

Tables of 64-bit Mersenne Twisters

TAKUJI NISHIMURA

Keio University

We give new parameters for a Mersenne Twister pseudorandom number generator for 64-bit word machines.

Categories and Subject Descriptors: G.3 [**Mathematics of Computing**]: Probability and Statistics—*Random number generation*

General Terms: Algorithms

Additional Key Words and Phrases: Finite fields, k -distribution, linear recurrence, Mersenne Twister, random number generation, 64-bit

1. INTRODUCTION

The Matsumoto and Nishimura [1998] Mersenne Twister (MT) is an algorithm for generating uniform pseudorandom numbers is based on a linear recurrence over the two-element field \mathbb{F}_2 , and is a special case of the multiple-recursive matrix method of Niederreiter [1993; 1995]. MT has the following properties: (1) long period, (2) efficient use of memory, (3) good k -distribution property (see Section 3), and (4) fast generation.

In a previous article by Matsumoto and Nishimura [1998], only 32-bit parameters were given. In this article, we give 64-bit parameters, which is practical because (1) 64-bit machines are growing in popularity, (2) they fit the sizeable demand for real numbers with 64-bit precision, and (3) the number of nonzero terms in a characteristic polynomial has increased to roughly twice as many as the 32-bit MT.

We also obtained a modified version by adding two more reference vectors, which dramatically increases the number of nonzero terms in the characteristic polynomial.

The work was partially supported by the Research Fellowships of the Japan Society for the Promotion of Science for Young Scientists.

Author's address: Department of Mathematics, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522, Japan; email: nisimura@comb.math.keio.ac.jp.

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

© 2001 ACM 1049-3301/00/1000-0348 \$5.00

ACM Transactions on Modeling and Computer Simulation, Vol. 10, No. 4, October 2000, Pages 348–357.

In Sections 2 and 3 we recall the MT recurrence relation and equidistribution property. In Section 4 we show tables of parameters with 64-bit MTs. In Section 5 we provide an implementation of 64-bit MTs in C.

2. RECURRENCE OF MERSENNE TWISTER

MT generates pseudorandom w -dimensional vectors over \mathbb{F}_2 (we identify w -dimensional vectors over \mathbb{F}_2 with w -bit integers) by the following recurrence:

$$\mathbf{x}_{k+n} := \mathbf{x}_{k+m} \oplus (\mathbf{x}_k^u \parallel \mathbf{x}_{k+1}^l)A \quad (k = 0, 1, \dots), \quad (1)$$

where \mathbf{x}_k is a w -dimensional vector over \mathbb{F}_2 , \oplus denotes the bitwise exclusive-or operation, n is the degree of the recurrence, m is an integer such that $1 < m < n$, A is a $w \times w$ matrix over \mathbb{F}_2 , and $(\mathbf{x}_k^u \parallel \mathbf{x}_{k+1}^l)$ is the w -dimensional vector formed by concatenating the leftmost $w - r$ bits of \mathbf{x}_k with the rightmost r bits of \mathbf{x}_{k+1} . We choose n , w and r so that $nw - r$ becomes a Mersenne exponent. We choose the form of A as follows:

$$A = \begin{pmatrix} & & & & 1 \\ & & & 1 & \\ & & & & \ddots \\ & & & & & 1 \\ a_{w-1} & a_{w-2} & \cdots & \cdots & a_0 \end{pmatrix}$$

This form of A makes multiplications fast [Matsumoto and Nishimura 1998, Section 2.1].

We must choose the parameters so that the characteristic polynomial of (1) is primitive. Then the sequence generated by the recurrence (1) attains the maximal period $2^{nw-r} - 1$. When $nw - r$ is a Mersenne exponent, there is an efficient algorithm, called the inverse-decimation method [Matsumoto and Nishimura 1998], which tests primitivity quickly. We search feasible parameters as follows: fix n , m , r , and w and select $\mathbf{a} := (a_{w-1}, a_{w-2}, \dots, a_0)$ randomly, then check the primitivity of the characteristic polynomial of (1) by the inverse-decimation method.

In addition to (1), we introduce the following slightly modified recurrence:

$$\mathbf{x}_{k+n} := \mathbf{x}_{k+m_2} \oplus \mathbf{x}_{k+m_1} \oplus \mathbf{x}_{k+m_0} \oplus (\mathbf{x}_k^u \parallel \mathbf{x}_{k+1}^l)A, \quad (k = 0, 1, \dots), \quad (2)$$

where m_0 , m_1 , and m_2 are integers such that $1 < m_0, m_1, m_2 < n$. It is easy to see that we can apply the theories and algorithms in Matsumoto and Nishimura [1998] to recurrence (2). The modification is beneficial in that, although it decreases generating speeds, the number of nonzero terms in the characteristic polynomial increases dramatically (see Table V).

3. EQUIDISTRIBUTION PROPERTY OF MT

The equidistribution property in higher dimensions is one of the strongest measures of the quality of pseudorandom number generators. The *k-distribution* is a measure of the high dimensional equidistribution property of a sequence generated by a linear recurrence over \mathbb{F}_2 .

3.1 Definition of *k*-Distribution

Definition 3.1 [Tootill et al. 1973]. A pseudorandom sequence \mathbf{x}_i of w -bit integers of period P satisfying the following condition is said to be *k-distributed to v -bit accuracy*. Let $\text{trunc}_v(\mathbf{x})$ denote the number formed by the leading v bits of \mathbf{x} , and consider P of the kv -bit vectors

$$(\text{trunc}_v(\mathbf{x}_i), \text{trunc}_v(\mathbf{x}_{i+1}), \dots, \text{trunc}_v(\mathbf{x}_{i+k-1})) \quad (0 \leq i < P). \quad (3)$$

Then, each of the 2^{kv} possible combinations of bits occurs the same number of times in a period, except for the all-zero combination that occurs once less often.

For each $v = 1, 2, \dots, w$, let $k(v)$ denote the maximum number such that the sequence is $k(v)$ -distributed to v -bit accuracy.

The geometric meaning of the above definition is as follows: Let $x_i := \mathbf{x}_i / 2^w$, i.e., normalize w -bit integers into real numbers in the $[0, 1]$ interval. Scatter the P points in the k -dimensional unit cube with coordinates $(x_i, x_{i+1}, \dots, x_{i+k-1})$ ($i = 0, 1, \dots, P - 1$). We divide equally each axis of the unit cube into 2^v pieces. Thus we have partitioned the unit cube into 2^{kv} smaller cubes. Then the sequence is *k-distributed to v -bit accuracy* if each cube contains the same number of points (except for the cube at the origin, which contains one less). Therefore, the higher $k(v)$ for each v means a higher dimensional equidistribution with v -bit precision.

Note that $2^{k(v)v} - 1 \leq P$ holds, since the number of possible patterns in (3) is $2^{k(v)v}$, and we admit the flaw at zero. In particular, $k(v) \leq \lfloor (nw - r)/v \rfloor$ holds in the case of MT.

To compute $k(v)$ for the sequences generated by recurrences (1) and (2), we use the lattice method as in Matsumoto and Nishimura [1998]. This method was developed by Couture et al. [1993]; Tezuka [1990; 1994a], using the Lenstra [1985] lattice-reduction algorithm.

3.2 Tempering

The sequences generated by (1) and (2) has poor *k-distribution* property [Matsumoto and Kurita 1994; Tezuka 1994b]. To improve the *k-distribution* to v -bit accuracy, mainly of the most significant bits, we multiply each generated word by a suitable $w \times w$ invertible matrix T from the right (called *tempering* [Matsumoto and Kurita 1994]). The

Table I. Parameters of 64-bit MTs. Recurrence (1)

ID	1	2
m	156	156
n	312	312
r	31	31
w	64	64
u	29	29
s	17	17
t	37	37
l	41	41
a	B5026F5AA96619E9	F6A3F020F058B7A7
b	D66B5EF5B4DA0000	28AAF6CDBDB40000
c	FDED6BE000000000	FDEDEAE000000000

tempering matrix, which transforms \mathbf{x} into $\mathbf{z} := \mathbf{x}T$, is determined implicitly by the following successive transformations:

$$\mathbf{y} := \mathbf{x} \oplus (\mathbf{x} \gg u) \quad (4)$$

$$\mathbf{y} := \mathbf{y} \oplus ((\mathbf{y} \ll s) \text{ AND } \mathbf{b}) \quad (5)$$

$$\mathbf{y} := \mathbf{y} \oplus ((\mathbf{y} \ll t) \text{ AND } \mathbf{c}) \quad (6)$$

$$\mathbf{z} := \mathbf{y} \oplus (\mathbf{y} \gg l), \quad (7)$$

where l , s , t , and u are integers, \mathbf{b} and \mathbf{c} are suitable bitmasks of size equal to the computer word size, $(\mathbf{x} \gg u)$ denotes the u -bit right shift of \mathbf{x} , and $(\mathbf{x} \ll u)$ denotes the u -bit left shift of \mathbf{x} . These transformations are the same as those used in Matsumoto and Nishimura [1998].

Note that it is desirable that $k(v)$ attains the trivial upper bounds $\lfloor (nw - r)/v \rfloor$ for each v , but MT cannot attain the bounds even after tempering, due to obstructions in the form of recurrences (1) and (2) [Matsumoto and Nishimura 1998]. We mention here that the L'Ecuyer [1996; 1999b] maximally equidistributed combined Tausworthe (or LFSR) generators do attain the best possible equidistribution in all dimensions and have a lot of nonzero terms in their characteristic polynomial. They are as fast as MT, although they have rather smaller periods than MT.

4. TABLES

Table I shows two sets of MT parameters of 64-bit machines for recurrence (1), with $nw - r = 19937$, a Mersenne exponent. The parameters m , n , r , w , and \mathbf{a} in the table correspond to those appearing in recurrence (1). The parameters u , s , t , l , \mathbf{b} , and \mathbf{c} in the table correspond to those in the transformations (4), (5), (6), and (7). In Table I, \mathbf{a} , \mathbf{b} , and \mathbf{c} are expressed in hexadecimals. Table II shows three sets of MT parameters for 64-bit machines for recurrence (2), with $nw - r = 19937$.

Table II. Parameters of 64-bit MTs. Recurrence (2)

ID	3	4	5
m_0	63	55	87
m_1	151	122	148
m_2	224	268	241
n	312	312	312
r	31	31	31
w	64	64	64
u	26	26	26
s	17	17	17
t	33	33	33
l	39	39	39
a	B3815B624FC82E2F	8EBD4AD46CB39A1E	CACB98F78EBCD4ED
b	599CFCBFCA660000	656BEDFFD9A40000	A51DBEFFDA6C0000
c	FFFAAFFE00000000	FDCECE7E00000000	FFEE9BF600000000

Table III. $k(v)$ ($1 \leq v \leq 64$)

ID	$k(1)$	$k(2)$	$k(3)$	$k(4)$	$k(5)$	$k(6)$	$k(7)$	$k(8)$
	$k(9)$	$k(10)$	$k(11)$	$k(12)$	$k(13)$	$k(14)$	$k(15)$	$k(16)$
	$k(17)$	$k(18)$	$k(19)$	$k(20)$	$k(21)$	$k(22)$	$k(23)$	$k(24)$
	$k(25)$	$k(26)$	$k(27)$	$k(28)$	$k(29)$	$k(30)$	$k(31)$	$k(32)$
	$k(33)$	$k(34)$	$k(35)$	$k(36)$	$k(37)$	$k(38)$	$k(39)$	$k(40)$
	$k(41)$	$k(42)$	$k(43)$	$k(44)$	$k(45)$	$k(46)$	$k(47)$	$k(48)$
	$k(49)$	$k(50)$	$k(51)$	$k(52)$	$k(53)$	$k(54)$	$k(55)$	$k(56)$
	$k(57)$	$k(58)$	$k(59)$	$k(60)$	$k(61)$	$k(62)$	$k(63)$	$k(64)$
Trivial Upper	19937	9968	6645	4984	3987	3322	2848	2492
Bound	2215	1993	1812	1661	1533	1424	1329	1246
$(P = 2^{19937} - 1)$	1172	1107	1049	996	949	906	866	830
	797	766	738	712	687	664	643	623
	604	586	569	553	538	524	511	498
	486	474	463	453	443	433	424	415
	406	398	390	383	376	369	362	356
	349	343	337	332	326	321	316	311
1	19937	9968	6645	4984	3826	3138	2511	2199
	1912	1875	1570	1560	1257	1251	1249	1246
	943	939	937	936	935	633	627	627
	625	625	624	624	624	623	623	622
	319	316	314	314	313	312	312	312
	312	312	312	312	312	312	312	311
	311	311	311	311	311	311	311	311
	311	311	311	311	311	311	311	311
2	19937	9968	6645	4984	3835	3134	2506	2197
	1901	1874	1568	1557	1253	1247	947	947
	942	938	936	935	639	629	627	626
	625	624	624	624	623	623	314	314
	314	314	314	313	313	312	312	312
	312	312	312	312	312	312	312	311
	311	311	311	311	311	311	311	311
	311	311	311	311	311	311	311	311

Table III shows the trivial upper bounds for $k(v)$ ($1 \leq v \leq 64$) when the period is $P = 2^{19937} - 1$ and the values of $k(v)$ for the generators in Table I.

Table IV. $k(v)$ ($1 \leq v \leq 64$)

ID	$k(1)$	$k(2)$	$k(3)$	$k(4)$	$k(5)$	$k(6)$	$k(7)$	$k(8)$
	$k(9)$	$k(10)$	$k(11)$	$k(12)$	$k(13)$	$k(14)$	$k(15)$	$k(16)$
	$k(17)$	$k(18)$	$k(19)$	$k(20)$	$k(21)$	$k(22)$	$k(23)$	$k(24)$
	$k(25)$	$k(26)$	$k(27)$	$k(28)$	$k(29)$	$k(30)$	$k(31)$	$k(32)$
	$k(33)$	$k(34)$	$k(35)$	$k(36)$	$k(37)$	$k(38)$	$k(39)$	$k(40)$
	$k(41)$	$k(42)$	$k(43)$	$k(44)$	$k(45)$	$k(46)$	$k(47)$	$k(48)$
	$k(49)$	$k(50)$	$k(51)$	$k(52)$	$k(53)$	$k(54)$	$k(55)$	$k(56)$
	$k(57)$	$k(58)$	$k(59)$	$k(60)$	$k(61)$	$k(62)$	$k(63)$	$k(64)$
3	19937	9968	6645	4984	3812	3134	2496	2181
	1877	1869	1569	1563	1281	1253	1250	1246
	944	939	937	936	935	634	627	626
	625	625	624	624	624	623	623	622
	321	316	312	312	312	312	312	312
	312	312	312	312	311	311	311	311
	311	311	311	311	311	311	311	311
	311	311	311	311	311	311	311	311
4	19937	9968	6645	4984	3816	3132	2494	2192
	1898	1875	1571	1562	1275	1253	1249	1246
	943	939	937	935	637	629	627	626
	625	625	624	624	623	623	623	622
	319	315	312	312	312	312	312	312
	312	312	312	312	311	311	311	311
	311	311	311	311	311	311	311	311
	311	311	311	311	311	311	311	311
5	19937	9968	6645	4984	3811	3134	2515	2181
	1877	1869	1569	1562	1285	1253	1250	1246
	944	939	937	936	935	631	627	626
	625	624	624	624	624	623	623	623
	319	315	312	312	312	312	312	312
	312	312	312	311	311	311	311	311
	311	311	311	311	311	311	311	311
	311	311	311	311	311	311	311	311

Table V. Number of Nonzero Terms in the Characteristic Polynomial

ID 1	ID 2	ID 3	ID 4	ID 5
285	319	5795	4701	6097

Table IV shows $k(v)$ for the generators of Table II. Note that $k(v)$ for the proposed generators degenerates for $v > 32$ with $k(v) \approx 312$, whereas $k(v)$ for the 32-bit generator MT19937 [Matsumoto and Nishimura 1998] degenerates for $v > 16$ with $k(v) \approx 624$. Compared to the 32-bit generator, these 64-bit generators improve the value of $k(v)$ only for $16 < v \leq 32$.

In Table V, the number of nonzero terms in the characteristic polynomial is listed. Note that the degree of the characteristic polynomial of each generator is 19937. For recurrence (1), the values are obtained from the explicit form of the characteristic polynomial in Matsumoto and Nishimura [1998]. For recurrence (2), the values are obtained from expression (8) in Appendix A. It is known that generators whose characteristic polynomials

have too few terms, such as trinomial-based GFSR, show poor randomness (see, e.g., Compagner [1991] and Matsumoto and Kurita [1996]). Thus it is preferable that the number of nonzero terms in the characteristic polynomial be large (and not too far from a half of the degree of the characteristic polynomial). In the case of recurrence (1), the number of terms is about twice as much as that of the 32-bit generator MT19937. Note that, although the number of nonzero terms in the characteristic polynomial is improved for recurrence (2), the equidistribution property is not improved.

5. IMPLEMENTATION IN C

We now give an implementation in C of generator 3 in Table II. The function `genrand()` returns a uniform pseudorandom real number in the interval $[0, 1]$ for each call. The function `sgenrand()` initializes the state array `mt[NN]` using a linear congruential generator whose modulus is 2^{64} . The multiplier is adopted from L'Ecuyer [1999a]. `sgenrand()` must be called once, before calling `genrand()` for the first time. Note that `sgenrand()` in the C code is just an example. Users can set any values in the state array except for the all zero state. Strictly speaking, the state corresponds to 19937 bits in `mt[NN]`: the 33 most significant bits of `mt[0]` and all the bits of `mt[1..NN-1]`.

```
/* Period parameters */
#define NN 312
#define M0 63
#define M1 151
#define M2 224
/* Constant vector a */
#define MATRIX_A 0xB3815B624FC82E2FULL
/* Most significant 33 bits */
#define UMASK 0xFFFFFFFF80000000ULL
/* Least significant 31 bits */
#define LMASK 0x7FFFFFFFULL

/* Tempering parameters */
#define MASK_B 0x599CFCBFA660000ULL
#define MASK_C 0xFFFAAFFE00000000ULL
#define UU 26
#define SS 17
#define TT 33
#define LL 39

/* The array for the state vector */
static unsigned long long mt[NN];
/* mti==NN+1 means mt[NN] is not initialized */
static int mti=NN+1;

void sgenrand(unsigned long long seed)
{
    unsigned long long ux, lx;

    for (mti=0; mti<NN; mti++) {
        ux = seed & 0xFFFFFFFF00000000ULL;
```

```

    seed = 2862933555777941757ULL * seed + 1ULL;
    lx = seed >> 32;
    seed = 2862933555777941757ULL * seed + 1ULL;
    mt[mti] = ux | lx;
  }
}

double genrand(void)
{
  int i;
  unsigned long long x;
  static unsigned long long mag01[2]={0ULL, MATRIX_A};

  if (mti >= NN) { /* generate NN words at one time */
    /* if sgenrand() has not been called, */
    /* a default initial seed is used */
    if (mti == NN+1) sgenrand(1ULL);

    for (i=0;i<NN-M2;i++) {
      x = (mt[i]&UMASK) | (mt[i+1]&LMASK);
      mt[i] = (x >> 1) ^ mag01[(int)(x&1ULL)];
      mt[i] ^= mt[i+M0] ^ mt[i+M1] ^ mt[i+M2];
    }
    for (;i<NN-M1;i++) {
      x = (mt[i]&UMASK) | (mt[i+1]&LMASK);
      mt[i] = (x>>1) ^ mag01[(int)(x&1ULL)];
      mt[i] ^= mt[i+M0] ^ mt[i+M1] ^ mt[i+M2-NN];
    }
    for (;i<NN-M0;i++) {
      x = (mt[i]&UMASK) | (mt[i+1]&LMASK);
      mt[i] = (x>>1) ^ mag01[(int)(x&1ULL)];
      mt[i] ^= mt[i+M0] ^ mt[i+M1-NN] ^ mt[i+M2-NN];
    }
    for (;i<NN-1;i++) {
      x = (mt[i]&UMASK) | (mt[i+1]&LMASK);
      mt[i] = (x>>1) ^ mag01[(int)(x&1ULL)];
      mt[i] ^= mt[i+M0-NN] ^ mt[i+M1-NN] ^ mt[i+M2-NN];
    }
    x = (mt[NN-1]&UMASK) | (mt[0]&LMASK);
    mt[NN-1] = (x>>1) ^ mag01[(int)(x&1ULL)];
    mt[NN-1] ^= mt[M0-1] ^ mt[M1-1] ^ mt[M2-1];

    mti = 0;
  }

  x = mt[mti++];
  x ^= (x >> UU);
  x ^= (x << SS) & MASK_B;
  x ^= (x << TT) & MASK_C;
  x ^= (x >> LL);

  return ((double)x/((double)0xFFFFFFFFFFFFFFFFULL));
}

```


APPENDIX

A. CHARACTERISTIC POLYNOMIAL OF RECURRENCE (2)

Let $\{\mathbf{x}_k\}$ be a sequence generated by recurrence (2), and B be a $(nw - r) \times (nw - r)$ matrix such that

$$(\mathbf{x}_{k+n}, \mathbf{x}_{k+n-1}, \dots, \mathbf{x}_{k+1}^u) = (\mathbf{x}_{k+n-1}, \mathbf{x}_{k+n-2}, \dots, \mathbf{x}_k^u)B$$

holds for $k = 0, 1, \dots$, i.e., B is the state transition matrix for recurrence (2). The explicit form B is as follows:

$$B = \left(\begin{array}{cccc|cccc} 0 & I_w & 0 & 0 & & & & \\ 0 & 0 & I_w & 0 & & & & \\ \vdots & & & & \ddots & & & \\ I_w & & & & & & & \\ \vdots & & & & & & & \\ I_w & & & & & & & \\ \vdots & & & & & & & \\ I_w & & & & & & & \\ \vdots & & & & & & & \\ 0 & & & & & 0 & I_w & 0 \\ \hline 0 & & & & & 0 & 0 & I_{w-r} \\ \hline S & & & & & 0 & 0 & 0 \end{array} \right) \begin{array}{l} \leftarrow m_2\text{th block} \\ \leftarrow m_1\text{th block} \\ \leftarrow m_0\text{th block} \\ \leftarrow 0\text{th block} \end{array}, S := \begin{pmatrix} 0 & I_r \\ I_{w-r} & 0 \end{pmatrix} A,$$

where I_k denotes the identity matrix of size k for any positive integer k . Then the characteristic polynomial of recurrence (2) is $\det(tI_{nw-r} - B)$. By applying elementary transformations to $(tI_{nw-r} - B)$, we get

$$\begin{aligned} \det(tI_{nw-r} - B) &= F^{w-r}G^r + a_0F^{w-r}G^{r-1} + \dots + a_{r-2}F^{w-r}G \\ &\quad + a_{r-1}F^{w-r} + a_rF^{w-r-1} + \dots + a_{w-2}F + a_{w-1}, \end{aligned} \quad (8)$$

where $F = t^n + t^{m_2} + t^{m_1} + t^{m_0}$ and $G = t^{n-1} + t^{m_2-1} + t^{m_1-1} + t^{m_0-1}$.

ACKNOWLEDGMENTS

The author thanks M. Matsumoto for his valuable advice., and also P. L'Ecuyer for his many helpful suggestions.

REFERENCES

- COMPAGNER, A. 1991. The hierarchy of correlations in random binary sequences. *J. Stat. Phys.* 63, 883–896.
- COUTURE, R., L'ECUYER, P., AND TEZUKA, S. 1993. On the distribution of k -dimensional vectors for simple and combined Tausworthe sequences. *Math. Comput.* 60, 749–761.

- L'ECUYER, P. 1996. Maximally equidistributed combined Tausworthe generators. *Math. Comput.* 65, 213, 203–213.
- L'ECUYER, P. 1999a. Tables of linear congruential generators of different sizes and good lattice structure. *Math. Comput.* 68, 225, 249–260.
- L'ECUYER, P. 1999b. Tables of maximally equidistributed combined LFSR generators. *Math. Comput.* 68, 225, 261–269.
- LENSTRA, A. K. 1985. Factoring multivariate polynomials over finite fields. *J. Comput. Syst. Sci.* 30, 235–248.
- MATSUMOTO, M. AND KURITA, Y. 1994. Twisted GFSR generators II. *ACM Trans. Model. Comput. Simul.* 4, 3 (July), 254–266.
- MATSUMOTO, M. AND KURITA, Y. 1996. Strong deviations from randomness in m -sequences based on trinomials. *ACM Trans. Model. Comput. Simul.* 6, 2, 99–106.
- MATSUMOTO, M. AND NISHIMURA, T. 1998. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.* 8, 1, 3–30.
- NIEDERREITER, H. 1993. Factorization of polynomials and some linear-algebra problems over finite fields. *Linear Alg. Appl.* 192, 301–328.
- NIEDERREITER, H. 1995. The multiple-recursive matrix method for pseudorandom number generation. *Finite Fields Appl.* 1, 3–30.
- TEZUKA, S. 1990. Lattice structure of pseudorandom sequences from shift register generators. In *Proceedings of the 1990 Winter Conference on Simulation* (WSC '90, New Orleans, LA, Dec.). ACM Press, New York, NY, 266–269.
- TEZUKA, S. 1994a. The k -dimensional distribution of combined GFSR sequences. *Math. Comput.* 62, 206 (Apr.), 809–817.
- TEZUKA, S. 1994b. A unified view of long-period random number generators. *J. Oper. Res. Japan Soc.* 37, 211–227.
- TOOTILL, J. P. R., ROBINSON, W. D., AND EAGLE, D. J. 1973. An asymptotically random Tausworthe sequence. *J. ACM* 20, 469–481.

Received: July 1999; revised: March 2000; accepted: March 2000