–191
6. Egnal, G., Wildes R.P.: Detecting Binocular Half-Occlusions: Empirical Comparisons of Five Approaches. IEEE Trans. PAMI, Vol. 24, No. 8, (2002) 1127-1132
7. Fua P.: A Parallel Stereo Algorithm that Produces Dense Depth Maps and Preserves Image Features, INRIA Technical Report No 1369 (1991)
8. Hartley, R.I., Zisserman A.: Multiple View Geometry in Computer Vision. CUP (2000)
9. Hauβecker, H., Jähne, B.: A Tensor Approach for Local Structure Analysis in Multi-Dimensional Images. Interdisciplinary Center for Scientific Computing, University of Heidelberg, (1998)
10. Jähne, B.: Digital Image Processing. 4th edition, Springer-Verlag, (1997)

Symbolic Calculation for Frölicher-Nijenhuis $\mathbb{R}$-Algebra for Exploring in Electromagnetic Field Theory* **

José de Jesús Cruz Guzmán and Zbigniew Oziewicz* * *

Universidad Nacional Autónoma de México,
Facultad de Estudios Superiores Cuautitlán,
Apartado Postal 25, Cuautitlán Izcalli, Edo. México C.P. 54714
{cruz, oziewicz}@servidor.unam.mx

Abstract. The principal aim of this work is the presentation of a symbolic calculation computer analysis for exploring electromagnetic fields for not inertial observer. Based on Frölicher-Nijenhuis super-Lie $\mathbb{R}$-algebra, we developed a learning environment for axiomatic classical electromagnetics and electrodynamics. A collection of programs developed on Mathematical programming environment has been builded for the $\mathbb{N}$-graded Graßmann Algebra, $\mathbb{Z}$-graded endomorphisms and graded commutators.

1 Introduction

In axiomatic classical electromagnetic is postulated that the density pseudo differential form J, grade $J = d - 1 \in \mathbb{N}$ and the strength or 'magnetic flux' F, grade $F = 2$ are absolute, are observer-free.

– Charge current conservation

$$\int_{\partial \mathrm{V}} J = 0 \quad \in \mathbb{R} \tag{1}$$

– Conservative force field interaction

$$\int_{\partial C} F = 0 \quad \in \mathbb{R}. \tag{2}$$

* Supported by el Consejo Nacional de Ciencia y Tecnología (CONACyT) de México, proyecto # U41214-F, and by UNAM, DGAPA, Programa de Apoyo a Proyectos de Investigacion e Innovacion Tecnologica, proyecto # IN 105402.

** Submitted to Lecture Notes in Computer Science: International Conference on Computational Science ICCS'04, June 2004, Krakow, Poland. Not yet in final form

*** Zbigniew Oziewicz is a member of Sistema Nacional de Investigadores, México, No de expediente 15337.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 552–556, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The first axioms is the conservation of the density pseudo-form of the Charge-Current. The second axiom is interpretd as the conservation of the work (on small test charge-current) in the space-time. See for complementary references and formal approach Cruz & Oziewicz [1, 2003].

Starting with such axiomatic approach we use the Frölicher and Nijenhuis[2, 3] Lie $\mathbb{R}$-operation to derive observer-dependent form of the four Maxwell equations.

In Section 2 we present some necessary definitions. Section 3 is devoted to present the principal lines of the algorithm. Finally in Section 4 we present conclusions and objectives for future work.

2 Basic Definitions

Let $\mathcal{F}$ be an $\mathbb{R}$-algebra of scalar fields on Space-Time. The $\mathcal{F}$-modul of the differential 1-forms is denoted by M.

Definition 1 (Graßmann algebra of differential forms). *The $\mathbb{N}$-graded Graßmann $\mathcal{F}$-algebra of differential forms is denoted by:*

$$M^\wedge \equiv \oplus M^{\wedge i} = M^{\wedge 0} \oplus M^{\wedge 1} \oplus M^{\wedge 2} + \cdots, \quad M^{\wedge 0} \equiv \mathcal{F}, \quad M^{\wedge 1} \equiv M. \tag{3}$$

Definition 2 (Observer field). *Is a not necessarily integrable product structure that splits $\mathcal{F}$-modules of the differential one-forms M and one-vector fields, M^*, into 'time-' and 'space-' $\mathcal{F}$-submodules. An idempotent $a^2 = a \in \mathrm{der}(M^\wedge)$ is said to be observer if tr $a = 1$, and dim im a = dim of time = 1.*

In the present paper we represent an observer by κ. A1-dimensional $\mathcal{F}$-modul $\mathrm{im}\kappa$, is an ideal in the Graßmann $\mathcal{F}$-algebra, and the projector (id $-\kappa$), is an $\mathcal{F}$-algebra map

$$(\mathrm{id} - \kappa) \in \mathrm{alg}_{\mathcal{F}}\left(M^\wedge, M^\wedge/\mathrm{im}\kappa\right). \tag{4}$$

Definition 3 (Derivations of Graßmann algebra). *Let $|\mathcal{A}| \equiv \mathrm{grade}\,\mathcal{A} \in \mathbb{Z}$. A map $\mathcal{A} : M^\wedge \to M^\wedge$ is said to be a* graded *$\mathcal{F}$-derivation, $\mathcal{A} \in der(M^\wedge)$, if Leibniz axiom holds:*

$$\mathcal{A}(\alpha \wedge \beta) = (\mathcal{A}\alpha) \wedge \beta + (-)^{|\mathcal{A}||\alpha|}\alpha \wedge \mathcal{A}\beta. \quad \forall\, \alpha, \beta \in M^\wedge \tag{5}$$

Example: every observer is $\mathcal{F}$-derivation, $a \in \mathrm{der}_{\mathcal{F}}(M^\wedge)$, $\mathrm{grade}\, a = 0 \in \mathbb{N}$.

Definition 4 (Lie Super ($\equiv \mathbb{Z}_2$-graded) $\mathcal{F}$-algebra of derivations). The graded commutator (bracket) *is:*

$$\begin{aligned}
(End\, M^\wedge) \otimes (End\, M^\wedge) &\xrightarrow[super\text{-}bracket]{\{\ ,\ \}} (End\, M^\wedge), \\
\{\mathcal{A}, \mathcal{B}\} &\equiv \mathcal{A} \circ \mathcal{B} - (-)^{|\mathcal{A}||\mathcal{B}|}\mathcal{B} \circ \mathcal{A} \in \mathrm{der}(M^\wedge), \\
\mathrm{grade}\{\mathcal{A}, \mathcal{B}\} &= \mathrm{grade}\,\mathcal{A} + \mathrm{grade}\,\mathcal{B}.
\end{aligned}$$

For the graded commutator the graded version of the Jacobi identity holds,

$$\{A,\{B,C\}\} = \{\{A,B\},C\} + (-)^{AB}\{B,\{A,C\}\}. \tag{6}$$

if $\mathcal{A}$ and $\mathcal{B}$ be graded derivations; $\mathcal{A},\mathcal{B} \in \mathrm{der}(M^\wedge)$. Then $\{\mathcal{A},\mathcal{B}\} \in \mathrm{der}(M^\wedge)$.

Definition 5 (Ślebodziński Lie derivation). *Let* $\mathcal{A} \in \mathrm{der}(M^\wedge)$ *be any derivation. The* $\mathbb{R}$*-derivation introduced in 1931 [4] is:*

$$\mathcal{L}_\mathcal{A} \equiv \{\mathcal{A}, d\} \in \mathrm{der}_\mathbb{R}(M^\wedge). \tag{7}$$

Definition 6 (Frölicher and Nijenhuis $\mathbb{R}$-algebra). *A derivation* $[\kappa,\rho]_{FN} \in \mathrm{der}_\mathcal{F}(M^\wedge)$ *exit, such that*

$$\begin{aligned} \mathcal{L}_{[\kappa,\rho]_{FN}} &\equiv \{\mathcal{L}_\kappa, \mathcal{L}_\rho\} \in \mathrm{der}_\mathbb{R}(M^\wedge), \\ [\kappa,\rho]_{FN} &= (-1)^{\kappa+\rho+\kappa\rho}\cdot[\rho,\kappa]_{FN}. \end{aligned} \tag{8}$$

The Frölicher-Nijenhuis [1956] binary operation on the Lie $M^\wedge$ *-module* $\mathrm{der}_\mathcal{F}(M^\wedge)$, *denoted by* $[\cdot,\cdot]_{FN} \in \mathrm{der}_\mathcal{F}(M^\wedge)$, *with* $\mathrm{grade}[\cdot,\cdot] = +1$, *is an example of the Gerstenhaber* $\mathbb{R}$*-algebra.*

An arbitrary derivation $D \in \mathrm{der}_\mathbb{R}(M^\wedge)$ possess the following unique decomposition

$$D = (\mathcal{L}\circ i + i\circ\mathcal{L})D = \{i_D, d\} + i_{\{D,d\}}. \tag{9}$$

The generalization of the Frölicher and Nijenhuis decomposition for an extension of DGA, was given in [Oziewicz 1991].

3 Outline of Algorithm

The builded functions contains:

- Appropriate data types designed for: $\mathbb{N}$-graded $\mathcal{F}$-algebra of the differential forms $M^\wedge$. $\mathbb{Z}$-homogeneous graded endomorphism End $(M^\wedge)$, Poisson graded commutator (bracket) $[\ ,\ ]_{FN}$, etc.
- Constructors for graded $\mathcal{F}$- and $\mathbb{R}$-derivations, Lie $\mathcal{F}$- and $\mathbb{R}$-derivation, observers $\in$ der, and creations and annihilation operators $\in$ End. Every annihilation opertator is a $\mathcal{F}$-derivation.
- Procedures for Graßmann multiplication $\wedge$, composition $\circ$ of graded endomorphisms and Jacobi identity for Lie and Poisson $\mathbb{R}$-brackets.

A grammar definition is necessary in the way that the abstract representation be in accordance with mathematical formal definition of objets. By example: for the $\mathbb{N}$-graded Graßmann $\mathcal{F}$-algebra of differential forms $M^\wedge$we have:

- $M^\wedge$ is generated by $\mathcal{F}$ and M ie $M^\wedge = \mathrm{gen}\{\mathcal{F}, M\}$.
- If $\alpha,\beta \in M^\wedge$ then $\alpha+\beta \in M^\wedge$ and $\alpha\wedge\beta \in M^\wedge$.

- $\text{grade}(\alpha \wedge \beta) = \text{grade}\,\alpha + \text{grade}\,\beta + \text{grade}\,\wedge, \text{grade}\,\wedge = 0 \in \mathbb{N}$.
- $\forall \alpha \in M \Rightarrow \text{grade}\,\alpha = 1$ and $\alpha \wedge \alpha = 0$.
- The creation Graßmann operators is defined by: $e_\alpha \beta \equiv \alpha \wedge \beta$, $e_\alpha \in \text{End}\,(M^\wedge)$.
- The 'inner' or 'interior' action 'i' of $M^\wedge$ on $M^{*\wedge}$ is said to be the annihilation operator $i_M \in \text{der}_{\mathcal{F}}(M^\wedge)$.

In practical computation operators e and i play an important role, e is a data type constructor and i is a data type selector:

$$M^\wedge \otimes M^\wedge \xrightarrow{\ e\ } M^\wedge$$

$$M^* \otimes M^\wedge \xrightarrow{\ i\ } M^\wedge.$$

For derivations operator we need the set of symbols: {name, grade,+}. With those and the Leibniz axiom we build the necessary expressions for the derivation operation, and for graded commutator we need {name, grade,+, ∘}.

For Frölicher-Nijenhuis binary operation we use the explicit form of the bracket: Let $\rho, \kappa \in \text{der}$ and $\delta_{\kappa \circ \rho} \in \text{der}$

$$[\rho, \kappa]_{FN} \equiv -\{\kappa, \mathcal{L}_\rho\} - \{\delta_{\kappa \circ \rho}, d\} \quad \in \text{der}_{\mathbb{R}} \tag{10}$$

Here we use a $\mathcal{F}$-modul map $p \in \text{hom}_{\mathcal{F}}(M, M^\wedge)$ lifted to the unique ($\mathbb{Z}_2$-graded) $\mathcal{F}$-derivation

$$\text{hom}_{\mathcal{F}}(M, M^\wedge) \simeq M^\wedge \otimes_{\mathcal{F}} M^* \ni p \mapsto \delta_p \in \text{der}_{\mathcal{F}}(M^\wedge),$$

with $\text{grade}(\delta) = 0$, such that $\delta_p | \mathcal{F} = 0$ and $\delta_p | M = p$.

4 Conclusions and Future Work

The axiomatic approach could possess pedagogical advantage in teaching the fundamentals of electromagnetic laws as explicitly observer-dependent and gives a power tool in the exploration of physical laws. Symbolic calculus offer an appropriate tool in the exploration of abstract algebra and its applications. Future work will be devoted to

- Get procedures for splitting differential forms and differential equation in Space Time,
- automatize algorithmically all expressions of four Maxwell equations for non inertial observers
- build higher order words of Lie Poisson brackets for Lie derivation of observer operators $\{k, d\}$ and find all identities inside the Lie algebra $\text{gen}\{d, k_1, k_2\}$.

References

1. Cruz J. and Oziewicz Z., *Frölicher-Nijenhuis Algebra and Non-Inertial Deformation of Maxwell Equations,* Bulletin de La Société des Sciences Et Des Lettres de Łódź, Vol LIII, (2003), 107-140, ISSN 0459-6854.

2. Frölicher A. and Nijenhuis A., *Theory of vector valued differential forms,* Part I, Indagationes Mathematicae **18**(1956), 338–359.
3. Oziewicz Z., Frölicher and Nijenhuis operations for noncommutative rings, Proceedings of the XXth International Conference on Differential Geometric Methods in Theoretical Physics, edited by S. Catto and A. Rocha (World Sci. Pub. Comp., Singapore, 1992), 584-593.
4. Ślebodziński W., Bulletin de l'Académie Royale Belgique **17** (5)(1931), 864–870.

Spherical Orthogonal Polynomials and Symbolic-Numeric Gaussian Cubature Formulas

Annie Cuyt[1], Brahim Benouahmane[2], and Brigitte Verdonk[1]

[1] Dept of Mathematics and Computer Science, University of Antwerp
Middelheimlaan 1, B2020 Antwerp, Belgium
[2] Faculté des Sciences et Techniques, Université Hassan II
BP 146 Mohammadia, 20650 Morocco

Abstract. It is well-known that the classical univariate orthogonal polynomials give rise to highly efficient Gaussian quadrature rules. We show how the classical orthogonal polynomials can be generalized to a multivariate setting and how this generalization leads to Gaussian cubature rules for specific families of multivariate polynomials.
The multivariate homogeneous orthogonal functions that we discuss here satisfy a unique slice projection property: they project to univariate orthogonal polynomials on every one-dimensional subspace spanned by a vector from the unit hypersphere. We therefore call them spherical orthogonal polynomials.

1 Spherical Orthogonal Polynomials

The orthogonal polynomials under discussion were first introduced in [1] in a different form and later in [3] in the current form. Originally they were not termed spherical orthogonal polynomials because of a lack of insight into the mechanism behind the definition.

In dealing with multivariate polynomials and functions we shall often switch between the cartesian and the spherical coordinate system. The cartesian coordinates $X = (x_1, \ldots, x_n) \in \mathbb{R}^n$ are then replaced by $X = (x_1, \ldots, x_n) = (\xi_1 z, \ldots, \xi_n z)$ with $\xi_k, z \in \mathbb{R}$ where the directional vector $\xi = (\xi_1, \ldots, \xi_n)$ belongs to the unit sphere $S_n = \{\xi : ||\xi||_p = 1\}$. Here $||\cdot||_p$ denotes one of the usual Minkowski norms. While ξ contains the directional information of X, the variable z contains the (possibly signed) distance information. With the multi-index $\kappa = (\kappa_1, \ldots, \kappa_n) \in \mathbb{N}^n$ the notations $X^\kappa, \kappa!$ and $|\kappa|$ respectively denote

$$X^\kappa = x_1^{\kappa_1} \ldots x_n^{\kappa_n} \qquad \kappa! = \kappa_1! \ldots \kappa_n! \qquad |\kappa| = \kappa_1 + \ldots + \kappa_n$$

Since z can be positive as well as negative and hence two directional vectors can generate X, we also introduce a signed distance function

$$\mathrm{sd}(X) = \mathrm{sgn}(x_1)||X||_p$$

For the sequel of the discussion we need some more notation. We denote by $\mathbb{R}[z]$ the linear space of polynomials in the variable z with real coefficients, by

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 557–560, 2004.
© Springer-Verlag Berlin Heidelberg 2004

$\mathbb{R}[\xi] = \mathbb{R}[\xi_1, \ldots, \xi_n]$ the linear space of n-variate polynomials in ξ_k with real coefficients, by $\mathbb{R}(\xi) = \mathbb{R}(\xi_1, \ldots, \xi_n)$ the commutative field of rational functions in ξ_k and with real coefficients, by $\mathbb{R}(\xi)[z]$ the linear space of polynomials in the variable z with coefficients from $\mathbb{R}(\xi)$ and by $\mathbb{R}[\xi][z]$ the linear space of polynomials in the variable z with coefficients from $\mathbb{R}[\xi]$.

Let us introduce the linear functional Γ acting on the variable z as $\Gamma(z^i) = c_i(\xi)$, where $c_i(\xi)$ is a homogeneous expression of degree i in the ξ_k. For our purpose we take

$$c_i(\xi) = \sum_{|\kappa|=i} c_\kappa \xi^\kappa \qquad c_\kappa = \frac{|\kappa|!}{\kappa!} \int \ldots \int_{||X||_p \le 1} w\,(||X||_p)\, X^\kappa \, dX \tag{1}$$

$$\Gamma(z^i) = \int \ldots \int_{||X||_p \le 1} w\,(||X||_p) \left(\sum_{k=1}^{n} x_k \xi_k \right)^i dX \qquad dX = dx_1 \ldots dx_n \tag{2}$$

The n-variate polynomials under investigation are of the form

$$V_m(X) = \mathcal{V}_m(z) = \sum_{i=0}^{m} B_{m^2-i}(\xi) z^i \qquad B_{m^2-i}(\xi) = \sum_{|\kappa|=m^2-i} b_\kappa \xi^\kappa \tag{3}$$

The function $V_m(X)$ is a polynomial of degree m in z with polynomial coefficients from $\mathbb{R}[\xi]$. The coefficients $B_{m(m-1)}(\xi), \ldots, B_{m^2}(\xi)$ are homogeneous polynomials in the parameters ξ_k. The function $V_m(X)$ does itself not belong to $\mathbb{R}[X]$ but since $V_m(X) = \mathcal{V}_m(z)$, it belongs to $\mathbb{R}[\xi][z]$. Therefore the function $V_m(X)$ is given the name spherical polynomial: with every $\xi \in S_n$ the function $V_m(X) = \mathcal{V}_m(z)$ is associated which is a polynomial of degree m in the variable $z = \mathrm{sd}(X)$. Imposing the orthogonality conditions

$$\Gamma(z^i \mathcal{V}_m(z)) = 0 \qquad i = 0, \ldots, m-1 \tag{4}$$

signifies for $\mathcal{V}_m(z)$ and for $i = 0, \ldots, m-1$:

$$\Gamma(z^i \mathcal{V}_m(z)) = \int \ldots \int_{||X||_p \le 1} w(||X||_p) \left(\sum_{k=1}^{n} x_k \xi_k \right)^i \mathcal{V}_m \left(\sum_{k=1}^{n} x_k \xi_k \right) dX = 0$$

With the $c_i(\xi)$ we define the polynomial Hankel determinants

$$H_m(\xi) = \begin{vmatrix} c_0(\xi) & \cdots & c_{m-1}(\xi) \\ \vdots & \iddots & c_m(\xi) \\ & & \vdots \\ c_{m-1}(\xi) & \cdots & c_{2m-2}(\xi) \end{vmatrix} \qquad H_0(\xi) = 1$$

and call the functional Γ definite if

$$H_m(\xi) \not\equiv 0 \qquad m \ge 0$$

In the sequel of the text we assume that $\mathcal{V}_m(z)$ satisfies (4) and that Γ is a definite functional. Also we use both the notation $V_m(X)$ and $\mathcal{V}_m(z)$ interchangeably to refer to (3).

2 Connection with Classical Orthogonal Polynomials

With $w(||X||_p) = 1$ in (4) so-called spherical Legendre polynomials are obtained and with $w(||X||_p) = 1/\sqrt{1 - ||X||_p^2}$ spherical Tchebyshev polynomials. Let us now fix $\xi = \xi^*$ and take a look at the projected polynomials

$$\mathcal{V}_{m,\xi^*}(z) = V_m(\xi_1^* z, \ldots, \xi_n^* z)$$

which are polynomials of degree m in z. Are these projected polynomials themselves orthogonal? If so, what is their relationship to the univariate Legendre and Tchebyshev polynomials?

We introduce the univariate linear functional c^* acting on the variable z, by

$$c^*(z^i) = c_i(\xi^*) = \Gamma(z^i)\,|_{\xi=\xi^*} \tag{5}$$

In what follows we use the notation $V_m(z)$ to denote the univariate polynomials of degree m orthogonal with respect to the linear

Theorem 1. *Let the monic univariate polynomials $V_m(z)$ satisfy the orthogonality conditions $c^*(z^i V_m(z)) = 0$ with c^* given by* (5) *and $i = 0, \ldots, m-1$, and let the multivariate functions $V_m(X) = \mathcal{V}_m(z)$ satisfy the orthogonality conditions* (4). *Then for $X^* = (\xi_1^* z, \ldots, \xi_n^* z)$,*

$$H_m(\xi^*)V_m(z) = p_m(\xi^*)\mathcal{V}_{m,\xi^*}(z) = p_m(\xi^*)V_m(X^*)$$

In words, Theorem 2 says that, up to a normalizing factor $p_m(\xi^*)/H_m(\xi^*)$, the orthogonality conditions and the projection operator commute.

With respect to the projection property it is important to point out that $c^*(z^i)$ does not coincide with the one-dimensional version of c_κ given by (2), meaning (2) for $n = 1$ and $\kappa = i$. While in the one-dimensional situation, the linear functional

$$c(z^i) = c_i = \int_{-1}^{1} w(|x|)x^i\,dx \tag{6}$$

gives rise to the classical orthogonal polynomials, we do not immediately retrieve these classical polynomials from the projection, because the projected functional c^* given by (5) does not coincide with the functional c given by (6).

3 Gaussian Cubature Formulas

If the functional Γ is positive definite, in other words if $H_m(\xi) > 0$ for $m \geq 0$, then the zeroes $z_i^{(m)}(\xi^*)$ of $\mathcal{V}_{m,\xi^*}(z)$ are real and simple. In a neighbourhood of $z_i^{(m)}(\xi^*)$ holds that $z_i^{(m)}(\xi^*) = \phi_i^{(m)}(\xi^*)$ for a unique holomorphic function $\phi_i^{(m)}(\xi^*)$. Let us denote

$$\mathcal{W}_{m-1}(u) = \Gamma\left(\frac{\mathcal{V}_m(z) - \mathcal{V}_m(u)}{z-u}\right)$$

$$A_i^{(m)}(\xi) = \frac{\mathcal{W}_{m-1,\xi}(z_i^{(m)})}{\mathcal{V}'_{m,\xi}(z_i^{(m)})} = \frac{\mathcal{W}_{m-1}(\phi_i^{(m)}(\xi))}{\mathcal{V}'_m(\phi_i^{(m)}(\xi))} \tag{7}$$

Here the functions $\mathcal{W}_{m-1}(z)$ are also spherical polynomials, now of degree $m-1$ in z. Then the following cubature formula can rightfully be called a Gaussian cubature formula [2].

Theorem 2. *Let $\mathcal{P}(z)$ be a polynomial of degree $2m-1$ belonging to $\mathbb{R}(\xi)[z]$. Let the functions $\phi_i^{(m)}(\xi)$ be mutually distinct. Then*

$$\int \dots \int_{||X||_p \le 1} w(||X||_p)\, \mathcal{P}\left(\sum_{k=1}^n \xi_k x_k\right) dX = \sum_{i=1}^m A_i^{(m)}(\xi) \mathcal{P}(\phi_i^{(m)}(\xi))$$

The m-point Gaussian cubature formula from Theorem 3, with its parametrized nodes and weights, in fact exactly integrates an entire parametrized family of polynomials, over a domain in $\mathbb{R}^n$. We illustrate Theorem 3 with a bivariate example. Consider again the ℓ_2-norm and take

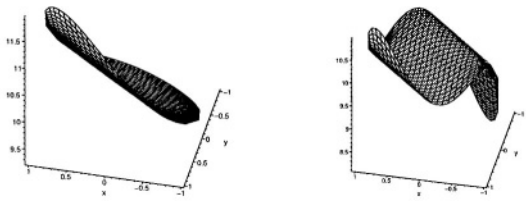
$$\mathcal{P}(z) = \frac{\xi_1}{\xi_2+1} z^3 + \frac{\xi_2}{\xi_1^2+1} z^2 + z + 10$$

Then

$$\phi_1^{(2)}(\xi) = \frac{1}{2}\sqrt{\xi_1^2+\xi_2^2} \qquad \phi_2^{(2)}(\xi) = -\frac{1}{2}\sqrt{\xi_1^2+\xi_2^2} \qquad A_1^{(2)}(\xi) = A_2^{(2)}(\xi) = \frac{\pi}{2}$$

$$\iint_{||(x,y)|| \le 1} \mathcal{P}(\xi_1 x + \xi_2 y) dx\, dy = \frac{\pi\left(\xi_2^3 + \xi_2\xi_1^2 + 40\xi_1^2 + 40\right)}{4\left(\xi_1^2+1\right)}$$
$$= A_1^{(2)} \mathcal{P}(\phi_1^{(2)}(\xi)) + A_2^{(2)} \mathcal{P}(\phi_2^{(2)}(\xi))$$

Two members of this family that are exactly integrated over the unit disk are for instance $\mathcal{P}_1(x,y)$ and $\mathcal{P}_2(x,y)$ which are obtained by choosing $(\xi_1,\xi_2) = (3/5, 4/5)$ and $(\xi_1,\xi_2) = (-\sqrt{2}/2, -\sqrt{2}/2)$ respectively.



$\mathcal{P}_1(x,y)$ $\mathcal{P}_2(x,y)$

Fig. 1.

References

1. B. Benouahmane. Approximants de Padé et polynômes orthogonaux à deux variables. Thèse de Doctorat, Rouen, 1992.
2. B. Benouahmane and A. Cuyt. Multivariate orthogonal polynomials, homogeneous Padé approximants and Gaussian cubature. *Numer. Algor.,* 24:1–15, 2000.
3. B. Benouahmane and A. Cuyt. Properties of multivariate homogeneous orthogonal polynomials. *J. Approx. Theory,* 113:1–20, 2001.

The Berlekamp-Massey Algorithm. A Sight from Theory of Pade Approximants and Orthogonal Polynomials

S.B. Gashkov[1]* and I.B. Gashkov[2]**

[1] Department Mechanics and Mathematics
Moscow State University, 119899 Moscow, Russia
gashkov@lsili.ru
[2] Department of Engineering Sciences,Physics and Mathematics,
Karlstad University,651 88 Karlstad, Sweden
Igor.Gachkov@kau.se

Abstract In this paper we interpret the Berlekamp-Massey algorithm (BMA) for synthesis of linear feedback shift register (LFSR) as an algorithm computing Pade approximants for Laurent series over arbitrary field. This interpretation of the BMA is based on a iterated procedure for computing of the sequence of polynomials orthogonal to some sequence of polynomial spaces with scalar product depending on the given Laurent series. It is shown that the BMA is equivalent to the Euclidean algorithm of a conversion of Laurent series in continued fractions.

1 Introduction

Let $f_0, \dots, f_{n-1}, \dots$ be a sequence of elements from arbitrary given field F. The given sequence is generated by the given LFSR iff this sequence satisfies the linear recurrence relation of order m, $\sum_{i=0}^{m} f_{i+k} q_i = 0, k = 0, 1, 2, \dots$ with the initial values $f_0, \dots, f_{m-1}$, where $Q(x) = \sum_{i=0}^{m} q_i x^i, q_m = 1$ is the characteristic polynomial of the given LFSR. (see [1].) The reciprocal characteristic polynomial $Q^*(x) = x^m Q(1/x)$ is called the feedback polynomial of the given LFSR. Denote by $L_n(f)$ the least degree of the polynomial Λ_n generating the sequence $f_0, \dots, f_{n-1}$. The number $L_n(f)$ is called *the linear complexity* of the sequences $f_0, \dots, f_{n-1}$. ([1].)The sequence $L_1(f), \dots L_n(f)$ is called *the linear complexity profile* of the sequence $f_0, \dots, f_{n-1}$. J. L. Massey [2] interpreted the Berlekamp algorithm [3] as the algorithm computing of the linear complexity profile for the given sequence and generating the corresponding sequence of the characteristic polynomials. (see [1].) Berlekamp's variant of the BMA is equivalent to the variant of the Euclidean algorithm (EA) given for BCH codes decoding ([4].) In [5], [6] was investigated connections between the BMA and continued fractions.

* The research of first author was support by The Russian Found of Fundamental Investigations, projects 02-01-10142, 00985 and by the grant Sc.Sch. 1807.2003.1
** The research of second author was support by The Royal Swedish Academy of Sciences

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 561–564, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In [7] was given the matrix generalization of the BMA. This generalization was used in [8] in the proof of the equivalence the BMA and the EA for decoding of BCH codes. We interpret the BMA from the point of view of theory of Pade approximants and orthogonal polynomials.

2 Pade Approximants for Laurent Series, Continued Fractions, Linear Complexity, and BMA

Any expression $z^n(c_0 + c_1/z + c_2/z^2 + \dots)$, $c_0 \neq 0$, with any integer n and coefficients $c_i \in F$ is called *a formal Laurent series.* The set $F((1/z))$ of all Laurent series forms the field with respect to the sum and product operation (see [9]). Any series $f(z)$ with null integral part may be expanded in continued fraction

$$f(z) = \cfrac{1}{a_1(z) + \cfrac{1}{a_2(z) + \cfrac{1}{a_3(z) + \dots}}}.$$

The proper fraction formed the first n levels of a given continued fraction, is called a $n-$th convergent to a given continued fraction and is denoted by τ_n. The numerator P_n and the denominator Q_n of the τ_n are calculated by the recurrent formulas $Q_n = a_n Q_{n-1} + Q_{n-2}, Q_1 = 1, Q_0 = 0$, $P_n = a_n P_{n-1} + P_{n-2}, P_1 = a_1, P_0 = 1$. (see [9].) The polynomials Q_n and P_n have degrees $s_n - 1$ and s_n, where $s_n = d_1 + \dots + d_n, s_0 = 0$, $d_n = \deg a_n$. We consider only Laurent series $f(z) = \sum_{i=0}^{\infty} f_i z^{-i-1}$ with null integral part. It is known

Theorem 1. *The following statements are equivalent:*

(i) the LFSR with the characteristic polynomial $Q(z)$ generates the sequence $f_0, \dots f_{L-1}$;

(ii) there exist the polynomials P, Q such that

$$f(z)Q(z) = P(z) + \frac{c}{z^{L-\deg Q+1}} + \dots, c \in F,$$

where $\deg P(z) < \deg Q(z)$;

(iii) there exist the polynomials P, Q such that

$$f(z) - \frac{P(z)}{Q(z)} = \frac{b}{z^{L+1}} + \dots, b \in F, \deg P < \deg Q.$$

For any n there exists a unique uncancelled proper fraction $P_n/G_n, \deg G_n \leq n$ such that $f(z)Q_n(z) = P_n(z) + \frac{c}{z^{n+1}} + \dots, c \in F$. (see [9]). This fraction is called $n-$th (diagonal) *Pade approximants* π_n of a number f. It a numerator P_n Suppose $\pi_n = P_n/G_n$ and $Q = G_n$ is the polynomial of minimal degree $m \leq n$ such that $f(z)Q(z) = P(z) + \frac{c_{n+1}}{z^{n+1}} + \dots$; then the sequence $f_0, \dots, f_{n+m-1}$ satisfies the recurrence relation $\sum_{i=0}^{m} f_{i+k} q_i = 0, k = 0, \dots, n-1$. Denote by Π_n degree of the fraction π_n.

Theorem 2. $L_{\Pi_n+n} = \Pi_n$.

If the degree of denominator of $n-$th Pade fraction is equal n,then the index n is called *normal.* If $n_0 < n_1$ there are adjacent normal indexes, then ([9]) for any $k, n_1 > k \geq n_0$, $f(z)G_{n_0}(z) - P_{n_0}(z) = G_{n_0}(z)(cz^{-n_0-n_1} + \dots) = ez^{-n_1} + \dots = bz^{-k-1} + \dots, c, e, b \in F$ and $G_k = G_{n_0}, n_0 = \Pi_{n_0} = \Pi_k = L_{k+\Pi_k} = L_{k+n_0}$. The sequence of normal indexes coincides with the sequence $s_0, s_1, s_2, \dots$ and Pade approximants $\pi_{s_n} = \tau_n = P_n/Q_n$. ([9].) Therefore, for any $k, s_n \leq k < s_{n+1}$, is valid $\pi_k = \pi_{s_n} = \tau_n, G_k = G_{s_n} = Q_n$ and for any sequence $\{f_0, \dots, f_{s_n+k}\}, k = s_n - 1, \dots, s_{n+1} - 2$ the minimal LFSR has the characteristic polynomial Q_n.

Theorem 3. $L_{k+s_n} = s_n,\ s_{n-1} \leq k < s_n$.

From theorem 3 easy follows well known

Theorem 4. *If the LFSR of the complexity $L_k(f)$ generates the sequence $f_0, \dots, f_k$ then $L_{k+1}(f) = L_k(f)$, else $L_{k+1}(f) = \max\{L_k(f), k + 1 - L_k(f)\}$.*

3 The Interpretation of the BMA in Terms of Orthogonal Polynomials

The following part of the paper *does not assume any knowledge about the BMA* and can be used for a *alternative description* of this algorithm.

Let $Pol(n)$ be the space of polynomials of degree less than n over a field F. For the given sequence $\{f_0, \dots, f_{n-1}\}$ over a field F we consider the linear functional $l_f(P) = \sum_{i=0}^{n-1} f_i p_i, P(z) = \sum\limits_{i=0}^{n-1} p_i z^i$. over the space $Pol(n)$. On the space $Pol(n)$ may be defined the scalar product $(P, Q) = (P, Q)_f$ of polynomials P, Q by equality $(P, Q) = l_f(PQ)$. Obviously is valid the identity $(P,Q) = (PQ, 1)$. Following [9], we rewrite the equalities $\sum_{i=0}^{m} f_{i+k} q_i = 0, k = 0, \dots, s-1$, where $Q(z) = \sum_{i=0}^{m} q_i z^i, q_m = 1$, as the equalities $(Q(z), z^k) = 0, k = 0, \dots, s-1$, where (P, Q) is the scalar product of polynomials P, Q. Orthogonality of vectors is denoted by the symbol $\perp$. Therefore, the system of equalities $(Q_n(z), z^k) = 0, k = 0, \dots, s_n - 1$ is equivalent to the relation $Q_n(z) \perp Pol_{s_n}$. Hence $Q_n \perp Q_{n-1}$ and the sequence of polynomials $Q_n(z) = \sum_{i=0}^{s_n} q_{n,i} z^i$, is uniquely determined by the mentioned above condition of the orthogonality.

Suppose that we have computed the polynomial Q_n by the given sequence $f_0, \dots, f_{2s_n-1}$. It is valid $\Lambda_{2s_n} = Q_n$. Computing $(Q_n(z), z^k) = \sum_{i=0}^{m} f_{i+k} q_{n,i}, m = s_n, k = m, m+1, \dots$ we find minimal k such that $(Q_n(z), z^k) \neq 0$. Hence, we can find s_{n+1}, because $k = s_{n+1} - 1$. Since the polynomial $Q_n(z)$ satisfies the condition $\sum_{i=0}^{s_n} f_{i+k} q_{n,i} = 0, k = 0, \dots, s_{n+1} - 2$, then the LFSR with the characteristic polynomial Q_n generates any sequence $f_0, \dots, f_k$, where $k = 2s_n, \dots, s_n + s_{n+1} - 2$. Therefore, we have $\Lambda_k = Q_n, k = 2s_n, \dots, s_n + s_{n+1} - 1$. Further, we find $d_{n+1} = s_{n+1} - s_n$. Let's look for the polynomial Q_{n+1} in the form $a_{n+1}(z)Q_n(z) + Q_{n-1}(z)$, where $\deg a_{n+1} = d_{n+1}$. The polynomial Q_{n+1} is uniquely determined (with an exactitude up to a constant factor) by the condition $Q_{n+1} \perp Pol_{s_{n+1}}$. By the

induction hypothesis $Q_n \perp Pol_{s_{n+1}-1}$, but the polynomial Q_n is not orthogonal to the space $Pol_{s_{n+1}}$. Hence, $(Q_n(z), z^{s_{n+1}-1}) = \Delta_{s_n+s_{n+1}-1} \neq 0$. Since $a_{n+1}(z)z^k \in Pol_{s_{n+1}-1}, z^k \in Pol_{s_n-1}$, we see that for any polynomial a_{n+1} of degree d_{n+1} $a_{n+1}(z)Q_n(z) + Q_{n-1}(z) \perp Pol_{s_n-1}$. To choose the polynomial a_{n+1} such that the polynomial $a_{n+1}(z)Q_n(z) + Q_{n-1}(z)$ is orthogonal to the space generated by the monomials $z^{s_n-1}, \ldots, z^{s_{n+1}-1}$, we need next condition. The projections of the polynomials $a_{n+1}(z)Q_n(z)$ and $Q_{n-1}(z)$ on this space are opposite, i.e. $(a_{n+1}(z)Q_n(z), z^k) = -(Q_{n-1}(z), z^k), k = s_n - 1, \ldots, s_{n+1} - 1$. These equalities concerning coefficients of the polynomial a_{n+1} determine the system of linear equations with a triangular matrix. This system may be solved by the following iterated algorithm.

A step with any number. At $i-$th step we correct the polynomial $Q_{n+1}^{(i-1)}$ iff $\Delta_{s_n+s_{n+1}+i-2} = (Q_{n+1}^{(i-1)}, z^{s_n+i-2}) \neq 0$. Then we look for the $Q_{n+1}^{(i)} = Q_{n+1}^{(i-1)} + cQ_n z^{d_{n+1}-i+1}$ such that $Q_{n+1}^{(i)} \perp z^{s_n+i-2}$. For this goal we search a constant c such that the projections $Q_{n+1}^{(i-1)}, cQ_n z^{d_{n+1}-i+1}$ on the monomial z^{s_n+i-2} are opposite. Hence, $c = -\Delta_{s_n+s_{n+1}+i-2}/\Delta_{s_n+s_{n+1}-1}$. Since $Q_{n+1}^{(i-1)} \perp Pol_{s_n+i-2}$ by the induction hypothesis, we have $Q_{n+1}^{(i-1)} \perp z^{s_n+k}$ for any $k, -1 \leqslant k \leqslant i-3$. Therefore, $(Q_{n+1}^{(i)}, z^{s_n+k}) = (Q_{n+1}^{(i-1)}, z^{s_n+k}) + (cQ_n z^{d_{n+1}-i+1}, z^{s_n+k}) = c(Q_n, z^{s_{n+1}+k+1-i}) = 0$. Since $Q_{n+1}^{(i)} \perp Pol_{s_n+i-1}$, we see that $\Lambda_{s_n+s_{n+1}+i-2} = Q_{n+1}^{(i)}$. **Last step.** Finally, at $d_{n+1}+1-$th step we get the polynomial $Q_{n+1}^{(d_{n+1}+1)} = Q_n a_{n+1} + Q_{n-1}, \deg Q_{n+1}^{(d_{n+1}+1)} = s_{n+1}$, such that $Q_{n+1}^{(d_{n+1}+1)} \perp Pol_{s_n+d_{n+1}} = Pol_{s_{n+1}}$. This polynomial coincides with the polynomial Q_{n+1}. Hence $\Lambda_{2s_{n+1}} = Q_{n+1}$.

References

1. Jungnickel D., *Finite fields. Structure and arithmetic,* Wissenschaftsverlag, Mannheim, Leipzig, Wien, Zurich (1993).
2. Massey J.L., *Feedback Shift Register Synthesis and BCH Decoding, IEEE Trans. Inform. Theory,* **IT15** (1969), 122-128.
3. Berlekamp E.R., *Algebraic coding theory,* McGraw Hill (1968).
4. Dornstetter J.L., *On the equivalence between Berlekamp's and Euclid's algorithms, IEEE Trans. Inf. Theory* **IT-33,** 3 (1987), 428-431.
5. Cheng U., *On the continued fractions and Berlekamp's Algorithm, IEEE Trans. Inf. Theory* **IT-30,** 3 (1984), 541-544.
6. Zongduo Dai, Kencheng Zeng, *Continued fractions and Berlekamp-Massey Algorithm, Advances in Cryptology - Auscript-90,* Springer Verlag, Berlin (1990), 24-31.
7. Feng G.-L., Tzeng K.K., *A generalization of the Berlekamp- Massey algorithm for multisequence shift-register sinthesis with applications To decoding cyclic codes, IEEE Trans. Inf. Theory* **IT-37,** (Sept.1991), 1274-1287.
8. Agnes E. Heydtmann, J.M.Jensen, *On the equivalence of the Berlekamp-Massey and Euclidean algorithms for decoding, IEEE Trans. Inf. Theory* **IT-46,** 7 (2000), 2614-2624.
9. Nikishin E.M., Sorokin V.N. *Rational approximations and Orthogonality,* Moscow, Nauka (1988), AMS, New York (1996).

An Advanced Version of the Local-Global Step Size Control for Runge-Kutta Methods Applied to Index 1 Differential-Algebraic Systems

Gennady Y. Kulikov

School of Computational and Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, Johannesburg, South Africa
gkulikov@cam.wits.ac.za

Abstract. In recent paper [5] a procedure was developed to control a step size for Runge-Kutta methods. Here, we present a new version of that step size selection to make it better in exceptional cases when the old version does not work appropriately.

In this paper, we consider a system of differential-algebraic equations (DAEs) of the form

$$x'(t) = g\big(x(t), y(t)\big), \quad y(t) = f\big(x(t), y(t)\big), \tag{1}$$

where $t \in [0,T]$, $x(t) \in \mathbf{R}^m$, $y(t) \in \mathbf{R}^n$, $g : D \subset \mathbf{R}^{m+n} \to \mathbf{R}^m$, $f : D \subset \mathbf{R}^{m+n} \to \mathbf{R}^n$, and where an initial conditions $x(0) = x^0$, $y(0) = y^0$ are also given and they are consistent; i. e., $y^0 = f(x^0, y^0)$. We suppose that problem (1) possesses an unique solution $z(t) \stackrel{\text{def}}{=} \big(x(t)^T, y(t)^T\big)^T$ on the interval $[0,T]$.

To solve problem (1) numerically, we applied an l-stage Runge-Kutta (RK) method and come to the following discrete system:

$$x_{ki} = x_k + \tau_k \sum_{j=1}^{l} a_{ij} g(x_{kj}, y_{kj}), \quad y_{ki} = f(x_{ki}, y_{ki}), \quad i = 1, 2, ..., l, \tag{2a}$$

$$x_{k+1} = x_k + \tau_k \sum_{i=1}^{l} b_i g(x_{ki}, y_{ki}),\ y_{k+1} = f(x_{k+1}, y_{k+1}),\ k = 0, 1, ..., K-1, \tag{2b}$$

We assume further that the RK formula is of order s, and that problem (1) satisfies the smoothness condition and the nonsingularity one in a neighborhood of the solution $z(t)$ on the interval $[0,T]$ (we refer to [5] for details).

It is well-known [1], [2], [3] that under the conditions mentioned above problem (1) is of index 1 and method (2) is convergent of order s. Moreover, it was proved in [3] that the local error of method (2) has the form

$$x(t_{k+1}) - \tilde{x}_{k+1} = \check{\psi}_{s+1}(t_k)\tau_k^{s+1} + O(\tau_k^{s+2}), \tag{3a}$$

$$y(t_{k+1}) - \tilde{y}_{k+1} = \check{\phi}_{s+1}(t_k)\tau_k^{s+1} + O(\tau_k^{s+2}), \tag{3b}$$

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 565–569, 2004.
© Springer-Verlag Berlin Heidelberg 2004

$k = 0, 1, \ldots, K-1$, as $\tau_k \to 0$, where $\tilde{z}_{k+1} \stackrel{\text{def}}{=} \left((\tilde{x}_{k+1})^T, (\tilde{y}_{k+1})^T\right)^T$ is the solution of (2) provided that $z_k = z(t_k)$, and the functions $\check{\psi}_{s+1}(t_k)$, $\check{\phi}_{s+1}(t_k)$ denote the coefficients of leading terms of the local error of RK method (2). In addition, Theorem 2.1 in [4] establishes that the global error of method (2) possesses the following asymptotic expansion:

$$x(t_{k+1}) - x_{k+1} = \psi_s(t_{k+1})(\tau_k^*)^s + O\left((\tau_k^*)^{s+1}\right), \tag{4a}$$

$$y(t_{k+1}) - y_{k+1} = \phi_s(t_{k+1})(\tau_k^*)^s + O\left((\tau_k^*)^{s+1}\right), \tag{4b}$$

$k = 0, 1, \ldots, K-1$, as $\tau_k^* \to 0$, where the pair of functions $\psi_s(t)$ and $\phi_s(t)$ is a solution of the DAEs

$$\psi_s'(t) = \partial_x g\big(x(t), y(t)\big)\psi_s(t) + \partial_y g\big(x(t), y(t)\big)\phi_s(t) + \check{\psi}_{s+1}(t), \tag{5a}$$

$$\phi_s(t) = \partial_x f\big(x(t), y(t)\big)\psi_s(t) + \partial_y f\big(x(t), y(t)\big)\phi_s(t), \tag{5b}$$

$$\psi_s(0) = 0, \quad \phi_s(0) = 0. \tag{5c}$$

The new step size τ_k^* in formulas (4) means a step size for integrating problem (1) in the local interval $[t_k, t_{k+1}]$ by a fixed step size version of method (2). Below, we will discuss how to define it more precisely.

Taking into account possible stiffness of problem (5), we apply the implicit Euler method to this problem and obtain the following recursion relation for an approximate computation of the functions $\psi_s(t)$ and $\phi_s(t)$:

$$\begin{pmatrix} \tilde{\psi}_s(t_{k+1}) \\ \tilde{\phi}_s(t_{k+1}) \end{pmatrix} = P_{k+1}^{-1} \begin{pmatrix} \tilde{\psi}_s(t_k) + \tau_k \hat{\psi}_{s+1}(t_{k+1}) \\ 0 \end{pmatrix}, \tag{6a}$$

$$\tilde{\psi}_s(0) = 0, \quad \tilde{\phi}_s(0) = 0, \quad k = 0, 1, \ldots, K-1, \tag{6b}$$

where

$$P_{k+1} \stackrel{\text{def}}{=} \begin{pmatrix} I_m - \tau_k \partial_x g\big(\tilde{x}_{k+1}, \tilde{y}_{k+1}\big) & -\tau_k \partial_y g\big(\tilde{x}_{k+1}, \tilde{y}_{k+1}\big) \\ -\partial_x f\big(\tilde{x}_{k+1}, \tilde{y}_{k+1}\big) & I_n - \partial_y f\big(\tilde{x}_{k+1}, \tilde{y}_{k+1}\big) \end{pmatrix}.$$

In formula (6a), 0 denotes the null vector of dimension n, but full particularities can be found in [5].

The next important stage in the numerical integration of semi-explicit DAEs of index 1 is a selection of the maximum step size keeping the truncated local and global errors bounded. Let ϵ_l and ϵ_g be given limits for the local and global errors of method (2), respectively. Considering that the global error of the numerical solution at the point t_k does not exceed ϵ_g, we now explain how to find an approximate solution of problem (1) at the point t_{k+1} with the given accuracy.

We first compute the local error at the point t_{k+1}, by using any classical approach, with the accuracy ϵ_l (see, for example, [3]). After that we determine

the coefficient of the principal term of the global error by formulas (6). Then we find the maximum step size τ_k^* guaranteeing the necessary accuracy[1]:

$$p = \left[\tau_k \left(\left\| \left(\tilde{\psi}_s(t_{k+1})^T, \tilde{\phi}_s(t_{k+1})^T \right)^T \right\| / \epsilon_g \right)^{1/s} \right] + 1, \quad \tau_k^* = \tau_k / p. \tag{7}$$

Further, we solve problem (1) on the local interval $[t_k, t_{k+1}]$ by method (2) with the fixed step size τ_k^* and obtain the numerical solution at the point t_{k+1} with the accuracy ϵ_g. Moreover, in order to compensate the influence of the rejected terms in global error expansion (4) we improve the numerical solution by means of Richardson extrapolation or two RK methods of different orders. Now we set the next step size $\tau_{k+1} = \tau_k^*$ and repeat all the computations at the next time point, and so on.

In [5], it was shown that, with a practical point of view, it is important to use the stable version of the local-global step size control with an additional bound ϵ_{gl}. To demonstrate the nice practical properties of the stable controller, we apply the Trapezoidal Rule with Newton iteration (TRN method) to the restricted three body problem in differential-algebraic formulation [5].

Table 1. Global errors of the variable step size TRN method with the stable local-global step size control and with extrapolation ($\epsilon_l = \epsilon_g$ and $\epsilon_{gl} = \epsilon_g/100$)

Number of iterations N	required accuracy $\epsilon_g = 10^{-03}$	$\epsilon_g = 10^{-05}$	$\epsilon_g = 10^{-07}$	$\epsilon_g = 10^{-09}$	$\epsilon_g = 10^{-11}$
1	$6.40 \cdot 10^{-04}$	$8.12 \cdot 10^{-06}$	$4.74 \cdot 10^{-08}$	$2.04 \cdot 10^{-10}$	$2.90 \cdot 10^{-12}$
2	$6.52 \cdot 10^{-04}$	$8.06 \cdot 10^{-06}$	$4.73 \cdot 10^{-08}$	$2.16 \cdot 10^{-10}$	$3.62 \cdot 10^{-13}$

From the data of Table 1 it follows that the variable step size TRN method with the stable local-global error control mechanism achieves automatically any reasonable accuracy preassigned for the numerical solution, even when only one Newton iteration per each time point has been fulfilled.

Table 2. Global errors of the variable step size TRN method with the stable local-global step size control and with extrapolation ($\epsilon_l = \epsilon_g$ and $\epsilon_{gl} = \epsilon_g/100$)

Number of iterations N	required accuracy $\epsilon_g = 10^{-03}$	$\epsilon_g = 10^{-04}$	$\epsilon_g = 10^{-05}$	$\epsilon_g = 10^{-06}$	$\epsilon_g = 10^{-07}$
1	$1.57 \cdot 10^{+02}$	$4.43 \cdot 10^{+00}$	$1.92 \cdot 10^{+00}$	$2.07 \cdot 10^{+00}$	$2.00 \cdot 10^{+00}$
2	$1.85 \cdot 10^{+00}$	$1.00 \cdot 10^{+00}$	$1.10 \cdot 10^{-01}$	$3.47 \cdot 10^{-02}$	$3.22 \cdot 10^{-03}$
3	$1.85 \cdot 10^{-01}$	$1.96 \cdot 10^{-02}$	$2.19 \cdot 10^{-03}$	$1.82 \cdot 10^{-04}$	$3.46 \cdot 10^{-05}$

[1] The square brackets in formula (7) means, as is customary, the integral part of the number.

Table 3. Global errors of the variable step size TRN method with the advanced stable local-global step size control and with extrapolation ($\epsilon_l = \epsilon_g$ and $\epsilon_{gl} = \epsilon_g/100$)

Number of iterations N	required accuracy $\epsilon_g = 10^{-03}$	$\epsilon_g = 10^{-04}$	$\epsilon_g = 10^{-05}$	$\epsilon_g = 10^{-06}$	$\epsilon_g = 10^{-07}$
1	$1.82 \cdot 10^{+00}$	$1.58 \cdot 10^{+00}$	$8.87 \cdot 10^{-01}$	$1.86 \cdot 10^{-01}$	$3.91 \cdot 10^{-02}$
2	$1.02 \cdot 10^{-02}$	$5.92 \cdot 10^{-04}$	$3.12 \cdot 10^{-05}$	$1.53 \cdot 10^{-06}$	$7.41 \cdot 10^{-08}$
3	$1.02 \cdot 10^{-05}$	$7.67 \cdot 10^{-08}$	$1.86 \cdot 10^{-10}$	$2.47 \cdot 10^{-09}$	$1.18 \cdot 10^{-09}$

Let us now show that the stable local-global step size control sometimes cannot provide the required accuracy. To illustrate this, we take the TRN method with trivial predictor and apply it to the restricted three body problem on the double interval $[0,2T]$; i. e. we have increased the length of the integration segment by a factor of 2. This makes the test problem more difficult for solving with the guaranteed accuracy because of round-off errors and the omitted terms in global error expansion (4). Indeed, the numerical results demonstrate that our error control mechanism cannot provide the required accuracy for any number of Newton iterations per time point (see Table 2).

To improve the local-global step size control in order to solve our test problem with the necessary accuracy, we involve more terms of the global error expansion in the error control procedure. First of all Richardson extrapolation gives us a way to find any reasonable number of the first terms in the local error expansion. Then, by substitution of the coefficients of these terms in the inhomogeneous part of equation (5a), we obtain systems of linear DAEs for the coefficients of relevant terms in the global error expansion. After that we can use the main ideas discussed above in order to compute and control some first leading terms of the global error of Runge-Kutta methods. If we now derive the advanced stable local-global step size control based on the computation of the two first leading terms of the global error expansion and apply it to our test problem then we see that the new advanced version of the step size control is more reliable for practical use (see Table 3). Finally, it is necessary to point out that, as the last example shows, the sufficient quantity of Newton iterations is very important for our step size control to be valid. We refer to [6] for more details.

References

1. Hairer, E., Wanner, G.: Solving ordinary differential equations II: Stiff and differential-algebraic problems. Springer-Verlag, Berlin, 1996
2. Kulikov, G.Yu.: Convergence theorems for iterative Runge-Kutta methods with a constant integration step, *(in Russian)* Zh. Vychisl. Mat. Mat. Fiz. **36** (1996) No. 8, 73–89; *translation in* Comput. Math. Math. Phys. **36** (1996) No. 8, 1041–1054
3. Kulikov, G.Yu.: On the numerical solution of the Cauchy problem for a system of differential-algebraic equations by means of implicit Runge-Kutta methods with a variable integration step. *(in Russian)* Vestnik Moskov. Univ. Ser. 1 Mat. Mekh. (1997) No. 5, 7–11; *translation in* Moscow Univ. Math. Bull. **52** (1997) No. 5, 6–10

4. Kulikov, G.Yu.: A theory of symmetric one-step methods for differential-algebraic equations. Russian J. Numer. Anal. Math. Modelling. **12** (1997) No. 6, 501–523
5. Kulikov, G.Yu.: A local-global version of a stepsize control for Runge-Kutta methods. Korean J. Comput. Appl. Math. **7** (2000) No. 2, 289–318
6. Kulikov, G.Yu.: One-step methods and implicit extrapolation technique for index 1 differential-algebraic systems. Russian J. Numer. Anal. Math. Modelling. (to appear)

INTEGRATOR: A Computational Tool to Solve Ordinary Differential Equations with Global Error Control

Gennady Y. Kulikov and Sergey K. Shindin

School of Computational and Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, Johannesburg, South Africa
{gkulikov,sshindin}@cam.wits.ac.za

Abstract. In recent papers [6]–[10] the technique for local and global errors estimation and the local-global step size control have been presented to solve both ordinary differential equations and semi-explicit index 1 differential-algebraic systems by multistep methods with any automatically obtained reasonable accuracy. Now we describe the object oriented library INTEGRATOR (ver. 1.0) built in C++ for portability and performance across a wide class of machine architectures. Also we give some computational tests and a comparison with the software implemented the well-known Gear's method which has been included into MATLAB (ver. 6.2, release 12).

The problem of an automatic global error control for ODEs of the form

$$x'(t) = g\big(t, x(t)\big), \quad t \in [t_0, t_0 + T], \quad x(t_0) = x^0, \tag{1}$$

where $x(t) \in \mathbf{R}^n$ and $g : D \subset \mathbf{R}^{n+1} \to \mathbf{R}^n$ is a sufficiently smooth function, is one of the challenges of modern computational mathematics. ODE (1) is quite usual in applied research and practical engineering (see, for example, [l]–[5]) Therefore any issue in that field possesses a great potential to develop intelligent software packages for mathematical modelling tasks. Thus, following the recent advances in computer algorithms for global error monitoring and controlling [7]–[10], we present the new user-oriented library INTEGRATOR for solving problem (1) with any accuracy set by the user (up to round-off). In this paper we give a brief outline of library types and routines as well as numerical experiments and data analysis. We clearly show an advantage of the new software at least over the implementation of Gear's method [3] taken from MATLAB (ver. 6.2, release 12) that is a good reason to use INTEGRATOR in practice.

Most of numerical algorithms imply vector and matrix computations in this or that form. Therefore, when constructing any object-oriented software, it is important to separate the algorithm from the vector-matrix arithmetic implementation for a convenience to use or to modify the code. Following this philosophy, we divided the library into two parts. The first one contains basic C++ classes

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 570–573, 2004.
© Springer-Verlag Berlin Heidelberg 2004

and routines providing facilities to code complex vectors-matrices algorithms closer to its natural mathematical notation. The second part includes routines implemented both multistep methods (with fixed or variable coefficients) and Nordsieck ones with the local-global step size control in order to solve ODEs of the form (1) with the accuracy set by the user.

Here, we do not describe the details concerning the vector and matrix classes because they are not essential for using INTEGRATOR in practice. We start with the common description of function-solvers implemented in the second part of our library. Each of these functions are defined in the standard way:

```
statistics <solver>(odestruct& _ODE, solveropt& _OPT);
```

The data structures `odestruct`, `solveropt` and `statistics` are declared in the header file `numerics.h` and contain input and output parameters of the function-solvers.

The input data structure `odestruct` has been designed to keep the input information about ODE (1) and it contains a pointer to the functions for evaluating $g(t,x)$ and its Jacobian $\partial_x g(t,x)$, a vector-column of the initial values x_0 for problem (1), and also the initial and final time points of the integration interval $[t_0, t_0+T]$.

The input data structure `solveropt` includes four groups of parameters needed for the function-solvers. The first one is aimed to set the order of multistep method, the type of iteration, and the number of iteration steps per grid point. At present INTEGRATOR supports the following three iterative methods: `solveropt::mt_NEWTON` is a full Newton iteration, `solveropt::mt_mNEWTON` is a modified Newton iteration, `solveropt: :mt_sITER` is a simple (or fixed-point) iteration. The lower bounds of enough iteration steps for each type of the iterative methods can be found in [7].

The second group of parameters is necessary to set particulars of the local-global step size control (the detailed description of that algorithm for multistep methods with both fixed coefficients and variable ones applied to ordinary differential equations or to differential-algebraic systems can be found in [6]–[10]). In particular, these parameters include the global error tolerance ϵ_g, the safety factor, the initial step size τ_1 for an starting procedure, and the initial restriction for the maximum step size τ.

The third group of parameters as well as the fourth one manages an output of statistical and integration data to the consol and the user defined `C++` streams, respectively.

All function-solvers return the data structure `statistics`. The data fields of this structure preserve the statistical data accumulated during the integration process of problem (1).

The header files `bdf.h` and `adams.h` declare all the function-solvers of the library. The first one contains algorithms based on Backward Differentiation Formulas (BDF methods): the function-solver `VBDF_SCL` implements fixed-coefficients BDF methods with polynomial interpolation of the numerical solution; the function-solver `VBDF_SCH` implements variable-coefficients BDF methods; the function-solver `NBDF_SCH` implements the Nordsieck representation for

BDF methods. The library INTEGRATOR supports the BDF methods of orders 2–6.

Table 1. Global errors obtained when treating problem (0.1) by the fixed-coefficients implicit Adams methods (with extrapolation) according to order s

order	required accuracy				
s	$\epsilon_g = 10^{-02}$	$\epsilon_g = 10^{-03}$	$\epsilon_g = 10^{-04}$	$\epsilon_g = 10^{-05}$	$\epsilon_g = 10^{-06}$
2	$2.11 \cdot 10^{-05}$	$2.42 \cdot 10^{-07}$	—	—	—
3	$3.15 \cdot 10^{-04}$	$1.08 \cdot 10^{-05}$	$5.03 \cdot 10^{-07}$	$2.01 \cdot 10^{-08}$	$1.14 \cdot 10^{-09}$
4	$5.41 \cdot 10^{-05}$	$2.70 \cdot 10^{-06}$	$1.69 \cdot 10^{-07}$	$8.70 \cdot 10^{-09}$	$3.18 \cdot 10^{-10}$
5	$1.26 \cdot 10^{-04}$	$1.94 \cdot 10^{-05}$	$2.68 \cdot 10^{-06}$	$2.46 \cdot 10^{-07}$	$2.68 \cdot 10^{-08}$
6	$4.11 \cdot 10^{-03}$	$1.75 \cdot 10^{-05}$	$8.49 \cdot 10^{-07}$	$5.60 \cdot 10^{-08}$	$3.37 \cdot 10^{-09}$
7	$1.76 \cdot 10^{-04}$	$7.60 \cdot 10^{-06}$	$8.82 \cdot 10^{-08}$	$2.80 \cdot 10^{-08}$	$4.00 \cdot 10^{-09}$

Table 2. Global errors obtained when treating problem (1.20) by the variable-coefficients BDF methods (with extrapolation) according to order s

order	required accuracy				
s	$\epsilon_g = 10^{-02}$	$\epsilon_g = 10^{-03}$	$\epsilon_g = 10^{-04}$	$\epsilon_g = 10^{-05}$	$\epsilon_g = 10^{-06}$
2	$2.54 \cdot 10^{-06}$	$3.08 \cdot 10^{-08}$	$7.53 \cdot 10^{-09}$	$8.19 \cdot 10^{-11}$	$3.15 \cdot 10^{-08}$
3	$1.11 \cdot 10^{-05}$	$6.57 \cdot 10^{-07}$	$2.38 \cdot 10^{-08}$	$1.83 \cdot 10^{-09}$	$4.16 \cdot 10^{-10}$
4	$5.40 \cdot 10^{-05}$	$2.96 \cdot 10^{-06}$	$1.72 \cdot 10^{-07}$	$1.35 \cdot 10^{-08}$	$5.87 \cdot 10^{-10}$
5	$6.03 \cdot 10^{-04}$	$4.24 \cdot 10^{-05}$	$2.99 \cdot 10^{-06}$	$1.44 \cdot 10^{-07}$	$8.30 \cdot 10^{-09}$
6	$2.64 \cdot 10^{-04}$	$7.12 \cdot 10^{-05}$	$3.37 \cdot 10^{-06}$	$2.85 \cdot 10^{-07}$	$2.03 \cdot 10^{-08}$

The second header file includes algorithms based on implicit Adams methods: the function-solver `VADM_SCL` implements fixed-coefficients implicit Adams methods with polynomial interpolation of the numerical solution; the function-solver `VADM_SCH` implements variable-coefficients implicit Adams methods; the function-solver `NADM_SCH` implements the Nordsieck representation for implicit Adams methods. Orders of all the Adams methods have to be from 2 up to 7.

Finally, we give a couple of numerical examples showing the efficiency of INTEGRATOR. As test problems, we take ODEs (0.1) and (1.20) from [4]. The first one is the restricted three body problem and possesses the periodic solution-path. The second ODE has the exact solution. Thus, we are capable to observe the work of our software in practice.

Now we apply the function-solvers of INTEGRATOR described above to these test problems. We use both Adams methods and BDF ones with fixed and variable coefficients to compute the numerical solutions. We also determine the real errors appeared in the integrations and compare them with the required accuracy. The lines in Table 1 mean that the second order function-solvers are not able to find the numerical solution with the set accuracy when $\epsilon_g \leq 10^{-04}$ because the required step size in this situation is smaller than the minimum admissible step size value $\tau_{\min}$.

Table 3. Global errors obtained when treating the test problems by the Gear's method from MATLAB (ver. 6.2, release 12)

test problem	required accuracy $\epsilon_l = 10^{-02}$	$\epsilon_l = 10^{-03}$	$\epsilon_l = 10^{-04}$	$\epsilon_l = 10^{-05}$	$\epsilon_l = 10^{-06}$	$\epsilon_l = 10^{-07}$
(0.1)	$2.09 \cdot 10^{+00}$	$2.07 \cdot 10^{+00}$	$1.94 \cdot 10^{+00}$	$1.45 \cdot 10^{+00}$	$6.00 \cdot 10^{-02}$	$1.00 \cdot 10^{-03}$
(1.20)	$1.40 \cdot 10^{+02}$	$1.09 \cdot 10^{+02}$	$9.50 \cdot 10^{+01}$	$2.15 \cdot 10^{+00}$	$2.92 \cdot 10^{-01}$	$6.23 \cdot 10^{-02}$

Tables 1, 2 display that all the function-solvers have achieved the goal; i.e., they have computed the numerical solutions of problems (0.1) and (1.20) with the set accuracy ϵ_g. To emphasis the advantage of INTEGRATOR over the standard software packages, we applied the Gear's method from MATLAB (ver. 6.2, release 12) to both test problems (see Table 3). A comparison of these data with Tables 1, 2 gives a right to conclude that the new software, in fact, controls automatically the global error of numerical solution. This is a good result for future testing of the library INTEGRATOR and for implementing it in practice.

References

1. Arushanyan, O.B., Zaletkin, S.F.: Numerical solution of ordinary differential equations using FORTRAN. *(in Russian)* Mosk. Gos. Univ., Moscow, 1990
2. Butcher, J.C.: Numerical methods for ordinary differential equations. John Wiley & Son, Chichester, 2003
3. Gear, C.W.: Numerical initial value problems in ordinary differential equations. Prentice-Hall, 1971
4. Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations I: Nonstiff problems. Springer-Verlag, Berlin, 1987
5. Hairer, E., Wanner, G.: Solving ordinary differential equations II: Stiff and differential-algebraic problems. Springer-Verlag, Berlin, 1996
6. Kulikov, G.Yu., Shindin, S.K.: A local-global stepsize control for multistep methods applied to semi-explicit index 1 differential-algebraic equations. Korean J. Comput. Appl. Math. **6** (1999) No. 3, 463–492
7. Kulikov, G.Yu., Shindin, S.K.: A technique for controlling the global error in multistep methods. *(in Russian)* Zh. Vychisl. Mat. Mat. Fiz. **40** (2000) No. 9, 1308–1329; *translation in* Comput. Math. Math. Phys. **40** (2000) No. 9, 1255–1275
8. Kulikov, G.Yu., Shindin, S.K.: On multistep extrapolation methods for ordinary differential equations. *(in Russian)* Dokl. Akad. Nauk, **372** (2000) No. 3, 301–304; *translation in* Doklady Mathematics, **61** (2000) No. 3, 357–360
9. Kulikov, G.Yu., Shindin, S.K.: On effective computation of asymptotically correct estimates of the local and global errors for multistep methods with fixed coefficients. *(in Russian)* Zh. Vychisl. Mat. Mat. Fiz. (to appear); *translation in* Comput. Math. Math. Phys. (to appear)
10. Kulikov, G.Yu., Shindin, S.K.: On interpolation type multistep methods with automatic global error control. *(in Russian)* Zh. Vychisl. Mat. Mat. Fiz. (to appear); *translation in* Comput. Math. Math. Phys. (to appear)

Reconstruction of Signal from Samples of Its Integral in Spline Subspaces*

Jun Xian, Yongjin Li, and Wei Lin**

Department of Mathematics, Sun Yat-sen University, Guangzhou, 510275, China

Abstract. In this paper, we study the reconstructing of signals based on regular and irregular incremental integral samples in some non-bandlimited space—spline subspace, and we obtain reconstructing formulas with a new method.

1 Introduction

In digital signal and image processing, digital communication, etc., a continuous signal is usually represented and processed by using its discrete samples. For a bandlimited signal of finite energy, it is completely characterized by its samples, by the famous classical Shannon sampling theorem. However, in many real applications sampling points are not always regularly and sampled value may not be values of a signal f precisely at times x_k for the inertia of the measurement aparatus. As for the signal spaces, they are not always bandlimited signal of finite energy. The problem arose initially in the design of an interferometer in which the interferogram is obtained using a continuously moving mirror, but may also have bearing in other problem in which the data are smoothed by an integrating sensor, such as CCD array with slow response time compared to the sample integral. So we need to give the reconstruction of signal from samples of its integral. In this paper, we discuss the reconstruction of signal from samples of its integral in spline subspaces.

Though Sun and Zhou studied this problem in [7], their result and method are not easy to apply in practical application and computation. We will give reconstruction formula from samples of its integral in spline subspaces with the simple method. And the reconstruction formula is more practical in application.

2 Reconstruction of Signal from Regular Incremental Integral Samples in Spline Subspaces

Let $y_k = \int_{-\infty}^{k+1} f(t)dt, y_{k-1} = \int_{-\infty}^{k} f(t)dt, z_k = y_k - y_{k-1}$. We refer to $\{y_k\}$ as the set of cumulative integral samples. The problem is to reconstruct the signal f

* This work is supported in part by the China-NSF, the Guangdong-NSF and the Foundation of Sun Yat-Sen University Advanced Research Centre.

** E-mail address: xianjun11@sina.com (Jun Xian), stslw@zsu.edu.cn (Wei Lin), stslyj@zsu.edu.cn (Yongjin Li)

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 574–577, 2004.
© Springer-Verlag Berlin Heidelberg 2004

from $\{z_k\}$ (incremental integral samples). We now introduce some notations and lemmas used in Section 2 and Section 3. In this paper, the Fourier transform of $f \in L^2(\mathbb{R})$ is defined by $\hat{f}(\omega) = \int_{\mathbb{R}} f(x)e^{-ix\omega}dx$. $V_N = \{\sum_{k\in Z} c_k\varphi_N(\cdot - k) : \{c_k\} \in \ell^2\}$ is spline subspace generated by $\varphi_N = \chi_{[0,1]} * \cdots * \chi_{[0,1]}$ (N convolutions), $N \geq 1$.

Lemma 2.1[5] *$\{\varphi_N(\cdot - k) : k \in Z\}$ is Riesz basis for V_N, $A_N = \sum_k |\hat{\varphi}_N(\pi + 2k\pi)|^2$ and $B_N = 1$ are its lower and upper bounds, respectively.*

Lemma 2.2[2] *If $\{x_n\}$ is real sequence with $\sup_i(x_{i+1} - x_i) = \delta < 1$, then there exist constants c and C such that for arbitrary $f \in V_N$ we have $c\|f\|_2^2 \leq \sum_i |f(x_i)|^2 \leq C\|f\|_2^2$.*

Lemma 2.3 *Suppose that $\hat{\tilde{\varphi}}_N(\omega) = \frac{\hat{\varphi}_N(\omega)}{|\Pi_N(e^{-i\omega})|}$, where $\Pi_N(\omega) = \sum_{n=0}^{2N-2} \varphi_{2N}(n+1)\omega^k$. Then $k_x(\omega) = k(\omega, x) = \sum_n \tilde{\varphi}_N(w - n)\overline{\varphi_N(x - n)}$ is reproducing kernel for V_N.*

Theorem 2.1[7] *For arbitrary $f \in V_N$, we have*

$$f(x) = \sum_{k\in Z} f(k + \frac{N+1}{2})S(x - k),$$

where $\hat{S}(\omega) = \frac{\hat{\varphi}_N(\omega)}{\sum \varphi_N(k+\frac{N+1}{2})e^{-ik\omega}}$ and V_N is spline subspaces generated by $\varphi_N = \chi_{[0,1]} * \cdots * \chi_{[0,1]} (N \geq 1$ convolutions).

Lemma 2.4 *Let $y(t) = \int_{-\infty}^{t} f(x)dx$ and $f \in V_N$, then $y \in V_{N+1}$.*

By Theorem 2.1 and Lemma 2.4, we have the following Theorem 2.2.

Theorem 2.2 *Let*

$$y_k = \int_{-\infty}^{k+\frac{1}{2}+1} f(t)dt, y_{k-1} = \int_{-\infty}^{k+\frac{1}{2}} f(t)dt, z_k = y_k - y_{k-1}.$$

Then for any $f \in V_N$, we have reconstruction formula

$$f(t) = \sum_{k\in Z} z_k h(t - k),$$

where h is defined by

$$h_k = h_{1k} + h_{k+1}, \hat{h}_1(\omega) = i\pi\omega \frac{\hat{\varphi}_{N+1}(\omega)}{\sum \varphi_{N+1}(k + \frac{N}{2} + 1)e^{-ik\omega}},$$

$h_k(\cdot) = h(\cdot - k)$ *and* $h_{1k}(\cdot) = h_1(\cdot - k)$.

Proof. From Lemma 2.4, we know $y \in V_{N+1}$. Then we use Theorem 2.1 to obtain

$$y(t) = \sum_{k\in Z} y(k + \frac{N}{2} + 1)S_1(t - k) \qquad (*),$$

where $\hat{S}_1(\omega) = \frac{\hat{\varphi}_{N+1}(\omega)}{\sum_k \varphi_{N+1}(k+\frac{N}{2}+1)e^{-ik\omega}}$. Through derivation calculus are acted in both sides of equality $(*)$, we can imply

$$f(t) = \sum_{k\in Z} y_k S_1'(t-k) = \sum_{k\in Z} y_k h_1(t-k) = \sum_{k\in Z} z_k h(t-k).$$

The above last equality derives from the assumption $h_k = h_{1k} + h_{k+1}$.

Remark 1. a.Actually, Theorem 2.2 show the reconstruction formula from regular incremental integral samples in V_N.

b. The above reconstruction formula can be rewritten as the following:

$$f(t) = \sum_{k\in Z} \int_{k+\frac{N}{2}+1}^{k+\frac{N}{2}} f(x)dx h(t-k) = \sum_{k\in Z} \langle f, u(\cdot - k)\rangle h(t-k),$$

where $u(\cdot) = \chi_{[\frac{N}{2}, \frac{N}{2}+1]}(\cdot)$. This induces that incremental integral samples can be regarded as a special weighted samples.

c.The interpolation function h is practical in application.

3 Reconstruction of Signal from Irregular Incremental Integral Samples on Spline Subspace

Next, we will study the reconstruction from irregular incremental integral samples. We will show the following Theorem 3.1 before we giving the main result.

Theorem 3.1 *Let sampling set $X = \{x_n\}$ be a real sequence and satisfy $\sup_i(x_{i+1} - x_i) = \delta < 1$. Then $\{k(\cdot, x_j)\}$ is frame for V_{N+1} and there exist dual frames $\{\tilde{k}_{x_j} : x_j \in X\} \subset V_{N+1}$ such that for any $f \in V_{N+1}$, we have*

$$f(x) = \sum_{j\in Z} \langle f, k_{x_j}\rangle \tilde{k}_{x_j}(x) = \sum_{j\in Z} f(x_j)\tilde{k}_{x_j}(x),$$

where $k(\cdot,\cdot)$as the same in Lemma 2.3.

Proof. From the discussion of [1, 3], we know Theorem 3.1 holds.

We make little changes for the proof of Theorem 2.2 to obtain the proof of the following Theorem 3.2. So we will omit its proof.

Theorem 3.2 *Let*

$$y_k = \int_{-\infty}^{x_k} f(t)dt, y_{k-1} = \int_{-\infty}^{x_{k-1}} f(t)dt, z_k = y_k - y_{k-1}$$

and sampling set $X = \{x_n\}$ *is real sequence with* $\sup_i(x_{i+1} - x_i) = \delta < 1$. *Then for any* $f \in V_N$ *we have reconstruction formula*

$$f(t) = \sum_{k \in Z} z_k \tilde{h}_{x_k}(x),$$

where $\{\tilde{h}_{x_k}\}$ *is given by* $h_{x_j} = \tilde{h}_{x_j} - \tilde{h}_{x_{j+1}} (j \in Z)$, $h_{x_j}(x) = \tilde{k}'_{x_j}(x)$ *is derivative of* $\tilde{k}_{x_j}(x)$ *and* $\tilde{k}_{x_j}(x)$ *as the same in Theorem 3.1.*

4 Conclusion

In this paper, we obtain reconstruction formula from samples of its integral in spline subspaces with the simple new method. And the reconstruction formula is more practical in application. Furthermore, we obtain reconstruction formula of signal from irregular incremental integral samples in spline subspace too. We conjecture the new method remains validity for a much large class of shift-invariant spaces. We will study it in future work.

References

1. Aldroubi, A.: Non-uniform weighted average sampling and reconstruction in shift-invariant and wavelet spaces, *Appl. Comput. Harmon. Anal* 13(2002)156-161.
2. Aldroubi, A., K. Gröhenig.: Beurling-Landau-type theorem for non-uniform sampling in shift invariant spline spaces. *J. Fourier. Anal. Appl,* 6(2000)93-103.
3. Aldroubi, A., K. Gröchenig.: Non-uniform sampling and reconstruction in shift-invariant spaces.,
   SIAM Rev 43(2001)585-620.
4. Aldroubi, A., Unser, M., Eden, M.: Cardinal spline filters: Stability and convergence to the ideal sinc interpolator," *Singal. Processing* 28(1992)127-138.
5. Chui, C. K.: An introduction to Wavelet, Academic Press, New York,1992
6. Liu, Y.: Irregular sampling for spline wavelet subspaces, *IEEE Trans. Inform. Theory* 42(1996)623-627.
7. Sun, W. C., Zhou., X. W.: Average sampling in spline subspaces. *Appl. Math. Letter,* 15(2002)233-237.
8. Sun, W. C., Zhou, X. W.: Reconstruction of bandlimited signals from local averages. *IEEE Trans. Inform. Theory,* 48(2002)2955-2963.
9. Wang, J.: Spline wavelets in numerical resolution of partial differential equations, *International Conference on Wavelet Analysis and its application, AMS/IP Studies in Advanced Mathematics,* 25(2002)257-276.
10. Xian, J., Lin, W.: Sampling and reconstruction in time-warped spaces and their applications. to appear in *Appl. Math. Comput,* 2004.

The Vectorized and Parallelized Solving of Markovian Models for Optical Networks

Beata Bylina and Jarosław Bylina

Department of Computer Science, Marie Curie-Skłodowska University
Pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland
{beatas, jmbylina}@hektor.umcs.lublin.pl

Abstract. The article presents two approaches to the WZ factorization – specific ones for solving Markov chains – and the results of their vectorization and parallelization.

1 Introduction

The probabilistic methods – especially Markov models – are the most useful ones to describe queueing models. A homogeneous continuous-time Markov chain can be described with one singular matrix $\mathbf{Q} = (q_{ij})_{i=1,\ldots,n}^{j=1,\ldots,n}$ called the transition rate matrix given by $q_{ij} = \lim_{\Delta t \to 0} \frac{p_{ij}(\Delta t)}{\Delta t}$ for $i \neq j$ and by $q_{ii} = -\sum_{j \neq i} q_{ij}$.

We are to find $\mathbf{x} = \pi^T$ – the vector of the stationary probabilities π_i that the system is in the state i at the time t – from:

$$\mathbf{Q}^T\mathbf{x} = 0, \qquad \mathbf{x} \geq 0, \qquad \mathbf{e}^T\mathbf{x} = 1, \quad \text{where } \mathbf{e} = (1, 1, \ldots, 1)^T. \tag{1}$$

2 The WZ Factorization

The WZ factorization is described in [3]. **A = WZ,** where (for an even n) $\mathbf{W} = (w_{ij})_{i=1,\ldots,n}^{j=1,\ldots,n}$ is shaped like a butterfly ($w_{ij} = 0$ for $i < j < n - i + 1$ and for $n - i - 1 < j < i$, $w_{ii} = 1$) and $\mathbf{Z} = (z_{ij})_{i=1,\ldots,n}^{j=1,\ldots,n}$ is shaped like a transposed butterfly ($w_{ij} = 0$ for $j < i < n - j$ and for $n - j < i < j$). After the factorization we can solve two linear systems: **Wc = b** and **Zx = c** instead of one **Ax = b.**

The sequential algorithm for solving the linear system with the WZ factorization is presented on the figure 1. On the figure 2 we present its vectorized counterpart (from [1]).

3 Replacing an Equation (RWZ)

The most intuitive approach to solving a homogenous linear system (1) is to replace an arbitrary equation of that system with the normalization equation $\mathbf{e}^T\mathbf{x} = 1$. Let $\widetilde{\mathbf{Q}}_p$ be the matrix $\mathbf{Q}$ with the pth column replaced with the vector **e.** Our modified system can be written $\widetilde{\mathbf{Q}}_p^T\mathbf{x} = \mathbf{e}_p$, where $\mathbf{e}_p = (\delta_{ip})_{i=1,\ldots,n}$.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 578–581, 2004.
© Springer-Verlag Berlin Heidelberg 2004

```
% elimination loop – steps of reduction from A to Z:
for k = 0 : m
  k2 = n-k-1;
  det = A(k,k)*A(k2,k2)-A(k2,k)*A(k,k2);
% finding elements of W
  for i = k+1 : k2-1
    wk1 = (A(k2,k)*A(i,k2)-A(k2,k2)*A(i,k))/det;
    wk2 = (A(k,k2)*A(i,k)-A(k,k)*A(i,k2))/det;
% updating A
    for j = k+1 : k2-1
      A(i,j) = A(i,j)+wk1*A(k,j)+wk2*A(k2,j);
% updating b
    b(i) = b(i)+wk1*b(k)+wk2*b(k2);
% finding x
for j = m : 0
% solving a 2 × 2 linear system
  j2 = n-j-1;
  det = A(j,j)*A(j2,j2)-A(j2,j)*A(j,j2);
  x(j) = (b(j)*A(j2,j2)-b(j2)*A(j,j2))/det;
  x(j2) = (b(j2)*A(j,j)-b(j)*A(j2,j))/det;
% updating b
  for i = j-1 : 0
    i2 = n-i-1
    b(i) = b(i)-x(j)*A(i,j)-x(j2)*A(i,j2);
    b(i2) = b(i2)-x(j)*A(i2,j)-x(j2)*A(i2,j2);
```

Fig. 1. The sequential algorithm for the WZ factorization

```
% elimination loop – steps of reduction from A to Z:
for k = 0 : m
  k2 = n-k-1;
  det = A(k,k)*A(k2,k2)-A(k2,k)*A(k,k2);
  for i = k+1 : k2-1
% finding elements of W
    wk1 = (A(k2,k)*A(i,k2)-A(k2,k2)*A(i,k))/det;
    wk2 = (A(k,k2)*A(i,k)-A(k,k)*A(i,k2))/det;
% updating A
    A(i,k+1:k2-1) = A(i,k+1:k2-1)+wk1*A(k,k+1:k2-1)+wk2*A(k2,k+1:k2-1);
% updating b
    b(i) = b(i)+wk1*b(k)+wk2*b(k2);
% finding x
for j = m : 0
% solving a 2 × 2 linear system – as on figure 1
% updating the upper and the lower part of b
  b(0:j-1) = b(0:j-1)-x(j)*A(0:j-1,j)-x(j2)*A(0:j-1,j2);
  b(n-j:n-1) = b(n-j:n-1)-x(j)*A(n-j:n-1,j)-x(j2)*A(n-j:n-1,j2);
```

Fig. 2. The vectorized algorithm for the WZ factorization

Let $\widetilde{\mathbf{Q}}_p^T = \widetilde{\mathbf{W}}\widetilde{\mathbf{Z}}$. Setting $\widetilde{\mathbf{Z}}\mathbf{x} = \mathbf{y}$ in the system $\widetilde{\mathbf{W}}\widetilde{\mathbf{Z}}\mathbf{x} = \mathbf{e}_p$ we get $\widetilde{\mathbf{W}}\mathbf{y} = \mathbf{e}_p$ from which it is obvious that $\mathbf{y} = \mathbf{e}_p$. So now we are to solve the system $\widetilde{\mathbf{Z}}\mathbf{x} = \mathbf{e}_p$.

This approach is likely to yield a less accurate result than the next one (section 4). When we compute x_p and x_q they will be contaminated with the round-off errors from all of the previous elimination steps. Moreover, this will propagate throughout the next backsubstitution steps.

4 Removing an Equation (DWZ)

Another approach is to remove an equation. We know that the rank of $\mathbf{Q}$ is $(n-1)$ – that is one of the equations can be written as a linear combinantion of other equations. If we drop an equation we get a linear system of $(n-1)$ equations with n unknowns (and the normalization equation).

In this approach we divide (after [4]) our matrix in blocks $\mathbf{Q}^T = \begin{bmatrix} \mathbf{B} & \mathbf{d} \\ \mathbf{c}^T & f \end{bmatrix}$ where $\mathbf{B}$ is a nonsingular matrix of the size $(n-1)$, $\mathbf{c}$ and $\mathbf{d}$ are $(n-1)$-element vectors and f is a real number.

Let us assign $x_n = 1$, now $\mathbf{x}^T = (\widehat{\mathbf{x}}^T, 1)$ and our equation (1) gives the equatoins: $\mathbf{B}\widehat{\mathbf{x}} + \mathbf{d} = 0$ and $\mathbf{c}^T\widehat{\mathbf{x}} + f = 0$.

Now we can solve the linear system without the last equation, that is only $\mathbf{B}\widehat{\mathbf{x}} = -\mathbf{d}$. We solve it using the WZ factorization – the matrix $\mathbf{B}$ is factorized: $\mathbf{B} = \mathbf{WZ}$ and the equations $\mathbf{Wy} = -\mathbf{d}$ and $\mathbf{Z}\widehat{\mathbf{x}} = \mathbf{y}$ are solved. Now we must normalize the solution vector $\mathbf{x}^T = (\widehat{\mathbf{x}}^T, 1)$.

Of course, whichever equation can be dropped, not only the last.

5 The Example, the Implementation, and the Experiments

A queuing model of an optical network edge node is presented in [2]. Information (electronical) packages arrive in a buffer (of an optical switch) of capacity of $N = 250$ blocks. The packages are of different sizes (sizes are integer, from one block up to 20 blocks). When the buffor is filled, the bigger (optical) package is formed and sent. The buffer is then emptied. We don't want to divide information from one electronical package between two optical packages so when an arriving package is too big to fit into the buffer we send an uncompleted optical package and package just received starts a new optical package. An uncompleted optical package is also sent when a given timeout is over. Arriving packages stream has a Poisson distribution and the probability that the received packages have i $(i = 1, \ldots, 20)$ blocks is $p_i = 0.05$. In our model the timeout is approximated with an Erlang distribution consisting of 10 phases of exponential distribution because we want to preserve a Markovian form of this model.

In our experiments the transition rates matrix ($\mathbf{Q}$) was not very big (only 2491 states) so we decided to store it in a traditional, two-dimensional array

with a lot of zeroes. This storing scheme is rather space consuming, but is the best when we have enough space and the computation time matters.

The algorithms **RWZ** and **DWZ** were implemented for the single precision numbers with the use of the language C. The programs were compiled with the *icc* (*Intel C Compiler*). We denoted the algorithms **RWZ** and **DWZ** improved with the vectorized algorithm presented in the figure 2 by **VRWZ** and **VDWZ,** respectively. The algorithms **VRWZ** and **VDWZ** were implemented with the use of the *BLAS1* functions from the *mkl* (Intel's *Mathematics Kernel Library*). All the programs were tested on a Pentium III 733 MHz machine.

Table 1. The performance and the residual for the tested algorithms

algorithm	$\|\|\mathbf{Q}^T\mathbf{x}\|\|_2$	time A [s]	time B [s]
DWZ	3.84383e–07	239.98	228.65
VDWZ	5.69805e–07	9.84	231.82
RWZ	1.91620e–06	232.96	229.08
VRWZ	1.93863e–06	15.47	220.02

All the described algorithms were tested as not parallelized ones and as parallelized (with the use of the *OpenMP* standard) algorithms – but the parallelization did not give the significiant improvement in performance (no more than 5%), so in the table 1 we present the residual (here: $||\mathbf{Q}^T\mathbf{x}||_2$) and performance (as **time A**) of the not parallelized ones only.

To better understand such a big improvement in performance (with the use of *BLAS1* only) we made some tests with the same algorithms but with other matrices. Namely we prepared some random matrices with all non-zero elements. The performance for such dense matrices (**time B** in table 1) for not vectorized algorithms was similar to performance for our sparse (54782 non-zeroes for $2491^2 = 6205081$ elements) matrix. However, for vectorized ones the performance for the sparse matrix was much better than the performance for the dense ones – because of the special treatment of huge number of zeroes by the *mkl* and the processor.

References

1. Bylina, B.: Solving lineaar systems with vectorized WZ factorization. Annales UMCS Informatica **1** (2003) 5–13
2. Domańska, J., Czachórski, T.: Model formowania pakietów w węźle brzegowym sieci optycznej. Studia Informatica **2A(53)** (2003) 51–62 (in Polish)
3. Evans, D.J., Hatzopoulos, M.: The parallel solution of linear system. Int. J. Comp. Math. **7** (1979) 227–238
4. Stewart, W.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press, Chichester, West Sussex (1994)

A Parallel Splitting up Algorithm for the Determination of an Unknown Coefficient in Multi Dimensional Parabolic Problem

Daoud S. Daoud and D. Subasi

Department of Mathematics, Eastern Mediterranean Univ., Famaqusta, North Cyprus – Mersin 10 Turkey
daoud.daoud@emu.edu.tr

Abstract. One of the global approach for solving the two dimensional inverse parabolic problem is the predictor corrector which takes place for evaluating the pair (u, p) and adjusting the evaluation for the desired accuracy. In this work we will present a new parallel algorithm(of non iterative type) for solving two or higher dimensional inverse control problem.

1 Introduction

In this article we study the finite difference method for approximating the source control parameter $p(t)$, and the solution $u = u(x, y, t)$ of the following inverse model problem defined by; Find $u = u(x, y, t)$ and $p = p(t)$ which satisfy

$$\begin{aligned} u_t = &\quad \triangle u + p(t)u + \phi(x, y, t) \quad \text{in } \Omega \times [0, T] \\ u(x, y, 0) = &\ f(x, y) \quad (x, y) \in \Omega \\ u(x, y, t) = &\ g(x, y, t) \quad on \partial\Omega \times [0, T] \end{aligned} \tag{1}$$

subject to the over specification condition at a point (x^*, y^*) in the spatial domain Ω given by;

$$u(x^*, y^*, t) = E(t) \tag{2}$$

One of the global techniques for solving problem (1) and (2) is the predictor corrector finite difference scheme introduced by J.R. Canon et al. [1].

The proposed splitting algorithm is defined to split the problem in accordance to the spatial variables into independent one dimensional sub problems which classified as *Additive Splitting Method* and its of non iterative type.

2 Parallel Splitting up Method

The general definition of the multi dimensional parabolic problem is given by

$$\frac{\partial u}{\partial t} = \sum_{j}^{m} \frac{\partial}{\partial x_j}(a_j \frac{\partial u}{\partial x_j}) + f \quad \text{in} \quad \Omega \times [0, T], \tag{3}$$

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 582–585, 2004.
© Springer-Verlag Berlin Heidelberg 2004

where $u(0) = u_0$, with Dirichlet boundary conditions over $\partial\Omega \times [0, T]$.

The domain Ω is a bounded domain in R^m $(m = 2, 3)$, a_j are scalars, and the assembled coefficients of the central difference discretization for $\frac{\partial}{\partial x_j}(a_j \frac{\partial u}{\partial x_j})$, and for $\sum_{j=1}^{m} \frac{\partial}{\partial x_j}(a_j \frac{\partial u}{\partial x_j})$ are denoted by A_j and A, respectively. The key to the splitting algorithm is the splitting up algorithm by Lu, et. al. [3], given by

Algorithm 1 *First order implicit splitting up method.*
Step1. Let $\delta t > 0$ be any time step. For $j = 1, \ldots m$ the solution $v^{n+\frac{j}{2m}}$ is given by:

$$(I - m\delta t A_j) v^{n+\frac{j}{2m}} = v^n + m\delta t f^{n+1} \tag{4}$$

Step 2. $v^{n+1} = \frac{1}{m} \sum_{j=1}^{m} v^{n+\frac{j}{2m}}$.
Step 3. If $T < (n+1)\delta t$ go to step 1, otherwise stop.

Theorem 2. *The splitting up method given by algorithm 1 is globally a first order splitting in δt.*

Proof. For the proof see [2]

Theorem 3. *With respect to each spatial variable x_j, $j = 1, \ldots m$ the solution $v^{n+\frac{j}{2m}}$ given by step 1, and the solution v^{n+1} by step 2 of algorithm 1 is unconditionally stable in ℓ_2 norm.*

Proof. For the proof see [2]

3 Inverse Control Problem Splitting Algorithm

In this section we will present the parallel splitting up algorithm for the inverse control problem (1-2).

Algorithm 4 *Inverse Problem Splitting up Method.*
Step1. Let $\delta t > 0$ be any time step. For $j = 1, \ldots m$ solve for $v^{n+\frac{j}{2m}}$:

$$\frac{v^{n+j/2m} - v^n}{m\delta t} = A_j v^{n+j/2m} + \frac{1}{m} p^{n+1} v^{n+j/2m} + \frac{1}{m} \phi^{n+1} \tag{5}$$

Step 2. $v^{n+1} = \frac{1}{m} \sum_{j=1}^{m} v^{n+\frac{j}{2m}}$.
Step 3. If $T < (n+1)\delta t$ go to step 1, otherwise stop.

In the predictor corrector algorithm we firstly assume that $v(x, y, t)$ and $p(t)$ are given at the time $t = n\delta t$, for $\delta t > 0$, and let $l = 0, 1, \ldots$ be the correction level. The generic splitting up and Predictor-Corrector algorithm is given by

Algorithm 5 *Predictor Corrector Method for Inverse Control Problem*
Step 1: For $l = 0$ let $p^{n+1(l)} = p^n$, solve the following splitted one dimensional problem

$$\begin{aligned} \frac{v^{n+1/4} - v^n}{2\delta t} &= A_x v^{n+1/4} + \tfrac{1}{2} p^{n+1(0)} v^{n+1/4} + \tfrac{1}{2} \phi^{n+1}, \\ \frac{v^{n+1/2} - v^n}{2\delta t} &= A_y v^{n+1/2} + \tfrac{1}{2} p^{n+1(0)} v^{n+1/2} + \tfrac{1}{2} \phi^{n+1}, \end{aligned} \tag{6}$$

the solution $v^{n+1(l)} = \frac{1}{2}(v^{n+1/4} + v^{n+1/2})$, *where* A_x, *and* A_y *represent the assembled coefficients of the central difference approximation for* u_{xx}, *and* u_{yy} *respectively.*

Step 2: Since $v(x_{i^*}, y_{i^*}, t) = E(t)$, where $(x_{i^*}, y_{j^*}) = (x^*, y^*)$ *i.e* $v(x_{i^*}, y_{j^*}, .)^{n+1(l)} = E^{n+1(l)}$, *then problem (1) for* $E(t)$ *is given by*

$$E'(t) = \triangle v(x_{i^*}, y_{j^*}, t) + p(t)E(t) + \phi(x_{i^*}, y_{j^*}, t), \tag{7}$$

then the splitting up of (7) in accordance to the splitting up algorithm is given by

$$\begin{aligned} \frac{E^{n+1/4(l)} - E^n}{2\delta t} &= v_{xx}(x_{i^*}, y_{j^*}, .)^{n+1(l)} + \tfrac{1}{2} p x^{n+1(l+1)} E^{n+1/4(l)} + \tfrac{1}{2}\phi^{n+1}, \\ \frac{E^{n+1/2(l)} - E^n}{2\delta t} &= v_{yy}(x_{i^*}, y_{j^*}, .)^{n+1(l)} + \tfrac{1}{2} p y^{n+1(l+1)} E^{n+1/2(l)} + \tfrac{1}{2}\phi^{n+1}. \end{aligned} \tag{8}$$

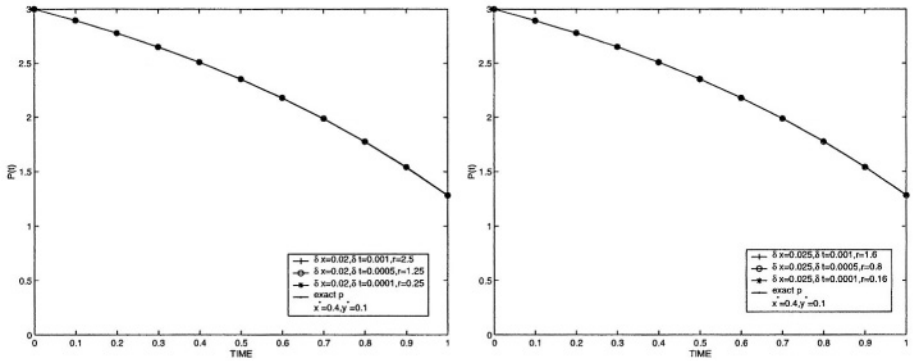


Fig. 1. The exact value and the estimated values of $p(t)$ for different δx and δt for problem 1, simulated over [0,1]

Table 1. The error due to the solution of problem 1 for $x^* = 0.4$ and $y^* = 0.1$

	$\delta x = 0.02$				
r	2.5	1.6	1.25	0.8	0.25
error	4.63008e(-4)	2.52485e(-4)	2.07695e(-4)	8.93606e(-5)	6.05821e(-5)

The solution of (8) for px and py is given by

$$\begin{aligned} p x^{n+1(l+1)} &= \frac{(E^{n+1/4(l)} - E^n) - 2\delta t v_{xx}(x_{i^*}, y_{j^*}, .)^{n+1(l)} - \delta t \phi^{n+1}}{\delta t E^{n+1/4(l)}}, \\ p y^{n+1(l+1)} &= \frac{(E^{n+1/2(l)} - E^n) - 2\delta t v_{yy}(x_{i^*}, y_{j^*}, .)^{n+1(l)} - \delta t \phi^{n+1}}{\delta t E^{n+1/2(l)}}, \end{aligned} \tag{9}$$

$$p^{n+1(l+1)} = \frac{1}{2}(p x^{n+1(l+1)} + p y^{n+1(l+1)}), \tag{10}$$

where px and py represent the correction with respect to x and y variables respectively.

Step3: Update the solution $v^{n+1(l)}$ using step 1 for $l = 1, 2, \ldots$ and considers the latest value of $p^{n+1(l+1)}$.

The value of $p^{n+1(l+1)}$ will then be adjusted till it satisfies the pre assigned tolerance(e.g. $|p^{n+1(l+1)} - p^{n+1(l)}| \leq 10^{-4}$) for the evaluation of the pair (v, p) over the time interval $[t_n, t_{n+1}]$.

Theorem 6. *The splitting up algorithm 4 of the inverse problem is a globally a first order algorithm in δt.*

4 Numerical Experiments and Conclusion

In this section we report the numerical results from solving an inverse control problem using algorithm 5 described in section 3 for evaluating the pair (u, p). In general we take an exact solution $u(x, y, t)$, source function $p(t)$ we refer to the general problem (1)-(2) to formulate our model problems using algorithm 5 to compute the value of the solution v^{n+1} and $p(t)$, and to compare the numerical results with the exact values.

Model problem 1. We consider the following problem

$$u_t = \triangle u + p(t)u + \phi(x, y, t),$$

defined over $\Omega \times [0, 1]$ with initial conditions $u(x, y, 0) = \sin(\frac{\pi}{4}(x + 2y))$ and the boundary conditions given by

$$u(0, y, t) = e^t \sin(\tfrac{\pi}{2} y) \ , \ u(1, y, t) = e^t \sin(\tfrac{\pi}{4}(1 + 2y)),$$
$$u(x, 0, t) = e^t \sin(\tfrac{\pi}{4} x) \ , \ u(x, 1, t) = e^t \sin(\tfrac{\pi}{4}(x + 2)),$$

From the numerical results we demonstrate the accuracy of the evaluated function $p(t)$ over the whole time interval [0,1] independently on the time stepping or the spatial mesh spacings.

References

1. J.R. Cannon, Yanping Lin, S.Wang, Determination of source parameter in parabolic equations. *Meccanica,* vol. 27:85-94,1992.
2. D.S. Daoud, Non Overlapping Domain Decomposition and Additive Splitting up Methods for Multi Dimensional Parabolic Problem, *Numer. Methods Partial Differerntial Eg.* to appear.
3. T.Lu, P. Neittaanmaki, and X-C Tai, A parallel splitting up method and its application to Navier-Stokes equations. *Applied Mathematics Letters,* Vol. 4, No. 2:25-29, 1991.

A-Posteriori Error Analysis of a Mixed Method for Linear Parabolic Problem

M.I. Asensio, J.M. Cascón, and L. Ferragut

Departamento de Matemática Aplicada, Universidad de Salamanca,
37008 Salamanca, Spain
{mas, casbar, ferragut}@usal.es

Abstract. In this paper we present a-posteriori error estimator for the mixed formulation of linear parabolic problem, and we use them in designing an efficient adaptive algorithm. Our space-time discretization consist of lowest order Raviart-Thomas finite element over graded meshes, and discontinuous Galerkin method with varying time-steps.

1 Introduction

A-posteriori error estimates are an essential component in the design of reliable and efficient adaptive algorithms for the numerical solutions of PDEs. At the same time, mixed formulations are suitable for certain problems, as they allow to directly approach certain derivatives of the solution.

In this paper we introduce an a-posteriori error estimation for mixed formulation of the linear parabolic problem obtaining,

$$\|u - U\|_{L^\infty(0,T;L^2(\Omega))} + \|\mathbf{p} - \mathbf{P}\|_{L^2(0,T;H^{-1}(\mathrm{div},\Omega))} \leq \mathcal{E}(u_0, f, T, \Omega; U, \mathbf{P}, h, k),$$

where u is the scalar variable, and $\mathbf{p}$ its gradient. Capital letters represent the numerical approximations obtained with the lowest order Raviart-Thomas finite element in space and a variable step discontinuous Galerkin method in time. The estimator $\mathcal{E}$ is computable in terms of: given data u_0, f, Ω, T, computed solutions U and $\mathbf{P}$, meshsize h and time step k.

As in [4], we obtain the error representation formulas of $u - U$ and $\mathbf{p} - \mathbf{P}$ from the residual equations. The evaluation of the residual in the corresponding norms and the stability properties of the associated dual problem allow the estimations to be concluded. The scalar error bound is an extension to the mixed formulation of the results developed in [3]. For the error estimation of $\mathbf{p} - \mathbf{P}$ we use the Helmholtz descomposition in $L^2(\Omega; \mathbb{R}^2)$, as in [1] and [2] for the stationary case.

2 Continuous Problem

Let $\Omega \subset \mathbb{R}^2$ be a bounded convex polyhedral domain, let $T > 0$ be the final time, and set $Q_T := \Omega \times (0, T)$ and $\Gamma_T := \partial\Omega \times (0, T)$. Given an initial condition

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 586–589, 2004.
© Springer-Verlag Berlin Heidelberg 2004

u_0 and a source term $f(x,t)$, let $(u,\mathbf{p})$ be the solution of the problem,

$$u_t + \operatorname{div}\mathbf{p} = f \quad \text{in} \quad Q_T, \tag{1}$$

$$\mathbf{p} + \nabla u = 0 \quad \text{in} \quad Q_T, \tag{2}$$

$$\nabla u \cdot \boldsymbol{\nu} = 0 \quad \text{on} \quad \Gamma_T, \tag{3}$$

$$u(.,0) = u_0 \quad \text{in} \quad \Omega, \tag{4}$$

where $\boldsymbol{\nu}$ is the unit normal vector to $\partial\Omega$.

In the following, $(\cdot,\cdot)$ denotes the inner product in $L^2(\Omega)$ or $L^2(\Omega;\mathbb{R}^2)$ and $<\cdot\,,\cdot>$ denotes the duality pairing between the corresponding spaces. We define the space $H_0(\operatorname{div},\Omega)$ as,

$$H_0(\operatorname{div},\Omega) := \left\{\mathbf{q} \in L^2(\Omega;\mathbb{R}^2) : \operatorname{div}\mathbf{q} \in L^2(\Omega),\ \ <\mathbf{q}\cdot\boldsymbol{\nu}, v>=0\ \ \forall v \in H^1(\Omega)\right\}$$

and denote by $H^{-1}(\operatorname{div},\Omega)$ its dual space. For the corresponding variational formulation of this problem we use the bilinear operator B, defined on $L^2(\Omega)\times H_0(\operatorname{div},\Omega)$ as,

$$\mathrm{B}(w,\mathbf{q};r,\mathbf{s}) = (w,r) + (\operatorname{div}\mathbf{q},r) + (\operatorname{div}\mathbf{s},w) - (\mathbf{q},\mathbf{s}).$$

Adding the equations (1) and (2) we obtain the following variational problem,

Continuous problem. *Find* u *and* $\mathbf{p}$ *such that*

$$u \in H^1(0,T;L^2(\Omega)),\quad \mathbf{p} \in L^2(0,T;H_0(\operatorname{div},\Omega)),\quad u(.,0)=u_0, \tag{5}$$

$$\mathrm{B}(u_t,\mathbf{p};v,\mathbf{q}) = (f,v)\quad \forall (v,\mathbf{q}) \in L^2(\Omega)\times H_0(\operatorname{div},\Omega)\ \text{a.e. } t\in(0,T). \tag{6}$$

3 Discretization

The fully discrete problem combines finite elements in space with the Galerkin discontinuous method in time.

Let k_n denotes the time step at the n-th step and set $t_n = \sum_{i=1}^{n} k_i$. Let N be the total number of time steps, that is, $t_N = T$. To each time step, $(t_{n-1},t_n]$ we associate $(\mathcal{T}_n,\mathbb{M}_n,\mathbb{X}_n)$, where $\mathcal{T}_n = \{\mathcal{S}_n\}$ is a uniformly regular partition of Ω in triangles. We denote h_n a piecewise constant function with $h_{n|\mathcal{S}} = \operatorname{diam}(\mathcal{S})$ for all $\mathcal{S}\in\mathcal{T}_n$, $\mathcal{B}_n$ is the set of all interior edges e of $\mathcal{T}_n$; h_e stands for the size of e. The mesh $\mathcal{T}_n$ is obtained from $\mathcal{T}_{n-1}$ by refining/coarsening. $\mathbb{M}_n \in L^2(\Omega)$ is the usual space of piecewise constant finite elements over $\mathcal{T}_n$. Finally, $\mathbb{X}_n \in H(\operatorname{div},\Omega)$ is the lowest order Raviart-Thomas finite element. Then, we have the problem,

Discrete problem. *Given* $U^{n-1}\in\mathbb{M}_{n-1}$, $\mathbf{P}^{n-1}\in\mathbb{X}_{n-1}$, *then* k_{n-1} *and* $\mathcal{T}_{n-1}$ *are modified as described in section* §5 *to get* k_n *and* $\mathcal{T}_n$ *and thereafter* $U^n\in\mathbb{M}_n$, $\mathbf{P}^n\in\mathbb{X}_n$ *computed according to,*

$$\frac{1}{k_n}(U^n - \pi_\Omega U^{n-1}, V) + (\operatorname{div}\mathbf{P}^n, V) = \frac{1}{k_n}\int_{t_{n-1}}^{t_n}(f,V)dt, \qquad \forall V\in\mathbb{M}_n, \tag{7}$$

$$(\mathbf{P}^n,\mathbf{Q}) - (\operatorname{div}\mathbf{Q},U^n) = 0, \qquad \forall \mathbf{Q}\in\mathbb{X}_n, \tag{8}$$

where π_Ω^n denotes the usual $L^2(\Omega)$-projection operator over $\mathbb{M}_n$.

4 A-Posteriori Estimator

In the following Theorems we present the a-posteriori error estimates for the schemes developed bellow.

Theorem 1. *Let* $(u, \mathbf{p})$ *be the solution of (5-6) and* $(U, \mathbf{P})$ *the solution of (7-8). If* Ω *is convex, for* $N \geq 1$,

$$\left\|e_u^{t_N}\right\|_{L^2(\Omega)} \leq \left\|e_u^0\right\|_{L^2(\Omega)} + \sum_{i=1}^{5} C_i \max_{1\leq n\leq N} \mathcal{E}_i,$$

with,

$$\mathcal{E}_1 := \left\|U^n - \pi_\Omega^n U^{n-1}\right\|_{L^2(\Omega)} \qquad \mathcal{E}_2 := \left\|U^{n-1} - \pi_\Omega^n U^{n-1})\right\|_{L^2(\Omega)},$$

$$\mathcal{E}_3 := \|h_n \mathbf{P}\|_{L^2(\Omega)}, \qquad \mathcal{E}_4 := \max_{t\in[t_{n-1},t_n)} \|h_n(f - div\,\mathbf{P})\|_{L^2(\Omega)},$$

$$\mathcal{E}_5 := \int_{t_{n-1}}^{t_n} \|f - div\,\mathbf{P}\|_{L^2(\Omega)} dt.$$

Theorem 2. *Let* $(u, \mathbf{p})$ *be the solution of (5-6) and* $(U, \mathbf{P})$ *the solution of (7-8). Let* J_τ *be the tangential jump of* $\mathbf{P} \in \mathbb{X}$ *along* e. *If* Ω *is convex, for* $N \geq 1$,

$$\|\mathbf{E_P}\|_{L^2(0,T;H^{-1}(div,\Omega))} \leq \left\|e_u^0\right\|_{L^2(\Omega)} + \sum_{i=6}^{11} C_i \mathcal{E}_i,$$

where,

$$\mathcal{E}_6 := \left(\sum_{n=1}^{N} k_n \sum_{e\in\mathcal{B}_n} h_e \|J_\tau\|^2_{L^2(e)}\right)^{\frac{1}{2}}, \mathcal{E}_7 := \left(\sum_{n=1}^{N} k_n \left\|U^n - \pi_\Omega U^{n-1}\right\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}},$$

$$\mathcal{E}_8 := \left(\sum_{n=1}^{N} k_n \left\|U^{n-1} - \pi_\Omega U^{n-1}\right\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}}, \mathcal{E}_9 := \left(\sum_{n=1}^{N} \int_{t_{n-1}}^{t_n} \|h_n(f - div\,\mathbf{P})\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}},$$

$$\mathcal{E}_{10} := \left[\sum_{n=1}^{N} \left(\int_{t_{n-1}}^{t_n} k_n^{1/2} \|f - div\,\mathbf{P}\|_{L^2(\Omega)}\right)^2\right]^{\frac{1}{2}}, \mathcal{E}_{11} := \left(\sum_{n=1}^{N} k_n \|h_n \mathbf{P}\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}}.$$

5 Adaptive Algorithm and Numerical Examples

We propose two adaptive strategies based on the former estimates. The goal is to determine the time steps and the meshes in order to obtain a uniform error distribution. As usually in parabolic problem we distinguish four kinds of terms in the a-posteriori estimate: initial error, space discretization error, coarsening error and time discretization error,

$$\mathcal{E}_n \leq \mathcal{E}_0 + \mathcal{E}_{h,n} + \mathcal{E}_{c,n} + \mathcal{E}_{k,n}.$$

We also select the parameters Γ_0, Γ_h, Γ_k,verifying, $\Gamma_0 + \Gamma_h + \Gamma_k \leq 1$, in a way that given a total error tolerance *tol*, the adaptive algorithm tries to select time steps and meshes verifying for all n,

$$\mathcal{E}_n \sim tol, \qquad \mathcal{E}_0 \sim \Gamma_0\, tol, \qquad \mathcal{E}_{h,n} + \mathcal{E}_{c,n} \sim \Gamma_h\, tol, \qquad \mathcal{E}_{k,n} \sim \Gamma_k\, tol.$$

The algorithm begins with an initial time step k_0. Given $\theta_1 \in (0,1)$ and $\theta_2 \in (0,\theta_1)$, if $\mathcal{E}_{k,n} > \Gamma_k\, tol$, then the time step diminishes in a factor $\delta_1 \in (0,1)$; on the contrary, if $\mathcal{E}_k < \Gamma_k\, tol$, the time step increases in a factor $\delta_2 > 1$. For the space refinement we use an error equidistribution strategy (see [3]).

We use an implicit adaptive strategy, that is for each time step we start from the previous step mesh and repeat the process solving-estimating-adapting until the estimated error being below the tolerance.

In the poster session we will present several examples showing that the proposed method is efficient and reliable. We compare the estimators with the exact errors, and we obtain an excelent correlation coefficient number (≈ 1). We also study the adaptation in time and in space.

The numerical experiments have been designed with the finite element toolbox ALBERT [5], extended with new designed tools for the lowest order Raviart-Thomas finite element and the a-posteriori error estimator developed in this paper.

Conclusions

We obtain a-posteriori error estimators for the mixed formulation of linear parabolic problem and we use them in designing an efficient adaptive algorithm.

Acknowledgements. Research partially supported by REN2001-0925-03-03, Ministerio de Ciencia y Tecnología (Spain) and SA089/01, Junta de Castilla y León (Spain).

References

1. A. Alonso, Error estimators for a mixed method, Numer. Math., Vol. 74 (1994), n^0 4, pp. 385-395.
2. C. Cartensen, A posteriori error estimate for mixed finite element method, Math. Comp., Vol. 66 (1997), n^0 218, pp. 465-776.
3. K. Eriksson and C. Johnson, Adaptive Finite Element Methods For Parabolic Problems *I*: A Linear Model Problem, SIAM J. Numer. Anal., Vol. 28 (1991), n^0 1, pp. 43-77.
4. R. H. Nochetto , A. Schmidt, and C.Verdi, A posterioi error estimation and adaptivity for degenerate parabolic problems, Math. Comp., Vol. 69, (2000), n^0 229, pp. 1-24.
5. A. Schmidt and K.G. Siebert, ALBERT: An adaptive hierarchical finite element toolbox, Preprint 06/2000, Freiburg (2000).

Analysis of Parallel Numerical Libraries to Solve the 3D Electron Continuity Equation

Natalia Seoane and A.J. García-Loureiro

University of Santiago de Compostela, Dept. Electronics and Computer Science,
Campus Sur, 15782 Santiago de Compostela, Spain
Phone: +34 981 563100 Ext. 13568, Fax: +34 981 528012
natalia@dec.usc.es, antonio@dec.usc.es

Abstract. In this paper we show an analysis of different parallel numerical libraries for solving the linear systems associated to the electron continuity equation from the 3D simulation of the semiconductor devices. We use domain decomposition techniques, such as Additive Schwarz, Multicolor SOR or Schur Complement methods, in order to find the best method of resolution considering the minimization of the execution time. The results were obtained in a Beowulf system with Myrinet 2000 network with MPI standard for message–passing communication.

1 Introduction

In this paper an analysis is presented of different parallel numerical libraries employed in resolution of sparse linear systems in 3D semiconductor device simulation. The main goal of this analysis is the selection of parameters to minimize execution time. The sparse matrices employed in our analysis are those resulting from the discretization of electron continuity equation for HEMT devices. In our case, these matrices have 29012 rows and 398102 nonzero elements. A similar study could be performed for any other type of semiconductor devices. Our results have been obtained in a Beowulf system with Myrinet 2000 network, using MPI for message–passing communication.

2 Numerical Libraries

The working sets are numerical libraries based on direct methods, SuperLU, and iterative methods, PSPARSLIB, PETSc and Aztec, analyzing in each case the parameters with the largest impact on execution time, such as solvers based on Krylov methods, parallel preconditioners, fill–in processes in Incomplete LU Factorizations, the Krylov subspace size and the number of processors employed.

2.1 SuperLU

SuperLU[1] is a general purpose library for the direct solution of large, sparse linear systems. The library performs a LU decomposition with partial pivoting

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 590–593, 2004.
© Springer-Verlag Berlin Heidelberg 2004

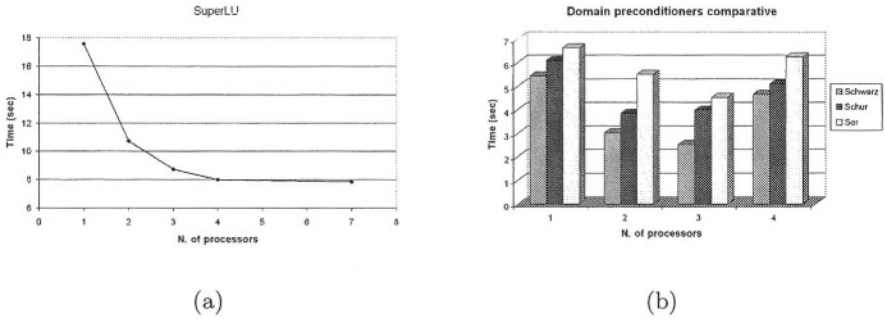


Fig. 1. (a) Factorization time for SuperLU library, (b) Comparison between the Additive Schwarz, Multicolor SOR and Schur Complement methods for PSPARSLIB

and triangular system solves through forward and back substitution. We can choose between sequential or parallel versions.

Using the distributed memory version, we analyze this library, solving in parallel a complete LU Factorization. Solving time decreases with the number of processors employed, which is becoming less marked (Figure 1(a)). For this type of matrix the use of this library is not very profitable in comparison with any of the analized libraries based on iterative methods.

2.2 PSPARSLIB

The solvers included in PSPARSLIB[2] are CG, GMRES, FGMRES, DQGMRES, BCGSTAB and TFQMR. It uses domain decomposition preconditioners[3], such as Additive Schwarz, Multicolor SOR and Schur complement methods [4].

Making a solver comparison, we find that GMRES obtain the lowest solving times. From the available preconditioners, Additive Schwarz is the most appropriate, as can be deduced in Fig. 1(b), where a comparison among the preconditioners named above is shown. For the computation of the internal points we use an incomplete LU Factorization depending on two parameters: the fill value ($2 \cdot fill$ will be the maximum number of fill–in elements) and a drop tolerance (fixed to $10e^{-4}$). The lowest solving times were obtained for a value of $fill = 25$. An example of this can be observed in Fig. 2(a), where the influence of fill value in solving time is represented for GMRES. The influence of Krylov subspace size is very small, since variations of 5% in the solving time are found by changing two magnitude orders in the Krylov subspace size value. With respect to the number of processors, the tendency shows minimum execution times with 3 processors. There are several reasons because using 4 processors is not a good choice. Firstly, communication times between processors become increasingly high when they are compared to the computational time. This is due to the size of local matrices which is getting too small. Moreover, reducing the size of matrices involves less

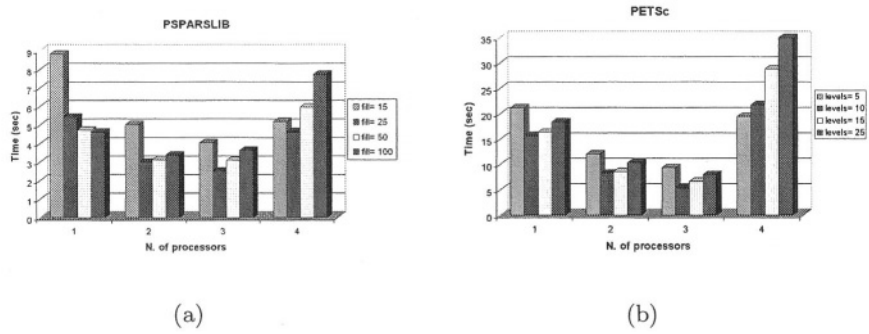


Fig. 2. Dependence of solving time with the fill value for GMRES solver with Additive Schwarz preconditioning for: (a) the PSPARSLIB library and (b) the PETSc library

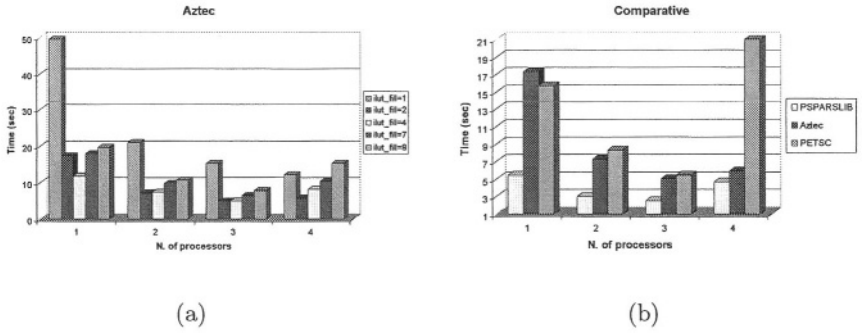


Fig. 3. (a) Dependence of solving time with the ilut_fill parameter for the Aztec library, (b) Comparison between PSPARSLIB, PETSc and Aztec for the best solving times

number of internal nodes and an increase of interface nodes. This is translated into a higher cost in both communication and computation times.

2.3 PETSc

PETSc[5] includes several solvers such as CG, GMRES, TCQMR, BCGS and TFQMR. Some preconditioners included in PETSc are Jacobi, Block Jacobi, SOR and Additive Schwarz. From all of the solvers, we are only going to study those common to the ones existing in PSPARSLIB in order to be able to make a comparative study. For the same reason, we choose the Additive Schwarz preconditioner, solving the internal points through an incomplete LU Factorization depending on a certain level of fill (*level* is the column number round the diagonal in which the fill–in is allowed).

The lowest solving times are found for the GMRES solver, when the levels of fill are between 10 and 12. With respect to the influence of the number of

processors employed, we find a decrease of the solving time when we increase the number of processors, although this tendency is not true for more than 3 processors, as can be observed in Fig. 2(b) relative to the GMRES solver.

2.4 Aztec

Aztec[6] includes a number of Krylov iterative methods such as CG, GMRES, TFQMR and BCGSTAB. These Krylov methods are used in conjunction with various preconditioners such as polynomial or domain decomposition methods.

In this library the same solvers of PETSc are studied, using as a preconditioner the domain decomposition method Additive Schwarz, the inner solve being an incomplete LU factorization depending on the parameter ilut_fill (which indicates that the final factorization can contain as most ilut_fill times the number of no zero elements from the original matrix) and a drop tolerance (fixed to zero).

The lowest execution times are found for BCGSTAB solver, the differences between BCGSTAB solver and the other solvers studied are considerable. The minimun execution times are achieved using 3 processors, with ilut_fill values between 2 and 4. An example of this can be observed in Fig. 3(a) in which we represent for the BCGSTAB solver the influence of ilut_fill in solving time.

Acknowledgments. This work was partly supported by the Spanish Government (CICYT) under the project TIC 2001–3694–C02–01

References

1. Demmel J.W., Gilbert J., Li X.S.: SuperLU Users Guide (1999)
2. Saad Y., Lo GC., Kuznetsov S.: PSPARSLIB Users Manual: A portable library of parallel sparse iterative solvers. Univ. of Minnesota, Dept. Computer Science (1997)
3. Saad Y.: Iterative Methods for Sparse Linear Systems. PWS Publishing Co.(1996)
4. Saad Y., Gen–Ching L.: Iterative Solution of General Sparse Linear Systems on Clusters of Workstations. Univ. of Minnesota, Dept. of Computer Science (1996)
5. Balay S., Gropp W.D., McInnes L.C., Smith B.F.: PETSc 2.0 Users Manual, ANL-95/11 - Revision 2.0.24. Argonne National Laboratory (1999)
6. Hutchinson S. A., Shadid J., Tuminaro R.S.: Aztec User's Guide, SAND95-1559. Sandia National Laboratories (1995)

Parallel Solution of Cascaded ODE Systems Applied to ^{13}C-Labeling Experiments

Katharina Nöh and Wolfgang Wiechert

University of Siegen
Paul-Bonatz-Str. 9-11, D-57068 Siegen, Germany
{noeh,wiechert}@simtec.mb.uni-siegen.de
http://www.simtec.mb.uni-siegen.de

Abstract. In the rapidly growing field of computational methods within the discipline of Metabolic Engineering, the simulation of *instationary* ^{13}C labeling experiments is a new research focus. The underlying mathematical model is a high-dimensional cascaded system of differential equations and must be exploited to obtain efficient simulation algorithms. Additionally the sensitivity matrices of the system have to be computed as efficient as possible. For this purpose several ways for parallel implementation are introduced, compared and discussed.

1 Cascaded Systems

In this contribution high-dimensional cascaded ODE systems are considered which have the general form

$$\dot{{}^i\mathbf{y}} = \mathbf{f}\left({}^0\mathbf{y}, {}^1\mathbf{y}, \ldots, {}^{i-1}\mathbf{y}, \mathbf{p}\right), \quad i = 1(1)m$$

with initial values ${}^i\mathbf{y}(t_0) = {}^i\mathbf{y}_0$ and parameters **p.** Herein the upper left index of the vectors ${}^i\mathbf{y}$ specifies the stage of the cascade. To demonstrate the general methods developed in this paper a class of biotechnological examples is taken that arise in the framework of Metabolic Flux Analysis (MFA) [1-3].
In case of metabolic stationarity (constant metabolic fluxes **v** and pool sizes **X**) the detailed dynamics of an isotopically instationary ^{13}C labeling experiment is described by a so called cumomer vector **x** [4] and has the cascaded form

$$\mathrm{diag}({}^i\mathbf{X}) \cdot \dot{{}^i\mathbf{x}} = {}^i\mathbf{A}(\mathbf{v}) \cdot {}^i\mathbf{x} + {}^i\mathbf{b}(\mathbf{v}, \mathbf{x}^{inp}, {}^0\mathbf{x}, {}^1\mathbf{x}, \ldots, {}^{i-1}\mathbf{x}), \quad i = 1(1)m \qquad (1)$$

with ${}^0\mathbf{x} = \mathbf{1}$ and for some given initial values ${}^i\mathbf{x}(0) = {}^i\mathbf{x}_0$ and known input labeling $\mathbf{x}^{inp}$ [3]. Here $\dot{{}^i\mathbf{x}}$ depends linearly on ${}^i\mathbf{x}$ but nonlinearly on ${}^0\mathbf{x}, {}^1\mathbf{x}, \ldots, {}^{i-1}\mathbf{x}$. The equation ${}^0\mathbf{x} = \mathbf{1}$ is a conservation law which is exactly fulfilled. In [3] it is shown that for a reasonable metabolic network under weak conditions ($\mathbf{v}, \mathbf{X} > 0$) the non-singularity of ${}^1\mathbf{A}$ is equivalent to the global stability of the complete cascaded system (1). In the context of MFA the ODE (1) can have a very high dimension and moreover it constitutes an inverse problem for the fluxes **v.** Accordingly, the cascade (1) has to be solved repeatedly in the course of a parameter fitting procedure for flux determination. An efficient and aligned ODE solver thus is desirable because a standard "black-box" solver will cause tremendous computational costs.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 594–597, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Sensitivity Equations: Because the biological system and its measurements are usually rather noisy a detailed statistical analysis for parameter fits is always required. For that purpose the time dependent sensitivity matrices have to be computed by another cascaded system

$$\frac{d}{dt}\frac{\partial^i \mathbf{y}}{\partial \mathbf{p}} = \sum_{k=1}^{i-1} \frac{\partial \mathbf{f}}{\partial^k \mathbf{y}} \cdot \frac{\partial^k \mathbf{y}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}}{\partial \mathbf{p}}, \quad \frac{\partial^i \mathbf{y}}{\partial \mathbf{p}}(t_0) = 0, \quad i = 1(1)m. \tag{2}$$

These systems take the main part of computational time and the way how the sensitivity matrices are computed has a large impact on the overall performance of the method. Each column of the matrices $\partial^i \mathbf{y}/\partial \mathbf{p}$ corresponds to one parameter. Consequently, (2) has a high dimension of dim $\mathbf{y} \cdot$ dim $\mathbf{p}$. It is well known that the sensitivity ODEs (2) in principle have the same form as the systems ODE (1) and hence have the same numerical stability characteristics.

Example Systems: As a typical example in MFA the amino acid producer *C. glutamicum* may serve. Simulating an experiment with this bacterium with (1) results in 5608 equations with 11 stages. The *C. glutamicum* network has a total of 244 parameters and so (1)+(2) has the overall dimension 1,368,352. In this contribution one important part of the network is taken as an example: the so called PPP network of cumomer dimension 522 with 214 parameters.

2 Implementations of the Cascaded Systems

Several ways for implementation are proposed in this section. In the first instance the cascade (1) is considered.

Implementation ('s'): For a first (serial) simulation the solution of (1) is implemented on a single processor system (AMD Athlon 1800+). The explicit high order Runge-Kutta method DOPRI8 [5] with embedded step size control and an accurate dense output is chosen. If the system is regarded to be stiff the ODE solver is changed to a linear singly diagonally implicit method of order 4. One simulation run of (1) is carried out until the stationary state of the system is reached. DOPRI8 needs 10.43 s without and 16.59 s with an inner stage interpolation. An absolute accuracy of 10^{-4} resp. 10^{-7} is reached which both satisfies the accuracy requirements.
Now the sensitivity system (2) is added. There are two main methods to perform a sensitivity analysis numerically. On the one hand the sensitivity matrices of an ODE can be deployed analytically by integrating the variational equation (2), where the complex derivative matrices are given explicitly. On the other hand the derivatives of the solution in stage i can be computed by a standard differential quotient (DQ). This has some well known obstacles: The discretization error must be very small to meet the desired sensitivity accuracy within a given error tolerance. Furthermore there is no accuracy control of the sensitivities. However, this approach is very simple to implement.

The so arising computational effort is very high and it should be shortened by using multiple processors. All programs are tested on a cluster / SPMD architecture with 10 dual processor PCs (1.5 GB main memory each, 1000 MBps Ethernet interface). This contribution will concentrate on *parallelism across the problem* [6] because this approach seems to be the most promising one.

Functional Parallelism ('p'): The most apparent type of functional parallelism is pipelining: give each processor its own unique task and allow the data to "flow" through the pipeline in only one direction. As one processor finishes its calculation, it passes the solution along to the next processor and receives another solution from the previous processor.
Clearly, this is a "natural" parallelism for both the simulation problem (1) and the sensitivity equations (2). Each processor calculates one stage and the time shifts between the stages are variable. A rather good speedup is expected, because the last stage should theoretically have completed only a short time after the first stage.

Trivial Parallelism ('t'): This method does not involve parallelizing the code at all, rather the sequential code is run on a number of processors in parallel, with each processor having its own variant of the problem. The processors are working independently and there is no communication between the processes – except for the very end.
This method fits perfectly to the requirements of a sensitivity analysis using differential quotients as numerical approximation. Each processor gets a slightly varied parameter set and makes a simulation run. At the end all results are collected and the sensitivity matrices are given by a standard differential equation formula (DQ).

Decomposition of Sensitivity Matrices ('d'): Instead of dividing tasks between processors, each processor works on its own section of a dataset of parameters. This parameter set is initially divided between the processors, the calculations are carried out using the data, and the results are collected at the end of the computational process. Thereby the data can be easily distributed in such a way that the processor load is well balanced.
This seems to be the favorite method for the variational differential equation system (2) provided the solution vector $\mathbf{x}$ from (1) is given. It can be implemented by solving the cascade (1) additionally on *each* processor. Again no communication overhead is needed until the very end.

3 First Results

Each of these methods exploits a special feature of the systems to solve and can be combined with others. The following table shows the computational time for reasonable combinations of the parallel approaches. For comparability the ODE solver here is restricted to a constant step size sequence (1000 steps).

eqns.	(solution, sensitivity) method					
	(s,s)	(s,t)	(s,d)	(p,p)	(p,t)	(p,d)
(1, 2)	1021.51s	–	205.82s	534.88s	–	[1]
(1, DQ)	1591.87s	279.86s	–	–	381.98s	–

[1]: The implementation is currently under work.

The trivial parallelism method 't' gives a good speedup because it is not limited by a permanent communication overhead. This is also the case for the decomposition method 'd' even though the cascade (1) has to be computed multiple times. Generally, the expected time ranges are achieved for both methods.
Unlikely, this does not apply for the functional parallelism 'p'. Under precise consideration this method has three main limits: First of all there are idle processors at the beginning and the end of the computation. A more serious limit is the problem of load balancing. The stages have not the same dimensions and these differences are amplified by the sensitivities. So the processors are in general not load balanced and this will cause a bottleneck in the pipeline. To overcome this problem an additional level of parallel tasks has to be introduced in future versions of the code. The third limitation is given by the network communication since the whole solution must flow through all processors.

4 Conclusions and Outlook

After the special cascaded structure of the mathematical model of a CLE is introduced, possible sequential and parallel approaches are figured out. Already a moderately sized metabolic network shows that a sequential implementation is too slow for a repeated call by a parameter fitting procedure. Generally speaking only these parallel approaches are competitive which run without too much network communication because all benefits gained by multiple processors are discarded through it. It would also be interesting to test the scalability of the methods on clusters with more knots and to confirm the results for more extended networks like *C. glutamicum.*

References

1. Van Winden, W.: ^{13}C-Labelling Technique for Metabolic Network and Flux Analysis. PhD. Thesis, Delft University of Technology, 2002.
2. El Massoudi M., Drysch, A., Spelthahn, J., de Graaf, A.A., Takors, R.: Production Process Monitoring by Serial Mapping of Microbial Carbon Flux Distributions Using a Novel Sensor Reactor Approach. Metab. Eng. **5** (2003), pp. 86–95.
3. Wiechert, W., Wurzel, M.: Metabolic isotopomer labeling systems. Part I: Global dynamic behavior. Math. Biosciences **169** (2001), pp. 173–205.
4. Wiechert, W.: ^{13}C Metabolic Flux Analysis. Metab. Eng. **3** (2001), pp. 195–206.
5. Hairer, E., Nørsett, S.P., Wanner G.: Solving ordinary differential equations I + II. 2^{nd} ed., Springer, 2000 resp. 1996.
6. Burrage, K.: Parallel and Sequential Methods for Ordinary Differential Equations. Oxford Science Publications, Clarendon Press, 1995.

A k-way Graph Partitioning Algorithm Based on Clustering by Eigenvector

Tae-Young Choe[1] and Chan-Ik Park[2]

[1] School of Computer Engineering, Kumoh National Institute of Technology, 188, ShinPyung Dong, Gumi, KOREA 730-701. choety@kumoh.ac.kr
[2] Department of Computer Science and Engineering, Pohang University of Science and Technology, San 31, HyoJa Dong, Pohang, KOREA 790-784. cipark@postech.edu

Abstract. The recursive spectral bisection for the k-way graph partition has been underestimated because it tries to balance the bipartition strictly. However, by loosening the balancing constraint, the spectral bisection can identify clusters efficiently. We propose a k-way graph partitioning algorithm based on clustering using recursive spectral bisection. After a graph is divided into a partition, the partition is adjusted in order to meet the balancing constraint. Experimental results show that the clustering based k-way partitioning generates partitions with 83.8 ~ 108.4% cutsets compared to the strict recursive spectral bisections or multi-level partitions.

1 Introduction

Given a graph $G = (V,E)$, where V is the set of $|V| = n$ vertices and E is the set of $|E| = e$ edges, the k-way graph partition is composed of disjoint k subsets, where the union of the subsets is V. We assume all vertices and edges are same size one. An edge whose endpoints are located in different subsets is called the *cut* edge. The *cutset* of a partition is the set of cut edges. The graph partitioning problem is to divides V into a balanced partition with the minimum cutset size. The graph partitioning problem is known as NP-hard even in 2-way partitioning [1]. Therefore, the partitioning algorithms heuristically find approximated solutions within acceptable computation time.

The multi-level partitioning algorithm, one of the clustering-based algorithms, partitions a graph in three phases: coarsening phase, partitioning phase, and uncoarsening phase [2]. They rely on local information like adjacent vertices and edges to determine candidates to be merged. The spectral method partitions graph using eigenvectors of the Laplacian matrix of the graph. It utilizes global information and have been widely used [3]. Because many graphs are not balanced, balanced partitions by the spectral method can be local optima.

If we do not stick on the balancing constraint, the spectral method is especially efficient for clustering. From the position of vertices expressed by the Fiedler vector, some relations between vertices are revealed: dense areas and parse areas. Based on these facts, we divide a graph into clusters by setting a

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 598–601, 2004.
© Springer-Verlag Berlin Heidelberg 2004

sparse area as a border between clusters. The remainder of the paper is organized as follows; Section 2 presents the clustering methods and a partitioning algorithm. Section 3 provides experimental results. Finally, Section 4 concludes the paper.

2 The Proposed Algorithm

Our partitioning algorithm is just composed of three sequential steps: given graph is clustered considering vertices connectivity in clustering step, the clusters are fitted to balanced blocks in balancing step, and finally, a simple partitioning algorithm refines the balanced blocks in refinement step.

2.1 Clustering Step

The base architecture of the clustering step is the recursive spectral method [4]. The main difference between the proposed step and other recursive spectral bisection algorithms is that our clustering step attaches more importance to clustering with smaller cutset rather than balanced partition.

Figure 1 shows function Clustering() for the clustering step. Element f_i in the Fiedler vector F_C is considered as an one-dimensional coordinate of vertex v_i in G_C. By projecting the vertices into one-dimension, we can see relative positions and connectivities between vertices. That is, as two vertices are more strongly connected each other, they are laid more closely in the one-dimension, and vise versa.

Function Clustering(G, k)
  //INPUT: graph $G = (V, E)$ to be clustered and k the number of clusters
  //OUTPUT: a set of clusters $\mathcal{C}$
  //CONSTANT l: the number of subregions
  $\mathcal{C} \leftarrow \{V\}$;
  while $|\mathcal{C}| < k$ **do**
    $C \leftarrow$ the largest cluster in $\mathcal{C}$;
    $G_C \leftarrow$ subgraph composed of C;
    compute Fiedler vector F_C of G_C;
    divide section $[\min_{f_i \in F_C} f_i, \max_{f_i \in F_C} f_i]$ to l subregions;
    partition G_C into $C_1, C_2, \ldots, C_k$ according to partition policy;
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{C_1, C_2, \ldots, C_k\} - \{C\}$;
  end while
  return $(\mathcal{C})$;

Fig. 1. Clustering step

In order to find clusters in a graph, we simplify the method proposed by Hagen and Kahng [5]. Our algorithm divides the entire region of the vertices coordinates into same sized subregions and counts the number of vertices in

Table 1. The cutset size and execution time of result partition of each algorithm

measure		cutset				run time	
No. subset	graph	Chaco	Sanchis	Metis	Proposed	Metis	Proposed
2	hammond	105	233	109	104	0.02	1.25
	barth5	164	204	149	**147**	0.09	5.28
	brack2	**747**	1886	779	1006	0.62	23.85
8	hammond	590	414	**376**	383	0.03	4.90
	barth5	**733**	970	744	823	0.09	44.00
	brack2	8986	10934	8802	**8341**	0.68	111.03
32	hammond	1174	1250	1066	**1047**	0.05	8.46
	barth5	1853	2160	1823	**1716**	0.13	66.34
	brack2	21674	24705	**18898**	20690	0.77	243.75

each subregion. The routine sets the initial number of subregions as the number of vertices $|C|$. Adjacent two subregions are merged until every subregions has one or more vertices. After the merge operations, we identify a cluster boundary which is a subregion with the least vertices If there are two or more subregions with the least vertices, a border that generates most balanced clusters is selected.

2.2 Balancing Step

In general, clusters generated during clustering step are unbalanced, even some cluster could be splited. Thus balancing step is indispensable for partition to meet the balancing constraint. We modify and combine Sanchis algorithm [6] and P-P algorithm [7]. Gain of a movement for a vertex to a subset is the number of removed cutsets and improvement in degree of balance [7]. The gain is managed by bucket data structure [6].

2.3 Refinement Step

The refinement step runs on the partition that satisfies the balancing constraint. The remainder is to reduce the cutset size further. Thus, Sanchis algorithm is a good refinement algorithm to reduce the cutset size. Since the partition becomes globally near optimal by the clustering step, it is not needed to consider escape from local optima. Thus movement of cells with negative gain is not necessary.

3 Experimental Results

The algorithms are implemented in C language. Since the programming code uses the standard C libraries, it can run in almost all systems. The Chaco, Sanchis algorithm, and the Metis algorithms are executed to compare the performance with our algorithms. Among the many options of the Chaco partitioning algorithm, recursive spectral octal-partition and K-L refinement are used. The Metis

runs with default options. The balancing constraint is 3% of $|V|/k$. Three test graphs are used: hammond with 4720 vertices and 13722 edges, barth5 with 15606 vertices and 45878 edges, and brack2 with 62631 vertices and 366559 edges.

The results of the algorithm executions are shown in Table 1. The data in Sanchis column show partition without clustering step. That is, it starts with a partition that all vertices are in one subset, runs the min-bal balancing steps, and finishes with Sanchis refinement. Each bold fonted number means that the number is the smallest cutset given mesh and the number of subsets. Execution times of Metis and the proposed algorithm are compared. A large proportion the execution times in the proposed algorithm is the refinement step. It shows that the refinement step need some performance tuning as future works.

4 Conclusions

We proposed an effective partitioning algorithm that generates a partition with a relatively small cutset. The algorithm clusters the graph using the recursive unbalanced bipartition.

Acknowledgments. The authors would like to thank the Ministry of Education of Korea for its support toward the Electrical and Computer Engineering Division at POSTECH through its BK21 program. This research was also supported in part by HYSDR IT Research Center and in part by grant No. R01-2003-000-10739-0 from the Basic Research Program of the Korea Science and Engineering Foundation.

References

1. Bui, T.N., Jones, C.: Finding good approximate vertex and edge partitions is NP-hard. Information Processing Letters **42** (1992) 153–159
2. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing **20** (1999) 359–392
3. Simon, H.D., Sohn, A., Biswas, R.: Harp: A dynamic spectral partitioner. Journal of Parallel and Distributed Computing **50** (1998) 83–103
4. Hsieh, S.H., Paulino, G.H., Abel, J.F.: Recursive spectral algorithms for automatic domain partitioning in parallel finite element analysis. Comput. Methods Appl. Mech. Engrg. (1995) 137–162
5. Hagen, Kahng: New spectral methods for ratio cut partitioning and clustering. IEEETCAD: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **11** (1992)
6. Sanchis, L.: Multiple-way network partitioning. IEEE Trans. Computers **38** (1989)
7. Park, C.I., Park, Y.B.: An efficient algorithm for vlsi network partitioning problem using a cost function with balancing factor. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **12** (1993)

Network of Networks*,**

José de Jesús Cruz Guzmán and Zbigniew Oziewicz* * *

Universidad Nacional Autónoma de México, Facultad de Estudios Superiores
Campus Cuautitlán, Apartado Postal # 25, C.P. 54714 Cuautitlán Izcalli
Estado de México, {Cruz,Oziewicz}@servidor.unam.mx

Abstract. We consider the categorical concepts of a 'network of networks': (a) each node is a host network (1-network or 1-graph) and super-links are analogous to a graph-functor, *i.e.* this is (1, 1)-network; (b) 2-network where there are 2-links among 1-links. The general notion of network-morphism is proposed.

1 Network

The dominant technological structure is a network (a graph), e.g. World Wide Web graph, each web page is a node and each (hyper)link is a directed edge; internet, social networks, networks in the molecular biology, a biological cell as a network of genes, neural networks, metabolic network, scientific citation network, energetic networks, phone calls, linguistics networks, networks in natural languages, ecological eco-networks, computer circuits, www.nd.edu/networks or http://www.internetmathematics.org.

A graph underline a category [4,6,7, Burroni 1981, Lawvere 1989]. Graphs provide useful concept in a computational science (integrating science with a computation), braided logic [5, Chávez Rodríguez et al. 2001], etc. Lawvere consider the category of directed graphs and their morphisms [Lawvere 1989]. The categorical aspects of graphs is not familiar outside of category theory. The value of categories for computational science is that the category theory, invented by Eilenberg and MacLane in 1945, has developed *language* for the study of structures, language for the formulation of problems, language for development of methods of calculation and deduction, language for discovering and exploiting analogies between various inter-disciplinary fields of science. Attraction of category theory is that the same algebraic tool=language is applicable in a variety of multi-disciplinary science. Study of the structure (process) involves studying the pre-categories ($\equiv$ graphs) and pre-functors $\equiv$ graph-morphisms.

* Supported by el Consejo Nacional de Ciencia y Tecnologia (CONACyT) de México, grant # 41214-F. Supported by UNAM, DGAPA, Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica, grant # IN 105402.

** International Conference on Computational Science, ICCS'2004, Kraków, Poland, June 2004. Accepted for Lecture Notes in Computer Science, Springer-Verlag, Editors: Marian Bubak, G. Dick van Albada, Peter M.A. Sloot, Jack J. Dongarra.

*** A member of Sistema Nacional de Investigadores in México, Expediente # 15337.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 602–605, 2004.
© Springer-Verlag Berlin Heidelberg 2004

A graph consists of collections of variously called objects/nodes/vertices/states/0-cells V/G_0, and arrows/edges/links/transitions/1-cells E/G_1, with two operations: domain/source $s : G_1 \to G_0$ and codomain/target $t : G_1 \to G_0$ [4,9, Burroni 1981, Mac Lane 1998]. Elements of G_0 are said also to be 0-morphisms/0-transitions, elements of G_1 are said to be 1-morphisms/1-transitions. For a cell $f \in G$, the value *sf* is said to be a domain or a source of f, and *tf* is a co-domain or target of f.

A (time-dependent) out-degree of a cell $x \in G$ is a cardinality of a fiber $s^{-1}x$. An in-degree of x is a cardinality of $t^{-1}x$. The power-law degree say that a distribution probability of the connecting links to the new node y (to a new 0-cell y) for *ty* is $\simeq$ (degree of y)$^{-(1\div 4)}$, known as a 'scale-free'. A power-law in-degree distribution means that the percentage of the number of cells with a given in-degree is proportional to the power $(\text{in-degree})^{-\text{constant}}$. A power-law degree was discovered by Lotka [8, 1926] in a scientific productivity.

The models of the time evolution of the networks, models of the graph-process and models of the interactions among nodes, attract most attention. The best known is the Barabási-Albert model, based on two axioms one new vertex $x \in V$ at a time with a fixed node-independent in-degree $t^{-1}x \in \mathbb{N}$, [1,2, Barabási & Albert 1999; Barabási 2002].

1.1 Aim of This Note

It was observed by Bharat et al. [3, 2001], that web-graph is a 2-level structure. We consider the categorical concepts of a 'network of networks' or an hierarchical network. For a natural number $m \in \mathbb{N}$ we are introducing a concept of a m-network, and moreover for $m, n \in \mathbb{N}$ a fine concept of (m, n)-network. In particular, our aim is to elucidate the differences and similarities among the following hierarchical structures.

(a) Each node is a usual (connected) host network (1-network or 1-graph) and super-links are analogous to a graph-functor. Such structure we call (1,1)-network or (1, 1)-graph, or 'graph of host-graphs', or a network of clusters
(b) 2-network (2-graph) where there are links among specified links of the same type.

The study of categorical aspects of networks leads to new insights. A graph is the same as 1-graph. We developed conceptual notion of a graph of graphs, (1,1)-graph, multi-graph m-graph, multi-graph of multi-graphs (m, n)-graph, and iterations $(n_1, \ldots, n_k)$-graph for $\forall\, n_i \in \mathbb{N}$. Graph of graphs can be used for refinement of the hierarchical structures and /or 'network of networks'. In particular the conceptual hierarchical structure 'graph of graphs' can be useful for description of the networks where the links are attribute-dependent and therefore are natural 'higher' links between attributes leading to multi-level systems [10,11, Obtułowicz 2001], for example the attribute-dependent links among employers and workers, professors and students.

2 Graph-Morphism

A multi-network G is a sequence of collections of i-cells/i-morphisms $\{G_i, i \in \mathbb{N}\}$, a sequence of surjective sources $\{s_i \equiv s|G_i\}$ and targets $\{t_i \equiv t|G_i\}$. A 0-cell in G_0 is variously also called an object, 0-morphism, node; 1-cell - a morphism, 1-morphism, an arrow, a functor. A 2-cell is called a 2-morphism, morphism of morphism, natural transformation, etc.

Let $E \equiv (E_0 \leftleftarrows E_1 \leftleftarrows \cdots \leftleftarrows E_k)$ and $F \equiv (F_0 \leftleftarrows F_1 \leftleftarrows \cdots \leftleftarrows F_k)$ be two k-graphs. A k-graph morphism $f : E \to F$, $f \in \mathrm{graph}(E, F)$, is a collection of functions $f_i : E_i \to F_i$, intertwining the source s and the target t maps (in this case a morphism is said to be totally covariant),

$$s \circ f = f \circ s \quad \text{and} \quad t \circ f = f \circ t,$$

$$\begin{array}{ccccc} E_0 & \leftleftarrows & E_1 & \leftleftarrows & E_2 \ldots \\ \downarrow f_0 & & \downarrow f_1 & & \downarrow f_2 \\ F_0 & \leftleftarrows & F_1 & \leftleftarrows & F_2 \ldots \end{array}$$

Set $E,\ F \in G_0$ to be 0-cells, and $\mathrm{graph}(E, F) \subset G_1$, then $G \equiv (G_0 \leftleftarrows G_1)$ is a graph of k-graphs, that we denote by $(1, k)$-graph. This means that every 0-cell of G is a k-graph and every 1-cell of $(1, k)$-graph G is a k-graph morphisms, $f \in G_0(E, F) \equiv \mathrm{graph}(E, F) \subset G_1$.

A 1-graph $G = (G_0 \leftleftarrows G_1)$ whose every 0-cell is a graph, and every 1-cell is a graph morphism, is said to be a graph of graphs. In particular, (0, 1)-graph is a collection of 1-graphs, and a 1-graph whose every 0-cell is a 1-graph and whose every 1-cell is 1-graph morphism is said to be a (1,1)-graph. Therefore we interpret (1,0)-graph as a more particular concept that a general 1-graph.

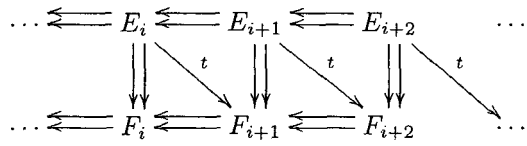
2.1 2-Graph of Graphs

Let G be a 2-graph such that every 0-cell is a k-graph and every 1-cell is a k-graph morphism, *i.e.* G is a $(2, k)$-graph. We need to define 2-cell as a k-graph natural transformation. The following definition generalize the notion of the natural transformation of (pre)functors from the case 2-graph of $k =$ 1-graphs invented by Eilenberg and MacLane [1945], to the case of arbitrary natural number $k \in \mathbb{N}$.

Let $f, g \in G_0(E, F) \subset G_1$. This means

$$G_0(E, F)(f, g) \quad \subset \quad G_1(f, g) \quad \subset \quad G_2.$$

A k-graph natural transformation $t \in \mathrm{nat}(f, g) \equiv G_1(f, g) \subset G_2$, is a collection of $(i + 1)$-cells-valued map intertwining f with g,



$$E_i \ni x \overset{t}{\longmapsto} tx \in F_i(fx, gx) \subset F_{i+1},$$

$$\forall\, a \in E_i(x, y) \subset E_{i+1}, \quad (ga) \circ (tx) = (ty) \circ (fa) \quad \in F_{i+1}. \tag{1}$$

The inter-twining condition (1) is referred as the naturality condition.

Let s_F denote F-source map. Then $s_F \circ t = f = s_G t$, therefore s_G is given be push-forward, $s_G = (s_F)_*$.

The k-graph natural transformation $t \in \mathrm{nat}(f, g)$ for the case of a graph of 1-graphs' is analogous to natural transformation of functors. A natural transformation $t \in \mathrm{nat}(f, g)$ can be illustrated also by means of a commutative diagram with pull-back and push-forward,

$$\begin{array}{ccc} E(x,y) & \xrightarrow{f} & F(fx, fy) \\ {\scriptstyle g}\downarrow & & {\scriptstyle (ty)_*}\downarrow \\ F(gx, gy) & \xrightarrow{(tx)^*} & F(fx, gy) \end{array}$$

Therefore $G_0(E, F) \subset G_1$ can be seen as an 1-graph (= pre-functor graph): vertices are the morphisms of graphs, $f, g \in \mathrm{graph}(E, F)$, and edges are natural transformations $\mathrm{nat}(f, g) \subset G_2$.

A (0, 2)-graph is a collection G_0 whose every element is a 2-graph. A (2, 0)-graph is not the same as a 2-graph, *i.e.* a 2-graph $G_0 \leftleftarrows G_1 \leftleftarrows G_2$ whose every 0-cell in G_0 is a $0\text{-graph} \simeq \mathrm{set}$, 1-cells are mappings and 2-cells are operators.

References

1. Barabási Albert-László, Albert Rebeca: Emergence of scaling in random networks. Science **286** (1999) 509–512
2. Barabási Albert-László: Linked: the new science of networks. Perseus Publishing, 2002
3. Bharat K., Cheng B., Henzinger M., Rühl M.: Who links to whom: mining linkage between web cities. Proceedings of the IEEE International Conference on Data Minig, Los Alamitos, California, IEEE Press 2001 pp. 51–58
4. Burroni Albert: Algèbres graphiques. Cahier (1981)
5. Chávez Rodríguez María Ernestina, López Gomez Angel: Braided logic: the simplest models. International Journal of Theoretical Physics **40** (1) (2001) 95–103
6. Lawvere F. William: Display of graphics and their applications, as exemplified by 2-categories and the Hegelian 'taco'. State University New York at Buffalo, 1989
7. Lawvere F. William: Qualitative distinctions between some toposes of generalized graphs. Contemporary Mathematics **92** (1989) 261–299
8. Lotka A. J.: The frequency distribution of scientific productivity. Journal of the Washington Academy of Sciences **16** (1926) 317
9. Mac Lane Saunders: Categories for the Working Mathematician. Springer-Verlag (1971, 1998)
10. Obtułowicz Adam: On n-dimensional graphs and n-graphs. Miscellanea Algebraicae (Kielce), Rok 5, No 1/2001, pp. 87–96, ISBN 83-87798-22-3
11. Obtułowicz Adam: A note on the connections of automata and weak double categories. Miscellanea Algebraicae (Kielce), Rok 5, No 1/2001, pp. 75–86, ISBN 83-87798-22-3

MSL: An Efficient Adaptive In-Place Radix Sort Algorithm

Fouad El-Aker[1] and Amer Al-Badarneh[2]

[1] Computer Science Department, New York Institute of Technology,
P.O. Box 940650, Amman 11194, Jordan
Fouad_elaker@yahoo.ca
[2] Computer Information Systems Department, Jordan University of Science and Technology
P.O. Box 3030, Irbid 22111, Jordan
Amerb@just.edu.jo

Abstract. This paper presents an in-place pseudo linear average case radix sorting algorithm. The proposed algorithm, MSL (Map Shuffle Loop) is a modification of the ARL algorithm. The MSL permutation loop is faster than the ARL counterpart since it searches for the root of the next permutation cycle group by group. The permutation cycle loop maps a key to its target group and shuffles the input array. The performance of MSL is compared with Java quicksort, as well as MSD and LSD radix sorting algorithms.

1 Introduction

Radix sorting [1] algorithms fall into two major categories, depending on whether they process bits left to right, or right to left. After the k^{th} step, MSD (Most Significant Digit) sorts the input keys according to their left most k digits. Similarly, LSD (Least Significant Digit) processes digits in the opposite direction, but it is non-recursive, unlike MSD. MSL processes bits left to right recursively. MSL is described in Section 2. In Section 3, we present the algorithm analysis and experimental results. Finally, Section 4 gives the conclusion and future work.

2 The MSL Algorithm

The main steps of MSL sorting algorithm are presented first. Each step executes a loop whose time complexity is indicated. N is the size of input data, while K is the number of groups.

```
Step 1: Compute groups' sizes.                          O(N)
Step 2: Compute groups' start and end addresses.        O(K)
Step 3: Permutation cycles (Map shuffle) loop.          O(N)
Step 4: Process groups recursively.                     O(K)
```

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 606–609, 2004.
© Springer-Verlag Berlin Heidelberg 2004

MSL is a modification of the ARL algorithm [2]. Both algorithms use permutation cycles, which is the primary reason for the in-place property of the algorithms. MSL and ARL differ in how they search for and select the root of the next permutation cycle, Step 3, and in how they compute the radix size.

Radix computing in MSL is similar to adaptive MSD (AMSD) [3]. AMSD radix setting is simple to understand, and is modified to work on bits instead of digits. The following modifications are applied to the implementations of both MSL and AMSD: (1) Recompute the radix size continually on recursive calls instead of only once, initially. (2) Use multiple of 4 radix size values and disallow the use of small radix size values, 1 to 3, to improve the running time. The radix sizes used in MSL are 4, 8, and 12, etc. The initial radix size value 15 is used in Section 3 in sorting large arrays of 31 bits integers, to avoid using radix size values 1 to 3.

In addition, the implementations of MSL and AMSD use one iteration of quicksort partitioning before calling insertion sort. This initial call inserts the smallest key of each of the two partitions in its correct place. This speeds up insertion sort, which is called frequently. Insertion sort in this case has a redundant if-condition that is removed from its inner loop.

The digit size is computed in MSL as follows: (1) For array sizes 1..25, no radix is used, and the quicksort partition method is called instead. (2) For array sizes 26..5000, 5001..80000, and larger than 80001, a digit size 4, 8, and 12 is used, respectively. In comparison, ARL computes the digit size in terms of the largest key value in the array, and applies a maximum threshold on digit sizes.

MSL, like ARL, requires 2 supporting arrays, each of size 2^d, where d is the current digit size. For MSL extra space requirements, see Section 3. ARL uses the supporting arrays to store groups' start addresses and sizes. MSL replaces the groups' sizes by the groups' end addresses. In Step 2 above, K extra assignments are executed to replace sizes by end addresses.

MSL inserts a key in its target group, at the key target group's end address. The key target group's end address is decremented immediately beforehand. MSL performs a total of N decrement operations for this purpose. In addition to these N decrement operations, ARL needs extra N addition operations to compute keys' target addresses. A key's target group's address is computed by summing its target group's start address and size.

ARL and MSL permutation cycles, Step 3, insert keys into their target groups. Each permutation cycle starts and ends at the same location, which is the root key location. The root key is the initial key used in the cycle, and is recomputed when a new permutation cycle is started.

In ARL, the exchange cycle root key is the first key, counting from the left end of the array that is not in its target group. ARL uses an independent loop of total N increment operations to find all roots for permutation cycles. In MSL, a total of K increment operations are needed instead.

MSL Step 3 is organized into 3 substeps. In substep 1, MSL finds the next root key, by searching through groups, which makes it faster in principle than ARL. The *originGroup* variable advances through groups until a group is found that is not fully processed (when the group's start and end addresses do not match). At this point, *originGroup* points to the left most group, which has a key that may not be in its target group (See example below). MSL permutation code stops when the last non-empty group is reached. The condition used is: if *(originGroup == maxGroup).*

In substep 2, root key information is updated. The root key *(root_key)* of the next exchange cycle is computed, in addition to its group *(root_key_group)* and its destination group *(dest_group)*. Target addresses are computed directly from target groups using the group's end address.

Substep 3, is the actual permutation cycle. As exchanges are performed, *root_key_group* remains unchanged. For each body execution of the permutation cycle, *root_key* is exchanged with the key that is in its destination address, where the destination address is computed from *dest_group*. Next, *dest_group* is updated to the destination group of the current *root_key*. The current exchange cycle stops when a key is found whose target group is equal to *root_key_group*. The condition used is: if *(dest_group == root_key_group)*.

The following is a trace of the 3 substeps above. The example assumes we are sorting a group of 8 bit integers on the left most 4 bits, where the radix size value is 4. There are a total of three groups, whose identities, start and end addresses are provided. The changes made by a substep are bolded. Only the modified group information after a substep execution is displayed.

Initial array data: ⟨3, 20, 1, 21, 41, 22, 40, 42, 2⟩;
Initial groups data: *minGroup* = 0; *maxGroup* = 2; (*Gid* = 0, *Start* = 0, *End* = 3);
(*Gid* = 1, *Start* = 3, *End* = 6); (*Gid* = 2, *Start* = 6, *End* = 9);
After substeps 1 & 2: *originGroup* = 0; *root_key_group* = 0; *root_key_address* = 2;
After exchanges in substep 3:
First exchange: ⟨3, 20, **1**, 21, 41, 22, 40, 42, 2⟩; (*Gid* = 0, *Start* = 0, *End* = **2**);
// Key value 1 is already in its target group
After substeps 1 & 2: *originGroup* = 0; *root_key_group* = 0; *root_key_address* = **1**;
After exchanges in substep 3:
First exchange: ⟨3, , 1, 21, 41, **20**, 40, 42, 2⟩; (*Gid* = 1, *Start* = 3, *End* = **5**);
Next exchange: ⟨3, , 1, 21, **22**, 20, 40, 42, 2⟩; (*Gid* = 1, *Start* = 3, *End* = **4**);
Next exchange: ⟨3, , 1, 21, 22, 20, 40, 42, **41**⟩; (*Gid* = 2, *Start* = 6, *End* = **8**);
Next exchange: ⟨3, **2**, 1, 21, 22, 20, 40, 42, 41⟩; (*Gid* = 0, *Start* = 0, *End* = **1**);
// Key value 2 is inserted in root key address, closing the exchange cycle
After substeps 1 & 2: *originGroup* = 0; *root_key_group* = 0; *root_key_address* = **0**;
After exchanges in substep 3:
First exchange: ⟨**3**, 2, 1, 21, 22, 20, 40, 42, 41⟩; (*Gid* = 0, *Start* = 0, *End* = **0**);
// Key value 3 is already in its target group
After substeps 1 & 2: *origin group* = **1**; *root_key_group* = **1**; *root_key_address* = **3**;
After exchanges in substep 3:
First exchange: ⟨3, 2, 1, **21**, 22, 20, 40, 42, 41⟩; (*Gid* = 1, *Start* = 3, *End* = **3**);
// Key value 21 is already in its target group
After substeps 1 & 2: *originGroup* = 2; Exit, since *originGroup* == *maxGroup*;

3 Experimental Results

In the following, assume that the average applied digit is *A,* and that we are sorting *B* bit integers. For the worst case, the algorithm is called recursively to the maximum possible number of levels, $L = B/A$, where partitioning is always severely unbalanced. The worst case is $O(N*L)$. The average case is $O(N \log_c N)$, when partitioning is always balanced, and the algorithm is called recursively to level $\log_c N$, where $C = 2^A$.

In Table 1, all the tests take the average of five runs. The machine used in the tests is Compaq, 500 MHZ Pentium III, with 128 MB RAM. The distributions used are random with varying densities. We divide the full integer range into x subranges and use the first subrange to randomly generate the data for the random distribution named R/x. The smallest integer value used in the distribution R/x is zero, and the largest integer value used is Java_Max_Integer/x.

MSL is compared with AMSD, LSD, and Java tuned quicksort. For all 4 algorithms, the run time improves as the range is decreased. MSL run time is more than twice as fast as Java quicksort, for all runs. MSL run time is better than AMSD. MSL and LSD running time are relatively superior compared to Java quicksort and AMSD. The time taken to allocate extra space is excluded. The extra space allocated for LSD is N. The extra space allocated for MSL and AMSD is 0.14*N, 0.07*N, 0.35*N, and 0.0175*N, for the array sizes 0.5, 1, 2, and 4 millions, respectively. For other random distributions, the algorithms behave similarly.

Table 1. A comparison shows MSL, AMSD, LSD, & Java tuned quicksort (QSort) running times in milliseconds. Sizes are multiple of 10^6

Sizes * 10^6		0.5	1	2	4		0.5	1	2	4
LSD	R/1	372	770	1550	3130	R/10^3	306	604	1204	2426
AMSD		474	868	1614	3186		362	682	1408	2988
QSort		614	1352	2868	6064		636	1330	2778	5734
MSL		340	726	1430	2910		272	570	1222	2332
LSD	R/10	330	660	1338	2702	R/10^6	198	406	808	1670
AMSD		428	680	1516	3186		322	582	1228	2558
QSort		670	1360	2866	6046		372	750	1528	3196
MSL		284	626	1408	2746		208	328	738	1500

4 Conclusion and Future Work

In-place radix sorting includes ARL and MSL. The run time of MSL is competitive with other radix sorting algorithms, such as LSD. Future work on in-place radix sorting algorithms includes engineered in-place radix sorting. In addition, testing in-place radix sorting algorithms on 64 and 128 bits integers is important.

References

1. Sedgewick, R.: Algorithms in Java, Parts 1-4, 3rd Edition, Addison-Wesley, (2003).
2. Maus, A.: "ARL: A Faster In-place, Cache Friendly Sorting Algorithm", Norsk Informatikkonferranse, NIK'2002, (2002) 85-95.
3. Anderson, A., Nilsson, S.: Implementing Radixsort, ACM Journal of Experimental Algorithms. Vol. 3(7). (1998) 156-165.

Parallel Chip Firing Game Associated with *n-cube* Edges Orientations

René Ndoundam[1], Claude Tadonki[2], and Maurice Tchuente[1]

[1] University of Yaoundé I, Faculty of Science,
Department of Computer Science, P.O. Box. 812 Yaoundé, Cameroon
[2] University of Geneva, Centre Universitaire Informatique,
Departement of Theoretical Computer Science 24, rue Général Dufour,
1211 Genève 4 - Switzerland
ndoundam@uycdc.uninet.cm, claude.tadonki@cui.unige.ch

Abstract. We study the cycles generated by the chip firing game associated with *n-cube* orientations. We consider a particular class of partitions of vertices of *n-cubes* called *left cyclic partitions* that induce *parallel periodic evolutions*. Using this combinatorical model, we show that cycles generated by *parallel evolutions* are of even lengths from 2 to 2^n on H_n $(n \geq 1)$, and of odd lengths different from 3 and ranging from 1 to $2^{n-1}-1$ on H_n $(n \geq 4)$. However, the question weather there exist *parallel evolutions* with period greater that 2^n remains opened.

1 Introduction

A state in the *parallel chip firing game* played on a directed graph $G = (X, A)$ is a mapping $x : V \rightarrow N$ which can be viewed as a distribution of chips onto the vertices of G. In a transition of the game, a state x is transformed into a new state by activating all nodes with more chips that its out-neighbors. The evolution is ultimately *periodic* because the total number of chips remains constant. More precisely, if $x^t, t \geq 0$, denotes the state of the system at time t, then there exists an integer q called *transient length* and another integer p called *period* or *cycle length* such that

$$x^{t+p} = x^t \text{ for } t \geq q, \text{ and } x^{t+p'} \neq x^t \text{ for } p' < p. \quad (1)$$

In this paper, we investigate the dynamics generated by the chip firing game associated with *n-cube* orientations and we provide a model to study possible periods generated in this particular case.

2 Recurrent Construction of Parallel Cycles

Definition 1. *A partition $S_0 \cup S_1 \cup ... \cup S_{k-1}$ of the vertices of an* n-cube *is called a* left cyclic *partition if the two following statements hold.*

- *For all i from 0 to $k-1$, every vertex of S_i has a neighbor in S_{i-1}, where index operations are performed modulo k.*
- *For all i from* 0 to $k-1$, *there is no edge between two vertices of S_i.*

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 610–613, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The model of *left cyclic partition* clearly gives a characteristic of *parallel evolutions* with unique firing within a cycle. In this paper, we investigate on possible configurations. We first present the construction of *left cyclic partitions* of even lengths.

Lemma 1. *An n-cube admits left cyclic partitions of all even lengths from 2 to 2^n.*

Proof. Let $H_n = (V, E)$ be an n-cube an let p be an even integer between 2 and 2^n. It is well known that, since p is even, there is a cycle $[x_0, x_1, ..., x_{p-1}, x_0]$ of length p in H_n. Now, for every vertex u, let $\Gamma(u)$ denote the set of all neighbors of u in H_n. This notation is naturally extended to a set of vertices. A *left cyclic partition* of order p is obtained as follows.

For $i = 0, ..., p-1$ **do**
  $S_i \leftarrow \{x_i\}$
endfor
$S = V - \{x_0, x_1, ...x_{p-1}\}$
while $(S \neq \emptyset)$ **do**
  For $i \leftarrow 0$ **to** $p-1$ **do**
    $S_{i+1} \leftarrow S_{i+1} \cup (\Gamma(S_i) \cap S)$
    $S \leftarrow S - (\Gamma(S_i) \cap S)$
  endfor
endwhile

It is obvious that $S_0, ..., S_{p-1}$ is a partition of V and that every vertex in S_i has at least one neighbor in S_{i-1}. So we just need to show that two vertices of the same subset S_i cannot be neighbors. Let a and b be two vertices of S_i.

- There is a path from a to x_0 of length ℓ_1 such that $\ell_1 = i \bmod p$,
- There is a path from b to x_0 of length ℓ_2 such that $\ell_2 = i \bmod p$,

Since p is even, it follows that $\ell_1 = \ell_2 \bmod 2$. Hence, if a and b were neighbors, there would exist a cyclic path of odd length $\ell_1 + \ell_2 + 1$ joining a and b in H_n, which is not possible since H_n is a bipartite graph. This shows that two vertices of the same subset cannot be neighbors.

We now turn to the construction of *left cyclic partitions* of odd lengths.

Lemma 2. *If S_0, S_1, S_2 is a* left cyclic partition *of $H_n, n \geq 2$, then every vertex of S_i has at least two neighbors in S_{i-1} for $i = 0, 1, 2$.*

Proof. Because of symmetry considerations, we can assume that $i = 2$. So let x be a vertex of S_2. From the definition of *left cyclic partitions,*

- x has a neighbor $x \oplus e_j$ in S_1, where $\oplus$ is the XOR operator and e_j is a vector of the canonical basis.
- similarly, $x \oplus e_j$ has a neighbor $x \oplus e_j \oplus e_k$ in S_0.

Now consider the vertex $x \oplus e_k$.

- It is a neighbor of x, hence it does not belong to S_2.
- It is a neighbor of $x \oplus e_j \oplus e_k$, hence it does not belong to S_0.

It then follows that $x \oplus e_k$ belongs to S_1, hence x admits two neighbors $x \oplus e_j$ and $x \oplus e_k$ which are both in S_1.

Lemma 3. *If H_n, $n \geq 3$ admits a* left cyclic partition *of order* 3, *then H_{n-1} admits a* left cyclic partition *of order* 3.

Proof. Obvious.

Proposition 1. *n-cubes do not admit left cyclic partitions of order 3.*

Proof. From lemma 2, if a hypercube H admits a *left-cyclic partition* of order 3, then $|H| \geq 6$, which is not the case for H_2. By application of lemma 3, we deduce that no *n-cube*, $n \geq 3$ admits a *left cyclic partition* of order 3.

Proposition 2. *If $S_0, ..., S_{p-1}$ is a left cyclic partition of odd order p of H_n, then $p \leq 2^{n-1} - 1$.*

Proof. We just have to show that in such a case, $|S_i| \geq 2$ for $i = 0, ..., p-1$. Indeed, starting from a vertex $a_{p-1} \in S_{p-1}$, we construct a chain $[a_{p-1}, a_{p-2}, ..., a_0, b_{p-1}, b_{p-2}, ..., b_0]$ such that $a_i, b_i \in S_i$ for $i = 0, ..., p-1$. It is clear that $a_i \neq b_i, i = 0, ..., p-1$, otherwise we would have displayed a closed path of odd length in H_n which is not possible.

Lemma 4. *If H_n admits a* left cyclic partition *of order p, then H_{n+1} admits left cyclic partition of order p.*

Proof. If $S_0, ..., S_{p-1}$ is a left cyclic partition of order p in H_n, then $1S_i \cup 0S_{i-1}, i = 0, ..., p-1$ is a also *left cyclic partition* of order p in H_{n+1}.

Lemma 5. *If H_n admits a* left cyclic partition *of odd order p, $p \geq 5$ then H_{n+1} admits a* left cyclic partition *of order $2p-1$. Moreover, if $p \geq 7$, then H_{n+1} admits a left cyclic partition of order $2p-3$.*

Proof. Let $S_0, S_1, ..., S_{p-1}$ be a *left cyclic partition* of odd order p.

- Case $p \geq 5$

The following sequence is a *left cyclic partition* of order $2p-1$ in H_{n+1}.
$0S_0$, $1S_0 \cup 0S_1$, $1S_1$, $1S_2$, $0S_2$, $0S_3$, $1S_3$ ..., $1S_{2i}$, $0S_{2i}$, $0S_{2i+1}$, $1S_{2i+1}$,..., $1S_{p-3}$, $0S_{p-3}$, $0S_{p-2}$, $1S_{p-2}$, $1S_{p-1}$, $0S_{p-1}$.

- Case $p \geq 7$

A *left cyclic partition* of order $2p-3$ in H_{n+1} is obtained from the *left cyclic partition* exhibited in the case $p \geq 5$ by replacing the subsequence $1S_2$, $0S_2$, $0S_3$, $1S_3$, $1S_4$, $0S_4$, $0S_5$, $1S_5$ by $1S_2$, $0S_2 \cup 1S_3$, $0S_3$, $0S_4$, $1S_4 \cup 0S_5$, $1S_5$.

Lemma 6. *H_4 admits left cyclic partitions of orders* 5 *and* 7.

Proof.

- A *left cyclic partition* of order 5 in H_4 is the following :
{0000,1101}, {0001,1100,0010,1111}, {0110,1011}, {0100,0111,1001,1010}, {0011,0101,1000,1110}.
- A *left cyclic partition* of order 7 in H_4 is the following :
{0000,1101}, {0001,1100}, {0011,1110}, {0010,1111}, {0110,1011}, {0100,0111,1001,1010}, {0101,1000}.

Lemma 7. *H_n, $n \geq 4$, admits a* left cyclic *partition of order* $2^{n-1}-1$.

Proof. Consider the sequence $\{u_i; 0 \leq i \leq 2^{n-1}-1\}$, defined by $u_i = bin(i) \oplus bin(i/2)$,where $bin(x)$ is the n-position binary representation of the integer x, and symbol / denotes integer division. It can be easilly checked that this sequence corresponds to a *hamiltonian cycle* in H_{n-1}. Now, let us denote $v_i = u_i \oplus 1 \oplus 2^{n-2}$ (i.e. v_i is obtained from u_i by changing the first and last bits) and $N = 2^n$. It is also easy to check that $\{v_i; 0 \leq i \leq 2^{n-1}-1\}$ is a *hamiltonian cycle* of H_{n-1}.Now, observe that $0u_i \oplus 1v_i = 2^{n-1} \oplus (u_i \oplus v_i) = 2^{n-1} \oplus 1 \oplus 2^{n-2}$. Hence, $0u_i$ and $1v_i$ are not neighbors in the hypercube H_n . On the other hand, $u_{N-4} = 100...010$, $u_{N-2} = 10...01$, $u_{N-1} = 10...0$ and $v_0 = u_0 \oplus 1 \oplus 2^{n-2} = 10...01 = u_{N-2}$. Hence, $0u_{N-4}, 0u_{N-1}, 0u_{N-2}, 1v_0$ is a chain of H_n. Moreover, $v_{N-4} = 0...011$, $v_{N-2} = 0...0 = u_0$ and $v_{N-1} = 0...01 = u_1$. Hence $1v_{N-4}, 1v_{N-1}, 1v_{N-2}, 0u_0$ is a chain of H_n. Hence, by considering the two chains and the two previous *hamiltonian cycles,* we see that the partition $\{0u_0, 1v_0\}, \{0u_{N-4}, 1v_{N-4}\}$, $\{0u_{N-3}, 0u_{N-1}, 1v_{N-3}, 1v_{N-1}\}$, $\{0u_{N-2}, 1v_{N-2}\}$ is a *left cyclic partition.*

Proposition 3. *H_n, $n \geq 4$, admits left cyclic partitions of all odd orders from 5 to* $2^{n-1}-1$.

Proof. We proceed by induction. For $n = 4$ the result follows from lemma 6. Assuming that the result holds for $n \geq 4$, let us consider an $(n+1)$-*cube* together with an odd integer $p \in [5, 2^n - 1]$.

- Case $1 : 5 \leq p \leq 2^{n-1}-1$. The result follows from the induction hypothesis and lemma 4.
- Case $2 : 2^{n-1}-1 < p < 2^n - 1$. There is an odd integer q, $7 < q < 2^{n-1}-1$, such that $p = 2q - 1$ or $p = 2q - 3$. The result follows from the induction hypothesis and lemma 5.
- Case 3 : $p = 2^n - 1$. The result follows from lemma 7.

Theorem 1. *The cycles generated by the parallel chip firing game associated with n-cube orientations, $n \geq 4$, are of even lengths from 2 to 2^n, and of odd lengths different from 3 and ranging from 1 to $2^{n-1}-1$.*

3 Conclusion

In this paper, we have given, in the particular case of *parallel evolutions,* a range of possible periods which can be generated by the chip firing game associated with n-*cube* orientations. The question of the existence of cycles with length greater than 2^n remains to be clarified. We suggest, for the case of n-cubes, to consider a recurrent approach based on the sub-evolutions induced in every face of the n-cube by a *block sequential evolution.*

References

1. M. A. Kiwi, R. Ndoundam, M. Tchuente and E. Goles *No polynomial bound for the period of the parallel chip firing game on graphs,* Theoretical Computer Science, 136, pp. 527-532, 1994.

A Fast Multifrontal Solver for Non-linear Multi-physics Problems*

Alberto Bertoldo, Mauro Bianco, and Geppino Pucci

Department of Information Engineering, University of Padova, Italy
{cyberto,bianco1,geppo}@dei.unipd.it

Abstract. The paper presents a highly optimized implementation of a multifrontal solver for linear systems arising in the FEM simulation of multi-physics problems related to the behaviour of porous media. The solver features a careful preprocessing phase that is crucial to considerably speed up both system assembly and Gaussian elimination. When run on a number of relevant test cases, the proposed solver compares very favourably with both its previous unifrontal counterpart and two general multifrontal solvers well known in the literature.

1 Introduction

The FEM simulation of non-linear fully coupled multi-physics problems is a challenging and, for many aspects, yet open problem. Due to their complexity, these problems require physical models based on very large, non-linear systems of PDEs. The specific multi-physics problem targeted in this paper is the simulation of porous media [4]. The computational kernel of the simulation is a solver of large unsymmetric linear systems, whose sparsity pattern does not change through iterations.

Due to their size, by virtue of their structure, and for stability reasons, the above mentioned linear systems are often solved using *frontal* approaches [6]. For the specific application under consideration, previous work [4] has studied the accuracy of standard pivoting strategies for the subfamily of *unifrontal* methods. Subsequently, Bianco [3] has proposed the novel *implicit minimum-degree* pivoting strategy, which affords an efficient implementation in terms of fast routines for dense matrices (e.g., the BLAS level 3 library.)

This paper extends previous work to the *multifrontal* approach, in which the assembly of the linear system proceeds in a tree-like fashion on the entire mesh. On four sizable test cases from our application domain, the resulting solver outperforms both the previous unifrontal solver of [3] tailored for the specific problem at hand, and other two prominent general multifrontal solvers: MUMPS [1] and SuperLU [5].

2 Logical Solution Algorithm

As stated above, the multifrontal approach entails a tree-like assembly of the linear system from repeated joins of subsystems corresponding to larger and larger regions of

* This work was supported, in part, by MIUR of Italy under COFINLAB and PRIN grants. Further support for the first and second author came from Consorzio "Roma Ricerche" and CISM of Italy, respectively.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 614–617, 2004.
© Springer-Verlag Berlin Heidelberg 2004

the FE mesh. Given an *assembly tree* and one of its internal nodes t, let $\mathbf{A}^{[t]}\mathbf{u}^{[t]} = \mathbf{b}^{[t]}$ be its corresponding system after assembly of the associated region, and $\bar{\mathbf{A}}^{[t]}\bar{\mathbf{u}}^{[t]} = \bar{\mathbf{b}}^{[t]}$ be the *reduced system* obtained by eliminating *Fully-Summed* (FS) variables. Let also $\mathrm{lx}(t)$ and $\mathrm{rx}(t)$ denote, respectively, the left and the right children (subregions) of node (region) t. The logical sequence of steps that must be performed once two children regions are ready to be joined is the following:

1. **Assembly:** Merge the (reduced) matrices of the two (children) regions into a single (not reduced) matrix relative to the joined (parent) region. In the merged matrix $\mathbf{A}^{[t]}$, rows/columns relative to variables in the left child region are placed first, while the ones relative to the right child region are appended later.
2. **Swap:** Pack FS rows and columns into blocks (called *FS blocks*) towards the bottom and right sides of the matrix. (In order to make the computation efficient, and to ease subsequent assembly phases, we employ a minimal swap algorithm which does not move those rows/columns which are already in their destination blocks.)
3. **Copy:** Copy the FS blocks into temporary buffers which will be passed as input parameters to optimized linear algebra routines for fast UL factorization.
4. **Elimination:** Let

$$\mathbf{A}^{[t]} = \left[\begin{array}{c|c} \mathbf{N} & \mathbf{R} \\ \hline \mathbf{C} & \mathbf{S} \end{array}\right], \text{ where } \begin{cases} \mathbf{N} \in \mathbb{R}^{(f_t - n_t)\times(f_t - n_t)}, \mathbf{R} \in \mathbb{R}^{(f_t - n_t)\times n_t}, \\ \mathbf{C} \in \mathbb{R}^{n_t \times (f_t - n_t)}, \mathbf{S} \in \mathbb{R}^{n_t \times n_t} \end{cases} \quad (1)$$

   $\mathbf{C}$, $\mathbf{R}$ and $\mathbf{S}$ represent the three FS blocks (f_t and n_t are defined later). Obtain the factorization $\mathbf{S} = \mathbf{UL}$, compute $\mathbf{U}' = \mathbf{RL}^{-1}$ and $\mathbf{L}' = \mathbf{U}^{-1}\mathbf{C}$, and finally compute the Schür complement with respect to $\mathbf{S}$ as $\bar{\mathbf{A}}^{[t]} = \mathbf{N} - \mathbf{U}'\mathbf{L}'$.
5. **Strip:** Strip blocks $\mathbf{U}'$ and $\mathbf{L}'$ and factors $\mathbf{U}$ and $\mathbf{L}$ and store them to finally solve the system with substitution algorithms. The region is now associated with the reduced matrix $\bar{\mathbf{A}}^{[t]}$.

In our application all the matrices involved in the above steps are dense [3], hence the partial factorization of $\mathbf{A}^{[t]}$ and matrix $\bar{\mathbf{A}}^{[t]}$ can be obtained efficiently in terms of BLAS level 3 routines.

We call the previous algorithm a *logical algorithm,* since it needs an actual implementation. To this aim, at the ouset we perform a preprocessing phase, called *symbolic analysis,* in order to gather relevant data of the solution process. In [3] it is shown how to merge the step 2 (swap) and the step 3 (copy) steps. Below, we describe a faster and more advanced merging of steps 1 (assembly), 2 (swap), and 3 (copy) into what we call a *super-assembly* phase.

3 Symbolic Analysis

Our symbolic analysis takes advantage of the fact that the rows and columns that become fully-summed at each node of the assembly tree are the same *at every iteration* of the solver for a given mesh, so, for each region, the position of each variable in the blocks **N, C, R,** and **S** can be computed beforehand. Moreover, we can keep the sub-matrices **N, C, R,** and **S** of $\mathbf{A}^{[t]}$ within separate buffers $\mathbf{B_N}$, $\mathbf{B_C}$ $\mathbf{B_R}$ and $\mathbf{B_S}$. During symbolic analysis, the following information is computed for each node of the assembly tree t:

f_t : the dimension of $A^{[t]}$, which is called *front size;*
n_t : number of variables of $A^{[t]}$ that become fully-summed;
$v_t(\cdot)$: for $i \in \{1, \ldots, f_t\}$, $v_t(i)$ is the global index of the i-th variable of $A^{[t]}$ soon after the swap step of the logical algorithm. Clearly, variables $v_t(i)$ for $i \in \{1, \ldots, f_t - n_t\}$ are not FS, while variables $v_t(j)$ for $j \in \{f_t - n_t + 1, \ldots, f_t\}$ are FS.

We assume to have available at a leaf t the corresponding Schür complement $\bar{\mathbf{A}}^{[t]}$ obtained from static condensation of the associated element [3]. For what concerns composite regions, let $\bar{\mathbf{A}}^{[\mathrm{lx}(t)]}$ and $\bar{\mathbf{A}}^{[\mathrm{rx}(t)]}$ be the Schür complements of the left and right subregions of t, respectively. From the previous steps, we have available the correspondence between their row (resp., column) indexes and global variable indexes, respectively in $v_{\mathrm{lx}(t)}(\cdot)$ and $v_{\mathrm{rx}(t)}(\cdot)$. From these, we can easily compute two index maps $\gamma_l^{[t]}(k)$ and $\gamma_r^{[t]}(k)$ that, for the k-th variable in $A^{[t]}$, indicate the indexes associated with that variable in the matrices of the two children as follows:

$$\gamma_l^{[t]}(k) = \begin{cases} h \;\; \text{if } \exists h < \infty : \\ \quad v_t(k) = v_{\mathrm{lx}(t)}(h); \\ \infty \text{ otherwise.} \end{cases} \qquad \gamma_r^{[t]}(k) = \begin{cases} h \;\; \text{if } \exists h < \infty : \\ \quad v_t(k) = v_{\mathrm{rx}(t)}(h); \\ \infty \text{ otherwise.} \end{cases} \tag{2}$$

Using these definitions, we can think of k as the destination row/column index of a variable in the assembled region and of $\gamma_l^{[t]}(k)$ and $\gamma_r^{[t]}(k)$ as the source indexes of that variable, respectively in the left component and in the right component during the assembly process (in case the k-th variable is not present in one of the two children, we map the variable to a "default" value of ∞, which is considered an index for a zero element). These quantities can be computed for each node in the tree by simulating the assembly process of the logical algorithm without any floating point operation.

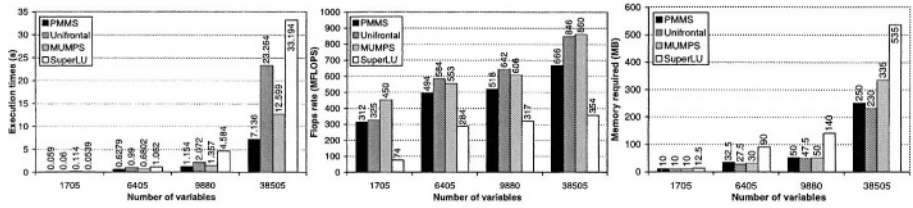


Fig. 1. Performance results for test cases using the four solvers.

The main purpose of the index maps defined in Eq. (2) is that of generating the submatrices **N, C, R** and **S** of $\mathbf{A}^{[t]}$ starting from $\bar{\mathbf{A}}^{[\mathrm{lx}(t)]}$ and $\bar{\mathbf{A}}^{[\mathrm{rx}(t)]}$, and placing them directly into buffers $\mathbf{B_N}$, $\mathbf{B_R}$, $\mathbf{B_C}$, and $\mathbf{B_S}$, thus merging together the first three steps of the logical algorithm. For efficiency reasons, we make sure that buffer $\mathbf{B_N}$ reuse as much memory space as possible from the one previously allocated to $\bar{\mathbf{A}}^{[\mathrm{lx}(t)]}$. In order to avoid overwriting entries of the latter matrix prior to their use, it is sufficient to compute $\mathbf{B_S}$, $\mathbf{B_R}$, and $\mathbf{B_C}$, before $\mathbf{B_N}$. In fact, observe that $\bar{\mathbf{A}}^{[\mathrm{lx}(t)]}(i,j)$ is used when updating $\mathbf{A}^{[t]}(m,n)$, with $\gamma_l^{[t]}(m) = i$ and $\gamma_l^{[t]}(n) = j$. By the properties of the precomputed maps, the only entries of $\bar{\mathbf{A}}^{[lx(t)]}$ in common with $\mathbf{B_N}$ that could be overwritten are

those entries (i, j) such that both $\gamma_l^{[t]}(i)$ and $\gamma_l^{[t]}(j)$ are greater than $f_t - n_t + 1$, which have already been used, since they contribute to entries in $\mathbf{B_S}$, $\mathbf{B_R}$, and $\mathbf{B_C}$.

4 Performance Results

This final section compares some performance figures of the presented solver (dubbed PMMS, *Porous Media Multifrontal Solver*) with the ones of the previous unifrontal version [3] and those of two well-known general solvers: SuperLU version 3.0 [5] and MUMPS version 4.3 [1]. The test cases are four FE (square) meshes yielding linear systems with sizes varying from 1705 to 38505 variables. Our platform is an IBM Power3 processor at 375MHz and 4 Gbyte of memory.

The results of the tests are shown in Figs. 1 .(a), 1 .(b), and 1 .(c) reporting, respectively the execution times, the FLOP rates, and the memory requirements for each iteration of the solver. On our grids, PMMS largely outperforms the unifrontal solver of [3] and is always faster than both MUMPS and SuperLU except for the smallest, and least significant test case (see Fig. l.(a)). The worse performance exhibithed by SuperLU w.r.t. MUMPS had also been observed in [2]. For the sake of fairness, the running times measured for SuperLU do not include the assembly of the full matrix of the system.

Fig. l.(b) shows that MUMPS exhibits a rather larger rate of execution of floating point operations than PMMS, which suggests that the former solver is better tuned than ours (which is still away from being optimized) although it resorts to an algorithm of higher complexity. From this point of view, matching the floating point performance of MUMPS appears as a challenging objective for the final release of our solver. For the memory usage, in Figure 1 .(c) it can be noted how MUMPS has a slightly smaller space requirement than PMMS for the biggest test case, while SuperLU always has the largest. Finally, we want to remark that the numerical accuracy of the various solvers with respect to *modified component-wise backward error* metric [4] (figures not shown here due to space constraints) are all comparable with the roundoff unit of the 64-bit floating point precision used.

References

1. P.R. Amestoy, I.S. Duff, J.Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Meth. Appl. Mech. Eng.* 184:501–520, 2000.
2. P.R. Amestoy, I.S. Duff, J.Y. L'Excellent, X.S. Li Analysis and comparison of two general sparse solvers for distributed memory computers. *ACM Trans. onMeth. Soft. (TOMS).* 24(4):388–421, 2001.
3. M. Bianco A high-performance UL factorization for the frontal method. In *Proc. Int. Conf. on Computational Science and its Applications (ICSSA 2003),* pages 893–902, Montreal, CA, May 2003. Lecture Notes in Computer Science 2667, Springer-Verlag.
4. M. Bianco, G. Bilardi, F. Pesavento, G. Pucci, and B. A. Schrefler. A frontal solver tuned for fully-coupled non-linear hygro-thermo-mechanical problems. *Int. J. Num. Meth.* Eng.57(13):1801–1818, 2003.
5. J.W. Demmel and S.C. Eisenstat and J.R. Gilbert and X.S. Li and J. W. H. Liu. A Supernodal Approach to Sparse Partial Pivoting. *SIAM J. Mat. Anal, and Appl.* 20(3):720–755, 1999.
6. I. S. Duff and J. A. Scott. The design of a new frontal code for solving sparse, unsymmetric *systems. ACM Trans. Math. Soft.* (TOMS) 22(l):30–45, 1996.

Modelling of Interaction between Surface Waves and Mud Layer

Lale Balas

Gazi University, Faculty of Engineering and Architecture, Civil Engineering Department, 06570 Ankara, Turkey
lalebal@gazi.edu.tr

Abstract. The analytical and numerical modelling of interaction between the mud layer at the sea bottom and the surface waves have been presented. In the simulations the theory for linear water wave movement in a two-layer viscous fluid system has been considered. The upper layer represents water and the lower layer represents fluid-mud. The type of the bottom material over which waves are propagating is assumed to be similar to a viscous fluid, characterized by a viscosity and density greater than the overlying fluid. It is assumed that the two fluids are incompressible and isotropic, and the rigid strata is smooth and impermeable. At the surface the height of the surface wave is specified and the forced interfacial wave is determined. Developed model solves the equations of motion for an incompressible fluid by composite finite difference-finite element approximations on a staggered scheme. Results of analytical and numerical solutions are compared with the experimental results and favourable agreement has been obtained.

1 Introduction

All over the world, at a great number of coastal areas, especially near the river mouths carrying large quantities of sediments, the sea bottom is covered by the mud layers. Surface waves start to interact with the mud layer at the sea bottom when they enter to the intermediate or shallow wave regions. This interaction generally causes the dissipation of surface waves step by step and suspension and transportation of mud due to the pressure generated at the sea bed. The dissipation caused by the mud layer is greater than the dissipation caused by the friction of rigid sea bottom. Under surface wave action, mass transport occurs in the mud layer. Gade[1] who is one of the former reasearchers that investigated the interaction of surface waves with the mud layer at the bottom, modelled the mud layer as a fluid with a high viscosity. Dalrymple and Liu[2], analytically investigated the interaction of surface waves with the viscous mud layer for variable layer thicknesses. Jiang and Zhao[3], investigated analytically the propagation of waves over fluid mud layer and worked with nonlinear surface waves. Wen and Liu[4], searched for the effects of viscous dissipation on mass transport for two dimensional small amplitude interfacial waves in a two-layered fluid system. In this study, the interaction of surface waves propagating over the mud layer at the sea bottom has been investigated analytically and numerically.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 618–621, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Theory

In this study, the water layer with a thickness of h and underlying mud layer with a thickness of d are modelled as a two layered fluid system. In the modelling, it is assumed that surface waves follow the linear wave theory; liquid mud layer exhibits viscous liquid properties; there is no mixture at the interface of water layer and mud layer; liquids are incompressible; system lies on a smooth, horizontal, impervious and rigid bottom; at the interface of the system normal and shear stresses are continuous. The density of the underlying mud layer is greater than the density of the water layer. The developed numerical model solves fluid particle velocities by Navier-Stokes equations. In the numerical solution of the equations central finite difference approximations are applied. In the vertical Galerkin finite element linear shape functions are used. Water depths are divided into same number of layers at each point in the solution domain. The model hydrodynamic equations in the Cartesian coordinate system is as follows:

$$\frac{\partial u_j}{\partial x}+\frac{\partial w_j}{\partial z}=0 \tag{1}$$

$$\frac{\partial u_j}{\partial t}+u_j\frac{\partial u_j}{\partial x}+w_j\frac{\partial u_j}{\partial z}=-\frac{1}{\rho_j}\frac{\partial P_j}{\partial x}+\upsilon_j\left(\frac{\partial^2 u_j}{\partial x^2}+\frac{\partial^2 u_j}{\partial z^2}\right) \tag{2}$$

$$\frac{\partial w_j}{\partial t}+u_j\frac{\partial w_j}{\partial x}+w_j\frac{\partial w_j}{\partial z}=-\frac{1}{\rho_j}\frac{\partial P_j}{\partial y}+\upsilon_j\left(\frac{\partial^2 w_j}{\partial x^2}+\frac{\partial^2 w_j}{\partial y^2}\right) \tag{3}$$

Where, x is horizontal coordinate, z is vertical coordinate; t is time; u, w are the fluid particule velocities in x,z directions respectively; υ is kinematic viscosity; p is pressure, j=1,2 are upper and lower layer indixes.

The analytical solution of the model has been performed as well. In the analytical solution the boundary layer approximations applied by Dalrymple and Liu[2] are considered. The numerical solution method is a composite finite element-finite difference method. The governing equations are solved by the Galerkin Weighted Residual Method in the vertical plane and by finite difference approximations in the horizontal plane. The water depths are divided into the same number of layers following the bottom topography. In the depth following coordinate system, the layer thickness is proportional to the local water depth. To increase the vetical resolution, wherever necessary, grid clustering can be applied in the vertical plane. The system of nonlinear equations are solved by the Crank-Nicholson Method which has second order accuracy in time.

3 Model Application

Numerical model predictions and analytical solution have been compared with the experimental studies published in the literature. Firstly the results of laboratory experiments of De Wit and Kranenburg[5] are compared with the simulations. In the experiments, various wave amplitudes (< 0,1 m.) are tested. Wave period is 1,5 s. and initial water depth is 0,30 m. dir. Experimental flume has a length of 40 m. and width of 0.8 m. Velocity amplitudes in and over the fluid mud layer are presented in Figure

(1-a) for China clay and in Figure (1-b) for Westwald Clay. Experimental values of H(m.),d(m), ρ_1 [kg.m^{-3}], ρ_2[kg.m^{-3}], Q[m^{2}s^{-1}] and a[m] are 0.325, 0.115, 1000, 1316, 2,7.10^{-3} and 0.02 respectively for china clay and 0.35, 0.15, 1000, 1186, 5,9.10^{-3} and 0.02 respectively for Westwald clay.

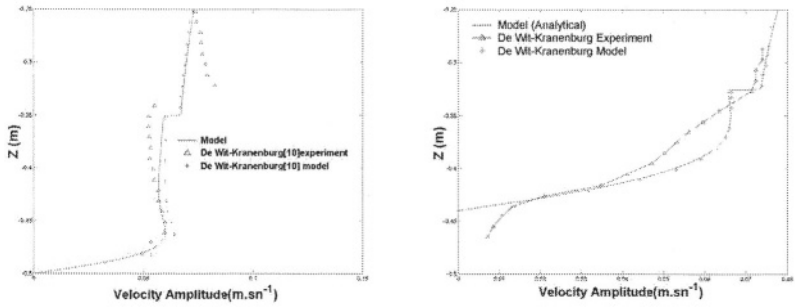


Fig. 1. Velocity amplitudes, (a) China Clay, (b) Westwald Clay

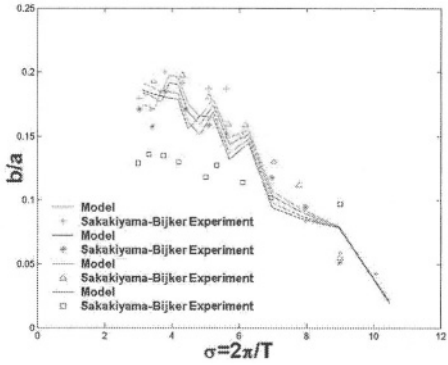


Fig. 2. Ratio of interfacial and surface wave amplitudes.

It has been observed that analytical and numerical model predictions well agree with the measurements except from the little deviations just at the interface of the mud and water layers. Secondly, the experimental results of Sakakiyama and Bijker[6] are compared with the simulations. They performed the experiment in a wave flume of 24,5 m. in length and 0.5 m. in width. For different wave periods, Sakakiyama and Bijker[6], observed the ratios of b/a (ratio of interfacial to surface wave amplitude). Simulated b/a ratios by the model are compared with the measurements and presented in Figure (2). It is seen that as the value of σ increases, the velocities in the mud layer decreases and accordingly b/a ratio decreases as well. Model simulations are favorable compared with the measurements. Calculated mean error is 0.2008% in A (□) where ρ_m is 1370 kg/m^3 and υ is 1,5 10^{-2} (m^2/s); it is 0,1971% in B (Δ) where ρ_m is 1300 kg/m^{3}and υ is 1,0.10^{-2} (m^2/s); it is 0,1489% in C (*) where ρ_m is 1210 kg/m^3 and υ is 4,0.10^{-3} (m^2/s) and it is 0,1799% in D (+) where ρ_m is 1140 kg/m^3 and υ is 1,0.10^{-3} (m^2/s). Errors are at acceptable levels. Closer to the rigid bottom, difference between

the simulations and measurements increases as also observed in the comparisons with the results of experiment done by De Wit – Kranenburg [5]. These discrepancies are basically due to the application of linear wave theory in the models.

4 Conclusions

The analytical and numerical modelling of interaction between the mud layer at the sea bottom and the surface waves have been presented. Variations in the mud layer due to the effect of waves and changes in the surface waves as a result of these variations are investigated. The numerical solution method is a composite finite difference-finite element method. The governing equations written in the Cartesian coordinates are solved by the Galerkin Weighted Residual Method in the vertical plane and by finite difference approximations in the horizontal plane. At each grid points in the solution domain, the water depth is divided into same number of layers following the bottom topography. In the analytical solution boundary layer approximation has been used. Results obtained from analytical and numerical solutions are compared with the measurements obtained from experimental studies, and comparisons have shown an encouraging agreement. The model that simulates the mud motion under wave action can be used as a practical tool in diverse coastal engineering applications.

References

1. Gade, H.G.: Effects of nonrigid, impermeable bottom on plane surface waves in shallow water. Journal of Marine Research, 16 (1958) 61-82
2. Dalrymple, R. A., Liu, P.L.: Waves over soft muds: a two-layer fluid model. Journal of Physical Oceanography. 8 (1978) 1121-1131
3. Jiang, L., Zhao, Z.: Viscous damping of solitary waves over fluid-mud seabed. Journal of Waterway, Port, Coastal and Ocean Engineering. 115(1989) 345-362.
4. Wen, J., Liu, P.,L.: Mass transport of interfacial waves in a two-layer fluid system. Journal of Fluid Mechanics. 297 (1995), 231-254
5. De Wit, P.,J.,Kranenburg, C.: On the effects of a liquefied mud bed on wave and flow characteristics. Journal of Hydraulic Research, 34 (1996), 3-18.
6. Sakakiyama, T., Bijker, E.,W.: Mass transport velocity in mud layer due to progressive waves. Journal of Waterway, Port, Coastal and Ocean Engineering, 115 (1989) 614-633.

Computational Modelling of Pulsating Biological Flow

X.S. Yang, R.W. Lewis, and H. Zhang

Civil and Computational Engineering Centre, School of Engineering,
University of Wales Swansea, Swansea SA2 8PP, UK x.s.yang@swansea.ac.uk

Abstract. Computational modelling of biological flow can have many in medicine and biomechanical engineering. Stenosis of blood vessels is the main cause of vascular disease. A mathematical model of pulsating flow through stenotic blood vessels has been formulated. The numerical simulations using finite element analysis show that the axial velocity profile is controlled by the shape of the stenotic plagues and the pulsating pressure gradient. Instability due to the vessel narrowing is closely related non-uniform stress distribution. In addition, the peak value of the shear stress may directly affect the degree and location of the rupture.

Keywords: Biological flow, finite element analysis, stenosis.

1 Introduction

Vascular disease is mainly caused by the stenosis of fatty plaque deposition and vessel narrowing. As the plaque slowly enlarges, it will cause a significant pressure gradient across the affected region, which may lead to the rupture of the blood vessels. The common clinical observations suggest that the atheromatous disease arise mainly at the bifurcation of arteries. The benefit of modelling this phenomena is obviously profound both in clinical treatment and biomedical research [1,2,5,6]. However, most models describing these phenomena only produce some basic features which are far from reality.

This paper aims at using the finite element method to simulate the flow velocity profiles for different shapes of stenotic plague and the possible rupture and patterns under the conditions of pulsating blood flow. The simulation of axial velocity distribution for regular arterial structure show that the rupture is controlled by the shape of the stenotic plaque, the pressure gradient and the amplitude of the pulsating flow. This can form the basis for further research. The implication of the simulation results will also be discussed.

2 Governing Equations

The Navier-Stokes equations for incompressible flow can be written as [3,6]:

$$\nabla \cdot \mathbf{u} = 0, \qquad \rho[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}] = \mu \nabla^2 \mathbf{u} - \nabla p, \tag{1}$$

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 622–625, 2004.
© Springer-Verlag Berlin Heidelberg 2004

where μ and ρ are the viscosity and density of the fluid, $\mathbf{u} = (U, V, W)$ and p are flow velocity and pressure of the fluid, respectively. As the blood flow is slow and viscous, so the equations for viscous flow become

$$\nabla \cdot \mathbf{u} = 0, \qquad \rho \frac{\partial \mathbf{u}}{\partial t} = \mu \nabla^2 \mathbf{u} - \nabla p. \tag{2}$$

In terms of the cylindrical coordinates (r, θ, z), the axial velocity $W(r, z, t)$ and the radial velocity $U(r, z, t)$, the nonlinear terms are negligible ($W\frac{\partial W}{\partial z} \ll 1$) in the case of the slow flow in a tube where the axial flow is dominant (i.e., $U \ll W$). The governing equation for the axial flow can be written in the cylindrical coordinates as

$$\rho \frac{\partial W}{\partial t} = \mu \frac{1}{r} \frac{\partial}{\partial r}(r \frac{\partial W}{\partial r}) - \frac{\partial p}{\partial z}. \tag{3}$$

This problem involves a moving or free boundary at $r = R(z, t)$. To simplify and convert the free boundary problem into a fixed-domain problem, we define a reduced radial coordinate $\xi = r/R(z, t)$, now equation (3) becomes

$$\rho \frac{\partial W}{\partial t} = \mu [\frac{1}{R^2} \frac{\partial^2 W}{\partial \xi^2} + \frac{1}{\xi R(z,t)^2} \frac{\partial W}{\partial \xi}] - \frac{\partial p}{\partial z} + \frac{\rho \xi}{R(z,t)} \frac{\partial R}{\partial t} \frac{\partial W}{\partial \xi}. \tag{4}$$

The boundary conditions then become

$$\frac{\partial W}{\partial \xi} = 0, \quad \text{at } \xi = 0, \qquad \mathrm{W}(\xi, \mathrm{z}, \mathrm{t}) = 0, \quad \text{at } \xi = 1. \tag{5}$$

It is clearly seen that this problem has been simplified as the one-dimensional flow problem if the pressure gradient is prescribed. Most earlier studies [1,2,6] assume the following form $\frac{\partial p}{\partial z} = A\cos(\omega t) + \gamma, \quad \frac{\partial p}{\partial r} = 0$, where γ is the background pressure gradient. In fact, we can assume any known function form for the pressure gradient in terms of a Fourier expansion.

3 Finite Element Formulation

From the implementation point of view, the continuity equation can be approximated by a penalty function as $\nabla \cdot \mathbf{u} = -\lambda^{-1} p$ or $p = -\lambda \nabla \cdot \mathbf{u}$ where $\lambda \to \infty$. For numerical purpose, $\lambda \gg 1$ is a large number [3,6]. We now have a general form

$$\mathbf{M}\dot{\mathbf{u}} = \lambda \mathbf{G}\mathbf{u} + \mathbf{K}\mathbf{u} + \mathbf{g}, \tag{6}$$

where $\mathbf{M} = \int_\Omega \rho N_i N_j d\Omega$ is the mass matrix, $\mathbf{K} = \int_\Omega \nabla N_i \mu \nabla N_j d\Omega$ is the stiffness matrix, $\mathbf{G}$ is the penalty term, and $\mathbf{g}$ is the term due to body force and natural boundary conditions. If $\lambda \to \infty$, the requirement $\mathbf{G}\mathbf{u} \to 0$ leads to the trivial solution $\mathbf{u} = 0$ if $\mathbf{G}$ is nonsingular. To obtain the nontrivial solution, this implies that $\mathbf{G}$ must be singular. The singularity of $\mathbf{G}$ can be fulfilled by the reduced integration for the λ-term with a Gauss-Legendre rule of one-order lower than normal rules [3,4,7].

The nonlinear finite element equation can be solved using the time-stepping and iteration technique. Once the flow velocity **u** has been computed at a time level, the pressure $p = -\lambda \nabla \cdot \mathbf{u}$ can be obtained using the smoothing technique for the flux calculation.

4 Simulations and Results

By using the finite element method in the previous section, the flow profiles are simulated for a given pulse. We will first focus on the axial velocity distribution, then calculate the mean shear stress at vessel walls and blood flux variation.

4.1 Axial Velocity Distribution

The velocity profiles for different stenotic shapes at the peak systole are shown in Figure 1. The left figure is the flow distribution through a tube where there is essentially no stenosis, and thus the velocity is in fact paraboloid. The middle profile is irregular stenosis leading to the distorted velocity profile. The velocity profile on the right is the regular stenosis on the hexagonal shape, and its velocity distribution is correspondingly modulated by the stenosis.

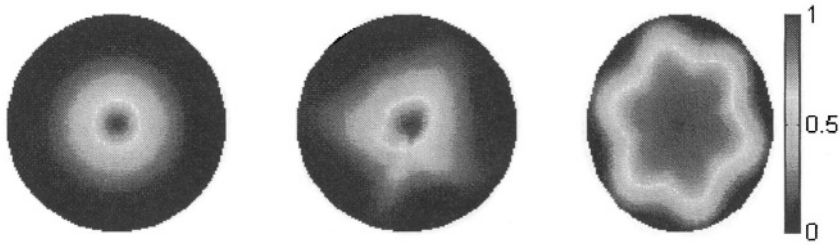


Fig. 1. Velocity profile for different stenotic shapes at the peak systole. The flow velocities are normalized so that $W = 0$ at $\xi = 1$ and $W = 1$ at $\xi = 0$.

From these velocity profiles, we can see that the velocity profile is controlled by the shape of the stenosis and thus the blood flux is subsequently affect the degree of the stenosis.

4.2 Shear Stress and Rupture Prediction

For a given pulse, one can calculate the wall shear stress and flux variation with time after the peak R-wave of the given pulse. The mean shear stress variation and blood flux variation with time. The characteristics of the computed the shear stress variation is consistent with the experimental results by Oyre et al [6]. It is worth pointing out that there exists a peak for shear stress for a given

pulsating pressure gradient, and thus the peak shear stress could directly affect the rupture of the stenotic plagues.

Mechanism of rupture of the stenotic plagues is complicated. Generally speaking, the stenosis weakens the lumen of the blood vessel, and thus the rupture could occur if the local shear stress reach to certain value or the yield stress. For a given level of yield stress, the level of shear stress is directly related to the possibility of the rupture. Since the wall shear stress varies with time after the peak R-wave, the peak stress determine the rupture, thus may explain why sudden increase of pulsating pressure could lead to the rupture of weak blood vessels especially near the branch bifurcations.

5 Discussion

A mathematical model of pulsating flow through stenotic blood vessels has been formulated, and the finite element method has been implemented after transforming the free boundary problem into a fixed domain problem. Numerical simulations show that the axial velocity profile is controlled by the shapes of the stenotic plagues and pulsating pressure gradient. For a fixed pressure gradient, the velocity profile is distorted by the stenosis.

For a pulsating pressure gradient, the wall shear stress varies with time. It first increases sharply to a peak short after the peak pressure is reached, then it decreases nearly exponentially to a background value. The blood flux also shows the similar characteristics. Thus, the peak value of the shear stress may directly affect the possibility and location of the rupture.

References

1. Berbich L, Bensalah A, Flaud P and Benkirane, R., Nonlinear analysis of the arterial pulsatile flow: assessment of a model allowing a noninvasive ultrasonic functional exploration, *Medical Engineering,* **23,** 175-183 (2001).
2. Chakravarty, S and Mandal P K, Mathematical modelling of blood flow through an overlapping arterial stenosis, *Math. Comput. Modelling,* **19**, 59-70 (1994).
3. Langtangen, H P, *Computational Partial Differential Equations: Numerical Methods and Diffpack Programming,* Springer, 1999.
4. Lewis R W and Schrefler B A, *The finite element method in the static and dynamics deformation and consolidation of porous media,* 2nd Ed., John Wiley & Son, England (1998).
5. Moore J A, Steinman D A, Ethier C R, Computational blood flow modelling: Errors associated with reconstructing finite element models from magnetic resonance images, *J. Biomechanics,* **31**, 179-184 (1998).
6. Oyre S, Ringgaard S, Kozerke, S, Paaske, W, Erlandsen, M, Boesiger, P, and Pedersen, E., Accurate noninvasive quantitation of blood flow, cross-sectional lumen vessel area and wall shear stress by three-dimensional paraboloid modelling of magnetic resonance imaging velocity dat, *JACC,* **32**, 128-134 (1998).
7. Zienkiewicz O C and Taylor R L, *The Finite Element Method,* vol. I/II, McGraw-Hill, 4th Edition, 1991.

Effect of Heterogeneity on Formation of Shear Zones in Granular Bodies

Jacek Tejchman

Civil Engineering Department, Gdansk University of Technology,
80-952 Gdansk, Poland
tejchmk@pg.gda.pl

Abstract. Heterogeneity of granular materials triggers shear zone formation. In the paper, the FE-analysis of the effect of heterogeneity on the formation of a spontaneous shear zone inside of granular material during a plane strain compression test is presented. The numerical calculations are performed with a finite element method on the basis of a hypoplastic constitutive law extended by polar quantities: rotations, curvatures, couple stresses and a mean grain diameter used as a characteristic length. The heterogeneity in the granular body is assumed in the form of spots with a different initial void ratio. The spots are single and distributed randomly or stochastically.

1 Introduction

Localization of deformation in narrow shear zones is a fundamental phenomenon of granular material behaviour. Since shear localization is a precursor of failure of soils, it has to be taken into account when modelling granular materials. Therefore, the constitutive law has to include a characteristic length to describe the thickness of shear zones and a spacing between them. The experimental results show that the thickness of shear zones depends on many different factors as: pressure level, void ratio, direction of deformation, specimen size, mean grain diameter, grain roughness, grain hardness, surface roughness and stiffness of the surrounding structure in contact with a granular body. In addition, the distribution of microscopic inhomogeneities inherently present in granular materials and triggering shear localization seems to be an influential factor [1]-[3]. Thus, while modeling shear localization it is of major importance to use a constitutive model taking all influential factors into account.

The intention of the paper was to show the effect of the distribution of heterogeneity on the formation of shear localization during a plane strain compression test with dry cohesionless sand specimen. The FE-calculations were performed with a polar hypoplastic constitutive model [4]-[6]. In this model, a characteristic length is directly related to the mean grain diameter of the granulate.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 626–629, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Constitutive Law and FE-Input Data

The polar hypoplastic constitutive model [4]-[6] can reproduce essential features of granular bodies during shear localization depending on the void ratio, pressure level, direction of the deformation and curvature rate, mean grain diameter and grain roughness. The material parameters can be found by means of standard element tests and simple index tests. They are correlated with grain properties, i.e. they can be estimated from granulometric properties encompassing grain size distribution curve, shape, angularity and hardness of grains. The FE-analyses were carried out with the following material constants (for so-called Karlsruhe sand): e_{i0}=1.3, e_{d0}=0.51, e_{c0}=0.82, ϕ_c=30°, h_s=190 MPa, n=0.5, α=0.3, d_{50}=0.5 mm, a_c=a_1^{-1} [4].

The FE-calculations of plane strain compression tests were performed with a sand specimen which was h_o=10 cm high and b=2 cm wide (length l=1.0 m).

A quasi-static deformation in sand was initiated through a constant vertical displacement increment prescribed at nodes along the upper edge of the specimen. The boundary conditions of the sand specimen were no shear stress and couple stress at the top and bottom. To preserve the stability of the specimen against the sliding along the bottom boundary, the node in the middle of the bottom was kept fixed.

To obtain a shear zone inside the granular specimen, single weaker and stronger spots and spots distributed randomly or stochastically were assumed. The single spots were inserted into the specimen in the form of a weaker finite element (with a higher initial void ratio) or a stronger finite element (with a lower initial void ratio). The location and the distance of imperfect elements varied.

The effect of a distribution of the initial void ratio e_o in the entire specimen was also investigated. In the first case, the initial void ratio e_o was randomly distributed (by means of a random generator) in such a way that the initial void ratio was increased in every element layer by the value $a \times r$, where a *is* a constant (*e.g.* 0.0001, 0.05) and r is a random number between 0 and 1.

In addition, the initial void ratio was distributed stochastically with an exponential frequency function proposed by Shahinpoor [3], [7]:

3 FE-Results

The effect of the spacing of weaker elements in the dense specimen is demonstrated in Fig.1. Three weak elements (e_0=0.90) are located uniformly along the left side of the specimen at distance of about 25 mm. At the beginning, two shear zones always emerge expanding outward from each weaker element. Afterwards and up to the end, only one shear zone dominates (starting from the lowest weaker spot). It forms intensively between u/h_0=2.25% and u/h_0=2.84%. The remaining shear zones become passive at the beginning of intense shearing in the dominated shear zone i.e. at u/h_0=2.25%. If the weaker elements are very close to each other, the shear zone forms at the same place. In contrast to FE-calculations of simple shearing of an infinite granular layer by Shi and Chang [8], the thickness of the shear zone does not grow and is the same (independently of the spacing of weak spots).

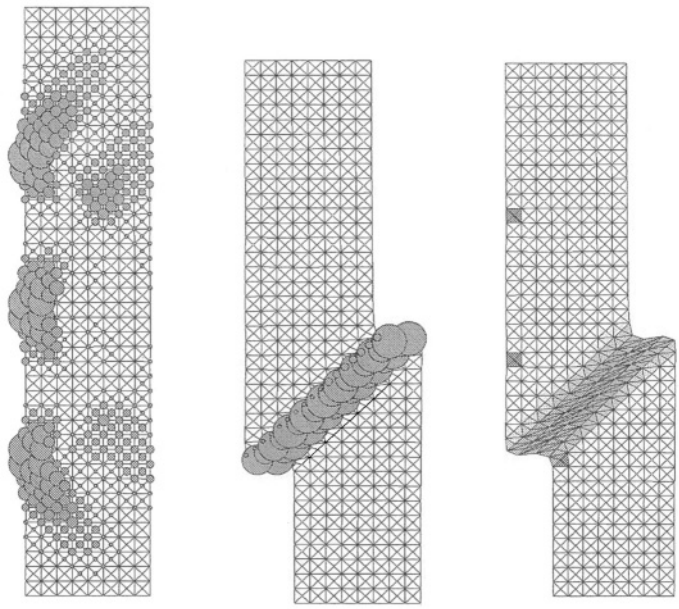


Fig. 1. Evolution of modulus of the deformation rate d in 3 weaker elements (a), deformed FE-mesh with the distribution of Cosserat rotation ω_c at u/h_0=2.5% (b) and Cosserat rotation ω_c and void ratio e at u/h_0=10% (C)

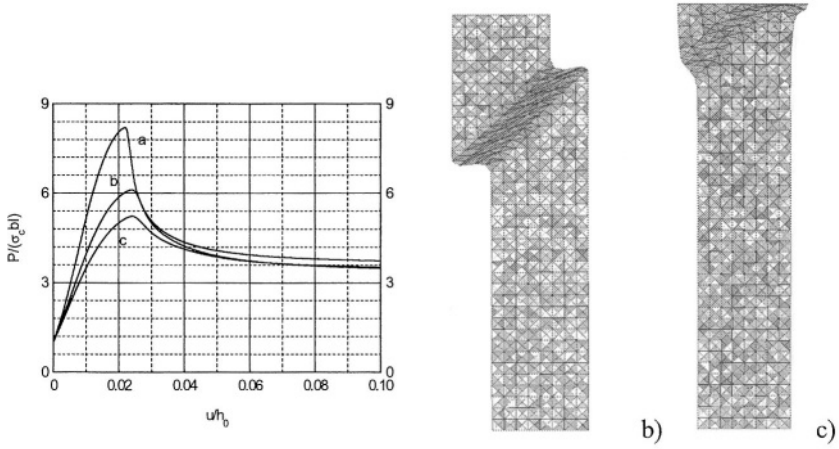


Fig. 2. FE-results: load-displacement curves and deformed FE-mesh with a distribution of void ratio at residual state (exponential frequency distribution of the initial void ratio):

a) $\tilde{e}_0$ =0.55, b) $\bar{e}_0$ =0.60 an d c) $\bar{e}_0$ =0.65 ($\bar{e}_0$ - mean void ratio)

The FE-results with a stochastic distribution of the initial void ratio in dense specimen (with a different mean void ratio) are shown in Fig.2.

As in the case of a random distribution of the initial void ratio, the shear zone can propagate in different parts of the granular specimen. The thickness of the shear zone slightly increases with increasing mean void ratio. Thus, this increase is significantly smaller as compared to the calculations with a random distribution of the initial void ratio (since the rate of softening is different in both cases).

4 Conclusions

Each single imperfection (weaker and stronger spot) induces shear zones at the beginning of loading independently of the initial density of the specimen. Thus, multiple shear zones appear which are not well developed.

The dominant single shear zone starts to form intensively in the dense specimen slightly before the peak state. It is well developed after the peak state.

The dominant shear zone is connected to a weaker spot but not to a stronger one.

The spacing of single imperfections does not influence the thickness of the shear zone. However, it influences the direction of propagation.

The distribution of the initial void ratio strongly influences the location of the shear zone and its propagation direction. The shear zone can appear in different parts of the dense specimen.

The results with a random distribution of the initial void ratio are similar as those with a stochastic distribution using an exponential frequency function.

References

1. Hobbs, B. E., Ord, A.: Numerical simulation of shear band formation in frictional-dilational materials. Ingenieur-Archiv 59 (1989) 209-220
2. Tejchman, J., Wu, W.: Numerical study on shear band patterning in a Cosserat continuum. Acta Mechanica 99 (1993) 61-74
3. Nübel, K., Karcher, C.: FE simulations of granular material with a given frequency of distribution of voids as initial condition. Granular Matter 1, 3 (1998) 105-112
4. Tejchman, J., Herle, I., Wehr, J.: FE-studies on the influence of initial void ratio, pressure level and mean grain diameter on shear localisation. Int. J. Num. Anal. Meth. Geomech. 23 (1999) 2045-2074
5. Tejchman, J., Gudehus, G.: Shearing of a narrow granular strip with polar quantities. Int. J. Num. and Anal. Methods in Geomechanics 25 (2001) 1-18.
6. Tejchman, J.: Patterns of shear zones in granular materials within a polar hypoplastic continuum. Acta Mechanica 155, 1-2 (2002) 71-95
7 Shahinpoor, M.: Statistical mechanical considerations on storing bulk solids. Bulk Solid Handling 1, 1 (1981) 31-36
8 Shi, Q., Chang, C. S.: Numerical analysis for the effect of heterogeneity on shear band formation. Proc. 16th ASCE Eng. Mech. Conf., Seattle USA (2003) 1-11

Effect of Structural Disorder on the Electronic Density of States in One-Dimensional Chain of Atoms

Maciej Wołoszyn and Bartłomiej J. Spisak

Faculty of Physics and Nuclear Techniques,
AGH-University of Science and Technology,
Al. Mickiewicza 30,
30-059 Kraków, Poland
woloszyn@novell.ftj.agh.edu.pl

Abstract. A simple model of chain containing Dirac's delta-like potentials is used to calculate the density of states in systems with structural disorder. The crystal case of equally spaced scattering centers may be seen as a reference case suitable both for numerical calculations as well as analytical ones. The structural disorder means distortion of ideal lattice seen as fluctuation of distances between adjacent sites. Such disorder is sometimes referred to as amorphous and may lead to smearing off the band boundaries resulting in a tail-like density of states.

1 Introduction

In the last decade we observe a great interest in the low dimensional systems. The rapid development of nanotechnology allows to fabricate the new class of artificial structures such as dots, wires and wells, where the motion of electrons is confined by the spatial dimensions of the system [1]. Therefore the dots, the wires and the wells are regarded as zero-, one- and two- dimensional systems, respectively.

The electronic structure of these systems is especially important for practical applications and its theoretical analysis can be based on the Bethe ansatz method [2], the concept of low energy Tomonaga-Luttinger liquid [3] and powerful numerical techniques such as the Quantum Monte-Carlo [4] or Density Matrix Renormalization Group Method [5]. Another important numerical techniques are Coherent Potential Approximation [6] and Recursion Method [7]. Especially the last two methods allow us to construct the density of states (DoS) [8] in a simple manner.

As it is well known the structural disorder, i.e. spatial fluctuations of potential, has a strong influence on the electronic phenomena in condensed matter [9]. In this paper we adopt the generalized Kronig-Penney model to the description of the chain of atoms (which can be treated as a model of quantum wire) and confine our attention to the modification of the DoS as a result of amorphisation.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 630–633, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Model

In the presented simple model which can be regarded as a generalization of the Kronig-Penney model [10], we consider the time-independent Schrödinger equation for the one-dimensional system of noninteracting electrons with effective mass m which move in a random potential $V(x)$, namely

$$\left[-\frac{\hbar^2}{2m}\frac{\mathrm{d}^2}{\mathrm{d}x^2}+V(x)\right]\psi(x)=\varepsilon\psi(x) \ , \tag{1}$$

where $\psi(x)$ is the wave function, ε – energy and $\hbar$ – the Planck constant.
In fact, the potential $V(x)$ represents the collection of N atoms and consequently scattering potentials well located at random positions x_i, namely

$$V(x)=\sum_{i=1}^{N}F_0\frac{\hbar^2}{m}\delta(x-x_i) \ , \tag{2}$$

where $F_0=\hbar^2V_0b/m$ and parameter b is defined as the width of the square interatomic barriers with V_0 being their height. The distribution of these scattering centers in the chain can be described by the probability distribution function, $P\{x_i\}$, for the random potential [11].

The limit $V_0\to\infty$ and $b\to 0$ allows us to define the constant strength of scattering potential, i.e. $V_0b=const$, and even in the case of disordered chain of atoms we can construct the wave function of electrons which satisfies (1), using the analytical expression for the wave function $\psi_i(x)$ inside the i-th square well

$$\psi_i(x)=A_i\cos(ux+\xi_i) \ , \tag{3}$$

where A_i is the amplitude, $u=\sqrt{2m\varepsilon}/\hbar$ and ξ_i is the phase.

Joining solutions (3) found at consecutive ranges gives the wave function $\psi(x)$ which must satisfy the following conditions:

$$\psi(x_i^+)=\psi(x_i^-) \ , \tag{4}$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\psi(x)\bigg|_{x_i^+}-\frac{\mathrm{d}}{\mathrm{d}x}\psi(x)\bigg|_{x_i^-}=2F_0\psi(x_i) \ , \tag{5}$$

where the finite discontinuity at the step results from integrating (1).
Of course the boundary values $\psi(x_1)$ and $\psi(x_N)$ also have to be provided to calculate A_i and ξ_i for each i-th ($i=1,\ldots,N-1$) potential well, which is achieved by applying conditions (4) and (5) at each site x_i.

The knowledge of exact form of $\psi(x)$ allows us to find the DoS function $\rho(E)$ from the relation

$$\rho(E)=\frac{\mathrm{d}S}{\mathrm{d}E} \ , \tag{6}$$

where *S(E)* is the number of zeros of the wave function $\psi(x)$ and corresponds to the number of excited states with energy $E\le\varepsilon$. Since each function $\psi_i(x)$ has

the simple form of a cosine finding all zeros is quite straightforward, however the whole procedure has to be repeated for all values of energy E for which we intend to obtain the DoS function. It allows us to calculate $S(E)$ and than the DoS from (6).

3 Results and Conclusions

The method described above provides the way to find the DoS function for disordered systems with $x_{i+1} = x_i + \overline{a} + \eta \Delta x$. The average distance between neighboring sites is denoted as $\overline{a} = \sum_{i=1}^{N} a_i$ where $a_i = x_{i+1} - x_i$.
η is a random value from the range $(-1;1)$ and Δx describes the degree of disorder. For calculations presented in this paper we assume the average distance between neighboring sites $\overline{a} = 1$, number of sites $N = 10^4$ and $F_0 = 1$ with arbitrary units.

For $\Delta x = 0$ we obtain a fully ordered chain of potentials corresponding to a pure crystal and the $\rho(E)$ function for that system is presented on all figures for reference with dashed line. It shows well known characteristics of the DoS function for one-dimensional ordered systems, that is $\rho(E) \propto E^{-1/2}$ at the edge of each band.

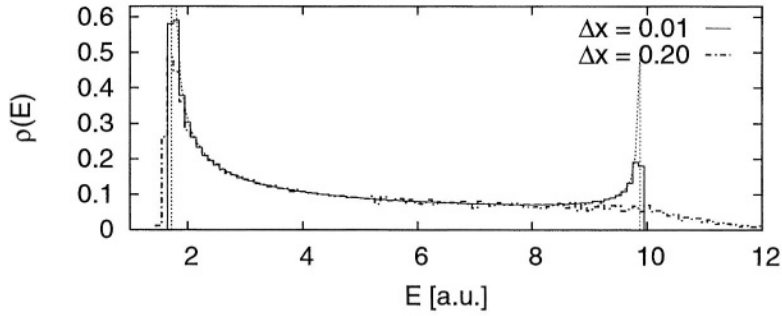


Fig. 1. DoS $\rho(E)$ for different values of Δx: 0.01 and 0.20. For reference, thin dashed line shows the DoS of the ordered system with the same F_0 and all $a_i = \overline{a}$

In case of disordered systems the characteristic deformation of the DoS function is observed (Fig. 1): previously sharp edges of the electronic bands smear off forming so-called "tails", where the localized states are expected (the problem of localization will be discussed in the forthcoming paper).

Broadening of the energy bands leads as it should be expected to filling the gaps when the bands start to overlap. The results on Fig. 2 show the vanishing gap between the first and the second band for the growing disorder.

It is well known that for example amorphous silicon exhibits band structure with the DoS showing characteristic "tails" which extend beyond the originally

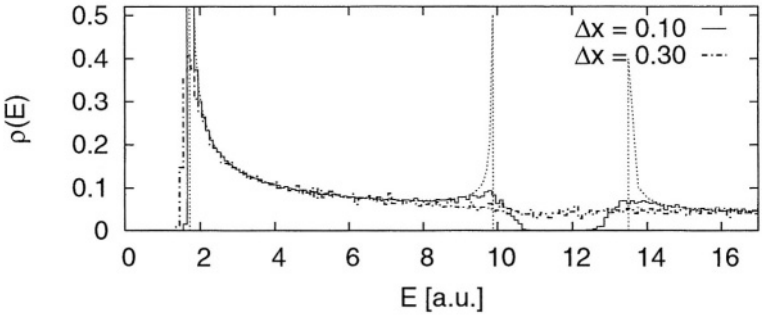


Fig. 2. The gap (or lack of it) in the DoS function $\rho(E)$ for various degrees of disorder: $\Delta x = 0.10$ and $\Delta x = 0.30$. Thin dashed line denotes the DoS of the ordered system with the same F_0 and all $a_i = \overline{a}$

placed band boundary for the crystal [12]. These tails correspond to localized states and so amorphisation may serve as an alternative possibility to introduce localized scattering centers, as it is known for the amorphous semiconducting silicon, instead of the standard doping technique.

The discussed computer simulation may help for the better understanding of how the amorphisation described in terms of parameter Δx alters the DoS. In the computer experiment we scan Δx from zero, the reference case of a pure crystal when the computer simulation restores the known analytical results, to larger $\Delta x > 0$ when we may observe how Dos reacts to the amorphisation. In particular, we report (a) the sharp boundary is smeared offwhen Δx increases and (b) the gap between bands even vanishes for Δx above a critical value.

References

1. Ando, T., Fowler, A.B., Stern, P. Rev. Mod. Phys. **54** (1982) 437
2. Castro-Alvaredo, O., Fring, A. Nucl. Phys. B **649** (2003) 449
3. Mahan, G.D.: Many-Particle Physics. Kluwer Academic/Plenum Publishers, New York, Boston,Dordrecht, London, Moscow (2000)
4. Evertz, H.G. Adv. Phys. **52** (2003) 1
5. Peschel, I., Wang, X., Kaulke, M., Hallberg, K., eds.: Density-Matrix Renormalization, A New Numerical Method in Physics. Springer, Berlin (1999)
6. Wołoszyn, M., Maksymowicz, A. TASK Quarterly **6** (2002) 669
7. Haydock, R. In Ehrenreich, H., Seitz, F., Turnbull, D., eds.: Solid State Phys. Volume 35. Academic Press, London (1980) 216
8. Ziman, J.M.: Principles of the Theory of Solids. Cambridge University Press, London (1972)
9. Anderson, P.W. Phys. Rev. **109** (1958) 1492
10. Kronig, R. de L., Penney, W.G. Proc. Roy. Soc. **A130** (1930) 499
11. Economou, E.N.: Green's Functions in Quantum Physics. Springer-Verlag, Berlin-Heidelberg-New York-Tokyo (1983)
12. Aljishi, S., Cohen, J.D., Jin, S., Ley, L. Phys. Rev. Lett. **64** (1990) 2811

The Estimation of the Mathematical Exactness of System Dynamics Method on the Base of Some Economic System

Elżbieta Kasperska and Damian Słota

Institute of Mathematics,
Silesian University of Technology,
Kaszubska 23, Gliwice 44-100, Poland,
{elakaspe, damslota}@polsl.gliwice.pl

Abstract. The purpose of this paper is to present a economic system, such as the base to estimate the mathematical exactness of the System Dynamics Method. An interesting point of view is to examine the reaction of such system to the sinusoidal characteristic of the market sale of a product and to estimate the mathematical preciseness of the method solution.

Keywords: System Dynamics Method, Simulation of the System of Production – Store – Sale, Simulation Language Dynamo.

1 Introduction

System Dynamics is a well known method of modelling, analysing and simulating dynamic processes occurring in economic and social systems. The world famous theory and application of Forrester [5,6] and Coyle [1,2,3,4] as well as and many other authors are basically concerned with continuous processes, but an addition of discrete events is also possible [7,8,9]. The attention is paid to the fact that relatively few authors are occupied with estimating the mathematical exactness of the System Dynamics Method. Naturally, this exactness may only be estimated for simple systems (the word "simple" refers to the order of the differential equation which was used to model the system).

In this paper one of such systems is presented. It contains: production, pipeline production, storage and sale. The sinusoidal characteristic of the market demand is assumed. In Figure 1 the scheme of the system assuming Lukaszewicz's symbols [10] is presented with the list of variables and parameters.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 634–637, 2004.
© Springer-Verlag Berlin Heidelberg 2004

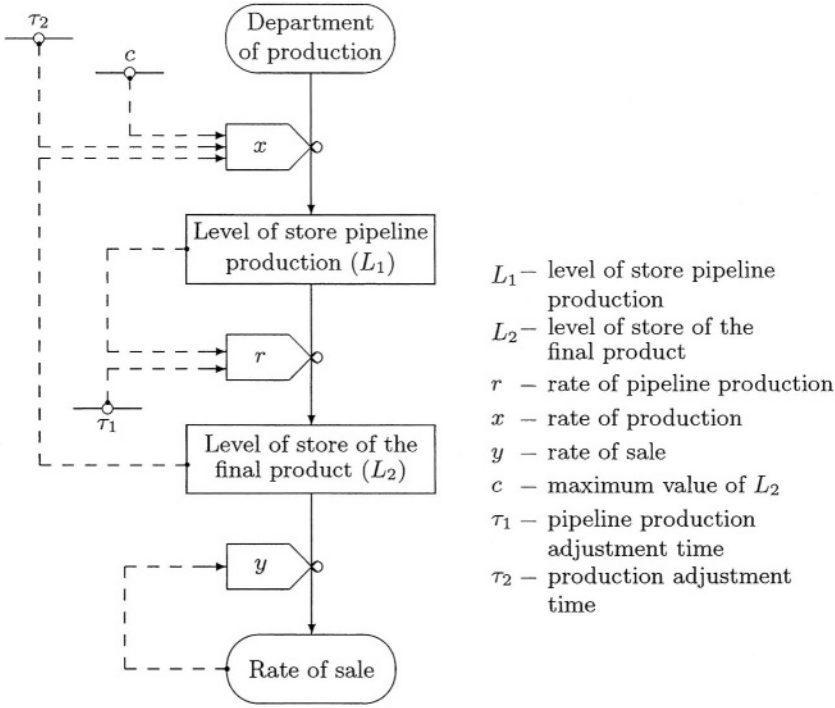


Fig. 1. The production – store – sale system with sinusoidal characteristic of sale

2 Mathematical Model of the System

As shown in Figure 1, the model consists of five equation:

$$y(t) = y(0) + p \sin\left(\frac{2\pi t}{\omega}\right), \tag{1}$$

$$x(t) = \frac{c - L_2(t)}{\tau_2}, \tag{2}$$

$$r(t) = \frac{L_1(t)}{\tau_1}, \tag{3}$$

$$\dot{L}_1(t) = x(t) - r(t), \tag{4}$$

$$\dot{L}_2(t) = r(t) - y(t), \tag{5}$$

with initial conditions:

$$x(0) = y(0), \tag{6}$$

$$\dot{x}(0) = 0. \tag{7}$$

After prognostic method we obtain (for $\tau_2 = 4\tau_1$):

$$\begin{aligned} x(t) = & \frac{6\pi\omega^3 p\tau_1 + 32\pi^3 p\omega\tau_1^3}{\left(\omega^2 - 16\pi^2\tau_1^2\right)^2 + 64\pi^2\omega^2\tau_1^2} e^{-\frac{t}{2\tau_1}} + \\ & + \frac{3\pi\omega^3 p + 16\pi^3 p\omega\tau_1^2 - 2\pi p\omega^3}{\left(\omega^2 - 16\pi^2\tau_1^2\right)^2 + 64\pi^2\omega^2\tau_1^2} t e^{-\frac{t}{2\tau_1}} + y(0) + \\ & + \frac{-6\pi\omega^3 p\tau_1 - 32\pi^3 p\omega\tau_1^3}{\left(\omega^2 - 16\pi^2\tau_1^2\right)^2 + 64\pi^2\omega^2\tau_1^2} \cos\left(\frac{2\pi t}{\omega}\right) + \\ & + \frac{p\omega^4}{\left(\omega^2 - 16\pi^2\tau_1^2\right)^2 + 64\pi^2\omega^2\tau_1^2} \sin\left(\frac{2\pi t}{\omega}\right). \end{aligned} \tag{8}$$

and:

$$L_1(t) = \left(y(0) + p\sin\left(\frac{2\pi t}{\omega}\right)\right)\tau_1 - 4\tau_1^2\dot{x}(t), \tag{9}$$

$$L_2(t) = -4\tau_1 x(t) + c. \tag{10}$$

So, the total level of storage $L(t)$ is:

$$L(t) = L_1(t) + L_2(t) = \left(y(0) + p\sin\left(\frac{2\pi t}{\omega}\right)\right)\tau_1 - 4\tau_1^2\dot{x}(t) - 4\tau_1 x(t) + c. \tag{11}$$

In the next section of the paper, we present the results of the simulation of this model usying System Dynamics method and Professional Dynamo 4.0 for Windows.

3 Simulation of the Dynamics of the Production – Store – Sale System

The Professional Dynamo 4.0 for *Windows* provides two types of integration:

- Euler method,
- Runge–Kutta method.

In Table 1 some of the errors in the simulations of these two types are presented. The analysis of errors underlines a considerable exactness, which, in turn, proves the usefulness of the System Dynamics Method, for simulating various economic systems. Moreover, it is used in real chance systems, when their size and complexity make it impossible to apply the exact methods. If the Euler's method is used, attention should be paid to the exactness of the results with a proper choice of the simulation step. Consequently, if the Runge-Kutta method is used, the simulation step is automatically partitioned to obtain a solution for a given exactness (with definite parameter *REL_ERR*). Moreover, this includes longer calculation time.

Table 1. The errors in the simulation of the rate of production $x(t)$ (Δ_{med} – medium absolute error, Δ_{max} – maximum absolute error, δ_{med} – medium relative error, δ_{max} – maximum relative error, σ – mean-square error)

	Euler method			Runge-Kutta method		
	dt			REL_ERR		
	0.1	0.5	1.0	0.1	0.01	0.001
Δ_{med}	2.62	7.49	14.35	1.99	1.99	1.98
Δ_{max}	6.85	12.89	27.60	5.67	5.67	5.65
δ_{med}	$7.51\ 10^{-3}$	$2.82\ 10^{-2}$	$5.60\ 10^{-2}$	$4.44\ 10^{-3}$	$4.44\ 10^{-3}$	$4.35\ 10^{-3}$
δ_{max}	$4.86\ 10^{-2}$	$2.05\ 10^{-1}$	$4.14\ 10^{-1}$	$1.93\ 10^{-2}$	$1.93\ 10^{-2}$	$1.82\ 10^{-2}$
σ	3.13	8.35	15.96	2.49	2.49	2.48

4 Conclusion

After modelling and simulation of the production – store – sale system, we have reached the following conclusions:

a) The System Dynamics Method is a very simple universal method, and, at the same time, its mathematical exactness is sufficient or can even be improved with more sophisticated methods of integration (like Runge-Kutta method).
b) The confidence for mathematical exactness of the System Dynamics Method in that paper was consolidated; that fact can help for development both:
   - further work for sophisticated tools of simulation complex system,
   - further application of the System Dynamics Method.

References

1. Coyle, R. G.: Management System Dynamics. John Wiley & Sons, New York (1977)
2. Coyle, R. G., Wolsterholm, E. P.: Modelling discret events in System Dynamics model. A case study. Dynamica **6** (1980) 21–27
3. Coyle, R. G.: System Dynamics Modelling. A Practical Approach. Chapman & Hall, London (1996)
4. Coyle, R. G.: The practice of System Dynamics: milestones, lessons and ideas from 30 years experience. System Dynamics Rev. **14** (1998) 343–365
5. Forrester, J. W.: Industrial Dynamics. MIT Press, Massachusetts (1961)
6. Forrester, J. W.: Principles of Systems. Cambridge Press, Massachusetts (1972)
7. Kasperska, E., Mateja-Losa, E., Słota, D.: Some extension of System Dynamics method – practical aspects. In Deville, M., Owens, R. (eds.): Proc. 16th IMACS World Congress. IMACS, Lausanne (2000) 718–11 1–6
8. Kasperska, E., Słota, D.: Two different methods of embedding the optimization in simulation on model DYNBALANCE(2-2). In Davidsen, P. I., Mollona, E. (eds.): Proc. 20th Int. Conf. of the System Dynamics Society. SDS, New York (2003) 1–23
9. Kasperska, E., Słota, D.: Mathematical Method in the Management in Conceiving of System Dynamics. Silesian Technical University, Gliwice (2000) (in Polish)
10. Łukaszewicz, R.: Management System Dynamics. PWN, Warsaw (1975) (in Polish).

Size of the Stable Population in the Penna Bit-String Model of Biological Aging

K. Malarz, M. Sitarz, P. Gronek, and A. Dydejczyk

Department of Applied Computer Science, Faculty of Physics and Nuclear Techniques, AGH University of Science and Technology, al. Mickiewicza 30, PL-30059 Kraków, Poland
malarz@agh.edu.pl

Abstract. In this paper the Penna model is reconsidered. With computer simulations we check how the control parameters of the model influence the size of the stable population.

1 Introduction

The Penna model [1] is popular and efficient way to simulate biological aging in genetically heterogeneous population. It is based on *the mutation accumulation* theory which claims that random hereditary deleterious mutations accumulate over several generations in the species genome.

The model is governed by a set of parameters such as environmental capacity N_{max} so that the actual population $N \leq N_{max}$. The birth rate B is the number of offsprings from a single parent per evolution time step, providing the parent reached age R and is still below maximum reproduction age E. Other parameters which control the growth of the population are: number of mutations M injected into the baby's genome, apart from the already inherited, and threshold value T of *active* bad mutations above which items are eliminated for this only reason.

In this paper we would like to check how model control parameters (described in the next section) influence the size $N(t)$ of the stable population for long times $t \to \infty$ when *mutational meltdown* is avoided.

2 Model

In the Penna model [1] each individual in the population is characterised by its genome — the N_{bit}-long binary string. Time is measured by a discrete variable t. In each time step $(t \to t+1)$, for each individual of age a, the number of bits set to one in the first a positions i in genome is calculated. Bit '1' represents a bad mutation activated when its position $1 \leq i \leq a$, however mutations above $a < i \leq N_{bit}$ are not harmful. If the number of *bad* and *active* mutations is greater or equal to the threshold value T an individual dies due to too many deleterious mutations and so called *genetic death* takes place.

Individuals compete among themselves for food and territory: each of them may be removed from the population with probability $N(t)/N_{max}$, where N_{max}

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 638–641, 2004.
© Springer-Verlag Berlin Heidelberg 2004

represents the maximal environmental capacity. In other words, we introduce the Verhulst's factor in order to avoid exponential growth of the population size to infinity. The number N_{bit} of bits in genome restricts also maximal age of individuals.

We start the simulation with N_{ini} individuals which genomes contain only '0' (no bad mutations). The population reproduces asexually. If the individual is older than minimum reproduction age R it is able with the probability b to give B offsprings. The offspring's genome is a copy of parent's one, exposed during replication to M harmful mutations. Each mutation occurs with probability m at randomly chosen position in the genome. We also introduce a maximum reproduction age E, so the individual older than E do not clone itself any more.

3 Results

We would like to refer our results to the standard one (labelled as 'std' in all figures) which we assume to be for $N_{bit} = 32$, $N_{max} = 10^6$, $n(0) = N(0)/N_{max} = 0.1$, $R = 8$, $E = 32$, $T = 3$, $M = 1$, $B = 3$, $m = 1.0$ and $b = 1.0$. The maximal environment capacity N_{max} is fixed during simulation. The initial concentration of individuals in respect to maximal environmental capacity do not influence the results (Fig. 1(a)).

As one may expect the larger probability of reproduction $0 \leq b \leq 1$ gives the larger population size (Fig. 1(b)). The population size decreases when the probability of mutations $0 \leq m \leq 1$ increases (Fig. 1(c)).

The same situation is for integer variables B and M (see Figs. 1(d) and 1(e), respectively). If the number of mutations is too large (i.e. $M > 7$) the mutational meltdown occurs. Also too large value of the number of offsprings (e.g. $B \geq 20$) may lead to catastrophe: too rapid growth of the population may cause that in one time step the environmental maximal capacity N_{max} will be exceeded. As a consequence of that the population will vanish *only* due to Verhulst's factor and not the genetic one.

In our implementation products Mm and Bb may be considered as the average number of mutations per individual's genome and as the average number of offsprings given every year per reproductive individual (i.e. in age $R \leq a \leq E$), respectively.

In Fig. 1(f) the influence of the threshold T on the population size is presented. Decreasing the threshold T results in increasing of the stable population size. Setting $T = N_{bit}$ eliminates genetic death and again only Verhulst's factor determines $N(t \to \infty)$.

Delay in starting the reproduction decreases the population size as presented in Fig. 1(g). Obviously, setting $R > N_{bit}$ destroys population because individuals never replicate themselves. For $R = 1$ — when all newly born babies become adults as quickly as possible — a very strong oscillations appear (not shown).

Fig. 1(h) shows the influence of the maximal reproduction age E on the stable population size. For $E = 8 (= R)$, when individual reproduce *only once* (semelparous organisms oscillations in the population size appears (not shown).

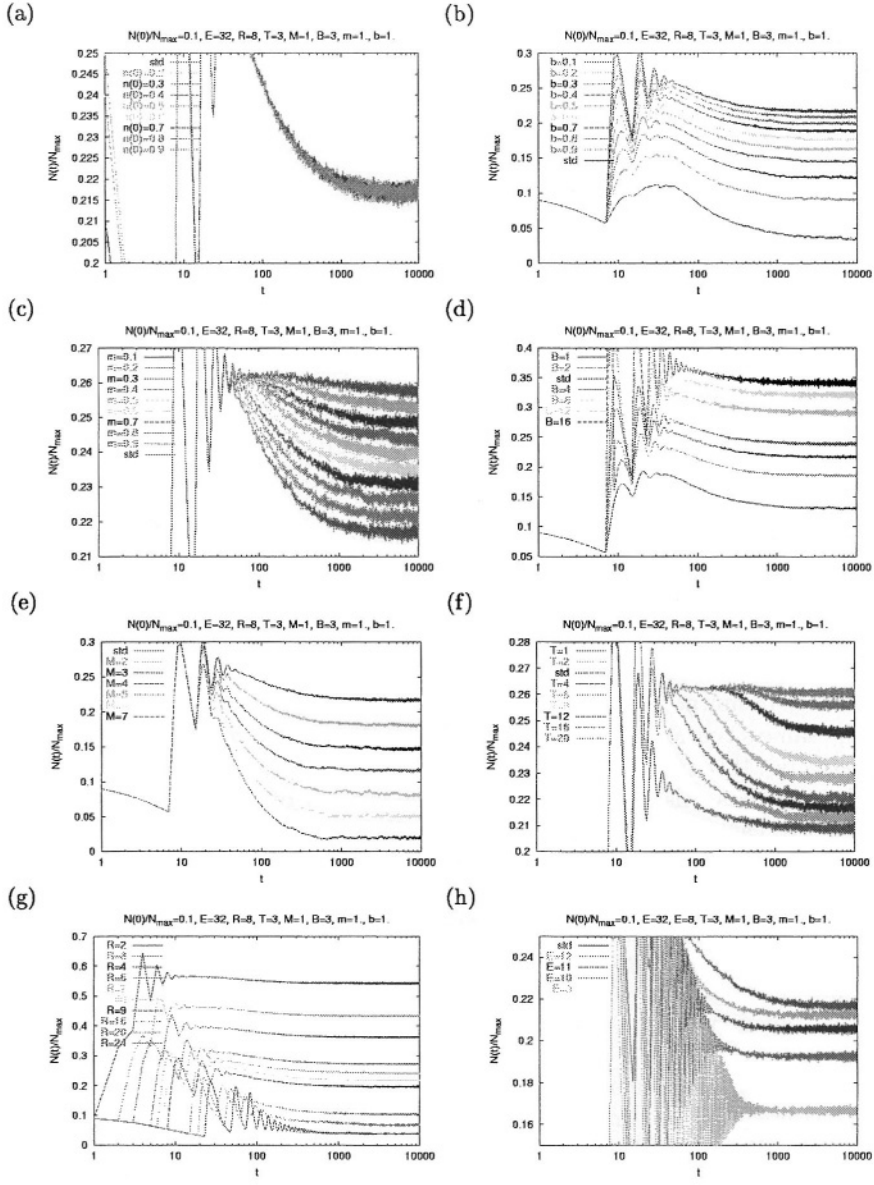


Fig. 1. The influence of the model control parameter (a) $n(0)$, (b) b, (c) m, (d) B, (e) M, (f) T, (g) R, and (h) E on the population size.

4 Conclusions

The population size of the stable population in the Penna model was already investigated for a certain purpose, e.g.: showing advantage of sexual reproduction on asexual one, studying sexual fidelity of males vs higher reproduction rate, modelling fishing and/or hunting, explaining mystery of Pacific salmon senescence, analysing pray-predator systems, recovering demographic data, modelling the oldest old effect, modelling consequence of parental care, studying the individuals fertility, studying the migration effect, and others.

In this paper we confine our attention to the basic features of the model. Main results were collected in Figs. 1. We may conclude the following:

Different time scales must be accounted for, if the 'stable' refers to different properties observed in the computer experiment; total population $N(t)$ shows values that do not change significantly after about one hundred iteration steps. However, we need at least a couple of thousand steps for age distribution and, especially, bad mutations distributions in a genome position to become stationary.

Obviously for smaller birth rates Bb the population is smaller. However, unlike the logistic model for which $N(t \to \infty) = BbN_{max}/(1 + Bb)$, the Penna model shows a critical B_C below which the population is extinct. This is due to the genetic death as result of pumping in more and more mutations M into each new generation, and so $B_C(M)$ with the limit $B_C = 0$ for $M = 0$, the logistic case.

When the number of bad mutations M injected into baby's genome is scanned from $M = 0$ (the logistic model), the upper limit M_C for the extinction strongly depends on fertility: if B is smaller, M_C is also smaller, and for sufficiently large B, $M_C \to \infty$ as the high birth rate gives a chance to new population (however rich in bad mutations) to reproduce before it dies out soon on reaching the threshold T. This is so if the minimum reproduction age R is sufficiently small to allow for the reproduction.

The upper age limit E for reproduction influence the stable population N and larger E makes bigger N, as it may be expected. This time, however, we do not observe any critical behaviour when we scan E.

Acknowledgements. The authors are grateful to Prof. A.Z. Maksymowicz for fruitful discussion. Simulations were carried out in ACK–CYFRONET–AGH. The machine time on SGI 2800 is financed by the Ministry of Scientific Research and Information Technology in Poland.

References

1. Penna, T.J.P.: J. Stat. Phys. **78** (1995) 1629.

Velocity Field Modelling for Pollutant Plume Using 3-D Adaptive Finite Element Method*

Gustavo Montero, Rafael Montenegro, José María Escobar,
Eduardo Rodríguez, and José María González-Yuste

University Institute of Intelligent Systems and Numerical Applications in Engineering,
Univ. of Las Palmas de Gran Canaria, Edif. Central Parque Científico-Tecnológico,
Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria, Spain
{rafa,gustavo}@dma.ulpgc.es, jescobar@dsc.ulpgc.es,
barrera@dma.ulpgc.es, jmgonzalezy@pas.ulpgc.es

Abstract. Air pollution models usually start from the computation of the velocity field of a fluid. In this paper, we present a model for computing that field based on the contribution of the observed wind flow and the vertical buoyancy or momentum plume rise defined by a Gaussian plume model. This initial velocity field is adjusted to verify incompressibility and impermeability conditions by using a mass consistent model. In this environmental modelling that occur in a three-dimensional domain defined over complex terrain, a mesh generator capable of adapting itself to the topographical data and to the numerical solution is essential. Here, the unstructured tetrahedral meshes are generated by combining the use of a refinement/derefinement algorithm for two-dimensional domains and a tetrahedral mesh generator based on Delaunay triangulation. Occasionally in this process, low quality or even inverted elements may appear. A simultaneous untangling and smoothing procedure allows to optimise the resulting meshes. Once we have constructed the adapted mesh in accordance with the geometrical characteristics of our domain, we use an adaptive local refinement in the plume trajectory. Finally, this model is applied in a test problem.

1 Introduction

In [1] the authors propose a mesh generator for environmental problems which is applied in this model. The studied domain is limited at the bottom by the terrain and at the top by a horizontal plane. The lateral walls are formed by four vertical planes. Specifically, we propose the construction of a regular triangulation of this upper boundary. Now, the refinement/derefinement algorithm [2] is applied over this regular mesh to define an adaptive node distribution of the layer corresponding to the terrain surface and the functions which represent the chimneys. Once the node distribution is defined on the terrain and the upper boundary, we begin to distribute the nodes located between both layers by using a vertical spacing function. Next, a three-dimensional mesh generator based on

* Supported by MCYT and FEDER. Grant contract: REN2001-0925-C03-02/CLI

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 642–645, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Delaunay triangulation [3] is applied. In a first stage, our 3-D mesh generator constructs meshes with poor quality and, in special cases, for example when node movement is required, inverted elements may appear. In [4] we propose a procedure for mesh untangling and smoothing in the same stage. In addition, along the plume trajectory, we apply a local refinement algorithm [5] based on the 8-subtetrahedron subdivision. This tetrahedral mesh is used in a mass consistent model for wind field adjustment [6]. To obtain the observed wind, horizontal interpolation of the station measures is carried out. Then, a log-linear wind profile is built in the surface layer taking into account the horizontal interpolation, the effect of roughness and the air stability. Above the surface layer, a linear interpolation is carried out using the geostrophic wind.

2 Vertical Velocity Correction along Plume Trajectory

The main idea is to add a vertical velocity along the trajectory of a pollutant plume arising from a chimney to the interpolated wind field, which usually only consider the horizontal components of wind velocities. Gaussian plume models allow to approximate the effective height of a plume z_H and the horizontal distance d_f from the stack to the point where the plume height reaches z_H, depending on the emission characteristics, the wind and the atmospheric stability. Gases rise from the stack if their density is lower than the air density (buoyancy rise) or due to emission velocity (momentum rise). In order to compute the effective height of the plume, we use Briggs' equations (see e.g. [7]). So, using the computed values of z_H and d_f for buoyancy rise, except for stable conditions and calm wind, we propose to adjust the vertical component of the velocity along the trajectory of the plume by a linearly unaccelerated motion. In addition, the horizontal motion from the source to the distance d_f is considered uniformly accelerated. Thus, the time t_f corresponding to the distance d_f is,

$$t_f = \frac{1}{a_d}\left(-|\boldsymbol{v}_0(x_c, y_c, z_c)| + \sqrt{|\boldsymbol{v}_0(x_c, y_c, z_c)|^2 + 2a_d d_f}\right) \tag{1}$$

where a_d is the module of horizontal acceleration (a_{dx}, a_{dy}) in the direction of wind velocity $\boldsymbol{v}_0$ at the centre of the emission surface (x_c, y_c, z_c). The trajectory of the plume, the vertical velocity w_0 and the vertical acceleration a_0, are then given by the following functions of the parameter t,

$$x(t) = x_c + u_0(x_c, y_c, z_c)t + \tfrac{1}{2}a_{dx}t^2 \tag{2}$$

$$y(t) = y_c + v_0(x_c, y_c, z_c)t + \tfrac{1}{2}a_{dy}t^2 \tag{3}$$

$$z(t) = z'_c + w_c t + K_1 t^2 + K_2 t^3 \tag{4}$$

$$w_0(t) = w_c + 2K_1 t + 3K_2 t^2 \tag{5}$$

$$a_0(t) = 2K_1 + 6K_2 t \tag{6}$$

where $K_1 = \frac{-2w_c t_f + 3(z_H - z'_c)}{t_f^2}$, $K_2 = \frac{w_c t_f - 2(z_H - z'_c)}{t_f^3}$, with w_c, the emission velocity of gases and z'_c, the effective height of the stack. The vertical component of

velocity v_0 is modified in any point of the domain Ω located inside a cylinder generated by the circular emission surface of the gases (of diameter D_c) which is moving parallel to the horizontal plane, along the parametric curve given by equations (2), (3) and (4) between $t = 0$ and $t = t_f$. Thus, constant vertical velocities are generated in the cylinder for each horizontal disk.

On the other hand, for momentum rise or buoyancy rise with stable condition and calm wind, the horizontal motion of the plume until reaching the effective height is very small. Thus the trajectory of the gases is nearly vertical. In this case, we consider a uniformly unaccelerated motion. Now $t_f = \frac{2}{w_c}(z_H - z'_c)$ and the acceleration $a_0 = \frac{-w_c}{t_f}$. Thus, the vertical velocity at a point of height z is $w_0(z) = w_c\sqrt{1 - \frac{2(z-z'_c)}{w_c t_f}}$. Here, the vertical component of the velocity is modified inside a standard cylinder of which base is the emission surface of gases in the stack and its height is $z_H - z'_c$.

3 Numerical Experiments

For air pollution modelling of a test power plant located in the region of La Palma Island, we add the chimney geometry to the topographical data and apply the 3-D mesh generator. Consider a chimney with a height of 200 m over the terrain and diameter of 20 m at its top and 40 m at its bottom. Since, the mesh must be able to detect the details of the chimney, a size of elements about 2×2 m in the chimney, starting from the uniform 2-D mesh τ_1 of the rectangular area with a size of elements about 2×2 *km,* means to make ten global refinement steps using Rivara 4-T algorithm [8]. However, we only need five global refinement steps over τ_1 and, after, five local refinement steps of elements inside the chimney.

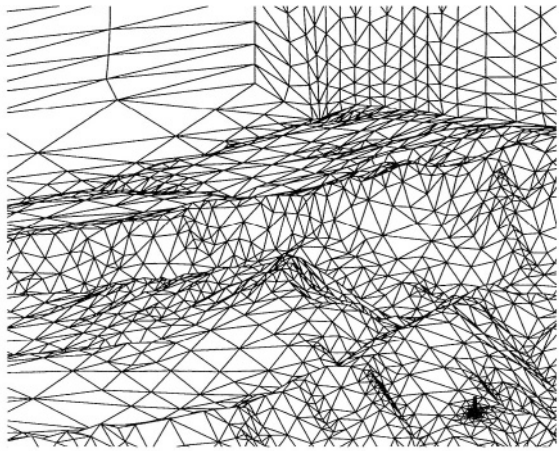


Fig. 1. Detail of a 3-D mesh with a chimney near the right bottom corner

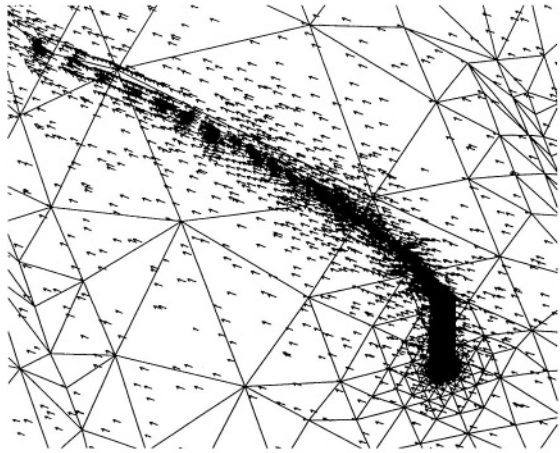


Fig. 2. Detail of the simulated velocity field in the surroundings of the chimney

Here we applied the derefinement algorithm with a parameter $\varepsilon = 40\ m$ considering that those nodes located inside the chimney could not be eliminated. Finally, we have applied six local refinement steps in the plume trajectory to previous resulting 3-D mesh in order to obtain a new mesh with 31555 nodes and 170784 tetrahedra. Figs. 1 and 2 show a detail of the mesh and the adjusted velocity field where the effect of chimney emission is included, respectively.

References

1. Montenegro, R., Montero, G., Escobar, J.M., Rodríguez, E., González-Yuste, J.M.: Tetrahedral Mesh Generation for Environmental Problems over Complex Terrains. Lect. Notes Comp. Sci., **2329** (2002) 335–344
2. Ferragut, L., Montenegro, R., Plaza, A.: Efficient Refinement/Derefinement Algorithm of Nested Meshes to Solve Evolution Problems. Comm. Numer. Meth. Engng. **10** (1994) 403–412
3. Escobar, J.M., Montenegro, R.: Several Aspects of Three-Dimensional Delaunay Triangulation. Adv. Engng. Soft. **27**(1/2) (1996) 27–39
4. Escobar, J.M., Rodríguez, E., Montenegro, R., Montero, G., González-Yuste, J.M.: Simultaneous Untangling and Smoothing of Tetrahedral Meshes. Comp. Meth. Appl. Mech. Engng. **192** (2003) 2775–2787
5. González-Yuste, J.M., Montenegro, R., Escobar, J.M., Montero, G., Rodríguez, E.: An Object Oriented Method for Tetrahedral Mesh Refinement. Adv. Engng. Soft., in press.
6. Montero, G., Montenegro, R., Escobar, J.M.: A 3-D Model for Wind Field Adjustment. J. Wind Engng. & Ind. Aer. **74-76** (1998) 249–261
7. Boubel, R.W., Fox, D.L., Turner, D.B., Stern, A.C.: Fundamentals of Air Pollution. Academic Press, San Diego, (1994)
8. Rivara, M.C.: A Grid Generator Based on 4-Triangles Conforming. Mesh-Refinement Algorithms. Int. J. Num. Meth. Engng. **24** (1987) 1343-1354

Organization of the Mesh Structure

Tomasz Jurczyk and Barbara Głut

AGH University of Science and Technology, Cracow, Poland,
{jurczyk,glut}@uci.agh.edu.pl

Abstract. The subject of this article concerns some aspects of the generation of 3D surface meshes, widely used in many scientific applications. This work contains a description of data structure used in the developed mesh generator and issues related to the management of this structure. The proper organization of this structure coupled with the appropriately selected algorithms, allow to achieve the satisfactory efficiency of the mesh generation process.

1 Introduction

In many areas such as numerical computation or computer graphics there are required surface meshes of three-dimensional objects. The developed mesh generator can create unstructured triangular, quadrangular or mixed meshes, basing on the geometrical description of the given objects. The quadrangular meshes are created from the triangular ones via the indirect techniques[1].

When creating meshes containing very large number of elements, the computational complexity and the memory requirements of the generator can prove to be a very important factor. Several data structures to represent and manipulate unstructured meshes have been published over the years [2] [3]. Such structures usually offer efficient solutions to represent and recover mesh topology and modify its geometric parameters. In this article we describe some aspects regarding the organization of the unstructured meshes, which allows obtaining practically linear time complexity of finite mesh generation.

2 Mesh Structure and Management

Finite element mesh is represented as a structure of linked entities such as vertices, edges or elements (triangles or quadrangles). The selection of the represented mesh entities and theirs interconnection are motivated by the diversity of employed procedures, which can treat the mesh as a graph of vertices and edges, a graph of connected elements, etc.

The process of finite mesh generation consists of a sequence of mesh transformation (mostly local), associated with such operations as insertion of the new node into the triangulation, or with conversion of two triangles into a single quadrilateral element. Depending upon the phase of the generation process, the

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 646–649, 2004.
© Springer-Verlag Berlin Heidelberg 2004

number of various mesh entities may gradually increase, decrease or be practically constant. The typical character of quantity changes for various mesh entities in different phases of the generation process (triangulation, conversion to quadrilaterals, mesh improvement) presents Fig. 1.

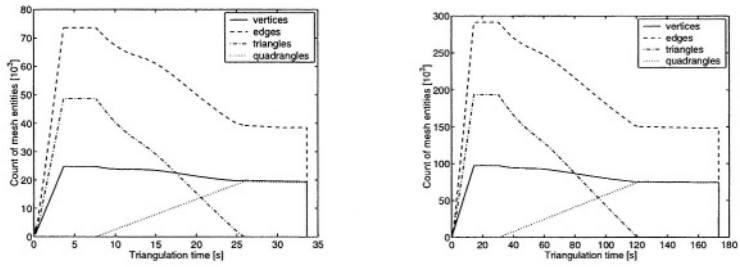


Fig. 1. Number entities during the process of mesh generation (mesh with NT=49k and NT=194k)

In order to optimize algorithms which relies on the removing/inserting of mesh entites (e.g. edges or triangles), there is being used an adequately adjusted procedure of allocation and deallocation of mesh elements. Instead of many single memory operations, a set of entities is allocated at once. Additionally, during the routines of mesh generation, where the quantity of elements has an inclination to grow, removed entities are not deallocated, but they are put aside for further use.

3 Prediction of the Mesh Size

In general case, the final size of the mesh isn't known a priori. However, such information can be very useful for precise establishing of the sizes of the dynamically created structures used for storing the mesh entities, thus increasing the efficiency of memory management – and the whole generation procedure. In the presented approach we try to assess the final size of the finite element mesh during the process of its creation, basing upon the property of the incremental Delaunay triangulation algorithm. During this procedure, all triangular elements are organized in the heap-structure, according to some quality criterion (the root of the heap contains the element with the lowest quality) [4]. The selected quality criterion is responsible for assessing how good the given triangle conforms to the prescribed control space (according both to geometrical shape and element size). It can be evaluated from the following formula:

$$Q_{\triangle} = \frac{S_{\triangle}}{\frac{3\sqrt{3}}{4\pi} S_{\circ}} \tag{1}$$

where $S_{\triangle}$ is the area of the ideal triangle (according to the current metric), and $S_{\circ}$ is the area of the circle circumscribed on the evaluated triangle.

At the subsequent steps of the triangulation algorithm, elements with the worst quality (i.e. too large or badly shaped) are taken from the root of the heap structure. For each taken triangle a refinement procedure is performed, which consists of insertion of a new node in the circumcentre of the triangle and local reconfiguration of the mesh according to the selected Delaunay criterion. The quality criterion (1) is mainly used for determining the order of triangles, selected for the refinement operation. However, this formula can be additionally used for predicting the final number of triangles in the mesh being created. Since this criterion always selects elements with the highest ratio of the current size to the expected size, the process of mesh refinement (according to the prescribed control space) is performed gradually throughout the whole mesh area.

The prediction of the final size of the mesh is calculated from the current number of elements in the mesh and the assessed number of triangles, which should be created instead of some average triangle from the current mesh:

$$NT_{\text{final}} = NT_{current}\frac{Q^*}{Q_\triangle} \tag{2}$$

where NT_{final} is the predicted number of triangles, $NT_{current}$ is the current number of triangles in the mesh, Q^* is the quality coefficient for the selected (average) triangle, and $Q_\triangle$ is the quality coefficient of the ideal triangle.

One of the most important factors influencing the accuracy of the prediction of the final number of elements in the mesh is the selection of triangle (or triangles), for which the quality criterion is inspected and used in the formula (2). Because of the efficiency aspects, the number of inspected triangles can't be too large – the sampling should be restricted to one or few elements.

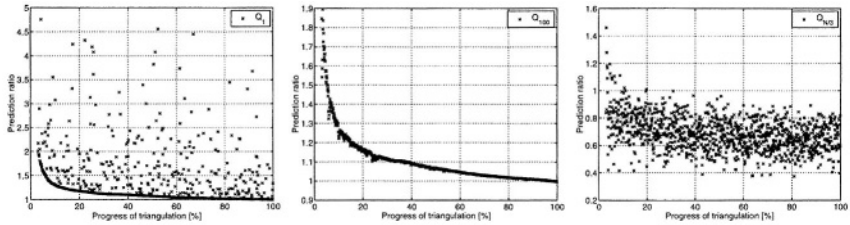


Fig. 2. Accuracy of prediction (ratio of assessed final number of elements to the real value) against the triangulation progress, for cases Q_1, Q_{100}, $Q_{\frac{N}{3}}$

An influence of the selection of the sampled triangle on the prediction accuracy was tested for several meshes. Figure 2 presents the prediction accuracy, evaluated as the ratio of the assessed final number of triangles to the real mesh size after the triangulation process. The best results of prediction were obtained using triangle near the root of the heap (the example triangle Q_{100} is taken from the sixth level of the heap), but not too close. Oscillations for the elements

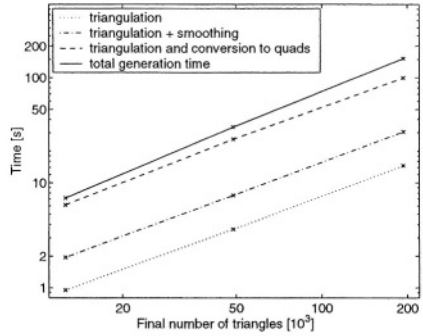


Fig. 3. Computational complexity for meshes with the final number of triangles: NT=12580, NT=48796, and NT=193672

Q_1 (taken from the root of the heap) and $Q_{\frac{N}{3}}$ follow from the variation of the quality of elements at these places.

4 Computational Complexity

The presented structure of the mesh coupled with the appropriately selected algorithms, allow to achieve practically linear computational complexity of the mesh generation. Figure 3 presents the time of generation of three meshes with sizes from 12.5 to 200 thousand of triangles.

Acknowledgment. The partial support of the Polish Committee for Scientific Research (KBN) Grant No. 4T11F00124 is gratefully acknowledged.

References

1. T. Jurczyk and B. Głut. Generation of Good Quality Quadrilateral Elements for Anisotropic Surface Meshes. In N.-E. Wiberg and P. Diez, editors, Proc. of the ADMOS2003 Conference on Adaptive Modeling and Simulation, Göteborg, Sweden, 29 September - 1 October 2003. Chalmers University of Technology, Sweden.
2. M.W. Beall and M.S. Shephard. A general topology-based mesh data structure. International Journal for Numerical Methods in Engineering, John Wiley & Sons, Ltd., 40:1573–1596, 1997.
3. R.V. Garimella. Mesh data structure selection for mesh generation and FEA applications. International Journal for Numerical Methods in Engineering, John Wiley & Sons, Ltd., 55(4):451–478, 2002.
4. T. Jurczyk and B. Głut. Generation of triangular meshes for complex domains. Computer Science, Annual of University of Mining and Metallurgy, Cracow, 3:71-93, 2001.

Kernel Maximum Likelihood Hebbian Learning

Jos Koetsier[1], Emilio Corchado[2], Donald MacDonald[1],
Juan Corchado[3], and Colin Fyfe[1]

[1] Applied Computation Intelligence Research Unit, University of Paisley, Scotland.
{koet-ci0,macd-ci0, fyfe-ci0}@paisley.ac.uk
[2]Departamento de Ingenieria Civil. Universidad de Burgos. Spain.
escorchado@ubu.es
[3]Departamento de Informática y Automática. Universidad de Salamanca. Spain.
corchado@usal.es

Abstract. We present a novel method based on a recently proposed extension to a negative feedback network which uses simple Hebbian learning to self-organise called Maximum Likelihood Hebbian learning [2]. We use the kernel version of the ML algorithm on data from a spectroscopic analysis of a stained glass rose window in a Spanish cathedral. It is hoped that in classifying the origin and date of each segment it will help in the restoration of this and other historical stain glass windows.

1 Introduction

One problem with the analysis of high dimensional data is identifying structure or patterns which exist across dimensional boundaries. By projecting the data onto a different basis of the space, these patterns may become visible. This presents a problem - how does one decide which basis is optimal for the visualisation of the patterns, without foreknowledge of the patterns in the data.

One solution is Principal Component Analysis (PCA), which is a statistical technique aimed at finding the orthogonal basis that maximises the variance of the projection for a given dimensionality of basis. This involves finding the direction which accounts for most of the data's variance, the first principal component; this variance is then filtered out. The next component is the direction of maximum variance from the remaining data and orthogonal to the 1^{st} PCA basis vector.

We [3, 4] have over the last few years investigated a negative feedback implementation of PCA defined by (1- 3). Let us have an N-dimensional input vector, $\mathbf{x}$, and an M-dimensional output vector, $\mathbf{y}$, with $\mathbf{W}_{ij}$ being the weight linking the j^{th} input to the i^{th} output. The learning rate, η, is a small value which will be annealed to zero over the course of training the network. The activation passing from input to output through the weights is described by (1). The activation is then fed back though the weights from the outputs and the error, $\mathbf{e}$, calculated for each input dimension. Finally the weights are updated using simple Hebbian learning.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 650–653, 2004.
© Springer-Verlag Berlin Heidelberg 2004

$$y_i = \sum_{j=1}^{N} W_{ij} x_j, \forall i \tag{1}$$

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \forall j \tag{2}$$

$$\Delta W_{ij} = \eta e_j y_i \tag{3}$$

We have subsequently modified this network to perform clustering with topology preservation [5], to perform Factor Analysis [8, 1] and to perform Exploratory Projection Pursuit (EPP) [7, 6].

2 Maximum Likelihood Hebbian Learning

This paper deals with a recently developed variation of the basic network which also performs Exploratory Projection Pursuit. We show the validity of our method by applying it to data from a spectroscopic analysis of the stained glass rose window in a Spanish cathedral. It is hoped that in classifying the origin and date of each segment it will help in the restoration of this and other historical stain glass windows.

Let us now consider the residual after the feedback to have probability density function.

$$p(\mathbf{e}) = \frac{1}{Z} \exp(-|\mathbf{e}|^p) \cdot \tag{4}$$

Then we can denote a general cost function associated with this network as

$$J = -\log p(\mathbf{e}) = |\mathbf{e}|^p + K \tag{5}$$

where K is a constant. Therefore performing gradient descent on J we have

$$\Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial J}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial W} \approx y(p\,|\mathbf{e}|^{p-1}\, sign(\mathbf{e}))^T \tag{6}$$

We would expect that for leptokurtotic residuals (more kurtotic than a Gaussian distribution), values of $p < 2$ would be appropriate, while platykurtotic residuals (less kurtotic than a Gaussian), values of $p > 2$ would be appropriate.

3 Kernel Maximum Likelihood

The first step in our kernel version of the ML algorithm can be performed by projecting our data onto a set of eigenvectors in feature space and thus we obtain sphered data in feature space. We can subsequently reduce the dimensionality as a further preprocessing step to the ML method. As the transformed data are actually points in feature space, we can simply apply the maximum likelihood method on the transformed data, as though it would be in data space.

Let $\mathbf{z}_k$ be the datapoint $\mathbf{x}_k$ transformed into feature space and projected onto the principal components in feature space. The Kernel Maximum Likelihood (KML) learning rules then become

$$\text{Feedforward:} \quad y_i = \sum_{j=1}^{N} W_{ij} z_{kj}, \forall_i \tag{7}$$

$$\text{Feedback:} \quad e_j = z_{kj} - \sum_{i=1}^{M} W_{ij} y_i \tag{8}$$

$$\text{Weightchange:} \quad \Delta W_{ij} = \eta y_i sign(e_j) |e_j|^{p-1} \tag{9}$$

4 Experiments

The data used to illustrate our method is composed of samples from 76 different sections of the stained glass rose window in a Spanish cathedral. The data contained 450 data vectors obtained from 90 samples each having been analysed 5 times. The data is 1020 dimensions, which after normalisation was reduced to 390 dimensions.

The data was analysed by [9] and during their analysis of the data they found that it contained three different clusters. These clusters were identified to be separate by their chemical composition, one belonging to the 16th century, and the other two classes which were from the 13th century. These clusters were classified by the proportion of sodium and potassium in the glass.

We visualise the data using our KML method on this data so we can look for clusters by eye. As we are interested in projection exhibiting clusters, we wish to extract sub-gaussian signals and therefore we use a value for $p = 0$. We compare our results to those obtained by normal ML.

(a) Kernel Maximum Likelihood (b) Maximum Likelihood

Fig. 1. Projection of the glass data using the KML (left) and the ML methods (right). The figures show the result of projection onto four directions found. It can be seen that the Kernel Maximum Likelihood method results in greater separation of the clusters.

4.1 Results

In Fig. 1., we show the projections obtained by ML (Right) and KML (Left). The first row of the figures show scatterplots of projections onto the first direction set out against projections onto first until the fourth directions. The second row shows projections onto the second direction set out against projections onto first until the fourth directions and so forth. In both figures we can identify clusters, but the projections obtained by the KML method generally result in a better separation of the clusters. The non-linear extension to the standard Maximum Likelihood method has given the method greater flexibility which allows us to get a different view on the data that certainly in this case yields much better clustering.

5 Conclusion

In this paper we have introduced a novel extension to the Maximum Likelihood Hebbian learning algorithm that allows the method to use non linear projections. The data is actively being used in research projects to help in future restoration of stain glass windows. Using our new clustering methods we visualise the data to identify relationships between the different chemical properties of the glass samples.
We have shown that our new non linear extension of Maximum Likelihood has found clusters in this data set that could not be identified by ML. This new method therefore allows us to better analyse the data and give us different visualisations of our data set.

References

1. Charles, D. and Fyfe, C. Modelling Multiple Cause Structure using Rectification constraints. Network: Computation in Neural Systems, 9:167-182, May 1998.
2. Corchado, E. and Fyfe, C. Maximum Likelihood Hebbian Rules. In Tenth European Symposium on Artificial Neural Networks, ESANN2002, pages 143-148, 2002.
3. Fyfe, C. PCA Properties of Interneurons. In From Neurobiology to Real World Computing, ICANN 93, pages 183-188, 1993.
4. Fyfe, C. Introducing Asymmetry into Interneuron learning. Neural Computation, 7(6):1167-1181, 1995.
5. Fyfe, C. Radial Feature Mapping. In International Conference on Artificial Neural Networks, ICANN95, Oct. 1995.
6. Fyfe, C. A Comparative Study of Two Neural Methods of Exploratory Projection Pursuit. Neural Networks, 10(2):257-262, 1997.
7. Fyfe, C. and Baddeley, R. Non-linear Data Dtructure Extraction using Simple Hebbian Networks. Biological Cybernetics, 72(6):533{541, 1995.
8. Fyfe, C. and Charles, D. Using Noise to Form a Minimal Overcomplete Basis. In Seventh International Conference on Artificial Neural Networks, ICANN99, 1999.
9. Lopez-Gejo, J. Colina, A. Lopez-Palacios, J. and Bravo, P. Principal Components Analysis in the Classification of Medieval Glasses by Scanning Electron Microscopy Coupled with Energy Dispersive X-ray Analysis. (submitted), 2003.

Discovery of Chemical Transformations with the Use of Machine Learning

Grzegorz Fic and Grazyna Nowak

Rzeszow University of Technology
6 Powstancow Warszawy Ave., 35-041 Rzeszow, Poland
{gfic, gnowak}@prz.rzeszow.pl

Abstract. Research activities in the computer reaction modeling and synthesis design have led to the development of a number of computer systems that are successively used as a research tools to study the organic synthesis problems. One of such systems is the CSB (Chemical Sense Builder), which combines four different logic- and knowledge-based models to generate possible structural modes of chemical conversions. In these paper we briefly discuss the architecture and specially the learning methods implemented in the CSB. They exploit different sources of knowledge and chemical reaction data for the classification, generalization, and derivation of rules in a form of the graph transformation schemes. Any transformation is general description of a collection of similar reactions from the reaction data base. Application of these learned rules in the course of reaction simulation enables us to predict – for a given set of reacting molecules – all possible reaction courses, having responses in real chemistry.

1 Introduction

A comprehensive study of fundamentals of formal-logic approaches (mathematical models) implemented in various systems for modeling chemical conversions has pointed out some problems dealing with the designing and exploitation of such system category. The most important is related to a compromise between two opposing interests: generating all the conceivable solutions (with wrong, trivial, and otherwise some valuable and novel one) or generating reliable solutions (avoiding combinatorial explosion). Thus, the solution space produced according to a given logic-based or knowledge based model for transformation of chemical structures may be either complete or reliable. The growing of the reliability implies reduction of completeness. Methods attempting to achieve a satisfactory compromise between completeness and reliability are based on:

- developing multimodel systems (CSB [1])
- developing a learning systems (SYNCHEM [2], CSB)
- using multilevel hierarchy of knowledge (CSB, in testing).

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 654–657, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The above three methods are implemented in recent version of the CSB system. In result, the user has the possibility to select the most suitable model of simulation, matched to the synthetic problem being solved.

2 The CSB Multimodel System for Simulation of Chemical Transformations

In the current version of the CSB system, the following four models are applied for the generation of chemical reactions:
- *Ugi-Dugundji (U-D) model,* based on the *matrix model of constitutional chemistry* [3], which enables the generating complete but not reliable solution space
- *common-sense model,* combines the fundamentals of U-D model and approximate reasoning of reaction centers [4] enabling to avoid a combinatorial explosion and to reduce unreal chemical transformations
- *similarity model,* allows the simulating conversions that are similar to reactions in the training set (a source for supervised learning) [1]
- *purely empirical model,* reproduces known chemical knowledge from the Real Reaction Library.

3 The CSB Learning System for Searching, Classification and Generation of Rules from Data

Two methods of supervised learning are implemented in the CSB system:
- monitoring and reproducing the courses for these simulations that have lead to *good solutions,*
- comparative analyzing based on the concept of hierarchical levels of knowledge.

First of these methods was implemented in the similarity model of the CSB system, the second is tested yet, and will be involved in the architecture of the CSB as novel simulation model – i.e., *model of the reaction circle.*

3.1 Monitoring of the Simulation Course

In the learning process, the following information sources may be utilized to derive and generalize of new rules for the Similar Reaction Library (SRL):
- reaction entered by chemist, or
- reaction generated by the common-sense module (in the prediction process), and verified by searching the Real Reaction Library (RRL).

The essence of this method is an attempt to imitate the course of a given real reaction, using the common-sense model and the actual chemical knowledge stored in the Real Reaction Library and REACT knowledge bases. When the simulation process of the source reaction has been successfully finalized (i.e., entered graphs of substrates have been transformed into the graph of product, which is identical with a product of some real reaction), then the Knowledge Generator (using the track of this simulation

stored in the memory), induces the transformation scheme of the shape: cause ⇒ consequence. It is a generalized description representing the source reaction and a collection of similar reactions (classifier). It contains the following information:

on the side of a cause: list of conditions needed for generating a real reaction as well as similar ones. It comprises: identifiers of active subgraphs that have to be present in graphs of substrates;

on the side of a consequence: transformation vector describing the course of the conversion of active subgraphs of substrates into product candidates. It contains: reaction model (number of reacting molecules); list of identifiers of edges (in active subgraphs) that are broken/made during the simulated transformation; a scheme (from Library of Reaction Generator [5]) allowing the formation of new edges in molecular graphs (products).

Each chemical reaction may be a source for generating at least one transformation scheme. Set of learned transformations kept in the Similar Reaction Library, may then be applied to generate (for a given ensemble of molecular graphs representing substrates) the reaction being the source for the learning process as well as all similar reactions.

The learning process is conduced here on expert-prepared rules for the knowledge base REACT. Uncompleted and subjective nature of this knowledge (reflecting the opinions of the individual expert) may cause some limitations of this learning method.

3.2 Hierarchical Levels of Knowledge – The Model of Reaction Circle

In the learning method discussed above, the cause-consequence correlation is created as result of a comparative analysis of molecular graph for substrates (cause) with molecular graphs of products (consequence). This operation involves the following six stages:

Input data: reaction graph = {molecular graphs of starting materials, molecular graphs of products}
step_1: topological analysis of molecular graphs
step_2: atom-atom mapping (recognizing in the graph of products the nodes which are equivalent to nodes in graph of substrates and creating the reaction map)
step_3: recognition of the molecular subgraph of reaction center {1st level of knowledge}
result: **transformation describing the 1st level of knowledge**
step_4: L=2 {2nd level of knowledge}
step_5: expansion of the L-1 level of knowledge (structural environment) by next bound length of the reaction center
result: **transformation of n-th level of knowledge**
step_6: checking whether all levels were tested
if yes: **STOP**
if not: L = L+1 and realization of the **step_5.**

The learning method considered here, in contrast to the first one, enables to avoid the influence of the subjective knowledge and intuition of the expert on results of computer simulation of chemical conversions. The main advantage is the capability to analyze and generalize the data from the Real Reaction Library, leading to derivation of rules for Library of Similar Reactions, arranged into multi-level, hierarchical structure. Application of various knowledge formalisms (reflecting various levels of generality/specificity) for generating chemical conversions, brought out the possibility of creating various solution spaces, depending on the goal of computer simulation. This method is capable to discover new chemical reactions (here the application of the most general knowledge, i.e., on the lowest level, is preferred), from the other hand, it is useful for the classification of chemical reactions (assuming as a criterion the highest level of knowledge). Additional possibility is the bi-directional generation of chemical transformation, i.e., forward for the simulation of organic reactions, and backward for the design of chemical syntheses.

4 Conclusion

The methodology implemented in the CSB system, integrates the concept of multi-model simulation with the knowledge processing and learning tools, supporting the process of the generation of chemical conversions. Recent advances in the development of the CSB provide new machine learning capabilities. Among them is the ability to learn different type of rules from various knowledge sources, and automatic application of learned or acquired rules in the reaction prediction process. The CSB can be applied as a tool to wide variety of problems in different disciplines of chemistry. Some of possible fields of CSB-applications are: the prediction of the reaction courses proceeding in different environment (natural atmosphere, soil, water, living cells, etc.), research on metabolic pathways of various drugs, research in drug-drug interactions, the simulation and prediction of reactions in the natural environment (as for example, the degradation reaction and their products, the study of the fate of chemicals in agriculture).

References

1. Fic, G., Nowak, G.: Implementation of similarity model in the CSB system for chemical reaction prediction. Comput. Chem. 25 (2001) 177-186.
2. Gelernter, H., Rose, J.R., Chen, C.: Building and refining a knowledge base for synthetic organic chemistry via the methodology of inductive and deductive machine learning. J. Chem. Inf. Comput. Sci. 30 (1990) 492-504.
3. Dugundji, J., Ugi, I.: An Algebraic Model of Constitutional Chemistry as a Basis for Chemical Computer Programs. Top. Curr. Chem. 39 (1973) 19-64.
4. Nowak, G., Fic, G.: Implementation of Commonsense Reasoning in Organic Reaction Prediction System CSB. Comput. Chem. 22 (1998) 147-152.
5. Fic, G., Nowak, G.: Extended Concept of Reaction Generators in an Organic Reaction Prediction System CSB. Comput. Chem. 22 (1998) 141-146.

Extraction of Document Descriptive Terms with a Linguistic-Based Machine Learning Approach*

Javier Fernández, Elena Montañés, Irene Díaz, José Ranilla, and Elías F. Combarro**

Artificial Intelligence Center – University of Oviedo (Spain)
ir@aic.uniovi.es

Abstract. In this paper we present a method for extracting relevant words from a document taking into account linguistic information. Form a set of words manually labelled as relevant or not and with the aid of a Machine Learning algorithm, we build a classifier which is able to decide which words from a unseen document must be regarded as relevant. This system is compared with some classical methods (based just on statistical information).

1 Introduction and Previous Work

One of the main processes in Information Retrieval (IR) and Text Categorisation (TC) is that of transforming documents into a form suitable for automatic processing. The most used representation consists of identifying each document with a vector whose components measure the importance of the different words. This representation is known as *bag of words* (see [1]). It is interesting to study methods for the removal of non-informative features. This paper explores the use of linguistic knowledge in this process.

A widely adopted approach to feature reduction consists of ordering the words according to a measure of their relevance and selecting those with the highest score. For instance, one can count the total number of appearances (*tf*) of each word in the corpus and then keep only the most frequent words. The dispersion of the word in the corpus can also be considered with *tfxidf* defined by $tfxidf = tf \log(\frac{N}{df})$ where N is the total number of documents and *df* is the number of different documents in which the word appears. If the corpus is a set of categorised documents the distribution of a word over the different categories can be considered. This is the case of *information gain* (IG) [2].

All these approaches have in common the use of mainly statistical information. However, including the syntactic and grammatical function of the words and the location in which they appear, may help to successfully determine the importance of the words in the documents. Some authors [3,4] do use some lexical information, but usually only a quite restricted one.

* This research has been supported under MCyT and Feder grant TIC2001-3579.
** The author acknowledges the support of FICYT grant PR-01-GE-15.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 658–661, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 The System and the Experiments

Our system builds a binary classifier to decide whether a word can be regarded as a descriptive term for a certain document or not. This inductive process starts from a set of attributes of a set of words that have been previously classified either as descriptive or not. In this work we use two Machine Learning (ML) systems to perform this task: Arni-rules [5] and C4.5-rules [6].

Each word is represented by a set of attributes which provide linguistic information of such word in the context of each document under study. Linguistic information is divided into **POS-tagging information,** representing the lexical functions that a word can play in a document (obtained with Eric Brill's tagger [7]); and **term location,** including information about the position of the word in the document (the word appears in the title, at the beginning of the document,...). The values of these attributes can be boolean or numerical. The influence of word frequency (*tf*) is also taken into account.

The attributes of a word in the different documents need to be aggregated. We have adopted two different methods to perform this task. The first one consists of adding the values of the attributes. We will call *tf-agg* this type of aggregation. The second one involves performing the *tf-agg* and then multiplying it by $\log(\frac{N}{df})$. This kind of aggregation will be called *tfxidf-agg.*

The experiments have been conducted with two corpora: Reuters-21578, which contains short news related to economy published by Reuters during 1987; and Ohsumed.91, which covers all references from 270 medical journals of 1991. We have used all the documents of the categories *Alum* and *Cocoa* from Reuters-21578 and 60 documents of the category *C06* from Ohsumed.91.

3 Results

To quantify the performance we use precision and recall. Precision (P) is the percentage of words classified as descriptive which really are, and recall (R) is the percentage of descriptive terms recognized by the system. They are combined by means of F_1 (cf. [8]) defined by $F_1 = \frac{1}{\frac{1}{2P}+\frac{1}{2R}}$. The experiments are performed using stratified cross validation with 10 folds and 5 repetitions.

In Tables 1, 2 and 3 we present the averaged F_1 obtained in the experiments. Results both without *stemming* (WS) and with *stemming* (S) according to Porter [9] are reported. We denote POS-tagging attributes by "L", boolean term location attributes by "B" and numeric term location attributes by "N".

The general trends in the three categories and with both systems are similar. First of all, it is clear, with few exceptions, that the use of *stemming* results in an improvement. On the other hand, the *tfxidf-agg* shows, in general, better results than the tf one. The combination of POS-tagging and term location seems to improve the performance in many cases, while the inclusion of tf has little impact (in many cases the result is exactly the same). In addition, although the results of *Cocoa, Alum* and *C06* follow the same trend, the F_1 of the first two is remarkably lower, possibly due to an overfitting problem.

Table 1. Resulting F_1 (in %) in Alum

Attributes	Arni-rules				C4.5-rules			
	tf-agg.		*tfxidf*-agg.		*tf*-agg.		*tfxidf*-agg.	
	WS	S	WS	S	WS	S	WS	S
L	40.16	42.40	43.40	44.83	44.87	34.27	43.83	50.34
L+Tf	40.16	45.54	43.40	47.51	44.87	32.49	43.83	49.54
L+B	34.72	41.33	35.33	39.94	41.22	46.01	38.36	51.23
L+B+Tf	34.72	42.65	35.33	40.87	41.61	44.29	39.19	49.85
L+N	39.20	42.74	40.55	38.70	39.85	40.82	37.10	45.94
L+N+Tf	39.20	43.54	40.55	40.08	39.78	38.91	37.11	44.99

Table 2. Resulting F_1 (in %) in Cocoa

Attributes	Arni-rules				C4.5-rules			
	tf-agg.		*tfxidf*-agg.		*tf*-agg.		*tfxidf*-agg.	
	WS	S	WS	S	WS	S	WS	S
L	16.67	23.82	18.00	21.14	13.10	15.33	24.94	19.17
L+Tf	13.79	23.11	14.97	22.75	13.31	16.40	24.46	16.20
L+B	20.93	23.36	27.74	27.46	26.42	27.86	30.14	28.57
L+B+Tf	20.69	22.48	26.15	28.39	26.27	19.07	28.72	27.27
L+N	21.32	27.60	25.25	26.98	11.63	23.81	22.42	22.38
L+N+Tf	21.88	26.65	25.71	26.60	12.77	17.96	21.92	19.81

The proposed method is compared with 3 measures of word relevance: tf, *tfxidf* and *IG*. Using them, we rank the words and calculate the resulting F_1 when the words with highest values are considered as descriptive terms. The results are presented in Table 4, where the filtering level indicates the percentage of words that are supposed to be non-informative. The filtering level obtained with our system ranges from 75% to 90% in *Alum,* from 90% to 98% in *Cocoa* and from 70% to 88% in *C06,* so only the corresponding levels are presented.

If we use both POS-tagging and term location data, *stemming* is performed, the aggregation is *tfxidf-agg* and the ML system is C4.5-rules, then the performance of the system is better than that of the traditional measures at all the comparable filtering levels. Namely, we obtain $F_1 = 51.31\%$ (filtering level of 75.39%) in *Alum,* $F_1 = 28.57\%$ (filtering level of 93.01%) in *Cocoa* and $F_1 = 56.97\%$ (filtering level of 77.47%) in *C06.* For other choices of parameters the results are also better than those of the traditional measures for most filtering levels.

4 Concluding Remarks and Future Work

We have presented a system for the extraction of informative words from a document which takes into account POS-tagging, term location and statistical information. When comparing the results obtained by traditional filtering measures with those got with a good selection of parameters in our system we find that our method performs better. Additionally, with this system it is not necessary to select the filtering level since it is automatically obtained.

However, some overfitting problems must be solved. Hence it is necessary to study which is the most appropriate number of examples. As an extension of this

Table 3. Resulting F_1 (in %) in C06

Attributes	*tf*-agg.		*tfxidf*-agg.		*tf*-agg.		*tfxidf*-agg.	
	Arni-rules				C4.5-rules			
	WS	S	WS	S	WS	S	WS	S
L	48.57	48.05	41.97	54.35	34.64	47.83	51.65	59.96
L+Tf	48.81	48.91	42.31	53.99	34.64	51.38	51.62	59.22
L+B	48.57	53.46	50.10	54.57	49.27	55.47	49.45	56.97
L+B+Tf	48.57	53.60	49.99	55.01	49.02	54.29	49.84	57.49
L+N	54.08	58.32	53.60	59.30	57.14	60.57	57.34	62.19
L+N+Tf	54.60	58.73	54.16	58.76	57.14	61.14	57.24	62.62

Table 4. F_1 (in %) of tf, $tfxidf$ and IG

Category	Filtering Level	tf	tfxidf	IG
Alum	75%	41.28	44.04	32.99
	80%	39.29	39.29	32.65
	85%	35.53	36.68	32.66
	90%	33.44	33.44	28.20
Cocoa	90%	22.29	24.80	19.10
	95%	21.20	20.60	16.00
C06	70%	40.21	53.12	33.80
	75%	45.78	48.19	30.45
	80%	39.86	44.90	27.85
	85%	32.98	39.63	24.20
	90%	25.93	31.89	18.78

work, we are also interested in studying the impact of considering syntactical information of the words. We would also like to check whether extracting words with the system proposed improves the overall performance of TC.

References

1. Salton, G., McGill, M.J.: An introduction to modern information retrieval. McGraw-Hill (1983)
2. Yang, T., Pedersen, J.P.: A comparative study on feature selection in text categorisation. In: Proceedings of ICML'97, 14th International Conference on Machine Learning. (1997) 412–420
3. Barker, K., Cornacchia, N.: Using noun phrase heads to extract document keyphrases. Lecture Notes in Computer Science **1822** (2000) 40–52
4. Turney, P.D.: Learning algorithms for keyphrase extraction. Information Retrieval **2** (2000) 303–336
5. Ranilla, J., Bahamonde, A.: Fan: Finding accurate inductions. International Journal of Human Computer Studies **56** (2002) 445–474
6. Quinlan, J.R.: Constructing decision tree in c4.5. In: Programs of Machine Learning. Morgan Kaufman (1993) 17–26
7. Brill, E.: A Corpus-Based Approach to Language Learning. PhD thesis, Philadelpha, PA (1993)
8. Sebastiani, F.: Machine learning in automated text categorisation. ACM Computing Survey **34** (2002)
9. Porter, M.F.: An algorithm for suffix stripping. Program (Automated Library and Information Systems) **14** (1980) 130–137

Application of Brain Emotional Learning Based Intelligent Controller (BELBIC) to Active Queue Management

Mahdi Jalili-Kharaajoo

Young Researchers Club, Azad University, Tehran, Iran
mahdijalili@ece.ut.ac.ir

Abstract. In this paper, Brain Emotional Learning Based Intelligent controller (BELBIC) is applied to Active Queue management (AQM). This type of controller is insensitive to noise and variance of the parameters, thus it is suitable to time varying network systems. Simulation results show the robust performance of BELBIC against the disturbance.

1 Introduction

Active Queue Management (AQM), as one class of packet dropping/marking mechanism in the router queue, has been recently proposed to support the end-to-end congestion control in the Internet [1-3]. AQM highlights the tradeoff between delay and throughput. By keeping the average queue size small, AQM will have the ability to provide greater capacity to accommodate nature-occurring burst without dropping packets, at the same time, reduce the delays seen by flow, this is very particularly important for real-time interactive applications.

Modeling emotions has attracted much attention in recent years, both in cognitive psychology and design of artificial systems [4]. Far from being a negative factor in decision making, emotions have shown to be a strong faculty for making fast satisficing decisions. In this paper, we will adapt a computational model based on the limbic system in the mammalian brain for control engineering applications. We applied the proposed controller (termed BELBIC) [5] for AQM. Our results demonstrate excellent control action, disturbance handling and system parameter robustness for BELBIC.

2 TCP Flow Control Model

In [6,7], a nonlinear dynamic model for TCP flow control has been developed based on fluid-flow theory. The state space description of this model can be as follows [1]

$$\begin{cases} \dfrac{dx_1}{dt} = x_2 \\ \dfrac{dx_2}{dt} = -a_1(t)x_1 - a_2(t)x_2 - b(t) + F(t) \end{cases} \tag{1}$$

$$a_{1\min} \le a_1 \le a_{1\max}, a_{2\min} \le a_2 \le a_{2\max}, 0 < b_{\min} \le b \le b_{\max} \tag{2}$$

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 662–665, 2004.
© Springer-Verlag Berlin Heidelberg 2004

where

$$a_1(t)=\frac{1}{T_1(t)T_2(t)}, a_2(t)=\frac{T_1(t)+T_2(t)}{T_1(t)T_2(t)}, b(t)=\frac{K(t)}{T_1(t)T_2(t)} \tag{3}$$
$$F(t)=\frac{d^2}{dt^2}q_o+\frac{T_1(t)+T_2(t)}{T_1(t)T_2(t)}\frac{d}{dt}q_o+\frac{1}{T_1(t)T_2(t)}q_o$$

For the purpose of the design of intelligent controller, the varying scope of parameters in TCP/AQM system is assumed as following

$$N(t):1-300, T_p=0.02\text{sec}, q_o:0-300 packets, C(t):1250-7500 packet/\text{sec}$$

Therefore,

$$a_{2\min}=3.8501 a_{2\max}=1250 a_{1\min}=0.015 a_{1\max}=60000 b_{\min}=26042, b_{\max}=2812500.$$

3 Brain Emotional Learning Based Intelligent Controller (BELBIC)

The model of BELBIC is illustrated in Fig. 1 [5]. BELBIC is essentially an action generation mechanism based on sensory inputs and emotional cues. In general, these can be vector valued, although in the benchmark discussed in this paper for the sake of illustration, one sensory input and one emotional signal (stress) have been considered. The emotional learning occurs mainly in amygdala. The learning rule of amygdala is

$$\Delta G_a = k_1.\max(0, EC-A) \tag{4}$$

where G_a is the gain in amygdala connection, k_1 is the learning step in amygdala and EC and A are the values of emotional cue function and amygdala output at each time.

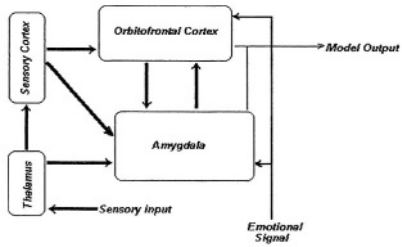


Fig. 1. The abstract structure of BELBIC

Similarly, the learning rule in orbitofrontal cortex is

$$\Delta G_o = k_2.(MO-EC) \tag{5}$$

where G_o is the gain in orbitofrontal connection, k_2 is the learning step in orbitofrontal cortex and MO is the output of the whole model, where it can be calculated as

$$MO = A-O \tag{6}$$

in which, O represents the output of orbitofrontal cortex. In fact, by receiving the sensory input S, the model calculates the internal signals of amygdala and orbitofrontal cortex by the relations

$$A = G_a.S \ , \ O = G_o.S \tag{7}$$

Since amygdala does not have the capability to unlearn any emotional response that it ever learned, inhibition of any inappropriate response is the duty of orbitofrontal cortex. Controllers based on emotional learning have shown very good robustness and uncertainty handling properties [4,5], while being simple and easily implementable.

The structure of the control circuit we implemented in our study is illustrated in Fig. 2. The functions we used in emotional cue and sensory input blocks are

$$EC = W_1.e + W_2.CO \ , \ SI = W_3.PO + W_4.PO \tag{8}$$

where EC, CO, SI and PO are emotional cue, controller output, sensory input and plant output and the W_1 through W_4 are the gains must tuned for designing a satisfactory controller.

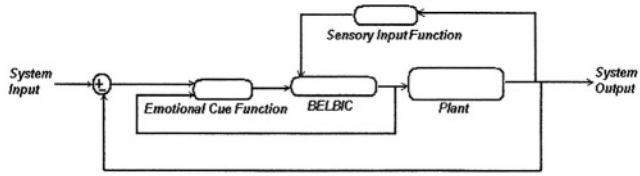


Fig. 2. Control system configuration using

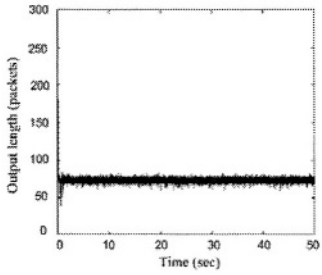


Fig. 3. Queue evaluation (BELBIC)

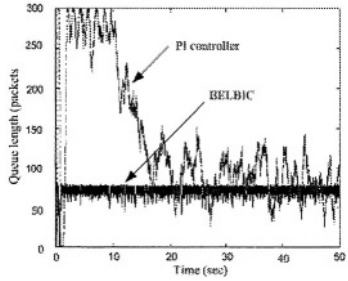


Fig. 4. Queue evaluation (FTP+HTTP)

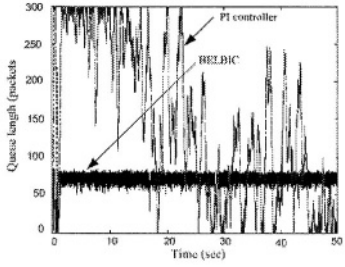


Fig. 5. Queue evaluation (FTP+UDP)

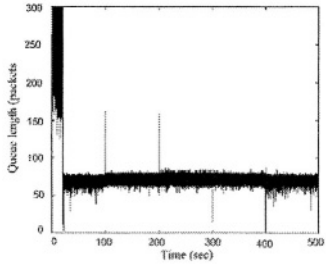


Fig. 6. Queue evaluation using BELBIC

4 Simulation Results

The network topology used for simulation is a FTP+UDP+HTTP network, which has been described in details in [1]. We introduced short-lived HTTP flows and non-

responsive UDP services into the router in order to generate a more realistic scenario, because it is very important for a perfect AQM scheme to achieve full bandwidth utilization in the presence of noise and disturbance introduced by these flows [1].

In the first study, we will use the most general FTP network configuration to testify whether the BELBIC controller can reach the goals of AQM, and freely control the queue length to stabilize at the arbitrary expected value. Let the expected queue length equal to 75 packets, the instantaneous queue length is depicted in Fig. 3. After a very short regulating process, the queue settles down its stable operating point. RED algorithm is unable to accurately control the queue length to the desired value. The queue length varies with network loads.

For FTP+HTTP configuration the evaluation of queue size is shown in Fig. 4 using BELBIC and classical PI controller. As it can be seen, BELBIC has better performance than that of PI one. This comparison is done for FTP+UDP in Fig. 5. Obviously, PI controller is very sensitive to this disturbance, while BELBIC operates in a relatively stable state. The queue fluctuation increases with introducing the UDP flows, but the variance is too much smaller comparing with PI one.

Finally, we evaluate the integrated performance of BELBIC using one relatively real scenario, i.e., the number of active flows is changeable, which has 270 FTP flows, 400 HTTP connections and 30 UDP flows (Fig. 6).

5 Conclusion

In this paper, BELBIC wad applied to active queue management. BELBIC is insensitive to system dynamic parameters and is capable of being against disturbance and noise, which is very suitable for the mutable network environment. BELBIC was very responsive, stable and robust, especially for the small reference queue system, but its performance was inferior when active TCP sessions were relatively small.

References

1. Jalili-Kharaajoo, M., Application of robust fuzzy second order sliding mode control to active queue management, *Lecture Notes in Computer Science,* 2957, pp.109-119, 2004.
2. Barden, B. et al., Recommendation on queue management and congestion avoidance in the internet, *REC2309,* April 1998.
3. Firoiu, V. and Borden, M., A study of active queue management for congestion control, *Proc. INFOCOM,* March 2000.
4. J. Moren, C. Balkenius, A Computational Model of Emotional Learning in The Amygdala, *From Animals to Animals: Proc. 6th Int. conf. Simulation of adaptive behavior,* Cambridge, MA, 2000.
5. Lucas, C. and Shahmirzadi, D., Introducing Brain Emotional Learning Based Intelligent Controller (BELBIC), *to appear in Intelligent Automation and Soft Computing,* 2004.
6. Misra, V., Gong, W.B. and Towsley, D., Fluid-based analysis of network of AQM routers supporting TCP flows with an application to RED, *Proc. ACM/SIGCOMM,* 2000.
7. Hollot, C., Misra, V., Towsley, D. and Gong, W.B., On designing improved controllers for AQM routers supporting TCP flows, *Proc. INFOCOM,* 2001.

A Hybrid Algorithm Based on PSO and SA and Its Application for Two-Dimensional Non-guillotine Cutting Stock Problem*

J.Q. Jiang[1,2], Y.C. Liang[1, 3**], X.H. Shi[1], and H.P. Lee[3]

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China
ycliang@public.cc.jl.cn
[2] College of Mathematics and Computer Science, National University of Inner Mongolia, Tongliao 028043, China
[3] Institute of High Performance Computing, Singapore 117528, Singapore

Abstract. In this paper we present a hybrid algorithm based on Particle Swarm Optimization (PSO) and Simulated Annealing (SA) approaches and apply it to two-dimensional non-guillotine cutting stock problem. The probability of trapping at the local optimum during the searching process can be reduced using the hybrid algorithm. Meanwhile, we propose a converting approach which is similar to the Bottom Left (BL) algorithm to map the cutting pattern to the actual layout. Finally, we implement the proposed algorithm on several test problems. The simulated results show that the performance of the hybrid algorithm is better than that of the standard PSO.

1 Introduction

The two-dimensional cutting stock problem can be stated as cutting small rectangular pieces of predetermined sizes from a large rectangular plate (the stock plate). Non-guillotine cut means that the cuts need not go from one edge of a rectangle to the opposite edge in a straight line. The aim is to minimize the unused area. Gilmore and Gomory [1, 2] used linear programming to solve such kind of a problem exactly. But because of the complexity of this problem, the exact algorithm only fits the case in which fewer pieces are being cut. These algorithms would fail if there are more pieces to be cut. A problem of cutting more than 20 pieces would cause some difficulty. Recently, with the extended application of various heuristic searching algorithms, Evolutionary Algorithms (EA) and Simulated Annealing (SA) approach have been applied to this kind of problems [3-5]. In this paper, we will consider applying Particle Swarm Optimization (PSO) to cutting stock problem. PSO was introduced by Kennedy and Eberhart [6]. In the process of searching for the solution, we add SA technique to PSO to reduce the probability of trapping at a local optimum.

* Supported by the science-technology development project of Jilin Province of China (Grant No. 20030520), the Key Science-Technology Project of the National Education Ministry of China (Grant No. 02090) and the doctoral funds of the National Education Ministry of China.
** To whom correspondence should be addressed. E-mail: ycliang@public.cc.jl.cn

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 666–669, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 A Hybrid Algorithm Based on PSO and SA

PSO is an evolutionary computational model based on swarm intelligence. Suppose that the search space is D-dimensional and m particles form the colony. The ith particle represents a D-dimensional vector X_i $(i=1, 2, ..., m)$. It means that the ith particle locates at $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ $(i=1, 2, ..., m)$ in the search space. The position of each particle is a potential solution. We calculate the particle's fitness by putting its position into a designated objective function. When the fitness is higher, the corresponding X_i is "better". The ith particle's "flying" velocity is also a D-dimensional vector, denoted as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ $(i=1, 2, ..., m)$. Denote the best position of the ith particle as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$, and the best position of the colony as $P_g (p_{g1}, p_{g2}, \ldots, p_{gD})$, respectively. The PSO algorithm could be performed using the following equations

$$V_i(k+1) = w\, V_i(k) + c_1 r_1 (P_i - X_i(k))/\Delta t + c_2 r_2 (P_g - X_i(k))/\Delta t \tag{1}$$

$$X_i(k+1) = X_i(k) + V_i(k+1)\Delta t \tag{2}$$

where $i=1, 2, ..., m$, k represents the iterative number, w is the inertia weight, c_1 and c_2 are learning rates, r_1 and r_2 are random numbers between 0 and 1, Δt is the time step value, $V_i \in [V_{min}, V_{max}]$ where V_{min} and V_{max} are the designated vectors. The termination criterion for the iterations is determined according to whether the maximum generation or a designated value of the fitness of P_g is reached [7].

Simulated annealing is a local search algorithm. The searching process starts with an initial solution. A neighbor of this solution is then generated and the change of cost is calculated. For a general local search process, if a reduction of cost is found then the current solution is replaced by the generated neighbor. Otherwise, the current solution is retained. The process is repeated until no further improvement can be found in the neighborhood of the current solution [4].

In order to reduce the probability of trapping at a local optimum, we introduce the SA technique to the PSO. We use PSO to search for the initial best position firstly. If the best position P_g does not change for a specified generation, then the SA is used, which starts with P_g for a local search. This process is repeated until the termination condition is achieved.

3 Stock Cutting Algorithm

For the sake of simplicity, we assumed that all the pieces have fixed orientation and all cuts on the stock plate are infinitesimally thin. The pieces cutting from the stock plate are rectangular in shape. We describe the stock plate and the pieces in a free coordinates. The left bottom corner of the stock plate is placed at the origin. Each piece is denoted by a four-dimension vector (x_k, y_k, l_k, w_k), where (x_k, y_k) is the position of the left bottom corner of the piece on the stock plate, l_k and w_k are the length and width of the piece, respectively. Each particle in the colony denotes a cutting pattern. A cutting pattern is consisted of a set of pieces.

In PSO, the fitness function of a particle is taken as the ratio of the summed areas of the pieces completely placed on the stock plate to the total area of the stock plate.

In the SA the objective function of a particle is the trim loss which is the ratio of the lost area of the stock plate to the area of the stock plate.

The cutting algorithm deals with the particles (the cutting pattern). In order to place the pieces on the stock plate, we should convert the cutting pattern to an actual layout. In this paper, we present a converting approach which is similar to the Bottom Left (BL) algorithm [5]. We call it the Coordinate-based Bottom Left Algorithm (CBL). In the BL algorithm, the piece is firstly put at the right upper corner of the stock plate and then it is moved to the left. In this paper, in order to use the coordinates of the piece's left bottom corner, we place the piece according to the coordinate then move it. We sort the pieces on x_k to reduce the probability of overlapping.

Table 1. Simulated results for the five test cases

Number of pieces	Trim loss of PSO (%)	Trim loss of hybrid algorithm (%)	Size of stock plate
10	0	0	100×80
10	0	0	40×20
15	7.5	3.7	40×20
20	6.3	3.4	40×40
30	7.1	5.5	65×45

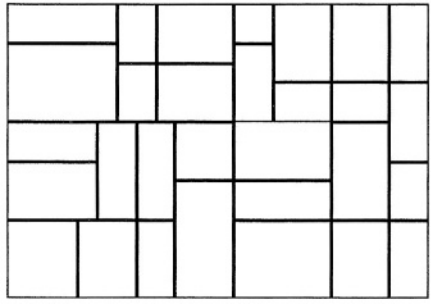


Fig. 1. A test problem with 30 pieces to be cut from the stock plate

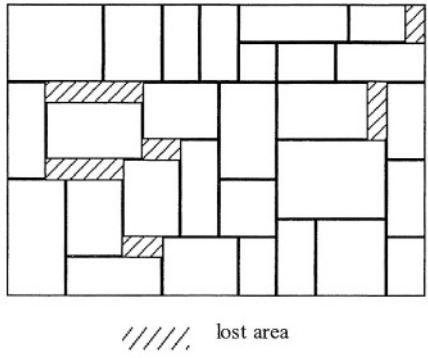


Fig. 2. Cutting results using the hybrid algorithm

4 Conclusions and Discussions

Five test problems are used to examine the proposed hybrid algorithm. Each of the five test problems has its own optimal solution of zero trim loss. Thus we can estimate easily the performance of the algorithm. The number of the rectangular pieces in each stock plate ranges from 10 to 30. The population size of the colony is taken as 60, and the parameters in the PSO are taken as $w = 0.3$, $c_1 = 0.8$ and $c_2 = 0.9$. The initial and final temperatures are taken as 80 and 1, respectively, in the SA. The algorithms are written in C and run in a Pentium 4 personal computer with 2.0GHz. The maximum number of iterations is taken as 1000.

Figure 1 shows one of the five test problems, where 30 pieces is cut from the stock plate. Figure 2 shows the actual layout generated using the hybrid algorithm. Twenty-eight pieces are cut from the stock pieces. The shade in the stock plate represents the lost area.

Table 1 shows the simulation results for the five test cases. From the results of the test cases it can be seen that the hybrid algorithm based on the PSO and the SA proposed in this paper has better ability to search for the global optimum for cutting stock problem. When the number of the pieces is smaller, both PSO and the hybrid algorithm work well. With lager number of pieces, the performance of the hybrid algorithm is better than the PSO's.

References

1. P. C. Gilmore and R. E. Gomory: A Linear Programming Approach to the Cutting Stock Problem. Operations Research, Vol.9 (1961) 849-859
2. P. C. Gilmore and R. E. Gomory: Multistage Cutting Stock Problems of Two and More Dimensions. Operations Research, Vol.13 (1965) 94-120
3. Stefan Jokobs: On Genetic Algorithms for the Packing of Polygons. European Journal of Operational Research, 88 (1996) 165-181
4. K. K. Lai and W. M. Chan: Developing a Simulated Annealing Algorithm for the Cutting Stock Problem. Computer and Industrial Engineering, Vol.33 (1997) 115-127
5. T. W. Leung, C. H. Yung and Marvin D. Troutt: Applications of Genetic Search and Simulated Annealing to the Two-dimensional Non-guillotine Cutting Stock Problem. Computer and Industrial Engineering, 40 (2001) 201-214
6. J. Kennedy, R. C. Eberhart: Particle Swarm Optimization. Proc. IEEE Conf. on Neural Networks, IV, Piscataway, NJ (1995) 1942–1948
7. X. H. Shi, L. M. Wan, H. P. Lee, X. W. Yang, L. M. Wang and Y. C. Liang: An Improved Genetic Algorithm with Variable Population-size and A PSO-GA Based Hybrid Evolutionary Algorithm. Second International Conference on Machine Learning and Cybernetics (2003) 1735-1740

Evolving TSP Heuristics Using Multi Expression Programming

Mihai Oltean and D. Dumitrescu

Department of Computer Science,
Faculty of Mathematics and Computer Science,
Babeş-Bolyai University, Kogălniceanu 1
Cluj-Napoca, 3400, Romania.
{moltean, ddumitr}@cs.ubbcluj.ro

Abstract. Multi Expression Programming (MEP) is used for evolving a Traveling Salesman Problem (TSP) heuristic for graphs satisfying triangle inequality. Evolved MEP heuristic is compared with Nearest Neighbor Heuristic (NN) and Minimum Spanning Tree Heuristic (MST) on some difficult problems in TSPLIB. The results emphasizes that evolved MEP heuristic is better than the compared algorithm for the considered test problems.

1 Introduction

Multi Expression Programming [4] is a Genetic Programming (GP) [3] that uses a linear representation of the chromosomes. MEP representation is similar to the way in which *C* and *Pascal* compilers translate mathematical expressions into machine code.

MEP technique is used here for discovering Traveling Salesman Problem (TSP) heuristics for graphs satisfying triangle inequality (TI graphs). This option was chosen due to the existence of a big number of real-world applications implying TI graphs (e.g. plains, trains and vehicles routes). MEP technique is used to learn a path function f that is used for evaluating the reachable nodes. This function serves as a heuristic for detecting the optimum path.

Evolved function f is compared with some well known heuristics. Numerical experiments emphasize that (for considered examples) MEP function outperforms dedicated heuristics.

2 MEP for TSP

TSP problem for TI graphs (i.e. satisfying triangle inequality) is stated as follows.

Consider a set $C = \{c_0, c_1, \ldots, c_{N-1}\}$ of cities, and a distance $d(c_i, c_j) \in \mathbb{Z}^+$ for each pair $c_i, c_j \in C, d(c_i, c_j) = d(c_j, c_i)$, and for each three cities $c_i, c_j, c_k \in C$, $d(c_i, c_j) \leq d(c_i, c_k) + d(c_k, c_j)$. The tour $< c_{\pi(0)}, c_{\pi(1)}, \ldots, c_{\pi(N-1)} >$ of all cities in C having minimum length is needed [2]

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 670–673, 2004.
© Springer-Verlag Berlin Heidelberg 2004

TSP problem with triangle inequality is an NP-complete problem [2]. No polynomial time algorithm for solving TSP problem is known. Several heuristics for solving TSP problem have been proposed. The most important are Nearest Neighbor and the Minimum Spanning Tree [1,2].

In this section we address the problem of discovering heuristics that can solve TSP rather than solving a particular instance of the problem. MEP technique is used for evolving a path function f that gives a way to choose graph vertices in order to obtain a Hamiltonian cycle. The fitness is assigned to a function f in the current population by applying f on several randomly chosen graphs (training set) and evaluating the results. Evolved path function may be used for solving particular instances of TSP. For each problem the graph nodes are evaluated using the path function f and are added one by one to the already build path. The algorithm for TSP using evolved path function f may be described as follows:

S_1. Let $c_{\pi(0)} = c_0$ {the path starts with the node c_0}
S_2. $k = 1$;
S_3. **while** $k < N - 1$ **do**
S_4. Using function f select $c_{\pi(k+1)}$ – the next node of the path
S_5. Add $c_{\pi(k+1)}$ to the already built path.
S_6. $k = k + 1$;
S_7. **endwhile**

S_4 is the key step of this algorithm. The procedure that selects the next node of the path in an optimal way uses the function f evolved by the MEP technique as described in the next sections.

Path function f has to use (as input) some information about already build path and some information about unvisited nodes. We consider a special terminal set which is independent with respect to the number of graph nodes.

Let us denote by y_1 the last visited node (current node). We have to select the next node to be added to the path. In this respect all unvisited nodes are considered. Let us denote by y_2 the next node to be visited.

For evolving path function f we consider a set T of terminals involving the following elements:

$d_y_1_y_2$ – distance between the graph nodes y_1 and y_2,

$min_g_y_1$ ($min_g_y_2$) – the minimum distance from the nodes y_1 (y_2) to unvisited nodes,

$sum_g_y_1$ ($sum_g_y_2$) – the sum of all distances between nodes y_1 (y_2) and unvisited nodes,

$prod_g_y_1$ ($prod_g_y_2$) – the product of all distances between nodes y_1 (y_2) and unvisited nodes,

$max_g_y_1$ ($max_g_y_2$) – the maximum distance from the nodes y_1 (y_2) to unvisited nodes,

length – the length of the already built path.

Table 1. The performance of evolved MEP heuristic, NN and MST on some problems in TSPLIB. *Length* is the length of the TSP path obtained with one of the considered heuristics. Each node of the graph has been considered as the first node of the path

Problem	MEP		NN		MST	
	Length	Error (%)	Length	Error (%)	Length	Error (%)
a280	2858.86	10.85	3084.22	19.58976	3475.23	34.75
att48	37188.2	10.93	39236.9	17.04227	43955.8	31.11
bier127	134945	14.08	127954	8.177068	152747	29.13
d198	17780.7	12.67	17575.1	11.37579	17957.6	13.79
d493	43071.3	23.05	41167	17.61328	41846.6	19.55
d657	56965.6	16.46	60398.7	23.48442	63044.2	28.89
fl417	13933.8	17.47	13828.2	16.58545	16113.2	35.85
gil262	2659.17	11.82	2799.49	17.72456	3340.84	40.48
kroB100	24801	12.01	25883	16.90077	28803.5	30.09
kroB200	33267.4	13.01	35592.4	20.91042	40619.9	37.98
pcb442	56948.3	12.15	57856.3	13.9397	73580.1	44.9
pr264	55827.1	13.61	54124.5	10.15468	65486.5	33.27
rat575	7573.6	11.82	7914.2	16.84925	9423.4	39.13
rat783	9982.96	13.36	10836.6	23.05928	11990.5	36.16
rd400	16973.3	11.07	18303.3	19.77816	20962	37.17
u574	43095.6	16.77	44605.1	20.86465	50066	35.66
u724	46545.7	11.06	50731.4	21.04844	60098.9	43.39

Set T of terminals is chosen in such way to be independent of the number of graph nodes. This choice confers flexibility and robustness to the evolved heuristic.

For evolving a MEP function for TSP problem we may consider the following set of function symbols: $F = \{+, -, /, *, cos, sin, min, max\}$.

The node y_2 that generates the lowest output of evolved function f is chosen to be the next node of the path. Ties are solved arbitrarily. For instance we may consider the node with the lowest index is selected.

Example

Consider the MEP linear structure:
1: $d_y_1_y_2$
2: $min_g_y_1$
3: + 1, 2
4: $sum_g_y_2$
5: * 2, 4
This MEP individual encodes the path functions f_1, f_2, f_3, f_4, f_5 given by:
$f_1 = d_y_1_y_2$,
$f_2 = min_g_y_1$,
$f_3 = d_y_1_y_2 + min_g_y_1$,
$f_4 = sum_g_y_2$,
$f_5 = min_g_y_1 * sum_g_y_2$.

The fitness (quality) of a detected path function f is defined as the sum of the TSP path length of graphs in the training set. Thus the fitness is to be minimized.

3 Numerical Experiments

In this experiment we evolve a heuristic for solving TSP problem and we assess its performance by applying it to some real-world instances taken from TSPLIB [5].

Let us denote by G_k the set of class of TI graphs having maximum k nodes.

MEP algorithm considers the class G_{50} (i.e. graphs having 3 to 50 nodes) for training and the class G_{100} for validation. MEP algorithm parameters are *Population Size* = 300; *Number of generations* = 100; *Chromosome length* = 40.

A path function evolved by the MEP algorithm is:

$$f = (sum_g(y2)) * (d_y_1_y_2 - (max(d_y_1_y_2, max_g(y_1))) + d_y_1_y_2).$$

Heuristic function f that is evolved by MEP technique is directly used for building the optimal path. For assessing the performance of evolved MEP heuristic we use it for solving several difficult problems in TSPLIB [5]. Other two well-known heuristics are used in this comparison: Nearest Neighbor and Minimum Spanning Tree heuristic [1,2]. Results are presented in Table 1.

Table 1 shows that evolved MEP heuristic performs better than NN and MST on all considered problems.

References

1. Cormen, T.H., Leiserson, C.E., Rivest, R. R.: Introduction to Algorithms, MIT Press, (1990)
2. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to NP–Completeness, Freeman & Co, San Francisco, CA, (1979)
3. Koza, J. R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, (1992)
4. Oltean, M.: Solving Even-Parity Problems using Multi Expression Programming, In: Ken Chen et. al (eds.): Proceedings of the the 7^{th} Joint Conference on Information Sciences (2003) 315-318
5. Reinelt, G.: TSPLIB - A Traveling Salesman Problem Library, ORSA, Journal of Computing, Vol. 3, no 4, (1991) 376-384

Improving the Performance of Evolutionary Algorithms for the Multiobjective 0/1 Knapsack Problem Using ε-Dominance

Crina Groşan and Mihai Oltean

Department of Computer Science
Faculty of Mathematics and Computer Science
Babeş-Bolyai University, Kogălniceanu 1
Cluj-Napoca, 3400, Romania.
{cgrosan, moltean}@cs.ubbcluj.ro

Abstract. The multiobjective 0/1 knapsack problem is a generalization of the 0/1 knapsack problem in which multiple knapsacks are considered. A new evolutionary algorithm for solving multiobjective 0/1 knapsack problem is proposed in this paper. This algorithm used a ε-dominance relation for direct comparison of two solutions. Several numerical experiments are performed using the best recent algorithms proposed for this problem. Experimental results clearly show that the proposed algorithm outperforms the existing evolutionary approaches for this problem.

1 Introduction

The 0/1 knapsack problem is a widely studied problem due its practical importance. In the last years a generalization of this problem was well studied and many algorithms for solving this variant have been proposed. Of great interest are the evolutionary approaches for solving the multiobjective 0/1 knapsack problem. Many papers on the multiobjective knapsack problem and on the algorithms proposed for solving it can be found in the literature [1,2,3]. In this paper, we propose a new evolutionary approach for multiobjective 0/1 knapsack problem. We use the ε-dominance concept which is a generalization of the standard Pareto concept.

In section 2 of the paper the multiobjective 0/1 knapsack problems are presented. The description of the newly proposed algorithm is given in section 3 of the paper. The definition of ε-dominance concept is also given in section 3. Some comparisons with the most recent algorithms (such as SPEA2, NSGA II, PESA) are performed in section 4.

2 Problem Statement

The classical 0/1 knapsack problem can be formulated as follow: a set of n items and a knapsack of capacity c are considered. Each item has a profit p_j and a

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 674–677, 2004.
© Springer-Verlag Berlin Heidelberg 2004

weight w_j. The problem is to select a subset of the items whose total weight does not exceed knapsack capacity c and whose total profit is maximum.

The problem can be extended for an arbitrary number of knapsacks. A set of n items and a set of k knapsacks are considered. Each item has k weight and k profits corresponding to the each knapsack. Each knapsack has its own capacity. The objective is to select a subset of the items with maximum overall profit among those whose weight does not exceed the knapsack capacity for any of the k dimensions.

3 Proposed Algorithm

The proposed algorithm uses ε-Pareto dominance relation between solutions.

We give here the definition of ε-Pareto dominance concept which is a generalization of the Pareto dominance concept.

Definition

Consider a maximization problem. Let x, y be two decision vectors (solutions) from the search space. Solution x *ε-dominate* y if and only if the following conditions are fulfilled:

$$f_i(x) \geq f_i(y), \forall \quad i = 1, 2, \ldots, n,$$
$$\exists j \in \{1, 2, \ldots, n\}: f_j(x) > f_j(y) + \varepsilon.$$

Each individual is a binary string. The value 1 for the position j of the chromosome means that the item j is selected to be included in a knapsack.

The proposed algorithm uses steady-state as its underlying mechanism and can be described as follows:

The algorithm starts with a population of randomly generated individuals. For each chromosome the total items weight for each knapsack is computed. If there are knapsacks for which the allowed capacity is exceeded the items starting with the one for which the proportion utility/weight has the smaller value are eliminated. This process continues until there are no knapsacks for which the capacity is exceeded. All nondominated solutions are computed using the ε-dominance concept. The following steps are repeated until a termination condition is reached: Two nondominates solutions (the parents) are randomly chosen. The parents are recombined using uniform crossover operator and the offspring are mutated. For each offspring the procedure for eliminating items if the capacity of one of knapsacks is exceeded is used. The offspring enters the population and the dominated solutions are removed.

This algorithm is called ε Multiobjective Knapsack Algorithm (ε-MOKA).

Table 1. The values of ε for the case of three knapsacks

Number of functions evaluations	The value of ε	Number of generations after which ε becomes ε-1
150,000	1000	500
288,000	1000	1000
576,000	1500	1000

Table 2. The results obtained for three knapsacks by ε-MOKA, NSGA II, SPEA2 and PESA considering 150,000 functions evaluations. Results are averaged over 30 runs

	ε-MOKA	NSGA II	SPEA 2	PESA
ε-MOKA		0.90301	0.939242	0.994266
NSGA II	0		0.976883	0.989555
SPEA 2	0	0.906996		0.990861
PESA	0	0.862837	0.908916	

Table 3. The results obtained for three knapsacks by ε-MOKA, NSGA II, SPEA2 and PESA considering 288,000 functions evaluations. Results are averaged over 30 runs

	ε-MOKA	NSGA II	SPEA 2	PESA
ε-MOKA		0.787402	0.910033	0.985815
NSGA II	0		0.996314	0.990203
SPEA 2	0	0.853698		0.985883
PESA	0	0.850779	0.93234	

4 Experimental Results

We test our algorithm considering 750 items and three knapsacks. The results obtained by ε-MOKA are compared to the results obtained by SPEA2 [3], NSGA II [2] and PESA [1]. For this comparison C metric introduced by Zitzler is used. Test data are taken from [4].

Table 4. The results obtained for three knapsacks by ε-MOKA, NSGA II, SPEA2 and PESA considering 576,000 functions evaluations. Results are averaged over 30 runs

	ε-MOKA	NSGA II	SPEA 2	PESA
ε-MOKA		0.58796	0.851949	0.931406
NSGA II	0		0.996237	0.991706
SPEA 2	0	0.832425		0.96905
PESA	0	0.809001	0.939329	

Using C metric two sets of nondominated solutions can be compared to each other. $C(A, B)$ return the proportion between number of solutions from B which are dominated by solutions from A over number of solutions from B. This number is a number between 0 and 1. The value $C(A, B) = 1$ means that all decision vectors in B are dominated by A. The value $C(A, B) = 0$ represent the situation when none of the points in B are dominated by A.

General parameters used by the considered algorithms are: number of knapsacks: 3, population size: 300, number of functions evaluations: 150,000, 288,000, 576,000.

The value of ε for ε-MOKA is chosen as follows: at the beginning of the search process this value is a large one. In this way a low dominance is ensured and this allow us to preserve many solutions in the first generations. After a number of iterations the value of ε is decreased by 1. Reducing the value of ε we will ensure the true Pareto dominance at the end of the search process.

The values of ε are presented in Table 1.

The results obtained by applying C metric for 150,000, 288,000 and 576,000 are presented in Table 2, Table 3 and Table 4, respectively.

Solutions obtained by ε-MOKA are not dominated by solutions obtained by NSGA II, SPEA 2 and PESA even for 576,000 functions evaluation. This means ε-MOKA has a very good convergence.

References

1. Corne, D.W., Knowles, J.D.: The Pareto-Envelope based Selection Algorithm for Multiobjective Optimization. In Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature, Springer-Verlag, Berlin (2000) 839–848
2. Deb, K., Agrawal, S., Pratap A., Meyarivan T.: A fast elitist non – dominated sorting genetic algorithm for multi-objective optimization: NSGA II. In: Shoenauer, M. et al. (eds), Parallel Problem Solving From Nature – PPSN VI, Springer-Verlag, Berlin (2000) 849–858
3. Zitzler, E., Marco Laumanns and Thiele, L.: SPEA 2: Improving the Strength Pareto Evolutionary Algorithm. TIK Report 103, Computer Engineering and Networks Laboratory (TIK), Departament of Electrical Engineering Swiss federal Institute of Technology (ETH) Zurich (2001)
4. http://www.tik.ee.ethz.ch/ zitzler/testdata.html

Genetic Evolution Approach for Target Movement Prediction

Sung Baik [1], Jerzy Bala [2], Ali Hadjarian[3], and Peter Pachowicz[3]

[1] School of Computer Engineering
Sejong University, Seoul 143-747, KOREA
sbaik@sejong.ac.kr
[2] School of Information Technology and Engineering
George Mason University
Fairfax, VA 22030, U.S.A.
jbala@gmu.edu
[3] Sigma Systems Research, Inc.
Fairfax, VA 22032, U.S.A.
{ahadjarian, ppach}@sigma-sys.com

Abstract. This paper presents a genetic evolution system, for target movement prediction, which includes functions inferring opponents' strategic movements and displaying such predicted movements in an interactive 3D visualization space. To speed up the analysts' ability to access and integrate information, the prediction approach generates new movements based on past behaviors and application of an inheritance mechanism. It applies Genetic Algorithms (GAs) learning techniques to evolve new individuals in the population of movements in order to converge the evolution process toward optimal movements. The approach is implemented into the GEM (Genetic Evolution of Movement) system and its performance has been experimentally evaluated.

1 Genetic Evolution of Target Movement for Prediction

GEM uses Genetic Algorithms (GAs) [1-4] to generate the population of movement generation operators. By using crossover and mutation, the population evolves and only the strongest elements survive, thus contributing to improved performance in terms of more probable/optimal movements. This contribution is used as an objective evaluation function to drive the generation process in its search for new and useful movement generation operators. There are 6 steps for GEM's processing (Fig. 1.):

Step 1: Movements Population Seeding. This step sets the initial population of movements. It can use a random selection of initial movements or the Historical Movement Database (HMD). Movements selected from HMD represent historical GMTI target data for scenarios (i.e., tracking and targeting engagements) similar to the one for which the prediction is requested. This step also defines the evaluation function.
Step 2: Binary Encoding. A string representation for each movement is generated using a genetic encoding schema.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 678–681, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Step 3: Application of Genetic Operators. Crossover and mutation operations are applied during each evolutionary cycle generating additional movements (offspring movements).
Step 4: Binary Decoding. This step executes a reverse process to the binary encoding step (i.e., Step 2 of the algorithm).
Step 5: Evaluation. New movements in the population are evaluated according to the evaluation function defined in Step 1.
Step 6: Recombination. In this step worse performing movements (according to the evaluation function) are discarded from the population leaving the same number of encoded movements in the population as initially set up in Step 1. Step 6 feeds back to Step 2 and restarts the next iteration of the movement generation process.

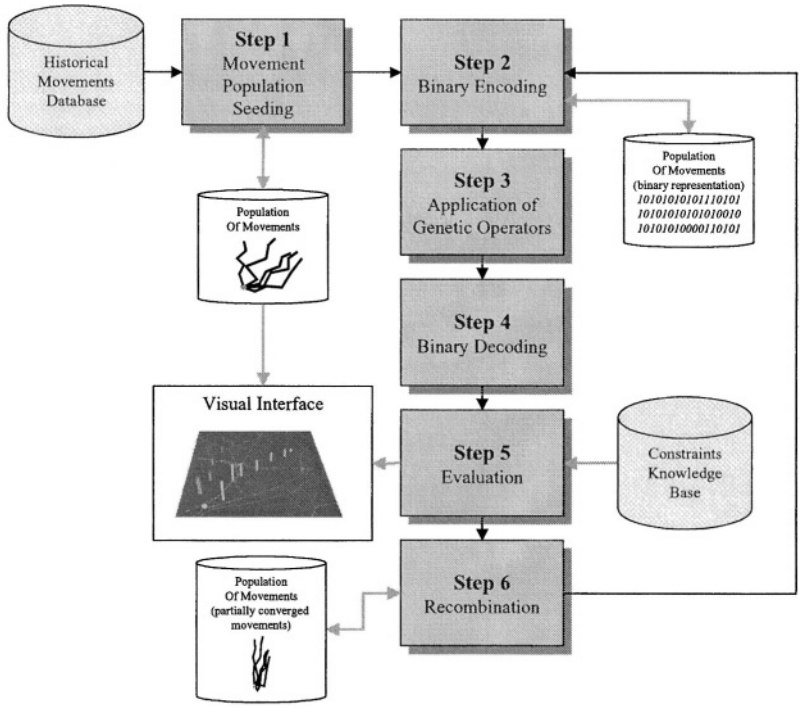


Fig. 1. Movement generation steps

1.1 GA Operations and Evaluation Function

The crossover operation, when applied to two operators produces offspring strings (operators). Offspring operators inherit movement characteristics from both their parents and the information about the relevant characteristics can be easily transferred to subsequent generations. This mechanism is possible because the data structure used to represent operators is "linearized." Linearized data structures are those string representations for which the relevance of genetic information is decreasing as one moves from the left towards the right fields/bits (moving through movement timeframes). As

a consequence, the last operation applied to the current movement causes only minor changes to the operator structure.

The mathematical analysis of GAs shows that they work best when the internal representation encourages the emergence of useful building blocks that can be subsequently recombine to produce improved performance. GEM's string decoding/encoding method identifies these building blocks as the movement fields and subfields in the string representation.

Mutation provides for occasional disturbances in the movement structure to insure diversity in the genetic individuals over long periods of time and to prevent stagnation in the search process for better movements.

The step 5 of GEM's processing evaluates movements in the population according to the evaluation function defined in Step 1. The function scores each movement by calculating partial scores for all <MSP, MP> pairs. The partial score is computed by linking <MSP, MP> pair's constraint parameters to the "constraint map". The map consists of several spatially arranged layers representing sets of constraints (e.g., geographical, tactical, communicational, and doctrinal). The location constraint on a given layer represents a probability distribution for the <MSP, MP> pair to originate at this location (i.e., MSP to be at this location at timeframe T, and with MP linking this location to the next MSP at timeframe T+1).

Each movement in the genetic population has to be evaluated according to an evaluation function. Each MSP probability at timeframe T is computed as the product of the location probability value at the (X,Y) location and the MSP probability at timeframe T-1.

1.2 Genetic Algorithms

GAs are iterative procedures which maintain a "population" of candidate solutions *P(t)* to the objective function *f(x):*

$$P(t) = \langle x1(t), x2(t), \ldots, xN(t) \rangle \quad (1)$$

Each structure *xi* in population *P* is simply a binary string of length L. Generally, each *xi* represents a vector of parameters to the function *f(x),* but the semantics associated with the vector is unknown to the GA. During each iteration step, called a "generation", the current population is evaluated, and, on the basis of that evaluation, a new population of candidate solutions is formed.

The initial population *P(0)* is usually chosen at random. Alternately, the initial population may contain heuristically chosen initial points. In either case, the initial population should contain a wide variety of structures. Each structure in *P(0)* is then evaluated. For example, if we are trying to minimize a function f, evaluation might consist of computing and storing *f(x1),... , f(xN).*

The structures of the population *P(t+1)* are chosen from the population *P(t)* by a randomized "selection procedure" that ensures that the expected number of times a structure is chosen is proportional to that structure's performance, relative to the rest of the population. That is, if *xj* has twice the average performance of all the structures in *P(t),* then *xj* is expected to appear twice in population *P(t+1).* At the end of the selection procedure, population *P(t+1)* contains exact duplicates of the selected structures in population *P(t).*

In order to search other points in the search space, some variation is introduced into the new population by means of idealized "genetic recombination operators." The most important recombination operator is called "crossover." Under the crossover operator, two structures in the new population exchange portions of their binary representation. This can be implemented by choosing a point at random, called the crossover point, and exchanging the segments to the right of this point. Termination may be triggered by finding an acceptable approximate solution to *f(x),* by fixing the total number of evaluations, or some other application dependent criterion.

2 System Interfacing and Experiments

A 3D visualization space is used in the GEM system to interface with the user. This interfacing space renders depiction of current movements in the evolved population of movements. It is also used as an editor. The editor can access the Historical Movement Database (HMB) to select and visualize the initial population of movements, to be used for population seeding, visualize intermittent results of the prediction process, and visually control various systems' parameters. This system-interfacing component has been implemented using In3D - a cross-platform Java class library designed to enable the creation of interactive 3D visualizations of complex information spaces (developed by Visual Insights Corp.)

The performance of the GEM system has been experimentally evaluated on its ability to converge the population of movements to the most probable movement in the lowest number of genetic cycles. After some evolutionary runs of the genetic engine, all movements and their corresponding significant points are mostly converged to one movement. During experiments, a non-convergence result was observed when the evaluation function is formed for a highly multi-modal initial search space. The population of individual movements converged to two separate movements. In such case, the search process has to start from a larger set of initial movements (i.e., search from a larger number of points in the initial cycle of GAs). A higher mutation rate can also potentially alleviate this problem.

References

1. Bethke, A. D., "Genetic algorithms as function optimizers," Ph.D. Thesis, Dept. Computer and Communication Sciences, Univ. of Michigan, 1981.
2. Brindle, A., "Genetic algorithms for function optimization," Ph.D. Thesis, Computer Science Dept., Univ. of Alberta, 1981.
3. DeJong, K. A., "Adaptive system design: a genetic approach," IEEE Trans. Syst., Man, and Cyber., vol. SMC-10, no. 9, pp. 566-574, Sept. 1980.
4. DeJong, K. A., "Analysis of the behavior of a class of genetic adaptive systems," Ph. D. Thesis, Dept. Computer and Communication Sciences, Univ. of Michigan, 1975.

Adaptive Transfer Functions in Radial Basis Function (RBF) Networks

Günther A. Hoffmann

Humboldt University Berlin, Department of Computer Science,
Rechnerorganisation und Kommunikation ROK, 10099 Berlin, Germany
gunho@informatik.hu-berlin.de

Abstract. The quality of Radial Basis Functions (RBF) and other nonlinear learning networks such as Multi Layer Perceptrons (MLP) depend significantly on issues in architecture, learning algorithms, initialisation heuristics and regularization techniques. Little attention has been given to the effect of mixture transfer functions in RBF networks on model quality and efficiency of parameter optimisation. We propose Universal Basis Functions (UBF) with flexible activation functions which are parameterised to change their shape smoothly from one functional form to another. This way they can cover bounded and unbounded subspaces depending on data distribution. We define UBF and apply them to a number of classification and function approximation tasks. We find that the UBF approach outperforms traditional RBF with the Hermite data set, a noisy Fourier series and a non φ-separable classification problem, however it does not improve statistically significant on the Mackey-Glass chaotic time series. The paper concludes with comments and issues for future research.

1 Introduction and Objectives

The quality of Radial Basis Functions (RBF) and other nonlinear learning networks such as Multi Layer Perceptrons (MLP) depends significantly on architecture, learning algorithms, initialisation heuristics and regularization techniques. Little attention has been given to the effect of flexible transfer functions in RBF networks on model quality and efficiency of parameter optimisation. Even though Gaussian and sigmoidal transfer functions prevail in literature, there is little a priori reason why models solely based on these transfer function will provide optimal model quality. In this paper we investigate the effects of adaptive transfer functions, which replace the Gaussian transfer functions in RBF.

The rest of the paper is organized as follows: Before we describe the universal basis function (UBF) approach in more detail in chapter three we review previous attempts and the state of the art of universal transfer functions in chapter two. RBF have been described and reviewed extensively in literature [1][6][13]. We will not revisit this topic. We will briefly comment on the problem of parameter optimisation in chapter four. We will then investigate the effect of different types of transfer func-

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 682–686, 2004.
© Springer-Verlag Berlin Heidelberg 2004

tions on selected problems such as the Hermite data set, the chaotic Mackey-Glass time series, a noisy Fourier series and a classification problem in chapter five. In chapter six we discuss our results and outlay future work.

2 Related Work

A straight forward integration of MLP and RBF was carried out in [3]. In [4] and [5] an extensive overview of transfer functions is given. Little practical expertise has been published. To the best of our knowledge a systematic comparison of transfer functions is yet missing.

3 Universal Basis Functions: RBF with Mixed Transfer Functions

We propose flexible kernels as in (2) to replace the standard Gaussian kernels (1) in RBF. We call this approach universal basis functions (UBF). These novel kernels can be parameterised to change their shape smoothly from one functional form to another. The Gaussian transfer function is a special case of UBF.

$$G(r) = e^{-r^2/2\sigma^2} \tag{1}$$

The kernels we propose consist of a mixture of two activation functions. To combine different types of activation functions in one kernel, additive and multiplicative mixtures have been proposed in [3] [5]. We propose a mixture of activation functions by smoothly morphing one activation function into the other as in (2). We combine Gaussian (4), sigmoid (5) and multiquadratic (6) activation functions.

$$G(\mathbf{x};\mathbf{r},\boldsymbol{\sigma},\omega') = \omega'\Phi_1(\mathbf{x};\mathbf{r},\boldsymbol{\sigma}) + (1-\omega')\Phi_2(\mathbf{x};\mathbf{r},\boldsymbol{\sigma}) \tag{2}$$

with

$$\omega' = \tanh\left(\frac{1}{2}\omega\right); \omega \in (-\infty,\infty) \tag{3}$$

and $\Phi \in \Phi'_{1..3}$

$$\Phi'_1 = e^{-r^2/2\sigma^2} \tag{4}$$

$$\Phi'_2 = \tanh(r/\sigma^2) \tag{5}$$

$$\Phi'_3 = \sqrt{r^2+\sigma^2} \tag{6}$$

If we do have information about the data distribution we can incorporate this knowledge by giving a bias towards a specific activation function by initialising the slider value ω' accordingly. However, as frequently is the case knowledge about the data distribution is either difficult to obtain or not available at all. In this case we start with an educated guess. If no further information is available to justify a bias towards a particular activation function we set $\omega' = 0.5$.

4 Parameter Optimisation

In UBF we have to optimise the joint set of parameters $\Theta = \{t, \sigma, \omega'\}$ with kernel positions **t,** kernel widths σ and the UBF slider value ω' which controls the shape of the kernels. We employ a global stochastic optimisation procedure, the derandomised Evolution Strategy with full covariance matrix adaptation (CMA-ES). This procedure is a stochastic optimisation scheme which has been described in detail in [7] and [14].

5 Results

Figure 1 depicts the mean square error of the classification problem as a function of the UBF slider value ω'. For a Gaussian RBF kernel network we get a straight line because the kernel shape does not change with a varying ω'. For a mixture UBF kernel approach we clearly get an optimal kernel shape minimizing the models classification error at $\omega' \approx 0.2$ implying a mildly Gaussian bias.

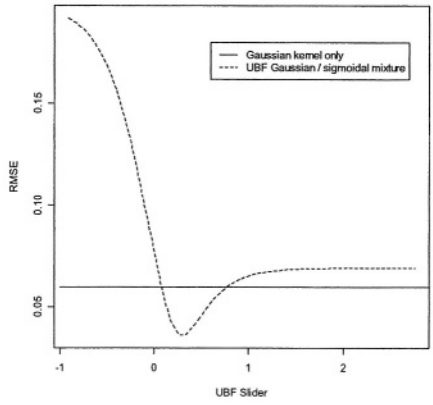


Fig. 1. The categorization error in terms of root-mean-square-errors (RMSE) is shown as the straight line for the RBF model. The UBF mixture model shows different errors as we smoothly change the activation function from Gaussian to sigmoidal. A UBF slider value larger than one, signals a strong Gaussian influence, a value below zero signals sigmoid influence.

6 Discussion and Future Work

In this paper we have proposed Universal Basis Functions (UBF) as an extension to Radial Basis Functions (RBF). We investigated the models generalisation capabilities in terms of means square errors. We compared UBF and RBF. We find that classification of a non trivial dichotomy and function approximation of the Hermite data set and a noisy Fourier series the UBF approach significantly outperforms the RBF approach. We also find that for the Mackey-Glass data series the UBF approach does not significantly outperform the traditional RBF approach. UBF offer the possibility to adapt the transfer function locally to the search space. RBF are a special case of UBF. This is achieved by evolving a transfer function which is best suited for covering the local parameter space. The activation function Φ can thereby take on any form ranging smoothly from a Φ_1 to a Φ_2. For a Gaussian $\Phi_1 = \Phi_2$ we get the traditional RBF network. If the data distribution is known the optimal activation function might be chosen based on analytical and theoretical consideration. However, in high dimensional space such knowledge about the data distribution might be difficult to obtain or not available at all. In these cases UBF might be a powerful extension of RBF. There are a number of open issues which we will address in future research. Further work is in progress to analyse the theoretical limits of UBF. Furthermore we will investigate how to reduce the complexity of the parameter space. Also currently we work with only one mixture value ω' for all kernels and all dimensions. We expect a higher impact on model quality if we introduce a full ω'-matrix allowing for adaptation of the activation function for each kernel.

References

1. Bishop C. M. (1995) *Neural Networks for Pattern Recognition;* Clarendon Press
2. Cybenko G. (1989) *Approximation by Superposition of a Sigmoidal Function;* Mathematics of Control Signals and Systems 2, pp. 303-314
3. Dorffner G. (1994) *A Unified Framework for MLPs and RBFNs: Introducing Conic Section Function Networks;* Dept. of Medical Cybernetics and Artificial Intelligence University of Vienna
4. Duch W., Jankowski N. (1999) *Survey of Neural Transfer Functions;* Neural Computing Surveys 2,163-212
5. Duch W., Jankowski N. (2001) *Transfer functions: hidden possibilities for better neural networks;* ESANN'2001 proceedings - European Symposium on Artificial Neural Networks Bruges (Belgium), pp. 81-94
6. Girosi F., Jones M., Poggio T. (1993) *Regularization Theory and Neural Networks;* MIT Cambridge; AI Memo 1430
7. Hansen, Müller, Koumoutsakos (2003); *Reducing the time complexity of the derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES);* Evolutionary Computation 11(1):1-18, MIT Press
8. Hoffmann G. A. (1997b); *Evolving Parameter Sets for Conditional Density Models;* Intl. Symposium on Multi-Techn. Inf. Proc.; Taiwan

9. Hutchinson James M., Lo Andrew, Poggio Tomaso (1994) *A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks;* MIT AI Memo No. 1471 or C.B.C.L. Paper No. 92
10. Jordan M. I., Bishop C. M. (1996); *Neural Networks;* AI Memo No. 1562; MIT
11. Lapedes A., Farber R. (1987) *Nonlinear Signal Processing using Neural Networks; Prediction and System Modeling;* Proceedings of Real Time Systems Symposium
12. Park J., Sandberg W. (1991); *Universal Approximation using radial basis function networks;* Neural Computation vol 3:(2):246-257
13. Poggio T., Girosi F. (1989); *A Theory of Networks for Approximation and Learning;* AI Memo 1140, MIT Cambridge
14. Ostermeier A., Gawelczyk A., Hansen N. (1993); *A Derandomized Approach to Self Adaptation of Evolution Strategies;* Technische Universität Berlin, Technical Report: TR-93-003

Disturbance Rejection Control of Thermal Power Plant Using Immune Algorithm

Dong Hwa Kim and Jae Hoon Cho

Dept. of Instrumentation and Control Eng., Hanbat National University,
16-1 San Duckmyong-Dong Yuseong-Gu, Daejon City, Korea, 305-719
kimdh@hanbat.ac.kr, ial.hanbat.ac.kr
Tel: +82-42-821-1170, Fax: +82-821-1164

Abstract. A PID Controller has been used to operate this system because of its implementational advantages. However, it is very difficult to achieve an optimal PID gain with no experience, since the gain of the PID controller has to be manually tuned by trial and error. This paper focuses on tuning of the PID Controller with disturbance rejection using immune network algorithm. To decide the performance of response, an ITSE (Integral of time weighted squared error) is used in this paper.

1 Introduction

The normal power plant operation requires tracking the steam demand while maintaining the steam pressure and the steam temperature of at their respective setpoints, despite variations of the steam load. However, if the overall system is to be driven to an operating point different from the design point, the interaction variables are very likely to vary from their design values. Therefore, the local controllers need to be robust in order to accommodate these variations. Up to now, a Proportional–Integral–Derivative (PID) controller has been used in the control system of power plant. However, it cannot effectively control such a complicated or fast running system, since the response of a plant depends on only the gain P, I, and D. In this paper, an intelligent tuning method by a immune algorithm is suggested for robust control with disturbance rejection on control system of power plant.

2 Control Characteristics of Thermal Power Plant for Controller Design

A thermal power plant is mainly composed of one boiler whose steam output feeds one or two turbine, driving an electric generator. There can be many available models for each subsystem with a varying degree of complexity and accuracy. The models

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 687–690, 2004.
© Springer-Verlag Berlin Heidelberg 2004

are nonlinear MIMO system, obtained through both physical and empirical methods and compared well against actual plant data [3].

$$\frac{d\lambda_d}{dt}=\omega_b\left[\frac{r}{l_d}(\lambda_{AD}-\lambda_d)-\omega\lambda_q-v_d\right],\ \frac{d\lambda_d}{dt}=\omega_B\left[\frac{r_F}{l_F}(\lambda_{AD}-\lambda_F)-v_F\right],\ \frac{d\lambda_d}{dt}=\omega_B\left[\frac{r_D}{l_D}(\lambda_{AD}-\lambda_D\right]$$

where λ_d, λ_F, *and* λ_D are the direct axis, field, and damper flux linkages, respectively, and ω_B and ω are the based frequency and actual frequency respectively. The mutual flux linkage is given by $\lambda_{AD}=L_{MD}(\lambda_d/l_d+\lambda_F+/l_F+\lambda_D/l_D)$, and the d-axis and field currents are given by $i_d=(1/l_d)(\lambda_d-\lambda_{AD})$, $i_F=(1/l_F)(\lambda_F-\lambda_{AD})$,

$$\frac{d\lambda_q}{dt}=\omega_b\left[\frac{r}{lq}(\lambda_{AQ}-\lambda_q)-\omega\lambda_d-v_d\right],\ \frac{d\lambda_Q}{dt}=\omega_b(\frac{r_Q}{l_Q}(\lambda_{AQ}-\lambda_Q)\ \lambda_{AQ}=L_{MQ}(\lambda_q/l_q+\lambda_Q+/l_Q),$$

$i_q=(1/l_q)(\lambda_q-\lambda_{AQ})$. The frequency deviation is given as a function of the mechanical torque and electric torque $\frac{d\omega_{\Delta u}}{dt}=\frac{1}{2H}(T_m-T_e-D\omega_{\Delta u})$, $\frac{d\delta}{dt}=\frac{180\omega_R}{\pi}\omega_{\Delta u}$, where $\omega_{\Delta u}$, T_m, T_e, *and* δ are the per-unit frequency deviation, mechanical torque, electric torque, and rotor angle, respectively.

3 PID Controller Tuning with Disturbance Rejection Function by Immune Algorithms

3.1 Condition for Disturbance Rejection

In Fig. 1, the disturbance rejection constraint can be given by $\max_{d(t)\in D}\frac{\|Y\|}{\|d\|}=\left\|\frac{w(s)}{1+K(s,c)G(s)}\right\|_\infty\langle\delta$. Here, $\delta\langle 1$ is constant defining by the desired rejection level and $\|\bullet\|_\infty$ denotes the H_∞-norm, which is defined as $\|G(s)\|_\infty=\max_{\omega\in[0,\infty)}|G(j\omega)|$. The disturbance rejection constraint becomes

$$\left\|\frac{w(s)}{1+K(s,c)G(s)}\right\|_\infty=\max_{\omega\in[0.\infty)}\left(\frac{w(j\omega)w(-j\omega)}{1+K(j\omega,c)G(j\omega,c)K(-j\omega,c)G(-j\omega,c)}\right)^{0.5}$$
$$=\max_{\omega\in[0.\infty)}(\sigma(\omega,c))^{0.5} \tag{1}$$

The controller K (s, c) is written as $K(s,c)=c_1+\frac{c_2}{s}+c_3 s$, $c=[c_1,\ c_2,\ c_3]^T$. Hence, the condition for disturbance rejection is given as $\max_{\omega\in[0.\infty)}(\sigma(\omega,c))^{0.5}\langle\delta$.

3.2 Performance Index for Optimal Controller Design

The performance index defined as ITSE (Integral of the Time-Weighted Square of the Error) is written by $PI = \int_0^\infty t(E(t))^2 dt$, $E(s) = B(s)/A(s) = \sum_{j=0}^{m} b_j s^{m-1} \Big/ \sum_{i=0}^{n} a_i s^{n-1}$.

Because *E(s)* contains the parameters of the controller *(c),* the value of performance index, *PI* for a system of n*th* order can be minimized by adjusting the vector c as $\min_c PI(c)$, The optimal tuning is to find the vector c, such that the ITSE performance index, *PI (c)* is a minimum and the constraint $\max_{\omega\in[0.\infty)} (\sigma(\omega,c))^{0.5} \langle \delta$ is satisfied through real coded immune algorithms.

3.3 Evaluation Method for Disturbance Rejection Based on Immune Algorithms

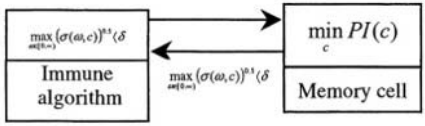


Fig. 1. Immune algorithm based computational structure for optimal parameter selection.

Let the ITSE performance index be *PI(c).* Then the value of the fitness of each individual of immune network $c_i (i = 1,.....n)$ is determined by the evaluation function, denoted by $\Gamma(c_i)$ as

$$\Gamma_1(c_i) = -(PI_n(c_i) + \Phi(c_i)),$$

The penalty function $\Phi(c_i)$ is discussed in the following. Let the disturbance rejection constraint be $\max(\alpha(\omega,c_i))^{0.5}$. The value of the fitness of each individual of memory cell $\omega_j (j = 1,...,m)$ is determined by the evaluation function, denoted by $\Omega(\omega_j)$ as $\Omega(\omega_j) = \alpha(\omega,c_i)$. The penalty for the individual c_i is calculated by means of the penalty function $\Phi(c_i)$ given by

$$\Phi(c_i) \begin{cases} M_2 \text{ if } c_i \text{ is unstable, } M_1 \max(\omega, c_i) \text{ if } \max(\alpha(\omega,c_i))^{0.5} > \delta, \\ 0 \text{ if } \max(\alpha(\omega,c_i))^{0.5} < \delta. \end{cases}$$

3.4 Computational Procedure for Optimal Selection of Parameter

[Step 1.] Initialization and recognition of antigen: That is, initialize the populations of network c_i $(i = 1,...,n)$ and memory cell $\omega_j (j = 1,...,m)$.

[Step 2.] Product of antibody from memory cell: For each individual c_i of the network population, calculate the maximum value of $a(\omega,c_i)$ using memory cell. If no individuals of the network satisfy the constraint $\max(a(\omega,c_i))^{0.5} < \delta$,

[Step 3.] Calculation for searching a optimal solution: Calculate the fitness value for each individual c_i.

[Step 4.] Stimulation and suppression of antibody: The expected value η_k of the stimulation of the antibody is given by $\eta_k = m_{\varphi k}/\sigma_k$,where σ_k is the concentration of the antibodies. Through this function, for each individual c_i of the network, calculate $\max(a(\omega, c_i))$ using memory cell, and initialize the gene of each individual $\omega_j (j = 1,...,m)$ in the population.

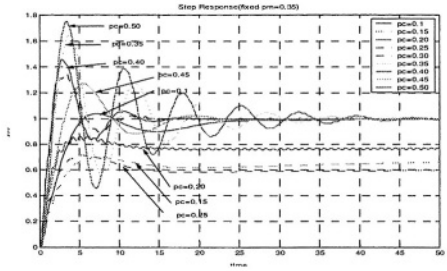


Fig. 2. Response to average values on parameter learning of immune network. (Pm=0.35, Pc=0.1 to

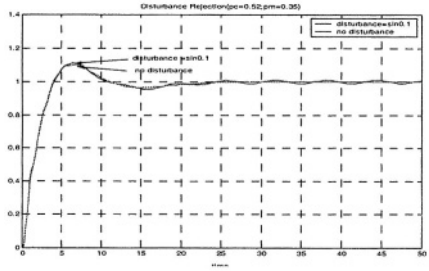


Fig. 3. Response to disturbance rejection. (Pc=0.52, Pm=0.35)

[Step 5.] Stimulation of Antibody: If the maximum number of generations of memory cell is reached, stop and return the fitness of the best individual $\max(a(\omega, c_i))$ to network; otherwise, go to step 3.

4 Simulations and Discussions

The simulation results are shown as Fig. 2-3. Fig. 2 represents response to average values on parameter learning of immune network on parameters, Pm=0.35, Pc=0.1 to 0.5. Fig. 3 illustrates comparison between immune based PID control depending on generation variation and genetic algorithm based PID control depending on generation variation.

References

1. David Lindsley: Boiler Control Systems. McGrawill (1991)
2. Matsummura, S.: Adaptive control for the steam temperature of thermal power plants. Proceedings the 1993 IEEE on Control applications. (1998) 1105-1109
3. Farmer, J. D., Packard, N. H and Perelson, A. S.: The immune system, adaptation, and machine learning, Vol. Physica. D, No. 22, (1986) 187 – 204.
4. Kazuyuki Mori and Makoto Tsukiyama: Immune algorithm with searching diversity and its application to resource allocation problem. Trans. JIEE, Vol. 113 - C, No. 10, (1993).
5. Dong Hwa Kim: Intelligent tuning of a PID controller using a immune algorithm. Trans. KIEE , vol. 51-D, no.1, (2002).

The Design Methodology of Fuzzy Controller Based on Information Granulation (IG)-Based Optimization Approach

Sung-Kwun Oh[1], Seok-Beom Roh[1], and Dong-Yoon Lee[2]

[1] School of Electrical, Electronic and Information Engineering, Wonkwang University, South Korea
{ohsk, nado}@wonkwang.ac.kr

[2] Department of Information and Communication Engineering, Joongbu University, South Korea
dylee@joongbu.ac.kr

Abstract. In this study, we present information granulation (IG)-based optimization approach to the design of fuzzy controller. The design procedure dwells on the use of evolutionary computing (genetic algorithms) and the estimation of the polynomial model that is carried out by effectively combining Hard C-Means Clustering (HCM) Method with Least Mean Square (LSM) method. The developed approach is applied to a nonlinear system such as an inverted pendulum where we show the results of comprehensive numerical studies and carry out a detailed comparative analysis

1 Introduction

Recently, the ongoing challenge we are facing when design advanced system control has resulted in a diversity of underlying methodologies, development platforms and detailed algorithms. As one of the representative design approaches, a design approach of fuzzy controller by means of IG-based estimation technique is introduced in this study. The intent of this study is to develop, optimize and experiment with the fuzzy controllers when developing a general design scheme of Computational Intelligence. One of the difficulties in the construction of the fuzzy controller is to derive a set of optimal control parameters of the controller such as linguistic control rules, scaling factors, and membership functions of the fuzzy controller[7,8]. Genetic algorithms (GAs) can be used to find the optimal control parameters. However, evolutionary computing is computationally intensive and this may be a point of concern when dealing with amount of time available to such search. For instance, when controlling a nonlinear plant such as an inverted pendulum of which initial states vary in each case, the search time required by GAs could be prohibitively high when dealing with dynamic systems. As a consequence, the parameters of the fuzzy controller cannot be easily adapted to the changing initial states of this system such as an angular position

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 691–694, 2004.
© Springer-Verlag Berlin Heidelberg 2004

and an angular velocity of the pendulum. To alleviate this shortcoming, we introduce a nonlinear mapping such as HCM-LMS tandem based polynomial. The development process consists of two main phases. First, using genetic optimization we determine optimal parameters of the fuzzy controller for various initial states (conditions). Second, we build up a nonlinear model that captures a relationship between the initial states of the system and the corresponding genetically optimized control parameters.

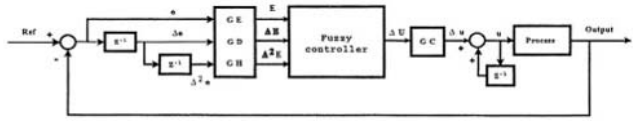


Fig. 1. An overall architecture of the fuzzy PID controller

2 Design Methodology of Fuzzy Controller

The block diagram of fuzzy PID controller is shown in Figure 1 [5,6].

2.1 Auto-tuning of the Fuzzy Controller by Gas

Genetic algorithms (GAs) are the search algorithms inspired by nature in the sense that we exploit a fundamental concept of a survival of the fittest as being encountered in selection mechanisms among species[1]. The overall design procedure of the fuzzy PID controller realized by means of GAs consists of the following steps

[Step 1.] Select the general structure of the fuzzy controller according to the purpose of control and dynamics of the process.

[Step 2.] We set the initial individuals of GAs for the scaling factor of fuzzy controller.

[Step 3.] Here, all the control parameters such as the scaling factors GE, GD, GH and GC are tuned at the same time.

2.2 HCM-LMS Tandem Based Polynomial Model

Step1: Fix $c(2 \le c < n)$ and initialize the partition matrix $\mathbf{U}^{(0)} \in M_c$

Step 2: Calculate the center vectors $\mathbf{v}_i$ of each cluster :

$$v_{ij}^{(r)} = \sum_{k=1}^{n} u_{ik}^{(r)} \cdot x_{kj} / \sum_{k=1}^{n} u_{ik}^{(r)} \tag{1}$$

Step 3: Update the partition matrix $\mathbf{U}^{(r)}$

$$d_{ik} = d(\mathbf{x}_k - \mathbf{v}_i) = \left\| \mathbf{x}_k - \mathbf{v}_i \right\| = \left[\sum_{j=1}^{m} (x_{kj} - v_{ij})^2 \right]^{1/2} \tag{2}$$

$$u_{ik}^{(r+1)} = \begin{cases} 1 & d_{ik}^{(r)} = \min\{d_{jk}^{(r)}\} \text{ for all } j \in c \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Step 4: Check a termination criterion. If

$$\| U^{(r+1)} - U^{(t)} \| \leq \varepsilon \text{ (tolerance level)} \tag{4}$$

stop ; otherwise set r = r + 1 and return to step 2[2][3].

3 Experimental Studies

In this section, we demonstrate the effectiveness of the fuzzy PID controller by applying it to the inverted pendulum system. The control goal is to balance the pole starting from nonzero conditions by supplying appropriate force to the cart[4].
First, we select several initial angular positions and angular velocity and obtain the auto-tuned control parameters by means of GAs according to the change of each selected initial angular positions and angular velocity, then build a table. Secondly, we use HCM-LMS tandem-based polynomial to estimate the control parameters in the case that the initial angular positions and angular velocity of the inverted pendulum are selected arbitrarily within the given range. Table 1 shows the estimated scaling factors of the fuzzy PID controller and describes performance index of the fuzzy PID controller with the estimated scaling factors. Figure 1 demonstrates pole angle for (a) Case 1 and (b) Case 2 respectively

Table 1. The estimated parameters by means of the HCM-LMS tandem based polynomial and performance index(ITAE, Overshoot(%)) of the fuzzy PID controller

Case	θ	θ	GE	GD	GH	GC	ITAE	OS(%)	Rising Time(sec)
1	0.22	0.22	2.951133	60.959881	243.663513	1.770746	0.169122	0.083650	0.145199
2	0.22	0.45	1.713128	63.498253	246.882355	1.886055	0.634893	0.000000	0.322975
3	0.22	0.78	2.600188	59.980007	269.163086	2.219068	0.285220	0.529590	0.169282
4	0.45	0.22	1.706843	42.527325	106.950111	6.492052	0.882041	0.000000	0.233517
5	0.45	0.45	1.789289	63.856773	84.001183	6.829825	1.267673	0.000000	0.285800
6	0.45	0.78	1.918489	61.384880	245.082535	3.747335	0.960818	0.000000	0.193290
7	0.78	0.22	1.427422	36.951550	174.614471	4.790281	2.421195	1.954839	0.209058
8	0.78	0.45	1.366799	39.255482	190.132248	3.867553	2.806530	1.624381	0.210987
9	0.78	0.78	1.290725	38.158276	181.288132	3.872698	3.501827	0.574551	0.217495

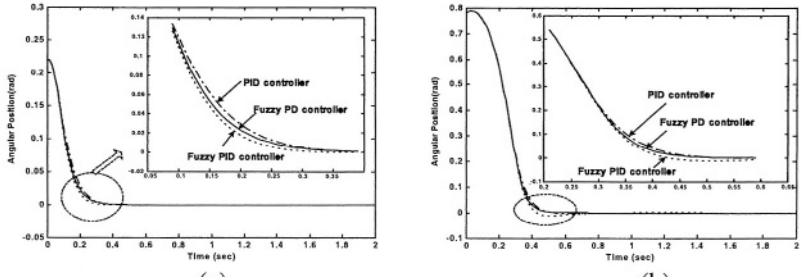


Fig. 2. Pole angle for (a) Case 1 and (b) Case 2

From the above Figures 2, we observe that the fuzzy PD and fuzzy PID controllers control effectively the inverted pendulum system. The proposed estimation algorithm such as HCM-LMS tandem-based polynomial generates the preferred model architecture.

4 Conclusion

In this paper, we have proposed a two-phase optimization scheme of the fuzzy PID and PD controllers. The parameters under optimization concern scaling factors of the input and output variables of the controller that are known to exhibit an immense impact on its quality. The first phase of the design of the controller uses genetic computing that aims at the global optimization of its scaling factors where they are optimized with regard to a finite collection of initial conditions of the system under control. In the second phase, we construct a nonlinear mapping between the initial conditions of the system and the corresponding values of the scaling factors. From the simulation studies, using genetic optimization by scaling factor estimation modes and the estimation algorithm of the HCM-LMS tandem based polynomial model, we showed that the fuzzy PD/PID controller controls effectively the inverted pendulum system in a nonlinear range of θ.

Acknowledgement. This work has been supported by EESRI(R-2003-0-285), which is funded by MOCIE (Ministry of commerce, industry and energy)

References

1. D.E. Goldberg: Genetic algorithms in Search, Optimization, and Machine Learning. Addison-Wesley (1989)
2. H.-S. Park, S.-K. Oh: Multi-FNN identification Based on HCM Clustering and Evolutionary Fuzzy Granulation. International Journal of Control, Automation, and Systems. 1(2) (2003) 194-202
3. J. Q. Chen, Y. G. Xi, Z.J. Zhang: A clustering algorithm for fuzzy model identification. Fuzzy Sets and Systems. 98 (1998) 319-329
4. J.R. Jang: Self-Learning Fuzzy Controllers Based on Temporal Back Propagation. IEEE Trans. On Neural Networks. 3(1992) 714-723
5. T.j. Procyk, E.H. Mamdani: A linguistic synthesis of fuzzy controller. Automatica. 15 (1979) 15- 30
6. H. X. Li: A comparative design and tuning for conventional fuzzy control. IEEE Trans. Syst., Man, Cybern. B. 27 (1997) 884-889
7. S.-K. Oh, W. Pedrycz: The Design of Hybrid Fuzzy Controllers Based on Genetic Algorithms and Estimation Techniques. Kybernetes 31(2002) 909-917
8. S.K. Oh, T. Ahn, H. Hwang, J. Park, K. Woo: Design of a Hybrid Fuzzy Controller with the Optimal Auto-tuning Method. Journal of Control, Automation and Systems Engineering. 1 (1995)63-70

PID Controller Tuning of a Boiler Control System Using Immune Algorithm Typed Neural Network

Dong Hwa Kim

Dept. of Instrumentation and Control Eng., Hanbat National University,
16-1 San Duckmyong-Dong Yuseong-Gu, Daejon City, Korea, 305-719.
kimdh@hanbat.ac.kr, ial.hanbat.ac.kr
Tel: +82-42-821-1170, Fax: +82-821-1164

Abstract. Dead time processes exist widely in many types of systems such as chemical processes, and the main steam temperature control system of the thermal power plant. A PID Controllers have been used to operate these systems. However, it is very difficult to achieve an optimal PID gain with no experience since the gain of the PID controller has to be manually tuned by trial and error. This paper suggests a tuning method of the PID Controller for a process with long dead time using an immune algorithm typed neural network, through computer simulation. Tuning results of immune algorithms based neural network are compared with the results of genetic algorithm.

1 Introduction

When using a PID controller in these plants, the plant is generally controlled without consideration of disturbance rejection. Therefore, an industrial experience is required for tuning in these systems: [1]. Traditionally, PID controllers applied to these plants are tuned with a reduction of gain so that overall stability can be obtained. This results in poor performance of control. Failure to tune in control will cause an inevitable plant shutdown, and a loss of production and considerable damage to the plant may result. An effective tuning is required to maintain the system reliability and stability following a system disturbance: [1], [2] In this paper auto-tuning scheme of the PID controller using reference model and immune network is suggested and simulated for an effective control of dead time process.

2 Problems of the PID Controller on the Dead Time Process

There are many well known PI and PID tuning formulas for stable processes. However, PID tuning formulas for unstable processes, complex plants, and dead time process are less common. Up to this time, many sophisticated tuning algorithms have been tried an attempt to improve the PID controller performance under such difficult conditions since the control performance of the system depends on the P, I, D parameter gains. In the PID tuning methods, the method proposed by Ziegler Nichols (1942) needs the ultimate gain and the period of the ultimate oscillation at stability limit. But it is difficult to determine their exact values experimentally in real

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 695–698, 2004.
© Springer-Verlag Berlin Heidelberg 2004

processes, since the oscillatory operation must be avoided. Hence, a new idea for automatic tuning of the PID control parameters, auto-tuning, is required. In utility boiler of Fig. 1, the boiler-turbine model is given as [1], [2]

3 Multiobjective Controller Design of Boiler-Turbine System Using Immune Algorithm Based Neural Structure

3.1 The Response of Immune System

Generally the concentration of i-*th* antibody, which is denoted by δ_i, is calculated as follows [7], [8]:

$$\frac{dS_i(t)}{dt} = \left(\alpha \sum_{j=1}^{N} m_{ji}\delta_j(t) - \alpha \sum_{k=1}^{N} m_{ik}\delta_k(t) + \beta m_i - \gamma_i \right) \delta_i(t), \quad \frac{d\delta_i(t)}{dt} = \frac{1}{1+\exp\left(0.5 - \frac{dS_i(t)}{dt}\right)}$$

In Eq., m_{ji} denotes affinities between antibody j and antibody i (i.e. the degree of interaction), m_i represents affinities between the detected antigens and antibody i, respectively. The affinity $m_{\alpha\beta}$ between antibody α and antibody β is given as $m_{\alpha\beta} = 1/\{1+\Omega(\alpha\beta)\}$, $\Omega(\alpha\beta) = f(x) = [f_1(x) + f_2(x) + f_3(x)]$, where $\Omega_{\alpha\beta}$ is an information which obtained by antibody α and antibody β. Generally $m_{\alpha\beta}$ is given by range of 0-1.

3.2 Controller Design by Immune Algorithm

In Fig. 1, $r_{=1,2,3}$ is reference input over 0, $y_{i=1,2,3}$ is plant output. Immune algorithm typed neural network has the following function in each layer;

Layer 1. As function comparing the reference input with the output of the given plant, comparing result, $r_{=1,2,3}$ is selected for affinity in sub-function and it is defined by the followings.

Layer 2. As layer for computing coupling degree between inputs of multivariable system as Fig. 1, the sub-function defined as f_a, f_b, f_c is used for coupling degree as;

$$f_a = f_1(x_1) + f_2(x_2),\ f_b = f_1(x_1) + f_3(x_3),\ f_c = f_2(x_2) + f_3(x_3). \tag{1}$$

Layer 3. As layer for detecting error between reference input and plant output, the following equation is defined as

$$f_i(x_i) = \sum_{t=0}^{n} \left(|r_1 - y_1| \right)/n,\ i = 1, 2, 3,\ \textit{if } |r_i - y_i|_t \geq r_i \textit{ then } f_i(x_i)_t = 1, \textit{ else } f_i(x_i)_t = |r_i - y_i|_t,\ i = 1, 2, 3$$

Layer 4. They provide stimulation action by algorithms defined as

$$f_{ri} : \textit{if } f_a < f_b \textit{ then stimulation } f_i(x_i), \textit{ else stimulation } f_i(x_i),\ i = 1, 2, 3. \tag{2}$$

Layer 5. Affinity is calculated with $H_s = f_1(x_1) + f_2(x_2) + f_3(x_3)$. When the value of H_s is smaller, more satisfactory results is obtained.

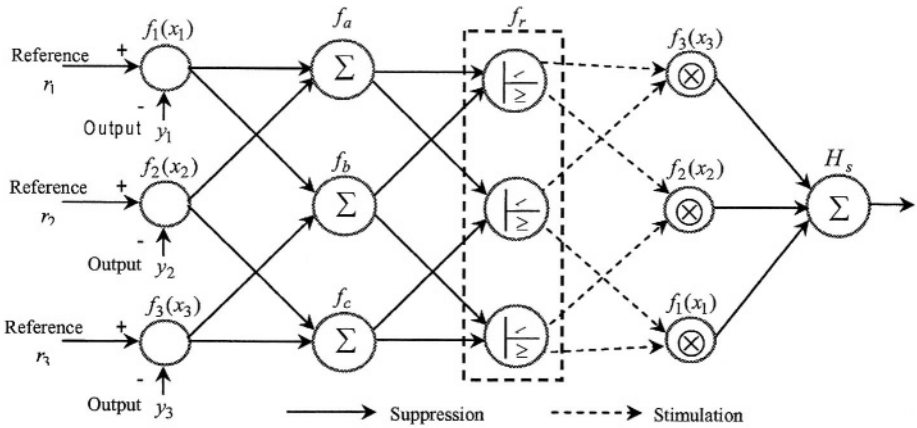


Fig. 1. Flow diagram of controller

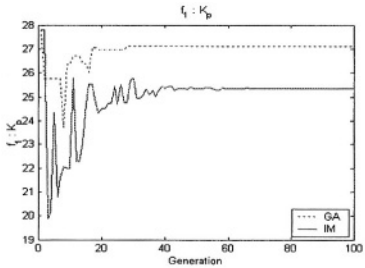


Fig. 2. Graph of sub-function f_1, K_p

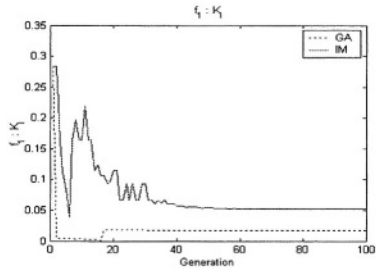


Fig. 3. Graph of sub-function f_1, K_i

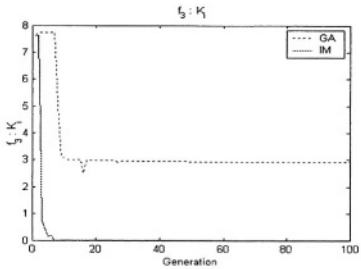


Fig. 4. Graph of sub-function f_3, K_i

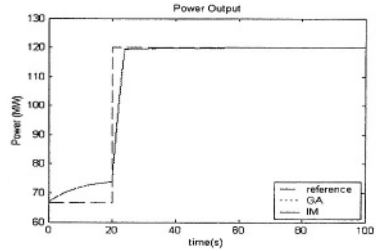


Fig. 5. Pressure response

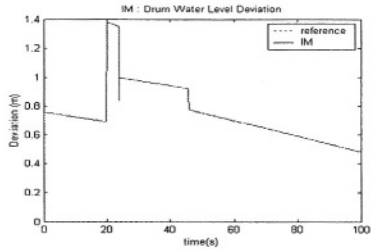


Fig. 6. Water level deviation using immune algorithm

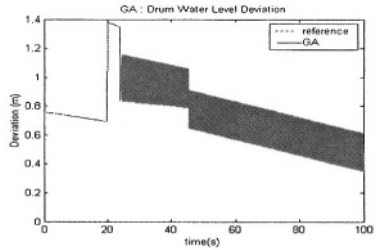


Fig. 7. Water level deviation using genetic algorithm

4 Simulation and Discussion

Figs. 4-7 represent the variation results of sub-function, f_1, f_2, f_3 and the PI controller parameters, K_p, K_i is compared on immune algorithm and genetic algorithm. Fig. 5 is pressure responses and Fig. 6 obtained by immune algorithm has no oscillation against many oscillation based on genetic algorithm as shown in Fig. 7. The range of initial value on parameter PI is P: 0-10, I: 0-5.

5 Conclusion

The immune system possesses a self organizing and distributed memory. Therefore, it is thus adaptive to its external environment and allows a PDP (parallel distributed processing) network to complete patterns against the environmental situation. This paper suggests a tuning method of the PI Controller for a steam temperature process with long dead time using an immune algorithm typed neural network, through computer simulation. Tuning results by immune algorithms based neural network are compared with the results of genetic algorithm.

References

1. Teng Fong-Chwee: Self-tuning PID controllers for dead time process. IEEE Trans., vol. 35, no. 1. (1988) 119-125
2. J. D. Farmer, N. H. Packard and A. S. Perelson: The immune system, adaptation, and machine learning, *Physica.* vol. D, no. 22. (1986) 187 - 204
3. Dong Hwa Kim: Intelligent tuning of a PID controller using a immune algorithm. Trans. KIEE , vo. 51-D, no.1. (2002) 78-91

A Framework to Investigate and Evaluate Genetic Clustering Algorithms for Automatic Modularization of Software Systems

Saeed Parsa and Omid Bushehrian

Iran University of Science and Technology , Narmak , Tehran , Iran
{Saeed_Parsa,Boushehrian}@yahoo.com

Abstract. In this paper a software environment, called DAGC, is described. The main idea behind the design of DAGC is to facilitate research works in design and development of genetic clustering algorithms for automatic re-modularization of software systems. Within the DAGC environment, clustering algorithms may be assembled or modified by simply selecting the parts from an extendable list of components. Using this distinguishing feature of the DAGC framework, a new algorithm with a new encoding and crossover operator was evolved.

Keywords: Genetic Clustering, encoding Algorithm, Automatic Modularization

1 Introduction

Automatic clustering algorithms are used within the context of program understanding to discover the structure (architecture) of the program under study. Clustering is a key activity in reverse engineering to discover a better design of the systems [3, 4]. Genetic algorithms are widely believed to be effective on NP-complete global optimization problems, such as clustering, and they can provide good sub-optimal solutions in reasonable time [1].

Well known frameworks such as Bunch [4], CRAFT [2] and GAME [6] have provided environments to run and view the results of clustering on software systems. It should be noted that even if there are reusable software components for GA, the success of respective implementations still depends on appropriate definitions of the concepts for the specific problem. Our objective has been to develop an environment to experiment with the effects of applying different schemes for the components of genetic clustering algorithms for software re-modularization. To achieve this, we have developed a flexible software environment called DAGC.

2 Genetic Clustering

The idea has been to develop a software environment to facilitate researchers' investigations on development of optimal genetic clustering algorithms for automatic

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 699–702, 2004.
© Springer-Verlag Berlin Heidelberg 2004

re-modularization of software systems. To achieve this, a comprehensive study of the existing algorithms was carried out [1,4, 5]. We arrived with the results listed below:

1. There are a fixed set of component types, mostly used in these algorithms.
2. For each component type, a standard interface could be defined.
3. Using standard interfaces, various schemes may be applied to define the components without any need to change the existing source code for the algorithm.
4. For each component, a number of implementations could be provided.
5. A general clustering algorithm, calling the components could be developed.
6. New algorithms could be created by selecting the components.

In general, there are a fixed set of operators and components appearing in genetic clustering algorithms. DAGC makes use of a generalized genetic clustering algorithm. By defining a function with a standard interface for each of the component types, various schemes for the components may be selected or created within the DAGC, without any need to change the body of the algorithm.

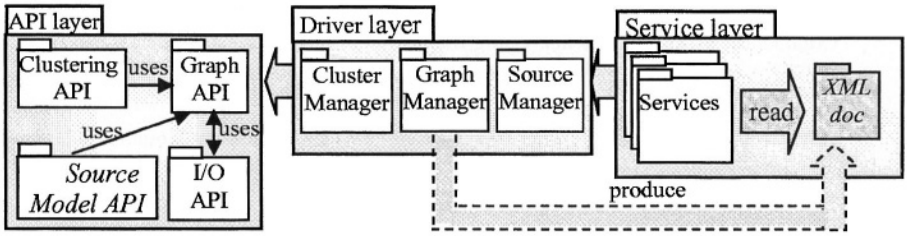


Fig. 1. The Architecture of the DAGC Framework

3 Architecture

As shown in figure 1, the DAGC architecture consists of three interacting layers: API, Driver and Services.

The API layer is an independent layer which includes a useful and complete set of java classes. These classes are arranged in four packages as follows:

- ClusteringAPI: The ClusteringAPI package provides useful classes and interfaces for the genetic components.
- GraphAPI: The GraphAPI package provides a set of classes which can be used to generate and display three kinds of benchmark graphs called Random, Caterpillar and software graphs [1].
- I/O API: This package provides a number of useful classes for loading and writing graph files. Here, we support three formats for importing or exporting graph files which are: Dotty-AT&T, XML and Text.
- SourceModelAPI: This package contains a number of classes to extract call graphs from a Java source code.

The service layer provides all the interfaces of the DAGC framework. At this moment DAGC presents the following services:

- Clustering Service: This service allows the user to create a new GA.

- Data Analyzer Service: This service analyzes the clustering results
- Customizing Service: This service provides some templates for genetic components and let us to customize them for building our new components.

4 DAGC Clustering Algorithm

A new algorithm, called DAGC was developed within the DAGC environment by simply substituting the parts of an existing algorithm called Bunch [4], with the corresponding components, selected from an extendable list of components. Substituting a component, the algorithm was executed within the environment on random graphs or class dependency graphs extracted from a given source code.

The DAGC clustering algorithm makes use of our new encoding and recombination schemes. In our new encoding scheme, each chromosome is a permutation of N integers. Here, the m^{th} gene of the chromosome, instead of holding a partition number k (which means node m of the graph resides in partition k), holds a value 1=<p<=N. To decode a chromosome into a clustering the following procedure can be used:

```
Suppose Array C holds a permutation of graph nodes 1 to N
  if C[i] >= i then Create a new cluster and assign node i to it.
  Else For i=1 to N do Assign node i to the same cluster as node C[i]
```

Since our coding is a permutation, we need an order based recombination operator to preserve the permutation form in each offspring. Here, a one point recombination operator is used. Suppose two parents P_1 and P_2 such that:

$P_1 = C_1C_2\ldots C_kC_{k+1}C_{k+2}\ldots C_L$ and $P_2 = D_1D_2\ldots D_kD_{k+1}D_{k+2}\ldots D_L$

To create two offspring O_1 and O_2, first, parents are split at position k. First parts of two offspring are created using the first parts of two parents as follows:

$O_1 = C_1C_2\ldots C_k$ and $O2 = D_1D_2\ldots D_k$

To fill the second parts of two offspring, we create the following list:

$C_{k+1}\ D_{k+1}C_{k+2}\ D_{k+2}\ldots C_L\ D_L$

Starting from the first element in the list, if the element does not belong to the first offspring, O_1, it will be augmented to O_1. Otherwise, it will be added to the second offspring, O_2. Analogously, we can consider the list $D_{k+1}\ C_{k+1}D_{k+2}\ C_{k+2}\ldots D_L\ C_L$ and generate two other offspring.

5 Evaluating DAGC Algorithm

To compare our new genetic algorithm, DAGC, with the Bunch [4], 9 different random graphs [1] with more than 200 vertices were used. Each graph was clustered 5 times, using both the DAGC and Bunch. The average and variance of the clustering results are shown in figure 2. Figure 2.b demonstrates the stability of the DAGC encoding scheme in comparison with the encoding scheme used by the Bunch.

To evaluate the performance of our permutation-based encoding scheme, the TurboMq fitness function was used. A comparison of the quality of the clustering results is presented in figure 2.a. Since the GA search space is highly reduced by using our permutation-based coding scheme, better clustering quality results were produced.

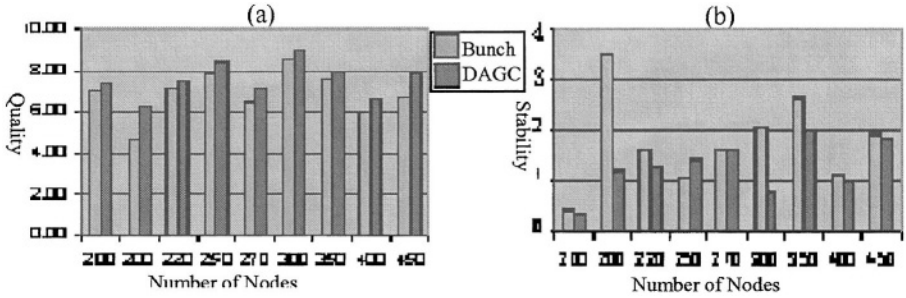


Fig. 2. Comparison of the results

6 Conclusions

In this paper we showed how a flexible environment for investigating clustering algorithms for re-modularization of software systems can be created. Using this environment, researchers can assemble new clustering algorithms by trying different schemes for the components and observing the effects of changing each component. The environment is capable of keeping different schemes for the components. Thereby, the components can be easily selected to assemble new algorithms. Using these distinguishing features of the DAGC, an efficient clustering algorithm was evolved by trying different schemes for the components of the Bunch genetic clustering algorithm. Since the GA search space is highly reduced by using our permutation-based coding scheme, better clustering quality results were produced.

References

1. Bui T. N., Moon B. R., "Genetic algorithm and graph partitioning," *IEEE Trans. Comput.,* vol. 45, pp. 841–855, July 1996.
2. Brian S. Mitchell, Spiros Mancoridis,"CRAFT: A Framework for Evaluating Software Clustering Results in the Absence of Benchmark Decomposition", IWPC, IEEE 2001.
3. A. F. Brito, "A Coupling Guided Cluster Analysis Approach to Reengineer the Modularity of Object Oriented Systems", Conf. on Soft. Maintenance and Reengineering, IEEE 2000.
4. Brian S. Mitchell, "Bunch: A Clustering tool for the Recovery and Maintenance of Software System Structure", International Conf. of Software Maintenance, IEEE 1999.
5. Cincotti A., Cuttelo V., Pavone M., "Graph Partitioning using Genetic Algorithms with ODPX", Proceedings of the World Congress on Computational Intelligence, IEEE 2002
6. Gokel N., Drechsler R., Becker B., "GAME: A Software Environment for Using Genetic Algorithms in Circuit design", Applications of Computer Systems, 240-247, 1997.
7. Brian S. Mitchell, Spiros Mancoridis, "Comparing the Decompositions Produced by Software Clustering Algorithms Using Similarity measurements ",in the proceedings of international conference on software maintenance (ICSM'01),Italy, IEEE 2001
8. David L. Woodruff, Optimization Software Class Libraries, Handbook, March 2002.

An Artificial Immune Algorithms Apply to Pre-processing Signals

Mariusz Święcicki[1], Wiesław Wajs[2], and Piotr Wais[3]

[1] Institute of Computer Modelling
Cracow University of Technology, Kraków, Poland
mswiecic@pk.edu.pl
[2] Institute of Automatics
University of Mining and Metallurg, Kraków, Poland
wwa@ia.agh.edu.pl
[3] The State Higher School of Vocational Education in Krosno
waisp@poczta.onet.pl

Abstract. Over the past few decades there has been a growing interest in the use of biology as a source of inspiration for solving computational problems. The motivation of this field is primarily to extract useful metaphors from natural biological systems, in order to create effective computational solutions to complex problems in a wide range of domain areas. The more notable developments have been the neural networks inspired by the working of the brain, and the evolutionary algorithms inspired by neo-Darwinian theory of evolution. This paper presents the theory of an immune network model, and it tries to apply to solve signal classification problems.

1 Introduction to Architecture Human Immune System

The immune system of vertebrates is composed of a great variety of molecules, cells, and organs spread all over the body. There is no central organ controlling the functioning of the immune system.The main task of the immune system is to survey the organism in the search for malfunctioning cells from their own body (e.g., cancer and tumour cells), and foreign disease causing elements (e.g., viruses and bacteria). Every element that can be recognised by the immune system is called an antigen (Ag). The cells that originally belong to our body and are harmless to its functioning are termed self (or self antigens), while the disease causing elements are named nonself (or nonself antigens). The immune system, thus, has to be capable of distinguishing between what is self from what is nonself; a process called self/nonsel discrimination, and performed basically through pattern recognition events. The main functions of the B cells include the production and secretion of antibodies (Ab) as a response to exogenous proteins like bacteria, viruses. Each B cell is programmed to produce a specific antibody. The antibodies are specific proteins that recognize and bind to another particular protein. The production and binding of antibodies is usually a way of sending signal for other cells to kill or remove the bound substance.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 703–707, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 An Artificial Immune Network for Signal Analysis

Main goal our system is classification signals so there are some problems that have to solve. The first problem is connected with algorithm of learning immune network. The next problem is related to structures of data, which are responsible for representation of signals. Solutions of these problems are presented below. The last paragraph shows result of classification signals by our immune network.

The signal comes inputs system, is interpreted as antibody (Ab) so task of immune network is found antigen Ag that will be suitable for Ab. The Ag-Ab representation will partially determine which distance measure shall be used to calculate their degree of interaction. Mathematically, the generalized shape of a molecule *(m),* either an antibody (Ab) or an antigen (Ag), can be represented by a set of real-valued coordinates $m = <m1, m2, ..., mL>$, which can be regarded as a point in an L-dimensional real-valued space.

$$D = \sqrt{\sum_{i=1}^{L} (ab_i - ag_i)^2} \tag{1}$$

The affinity between an antigen and an antibody is related to their distance that can be estimated via any distance measure between two vectors, for example the Euclidean or the Manhattan distance. If the coordinates of an antibody are given by $<ab1, ab2, ..., abL>$ and the coordinates of an antigen are given by $<ag1, ag2, ..., agL>$, then the distance *(D)* between them is presented in equation (1), and that use real-valued coordinates and that measure distance are called Euclidean shape-spaces.

In this article, we based on algorithm that was proposed in paper [1, 3] by de Castro and Von Zuben. This learning of immune net algorithm was adapted to classifying signals. Below this algorithm is presented. The learning algorithm lets building of set that recognizes and represents the data structural organization. The more specific the antibodies, the less parsimonious the network (low compression rate), whilst the more generalist the antibodies, the more parsimonious the network with relation to the number of antibodies. The suppression threshold controls the specificity level of the antibodies, the clustering accuracy and network plasticity.

For each $Agj \in Ag$ *do*

Determine its affinity fi,j, i = 1,...,N, to all Abi. fi,j =1/Dij, i = 1,...,N
A subset Ab{n} contains the n highest affinity antibodies is selected;
The n selected antibodies are going to clone proportionally to their antigenic affinity
*The set C is submitted to a directed affinity maturation process generating a mutated set C**
Determine the affinity dk,j = 1/Dk,j among Agj and all the elements of C:*
From C, re-select* $\zeta\%$ *of the antibodies with highest put them into matrix of clonal memory;*
Apoptosis: eliminate all the memory clones from Mj whose affinity $Dk,j > \sigma d$*:*
Determine the affinity si,k among the memory clones:

Clonal suppression: eliminate those memory clones whose si,k < σs:
Concatenate the total antibody memory matrix with the resultant clonal memory Mj
Determine the affinity among all the memory antibodies from Ab{m}:
Network suppression: eliminate all the antibodies such that si,k < σs:

3 Result of Experiments

This part of our paper includes results of which will be discussed. As was mentioned, we tried to construct system, which was able to classifying signals. Our experiments will consist of two phases, the first phase is phase of learning process of immune network. The second phase is phase of testing process. During Phase of testing, the signals will contain noise so it will make that process of recognizing will be either difficult or even impossible. For that reason the trained artificial immune net have to be cable of generalizing of signals. All Samples of signals have the same number of values. A Distance between either Ag and Ab or Ab and Ab will be counted by using the equation (1). A Set, which is used for training artificial immune net, includes patterns of signals. As was mentioned, in our case, the samples of signals that are included will be interpreted as antigens during the teaching process by having been trained net. A first task of system will be classification of samples, which belong to family of functions that are described by equations (2).

$$f_1(t,i) = \sin(\ i \cdot x) \quad f_2(t,i) = sign\ \ (\sin(\ i \cdot x)) \tag{2}$$

$$where \quad : i \in [0.2\ ,1.2]; \quad t \in [0\ ,5 \cdot \pi\ ]$$

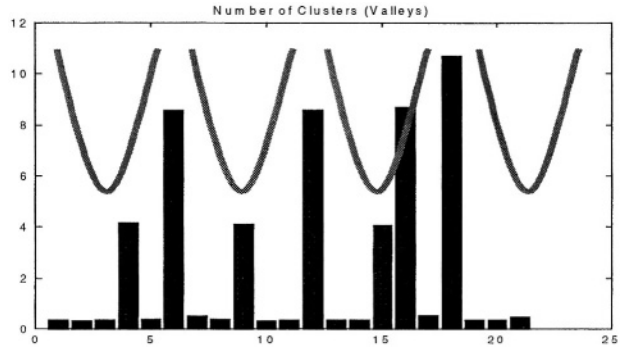


Fig. 1. Number of clusters (Peaks + 1, or Valleys) for this MST

There are 60 samples subdivided into 4 clusters (non-overlapping classes). Fig. 1 depict result of training of the immune network. Each of classes contains 6 cells. The net had following training parameters: $n = 4$, $\zeta = 0.2$, $\sigma d = 1.0$, $\sigma s = 0.001$ and d=10.

The stopping criterion is a fixed number of generations: *Ngen = 200.* The resulting network contains 22 cells. The suppression threshold *(σs)* controls the specificity level of the antibodies, the clustering accuracy and network plasticity. In order to provide the user with important information on how to set up the immune network algorithm parameters.

The change of parameters algorithm have influence on sensitivity analysis of the algorithm. An Especially, the σs suppression threshold is responsible for sensitivity work of the immune net.

3.1 A Phase of Testing of Artificial Immune Net

On input of the trained immune net will be delivered samples of signals don't belong to the learning set. The samples of signals were intentionally deformed by added noise. In the Fig. 2 is result of the immune net answer. The dashed line presents values of input sample. The continuous line shows the best answer of our net. The algorithm that was described in the third paragraph, returns a queue of possible answer. The queue is ordered and first place in the queue is occupied by vector of signal values that are connected with class that is matched an input signal.

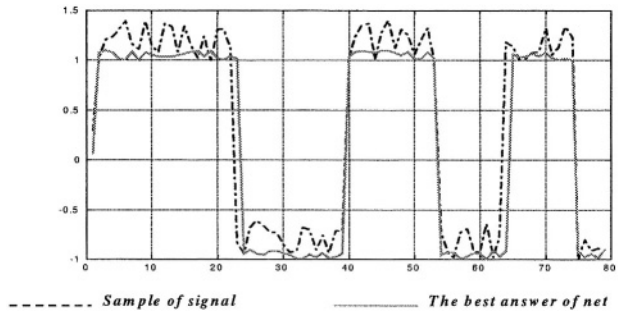


Fig. 2. Result of process classification input signal

As we can see, the Fig.2 shows very interesting case, because although the input signal was very deformed, the artificial immune network correctly classified it. That is mean that the net is able to generalization of input signals.

4 Conclusion

This paper described mechanics of immune network in human organism. In this paper is included describing positive selection. It is known that these the phenomenon are used for building of artificial immune systems.. The model is connectionist in nature but it follows an evolutionary-like learning algorithm that is the immune clonal selec-

tion principle. Finally, there was presented model of application that based on immune network theory. The application was solving problems of identification signals.

References

1. De Castro, L. N., Von Zuben, F. J. (2000a), An Evolutionary Immune Network for Data Clustering, Proc. of the IEEE SBRN, pp. 84-89.
2. De Castro, L. N., Von Zuben, F. J. (2000b), The Clonal Selection Algorithm with Engineering Applications, GECCO'00 – Workshop Proceedings, pp. 36-37.
3. De Castro, L. N., Von Zuben, F. J. (1999), Artificial Immune Systems: Part I – Basic Theory and Applications, Technical Report – RT DCA 01/99, p. 95.

Identification and Control Using Direction Basis Function Neural Network

Mahdi Jalili-Kharaajoo

Young Researchers Club, Azad University, Tehran, Iran
mahdijalili@ece.ut.ac.ir

Abstract. In this paper, adaptive identification and control of nonlinear dynamical systems are investigated using Two Synaptic Weight Neural Networks (TSWNN). The identification algorithm has the properties of rapid convergence and persistent adaptability that make it suitable for real-time control. A nonlinear example is simulated to demonstrate the effectiveness of the identification and control algorithms.

1 Introduction

Similar to the Multilayer Feedforward Neural Networks (MFNN), Two Synaptic Weight Neural Networks (TSWNN) possesses the capacity of universally approximating nonlinear multi-variable functions. An n-means clustering technique was regarded as a better method for updating the kernels by Feng Cao [1]. For on-line and adaptive applications of neural network model, some kinds of recursive identification algorithms are naturally required [2-5].

In this paper, the kernels and weights of the TSWNN are also updated simultaneously. A novel approach, however, is presented to train the TSWNN. Specifically, an Adaptive Fuzzy Generalized Learning Vector Quantization (AFGLVQ) technique is adopted to update the kernels, its weights and recursive least squares algorithm with variable forgetting factor (VRLS) is used to estimate the weights.

2 Two Synaptic Weight Neural Networks for Modeling Nonlinear System

Many single-input single-output non-linear systems can be described as

$$y(t) = f_s(y(t-1),...,y(t-n_y),u(t-1),...,u(t-n_u)) \tag{1}$$

where *y(t)* and *u(t)* are the system output and input, respectively; n_y and n_u the lags of the output and input respectively; and $f_s(.)$ some non-linear function. The TSWNN is a two-layer processing structure. In the first layer, the neurons are represented by TSWNN with the kernels c_i, which are interconnected by weights w_{ij}, w'_{ji}. Factor *s, p* is chosen. The second layer is essentially a linear combiner.

The overall response of such a network is a mapping $f_r : R^m \to R$, that is

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 708–712, 2004.
© Springer-Verlag Berlin Heidelberg 2004

$$f_r(x) = \sum_{i=1}^{n} \theta_i \phi(x, c_i, w_i) \tag{2}$$

where $x \in R^m$, m is network input vector, $c_i \in R^M, 1 \le i \le n$ are the kernels and $\phi(.): R \to R$. θ_i is the connection weights; and n is the number of neurons. When the TSWNN is used to approximate the dynamical system (1), define $m = ny + nu$ and let

$$x(t) = [y(t-1),..., y(t-n_y); u(t-1),..., u(t-n_u)] \tag{3}$$

Then the TSWNN output

$$\hat{y}(t) = f_r(x(t)) \tag{4}$$

acts as the estimator of the $y(t)$. Let

$$e(t) = \hat{y}(t) - y(t) \tag{5}$$

Hence, the goal of training the TSWNN is to make $e(t)$ as small as possible. In this paper, the function is chosen as function

$$\phi(x, c_i, w_i) = \cos\left[\sum_{j=1}^{m}\left(\frac{w_j}{|w_j|}\right)^s |w_j(X_j - c_j)|^p - \theta\right] \tag{6}$$

3 Hybrid AFGLVQ and VRLS Algorithm

3.1 Adaptive Fuzzy Generalized Learning Vector Quantization (FGLVQ)

3.1.1 FGLVQ

In the GLVQ algorithm, for the winner node $i(i=arg\ min|\ x(t)-c_j(t)|,\ 1< i< n)$, the updating rules of the kernels and its weights are

$$c_i(t+1) = c_i(t) + b(t)\left[\frac{D^2 - D + \|x(t)-c_i(t)\|}{D^2}\right](x(t)-c_i(t)), w_i(t+1) = w_i(t) + \eta_o \|x(t)-c_i(t)\|(x(t)-c_i(t)) \tag{7}$$

For the rest nodes j $(j=1,..., n,\ j \neq i)$, the rules are

$$c_j(t+1) = c_j(t) + b(t)\left[\frac{D^2 - D + \|x(t)-c_j(t)\|}{D^2}\right](x(t)-c_j(t)), w_j(t+1) = w_j(t) + \eta_o \|x(t)-c_j(t)\|(x(t)-c_j(t)) \tag{8}$$

where $D = \sum_{i=1}^{n} \|x(t)-c_i(t)\|$, η_o, η_1 1 is proportion factor; $b(t)$ is the learning rate which is chosen satisfy the two conditions: As $t \to \infty$; $b(t) \to 0$ and $b(t) \to \infty$. $b(t)$ can be taken as

$$b(t) = b(t-1)/\sqrt{1+t/n} \tag{9}$$

The algorithm can give more reasonable solution when $D>1$. But it has been found that GLVQ behaves exactly opposite to what was desired when $D<1$. To overcome this problem, we can give a fuzzy modification of GLVQ. Let L_x be a loss function which measures the locally weighted mismatch (error) with respect to the winner

$$L_x = L(x, c_1, c_2,..., c_n) = \sum_{j=1}^{n} \mu_j \|x - c_j\|^2 \tag{10}$$

where μ_j is chosen to be a fuzzy member function as

$$\mu_j = \left(\sum_{p=1}^{n} \frac{\|x - c_j\|^2}{\|x - c_p\|^2} \right)^{-1} \tag{11}$$

Assume $\|x - c_p\| > 0$, the gradient of L_x with respect to c_p can be calculated as

$$\nabla_{c_p} L = -2n\mu_p^2 (x - c_p) \tag{12}$$

Thus, the updating equation of the kernels can be rewritten as

$$c_j(t+1) = c_j(t) + 2b(t)n\mu_j^2 \left(x(t) - c_j(t) \right) \tag{13}$$

3.1.2 Adaptive Modification of FGLVQ

In adaptive identification and control, since a plant is unknown and the structure of a given TSWNN is limited, therefore the modeling error exists inevitably. Here we give an updating equation of *b(t)*

$$b(t) = b_o \|\varepsilon(t)\| / \left(1 + \|\varepsilon(t)\|\right) \tag{14}$$

where $0 < b_o < 1$, thus $0 < b(t) < 1$. $b(t)$ varies with $\varepsilon(t)$. Of course, when $\varepsilon(t)$ converges to zero, *b(t)* will tend to zero too.

3.2 Recursive Least Squares with Variable Forgetting Factor

Define the hidden layer output vector at the instant *t* as

$$\Phi_t = [\phi_1(t), \phi_2(t), ..., \phi_n(t)]^T \tag{15}$$

The connection weight vector at *t* as

$$\Theta_t = [\theta_1(t), \theta_2(t), ..., \theta_n(t)]^T \tag{16}$$

Thus, Θ_t can be recursively updated by the equations

$$\Theta_{t+1} = \Theta_t + K_t \varepsilon(t), \quad K_t = \frac{P_t \Phi_t}{1 + \Phi_t^T P \Phi_t}, \quad P_{t+1} = (I - K_t \Phi_t^T) P_t / \rho_t, \quad \rho_t = 1 - (1 + \Phi_t^T K_t) \varepsilon_t^2 / \sum_o \tag{17}$$

If $\rho_t < \rho_{\min}$, set $\rho_t = \rho_{\min}$; $\sum_o > 0$ reflects the amplitude of the noise.

4 Adaptive Control Law Optimization

Define the predictive control error

$$e(t+1) = \hat{y}(t+1) - [\alpha y(t) + (1 - \alpha) y_r(t+1)] \tag{18}$$

where $y_r(t+1)$ is the reference output. Let the initial value of *u(t)* be

$$\mu_o(t) = \mu(t-1) \tag{19}$$

and the corresponding output of the TSWNN be

$$\hat{y}_o(t+1) = f_r(x_o(t+1)); \quad x_o(t+1) = [y(t), ..., y(t - n_y + 1); u(t), ..., u(t - n_u + 1)] \tag{20}$$

Similarly, $u_k(t)$ denotes the control value. The corresponding output of the TSWNN is

$$\hat{y}_k(t+1) = f_r(x_k(t+1)); \quad x_k(t+1) = [y(t), ... y(t - n_y + 1); u_k(t), ... u_k(t - n_u + 1)]; \quad e_k(t+1) = \hat{y}(t+1) - [\alpha y(t) + (1 - \alpha) y_r(t+1)] \tag{21}$$

The optimization procedure is described as follows.

- Let $k_{\max}$ be the maximum iteration number and $e_{\max}$ be the desired control accuracy. Set the initial Hessian matrix $H_o = h_o I, 0 < h_o < 1, k = 0$
- Let $k=k+1$. $u(t)$ at the k iteration is calculated by

$$\mu_k(t) = \mu_{k-1}(t) - H_k \nabla \hat{y}_{k-1} e_{k-1}, \quad H_k = (\lambda_c H_{k-1}^{-1} + \nabla^T \hat{y}_{k-1} + \alpha_c I)^{-1} \tag{22}$$

where $0 < \lambda_c < 1, 0 < \alpha_c < 1$; $\nabla \hat{y}_{k-1}$ is the gradient of $y_{k-1}(t+1)$ with respect to $u_{k-1}(t)$.

- If $k = k_{\max}$ or $e_k(t+1) = e_{\max}$, then stop calculating; Otherwise, go to the previous step. In the DLS algorithm, the updating equation for H_k is

$$H_k = (\lambda_c H_{k-1}^{-1} + \nabla \hat{y}_{k-1} \nabla^T \hat{y}_{k-1})^{-1} \tag{23}$$

The item $\alpha_c I$ of the equation (22) is not included in the equation (23). It has been found that when $\hat{y}_{k-1}$ is small and $0 < \lambda_c < 10$, the H_k will become divergent infinitely as k increases.

5 Simulation Results

Consider $y(k) = \dfrac{y(k-10)}{1+y^2(k-1)} + u^3(k-1)$, $m = 7$, $h_o = 0.09, b_o = 0.21, \lambda_c = 0.001, \varepsilon(t) = e^{-800t}$

In Fig. 1 the problem of set point tracking of the closed-loop system is shown, while Fig. 2 indicates growth pattern (number of rules).

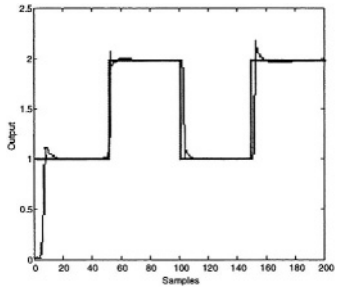


Fig. 1. Set point trajectory and output

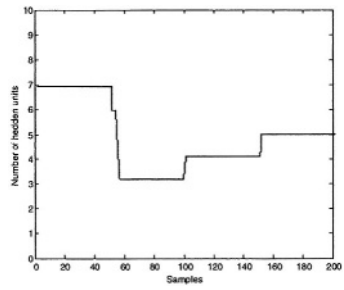


Fig. 2. Growth pattern (number of rules)

6 Conclusion

In this paper, firstly, we proposed a novel approach to train the TSWNN. An AFGLVQ technique was adopted to adjust the kernels of the TSWNN, and VRLS was applied to update the connection weights of the network. Secondly, on the basis of the one-step ahead TSWNN predictor, a numerically stable Davidon's least squares-based minimization approach was used to optimize the control law iteratively in each sample period. The simulations demonstrated the rapid learning and adaptive property of the identification algorithm and the effectiveness of control algorithm.

References

1. Cao S.G., Rees N.W., Feng G. Analysis and design for a complex control systems, part I: fuzzy modeling and identification. *Automatica,* 6(8), pp. 1017-1028, 1997.
2. Yingwei Lu and P. Saratchandran, Identification of time-varying nonlinear systems using minimal radial basis function neural networks, *IEE Proc.: Cont. Theory and App.*144(2), pp. 202-208, 1997.
3. Fabri S., and Kadirkamanatham V., Dynamic Structure Neural Networks for Stable Adaptive Control of Nonlinear Systems. *IEEE Transaction on Neural Networks,* 7(5), pp.1151-1167, 1996.
4. Jeffrey T.S. and Kevin M. Passino, Stable Adaptive Control Using Fuzzy Systems and Neural Networks, IEEE Transactions on Fuzzy Systems 4(3), pp.339-359, 1996.
5. Sanner R.M. and Jean-Jacques E. Slotine, Gaussian Networks for Direct Adaptive Control, *IEEE Transaction on Neural Networks,* 3(6), pp.837-867, 1992.

A New Dynamic Structure Neural Network for Control of Nonlinear Systems

Mahdi Jalili-Kharaajoo

Young Researchers Club, Azad University, Tehran, Iran
mahdijalili@ece.ut.ac.ir

Abstract. In this paper, Dynamic structure neural network controller based on feedback linearization is proposed. The proposed method can adapt the neural network structure dynamically while it can guarantee the stability and tracking precision of system.

1 Introduction

Neural network provides a nice form for describing the nonlinear characteristic of systems in certain degree and it has the vast vistas in the control of nonlinear systems [1,2,3]. There is a strong appeal to the research field of nonlinear control theory and neural network control if combining neural network with nonlinear adaptive control [4,5]. But there is the short of systematic study on the stability and robust property for the nonlinear neural network control system so far. In this note, we will propose an efficient approach to combine the dynamic structure neural network algorithm with adaptive feedback linearization. The algorithm of dynamic structure neural network controller is given and the stability of system is analyzed. It is satisfactory with the results of system simulation.

2 Dynamic Structure Neural Network Controller

Supposing the nonlinear system (SISO) is described

$$\begin{aligned} \dot{X} &= f(X) + g(X)u \\ y_p &= h(X) \end{aligned} \tag{1}$$

Where $X \in R^n$, y_p and u are the state, output and control input variables respectively. If the order and relative degree of system (1) are n and r, then we let

$$u = \frac{1}{g_1(X)}(-f_1(X) + v) \tag{2}$$

Where v is pseudo-input, $f_1 = L_f^r h(X), g_1 = L_g L_f^{r-1} h(X)$. (L is Lee derivative operator)[6]. The original system can be changed into linear system and

$$y^r = v \tag{3}$$

Generally, the objective of control system for (1) is to find the control law u so that the outputs of system follow the reference trajectory y_m. The relative degree of reference model n is greater than r. For minimum-phase system, we let

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 713–716, 2004.
© Springer-Verlag Berlin Heidelberg 2004

$$v = y_m^r - \alpha_r e^{r-1} - ... - \alpha_1 e \tag{4}$$

where $e = y_p - y_m$ and $\alpha_r, \alpha_{r-1}, ..., \alpha_1$ satisfy $\Gamma(s) = (s^r + \alpha_r s^{r-1} + ... + \alpha_r)$ is Hurwitz polynomial. In order to guarantee the stability and tracking precision of system, we take the control law

$$u = u_{al} + u_{sl}, \; u_{al} = \frac{1}{\hat{g}(X)}(-\hat{f}(x) + v) \tag{5}$$

and u_{sl} is sliding controller.

f(X), g(X) are the approaches of $f_1(X), g_1(X)$ respectively. Using the method of [1], we let

$$\hat{f}(X) = m(X)\hat{W}_f^T \Phi_f(X) + f_o(X), \hat{g}(X) = m(X)\hat{W}_g^T \Phi_g(X) + g_o(X) \tag{6}$$

where $\Phi_f(X)$ and $\Phi_g(X)$ are the output of hidden layers of neural network and $\hat{W}_f$, $\hat{W}_g$ are the parameters of corresponding output layers.

Using the method of feedback linearization, the state locus of system is in the sub-space A_d of the state space. It can be decided by reference model. Neural network is only employed to approach the characteristic of system in subspace A_d. To ensure safety we expand the approximation area to *A* where $A_d \subset A$. The *m(x)* expresses the switch function to control neural network so that neural network only approaches the characteristic of system in subspace *A*. It is defined as

$$m(X) = \begin{cases} 1 & X \in A_d \\ 0-1 & X \in (A - A_d) \\ 0 & X \notin A_d \end{cases} \tag{7}$$

The f_o, g_o are the known characteristic of f_1, g_1 receptively.

According to the approaching theory of neural network, we have:

Supposition (1) for any $\varepsilon_f, \varepsilon_g \in R^+$ and f_o, g_o there exist W_f^*, W_g^* and *m(X)* so that for any $x \in A$ in (6) satisfy

$$|\Delta f| = \left|f^*(X) - f_1(X)\right| \le \varepsilon_f, |\Delta g| = \left|g^*(X) - g_1(X)\right| \le \varepsilon_g \tag{8}$$

where

$$f^* = \hat{f}(W_f^*, X), \Delta f = f^*(X) - f_1(X), g^* = \hat{g}(W_g^*, X), \Delta g = g^*(X) - g_1(X) \tag{9}$$

The neurons in dynamic structure neural network can be regarded as to be chosen from the set of candidate neural neurons. The set of all candidate neurons is *Kpot* which is determined according to the properties of hidden layer neurons and the property of system. The neurons in *Kpot* can be divided into three sets, 1) inactive neurons *Kp*; 2) active neurons to be removed from neural network *Kd;* 3) active neurons to be kept in the neural networks *Ka*. Then, we have

$$f(X) = mW_f^* \Phi_f - \Delta f + f_o = m\begin{bmatrix} W_{fa}^* \\ W_{fd}^* \\ W_{fp}^* \end{bmatrix}^T \begin{bmatrix} \Phi_{fa} \\ \Phi_{fd} \\ \Phi_{fp} \end{bmatrix} - \Delta f + f_o, \; g(X) = mW_g^* \Phi_g - \Delta g + g_o = m\begin{bmatrix} W_{ga}^* \\ W_{gd}^* \\ W_{gp}^* \end{bmatrix}^T \begin{bmatrix} \Phi_{ga} \\ \Phi_{gd} \\ \Phi_{gp} \end{bmatrix} - \Delta g + g_o \tag{10}$$

Let

$$e_1 = \beta_r e^{r-1} + \beta_{r-1} e^{r-2} + ... + \beta_1 \tag{11}$$

where $\beta_r,...,\beta_1$ satisfy $\varphi(s)=\beta_r s^{r-1}+\beta_{r-1}s^{r-2}+...+\beta_1$ is Hurwitz polynomial and $\varphi(s)/\Gamma(s)$ is strict positive real function.

We can choose $\varphi(s),\Gamma(s)$ such that

$$\varphi(s)/\Gamma(s)=\frac{1}{s+k_d},k_d>0$$

In order to avoid the parameters drift, increase the robustness of the system, and add dead-zone to e_1, Let

$$e_{1\Delta}=e_1-\phi sat(\frac{e_1}{\phi}) \tag{12}$$

and take the sliding mode control as

$$u_{sl}=-k_{sl}sat(\frac{e_1}{\phi}) \tag{13}$$

The adaptive rate of neural network will be taken as

$$\frac{d\hat{W}_{fa}}{dt}=\eta_f me_{1\Delta}\Phi_{fa},\frac{d\hat{W}_{fd}}{dt}=\eta_f m|e_{1\Delta}|[-\mathrm{sgn}(\hat{W}_{fd})],\frac{d\hat{W}_{fp}}{dt}=0,\frac{d\hat{W}_{ga}}{dt}=\eta_g me_{1\Delta}\Phi_{ga}$$

$$\frac{d\hat{W}_{gd}}{dt}=\eta_g m|e_{1\Delta}|[-\mathrm{sgn}(\hat{W}_{gd})],\frac{d\hat{W}_{gp}}{dt}=0 \tag{14}$$

3 Simulation Results

Suppose that a nonlinear system is described as

$$\begin{aligned}\dot{x}_1&=f(X)+g(X)u\\ \dot{x}_2&=x_1-x_2\\ y_p&=x_1\end{aligned},\quad\begin{cases}X=\begin{bmatrix}x_1\\x_2\end{bmatrix}\\ f(X)=-4\frac{\sin(4\pi x_1)}{(4\pi x_1)}\frac{(\sin(4\pi x_2))^2}{(4\pi x_2)^2},\quad g(X)=2+\cos(7x_1x_2)\end{cases} \tag{15}$$

and system order n=2, relative degree r=1.

The reference model is $y_d=1/(1+s/10)^3$ and its input is the square wave with *0.4* Hz frequency and *0.9* amplitude. Let $\beta_1=1,\alpha_r=k_d=5,\phi=0.005$.

From the reference model, we can know that the state locus of system is in the region of [-1,1]*[-1,1]. So, we can take A_d =[−1,1]×[−1,1],A=[−1.5,1.5]x[−1.5,1.5].

In system simulation, the neural network controller based on dynamic structure wavelet network is employed. The *2-d* mother wavelet $\psi(X)=(2-\|X\|^2)\exp(-\|X\|^2/2)$. And the corresponding *2-D* wavelet frame is $\psi_{p,K}(X)=a_o^{-pn/2}\psi(a_o^{-p}X-diag(K)b_o)$.

Where $p\in Z,K\in Z^2$ are the scale and translation coefficients. And $a_o=2,b_o=1$, $p\in[-1,4]$. The range of K is chosen so that $a_o{}^p b_o K\in A$.

The adaptive rate of neural network $\eta_f=\eta_g=25$. The structure adaptive law coefficients are $u_f=1,u_g=5,\theta_f=3,\ \theta_g=15,f_o=0,g_o=1.9,g_l=0.895$. The sliding mode controller is $k_{sl}=0.5$ and the initial states of system are $X(0)=[1,1]^T$.

The simulated results are given in Figs. 1 and 2. It shows that the neural network structure is dynamic and tending towards stable and the tracking error of system is reaching the required zone after 10 seconds. The neural network sizes for approximating *f(X), g(X)*are 40 and 8. While, *f(X)* and *g(X)* are approached using *1403* and *556* RBF neurons in [3] and [5] respectively. It is illustrated that the size of neural network is decreased greatly using the dynamic structure wavelet network.

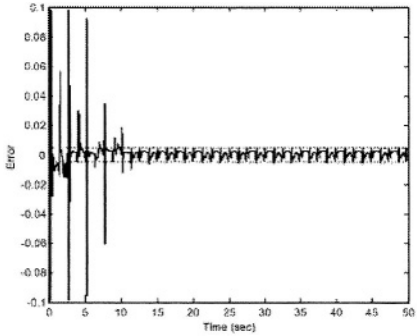


Fig. 1. Tracking Error; the dash line is the upper bound of tracking error.

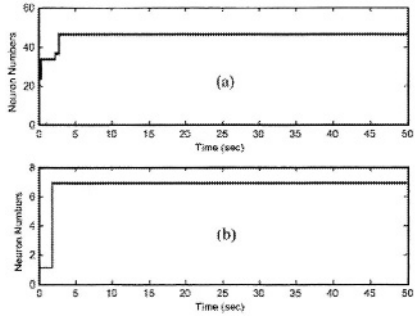


Fig. 2. The sizes of neural network for approximating *f*(x), *g(x),* (a) Neural Network for *f*(x), (b) Neural Network for *g*(x).

References

1. Yingwei Lu and P. Saratchandran, Identification of time-varying nonlinear systems using minimal radial basis function neural networks, *IEE Proc.: Con. Theory and Appl.* 144(2), pp. 202-208, 1997.
2. Cannon, M., Slotine, Jean-Jacques E., Space frequency Localized Basis Function Networks for Nonlinear System Estimation and Control, *Neurocomputing,* 9(3), pp.293-342, 1995.
3. Fabri S., and Kadirkamanatham V., Dynamic Structure Neural Networks for Stable Adaptive Control of Nonlinear Systems. *IEEE Transaction on Neural Networks, 7*(5), pp.1151-1167, 1996.
4. Jeffrey T.S. and Kevin M. Passino, Stable Adaptive Control Using Fuzzy Systems and Neural Networks, *IEEE Transactions on Fuzzy Systems* 4(3), pp.339-359, 1996.
5. Sanner R.M. and Jean-Jacques E. Slotine, Gaussian Networks for Direct Adaptive Control, *IEEE Transaction on Neural Networks,* 3(6), pp.837-867, 1992.
6. Khalili ,H. *Nonlinear systems,* John Wiley &Sons. NY, 1996.
7. Sastry S. and Alberto Isidori, Adaptive Control of Linearzable System, *IEEE Transaction on Automatic Control,* 34(11), pp.2247-2251, 1989.
8. Yesildirek A., Feedback Linearization Using Neural Networks, Automatica., 31(11), pp.1659-1664, 1995.

Proposing a New Learning Algorithm to Improve Fault Tolerance of Neural Networks

Mahdi Jalili-Kharaajoo

Young researchers Club, Azad University, Tehran, Iran
mahdijalili@ece.ut.ac.ir

Abstract. Fault tolerant neural network architecture, based on Multilayer Perceptron (MPL) is presented. We modify the conventional Back error propagation (BP) algorithm to be applied to this architecture with the least learning degradation for fault tolerant nodes. Simulation results for random s-a-0 faults demonstrating the fault tolerance improvement are presented.

1 Introduction

Neural Networks have been successfully used for fault diagnosis in nonlinear systems [1]. Using conventional fault tolerant techniques, such as Triple Modular Redundancy (TMR) and Triple Time Redundancy (TTR) [2], yields to either a very expensive and large system or a long time overhead. Two main approaches have been proposed to improve fault tolerance in an artificial neural network: 1) working on learning algorithms and 2) working on architectures [3,4].

In this paper we will first introduce fault tolerant neural network architecture, based on Multilayer Perceptron (MLP) and a new learning algorithm based on conventional Back error Propagation (BP) algorithm.

2 MLP and BP Algorithm

MLP consists of several cascaded layers of neurons with sigmoid activation functions [5]. The input vector, feeds into each of the first layer neurons, the outputs of this layer feed into each of the second layer neurons and so on, as in Fig. 1.

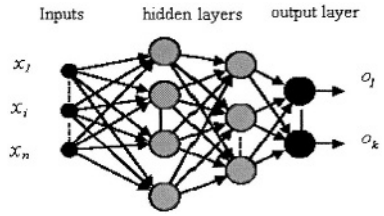


Fig. 1. Architecture of a typical MLP

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 717–721, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Most often the nodes are fully connected, i.e., every node in layer l is connected to every node in layer $l+1$. In this paper, we assume input vector as the first layer in the neural network. Usually output neurons use linear activation functions rather than nonlinear sigmoid, since this tends to make learning easier. We assume that the number of hidden layers is one and activation function of each neuron in hidden layer is a bipolar sigmoid by the following equation

$$f(u_i)=\frac{1-\exp(u_i)}{1+\exp(u_i)};\quad u_i=\sum_j w_{ij\times}x_i-\theta_i \tag{1}$$

where w_{ij} is the connection weight between neuron j in the preceding layer and neuron i. BP algorithm changes w_{ij} in order to reduce the error of output layer defined by

$$E=\frac{1}{2}\sum_i (t_i-o_i)^2 \tag{2}$$

where t_j is output target and o_j is the estimated output [7]. Using the steepest-descent gradient rule, the change of w_{ij} is expressed as

$$\Delta w_{ij}=\eta\frac{\partial E}{\partial w_{ij}} \tag{3}$$

Selecting a suitable value plays an important role in network learning convergence. According to BP algorithm

$$\Delta w_{ij}=\eta\delta_i^P o_j^P,\ \delta_i^P=(t_i^P-o_i^P).f'(u_i^P),\ \delta_i^P=(\sum_k w_{ki}.\delta_k^P).f'(u_i);\ f'(x)=2f(x).(1-f(x)) \tag{4}$$

3 Fault Model

There are usually three kinds of faults, considered in a neural network: 1) connection, 2) weights and 3) neuron body itself. The first two faults are often modeled as stuck-at-0 and most often occur during a memory disappearance or a link disconnection in VLSI. On the other hand faults due to a neuron cell, usually subject its output to one of the positive or negative saturation voltages. This kind of faults is modeled as stuck-at-(1) or stuck-at-(-1). In this paper we consider only stuck-at-0 (s-a-0) faults.

4 FTNN Architecture

It is evident that a simple system has naturally more fault tolerance capabilities than a large and complex one. Table 1. summarizes the results of injecting 1000 random s-a-0 fault patterns to links of a sample MLP, consisting of 2 inputs, 15 neurons in hidden layer and one output node. It is clear that the links in hidden layer have tolerated faults better than links in output layer, which are sensitive and can cause remarkable errors. So, replacing the conventional output nodes with nonfaulty nodes will improve significantly the total fault behavior in the MLP.

Table 1. Average absolute errors due to 1000 random stuck-at-0 faults in a 2-15-1 MLP.

Condition	No faults	Fault in hidden layer	Fault in output layer
Average Error	0.006	0.026	0.058

To have such a robust node we first use suggested linear activation function, which is a simple adder indeed, rather than the nonlinear sigmoid function. Then, we use a wired connection as links and eliminate memory usage by the elimination of connection weights. Fig. 2 shows our suggestion to obtain such nodes in two steps, while preserving the learning ability of neural network in output nodes, as much as possible. We call the resultant node as FTN (Fault Tolerant Neuron).

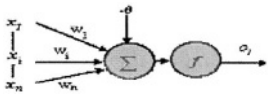


Fig. 2(a). A conventional nonlinear node

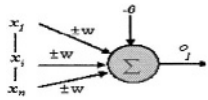


Fig. 2(b). A linear node with similar weights

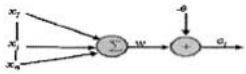


Fig. 2(c). Fault Tolerant node with wired links

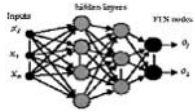


Fig. 2(d). An FTNN architecture using FTN nodes in output layer (shadowed nodes).

5 UWLA Learning Algorithm

According to the architecture described in previous section, a learning algorithm is introduced for output layer, which we call it as UWLA (Uniform Weight Learning Algorithm). To make this term as close as possible to zero, we add another criterion, variance of w_{ij}, to the learning process, which is defined as

$$\sigma_j = \sum_i (w_{ij} - m_j)^2, \quad m_j = \frac{1}{n}\sum_i w_{ij} \tag{5}$$

where j and i denote the neurons in output and hidden layer, respectively. m_j is the mean value of w_{ij} for output neuron j that is calculated after each training step. Our aim is to diminish sj as much as possible. The following algorithm is suggested:

UWLA Algorithm:
Step 0: Initialize weights with small values,
Step 1: While stopping condition is false do steps 2-9,
Step 2: For each input vector do Steps 3-8,
Step 3: Each input unit receives input signal and broadcast it to hidden layer units,
Step 4: Each hidden unit sums its weighted input signal and applies its activation function according to equations (1),
Step 5: Each output unit sums over its all input signals and produces its output, too.
Step 6:
Step 6-1: For each output unit compute its error information term, Calculate mean[wij (old)],
Calculate: w_{ij}(new)= w_{ij}(old)+• w_{ij}. •••••• using equation (4).

If abs[w_{ij}(new)-mean [w_{ij}(old)]] is greater than abs[w_{ij}(old) -mean [w_{ij}(old)]], then choose = $_2$,which $_2$< $_1$.

Step 7: for each hidden unit compute its error information term, using equation (4),

Step 8: For each output and hidden unit update weights according to: w_{ij}(new)= w_{ij}(old)+• w_{ij}. ,

Step 9: Test stopping condition.

6 FTNN Behavior

To evaluate the UWLA algorithm, an experiment is managed in which an MLP network with two inputs, fifteen hidden nodes and one output node is to approximate the nonlinear function of $z = 0.5 \times Sin(x+y)+0.3$

For comparison, we consider standard BP and MFTA (Multiple Fault Training Algorithm) algorithm introduced by [4], for single s-at-0 faults. In all simulations the training process lasts after 20000 iterations, with a decreasing learning rate of = 0.2 to = 0.95. Tabs. 2 and 3 summarize the results obtained after 1000 injection of random s-a-0 faults in hidden layer and output layer weights. It is clear that FTNN trained through UWLA has slightly improved fault tolerance in the hidden layer (first layer) as compared to MFTA. Figs. 3 and 4 show the results for 1000 random s-a-0 fault injections for the trained networks. Clearly, Fault tolerance performance of FTNN is twice the MFTA and approximately four times the standard BP algorithm.

Table 2. Average absolute errors due to 1000 random s-at-0 faults in hidden layer of a 2-15-1 network.

Fault numbers	No faults	One fault	Two faults
Standard BP	0.006	0.026	0.044
MFTA	0.009	0.019	0.028
FTNN	0.007	0.015	0.026

Table 3. Average absolute errors due to 1000 random s-at-0 faults in output layer of a 2-15-1 network.

Fault numbers	No faults	One fault	Two faults
Standard BP	0.006	0.058	0.090
MFTA	0.009	0.021	0.038
FTNN	0.008	0.022	0.039

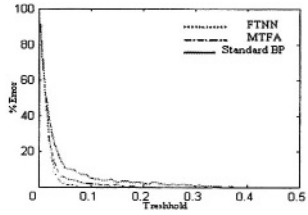


Fig. 3. Error percentage versus threshold value after 1000 random s-at-0 faults in hidden layer.

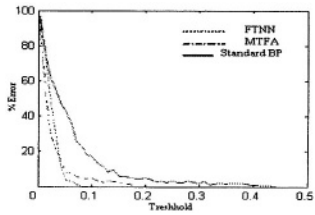


Fig. 4. Error percentage versus threshold value after 1000 random s-at-0 faults in output layer.

7 Conclusion

In this paper, we developed FTNN architecture by introducing FTN nodes and implementing them in output layer, which are the critical nodes in MLP networks. The proposed fault tolerant node is subjected to have the least memory usage. Then, we introduced UWLA, to extend BP. Simulations demonstrate that the resultant network shows a superior performance over the standard BP and commonly used fault injection training algorithms such as MFTA.

References

1. Khunasaraphan C., K.Vanapipat and C. Lursinsap, Weight Shifting Techniques for Self-Recovery Neural Networks, *IEEE Tran. Neural Networks* Vol 5, No. 4, July 1994.
2. Johnson B.W., *Design and Analysis of Fault Tolerant Digital Systems,* Addison Wesley, 1989.
3. Murrayand A.F., Peter J. Edwards, Synaptic Weight Noise during Multilayer Perceptron Training: Fault Tolerance and Training Improvement, *IEEE Tran. Neural Networks,* Vol 4, No.4, July 1993.
4. Tai-Chiu C., K. Mehrotra, Chiukuri K. Mohan and S. Ranka, Training Techniques to Obtain Fault Tolerant Neural Networks, *Proc. 24th international symposium on fault tolerant computing,* 1994.
5. Fausell L., *Fundamental of Neural Networks,* Prentice Hall, 1994.
6. Hush and Home, Progress in Supervised Neural Networks, *IEEE Signal Proc. Mag.,* Jan 1993.
7. Beale R. and T. Jackson, *Neural Computing: An introduction,* York University, IOP Pub., 1990.

Nonlinear Parametric Model Identification and Model Based Control of *S. cerevisiae* Production

Bulent Akay

Ankara University, Faculty of Engineering, Department of Chemical Engineering,
06100 Tandogan,Ankara,Turkey
bakay@eng.ankara.edu.tr

Abstract. This work shows the application of Parametric Model Identification method which is Recursive Least Square for the estimation of nonlinear model parameters. The model of the bioreactor dynamics was assumed in the form of regression model. DMC (Dynamic Matrix Control) and PID Control systems of *S.cerevisiae* yeast production in a batch bioreactor have been investigated experimentally to achieve the optimal operating condition. The performance of these control algorithms was compared with each other.

1 Introduction

In many cases the processes are so complex that it is not possible to obtain reasonable models using only physical insight. In such cases one is forced to use identification techniques. However, this technique needs an appropriate model structure.

DMC has provided researchers with a powerful tool for dynamic optimization and control for process industry. These control systems employing linear models include DMC [1] and simplified MPC. Recently there has been an increasing interest in the use of DMC for controlling plant-wide control [2,3].

In this study, a bioreactor which produces *S.cerevisiae* is modelled by parametric nonlinear models. Several identification techniques were used to define the model parameters. Comparison general nonlinear NARMAX model and experimental data from related system was demonstrated. Identified model was used to find the system matrix for DMC control.

2 Nonlinear Model Parametric Identification

A polynomial NARMAX model was constructed for design of nonlinear controller system . The general, discrete-time NARMAX Model given by eq. (1):

$$y(k+1) = f(y(k),.....,y(k-n_y),U(k),.....,U(k-n_u),e(k),....,e(k-n_c)) \quad (1)$$

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 722–725, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Square wave signal has been used in the identification of nonlinear system. The model parameters have been evaluated by using Recursive Least Square Method. NARMAX model which is used in this study was given below:

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) + b_0 u^{0.78}(k-1) + e(k) \tag{2}$$

Model parameters were calculated as $a_1 = -0.5408$, $a_2 = -0.4641$, $b_0 = 0.0339$.

3 Model Based Control

In DMC, the manipulated variable is adjusted previously and the controlled variable is influenced by these adjustments as well as disturbances. To minimize performance index J, the sum of the error squared is calculated

$$J = \sum_{i=1}^{NP}\left[X^{set}(k+i) - X_{new}(k+i)\right]^2 + f_t \sum_{i=1}^{NC}\left[\Delta u(k+i)\right]^2 \tag{3}$$

Where $\mathbf{X}_{new}$ is the closed-loop real response for the value at the i th step into the future and calculate by using eq. 4.

$$X_{new} = X_{past} + \underline{\underline{A}}\,\underline{\Delta u} + \underline{d} \tag{4}$$

Here NP denotes the future time over which the control performance is evaluated and termed the output horizon. NC is the number of adjustments and is given by the input horizon. NC must be less than NP.

A is the dynamic matrix that is composed of the step response coefficients. $\mathbf{X}^{set}$ is the set point and can remain constant at its current value in the future. The goal of perfect controlled-variable performance would be to have zero error for all samples in the future. The solution to this problem is a least-square solution in the form of the following control increments:

$$\underline{\Delta u} = \left[\underline{\underline{A}}^T \underline{\underline{A}} + f_t^2 \underline{\underline{I}}\right]\underline{\underline{A}}\left(X^{set} - X_{past} - \underline{d}\right) \tag{5}$$

4 Control Results

The parameters necessary for the DMC and PID algorithms were found from the response curve obtained by performing simple open-loop step tests (Figure 1). As can be seen from this figure the agreement between the response obtained from the computer simulation program and experimental data is sufficiently good. Dynamic Matrix was calculated from this reaction curve and given below.

$$A = \begin{bmatrix} 30.9 & 0 & 0 & 0 \\ 30.7 & 30.9 & 0 & 0 \\ 31.2 & 30.7 & 30.9 & 0 \\ 31.4 & 31.2 & 30.7 & 0 \end{bmatrix} \tag{6}$$

The optimum values of the coefficients of PID controller were found using Cohen-Coon Method [4] as $K_C = 68.6$, $\tau_I = 2.4$ min and $\tau_D = 0.36$ min. The process model was chosen as first order dead time process. Dead time and open-loop process time constant were found as 0.5 min and 30.5 min, respectively.

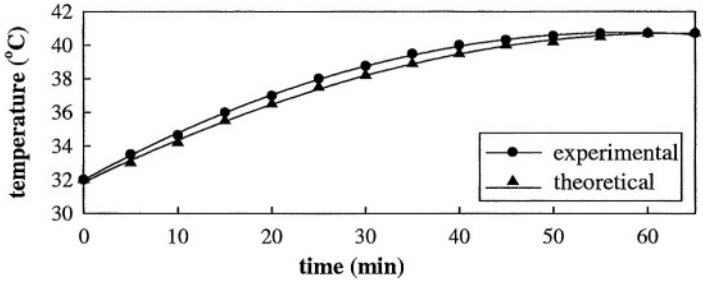


Fig. 1. Open loop temperature response

The selected DMC coefficients (NP, NC and f) were adjusted by a least-square technique applied to the error between observation and prediction. They are found as NP=4, NC=1 and f=0.1. Time variations of the bioreactor temperature controlled by under DMC system in the experimental and theoretical studies are shown in Figure 2.

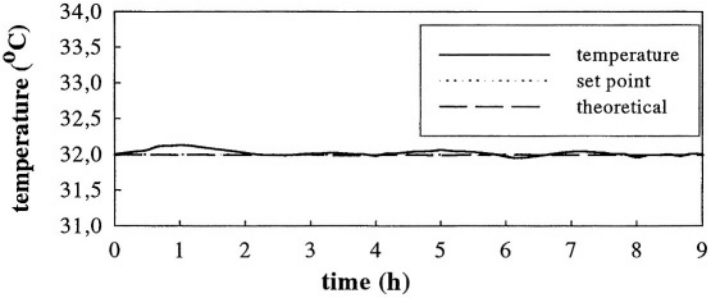


Fig. 2. Experimental and theoretical DMC of the bioreactor temperature

For the same condition, the PID control is applied to keep the bioreactor temperature at the desired set point. These experimental and theoretical time variations of the bioreactor temperature are given in Figure 3. In comparison with Figure 2, Figure 3

shows that temperature moves around the desired trajectory at a higher frequency with some offset.

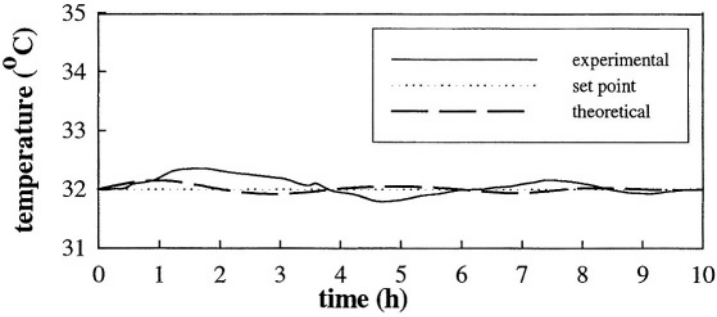


Fig. 3. Experimental and theoretical PID control of the bioreactor temperature

5 Discussion

The performance of DMC control was compared with the PID controller at the optimal set point in a batch bioreactor. This system was chosen because of its industrial importance and its nonlinearities. The oscillations of the temperature around the optimal path in the experiments were unavoidable. On the other hand, simulation results show excellent performance for DMC and PID controllers. Although there is no significant improvement in the experimental DMC results, it was seen that DMC provides a satisfactory control at least equal to a well-tuned PID controller.

Acknowledgement. This research has been supported by Ankara University Biotechnology Institute (Grant No. 29).

References

1. Cutler, C.R.: Dynamic matrix control: An optimal multivariable control algorithm with constraints, Ph.D. Thesis, University of Huston (1983)
2. Desphande, P.B. and Arulalan, G.R.: Simplified model predictive control, Ind. Eng. Chem. Res., Vol. 26., (1987) 347-357.
3. Seborg, D.E., Edgar, T.F. and Mellichamp, D.A.: Process dynamics and control, John-Wiley and Sons, New York (1989)
4. Omstead D. R.: Computer Control of Fermentation Processes CRC Press, Inc., Boca Raton, Florida (2000)

The Notion of Community in United States Computational Science Education Initiatives

Mary E. Searcy and Jill T. Richie

Appalachian State University, Department of Mathematical Sciences
Boone, NC 28608

Abstract. The computational science education community in the United States can be perceived at a variety of levels: a community of stakeholders, educators, colleagues, or learners. Each level is illustrated by examples and their roles are discussed.

1 Introduction

In the last decade, addressing international problems such as communicating SARS medical information [1] and researching microgravity combustion for international space station fire prevention [2] has required computational methods. Rapid technological growth has made it possible to meet larger challenges and resolve long-unanswered questions. As a consequence, there has been an increased pressure on United Stated academic institutions to prepare students for such a world [3]. As the number of computational science education initiatives increases, the notion of *community* – a social group sharing common characteristics or interests and perceived or perceiving itself as distinct in some respect from the larger society within which it exists [4], has become important to their development and implementation. This article explores the United States computational science education community and ways to strengthen it in the future.

2 A Community of Stakeholders

The United States Department of Energy's (DOE) Office of Science [5] suggested the following steps to help equip scientists for upcoming computational challenges: establish partnerships with computer vendors, strengthen existing high-end computing capability and networking, invest in advanced architecture, and collaborate with federal agencies and the academic community to find better ways to use existing and new capabilities. They advocate establishing connections among key computational science *stakeholders,* those individuals or agencies, with direct interest, involvement and/or investment in computational science endeavors. For example, the DOE's Scientific Discovery through Advanced Computing (SciDAC) funded projects (http://www.osti.gov/scidac/index.html) and collaborations with the United States national laboratories (http://www.scied.science.doe.gov/scied/sci_ed.htm) help support innovative partnerships between academia and other stakeholders. Also, the National Lambdarail (NLR) [6], which aims to provide a network infrastructure to

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 726–729, 2004.
© Springer-Verlag Berlin Heidelberg 2004

facilitate collaborative research, could possibly be one of the most significant computational science collaborations impacting education in the future.

There are many efforts to strengthen high-end computing. The Terascale Cluster project [7] at Virginia Tech has developed one of the fastest and most inexpensive supercomputers using currently available off-the-shelf components and working with industry partners. This effort has made complex projects accessible to local educators and researchers. For those without connections to super-computing facilities and who still want to give their students experiences with parallel processing, take an open lab of Pentium computers and boot up a pre-configured distributed computing environment with the Bootable Cluster CD (http://www.cs.uni.edu/~gray/bccd/). This CD, motivated by collaboration between educators and LinuxCare, was created solely "to facilitate instruction of parallel computing." [8]

3 A Community of Educators

We now shift our focus to the community of educators. Two major 2003 initiatives were the SC03 Supercomputing Conference Education Program (http://www.sc-conference.org/sc2003/education/edu_program.html) and the National Computational Science Institute (NCSI) (http://www.computationalscience.org/), both evaluated by the authors. Teams of educators from across the United States were chosen to spend four days learning about computational science education opportunities and technologies. The teams came from a variety of backgrounds. More than half were male (56%); two-thirds were PhD's and 20% were K-12 teachers. Although most participants had some experience with statistics, linear algebra, differential equations and numerical analysis, only 25% said they had advanced knowledge of calculus. They also had a variety of science knowledge in biology, chemistry, physics and computer science; 10% to 15% rated themselves as advanced in one or more of these fields. Approximately 75% said that they already incorporated modeling in their teaching, though that activity was a less common teaching strategy than traditional lecture (96%) or hands-on activities (97%).

Participants entered into the program with personal definitions of computational science that were as varied as their backgrounds. Here are a few quotes taken from participants' pre-survey in response to the question, "What does computational science mean to you?"

- *The science of computation using sequential and parallel programming, for solving engineering and science problems*
- *Numbers brought together to create meaning.*
- *Access to ideas, tools, data, and people.*
- *AI, representational theory, complex systems applications and theory, automated feedback to learners, adaptive testing, responsive dissemination of online resources, …*
- *It means using computers and other forms of technology to model or simulate natural systems. The model may be numeric data and equations based on the data or a visual representation using the data to generate graphics.*
- *problem solving*
- *At this point, nothing.*

Yasar and Landau [3] state that three elements have developed at the heart of defining *computational science:* a collection of computational tools, a collection of computational methods, and a problem-solving mindset that uses one discipline's information to solve another discipline's problems. Some elements can be seen in the definitions above; but there is a distinct lack of coherence among responses. Community, at this level, helps educate educators by providing an arena for exposure to new ideas and personal construct development.

The other current major program in the United States, NCSI, targets primarily university and college faculty. They address the community notion by partnering with universities, supercomputing facilities, non-profit foundations, and professional organizations. NCSI holds weeklong computational science education workshops where faculty are introduced to two major interpretations of computational science education [9]: 1) (computational science) education which emphasizes the modeling and visualization concepts, tools, and applications and how they should be used and 2) computational (science education) where the emphasis is on using visualization and modeling concepts, tools and applications to help students understand science ideas. In the summer of 2003, over 200 faculty were involved in workshops across the United States. NCSI participants were largely male (64%) and white (62%); 82% were PhD's; 13% were from community colleges; 82% were from a college or university. About 95% reported that a primary reason for attending a NCSI workshop was because they were interested in using modeling and visualization in instruction. About 80% cited the desire to network with other instructors, and 73% said that they believed attendance could help facilitate changes in their home departments. Participants were encouraged to collaborate and required to produce an educational product by the week's end. After the workshops ended, participants' short-term goals included learning more about computational science and exploring new technologies, demonstrating existing models in their courses, redesigning laboratory experiences, setting up clusters, and designing new courses or new degree programs.

4 A Community of Colleagues

There is a critical level of community missing at many institutions, a community of colleagues on a given campus who are interested in using computational science activities in their classrooms. Support from other local faculty can be a powerful resource for instructors. At a 2003 NCSI workshop, Tennessee community college faculty from three different disciplines developed a genetics model to be used in biology courses. Another institution, with six campuses outside of Atlanta, Georgia, will begin an institution-wide computational science education initiative in 2004. Their Center for Learning and Teaching (CLT) knows of "pockets" of faculty already involved in such activities and would like to see the development of a support network for the faculty that can be sustained over time.

5 Future Directions

This paper has shown there is substantial interest in computational science education; yet, the *community* at most levels is still somewhat fragmented. There are two primary

suggestions the authors offer for community development. Although the notion of computational science education *community* can be perceived at different levels, the community that truly motivates this discussion is the community of learners. As computational science opens doors to a deeper, more dynamic, understanding of our world for researchers and industry, computational science education is opening doors to a deeper, more dynamic, understanding of course content and its applications to our world for both faculty and students. The community of learners is essential to our progress as educators and must remain the focus in this rapidly changing environment. Along with keeping this critical focus, the authors also advocate an annual computational science education conference. Conferences such as ICCS 2004 that have an education component are beneficial, particularly in bringing together stakeholders. However, a focused conference might provide a catalyst for a more cohesive notion of community in the future.

References

1. Gannis, M. (June 2003). SARS grid aids Taiwan. *Access Online.* Retrieved February 17, 2004 from http://access.ncsa.uiuc.edu/Stories/SARS/
2. NASA Center for Computational Science. (2002). *NCCS highlights FY 2000: Enabling NASA earth and space sciences.* Retrieved February 17, 2004 from http://sdcd.gsfc.nasa.gov/SCB/Highlights/FY2000/microgravity.10.html
3. Yasar, O. & Landau, R. (2003). *Elements of computational science and engineering education,* SIAM Review, 45(4), 787-805.
4. *Webster's Encyclopedic Unabridged Dictionary of the English Language.* (1989). New York, New York: Portland House.
5. U.S. Department of Energy Office of Science. (2002). *The challenge and promise of scientific computing.* Retrieved February 17, 2004 from http://www.science.doe.gov/Sub/Occasional_Papers/Occasional_Papers.htm
6. The National Lambdarail, (2003). Retrieved February 17, 2004 from http://www.nationallambdarail.org/
7. Terascale cluster press release – computing.vt.edu. (2003). Retrieved February 17, 2004 from http://computing.vt.edu/research_computing/terascale/pressrelease.html
8. Gray, P. (November, 2003), *Implementing computational science in the [computer science] classroom.* Paper presented at the SC2003 Education Program, Phoenix, AZ.
9. Panoff, R. (July 2003). *Modeling as an interactive learning environment: (Beyond PowerPoint-lessness).* Paper presented at a National Computational Science Institute Workshop, Champaign, IL.

This page intentionally left blank

Author Index

Abad, A. IV-327
Abawajy, J.H. I-27,III-66, III-339
Abdalhaq, B. II-220
Abe, S. III-1046
Abrahamyan, L. IV-548
Abramson, D. I-148,I-164
Acacio, M.E. I-541
Acosta-Elías, J. I-561,I-615
Adamiak, R.W. III-1013
Adelmann, A. II-459
Adriaansen, T. III-905
Ahn, J. I-192
Ahn, S. I-349
Ahn, T.-C. II-188
Akay, B. II-722
Akçelik, V. III-717
Akdim, B. II-260
Akker, H.E.A. Van den IV-637
Al-Badarneh, A. II-606
Albuquerque, P. IV-540
Alemani, D. IV-540
Alexandrov, V.N. IV-475, IV-491
Alique, Á. I-523
Alonso, J.M. I-482
Altas, I. IV-364
Amaral, J.F.M. III-1273
Amaral, J.L.M. III-1273
Anai, H. IV-286
Anderluh, J. IV-851
Angelova, D. IV-531
Antonio, A. de I-433,I-545, III-575, III-1004, IV-252
Antoulas, A. III-740
Araújo, S.G. III-1265
Ardaiz, O. I-67
Arenas, A. III-1078
Arodź, T. I-462
Artoli, A.M. IV-548
Asensio, M.I. II-586
Asensio-Pérez, J.I. I-495
Asmussen, S. IV-795
Astalos, J. I-124
Austin, A. IV-1165
Austin, K. III-1054
Babuška, I. III-756
Bacak, G. IV-376
Bachmeir, C. III-187
Bae, S.H. III-234
Bai, Y. IV-900
Baik, S. II-678
Bajaj, Chandrajit II-419,III-756
Bakar, N. III-379
Bala, J. II-678
Bała, P. I-99
Balas, L. II-618
Balázs, F. I-318
Baldridge, K.K. I-75,I-148
Baliś, B. I-107,III-26
Balk, I. IV-1012, IV-1045
Balogh, Z. III-583, III-599
Balos, K. III-114
Banaś, K. II-155
Banaszak, J. IV-687, IV-694
Bandera, G. I-409
Bang, Y.-C. I-582,I-602,I-611
Bang Nielsen, M. III-921
Bansevicius, R. IV-278
Barbosa, J.L.V. III-2
Barreiro, E. IV-139
Bass, J. III-756
Bassi, A. III-202
Batty, M. III-1109
Batura, T. III-545
Beck, M. III-202
Bein, D. III-1233
Beletskyy, V. I-635, III-398
Benedyczak, K. I-99
Benoit, A. III-289
Benouahmane, B. II-557
Benyon, D. III-936
Bergamaschi, L. IV-434
Bernard, M. III-986
Bernatík, R. III-631
Bernholdt, D.E. III-1
Bernsen, N.O. III-946
Berthold, J. III-297
Bertoldo, Alberto II-614
Bertolotto, M. IV-948
Beseri, T. IV-376

Bešter, J. III-141,III-179
Bevzushenko, M. IV-1049
Beyls, K. III-448
Bianco, M. II-614
Bidarra, J. III-1025
Bielak, J. III-717
Bierbrauer, M. IV-859
Bijl, H. IV-604
Biros, G. III-717
Blažič, B.J. III-141,III-179
Blesa, M. I-429
Boada, I. IV-98
Bogdanov, A.V. I-239
Bojanic, S. I-644
Boku, T. IV-629
Bollapragada, R. IV-1017
Bonelli, A. II-459
Bonnefoi, P.-F. IV-80
Borgosz, Jan II-507
Boss, M. III-1070
Bote-Lorenzo, M.L. I-495
Boufflet, J.P. II-163
Boukhanovsky, A.V. I-239
Bourchtein, A. IV-667
Bourchtein, L. IV-667
Brasser, P. IV-637
Breitkopf, P. II-163
Bretto, A. IV-343
Broeckhove, J. I-51,II-443
Brogan, D. III-789
Brotzge, J. III-670
Browne, J.C. III-756
Bruno, G. I-75
Brunton, R. III-835
Bruti Liberati, N. IV-771
Brutzman, D. III-835
Brzezinski, J. II-475,III-82
Bubak, M. I-107,I-462,III-26, III-34, III-472, III-848
Buchtela, D. IV-1133
Budinska, I. III-599
Budzyńska, Ł. III-1241
Bungartz, H.-J. IV-394, IV-1180
Buono, N. Del IV-459
Burak, D. I-635
Burgos, P. II-204
Burnecki, K. IV-868
Burns, J. III-1094
Bushehrian, O. II-699
Byeon, O.H. III-250
Bylina, B. II-578
Bylina, J. II-578
Byun, Y. I-334

Caballer, M. III-54
Čada, V. IV-123
Cai, G. II-404, IV-956, IV-981, IV-989
Cai, J.-M. II-435
Cai, W. III-856
Cai, X. III-889
Cała, J. IV-1172
Caliari, M. IV-434
Calleja, M. IV-483
Calteau, A. IV-700
Camahort, E. IV-197
Campos, C. IV-197
Cánovas, Ó. III-158
Cant, S. I-478
Cantillo, K. I-523
Capitelli, M. IV-580, IV-588
Čapkovič, F. III-529
Carlucci Santana, R.H. I-537,I-661
Carmichael, G.R. III-709
Carvalho, R. III-1109
Cascón, J.M. II-130,II-586
Castro, F. IV-189
Catalyurek, U. III-805
Cebrat, S. IV-687, IV-694, IV-709, IV-765
Cepulkauskas, A. IV-278
Cermak, M. II-412
Cetnarowicz, K. III-607
Cha, W. I-648
Chai, T. III-709
Chandresakar, V. III-670
Chang, J.H. IV-1088
Changhuang, C. I-59
Chanussot, F. III-202
Chapman, B. III-408
Chaturvedi, A. III-779
Chen, H. I-59,I-470,I-623
Chen, L. I-310,I-507
Chen, M. III-725
Chen, R. III-889
Chen, W. III-1131
Chen, Y. III-388, III-414
Chen, Z. I-657, IV-931
Cheng, C. IV-972
Cheng, L. III-141
Chi, J. III-779

Chin Jr., G. II-540
Cho, G. I-648
Cho, H.-S. I-381
Cho, J.-W. I-43
Cho, J.H. II-687
Cho, K. III-1117
Cho, S. I-255
Cho, S.-P. I-590
Cho, T.-S. IV-1095
Choe, T.-Y. II-598
Choe, Y.H. IV-1095
Choi, J.K. I-607
Choi, I.-B. I-18
Choi, J. I-445
Choi, K.H. III-234
Choi, M.-H. II-300
Choi, S. I-499
Choi, S.J. II-121
Choi, W.-H. I-569
Choo, H. I-565,I-573,I-582,I-602, I-611,III-266,III-1138
Chopard, B. IV-540
Chover, M. IV-107, IV-173
Chrastek, R. IV-41,IV-49
Chun, K.J. III-250
Chung, C.H. I-687
Chung, J. I-349
Chung, M.G. III-1193
Chung, M.Y. I-573,III-1201
Chung, T.M. III-250
Chung, W.-G. III-1170
Ciereszko, A. II-451
Ciesielski, M. II-348
Çinar, A. II-523
Ciskowski, P. II-196
Coen, J.L. III-725
Cole, M. III-289
Cole, M.J. III-701
Coll, N. IV-98
Combarro, E.F. I-666,II-658
Conversin, P. IV-1165
Coppex, F. IV-742
Corchado, E. II-204,II-650
Corchado, J. II-650
Corradini, A. III-913, III-946
Correia, M.V. II-380
Cortés, A. II-220
Cortes, T. I-10
Costa, C.A. da III-2
Créput, J.-C. III-638,III-1102
Cruz Guzmán, J. de II-552,II-602
Cuyt, A. I-695,II-557
Cyganek, B. II-507,II-548
Czekierda, Ł. IV-1172

D'Apice, C. IV-351
D'Hollander, E.H. III-448
Daescu, D. III-709
Danion, F. IV-1056
Danon, L. III-1078
Daoud, D.S. II-582
Darema, F. III-662
Datta, A.K. III-1233
Datta, D. III-1209, III-1224
Daubin, V. IV-700
Davinelli, M. IV-1073
Demeester, P. III-149
Demkowicz, L.F. III-756
Deng, H. I-507
Deng, Q. I-507
Deng, S. I-627
Denis, C. II-163
Deschrijver, D. II-443
DeTar, C. IV-1202
Dew, R. I-397
Dhaene, T. II-443
Dhar, P. II-284
Dhoedt, B. III-149
Dhooge, A. II-42
Dias, A. III-1025
Díaz, I. I-666,II-658
Díaz-Guilera, A. III-1078
Diele, F. IV-426
Dimitriadis, Y.A. I-495
Diomede, P. IV-580
Dirnstorfer, S. IV-394
Doallo, R. I-132
Dobrucky, M. I-124
Dolk, D. III-779
Dong, H. I-174
Dong, S. I-287
Dongarra, J. III-432, IV-270
Dooren, P. Van III-740
Dopico, A.G. II-380
Douglas, C.C. III-701,III-725
Dove, M.T. IV-483
Downar, T. IV-1188
Drake, D. III-835
Dreżewski, R. III-654
Droegemeier, K. III-670

Droz, M. IV-742
Duan, X. II-260
Dubey, P. I-657
Dubu, D. I-490
Dudek, M.R. IV-687, IV-694
Dudkiewicz, M. IV-687, IV-694
Dumitrescu, D. II-670
Dumitriu, L. III-497, III-560
Duplaga, M. IV-1118
Dupuis, A. IV-556
Dureisseix, D. IV-612
Duwe, H. IV-1140
Dydejczyk, A. II-638
Dzemyda, G. I-652
Dziewierz, M. III-583
Dziurzanski, P. III-398

Eberspächer, J. III-187
Efendiev, Y. III-701
El-Aker, F. II-606
El-Amrani, M. II-89
Ellahi, T.N. III-210
Enticott, C. I-148
Epanomeritakis, I. III-717
Erciyes, K. I-357
Erdogan, N. I-184,I-437
Ergenc, T. IV-364
Ermolaev, V. IV-1049
Escobar, J.M. II-642
Esparza, O. III-174
Evangelinos, C. III-685
Ewing, R. III-701

Fahringer, T. II-459,III-18,III-42, III-424, III-456
Falcou, J. III-314
Fan, J. II-435
Fangohr, H. IV-1210
Fantozzi, S. IV-1073
Farhat, C. III-693
Farreras, M. I-10
Federl, P. II-65,II-138
Feng, Y. II-26, III-756
Fernández, J. II-658
Ferragut, L. II-130,II-586
Fic, G. II-654
Fidanova, S. IV-523
Fogelson, A.L. IV-1202
Forné, J. III-174
Forster, F. III-90
Frączak, M. II-451
Franca, L.P. III-725
Fredriksson, M. III-946
Freitag, F. I-10
Frijns, A.J.H. IV-661
Fritsche, W. IV-1165
Fudos, I. IV-258
Funika, W. I-462, III-472
Fyfe, C. II-650

Gabrijelčič, D. III-179
Gaffiot, F. IV-1037
Galán, R. I-523
Galis, A. III-141,III-166
Gall, M. II-467, IV-673
Gallivan, K.A. III-740
Gao, F. IV-1056
García, F.J. III-158
García, J.M. I-541
García, P. III-98
García-Loureiro, A.J. II-590
Gardner, H. IV-1218
Gargiulo, G. IV-351
Garic, S. I-148
Garstecki, Ł. III-74
Gashkov, I. IV-370
Gashkov, I.B. II-561
Gashkov, S.B. II-561
Gava, F. III-50, III-331
Gavidia, D.P. III-970
Gdaniec, Z. III-1013
Gecow, A. III-552
Gelas, J.-P. III-202
Geyer, C.F.R. III-2
Ghattas, O. III-717
Ghim, H. I-43
Giczi, D. I-83
Gillibert, L. IV-343
Gilmore, S. III-289, III-306
Gimenez, J. I-466
Ginting, V. III-701
Glasa, J. IV-18
Głut, B. II-163, II-646
Goh, K.-I. III-1038
Goh, Y.K. II-332
Goldstein, R.A. IV-718
Gomes, A.J.P. IV-221
Gómez-Sánchez, E. I-495
González, P. I-132
González-Castaño, F.J. I-326

González-Yuste, J.M. II-642
Gordon, B.W. II-292
Górriz, J.M. II-50
Goscinski, A. I-397
Gou, J. I-503
Govaerts, W. II-42
Gracio, D.K. II-540
Grama, A. III-740
Gray, L. III-756
Greenberg, J.P. I-75
Gregory, R.W. IV-1056
Griffith, V. IV-1165
Grochowski, M. III-623
Grønbæk, K. III-921
Gronek, P. II-638
Groşan, C. II-674,III-1257,III-1281
Großmann, S. III-865
Groumpos, P. II-515
Gschaider, B.F.W. IV-564
Guan, Y. IV-964
Gubała, T. III-34
Guimerà, R. III-1078
Gunn, C. III-905
Guo, W. II-388
Gurov, T.V. IV-507

Ha, J.S. IV-90, IV-131
Habala, O. I-124
Haber, R.E. I-523
Haddow, P.C. III-1249
Hadjarian, A. II-678
Haley, P.J. III-685
Hammami, M. I-310
Hampton, S.S. II-268
Han, H.S. III-258
Han, K. I-271,I-279,I-334,II-309,II-317
Han, K.-H. I-598
Han, N. II-309
Han, S. I-499,I-586,III-1185
Han, S.K. I-255
Han, S. II-427
Han, Z. I-623
Hanzliček, P. IV-1133
Harakaly, R. III-202
Havran, V. IV-164
Hayes, M. I-478
Henze, D. III-709
Heritage, J.P. III-1224
Hernández, V. I-482,III-54
Herrero, P. I-433,I-545, III-575, III-1004, IV-252
Herruzo, E. I-409
Hertzberger, B. I-515, III-771
Heyfitch, V. IV-1004
Hieb, M. III-835
Higaki, H. I-691
Hilaire, V. III-638
Hilbers, P.A.J. IV-661
Hillston, J. III-289
Hippold, J. II-146
Hluchy, L. I-124,I-425, III-599
Hochreiter, R. I-83
Hoekstra, A.G. IV-548
Hofacker, I.L. IV-728
Hoffmann, C. III-740
Hoffmann, G.A. II-682
Holmgren, S. II-9
Honeger, C.C. IV-564
Hong, I. I-611
Hong, J. III-1185
Hong, M. II-491,II-499
Hong, M. II-300
Hong, V. IV-34
Horák, B. III-631
Horan, P. I-397
Horie, K. IV-915
Houstis, E. III-693, III-732
Hsieh, S.-Y. I-223
Hu, J. I-174
Hu, Q. IV-923
Hu, Y. II-404, IV-956, IV-981, IV-989
Huang, G. IV-907
Huang, L. I-627, III-408
Huang, W. I-206
Huang, Y. III-1209, III-1224
Huerta, J. IV-197
Huh, E.N. III-1162,III-1193
Huttunen, P. III-10
Hwang, D.-U. I-255
Hwang, I.-C. I-43
Hwang, I.-S. I-594
Hwang, S. I-116,I-445
Hyun, Y.K. IV-181

Iglesias, A. IV-229, IV-237, IV-303
Ihm, I. II-419
Ikonen, J. III-10
Im, E.-J. I-116
Imamiya, A. III-897

Imre, S. I-318
Ingram, D. IV-1165
Ipanaqué, R. IV-303
Ivanova, T. IV-388
Ivanovska, S. IV-499
Iwadate, Y. I-519
Ixaru, L.G. IV-443
Izaguirre, J.A. II-268

Jacobs, R.L. II-332
Jaeschke, M. III-218
Jalili-Kharaajoo, M. II-662,II-708, II-713,II-717
Jang, J. III-124
Janicki, A. IV-803
Jankowski, G. I-449
Januszewski, R. I-449
Jaworek, K. IV-1103
Jean, K. III-166
Jeanblanc, M. IV-787
Jee, J.-G. I-342
Jeltsch, R. IV-1196
Jenkins, K. I-478
Jeong, C.-S. III-873
Jeong, H.Y. III-266
Jeong, K. I-116
Jeong, S.-J. III-1146
Jeong, Y. III-281
Jessup, E.R. IV-1242
Jezierski, J. I-532
Jiang, J. II-113
Jiang, J.Q. II-666
Jiang, L. III-58
Jiang, M. IV-923
Jiang, Y. III-591
Jigang, W. I-200
Jiménez, J.E. I-523
Jin, H. I-623
Jing, Z. I-373
Jinshu, S. I-373
Jo, G.-S. III-513
Joan-Arinyo, R. IV-139
Johannesson, L. III-946
Johns, C. III-725
Johnson, C.R. III-701, IV-1202
Jolivet, V. IV-115
Jones, G. III-701
Joo, B.-g. III-1185
Joshi, A. III-732
Jost, G. I-466
Jovanovic, V. III-748
Jugravu, A. III-18
Jung, H. I-43,II-317
Jung, J.J. I-247, III-513
Jung, K.-Y. I-295
Jurczyk, T. II-646
Jyung, K.J. III-274

Kacsuk, P. IV-475
Kahng, B. III-1038
Kalkuhl, M. II-228
Kallivokas, L.F. III-717
Kalra, D. IV-1165
Kamiński, M. II-171
Kaminski, M. IV-1029
Kan, Y. I-373
Kang, C. I-590
Kang, D. I-421,II-113
Kang, H.-K. IV-1110
Kang, K. III-1117
Kang, S.-S. I-670
Kang, Y. III-1117
Kapałka, M. III-34
Kar, T. II-260
Karaivanova, A. IV-499
Karniadakis, G.E. III-827
Kasperska, E. II-634
Katarzyniak, R. III-567
Katsionis, G. III-962
Katz, M. I-75
Katzfey, J. I-164
Kawulok, L. III-218
Kechadi, M.T. III-210
Kendall, R. III-408
Khan, J.I. III-978, III-1008
Khanal, M. II-364
Kharitonovich, A. IV-1049
Khrustaleva, E.Y. II-73
Kim, B.J. IV-1065
Kim, C. II-491
Kim, D. III-1038
Kim, D.-H. I-43
Kim, D.-W. III-363, III-371
Kim, D.H. II-687,II-695
Kim, E.J. III-717
Kim, H. I-43,I-577,I-594
Kim, H.-K. II-179
Kim, H.S. IV-1095
Kim, J.-H. I-598
Kim, J.-S. I-569,III-1170

Kim, J.A. I-255
Kim, J.H. IV-1088
Kim, J.O. I-687
Kim, J.R. I-519
Kim, K. III-1185
Kim, K.J. I-631
Kim, M. I-582,I-602, III-1217
Kim, S.-M. IV-1095,IV-1110
Kim, S.S. III-274
Kim, T.K. III-250
Kim, W. II-212, II-491, II-499
Kim, W.-H. IV-1110
Kim, Y. I-499,I-639
Kim, Y.-C. III-1209,III-1224
Kim, Y.H. IV-1081,IV-1095
Kirstein, P. IV-1165
Kisiel-Dorohinicki, M. III-646
Kitowski, J. I-458, III-583
Kleijn, C.R. IV-637
Klicker, K.R. II-540
Klie, H. III-805
Knight, D. III-748
Ko, Y.-B. I-598
Königsmann, J. III-946
Koestler, H. IV-410
Koetsier, J. II-650
Kokosiński, Z. I-215
Kokoszka, P. IV-827
Kolingerová, I. IV-123, IV-147
Kołodziej, M. I-215
Kominek, P. III-1241
Kommineni, J. I-164
Kondratieva, P. IV-164
Kong, J. II-388
Koo, J. I-349
Korczak, J.J. IV-835
Kormentzas, G. III-195
Kosacki, K. III-623
Koshigoe, H. II-105
Kou, G. IV-931
Koukam, A. III-638, III-1102
Kowalczuk, M. IV-687, IV-694
Kowarschik, M. III-440
Kozlak, J. III-638, III-1102
Kozlowski, J. III-521
Kozlowski, T. IV-1188
Krammer, B. III-464
Kramp, G. III-921
Krawczyk, H. I-557
Krawczyk, K. III-583, III-599
Kremens, R. III-725
Krishnan, A. II-284
Krumm-Heller, A. III-905
Kudela, H. II-356
Kuksin, A.Y. IV-596
Kulikov, G.Y. II-73,II-565,II-570
Kulikowski, J.L. III-488
Kulvietiene, R. IV-278
Kulvietis, G. IV-278
Kuo, T.W. III-226
Kupczyk, M. I-91
Kurc, T. III-805
Kurdziel, M. I-462
Kurgan, E. II-244,II-252
Kurose, J. III-670
Kurzyniec, D. I-140
Kutner, R. II-467, IV-673
Kuznetsov, Y.A. II-42
Kwak, B.M. IV-1088
Kwarciany, K. I-215
Kwiatkowska, T. IV-787
Kwon, C.H. III-234
Kwon, O. I-499
Kwon, S.Y. IV-1065
Kwon, Y. I-607
Kyriazis, I. IV-258

Labarta, J. I-10,I-466
Laclavik, M. III-599
Ladevèze, P. IV-612
Lai, K.K. IV-876
Lamantia, F. IV-779
Lambert, S. III-583
Landman, I. IV-1045
Langer, M. IV-1021, IV-1037
Lannocca, M. IV-1073
Latash, M.L. IV-1056
Lawenda,M. III-1013
Lawrence, S. III-764
Lawson, S.W. III-936
Lazarov, R. III-701
Lee, B.-H. III-1201
Lee, B.G. III-258
Lee, C. II-300
Lee, D.-Y. II-691
Lee, D.C. I-594,I-631
Lee, G.-S. I-389
Lee, H. I-639,II-499
Lee, H.-H. I-365
Lee, H.-J. III-1209

Lee, H.K. III-281, III-1124, III-1217
Lee, H.P. II-666
Lee, H.S. II-121
Lee, J. I-590
Lee, J.-B. II-179
Lee, J.-D. I-18
Lee, J.-H. I-295
Lee, K.-H. I-381
Lee, K.-J. I-569
Lee, K.-W. I-381
Lee, S. I-577,I-639
Lee, S.-G. IV-1081
Lee, S.-H. I-365, III-371
Lee, S.J. IV-1065
Lee, T.-D. III-873
Lee, T.-J. I-573, III-1201
Lee, W.-G. III-363
Lee, Y.-H. I-683
Leemput, P. Van IV-572
Lees, M. III-881
Lefèvre, L. III-202
Leone, P. IV-540
Lermusiaux, P.F.J. III-685
Leszczynski, J.S. II-348
Lewis, R.W. II-622
Li, J. III-388, III-414
Li, J. III-889
Li, J. IV-892
Li, M. I-174
Li, S. I-441, IV-1056
Li, X. I-441
Li, Y. II-26,II-574
Li, Z.-M. IV-1056
Liang, Y.C. II-666
Liao, W. III-709
Lim, D. I-279
Lim, H.W. I-474
Lim, J. III-266
Lim, S. II-396
Lin, D. IV-295
Lin, W. II-58,II-81,II-574
Lin, W.D. I-549
Lipinski, P. IV-835
Lipowski, A. IV-742
Lipowsky, R. III-1062
Lisik, Z. IV-1021,IV-1037
Lisowska, A. IV-1
Liu, C.S. III-226
Liu, H. III-58
Liu, J. IV-892
Liu, L. II-372
Liu, R. II-372
Liu, Z. II-26, III-408
Lloyd, D. IV-1165
Lluch, J. IV-197
Lobry, J.R. IV-679
Löf, H. II-9
Loffeld, O. II-228
Logan, B. III-881
Loitière, Y. III-789
Lombardo, S. III-106
Longo, S. IV-580, IV-588
Lopez, L. IV-459
Loulergue, F. I-401
Low, M.Y.H. III-856
Lu, F. IV-964, IV-972
Lu, J. I-421,II-113
Lu, T. IV-653
Lu, X. I-287
Lucor, D. III-827
Luengo, F. IV-229, IV-237
Lüthi, J. III-865
Lukac, R. IV-72
Lukasik, J. III-1102
Luna, J.M. I-561
Luo, S.P. II-58
Luo, Y. II-404, IV-940, IV-956, IV-981,IV-989,IV-997
Luo, Z. III-897
Luque, E. II-220
Lust, K. IV-572
Luszczek, P. IV-270
Luzón, M.V. IV-139

Ma, Y. III-978
MacDonald, D. II-650
Macedo Mourelle, L. de III-1289
Machì, A. III-106
Machiraju, R. III-764
Machmoum, A. II-89
Mackiewicz, D. IV-687, IV-694
Mackiewicz, P. IV-687, IV-694
Maeng, S.-R. I-43
Magaña, E. III-133
Magdoń-Maksymowicz, M. IV-750, IV-758
Majerowski, A. IV-673
Majewska, M. III-583
Maksymowicz, A.Z. IV-758
Malarz, K. II-638

Malawski, M. III-34, III-848
Malowiecki, M. III-537
Mandel, J. III-725
Manohar, P. III-1178
Manos, K. III-962
Mantiuk, R. IV-264
Mao, Y. I-470
Marangi, C. IV-426
Marco, J. IV-245
Margalef, T. II-220
Mari, J.-L. IV-156
Marín, M. III-480
Marinov, D. III-187
Markvoort, A.J. IV-661
Martín, M.J. I-132
Martínez, G. III-158
Martucci, S. IV-418
Mat Deris, M. III-379
Matossian, V. III-805
Matsuhisa, T. IV-884, IV-915
Mazurkiewicz, J. II-236
McCarthy, J.J. III-685
McGregor, J.L. I-164
McLaughlin, D. III-670
Mead, J. IV-451
Meer, H. De III-90
Mehta, M. III-946
Mehta, S. III-779
Melnik, R.V.N. IV-645
Meng, Z. IV-923
Merkulov, A.I. II-73
Mesquita, A. III-1265
Metaxas, D. III-813
Meyer, A. II-146
Meyer, N. I-91,I-449,III-1013
Meyer, R.R. I-326
Miatliuk, K. IV-1103
Michelson, G. IV-41,IV-49
Michopoulos, J. III-693, III-732, IV-621
Midura, J. III-114
Mielikäinen, T. I-231
Mihajlović, M.D. II-97
Mihaylova, L. IV-531
Mijalković, S. II-97
Mikolajczak, R. I-449
Min, B.-M. I-342
Mock, S. I-75
Moltó, G. I-482
Momot, J. III-623
Mondéjar, R. III-98
Montañés, E. I-666, II-658
Montenegro, R. II-642
Montero, G. II-642
Moon, K. III-124
Moon, S. I-334
Moore, S. III-432
Moore, T. III-202
Moreno, F. IV-213
Morey, J. III-996
Morgado, J.F.M. IV-221
Morozov, I.V. IV-596
Morse, K.L. III-835
Morzy, T. I-532
Mouriño, J.C. I-132
Mucci, P. III-432
Müller, M.S. III-464
Mukherjee, B. III-1209, III-1224
Mun, J.H. IV-1095
Mun, Y. III-281, III-1124, III-1217
Muñoz, J.L. III-174
Murawko-Wisniewska, K. IV-264
Murzin, F. III-545
Murzyn, K. II-325

Nachev, G. III-187
Napieralski, A. IV-1029
Navarro-Moldes, L. I-615
Ndoundam, R. II-610
Nedea, S.V. IV-661
Nedjah, N. III-1289
Nenortaite, J. IV-843
Néron, D. IV-612
Neuman, L. III-521
Nguyen, N.T. III-537
Ni, J. I-619
Nikolow, D. I-458
Nipp, K. IV-1196
Nobile, F. III-756
Nöh, K. II-228, II-594
Noël, A.G. IV-335
Noh, B. I-639
Noh, B.-N. I-365
Noh, M. III-281
Noh, S.-C. I-631
Nordén, M. II-9
Norman, G.E. IV-596
Novikava, S. IV-1103
Nowak, G. II-654
Nowiński, A. I-99
Nowiński, K.S. I-99

Nowicka, A. IV-687, IV-694
Nowicki, E. II-483
Nowostawski, M. III-552
Nunes, L.M. II-380

O'Conor, I. IV-1037
O'Hare, G. III-954
Ochmańska, E. I-527
Oden, J.T. III-756
Odintsov, E. IV-1049
Oguara, T. III-881
Oh, H.S. I-342
Oh, S.-C. II-212
Oh, S.-K. II-179, II-188, II-691
Okoń, M. III-1013
Oltean, M. III-1257, III-1281
Oltean, M. II-670, II-674, III-1257, III-1281
Ono, M. I-691
Onuma, K. IV-629
Orlando, S. I-263
Ortobelli, S. IV-779
Oziewicz, Z. II-602

Pacheco, M.A.C. III-1273
Pachowicz, P. II-678
Pachter, R. II-260
Pairot, C. III-98
Palak, B. I-91
Palus, H. IV-34
Pan, G. II-427
Pan, Y. I-441
Papadopoulos, P. I-75
Papageorgiou, E. II-515
Paprzycki, M. I-490
Parashar, M. III-58, III-805
Pareschi, L. II-276
Park, B. I-271
Park, B.-J. II-188
Park, C.-I. III-363, III-371
Park, C.-I. II-598
Park, D.-H. I-295
Park, G.-L. I-453
Park, H. I-116, I-445, I-499
Park, H.-K. III-1224
Park, H.J. III-274
Park, J.D. III-266
Park, K. I-499, III-1185
Park, N. III-124
Park, S. I-499, II-419
Parsa, S. II-699
Parthasarathy, S. III-764
Pasenkiewicz-Gierula, M. II-325
Paternoster, B. IV-418, IV-443
Patrikalakis, N.M. III-685
Patterson, D. IV-1165
Paulus, D. IV-34
Pavlovsky, I. IV-1045
Pawlaczyk, L. IV-49
Payne, D.A. II-540
Paz-Ramos, M.A. II-532
Pecheanu, E. III-497
Pedroza, A.C.P. III-1265
Pedrycz, W. II-179, II-188
Pegueroles, J. III-174
Pękalski, A. IV-765
Peña, J.M. III-347
Peng, Y. IV-931
Perego, R. I-263
Pérez, F. III-347
Pérez, F.A. II-130
Pérez, G. I-561
Pérez, M.S. III-347
Perng, N.C. III-226
Perrieère, G. IV-700
Petcu, D. I-490
Philips, B. III-670
Pieczyńska-Kuchtiak, A. III-567
Pineda, U. I-561
Pingali, K. III-797
Piotrowski, M. I-557
Plank, J. III-202
Plata, O. I-409
Plataniotis, K.N. IV-72
Platen, E. IV-771
Plemenos, D. IV-80, IV-115
Pllana, S. III-42, III-456
Płóciennik, M.P. I-91
Polak, N. IV-687, IV-694
Polgár, B. III-1153
Politi, T. IV-467
Polymenakos, L. III-195
Pommereau, F. III-322
Popenda, Ł. III-1013
Porras, J. III-10
Porwik, P. IV-1
Pothoven, S. III-986
Poulingeas, P. IV-115
Pramanik, S. III-889
Preißner, O. III-929

Press, S. IV-1218
Presti, G.L. I-156
Preston, M. III-670
Primet, P. III-202
Prodan, R. II-459
Prudhomme, S. III-756
Prusinkiewicz, P. II-65, II-138
Prykarpatskyy, Y. IV-653
Pucci, G. II-614
Puhalskii, A. III-725
Pullen, J.M. III-835
Puntonet, C.G. II-50
Purvis, M. III-552

Qi, H. I-503
Qi, L. III-388
Qi, X. I-405
Qiu, X. III-1209, III-1224
Quevedo, J.R. I-666
Quintero-Marmol-Marquez, E. II-532
Quirós, R. IV-197

Ra, I.H. III-234
Rabenseifner, R. I-1
Rabiei, M. III-379
Rachev, S. IV-779
Radomski, J.P. IV-709
Ragni, S. IV-426
Rajtar, T. III-1013
Rakthanmanon, T. I-303
Ramos, J.F. IV-107
Ranilla, J. I-666, II-658
Rasmus, S. IV-795
Ravantti, J. I-231
Ray, A. I-200
Re, G.L. I-156
Rebollo, C. IV-173
Redl, C.E.P. IV-564
Regucki, P. II-356
Reilly, R. III-954
Remolar, I. IV-173
Rendell, A.P. II-17
Resch, M.M. III-464
Reynolds, P. III-789
Rhee, S.-B. I-683
Ribelles, J. IV-173
Richie, J.T. II-726
Richman, M.B. III-678
Ridwan, A.M. II-284
Rinaldi, N. IV-1021
Ro, Y.M. IV-1110
Roberts, S. IV-1218
Robinson, A.R. III-685
Robles, V. III-347
Robshaw, M.J.B. I-474
Rodgers, G.J. III-1054
Rodionov, A.S. I-565
Rodionova, O.K. I-565
Rodríguez, E. II-642
Rodríguez, M. II-204
Róg, T. II-325
Roh, S.-B. II-691
Rojek, G. III-607
Román, J.E. III-54
Romero, S. IV-213
Rosanna, M. IV-351
Ruchaud, W. IV-80
Ruede, U. IV-410
Rünger, G. II-1, II-146
Rum, F. II-292
Ruskin, H.J. III-1094
Rust, B.W. IV-1226
Rycerz, K. III-34, III-848
Rząsa, W. I-107

Sacerdoti, F. I-75
Şahan, A. I-357
Sahingoz, O.K. I-184, I-437
Salamanca, E. III-133
Salmerón, M. II-50
Saltz, J. III-805
Sameh, A. III-740
Samulyak, R. IV-653
Sánchez, A. III-347
Sands, J. III-936
Sandu, A. III-709
Sanguesa, R. I-67
Sanjeevan, K. I-67
Santana, M.J. I-537, I-661
Santini, C.C. III-1273
Santos, J.A. II-380
Santosa, B. III-678
San Juan, J.F. IV-327
Sarafian, H. IV-319
Sas, C. III-954, III-1017
Sato, M. IV-629
Savanović, A. III-141, III-179
Sbert, M. IV-189, IV-245
Schaefer, R. III-623
Schikuta, E. I-486

Schmaranz, K. IV-1156
Schmidt, H. III-685
Schmidt, P. II-244
Schneider, R. IV-588
Schrefler, B.A. IV-612
Schubert, W. II-364
Schwartmann, D. IV-1148, IV-1156
Seaïd, M. II-89, II-276
Searcy, Mary E. II-726
Sedig, K. III-996, III-1030
Segal, C. III-497, III-560
Segovia, J. III-1004
Seidel, H.-P. IV-164
Seinfeld, J.H. III-709
Sekelsky, S. III-670
Sellarès, J.A. IV-98
Semerdjiev, T. IV-531
Semovski, S.V. IV-736
Senger, L.J. I-661
Seo, S.J. I-687
Seoane, N. II-590
Sequeira, J. IV-156
Sérot, J. III-314
Serrano, S. IV-327
Serrat, J. III-133
Seymour, K. III-432
Shadwick, B.A. IV-1234
Shakhov, V.V. III-1138
Shannon, C.E. III-701
Shen, M. III-388
Shen, M.M. III-242
Shen, P. IV-972
Shi, X.H. II-666
Shi, Y. I-657, IV-892, IV-931
Shi, Z. III-355
Shin, B.-S. II-396
Shin, D.-R. III-1170
Shin, M.H. III-234
Shin, Y.-s. IV-205
Shindin, S.K. II-570
Sidi, A. IV-402
Sikorski, C.A. IV-1202
Silver, D. III-58
Silvestri, F. I-263
Simo, B. I-124
Simpson, J. III-701
Sirmutis, R. IV-843
Singhal, M. II-540
Sitarz, M. II-638
Siver, A.S. IV-358
Skala, K. III-119
Skala, V. II-412
Skarmeta, A.F.G. III-98, III-158
Skitał, Ł. I-458
Sławińska, M. I-35
Słomiński, L. IV-811
Sloot, P.M.A. III-848, III-970
Słota, D. II-634
Słota, R. I-458, III-583
Slusarczyk, K. IV-1029
Smętek, M. III-472
Smolarczyk, K. IV-687, IV-694
Smutnicki, C. II-483
Snášel, V. III-631
Snider, R. III-822
Sobachkin, A. IV-1049
Sobaniec, C. III-82
Sobecki, J. III-505
Sobera, M.P. IV-637
Sohn, S. III-124
Sojat, Z. III-119
Soldatos, J. III-195
Song, J. I-577
Song, S.K. I-453
Songsiri, C. I-303
Soon, Y.H. IV-181
Sorensen, D. III-740
Soriano, M. III-174
Špidlen, J. IV-1133
Spisak, B.J. II-630
Sridhar, V. III-1178
Srovnal, V. III-631
Stadler, P.F. IV-728
Stagni, R. IV-1073
Stals, L. IV-1218
Stąpor, K. IV-41, IV-49
Stauffer, D. IV-709
Steenhoven, A.A. van IV-661
Stegailov, V.V. IV-596
Stephan, E.G. II-540
Stevens, E. I-561
Stodghill, P. III-797
Stokłosa, D. III-1013
Storniolo, P. I-156
Strazdins, P.E. II-17
Stroiński, M. I-91,I-449,III-1013
Strych, V. IV-123
Stuer, G. I-51
Su, C.-H. III-827
Subasi, D. II-582

Sudholt, W. I-148
Summer, M. III-1070
Sun, J. I-623
Sun, X. IV-956, IV-981, IV-989
Sun, X. II-404
Sunderam, V. I-51, I-140
Sung, H. I-586
Susa, H. IV-629
Suttles, D. III-889
Suzuki, N. III-1046
Suzuri, H.M. III-379
Święcicki, M. II-703
Świtonski, A. IV-41
Szatzschneider, W. IV-787
Szegő, D. I-553
Szepieniec, T. I-107
Szwarcman, M.H. III-1273
Szychowiak, M. II-475
Szymczyk, M. I-417
Szymczyk, P. I-417

Tabery, P. III-187
Taccogna, F. IV-588
Tack, G.R. IV-1065
Tadić, B. III-1086
Tadonki, C. I-679, II-610
Tae, K.S. I-453
Takahashi, D. IV-629
Tan, A. III-166
Tang, J. II-404, IV-956, IV-981, IV-989
Tanscheit, R. III-1273
Taskinoglu, E. III-748
Tchuente, M. II-610
Teixeira, M.M. I-537
Tejchman, J. II-340, II-626
Tempone, R. III-756
Testori, J. III-456
Teyssière, G. IV-827
Thambipillai, S. I-200
Thandavan, A. IV-475
Thanh, N.N. IV-1110
Theodoropoulos, G. III-881
Thompson, D. III-764
Thurner, S. III-1070
Thysebaert, P. III-149
Tian, J. IV-907
Tian, R. III-685
Tolk, A. III-835
Tomas, J. II-364
Tomaševic, V. I-644
Tong, C. II-372
Tong, W. I-511
Torres-Jimenez, J. II-532
Tosik, G. IV-1037
Trafalis, T.B. III-678
Tran, V.D. I-124, I-425
Trease, L.L. II-540
Tricio, V. II-204
Trinitis, C. III-440
Trück, S. IV-859
Truong, H.-L. III-424
Truong, T. IV-1202
Tsifakis, D. II-17
Tsishkou, D. I-310
Tsompanopoulou, P. III-732
Tufo, H.M. IV-1242
Tufte, G. III-1249
Turck, F. De III-149
Turner, S.J. III-856

Überhuber, C. II-459
Ufuktepe, Ü. IV-376
Uhruski, P. III-623
Ukkonen, E. I-231
Um, T.-W. I-607
Umemura, M. IV-629
Urso, A. I-156
Ushakov, A. IV-1045

Vaquero-González, L.M. I-495
Vargas, C. I-561
Varnuška, M. IV-147
Vayssade, M. II-163
Vázquez, P.-P. IV-245
Vega-Gorgojo, G. I-495
Venetsanopoulos, A.N. IV-26, IV-72
Venkataraman, S. III-813
Verdonk, B. I-695, II-557
Vernov, S.Y. IV-382
Veselý, A. IV-1133
Vial, J.-P. I-679
Vianello, M. IV-434
Villa, F.J. I-541
Villazon, A. III-42
Virvou, M. III-962
Visser, A. I-515, III-771
Vivó, R. IV-197
Vlček, M. IV-56, IV-64
Vodacek, A. III-725
Vogler, C. III-813

Volckaert, B. III-149
Voss, H. II-34
Vuong, S. III-889

Wais, P. II-703
Waiyamai, K. I-303
Wajs, W. II-703
Walentyński, R.A. IV-311
Walkowiak, T. II-236
Wang, D. I-174, III-414, III-1131
Wang, G. I-648
Wang, J. II-372, II-404, IV-956, IV-981, IV-989
Wang, L. IV-645
Wang, R. I-413
Wang, S. III-414
Wang, S. IV-876
Wang, S. III-1209
Wang, X. IV-940, IV-997
Wang, Y. II-404, II-427, IV-956, IV-981, IV-989
Wang, Y. IV-900
Węgiel, M. III-26
Weide, H. van der IV-851
Weidendorfer, J. III-440
Weishäupl, T. I-486
Weron, R. IV-859, IV-868
Wesela, W. II-467
Wheeler, M.F. III-805
Whitlock, P.A. IV-507
Wiechert, W. II-228, II-594, IV-1249
Wiesinger, C. I-83
Wiktorsson, M. IV-795
Winnem, O.M. IV-1126
Wismüller, R. I-462
Witek, H. I-140
Wojciechowski, T. IV-811
Wołoszyn, M. II-630
Won, Y. I-519
Workman, D. III-986
Woźniak, A. III-615
Wozniak, M. I-675
Wozny, J. IV-1021
Wu, Y. IV-295, IV-876
Wu, Z. I-59, I-441, I-470, I-627, II-427
Wu, Z. I-507
Wypychowski, J. I-99

Xhafa, F. I-429
Xia, Z. III-591
Xian, J. II-58, II-574
Xu, B. I-405, I-421, II-113
Xu, C. I-441
Xu, J. I-59
Xu, W. IV-892
Xu, X. III-978
Xu, Z. IV-940, IV-997
Xue, J. II-26
Xue, Y. II-404, IV-956, IV-981, IV-989

Yakali, H. I-515, III-771
Yakutovich, M. IV-515
Yamin, A.C. III-2
Yan, L. I-619
Yanami, H. IV-286
Yang, G. IV-295
Yang, J. I-503
Yang, L. II-26
Yang, S.S. III-1008
Yang, S.-Y. II-81
Yang, X. I-478
Yang, X.S. II-622
Yang, Z. IV-900
Yaroslavtsev, A.F. I-573
Yazici, A. IV-364
Ye, Yang II-435
Yeguas, E. IV-139
Yeomans, J.M. IV-556
Yeon, L.S. IV-181
Yijie, W. I-373
Yoo, J.H. III-258
Yoo, K.-H. IV-90, IV-131
Yoo, S.-H. III-873
Yoon, H.-W. III-1201
Yoon, K. II-212
Yoon, W.-S. I-598
You, H. III-432
Youn, H.Y. I-453, II-121
Yu, H.L. III-242
Yu, L. IV-876
Yu, Y.-H. III-513
Yu, Z. I-627
Yuan, Z. III-856

Zahradnik, P. IV-56, IV-64
Zatsiorsky, V.M. IV-1056
Zdrojewska, D. IV-264
Żebrowski, D. IV-673
Zgrzywa, A. III-521
Zgrzywa, M. III-537

Zhang, A. IV-827
Zhang, A. II-404, IV-956, IV-981
Zhang, B. II-388
Zhang, D. I-206
Zhang, H. II-622
Zhang, J. III-355
Zhang, J. III-1209, III-1224
Zhang, L. I-287
Zhang, M. IV-972
Zhang, S. III-591, IV-907
Zhang, X. IV-900
Zhang, Y. I-413
Zhao, H. III-748
Zhao, W. III-725
Zhen, W.M. III-242
Zheng, G. I-59
Zheng, W. I-174, III-1131, IV-295
Zhi, X. I-511
Zhong, S. II-404, IV-956, IV-981, IV-989
Zhong, Y. III-591
Zhou, H. III-1062
Zhou, M. IV-948
Zhou, X. III-355
Zhu, K. III-1209
Zhu, L. IV-907
Zhu, Y. III-822
Zieliński, K. III-114, III-218, IV-1172
Zieliński, S. IV-1172
Ziemkiewicz, B. IV-819
Zlatev, Z. IV-491
Zoetebier, J. I-515, III-771
Zorin, S. IV-1012, IV-1049
Zubik-Kowal, B. IV-451
Zudilova, E.V. III-970
Zuijlen, A. van IV-604
Zvárová, J. IV-1133
Zwierz, J. IV-803

This page intentionally left blank

Lecture Notes in Computer Science

For information about Vols. 1–2970

please contact your bookseller or Springer-Verlag

Vol. 3083: W. Emmerich, A.L. Wolf (Eds.), Component Deployment. X, 249 pages. 2004.

Vol. 3074: B. Kuijpers, P. Revesz (Eds.), Constraint Databases and Applications. XII, 181 pages. 2004.

Vol. 3065: A. Lomuscio, D. Nute (Eds.), Deontic Logic. X, 275 pages. 2004. (Subseries LNAI).

Vol. 3064: D. Bienstock, G. Nemhauser (Eds.), Integer Programming and Combinatorial Optimization. XI, 445 pages. 2004.

Vol. 3063: A. Llamosí, A. Strohmeier (Eds.), Reliable Software Technologies - Ada-Europe 2004. XIII, 333 pages. 2004.

Vol. 3060: A.Y. Tawfik, S.D. Goodwin (Eds.), Advances in Artificial Intelligence. XIII, 582 pages. 2004. (Subseries LNAI).

Vol. 3059: C.C. Ribeiro, S.L. Martins (Eds.), Experimental and Efficient Algorithms. X, 586 pages. 2004.

Vol. 3058: N. Sebe, M.S. Lew, T.S. Huang (Eds.), Computer Vision in Human-Computer Interaction. X, 233 pages. 2004.

Vol. 3056: H. Dai, R. Srikant, C. Zhang (Eds.), Advances in Knowledge Discovery and Data Mining. XIX, 713 pages. 2004. (Subseries LNAI).

Vol. 3054: I. Crnkovic, J.A. Stafford. H.W. Schmidt, K. Wallnau (Eds.), Component-Based Software Engineering. XI, 311 pages. 2004.

Vol. 3053: C. Bussler, J. Davies, D. Fensel, R. Studer (Eds.), The Semantic Web: Research and Applications. XIII, 490 pages. 2004.

Vol. 3052: W. Zimmermann, B. Thalheim (Eds.), Abstract State Machines 2004. Advances in Theory and Practice. XII, 235 pages. 2004.

Vol. 3047: F. Oquendo, B. Warboys, R. Morrison (Eds.), Software Architecture. X, 279 pages. 2004.

Vol. 3046: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications - ICCSA 2004. LIII, 1016 pages. 2004.

Vol. 3045: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications - ICCSA 2004. LIII, 1040 pages. 2004.

Vol. 3044: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1140 pages. 2004.

Vol. 3043: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1180 pages. 2004.

Vol. 3042: N. Mitrou, K. Kontovasilis, G.N. Rouskas, I. Iliadis, L. Merakos (Eds.), NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications. XXXIII, 1519 pages. 2004.

Vol. 3039: M. Bubak, G.D.v. Albada, P.M. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004, Part IV. LXVI, 1271 pages. 2004.

Vol. 3038: M. Bubak, G.D.v. Albada, P.M. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004, Part III. LXVI, 1311 pages. 2004.

Vol. 3037: M. Bubak, G.D.v. Albada, P.M. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004, Part II. LXVI, 745 pages. 2004.

Vol. 3036: M. Bubak, G.D.v. Albada, P.M. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004, Part I. LXVI, 713 pages. 2004.

Vol. 3035: M.A. Wimmer (Ed.), Knowledge Management in Electronic Government. XII, 326 pages. 2004. (Subseries LNAI).

Vol. 3034: J. Favela, E. Menasalvas, E. Chávez (Eds.), Advances in Web Intelligence. XIII, 227 pages. 2004. (Subseries LNAI).

Vol. 3033: M. Li, X.-H. Sun, Q. Deng, J. Ni (Eds.), Grid and Cooperative Computing. XXXVIII, 1076 pages. 2004.

Vol. 3032: M. Li, X.-H. Sun, Q. Deng, J. Ni (Eds.), Grid and Cooperative Computing. XXXVII, 1112 pages. 2004.

Vol. 3031: A. Butz, A. Krüger, P. Olivier (Eds.), Smart Graphics. X, 165 pages. 2004.

Vol. 3030: P. Giorgini, B. Henderson-Sellers, M. Winikoff (Eds.), Agent-Oriented Information Systems. XIV, 207 pages. 2004. (Subseries LNAI).

Vol. 3029: B. Orchard, C. Yang, M. Ali (Eds.), Innovations in Applied Artificial Intelligence. XXI, 1272 pages. 2004.

Vol. 3028: D. Neuenschwander, Probabilistic and Statistical Methods in Cryptology. X, 158 pages. 2004.

Vol. 3027: C. Cachin, J. Camenisch (Eds.), Advances in Cryptology - EUROCRYPT 2004. XI, 628 pages. 2004.

Vol. 3026: C. Ramamoorthy, R. Lee, K.W. Lee (Eds.), Software Engineering Research and Applications. XV, 377 pages. 2004.

Vol. 3025: G.A. Vouros, T. Panayiotopoulos (Eds.), Methods and Applications of Artificial Intelligence. XV, 546 pages. 2004. (Subseries LNAI).

Vol. 3024: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 621 pages. 2004.

Vol. 3023: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 611 pages. 2004.

Vol. 3022: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 621 pages. 2004.

Vol. 3021: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 633 pages. 2004.

Vol. 3019: R. Wyrzykowski, J. Dongarra, M. Paprzycki, J. Wasniewski (Eds.), Parallel Processing and Applied Mathematics. XIX, 1174 pages. 2004.

Vol. 3016: C. Lengauer, D. Batory, C. Consel, M. Odersky (Eds.), Domain-Specific Program Generation. XII, 325 pages. 2004.

Vol. 3015: C. Barakat, I. Pratt (Eds.), Passive and Active Network Measurement. XI, 300 pages. 2004.

Vol. 3014: F. van der Linden (Ed), Software Product-Family Engineering. IX, 486 pages. 2004.

Vol. 3012: K. Kurumatani, S.-H. Chen, A. Ohuchi (Eds.), Multi-Agnets for Mass User Support. X, 217 pages. 2004. (Subseries LNAI).

Vol. 3011: J.-C. Régin, M. Rueher (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. XI, 415 pages. 2004.

Vol. 3010: K.R. Apt, F. Fages, F. Rossi, P. Szeredi, J. Váncza (Eds.), Recent Advances in Constraints. VIII, 285 pages. 2004. (Subseries LNAI).

Vol. 3009: F. Bomarius, H. Iida (Eds.), Product Focused Software Process Improvement. XIV, 584 pages. 2004.

Vol. 3008: S. Heuel, Uncertain Projective Geometry. XVII, 205 pages. 2004.

Vol. 3007: J.X. Yu, X. Lin, H. Lu, Y. Zhang (Eds.), Advanced Web Technologies and Applications. XXII, 936 pages. 2004.

Vol. 3006: M. Matsui, R. Zuccherato (Eds.), Selected Areas in Cryptography. XI, 361 pages. 2004.

Vol. 3005: G.R. Raidl, S. Cagnoni, J. Branke, D.W. Corne, R. Drechsler, Y. Jin, C.G. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G.D. Smith, G. Squillero (Eds.), Applications of Evolutionary Computing. XVII, 562 pages. 2004.

Vol. 3004: J. Gottlieb, G.R. Raidl (Eds.), Evolutionary Computation in Combinatorial Optimization. X, 241 pages. 2004,

Vol. 3003: M. Keijzer, U.-M. O'Reilly, S.M. Lucas, E. Costa, T. Soule (Eds.), Genetic Programming. XI, 410 pages. 2004.

Vol. 3002: D.L. Hicks (Ed.), Metainformatics. X, 213 pages. 2004.

Vol. 3001: A. Ferscha, F. Mattern (Eds.), Pervasive Computing. XVII, 358 pages. 2004.

Vol. 2999: E.A. Boiten, J. Derrick, G. Smith (Eds.), Integrated Formal Methods. XI, 541 pages. 2004.

Vol. 2998: Y. Kameyama, P.J. Stuckey (Eds.), Functional and Logic Programming. X, 307 pages. 2004.

Vol. 2997: S. McDonald, J. Tait (Eds.), Advances in Information Retrieval. XIII, 427 pages. 2004.

Vol. 2996: V. Diekert, M. Habib (Eds.), STACS 2004. XVI, 658 pages. 2004.

Vol. 2995: C. Jensen, S. Poslad, T. Dimitrakos (Eds.), Trust Management. XIII, 377 pages. 2004.

Vol. 2994: E. Rahm (Ed.), Data Integration in the Life Sciences. X, 221 pages. 2004. (Subseries LNBI).

Vol. 2993: R. Alur, G.J. Pappas (Eds.), Hybrid Systems: Computation and Control. XII, 674 pages. 2004.

Vol. 2992: E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, E.Ferrari (Eds.), Advances in Database Technology - EDBT 2004. XVIII, 877 pages. 2004.

Vol. 2991: R. Alt, A. Frommer, R.B. Kearfott, W. Luther (Eds.), Numerical Software with Result Verification. X, 315 pages. 2004.

Vol. 2990: J. Leite, A. Omicini, L. Sterling, P. Torroni (Eds.), Declarative Agent Languages and Techniques. XII, 281 pages. 2004. (Subseries LNAI).

Vol. 2989: S. Graf, L. Mounier (Eds.), Model Checking Software. X, 309 pages. 2004.

Vol. 2988: K. Jensen, A. Podelski (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. XIV, 608 pages. 2004.

Vol. 2987:I. Walukiewicz (Ed.), Foundations of Software Science and Computation Structures. XIII, 529 pages. 2004.

Vol. 2986: D. Schmidt (Ed.), Programming Languages and Systems. XII, 417 pages. 2004.

Vol. 2985: E. Duesterwald (Ed.), Compiler Construction. X, 313 pages. 2004.

Vol. 2984: M. Wermelinger, T. Margaria-Steffen (Eds.), Fundamental Approaches to Software Engineering. XII, 389 pages. 2004.

Vol. 2983: S. Istrail, M.S. Waterman, A. Clark (Eds.), Computational Methods for SNPs and Haplotype Inference. IX, 153 pages. 2004. (Subseries LNBI).

Vol. 2982: N. Wakamiya, M. Solarski, J. Sterbenz (Eds.), Active Networks. XI, 308 pages. 2004.

Vol. 2981: C. Müller-Schloer, T. Ungerer, B. Bauer (Eds.), Organic and Pervasive Computing - ARCS 2004. XI, 339 pages. 2004.

Vol. 2980: A. Blackwell, K. Marriott, A. Shimojima (Eds.), Diagrammatic Representation and Inference. XV, 448 pages. 2004. (Subseries LNAI).

Vol. 2979:I. Stoica, Stateless Core: A Scalable Approach for Quality of Service in the Internet. XVI, 219 pages. 2004.

Vol. 2978: R. Groz, R.M. Hierons (Eds.), Testing of Communicating Systems. XII, 225 pages. 2004.

Vol. 2977: G. Di Marzo Serugendo, A. Karageorgos, O.F. Rana, F. Zambonelli (Eds.), Engineering Self-Organising Systems. X, 299 pages. 2004. (Subseries LNAI).

Vol. 2976: M. Farach-Colton (Ed.), LATIN 2004: Theoretical Informatics. XV, 626 pages. 2004.

Vol. 2973: Y. Lee, J. Li, K.-Y. Whang, D. Lee (Eds.), Database Systems for Advanced Applications. XXIV, 925 pages. 2004.

Vol. 2972: R. Monroy, G. Arroyo-Figueroa, L.E. Sucar, H. Sossa (Eds.), MICAI 2004: Advances in Artificial Intelligence. XVII, 923 pages. 2004. (Subseries LNAI).

Vol. 2971: J.I. Lim, D.H. Lee (Eds.), Information Security and Cryptology -ICISC 2003. XI, 458 pages. 2004.