A Bird's-Eye View of Gröbner Bases

André Heck
CAN Expertise Center, email: heck@can.nl

October 11, 1996

Abstract

In this expository paper we give a short introduction to Gröbner basis theory and its main applications. Quite a few books ([1, 3, 11, 17]) and overview articles ([2, 7, 8, 20, 27]) on Gröbner basis theory already exist. This one differs in style and in choice of examples. The style is concrete: examples illustrate the main techniques and the use of a computer algebra system, in our case Maple, is not shunned. Most examples are taken from the science literature and illustrate the Gröbner basis techniques on real applications.

1 Introduction

Many problems in science and engineering lead to systems of equations of the type

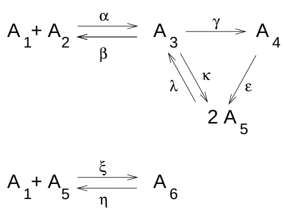
$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, x_2, \ldots, x_n) &= 0 \end{aligned}$$

where $f_1, f_2, \ldots f_m$ are polynomials in n unknowns. Then the main task is to solve the system of polynomial equations, i.e., to find the set $V(f_1, f_2, \ldots, f_m)$ of common zeros. Some problem classes with examples from existing literature are:

- **Steady state analysis of an ODE**. Given an explicit system of ODEs with polynomial right-hand sides, the steady state is described by vanishing left-hand sides.
  - NOONBURG [28]: Modeling of neural networks.

    $$\begin{aligned} cx + xy^2 + xz^2 - 1 &= 0 \\ cy + yx^2 + yz^2 - 1 &= 0 \\ cz + zx^2 + zy^2 - 1 &= 0 \end{aligned} \tag{1}$$

    where c is a parameter taking only rational values.
  - FEINBERG [14]: Modeling of a chemical reaction system of type



The steady state solutions are determined by

$$
\begin{aligned}
-\alpha c_1 c_2 + \beta c_3 - \xi c_1 c_5 + \eta c_6 &= 0 \\
-\alpha c_1 c_2 + \beta c_3 &= 0 \\
\alpha c_1 c_2 - (\beta + \gamma + \kappa) c_3 + \lambda c_5^2 &= 0 \\
\gamma c_3 - \epsilon c_4 &= 0 \\
2\kappa c_3 + 2\epsilon c_4 - 2\lambda c_5^2 + \eta c_6 - \xi c_1 c_5 &= 0 \\
\xi c_1 c_5 - \eta c_6 &= 0
\end{aligned} \tag{2}
$$

- **Geometrical descriptions**. Think of variables representing coordinates, distances, and angles (embedded in trigonometric functions).
  - van der Blij [5]: Molecular structure of cyclohexane. It is a special case of a regular hexagon in 3-space with sides of equal length (see Figure 1).

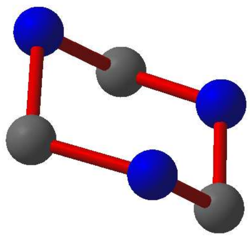


Figure 1: A cyclohexane configuration

Let x, y, z represent the squares of the lengths of the "long diagonals" of the cyclic structure, a be the square of the length of a side, and let b be the length of the "short diagonals". Consider the determinant

$$
f(a,b,x,y) = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & a & b & x & b \\ 1 & a & 0 & a & b & y \\ 1 & b & a & 0 & a & b \\ 1 & x & b & a & 0 & a \\ 1 & b & y & b & a & 0 \end{vmatrix}
$$

For cyclohexane (bond angle about 110.4°) you may approximate $a = 1, b = 8/3$. The system of polynomial equations for the general cyclic structure is

$$
f(a,b,x,y) = 0, \quad f(a,b,y,z) = 0, \quad f(a,b,z,x) = 0
$$

and we want to find the real solutions.
  - González-López/Recio [19]: Inverse kinematics of the ROMIN robot (see Figure 2).

$$
\begin{aligned}
-\sin\theta_1 (l_2 \cos\theta_2 + l_3 \cos\theta_3) - x &= 0 \\
\cos\theta_1 (l_2 \cos\theta_2 + l_3 \cos\theta_3) - y &= 0 \\
l_2 \sin\theta_2 + l_3 \sin\theta_3 - z &= 0
\end{aligned} \tag{3}
$$

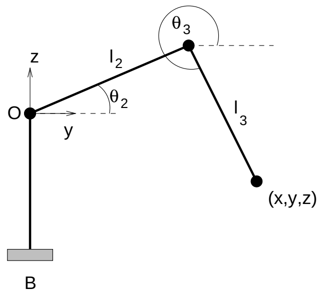


Figure 2: projection of ROMIN robot on yz-plane.

The task is to find for fixed lengths l_2 and l_3 of robot arms and for a given triple (a, b, c) the allowed values of joint angles θ_1 , θ_2 and θ_3. The equations are polynomial in terms of sines and cosines of the joint angles θ_1, θ_2, θ_3, and we may add trigonometric identities such as $\cos^2\theta_1 + \sin^2\theta_1 = 1$.

– Heck [22]: Geodesy.

$$\begin{aligned} x &= (N + h)\cos\phi\cos\lambda \\ y &= (N + h)\cos\phi\sin\lambda \\ z &= (N(1 - e^2) + h)\sin\phi \\ N &= \frac{a}{\sqrt{1 - e^2\sin^2\phi}} \\ e &= \sqrt{\frac{a^2 - b^2}{a^2}} \end{aligned} \tag{4}$$

The problem is to express the geodetic coordinates h (height), λ (longitude), and ϕ (latitude) of a point on or near the surface of the earth in terms of the geocentric Cartesian coordinates x, y, and z. This problem can be transformed into a problem of polynomial equations and solved as such.

- **Truncated power series**. An ansatz of a truncated power series substituted into an equation and the vanishing of coefficients of corresponding powers often gives a solvable system of polynomial equations.

  – K. Forsman [15]: Frequencies of periodic solutions of the van der Pol equation

$$y'' - a(1 - by^2)y' + y = 0$$

  through the method of harmonic balancing. The ansatz

$$y(t) = \sum_{k=-3}^{3} c_k e^{-ik\omega t}$$

  leads to

$$\begin{aligned} -c_1\omega^2 + abc_1^2c_{3i}\omega + c1 &= 0 \\ 2abc_1c_{3r}^2\omega + abc_1^2c_{3r}\omega + 2abc_1c_{3i}^2\omega + abc_1^3\omega - ac_1\omega &= 0 \\ -9c_{3r}\omega^2 + 3abc_{3i}c_{3r}^2\omega + 3abc_{3i}^3\omega + 6abc_1^2c_{3i}\omega - 3ac_{3i}\omega + c_{3r} &= 0 \\ -3abc_{3r}^3\omega - 9c_{3i}\omega^2 - 3abc_{3i}^2c_{3r}\omega - 6abc_1^2c_{3r}\omega + 3ac_{3r}\omega - abc_1^3\omega + c_{3i} &= 0 \end{aligned} \tag{5}$$

where $c_3 = c_{3r} + c_{3i}\, i$ and the equations $c_0 = c_2 = 0$ are left out.

- **Structural identifiability of compartmental systems**. Basic properties of compartmental systems such as reachability, observability, and identifiability are often translated into properties of systems of polynomial equations.

  – Heck [23]: Structural identifiability of a 3-compartmental model of cadmium transfer through the human body. The model is shown in Figure 3 below.

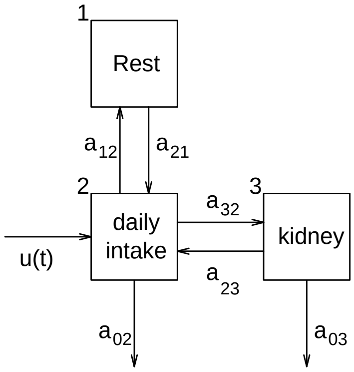


Figure 3: Cadmium transfer in the human body.

The transfer-function method leads to the following polynomials in a_{02}, a_{03}, a_{12}, a_{21}, a_{23}, a_{32}:

$$
\begin{aligned}
f_1 &= a_{32} \\
f_2 &= a_{21}a_{32} \\
f_3 &= a_{02} \\
f_4 &= a_{02}a_{03} + a_{02}a_{21} + a_{02}a_{23} + a_{03}a_{32} \\
f_5 &= a_{02}a_{03}a_{21} + a_{02}a_{21}a_{23} + a_{03}a_{21}a_{32} \\
f_6 &= a_{02} + a_{03} + a_{12} + a_{21} + a_{23} + a_{32} \\
f_7 &= a_{02}a_{03} + a_{02}a_{21} + a_{02}a_{23} + a_{03}a_{12} + a_{03}a_{21} \\
&\quad + a_{03}a_{32} + a_{12}a_{23} + a_{21}a_{23} + a_{21}a_{32}
\end{aligned}
\tag{6}
$$

Identifiability testing is equivalent to solving the polynomial equations

$$f_k(a_{02}, a_{03}, a_{12}, a_{21}, a_{23}, a_{32}) = f_k(b_{02}, b_{03}, b_{12}, b_{21}, b_{23}, b_{32})$$

for $k = 1, 2, \ldots, 7$ and where we consider the a's as unknowns and the b's as formal parameters.

Not only finding the exact solutions of systems of polynomial equations is interesting. Without solving the system of equations, you can also answer questions such as

- Is the system solvable?
- Are there a finite number of solutions and, if so, how many?

- Do equivalent systems of equations exist that give more insight to their solutions?

Gröbner basis theory is a systematic approach to answering such questions and to solving polynomial equations. In subsequent sections we shall use above examples to illustrate the theory and its applications.

2 Elementary Solution Methods

Let us consider

$$\begin{aligned} f_1 &= x - y - z \\ f_2 &= x + y - z^2 \\ f_3 &= x^2 + y^2 - 1 \end{aligned} \tag{7}$$

and let us try to find the set $V(f_1, f_2, f_3)$ of common zeros.

2.1 Heuristic Method

A heuristic approach may go as follows:

$f_1 + f_2 = 2x - z - z^2$ and $f_2 - f_1 = 2y + z - z^2$
So, $x = \frac{1}{2}(z^2 + z)$ and $y = \frac{1}{2}(z^2 - z)$.
Substitution in f_3 gives the polynomial $\frac{1}{2}z^4 + \frac{1}{2}z^2 - 1$.
Its solutions are $z = 1$, $z = -1$, $z = \sqrt{2}i$, and $z = -\sqrt{2}i$.

2.2 Gaussian Elimination-Like Method

Step 1. We choose x in the first polynomial as the first term suitable for eliminating terms in other polynomials. We multiply the first polynomial by another polynomial and subtract it from the second polynomial in order to eliminate the terms containing x. We do the same for the third polynomial.

$$V(f_1, f_2, f_3) = V(f_1, f_2 - f_1, f_3 - (x + y + z)f_1) = V(x - y - z, 2y - z^2 + z, 2y^2 + 2yz + z^2 - 1)$$

The resulting second and third polynomial have no terms that contain x. Let us call the new polynomials g_1, g_2, and g_3, respectively.

Step 2. We choose the variable y in g_2 as the most important variable. Then we multiply g_2 by another polynomial and subtract it from $2g_2$ in order to eliminate the terms containing y. We do the same for the third polynomial.

$$V(g_1, g_2, g_3) = V(2g_1 + g_2, g_2, 2g_3 - (2y + z^2 + z)g_2) = V(2x - z^2 - z, 2y - z^2 + z, z^4 + z^2 - 2)$$

The new generators are in upper triangular form: The last polynomial is only in z, the second one is only in y, z, and the first one is a polynomial in x, y, z (because of the special form of the second generator we could actually reduce the first generator to a polynomial in x, z only; in general this cannot be done).

2.3 Conclusion

The above methods have in common that they replace the original polynomials by nice polynomials that have the same solution set :

$$V(x-y-z, x+y-z^2, x^2+y^2-1) = V(2x-z^2-z, 2y-z^2+z, z^4+z^2-2)$$

"Nice" means here that the set of common zeros can be more easily computed from the new polynomials than from the original ones.

Let us go back to the Gaussian elimination-like method. During the elimination new polynomials p are formed from pairs of old ones f, g by $h = \alpha f + \beta g$, where α is a polynomial and β a scalar. h has the same common zeros as f and g and $V(f,g) = V(f,h)$ The set $I(f,g)$ of all linear combinations $\alpha f + \beta g$, where α and β are polynomials, is called the ideal generated by f and g. The set of common zeros of the ideal $I(f,g)$ equals the set of common zeros of f and g, i.e. $V(f,g) = V(I(f,g))$. So, all we do in the Gaussian elimination-like method is to choose at each step f, g, α, and scalar β clever enough such that in the end the new generators form a triangular system that can be easily solved

The Gröbner basis is another nice set of generators of an ideal. Actually, the generators computed above form a Gröbner basis. In the next section we shall introduce Gröbner bases and give the basic algorithm for computing them.

3 Basics of the Gröbner Basis Method

The mathematical ingredients of Gröbner bases will be introduced step by step.

3.1 Term Ordering

In the Gaussian elimination-like method of the previous section we selected in each step the term most suitable for further elimination of terms. We call such term the *leading term* in a polynomial. The rule of choosing leading terms is an example of a *term ordering*. For multivariate polynomials in $x_1, x_2, \ldots, x_n$ the *pure lexicographic ordering* is the linear ordering determined by

$$\begin{aligned} &{x_1}^{i_1}{x_2}^{i_2}\cdots{x_n}^{i_n} \prec {x_1}^{j_1}{x_2}^{j_2}\cdots{x_n}^{j_n} \text{ if and only if,} \\ &\quad \text{for some } l \in \{1,\ldots n-1\}, i_k = j_k \text{ for all } k < l \text{ and } i_l < j_l\,. \end{aligned}$$

For example, in the pure lexicographic ordering of three variables with $z \prec y \prec x$ we have

$$\begin{aligned} 1 &\prec z \prec z^2 \prec \ldots \prec y \prec yz \prec yz^2 \prec \ldots \\ &\prec y^2 \prec y^2z \prec y^2z^2 \prec \ldots \prec x \prec xz \prec xz^2 \prec \ldots \\ &\prec xy \prec xy^2 \prec \ldots \prec x^2 \prec \ldots \end{aligned}$$

Another ordering is the *total degree inverse lexicographic ordering* defined by

$$\begin{aligned} &{x_1}^{i_1}{x_2}^{i_2}\cdots{x_n}^{i_n} \prec {x_1}^{j_1}{x_2}^{j_2}\cdots{x_n}^{j_n} \text{ if and only if} \\ &\quad \text{either } \sum_{k=1}^{n} i_k < \sum_{k=1}^{n} j_k \\ &\quad \text{or } \sum_{k=1}^{n} i_k = \sum_{k=1}^{n} j_k \text{ and, for some } l \in \{1,\ldots n-1\}, i_l < j_l \text{ and } i_k = i_k \text{ for all } k > l\,. \end{aligned}$$

In this ordering, the total degree of a term (i.e. the sum of the exponents of the variables) is the most important thing and terms of equal total degree are ordered using the so-called inverse lexicographic ordering. In three variables with $z \prec y \prec x$:

$$\begin{aligned} 1 &\prec z \prec y \prec x \\ &\prec z^2 \prec yz \prec xz \prec y^2 \prec xy \prec x^2 \\ &\prec z^3 \prec yz^2 \prec xz^2 \prec y^2z \prec xyz \\ &\prec x^2z \prec y^3 \prec xy^2 \prec x^2y \prec x^3 \prec \dots \end{aligned}$$

Other term orderings are possible in Gröbner basis theory. The ordering $\prec$ only has to be *admissible*, i.e., satisfy

$$\begin{array}{ll} \text{(i)} & 1 \prec t \text{ for every term } t \neq 1 \\ \text{(ii)} & s \prec t \Longrightarrow s \cdot u \prec t \cdot u \text{ for all terms } s, t, u \end{array}$$

To each nonzero polynomial f we can associate the *leading term*:

$$\mathrm{lt}(f) = \textit{term that is maximal among those in } f$$

The *leading coefficient* is defined as

$$\mathrm{lc}(f) = \textit{the coefficient of the leading term of } f$$

The leading term of f is the product of the leading coefficient of f and the *leading monomial* $\mathrm{lm}(f)$ of f:

$$\mathrm{lt}(f) = \mathrm{lc}(f) \cdot \mathrm{lm}(f)$$

An example in the pure lexicographic ordering $z \prec y \prec x$:

$$\mathrm{lt}(2y - z^2 + z) = 2y, \quad \mathrm{lc}(2y - z^2 + z) = 2, \quad \mathrm{lm}(2y - z^2 + z) = y.$$

3.2 Polynomial Reduction and Normal Form

For nonzero polynomials f, g and polynomial $\tilde{f}$ we say that f *reduces to* $\tilde{f}$ *modulo* g and denote it by $f \longrightarrow_g \tilde{f}$ if there exists a term t in f that is divisible by the leading term of g and $\tilde{f} = f - \dfrac{t}{\mathrm{lt}(g)} \cdot g$. Admissibility of the term ordering $\prec$ guarantees that if the terms in f and $\tilde{f}$ are ordered from high to low terms, then the first terms in which these polynomials differ are t in f and some lower term in $\tilde{f}$

Some examples in the pure lexicographic ordering $z \prec y \prec x$:

$$x + y - z^2 \longrightarrow_{x-y-z} 2y - z^2 - z$$

$$x^2 + y^2 - 1 \longrightarrow_{x-y-z} x^2 + y^2 - 1 - \frac{x^2}{x}(x - y - z) = xy + xz + y^2 - 1$$

$$\begin{aligned} xy + xz + y^2 - 1 &\longrightarrow_{x-y-z} xz + 2y^2 + yz - 1 \\ &\longrightarrow_{x-y-z} 2y^2 + 2yz + z^2 - 1 \end{aligned}$$

In the last example, after two steps, a so-called *normal form* of $xy+xz+y^2-1$ with respect to $x-y-z$ is obtained and no further reductions are possible. We denote it by

$$\operatorname{normalf}(xy+xz+y^2-1, x-y-z) = 2y^2+2yz+z^2-1$$

Let $G = g_1, g_2, \ldots, g_m$ be a set of polynomials. A polynomial f *reduces to* $\tilde{f}$ *modulo* G if there exists a polynomial g_i in G such that $f \longrightarrow_{g_i} \tilde{f}$. A *normal form* $\operatorname{normalf}(f, G)$ *of* f *with respect to* G is a polynomial obtained after a finite number of reductions which contains no term anymore that is divisible by leading terms of polynomials of G. The normal form is in general not unique: consider the pure lexicographic ordering of two variables x and y such that $y \prec x$. Let $g_1 = x^2y-1, g_2 = xy^2-1$ and $f = x^2y^2$. Then $f \longrightarrow_{g_1} = f - yg_1 = y$ and $f \longrightarrow_{g_2} = f - xg_2 = x$. After each step no more reductions are possible. Both x and y are normal forms of f with respect to $\{g_1, g_2\}$.

3.3 Characterization of a Gröbner Basis

G is a Gröbner basis (with respect to an admissible ordering) if and only if normal forms modulo G are unique, i.e., for all f, g, h: if $g = \operatorname{normalf}(f, G)$ and $h = \operatorname{normalf}(f, G)$, then $g = h$.

Alternatively,

G is a Gröbner basis if and only if $\operatorname{normalf}(g, G) = 0$ for all g in the ideal generated by G.

Essentially, these characterizations are not much more than definitions of a Gröbner basis. They do not give any clue of how to verify the property of normal forms or how to compute such a basis. For this we have to introduce the concept of a *S-polynomial*: the *S-polynomial* $\operatorname{spoly}(f, g)$ of polynomials f and g is defined by

$$\operatorname{spoly}(f, g) = \operatorname{lcm}\Big(\operatorname{lt}(f), \operatorname{lt}(g)\Big) \cdot \left(\frac{f}{\operatorname{lt}(f)} - \frac{g}{\operatorname{lt}(g)}\right),$$

where $\operatorname{lcm}(p, q)$ denotes the least common multiple of polynomials p and q. Alternatively, we could define $\operatorname{spoly}(f, g) = \alpha \cdot f - \beta \cdot g$, where α and β are chosen such that the leading terms cancel in the difference and the degree of $a\,b$ is minimal. Two examples in the lexicographic ordering with $y \prec x$:

- For $f = x-y-z$ and $g = x^2+y^2-1$ we have $\operatorname{spoly}(f, g) = x \cdot f - g = -xy - xz - y^2 + 1$.
- For $f = x^2y-1$ and $g = yx^2-1$ we have $\operatorname{spoly}(f, g) = y \cdot f - x \cdot g = x - y$.

An algorithmic characterization of a Gröbner basis is the following:

A finite set G of polynomials is a Gröbner basis if and only if $\operatorname{normalf}(\operatorname{spoly}(f, g), G) = 0b$ for all pairs (f, g) in G.

3.4 The Buchberger Algorithm

The following algorithm written in Algol-like pseudo code computes for a given finite set G of polynomials a Gröbner basis GB such that the ideal generated by GB equals the ideal generated by G.

procedure groebnerBasis(G)
$GB \longleftarrow G$
$B \quad \longleftarrow \{(f,g) | f,g \in G, f \neq g\}$
while $B \neq \emptyset$ **do**
$(f,g) \longleftarrow$ select a pair in B
$B \longleftarrow B \setminus \{(f,g)\}$
$h \longleftarrow$ normalf(spoly(f,g), GB)
if $h \neq 0$
then $GB \longleftarrow GB \cup \{h\}$
$B \quad \longleftarrow B \cup \{(f,h) \,|\, f \in GB\}$
fi
od
return(GB)
end

Buchberger ([6]) invented this algorithm and proved that it terminates with a Gröbner basis as final result.

Let us see how the Buchberger algorithm goes for our example

$$f_1 = x - y - z, \quad f_2 = x + y - z^2, \quad f_3 = x^2 + y^2 - 1$$

We choose the pure lexicographic ordering with $z \prec y \prec x$.

1. $GB \longleftarrow \{f_1, f_2, f_3\}$; $B \longleftarrow \{(f_1,f_2),(f_1,f_3),(f_2,f_3)\}$;
   select (f_1,f_2); $B \longleftarrow \{(f_1,f_3),(f_2,f_3)\}$;
   spoly$(f_1,f_2) = -2y - z + z^2$ is already in normal form;
   $f_4 \longleftarrow -2y - z + z^2$; $GB \longleftarrow \{f_1,f_2,f_3,f_4\}$;
   $B \longleftarrow \{(f_1,f_3),(f_2,f_3),(f_1,f_4),(f_2,f_4),(f_3,f_4)\}$.

2. select (f_1,f_3); $B \longleftarrow B \backslash \{(f_1,f_3)\}$;
   spoly$(f_1,f_3) = -xy - xz - y^2 + 1 \overset{\text{normalform}}{\longrightarrow}_{GB} -\frac{1}{2}z^4 - \frac{1}{2}z^2 + 1$;
   $f_5 \longleftarrow -\frac{1}{2}z^4 - \frac{1}{2}z^2 + 1$; $GB \longleftarrow \{f_1,f_2,f_3,f_4,f_5\}$;
   $B \longleftarrow \{(f_2,f_3),(f_1,f_4),\ldots,(f_3,f_4),(f_1,f_5),\ldots,(f_4,f_5)\}$.

3. select (f_2,f_3); $B \longleftarrow B \backslash \{(f_2,f_3)\}$;
   spoly$(f_2,f_3) = xy - xz^2 - y^2 + 1 \overset{\text{normalform}}{\longrightarrow}_{GB} 0$.

4. select (f_1,f_4); $B \longleftarrow B \backslash \{(f_1,f_4)\}$;
   spoly$(f_1,f_4) = 2y^2 + 2yz + xz - xz^2 \overset{\text{normalform}}{\longrightarrow}_{GB} 0$.

5. all remaining pairs have S-polynomials whose normal form with respect to GB equals zero.

6. $GB = \{f_1,\ldots,f_5\} = \{x - y - z, x + y - z^2, x^2 + y^2 - 1, -2y - z + z^2, -\frac{1}{2}z^4 - \frac{1}{2}z^2 + 1\}$ is a Gröbner basis.

The following Maple code implements the Buchberger algorithm:

```
groebnerBasis := proc( polys::list(polynom), vars::list(name) )
local B,GB,p,h,i,j,f;
with(grobner,normalf); with(grobner,spoly);
B := [ seq( seq( [polys[i],polys[j]], i=1..j-1 ), j=2..nops(polys) ) ];
GB := polys;
while not B=[] do
  p := B[1];
  B := B[2..-1];
  h := normalf( spoly( p[1], p[2], vars, plex), GB, vars, plex );
  if h <> 0
  then GB := [op(GB),h];
       B := [ op(B), seq( [f,h], f=GB ) ];
  fi;
od;
GB
end;
```

It is an almost direct translation of the pseudo code given before into the Maple language.

The Gröbner basis is not unique: $\{f_1, f_4, f_5\} = \{x - y - z, -2y - z + z^2, -\frac{1}{2}z^4 - \frac{1}{2}z^2 + 1\}$ is also a Gröbner basis. A *reduced, monic Gröbner basis* G is a basis such that for all g in G: $g = \text{normalf}\Big(g, G \setminus \{g\}\Big)$ and $\text{lc}(g)$=1. This basis is unique. It is created from the computed basis in the above Buchberger algorithm by the following two steps:

Step 1.

for g in GB **do**
  if there exists a $p \in GB \setminus \{g\}$ such that $\text{lt}(p)$ divides $\text{lt}(g)$
  then remove g from GB
  fi
od

Step 2.

for g in GB **do**
  $g \longleftarrow \text{normalf}\Big(g, GB \setminus \{g\}\Big)$
  $g \longleftarrow \frac{1}{\text{lc}(g)}\, g$
od

In our example: f_2, and f_3 are removed in step 1. $f_6 \stackrel{\text{def}}{=} \text{normalf}(f_1, \{f_4\}) = x - \frac{1}{2}z^2 - \frac{1}{2}z$; So, in step 2 we get that $\{f_6, -\frac{1}{2}f_4, -2f_5\} = \{x - \frac{1}{2}z^2 - \frac{1}{2}z, y - \frac{1}{2}z^2 - \frac{1}{2}z, z^4 + z^2 - z\}$ is the reduced, monic Gröbner basis.

3.5 Improvements of Buchberger's Algorithm

The basic Buchberger algorithm from the previous subsection leaves a lot of freedom in the computational process:

- at the beginning a particular admissible ordering is chosen;

- at each step of the while-loop, a pair of polynomials in the current basis is selected.
- normal form reduction is in general not unique, so there may be different reduction steps leading to different normal forms.

The choices made in the computational process are of great influence on the performance of the algorithm, not on the final output, but on the complexity of the algorithm: number of pairs to process, growth of the coefficients, in general time and space complexity. Therefore, a lot of effort has been and is still being put into adapting the algorithm towards efficiency. We mention the main improvements and other observations:

- **Choice of admissible ordering**
  Experience shows that the total degree inverse lexicographic ordering is most efficient ordering for Buchberger's algorithm. For solving systems of equations however, the pure lexicographic ordering is more useful. Luckily, for zero-dimensional systems, i.e. polynomial systems with a finite set of solutions, the Faugère-Gianni-Lazard-Mora method (abbreviated as FGLM-method, [13] can be applied to convert Gröbner bases with respect to any ordering by linear algebra methods into Gröbner bases with respect to the pure lexicographic ordering.
- **Selection of pairs for the reduction process**
  Two well-known, heuristic strategies are:
  - *normal strategy* ([7]). Choose a pair (f, g) such that the least common multiple of the leading terms $\mathrm{lt}(f)$ and $\mathrm{lt}(g)$ is minimal in the current term ordering.
  - *sugar strategy* ([18]). The pairs are ordered with respect to a phantom degree called sugar and some tie-breaking algorithm.

  The sugar strategy appears to be the winning strategy in many cases.
- **Avoidance of useless computations of S-polynomials**
  The computation of an S-polynomial and the reduction to a normal form is a time- and space-consuming step that must possibly be avoided in the algorithm. The following criteria are used:
  - *Criterion 1*: If polynomials f and g have disjoint leading terms, i.e. if $\mathrm{lt}(f)$ and $\mathrm{lt}(g)$ have no variables in common (in other words, $\gcd(\mathrm{lt}(f), \mathrm{lt}(g)) = 1$), then $\mathrm{spoly}(f, g)$ reduces to zero with respect to $\{f, g\}$. So, for such pairs the computation of the S-polynomial and the reduction to normal form can be skipped in the Buchberger algorithm.
  - *Criterion 2*: If there are elements p, f, and g in the current basis GB such that $\mathrm{lt}(p)$ divides the least common multiple of $\mathrm{lt}(f)$ and $\mathrm{lt}(g)$, and if the pairs (f, p) and (g, p) have already been dealt with, i.e. $(f, p) \notin B$ and $(g, p) \notin B$, then the pair (f, g) can be discarded.
- **Use of polynomials in normal form**.
  In the computation of the reduced, monic Gröbner basis via Buchberger's algorithm we brought the polynomials in the Gröbner basis to monic normal form with respect to each other only as a final step. Experience shows that it is better to do this at every step in the algorithm.

- **Removal of superfluous polynomials**.
  In the computation of the reduced, monic Gröbner basis via Buchberger's algorithm we removed superfluous polynomials in the Gröbner basis after it has been computed. It is possible to remove superfluous polynomials in the intermediate bases GB in the following way: for $g, g' \in GB, g \neq g'$, if $\mathrm{lt}(g)$ divides $\mathrm{lt}(g')$, then g' can be expressed by g and the S-polynomial $\mathrm{spoly}(g, g')$. After you have reduced $\mathrm{spoly}(g, g')$ into normal form, g' as superfluous and all pairs (g', g'') can be deleted from B. Although g' are discarded you can still use it in the normal form algorithm.

- **Normal form calculation**.
  As we have seen normal form computation in general does not give a unique result: in each reduction step one has to choose the term to be eliminated and different choices may lead to different normal forms. A good strategy in the Buchberger algorithm is always to eliminate the largest term that can be eliminated in the normal form algorithm.

The following pseudo code is a better Buchberger algorithm that uses the normal selection strategy and brings the polynomials in the initial basis in normal form with respect to each other.

procedure groebnerBasis(G)
    $GB \longleftarrow$ reduceAll(G)
    $B \quad \longleftarrow \{(f,g) | f, g \in G$ with non-disjoint *leading terms*, $f \neq g\}$
    while $B \neq \emptyset$ **do**
        $(f,g) \longleftarrow$ select a pair in B with minimal least common multiple of leading terms
        $B \longleftarrow B \setminus \{(f,g)\}$
        if there exists no $p \in GB$ such that $(f,p) \notin B, (g,p) \notin B$,
           and $\mathrm{lt}(p)$ divides $\mathrm{lcm}(\mathrm{lt}(f), \mathrm{lt}(g))$
        then $h \longleftarrow$ normalf(spoly(f,g), GB)
                if $h \neq 0$
                then $GB \longleftarrow GB \cup \{h\}$
                        $B \quad \longleftarrow B \cup \{(f,h) \,|\, f \in GB\}$
                fi
        fi
    od
    return(GB)
end

procedure reduceAll(F)
    $F \longleftarrow \bigcup_{f \in F \setminus \{0\}}$ makeMonic(f)
    while normalf$\Big(f, F \setminus \{f\}\Big) \neq f$ for some $f \in F$ **do**
        $F \longleftarrow F \setminus \{f\}$
        $f \longleftarrow$ normalf(f, F)
        if $f \neq 0$ **then** $F \longleftarrow F \cup \{f\}$ **fi**
    od
    return(F)
end

These improvements have already a drastic effect on the computation of the reduced, monic Gröbner basis of our previous example $\{x-y-z, x+y-z^2, x^2+y^2-1\}$ with respect to the pure lexicographic ordering $z \prec y \prec x$. It turns out that bringing the original polynomials into monic normal form with respect to each other already produces the requested Gröbner basis $\{x - \frac{1}{2}z^2 - \frac{1}{2}z, y - \frac{1}{2}z^2 - \frac{1}{2}z, z^4 + z^2 - z\}$.

More sophisticated improvements of the basic Buchberger algorithm can among others be found in [7] (algorithm 6.3) and [3] (algorithm GRÖBNERNEW2).

3.6 Programs for Computing Gröbner Bases

Almost every modern computer algebra system contains an implementation of a Gröbner basis algorithm, some more advanced than others.

For example, the basic algorithm implemented in *Mathematica* (version 2.0 and later) is poor in the sense that

- it only works for lexicographic ordering and for rational coefficients;
- apparently works fast enough only for small problems. (cf. [10])

A little bit more useful but still not too impressive is the *grobner* package in *Maple* (release 4.0 and later):

- it allows both pure lexicographic ordering and total degree inverse lexicographic ordering;
- coefficients may be polynomials with rational coefficients;
- some algorithmic improvements to the basic Gröbner algorithm have been implemented;
- some utility functions are available. In the applications section we shall see the most important ones;
- A factorization version of Buchberger's algorithm can be used. In the applications section we shall discuss this further.

For both *Mathematica* and *Maple* improved and extended Gröbner bases are publicly available ([9, 26, 25, 16, 21]).

The *groebner* package of *Reduce* (version 3.4 and later) goes further than the two general purpose systems mentioned before:

- it can be used over a variety of different coefficient domains,
- it allows pure lexicographic, inverse lexicographic, total degree lexicographic, and total degree inverse lexicographic term ordering s.
- some algorithmic improvements to the basic Gröbner algorithm have been implemented and you can influence the computation by setting different modes;
- in the zero-dimensional case, i.e. if the set of polynomials have only a finite set of solutions, you can convert a basis from any ordering into a basis under pure lexicographic ordering;

- A factorization version of Buchberger's algorithm can be used.

Some non-commercial packages that allow Gröbner basis computations are:

- *Macaulay 1*. This system is for computation in algebraic geometry and commutative algebra and contains a fast Gröbner basis implementation; its drawback is that it only works for homogeneous polynomials and for polynomials rings over small prime fields.
- *GB* of J.-C. Faugère. Its purpose is to compute Gröbner bases of polynomial ideals and solve systems of polynomial equations quickly. Implementations of GB exist in *Axiom* and in both C and C++.
- *POSSO*. This package, which is an acronym of POlynomial System SOlver, is developed in the POSSO Esprit project n.6846 and is meant to provide researchers with fast, reliable software for computing Gröbner bases with tools to get insight in and control over the computational process. The C++ library of routines contains an implementation of Buchberger's algorithm customizable by its user with respect to pair handling strategies, normalization strategies, kind of reduction, policies for inserting new elements in the bases, etc. We refer to the WWW-site `http://posso.dm.unipi.it` for more information.

4 Properties and Applications of Gröbner Bases

We present the main properties of Gröbner bases in the form of recipes without proof and apply them if possible to one the examples in the introduction. The coefficient field is almost always assumed to be an algebraically closed field with characteristic zero, say the complex numbers. When we speak about a Gröbner basis we shall always mean the unique reduced, monic Gröbner basis. All examples will be computed with Maple. We shall always assume that the *grobner* package has been loaded via the command `with(grobner)`. The main procedure in this package is of course `gbasis` to compute a Gröbner basis. In the examples below we shall encounter other procedures of the *grobner* package.

4.1 Equivalence of Polynomial Equations

The system of polynomial equations $f_1(x_1,\ldots,x_n) = 0,\ldots\ldots,f_m(x_1,\ldots,x_n) = 0$ has the same solutions as the system arising from any Gröbner basis of $f_1,\ldots,f_m$ with respect to any term ordering.

This is the main property. It can also be formulated as

Two sets of polynomials generate the same ideal if and only if their Gröbner bases are equal (any term ordering may be chosen).

The next Maple session shows the computation of the Gröbner basis with respect to a pure lexicographic ordering for the system of polynomial equations (2), which describes the steady state of a chemical reaction.

```
> polys := [ -alpha*c[1]*c[2] + beta*c[3] - xi*c[1]*c[5] + eta*c[6],
>  -alpha*c[1]*c[2] + beta*c[3], gamma*c[3] - epsilon*c[4],
>  alpha*c[1]*c[2] - (beta+gamma+kappa)*c[3] + lambda*c[5]^2,
```

```
>  2*kappa*c[3] + 2*epsilon*c[4] - 2*lambda*c[5]^2 + eta*c[6] - xi*c[1]*c[5],
>  xi*c[1]*c[5]-eta*c[6] ]:
> reactionConstants := alpha, beta, gamma, epsilon, kappa, lambda, eta, xi:
> concentrations := c[1], c[2], c[3], c[4], c[5], c[6]:
> vars := [ reactionConstants, concentrations ]:
> gbasis( polys, vars, plex );
```

$$[-\beta\, c_3 + \alpha\, c_1 c_2, \epsilon\, c_4 - \gamma\, c_3, -\lambda\, {c_5}^2 + \gamma\, c_3 + \kappa\, c_3, \eta\, c_6 - \xi\, c_1 c_5]$$

The original set of polynomials and the Gröbner basis have the same solutions.

```
> solve( {op(")}, {c[1],c[2],c[3],c[4]} );
```

$$\{c_4 = \frac{\gamma\,\lambda\,{c_5}^2}{\epsilon\,(\gamma+\kappa)}, c_2 = \frac{\beta\,\lambda\,{c_5}^3\xi}{\alpha\,\eta\,c_6\,(\gamma+\kappa)}, c_3 = \frac{\lambda\,{c_5}^2}{\gamma+\kappa}, c_1 = \frac{\eta\,c_6}{\xi\,c_5}\}$$

The conclusion is that the chemical reaction system has a two-dimensional solution space of positive steady states.

4.2 Solvability of Polynomial Equations

The polynomial system of equations $f_1(x_1,\ldots,x_n)=0,\ldots\ldots,f_m(x_1,\ldots,x_n)=0$ *is solvable if and only if the Gröbner basis of* $f_1,\ldots,f_m$ *is not equal to* $\{1\}$.

For example, this criterion allows to show that the polynomial system of equations

$$x + xy^2 - 1 = 0, x^2y + y - 1 = 0, x^2 + y^2 - 1 = 0$$

has no solutions.

```
> gbasis( [ x + x*y^2 - 1, x^2*y + y - 1, x^2 + y^2 - 1 ], [x,y], tdeg );
```

$$[1]$$

The procedure `solvable` in the *grobner* package uses this method to verify whether a system is solvable or not.

4.3 Finite Solvability of Polynomial Equations

The polynomial system of equations $f_1(x_1,\ldots,x_n)=0,\ldots\ldots,f_m(x_1,\ldots,x_n)=0$ *has a finite number of solutions if and only if any Gröbner basis of* $f_1,\ldots,f_m$ *has the following property:*

> *For every variable* x_i*, there exists a polynomial such that its leading term with respect to the chosen term ordering is a power of* x_i.

Let us use this criterion to check whether system (1), which describes the steady state of an ODE associated to a neural network, has finite solutions.

```
> polys := [ c*x + x*y^2 + x*z^2 -1, c*y + y*x^2 + y*z^2 -1,
>            c*z + z*x^2 + z*y^2 -1 ]:
> gbasis( polys, [c,x,y,z], plex );
```

$$[cx + xy^2 + xz^2 - 1, cy + yx^2 + yz^2 - 1, cz + zx^2 + zy^2 - 1,
x^3y - y^3x - x + y, -z^3x + z + zx^3 - x, -yz^3 + z + zy^3 - y]$$

Note that no power of c appears in the gröbner basis. From the criterion it follows that there is no finite set of solutions. The procedure `finite` in the *grobner* package automates checking of finite solvability of polynomial equations via the above criterion.

```
> finite( polys, {c,x,y,z} );
```

$$false$$

Let us use this procedure to check if a finite number of solutions exist if we consider c as a parameter, i.e. in the Gröbner basis computation coefficients are considered as rational functions in c.

```
> finite( polys, {x,y,z} ):
```

$$true$$

4.4 Counting of Finite Solutions

Suppose that the system of polynomial equations $f_1(x_1, \ldots, x_n) = 0, \ldots\ldots, f_m(x_1, \ldots, x_n) = 0$ has a finite number of solutions.

> *The number of solutions (counted with multiplicities and solutions at infinity) is equal to the cardinality of the set of monomials that are no multiples of the leading monomials of the polynomials in the Gröbner basis (any term ordering may be chosen).*

We apply this criterion on the previous example, with c considered as a parameter. First we compute the leading monomials in the Gröbner basis with respect to the total degree inverse lexicographic ordering for which $z \prec y \prec x$.

```
> polys := [ c*x + x*y^2 + x*z^2 - 1, c*y + y*x^2 + y*z^2 - 1,
>            c*z + z*x^2 + z*y^2 - 1 ]:
> gbasis( polys, [x,y,z], tdeg ):
> map( f -> op(2,leadmon(f,[x,y,z],tdeg)), " );
```

$$[xy^2, zx^2, x^4, yx^2, z^5, y^2z^2, yz^4, y^4, xz^4, xz^3y, zy^3]$$

So, the set of monomials that are no multiples of these leading monomials equals

$$\{1, z, y, x, z^2, yz, xz, y^2, x^2, xy, z^3, yz^2, xz^2, y^2z, y^3, xy, z^4\}$$

and has cardinality 17. So, according to the above criterion there are 17 finite solutions. If you would use the `solve` command in Maple to find all solutions of this system of polynomial equations you could easily verify that this is indeed the correct number of solutions.

4.5 Converting a System of Polynomial Equations into Triangular Form

If a system of polynomial equations $f_1(x_1,\ldots,x_n)=0,\ldots\ldots,f_m(x_1,\ldots,x_n)=0$ has a finite number of solutions then the Gröbner basis of $\{f_1,\ldots,f_n\}$ with respect to the pure lexicographic ordering $x_1 \succ x_2 \succ \cdots \succ x_n$ has the following upper triangular structure:

$$
\begin{array}{rrrrr}
g_1(x_1, & x_2, & x_3, & \ldots, & x_n) \\
\vdots & & & & \\
g_p(x_1, & x_2, & x_3, & \ldots, & x_n) \\
 & g_{p+1}(x_2, & x_3, & \ldots, & x_n) \\
 & \vdots & & & \\
 & g_q(x_2, & x_3, & \ldots, & x_n) \\
 & & g_{q+1}(x_3, & \ldots, & x_n) \\
 & & \vdots & & \\
 & & & \vdots & \\
 & & & & g_r(x_n) \\
 & & & & \vdots \\
 & & & & g_s(x_n)
\end{array}
$$

Shape lemma. (cf. [4])
If a system of polynomial equations $f_1(x_1,\ldots,x_n)=0,\ldots\ldots,f_m(x_1,\ldots,x_n)=0$ has a finite number of solutions, then the Gröbner basis of $\{f_1,\ldots,f_n\}$ with respect to the pure lexicographic ordering $x_1 \succ x_2 \succ \cdots \succ x_n$, under some suitable assumptions verified in most of the cases, has the following structure:

$$\Big\{x_1 - g_1(x_n),\quad x_2 - g_2(x_n),\quad \ldots,\quad x_{n-1} - g_{n-1}(x_n),\quad g_n(x_n)\Big\},$$

where each g_i is a univariate polynomial.

The Gröbner basis of system (7) with respect to the pure lexicographic ordering $x \succ y \succ z$ has been computed in §3.4: $\{x - \frac{1}{2}z^2 - \frac{1}{2}z, y - \frac{1}{2}z^2 - \frac{1}{2}z, z^4 + z^2 - z\}$. It clearly satisfies the shape lemma.

As another illustration of the shape lemma we look at the polynomials (6), which define a system of polynomial equations that must be solved to verify structural identifiability of the given 3-compartmental model for cadmium transfer in the human body. The following Maple session does the work via the Gröbner basis method.

First we introduce short-hand notation for variables and create the polynomials.

```
> sequence :=
>   seq(seq( a.i.j = a[i,j], i=0..3 ), j=0..3 ),
>   seq(seq( b.i.j = b[i,j], i=0..3 ), j=0..3 ),
>   seq(seq( t.i.j = t[i,j], i=0..3 ), j=0..3 ), c3 = c[3]:
> (eval@subs)( S=sequence, 'macro(S)' ):
> f[1] := a32;
> f[2] := a21*a32;
> f[3] := a02;
> f[4] := a02*a03 + a02*a21+a02*a23 + a03*a32;
```

```
> f[5] := a02*a03*a21 + a02*a21*a23 + a03*a21*a32;
> f[6] := a02 + a03 + a12 + a21 + a23 + a32;
> f[7] := a02*a03 + a02*a21 + a02*a23 + a03*a12 + a03*a21
>          + a03*a32 +a12*a23 + a21*a23 + a21*a32;
```

$$f_1 := a_{3,\,2}$$

$$f_2 := a_{2,\,1}\,a_{3,\,2}$$

$$f_3 := a_{0,\,2}$$

$$f_4 := a_{0,\,2}\,a_{0,\,3} + a_{0,\,2}\,a_{2,\,1} + a_{0,\,2}\,a_{2,\,3} + a_{0,\,3}\,a_{3,\,2}$$

$$f_5 := a_{0,\,2}\,a_{0,\,3}\,a_{2,\,1} + a_{0,\,2}\,a_{2,\,1}\,a_{2,\,3} + a_{0,\,3}\,a_{2,\,1}\,a_{3,\,2}$$

$$f_6 := a_{0,\,2} + a_{0,\,3} + a_{1,\,2} + a_{2,\,1} + a_{2,\,3} + a_{3,\,2}$$

$$f_7 := a_{0,\,2}\,a_{0,\,3} + a_{0,\,2}\,a_{2,\,1} + a_{0,\,2}\,a_{2,\,3} + a_{0,\,3}\,a_{1,\,2} + a_{0,\,3}\,a_{2,\,1} + a_{0,\,3}\,a_{3,\,2} + a_{1,\,2}\,a_{2,\,3} + a_{2,\,1}\,a_{2,\,3} \\ + a_{2,\,1}\,a_{3,\,2}$$

Identifiability testing is equivalent to solving the polynomial equations

$$f_k(a_{02}, a_{03}, a_{12}, a_{21}, a_{23}, a_{32}) - f_k(b_{02}, b_{03}, b_{12}, b_{21}, b_{23}, b_{32}) = 0$$

for $k = 1, 2, \ldots, 7$ and where we consider the a's as unknowns and the b's as formal parameters. Below, we create this list of polynomials and compute its Gröbner basis with respect to the pure lexicographic ordering $a_{12} \succ a_{21} \succ a_{23} \succ a_{32} \succ a_{02} \succ a_{03}$.

```
> polys := [ seq( subs( { a12=b12, a21=b21, a23=b23, a32=b32,
>   a23=b23, a03=b03, a02=b02 }, f[i] ) - f[i], i=1..7 ) ]:
> vars := [a12,a21,a23,a32,a02,a03]:
> gbasis( polys, vars, plex );
```

$$[b_{0,\,2}\,a_{1,\,2} - b_{0,\,2}\,b_{1,\,2} + b_{0,\,3}\,b_{3,\,2} - b_{3,\,2}\,a_{0,\,3},\ -b_{2,\,1} + a_{2,\,1}, \\ b_{0,\,2}\,a_{2,\,3} - b_{0,\,2}\,b_{0,\,3} - b_{0,\,2}\,b_{2,\,3} - b_{0,\,3}\,b_{3,\,2} + (b_{3,\,2} + b_{0,\ ,2})\,a_{0,\,3},\ a_{3,\,2} - b_{3,\,2},\ -b_{0,\,2} + a_{0,\,2}, \\ b_{0,\,2}\,{b_{0,\,3}}^2 + {b_{0,\,3}}^2\,b_{3,\,2} - b_{0,\,2}\,b_{0,\,3}\,b_{2,\,1} + b_{0,\,2}\,b_{2,\,3}\,b_{0,\,3} - b_{0,\,2}\,b_{0,\,3}\,b_{1,\,2} \\ + (-2\,b_{0,\,3}\,b_{3,\,2} - b_{0,\,2}\,b_{0,\,3} + b_{0,\,2}\,b_{2,\,1} + b_{0,\,2}\,b_{1,\,2} - b_{0,\,2}\,b_{2,\,3})\,a_{0,\,3} + b_{3,\,2}\,{a_{0,\,3}}^2]$$

This Gröbner basis has the form described by the shape lemma. It follows that

- the 3-compartmental model is not structurally identifiable because there are two solutions and not just the trivial solution.
- the parameters a_{02}, a_{21}, and $_{32}$ are identifiable because in each solution they are equal to b_{02}, b_{21}, b_{32}, respectively.

So, the Gröbner basis method not only determines whether the compartmental model as a whole is structurally identifiable, but it also gives, in case of non-identifiability, information on which variables are identifiable and which not.

4.6 Finding a Univariate Polynomial

In the previous subsection we have seen that in the case of systems of polynomial equations with finite solutions, the system can be brought into triangular form with respect to the pure lexicographic ordering or under some assumptions into the form described by the shape lemma. The last polynomial is univariate in the lowest variable.

More often it is possible that the system of polynomials can be brought into a triangular form in which the last polynomial is univariate in the lowest variable in the chosen term ordering. We shall illustrate this on the example from geodesy with the equations (4).

An important problem in geodesy is to express the geodetic coordinates h (height), λ (longitude), and ϕ (latitude) of a point on or near the surface of the earth in terms of the geocentric Cartesian coordinates x, y, and z. The longitude λ can be found in a direct way by dividing the second equation of (4) by the first one: $\tan\lambda = \frac{y}{x}$. Squaring the first two equations of (4) and adding them gives $x^2 + y^2 = (N+h)^2\cos^2\phi$, where N is only depending on ϕ. The height h would be known once we have an expression for the latitude ϕ. But this turns out to be the most difficult part.

If you use the `Solve` procedure of *Mathematica*, the system answers that it cannot solve the problem. If you submit the problem to Maple, the `solve` procedure does its job, but returns a complicated answer that is not suitable for numeric processing. We shall use the Gröbner basis method that $\tan\phi$ is a solution of a univariate fourth degree polynomial. Cardano's formula provides then a closed form solution.

First, we bring the original system of equations that contains a square root and trigonometric functions into a system of polynomial equations. The square root has a defining polynomial in $\sin\phi$ associated:

$$S^2 = 1 - e^2\sin^2\phi.$$

We get rid of the trigonometric functions by considering them as formally independent variables and adding well-known polynomial relations for them:

$$cf = \cos\phi, \quad sf = \sin\phi, \quad tf = \frac{\sin\phi}{\cos\phi}, \quad cl = \cos\lambda, \quad sl = \sin\lambda$$

with relations

$$cf^2 + sf^2 = 1, \quad cl^2 + sl^2 = 1, \quad cf\ tf = sf$$

It appears to be convenient to introduce $d = (N+h)cf$ that satisfies the relation $d^2 = x^2 + y^2$. In this way, we come up with the following system of ten polynomials

```
> sys := [ x - (N+h)*cf*cl, y - (N+h)*cf*sl,
>   z - (N*(1-e^2)+h)*sf, cf^2 + sf^2 - 1, cl^2 +sl^2 - 1, tf*cf - sf,
>   N*S - a, S^2 + e^2*sf^2 - 1, (N+h)*cf - d, d^2 - x^2 - y^2 ];
```

$$sys := [x - (N+h)\,cf\,cl,\ y - (N+h)\,cf\,sl,\ z - (N\,(1-e^2)+h)\,sf,\ cf^2 + sf^2 - 1,$$
$$cl^2 + sl^2 - 1,\ tf\,cf - sf,\ N\,S - a,\ S^2 + e^2\,sf^2 - 1,\ (N+h)\,cf - d,\ d^2 - x^2 - y^2]$$

We have 10 polynomial equations in 14 variables. We compute the Gröbner basis with respect to the following pure lexicographical ordering of the variables

$$tf \prec sf \prec cf \prec sl \prec cl \prec h \prec y \prec x \prec S \prec N.$$

Note that we consider d, z, a, and e as formal parameters; this rules out points on the equator plane and on the north–south axis. These cases must be treated separately.

```
> vars := [N,S,x,y,h,cl,sl,cf,sf,tf]:
> gsys := gbasis( sys, vars, plex ):
```

The complete Gröbner basis is too big to be presented here. Besides, we are only interested in the univariate polynomial in *tf*, which is expected to be the last polynomial in the Gröbner basis.

```
> collect( gsys[-1], tf );  # get the polynomial in tf
```

$$-z^2 + 2\,z\,d\,tf + (-z^2 + e^2\,z^2 - d^2 + a^2\,e^4)\,tf^2 + (-2\,d\,z\,e^2 + 2\,z\,d)\,tf^3 + (-d^2 + d^2\,e^2)\,tf^4$$

So, we end up indeed with a fourth degree polynomial in $\tan\phi$ which can be solved analytically. The same answer can be found by the procedure `finduni` of the *grobner* package. This procedure uses the total degree ordering of variables to compute a Gröbner basis and uses this to construct the univariate polynomial (in the lowest variable) of least degree in the ideal generated by the polynomials.

We shall also illustrate the method of finding a univariate polynomial for the system (5). This allows us to approximate frequencies of periodic solutions of the van der Pol equation

$$y'' - a(1 - by^2)y' + y = 0.$$

We assume that the defining polynomials have been entered in Maple:

$$\begin{aligned} sys := [&-c_1\,\omega^2 + a\,b\,{c_1}^2\,c_{3,\,i}\,\omega + c_1,\\ &2\,a\,b\,c_1\,{c_{3,\,r}}^2\,\omega + a\,b\,{c_1}^2\,c_{3,\,r}\,\omega + 2\,a\,b\,c_1\,{c_{3,\,i}}^2\,\omega + a\,b\,{c_1}^3\,\omega - a\,c_1\,\omega,\\ &-9\,c_{3,\,r}\,\omega^2 + 3\,a\,b\,c_{3,\,i}\,{c_{3,\,r}}^2\,\omega + 3\,a\,b\,{c_{3,\,i}}^3\,\omega + 6\,a\,b\,{c_1}^2\,c_{3,\,i}\,\omega - 3\,a\,c_{3,\,i}\,\omega + c_{3,\,r},\\ &-3\,a\,b\,{c_{3,\,r}}^3\,\omega - 9\,c_{3,\,i}\,\omega^2 - 3\,a\,b\,{c_{3,\,i}}^2\,c_{3,\,r}\,\omega - 6\,a\,b\,{c_1}^2\,c_{3,\,r}\,\omega + 3\,a\,c_{3,\,r}\,\omega - a\,b\,{c_1}^3\,\omega c_{3,\,i}] \end{aligned}$$

To make computations simpler we discard solutions with $c_1 = 0$ and divide the first polynomial by c_1 and the second polynomial by $ac_1\omega$.

```
> sys[1] := expand( sys[1]/c[1] ):
> sys[2] := sys[2] := expand( sys[2]/(a*c[1]*omega) ):
> sys;
```

$$\begin{aligned} [&a\,b\,c_{3,\,i}\,\omega\,c_1 - \omega^2 + 1,\; b\,{c_1}^2 + b\,c_{3,\,r}\,c_1 + 2\,b\,{c_{3,\,r}}^2 + 2\,b\,{c_{3,\,i}}^2 - 1,\\ &\quad -9\,c_{3,\,r}\,\omega^2 + 3\,a\,b\,c_{3,\,i}\,{c_{3,\,r}}^2\,\omega + 3\,a\,b\,{c_{3,\,i}}^3\,\omega + 6\,a\,b\,{c_1}^2\,c_{3,\,i}\,\omega - 3\,a\,c_{3,\,i}\,\omega + c_{3,\,r},\\ &\quad -3\,a\,b\,{c_{3,\,r}}^3\,\omega - 9\,c_{3,\,i}\,\omega^2 - 3\,a\,b\,{c_{3,\,i}}^2\,c_{3,\,r}\,\omega - 6\,a\,b\,{c_1}^2\,c_{3,\,r}\,\omega + 3\,a\,c_{3,\,r}\,\omega - a\,b\,{c_1}^3\,\omega\\ &\quad + c_{3,\,i}] \end{aligned}$$

We choose the pure lexicographic ordering with $c_{3r} \succ c_{3i} \succ c_1 \succ \omega$ and we compute the Gröbner basis with respect to this term ordering. We are interested in the last polynomial in this basis.

```
> vars := [ c[3,r], c[3,i], c[1], omega ]:
> gsys := gbasis(sys, vars, plex ):
> collect( gsys[-1], omega );
```

$$3249\,\omega^{10} + (549\,a^2 - 6213)\,\omega^8 + (3754 - 567\,a^2)\,\omega^6 + (159\,a^2 - 842)\,\omega^4 + (-13\,a^2 + 53)\,\omega^2 - 1$$

It is a polynomial in ω^2 of fifth degree with one real solution for any positive a. The `finduni` procedure of the *grobner* package would give the same result. The Taylor series of ω^2 about $a = 0$ can easily be computed.

```
> subs( omega=sqrt(w), " ):
> omega^2 = series( RootOf(",w), a, 8 );
```

$$\omega^2 = 1 - \frac{1}{8}\,a^2 + \frac{3}{256}\,a^4 + \frac{1}{512}\,a^6 + \mathrm{O}(a^8)$$

4.7 Decomposition of Ideals

Suppose that during a Gröbner basis computation in the intermediate set GB of generators there appears a polynomial g that factorizes into two polynomials, say $g = g_1 \cdot g_2$. Then g vanishes if either g_1 or g_2 vanishes. A common zero of GB is either a common zero of $GB_1 = GB \cup \{g_1\}$ or of $GB_2 = GB \cup \{g_2\}$. The Gröbner basis for GB_1 and GB_2 can be computed separately. This decomposition of the Gröbner basis computation makes the algorithm more efficient and it gives more insight in the structure of the problem.

The procedure `gsolve` of Maple's *grobner* package inspects all intermediate polynomials in the Buchberger algorithm with respect to factorization and if possible forks the computation. Let us illustrate this first on system (1):

```
> polys := [ c*x + x*y^2 + x*z^2 -1, c*y + y*x^2 + y*z^2 - 1,
>            c*z + z*x^2 + z*y^2 -1 ]:
> gsolve( polys, [x,y,z] );
```

$$\begin{aligned}&[[-z + x,\, y - z,\, c\,z - 1 + 2\,z^3],\, [-1 + 2\,c\,z + c\,x + 2\,z^3,\, y - z,\, 2\,z^4 + 3\,c\,z^2 - z + c^2],\\ &\quad [x + y + z,\, y^2 + c + z^2 + y\,z,\, 1 + z^3 + c\,z], [-c^2 + (-c^3 - 2)\,z + 4\,c\,z^2 + 2\,x - 2\,c^2\,z^3,\\ &\quad -c^2 + (-c^3 - 2)\,z + 4\,c\,z^2 - 2\,c^2\,z^3 + 2\,y,\, c^2\,z^2 + 1 + 2\,c\,z - 2\,z^3 + 2\,c\,z^4],\\ &\quad [-z + x,\, -1 + 2\,c\,z + 2\,z^3 + c\,y,\, 2\,z^4 + 3\,c\,z^2 - z + c^2]]\end{aligned}$$

The above lists of polynomials define systems of equations that are more easily solved than the original system of polynomial equations.

An example where the decomposition of the Gröbner basis computation gives more insight is the following study of the conformational geometry of cyclohexane. Actually we treat the more general case of a closed linkage system M_6 in 3-space with sides of equal length. More precisely, any pair of subsequent edges is at a fixed angle, but the plane through these edges is not constrained. Figure 4 shows M_6 for right joint angles.

A picture of cylohexane can be found at URL `http://www.inria.fr/safir/DEMO/cyclohex/`. At this location you can also start a Maple demonstration of various geometries of the molecule. The (twisted) chair and boat configurations can be computed and can visually be played with. Below we look the computational part.

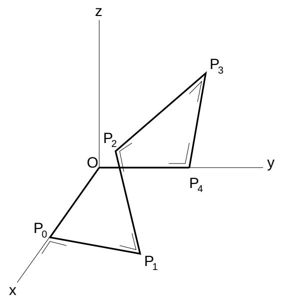


Figure 4: Closed linkage system M_6.

Let x, y, z represent the squares of the lengths of the "long diagonals" of the cyclic structure, a be the square of the length of a side, and let b be the square of length of the "short diagonals". The geometric restrictions can be formulated in algebraic terms as vanishing of the following three determinants.

```
> array( [ [0,1,1,1,1,1], [1,0,a,b,x,b], [1,a,0,a,b,y],
>          [1,b,a,0,a,b], [1,x,b,a,0,a], [1,b,y,b,a,0] ] );
```

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & a & b & x & b \\ 1 & a & 0 & a & b & y \\ 1 & b & a & 0 & a & b \\ 1 & x & b & a & 0 & a \\ 1 & b & y & b & a & 0 \end{bmatrix}$$

```
> f[1] := collect( linalg[det](''), [x,y,z], distributed, factor );
```

$$f_1 := (-4\,a^2 - 4\,b\,a - 4\,b^2)\,x\,y - 2\,(b-a)^3\,y + (b+a)\,(7\,b-a)\,(b-a)^2 + (2\,a+2\,b)\,y\,x^2 \\ + (2\,a+2\,b)\,x\,y^2 - y^2\,x^2 - (b-a)^2\,y^2 - 2\,(b-a)^3\,x - (b-a)^2\,x^2$$

Cyclic permutation gives the other two defining polynomials.

```
> f[2] := subs( {x=y,y=z}, f[1] ):
> f[3] := subs( {y=z,z=x}, f[2] ):
```

Of course there are restrictions on values of a, b, x, y, and z: $0 \le b \le 4a$ and x, y, z are between $\frac{(b-a)^2}{a}$ and $a+2b$. Let us now compute the decomposed Gröbner basis.

```
> sys := gsolve( [f[1],f[2],f[3]], [z,y,x] ):
> nops(sys);
```

$$4$$

There are 4 sets of Gröbner bases. The first three describe discrete geometries as special cases. For example,

```
> sys[1];
```

$$[-b^2\,a-b\,a^2+b^3+a^3+z^2\,a+(-b^2+b\,a-2\,a^2)\,z,\ y-b-a,\ -a-b+x]$$

This is the special case that $x=y=a+b$; solutions for z are:

```
> solve( sys[1], z );
```

$$b+a,\ \frac{b^2-2\,b\,a+a^2}{a}$$

The fourth set does not have a finite set of solutions as can be seen from its form:

```
> sys[4];
```

$$\begin{aligned}&[y\,z+3\,b^2+6\,b\,a+3\,a^2+(-2\,a-2\,b)\,x+(-2\,a-2\,b)\,y+z\,x+x\,y+(-2\,a-2\,b)\,z,-2\,a^3\\&\quad+2\,b^3-6\,b^2\,a+6\,b\,a^2+(4\,b\,a+4\,a^2+4\,b^2)\,x+(-2\,a-2\,b)\,x^2+(b^2-2\,b\,a+a^2)\,y\\&\quad+(-2\,a-2\,b)\,z\,x+y\,x^2+(-2\,a-2\,b)\,x\,y+z\,x^2+(b^2-2\,b\,a+a^2)\,z,(b^2-2\,b\,a+a^2)\,y^2\\&\quad+a^4-7\,b^4+6\,b^2\,a^2+8\,b^3\,a-8\,b\,a^3+(-2\,a^3+2\,b^3-6\,b^2\,a+6\,b\,a^2)\,x\\&\quad+(b^2-2\,b\,a+a^2)\,x^2+y^2\,x^2+(-2\,a-2\,b)\,x\,y^2\ +(4\,b\,a+4\,a^2+4\,b^2)\,x\,y\\&\quad+(-2\,a-2\,b)\,y\,x^2+(-2\,a^3+2\,b^3-6\,b^2\,a+6\,b\,a^2)\,y]\end{aligned}$$

The solution set will depend on one parameter and describes the situation where there is one degree of freedom. This freedom comes actually from the symmetry of the molecule. Once this is broken, there are only at most 16 discrete configurations. In order to get a better view on the situation, we introduce symmetric coordinates

$$s=x+y+z,\quad t=xy+yz+zx,\quad p=xyz.$$

We add these defining polynomials to the fourth basis and compute the Gröbner basis with respect to the ordering $z\succ y\succ x\succ p\succ t\succ s$.

```
> polys := [ op(sys[4]), s-x-y-z, t-x*y-y*z-z*x, p-x*y*z ]:
> gsys := gbasis( polys, [z,y,x,p,t,s], plex );
```

$$\begin{aligned}gsys:=&[-s+z+y+x,\ -3\,b^2-6\,b\,a-3\,a^2+(2\,a+2\,b)\,s+x^2-y\,s-x\,s+x\,y+y^2,\\&2\,a^3-2\,b^3+6\,b^2\,a-6\,b\,a^2+x^3+(2\,b\,a-a^2-b^2)\,s+(2\,a+2\,b)\,x\,s-x^2\,s\\&+(-3\,b^2-6\,b\,a-3\,a^2)\,x,\ 2\,a^3-2\,b^3+6\,b^2\,a-6\,b\,a^2+(2\,b\,a-a^2-b^2)\,s+p,\\&3\,b^2+6\,b\,a+3\,a^2+(-2\,a-2\,b)\,s+t]\end{aligned}$$

From the last two basis elements we can express p and t in terms of s.

```
> solve( { gsys[4], gsys[5] }, {t,p} ):
> map( collect, ",  s, factor);
```

$$\{p=(b-a)^2\,s+2\,(b-a)^3,\ t=(2\,a+2\,b)\,s-3\,(b+a)^2\}$$

The third basis element gives x as a root of a third degree polynomial in s.

```
> collect( gsys[3], [x,s], recursive, factor );
```

$$x^3-x^2\,s+((2\,a+2\,b)\,s-3\,(b+a)^2)\,x-(b-a)^2\,s-2\,(b-a)^3$$

Once x is known in terms of s then the first two basis elements can be used to determine y and z.

4.8 An Example From Robotics

The study of the inverse kinematics of the ROMIN manipulator ([19]) shall be used to give an idea how helpful Gröbner bases are for this kind of work. The system of equations is as follows:

$$\begin{aligned}
-\sin\theta_1(l_2\cos\theta_2 + l_3\cos\theta_3) - x &= 0\\
\cos\theta_1(l_2\cos\theta_2 + l_3\cos\theta_3) - y &= 0\\
l_2\sin\theta_2 + l_3\sin\theta_3 - z &= 0
\end{aligned}\tag{8}$$

where l_2 and l_3 denote the length of the first and second arm and the robot θ_1, θ_2, θ_3 represent rotation angles around the base and the robot arms. With these equations, the angles should be computed given a position (x, y, z) of the tip of the robot. Actually we are satisfied if we can find the cosines or sines of these angles. There are two ways of converting the equations into polynomial form.

- A rational parametrization of trigonometric functions is used:

  $$\cos\theta_i = \frac{1-t_i^2}{1+t_i^2}, \quad \sin\theta_i = \frac{2t_i}{1+t_i^2},$$

  for $i = 1, 2, 3$. When you use these rational expressions make sure that you multiply the equations with products of $(1+t_1^2)$, $(1+t_2^2)$, and $(1+t_3^2)$ so that polynomial equations arise. In our case, we get three polynomials in $l_2, l_3, x, y, z, t_1, t_2, t_3$ from which we could solve t_1,t_2, and t_3 by computing the decomposed Gröbner basis with respect to lexicographic ordering $x \succ y \succ z \succ t_3 \succ t_2 \succ t_1$. Drawback of this method is that joint angles of 180 degrees are not possible in this parametrization and joint angles close to 180 degrees give awkwardly large numerical values.

- The cosines and sines are considered as variables and trigonometric relations are added in the format of polynomial equations. For the ROMIN manipulator you get:

  $$\begin{aligned}
  -s_1(l_2c_2 + l_3c_3) - x &= 0\\
  c_1(l_2c_2 + l_3c_3) - y &= 0\\
  l_2s_2 + l_3s_3 - z &= 0\\
  c_1^2 + s_1^2 - 1 &= 0\\
  c_2^2 + s_2^2 - 1 &= 0\\
  c_4^2 + s_3^2 - 1 &= 0
  \end{aligned}\tag{9}$$

  where $s_i = \sin\theta_i$, $c_i = \cos\theta_i$ for $i = 1, 2, 3$. The set of defining polynomials can now be converted into a Gröbner basis with respect to the pure lexicographic ordering $s_1 \succ c_1 \succ s_2 \succ c_2 \succ s_3 \succ c_3$. We consider l_2, l_3, x, y, z as parameters.

```
> polys := [ -s[1] * (l[2]*c[2] + l[3]*c[3]) - x,
>             c[1] * (l[2]*c[2] + l[3]*c[3]) - y,
>             l[2]*s[2] + l[3]*s[3] - z,
>             c[1]^2 + s[1]^2 - 1, c[2]^2 + s[2]^2 - 1,
>             c[3]^2 + s[3]^2 - 1 ];
```

$$polys := [-s_1\,(l_2\,c_2 + l_3\,c_3) - x,\; c_1\,(l_2\,c_2 + l_3\,c_3) - y,\; l_2\,s_2 + l_3\,s_3 - z,\; {c_1}^2 + {s_1}^2 - 1,\; {c_2}^2 + {s_2}^2 - 1,\; {c_3}^2 + {s_3}^2 - 1]$$

```
> gbasis( polys, [c[3],s[3],c[2],s[2],c[1],s[1]], plex );
```

$$[2\,l_3\,x\,c_3 + 2\,z\,l_2\,s_2\,s_1 + (x^2 + y^2 + {l_3}^2 - z^2 - {l_2}^2)\,s_1,\; l_2\,s_2 + l_3\,s_3 - z,\\ -2\,z\,l_2\,s_2\,s_1 + 2\,l_2\,x\,c_2 + (-{l_3}^2 + z^2 + x^2 + y^2 + {l_2}^2)\,s_1,\\ (4\,{l_2}^2\,x^2 + 4\,{l_2}^2\,y^2 + 4\,{l_2}^2\,z^2)\,{s_2}^2 - 2\,{l_2}^2\,x^2 - 2\,{l_2}^2\,y^2 + {l_2}^4 + {l_3}^4 + z^4 - 2\,{l_3}^2\,y^2\\ + 2\,z^2\,y^2 + x^4 + y^4 - 2\,x^2\,{l_3}^2 + 2\,x^2\,z^2 + 2\,x^2\,y^2 + 2\,{l_2}^2\,z^2 - 2\,{l_3}^2\,z^2 - 2\,{l_3}^2\,{l_2}^2\\ + (4\,l_2\,z\,{l_3}^2 - 4\,l_2\,z^3 - 4\,z\,l_2\,x^2 - 4\,z\,l_2\,y^2 - 4\,z\,{l_2}^3)\,s_2,\, y\,s_1 + x\,c_1,\\ -x^2 + (x^2 + y^2)\,{s_1}^2]$$

The Gröbner basis is in triangular form. So, in principle the inverse kinematics problem is solved. However, the key problem is whether for numerical values of the parameters the above basis stays a Gröbner basis. If so, everything is ok. In the example we are considering, if we choose $l_2 \neq 0, l_3 \neq 0, x^2 + y^2 \neq 0$, then the above set is still a Gröbner basis. This specialization problem can be avoided by using so-called comprehensive Gröbner bases ([29]).

4.9 Implicitization of Parametric Objects

Consider the parametric equations

$$\begin{aligned} x_1 &= f_1(t_1, t_2, \ldots, t_m) \\ x_2 &= f_2(t_1, t_2, \ldots, t_m) \\ &\vdots \\ x_n &= f_n(t_1, t_2, \ldots, t_m) \end{aligned}$$

where $f_1, f_2, \ldots f_n$ are polynomials in m unknowns. The question is to eliminate the t's and find polynomial equations in $x_1, \ldots, x_n$ that define the parametric object. More precisely, we search for polynomials $g_1, \ldots, g_k$ in the unknowns $x_1, \ldots, x_n$ such that for all $a_1, \ldots, a_n$:

$$g_1(a_1, \ldots, a_n) = \ldots = g_k(a_1, \ldots, a_n) = 0 \text{ if and only if}\\ a_1 = f_1(b_1, \ldots, b_m), \ldots, a_n = f_n(b_1, \ldots, b_m) \text{ for some } b_1, \ldots, b_m.$$

Implicitization algorithm

Take the pure lexicographic ordering determined by $t_1 \succ \ldots \succ t_m \succ x_1 \succ \ldots \succ x_n$ *and compute the Gröbner basis GB of* $\{x_1 - f_1(t_1, \ldots, t_m), \ldots, x_n - f_n(t_1, \ldots, t_m)\}$. *Then* $\{g_1, \ldots, g_k\} = \{g \in GB \mid g \text{ is a polynomials in } x_1, \ldots, x_n \text{ only}\}$.

An example: Enneper's minimal surface defined by

$$x = \frac{1}{2}s - \frac{1}{6}s^3 + \frac{1}{2}st^2, y = -\frac{1}{2}t + \frac{1}{6}t^3 - \frac{1}{2}s^2t, z = \frac{1}{2}s^2 - \frac{1}{2}t^2.$$

The surface looks as follows:

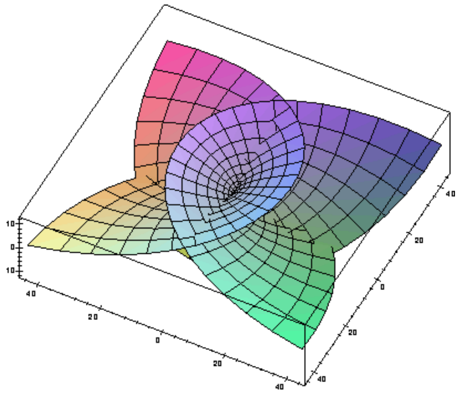


Figure 5: Enneper's minimal surface.

```
> polys := [x-s/2+s^3/6-s*t^2/2, y+t/2-t^3/6+t*s^2/2, z-s^2/2+t^2/2];
```

$$polys := [x - \frac{1}{2}\,s + \frac{1}{6}\,s^3 - \frac{1}{2}\,s\,t^2,\ y + \frac{1}{2}\,t - \frac{1}{6}\,t^3 + \frac{1}{2}\,t\,s^2,\ z - \frac{1}{2}\,s^2 + \frac{1}{2}\,t^2]$$

Next we compute the Gröbner basis with respect to the pure lexicographic ordering defined by $s \succ t \succ x \succ y \succ z$. We do not show the result here but after quite some computing time a Gröbner basis is obtained with only one polynomial in the coordinates x, y, z. This is the one shown below.

```
> GB := gbasis( polys, [s,t,x,y,z], plex ):
> collect( GB[-1], z, factor );
```

$$\begin{aligned}-1024\,z^9 &+ 4608\,z^7 + 3456\,(x-y)\,(x+y)\,z^6 + (-5184 + 15552\,x^2 + 15552\,y^2)\,z^5 \\ &+ 12960\,(x-y)\,(x+y)\,z^4 + (25272\,y^2\,x^2 - 3888\,x^2 + 4860\,y^4 + 4860\,x^4 - 3888\,y^2)\,z^3 \\ &+ 8748\,(x-y)\,(x+y)\,(x^2+y^2)\,z^2 - 729\,(x-y)^2\,(x+y)^2\,z + 1458\,(x-y)^3\,(x+y)^3\end{aligned}$$

4.10 Invertibility of Polynomial Mappings

The Jacobian conjecture states that a polynomial mapping has an inverse which is itself a polynomial mapping if and only if the determinant of the jacobian of the mapping is nonzero. In an attempt of proving the conjecture the following criterion of invertibility has been found ([12]):

Let $f_1, \ldots, f_n$ be the coordinate functions of a polynomial mapping in the variables $x_1, \ldots, x_n$. Let $y_1, \ldots, y_n$ be new indeterminates and let $\prec$ be an admissible ordering such that $y_1 \prec y_2 \prec \ldots \prec y_n \prec x_1 \prec x_2 \prec \ldots \prec y_n$. Then the mapping is invertible if and only if the Gröbner basis of $y_1 - f_1, y_2 - f_2, \ldots, y_n - f_n$ has the form $x_1 - g_1, x_2 - g_2, \ldots, x_n - g_n$. $g_1, g_2, \ldots, g_n$ are the coordinate functions of the inverse mapping.

We take the example from [24]:

$$(x,y,z) \longmapsto (x^4 + 2(y+z)x^3 + (y+z)^2x^2 + (y+1)x + y^2 + yz, x^3 + (y+z)x^2 + y, x + y + z)$$

```
> F := [ x^4 + 2*(y+z)*x^3 + (y+z)^2*x^2 + (y+1)*x + y^2 + y*z,
>        x^3 + (y+z)*x^2 + y, x + y + z ];
```

$$F := [x^4 + 2\,(y+z)\,x^3 + (y+z)^2\,x^2 + (y+1)\,x + y^2 + y\,z,\ x^3 + (y+z)\,x^2 + y,\ x + y + z]$$

```
> gbasis( [ X - F[1], Y - F[2], Z - F[3] ], [x,y,z,X,Y,Z], plex );
```

$$[x - X + Z\,Y,\ y + Z\,X^2 - Y + Z^3\,Y^2 - 2\,Z^2\,Y\,X,\ Y - Z\,Y - Z^3\,Y^2 + 2\,Z^2\,Y\,X + X - Z\,X^2 - Z + z]$$

So, the mapping has inverse

$(X, Y, Z) \longmapsto (X - ZY, Y - X^2Z + 2XYZ^2 - Y^2Z^3, -X - Y + Z + YZ + X^2Z - 2XYZ^2 + Y^2Z^3)$.

It turns out that verification of the result by composition of mappings takes more time then the Gröbner basis computation of the inverse mapping.

4.11 Simplification of Expressions

For a reduced monic Gröbner basis, the normal form of a polynomial is unique: it is a "canonical form" in the sense that two polynomials are equivalent when their normal forms are equal. This can be used to simplify mathematical expressions with respect to polynomial relations. An example from the Dutch Mathematics Olympiad of 1991, which is described in detail in [24], section 14.7.

Let a, b, c be real numbers such that

$$a + b + c = 3, \quad a^2 + b^2 + c^2 = 9, \quad a^3 + b^3 + c^3 = 24.$$

Compute $a^4 + b^4 + c^4$.

```
> siderels := [ a+b+c=3, a^2+b^2+c^2=9, a^3+b^3+c^3=24 ]:
> polys := map( lhs - rhs, siderels );
```

$$polys := \{\, a + b + c - 3, a^2 + b^2 + c^2 - 9, a^3 + b^3 + c^3 - 24 \,\}$$

Next, we compute the Gröbner basis of these polynomials with respect to the pure lexicographic ordering $a \succ b \succ c$.

```
> G :=  gbasis( polys, [a,b,c], plex );
```

$$G := [\, a + b + c - 3, b^2 + c^2 - 3\,b - 3\,c + b\,c, 1 - 3\,c^2 + c^3 \,]$$

The normal form of $a^4 + b^4 + c^4$ turns out to be 69.

```
normalf( a^4+b^4+c^4, G, [a,b,c], plex );
```

$$69$$

So, this number is the answer to the question. In Maple, this method is actually carried out when simplification with respect to side relations is requested.

```
> simplify( a^4+b^4+c^4, siderels, [a,b,c] );
```

$$69$$

5 Conclusion

This expository paper introduced Gröbner basis theory as a rich, systematic approach to solving many problems in polynomial theory. We presented the Buchberger algorithm and improvements of it. Finally, we looked at many applications.

References

[1] W. Adams, P. Lousraunau. *An introduction to Gröbner Bases.* Graduate Studies in Math., Vol 3, AMS, Oxford University Press, 1994.

[2] I. Ajwa, Z. Liu, P. Wang. *Gröbner Bases Algorithm.* ICM Technical Reports Series (ICM-199502-00), 1995. available by WWW at URL `http://symbolicnet.mcs.kent.edu/areas/groebner/gb_tutorial`

[3] T. Becker, V. Weispfenning. *Gröbner Bases: A Computational Approach to Commutative Algebra.* Springer Verlag, 1993.

[4] E. Becker, M. Marinari, T. Mora, C. Traverso. The shape of the Shape Lemma, In *Proc. ISSAC'94*, pages 129–133. ACM Press, 1994.

[5] F. van der Blij. Rekenen met ellebogen: computer algebra in dienst van de meetkunde. In Grootendorst, editor, *Vakantiecursus 1994*, CWI Sylabus 36, pages 50–76, 1994 (in Dutch).

[6] B. Buchberger. An Algorithm for Finding a Basis for a Residue Class Ring of a Zero-Dimensional Polynomial Ideal. Ph.D. Thesis, Univ. of Innsbruck (Austria), 1965 (in German).

[7] B. Buchberger. Gröbner Bases: An algorithmic Method in Polynomial Ideal Theory. In N.K. Bose, editor, *Multidimensional Systems Theory*, chapter 6, pages 184–232. Reidel Publishing Company, 1985.

[8] B. Buchberger. Applications of Gröbner Bases in Non-linear Computational Geometry. In J.R. Rice, editor, *Mathematical Aspects of Scientific Software*, IMA Volume in Mathematics and its Applications 14, pages 59–87. Springer Verlag, 1988. reprinted in D. Kapur, J.L Mundy, editors, *Geometric Reasoning*, pages 413–446, MIT Press, 1989.

[9] B. Buchberger. An Implementation of Gröbner Bases in *Mathematica.* Technical Report RISC-Linz 90-58, Univ. Linz, RISC, Linz, Austria, 1990.

[10] B. Buchberger. Gröbner Bases in Mathematica: Enthusiasm and Frustration. In P.W. Gaffney, E.N. Houstis, editors, *Proc. IFIP TC2/WG2.5 Working Conference on Programming Environments for High-Level Scientific Problem Solving*, Karlsruhe, Germany, 1991.

[11] D. Cox, J. Little, D. O'Shea. *Ideals, Varieties, and Algorithms.* Springer Verlag, 1992.

[12] A. v.d. Essen. Polynomial Maps and the Jacobian Conjecture. In A.M. Cohen, editor, *Computational Aspects of Lie Groups and Related Topics*, CWI Tract 84, pages 29–44, CWI, 1991.

[13] J.C. Faugère, P. Gianni, D. Lazard, T. Mora. Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering *J. Symbolic Computation*, 16(4):329–344, 1993.

[14] M. Feinberg. Reaction Network Structure, Multiple Steady States, and Sustained Composition Oscillations. In K. Ebert, P. Deuflard, W. Jäger, editors, *Modelling of Chemical Reaction Systems*, Springer Series in Chemical Physics 18, pages 56–68. Springer Verlag, 1981.

[15] K. Forsman and T. Glad. Constructive Algebraic Geometry in Nonlinear Control. In *Proceeding 29th CDC*, volume 5, pages 2825-2827, Honolulu, Hawaii, 1990. IEEE CSS.

[16] K. Gaterman. The moregroebner Package, 1996. available by WWW at URL `http://www.zib-berlin.de/~bzfgatte/moregroebner.html`.

[17] K.O. Geddes, S.R. Czapor, G. Labahn. *Algorithms for Computer Algebra*, chapter 10. Kluwer Academic Publishers, 1992.

[18] A. Giovinni, T. Mora, G. Niesi, L. Robbiano, C. Traverso. One sugar cube, please, or selection strategies in the Buchberger algorithm, In S.M. Watt, editor, *Proc. ISSAC'91*, pages 49–54. ACM Press, 1991.

[19] M.J. González-López and T. Recio. The ROMIN inverse geometric model and the dynamic evaluation method. In A.M. Cohen, editor, *Computer Algebra for Industry: Problem Solving in Practice*, pages 117–141. John Wiley & Sons, 1993.

[20] L. González-López. Symbolic Recipes for Polynomial System Solving: Real Solutions. Tutorial at ISSAC'96, available by anonymous ftp at `ftp://matsun1.matesco.unican.es/pub/Grupo_de_Algebra/lalo/`

[21] D. Gruntz. Groebner Bases in Gauss. *MapleTech*, 9:36–46, 1993.

[22] A.J.P. Heck. Transformation between Geocentric and Geodetic Coordinates. In A.M. Cohen, editor, *Computer Algebra for Industry: Problem Solving in Practice*, ages 203–219. John Wiley & Sons, 1993.

[23] A.J.P. Heck. Computer Algebra: A Tool in Identifiability Testing. In A.M. Cohen, L. van Gastel and S. Verduyn Lunel, editors, *Computer Algebra for Industry 2: Problem Solving in Practice*, pages 267–289. John Wiley & Sons, 1995.

[24] A.J.P. Heck. *Introduction to Maple.* 2nd edition, Springer Verlag, 1996.

[25] G. Helzer. Gröbner Bases. *Mathematica Journal*, 5(1):67–73, 1995.

[26] G. Nakos, N. Glinos. Gröbner Bases over the Integers. *Mathematica Journal*, 4(3):70–75, 1994.

[27] H. Melenk. Solving Polynomial Equation Systems by Gröbner Type Methods. *CWI Quarterly*, 3:121-136, 1990.

[28] V.W. Noonburg. A neural network modeled by an adaptive Lotka-Volterra system. *SIAM J. Appl. Math*, 49:1779–1792, 1989.

[29] V. Weispfenning Comprehensive Gröbner Bases. *J. Symbolic Computation*, 14(1):1–29, 1993.