Annales Mathematicae et Informaticae
56 (2022) pp. 95–108
DOI: https://doi.org/10.33039/ami.2022.12.003
URL: https://ami.uni-eszterhazy.hu

Generalized Middle-Square Method*

Viktória Padányi, Tamás Herendi

Department of Computer Science, Faculty of Informatics, University of Debrecen
padanyi.viktoria@inf.unideb.hu
herendi.tamas@inf.unideb.hu

Abstract. In this paper, we generalize John von Neumann's Middle-Square Method (MSM) to canonical number systems (CNS). Additionally, we present some observations and statistical tests of the sequences generated by the described generators.

Keywords: pseudorandom number generator, middle square method, canonical number system

AMS Subject Classification: 11K45, 11A63, 11B37, 62-08

1. Introduction

Pseudorandom number generators (PRNG) are often used in solving different theoretical and practical problems. The particular applications expect appropriate properties. The most important properties are the distribution of elements produced by the generators, the low correlation between the consecutive elements, and the large period length. In terms of usage, the speed, the resource requirements, and the qualities of the generators are interesting issues. A general approach for constructing pseudorandom number sequences is the following: the elements of the sequence are computed from the previous elements recursively. Recursion can be resolved by the use of a seed. The next seed is computed iteratively from the preceding seeds, and the random values are extracted from them.

John von Neumann's Middle Square Method is an interesting way to construct uniformly distributed PRNG since this was the first practical random number gen-

*The presented research has been partially supported by the SETIT Project (no. 2018-1.2.1-NKP-2018-00004), which has been implemented with the support provided by the National Research, Development and Innovation Fund of Hungary, financed under the 2018-1.2.1-NKP funding scheme. The research has been partially supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union and co-financed by the European Social Fund.

Submitted: October 27, 2022
Accepted: December 3, 2022
Published online: December 28, 2022

erator. In 1946 John von Neumann introduced the method (first published in [11]). It was simple and fast to execute with ENIAC. He used a recursive definition, where the initial value x_0 is some $2k$-digit decimal number. For $n > 0$ he defined $x_n = \lfloor x_{n-1}^2/10^k \rfloor \pmod{10^{2k}}$. The period length of the constructed sequence depends on the initial value. In general, the longest period has length at most 8^{2k}, but very often, it is much shorter. Practically, it is a rather weak generator. If the seed became 0, it is 0 for all consecutive members. N. Metropolis [9] investigated the MSM in binary number systems. He showed that in the case of 20-bit numbers, there are only 13 different cycles. The longest period amongst the 13 cases is 142, which is rather short. It is not obvious to recognize this short period because of the long preperiod.

A more detailed description of PRNGs can be found in [4] and [12].

In this paper, we also deal with canonical number systems. It has long been well-known that positive integers can be represented in a digital way. One of the generalizations of this was studied in [4, p. 189] by D. E. Knuth. He defined a number system for the Gaussian integers similar to the rational integers, where the number system had base $-1+i$. This was later further generalized by I. Kátai and J. Szabó [3], where the existence of other number system bases was proved but still in the Gaussian integers. Later, I. Kátai and B. Kovács in [2], B. Kovács in [6] and B. Kovács and A. Pethő in [7], and [8] even further generalized the definition of the CNS for the ring of algebraic integers. They also proved their existence. In [6], a simple condition is given for the construction of CNSs, but in general, it is not obvious to find them. For the sake of simplicity, we will focus on binary CNSs. A. Kovács [5] presented a complete description of binary canonical number systems of degrees not greater than 8. Later, P. Burcsi and A. Kovács [1] extended the results for CNSs of degrees $9, 10$ and 11. The arithmetic in these number systems is similar to the rational integers with the classical digit representation. However, the calculation of the next digit requires a more difficult reduction operation. Our discussion will focus on binary CNSs.

For instance, let α be a root of the polynomial x^2+x+2, e.g., $-\frac{1}{2}+i\frac{1}{2}\sqrt{7}$. One can prove that in the ring of algebraic integers $\mathbb{Z}[\alpha]$, every number can be written in the canonical form $\gamma = \sum_{i=0}^{h} c_i \alpha^i$, where $c_i \in \{0, 1\}$.

Example 1. Addition in a binary number system

				1	1	0	1	0	1
			+	1	0	1	1	1	1
				2	1	1	2	1	2
			-				1	1	2
				2	1	1	1	0	0
	-	1	1	2					
		-1	-1	0	1	1	1	0	0
+	1	1	2						
	1	0	1	0	1	1	1	0	0

Let $\gamma_1 = 110101$ and $\gamma_2 = 101111$ be 6-bit long binary numbers in the above representation. Then $\gamma = \gamma_1 + \gamma_2$ can be calculated according to Example 1. Here we used the fact that 1 1 2 represents 0.

In the following sections, we define the generalized version of the MSM in binary CNS and analyze some properties of these generators.

2. Definitions and preliminary results

In this chapter, we define some necessary notions and state important results.

Definition 2.1. Let A be a finite set, and u be a sequence over A. We say that $u \in A^\infty$ is periodic with period length $\varrho \in \mathbb{N}$, if there exists $\varrho_0 \in \mathbb{N}$, such that

$$u_{n+\varrho} = u_n \quad \text{for all} \quad n \geq \varrho_0 \ .$$

The smallest ϱ_0 and ϱ with the previous property will be called the preperiod and minimal period length of u, respectively.

If $n \geq \varrho_0$, then the subsequence $u_n, \ldots, u_{n+\varrho-1}$ is called a period of the sequence.

Remark 2.2. Let A be a finite set, $u \in A^\infty$, $0 < k \in \mathbb{N}$ and $F : A^k \to A$. If the sequence satisfies the recurrence defined by $u_n = F(u_{n-1}, \ldots, u_{n-k})$ for all $n \geq k$, then u is periodic with period length $\varrho \leq |A|^k$.

The following definition is the generalization of number systems for complex numbers given by I. Kátai and J. Szabó in [3].

Definition 2.3. Let R be an integral domain, $\alpha \in R$ and $N = \{n_1, \ldots, n_m\} \subseteq \mathbb{Z}$. The pair (α, N) is called a number system in R, if any $\gamma \in R$ has a unique representation in the form $\gamma = \sum_{i=0}^{h} c_i \alpha^i$, where $c_i \in N$ for all $0 \leq i \leq h$ and $c_h \neq 0$, if $h \neq 0$. The number system is called canonical, if $N = \{0, 1, \ldots, m-1\}$.

We will use the notation $L(\gamma, \alpha, N) = h+1$, i.e. the length of the representation of γ in the number system (α, N).

Theorem 2.4. *Let $p \in \mathbb{Z}[x]$ be an irreducible polynomial with $\deg(p) = n$, and $p(x) = a_n x^n + \cdots + a_0$ such that $1 = a_n \leq a_{n-1} \leq \ldots a_0$ and $2 \leq a_0$. Furthermore, let α be a root of p and $N = \{0, 1, \ldots, a_0 - 1\}$. Then (α, N) is a canonical number system in $\mathbb{Z}[\alpha]$.*

Proof. The theorem is proven in a more general setting in [6]. □

Let β be an algebraic number of degree $n \geqq 1$. Then $\beta^{(i)}$ denotes the i^{th} conjugates of β for all $i = 1, \ldots, n$.

Let $\alpha, \gamma \in \mathbb{Q}[\beta]$. For the sake of simplicity, we use the notation

$$|\log|_\alpha \gamma = \max_{1 \leq i \leq n} \frac{\log|\gamma^{(i)}|}{\log|\alpha^{(i)}|} \ .$$

Theorem 2.5. *Let β be an algebraic integer of degree $n \geqq 1$, and let (α, N) be a number system in $\mathbb{Z}[\beta]$. Then there exist effectively computable constants $C_1 = C_1(\alpha, N)$ and $C_2 = C_2(\alpha, N)$ depending only on α and N, such that*

$$|\log|_\alpha \gamma + C_1 \leqq L(\gamma, \alpha, N) \leqq |\log|_\alpha \gamma + C_2 \tag{2.1}$$

holds for every $0 \neq \gamma \in \mathbb{Z}[\beta]$.

Proof. The theorem is proven in [8]. □

Remark 2.6. John von Neumann's MSM uses squaring as the only arithmetic operation. We observe how the length of the numbers changes after squaring.

We fix α and the corresponding CNS, and we use the notation $C_1 = C_1(\alpha, N)$, $C_2 = C_2(\alpha, N)$ and $L(\gamma) = L(\gamma, \alpha, N)$.

For example in the usual binary representation $\alpha = 2$. The length of the binary representation of an integer n can be expressed by

$$L(n, 2) = \lfloor \log_2(n) \rfloor + 1 = \left\lfloor \frac{\log n}{\log 2} \right\rfloor + 1 \ ,$$

which means that $C_1 = 0$ and $C_2 = 1$.

With our simplified notation, equation (2.1) is simplified to

$$|\log|_\alpha \gamma + C_1 \leq L(\gamma) \leq |\log|_\alpha \gamma + C_2 \ . \tag{2.2}$$

Let $\gamma \in \mathbb{Z}[\beta]$ be an algebraic integer with length $L(\gamma)$. By (2.2),

$$|\log|_\alpha \gamma \leq L(\gamma) - C_1 \ , \tag{2.3}$$

and

$$L(\gamma) - C_2 \leq |\log|_\alpha \gamma \ . \tag{2.4}$$

Applying (2.2), (2.3) and (2.4) to the length of γ^2, we obtain

$$\begin{aligned} L(\gamma^2) &\geq |\log|_\alpha \gamma^2 + C_1 = 2|\log|_\alpha \gamma + C_1 \\ &\geq 2(L(\gamma) - C_2) + C_1 = 2L(\gamma) - 2C_2 + C_1 \end{aligned}$$

and

$$\begin{aligned} L(\gamma^2) &\leq |\log|_\alpha \gamma^2 + C_2 = 2|\log|_\alpha \gamma + C_2 \\ &\leq 2(L(\gamma) - C_1) + C_2 = 2L(\gamma) - 2C_1 + C_2 \ . \end{aligned}$$

With the notations $C_3 = C_1 - 2C_2$ and $C_4 = C_2 - 2C_1$, we have

$$2L(\gamma) + C_3 \leq L(\gamma^2) \leq 2L(\gamma) + C_4 \ . \tag{2.5}$$

We should remark that C_1 and C_3 may have negative values. In Section 4, we show some estimates on the values of C_3 and C_4 for different α's.

Since $L(\gamma)$ and $L(\gamma^2)$ are integers, thus C_3 and C_4 can be chosen to be integers without losing precision.

B. Kovács and A. Pethő in [8] prove not only the existence of the constants but also provide a way how to determine them. Their formula is explicit for C_1 but implicit for C_2. Based on the described method, we calculated the values of C_1, C_2, C_3, and C_4 for some polynomials.

By the proof of the Thoerem of [8]

$$C_1 = \min_{1\le i\le n} \frac{\log\left(\left|\alpha^{(i)}\right| - 1\right) - \log(a_0 - 1)}{\log\left|\alpha^{(i)}\right|} \ .$$

For the determination of C_2, one has to compute first some intermediate bounds

$$C_{2,i} = \frac{a_0 - 1}{\left|\alpha^{(i)}\right| - 1} \ .$$

Now, let

$$\Gamma = \left\{ \delta \,|\, \delta \in \mathbb{Z}[\alpha],\ \left|\delta^{(i)}\right| \le C_{2,i} \right\} ,$$

and

$$C_2 = \max_{\delta\in\Gamma} L(\delta, \alpha) \ .$$

In the following, we show the values of the constants for some binary number systems. The structures are defined by $\mathbb{Z}[\alpha]$, where α's are given by their defining polynomials $p(x)$. In these computations, the symbol $\mathbf{i}$ denotes the imaginary unit (everywhere else in the paper, i is an integer).

$p(x) = x^2 + x + 2$:

$$\alpha_{1,\,2} = -\frac{1}{2} \pm \mathbf{i}\frac{1}{2}\sqrt{7} \qquad |\alpha_1| = |\alpha_2| = \sqrt{2}$$
$$C_{2,1} = C_{2,2} \approx 2.41$$
$$C_1 \approx -2.54 \qquad C_2 = 6$$
$$C_3 = -12 \qquad C_4 = 10$$

$p(x) = x^2 + 2x + 2$ (the case of Gaussian integers, considered by Knuth in [4, p. 189]):

$$\alpha_{1,\,2} = -1 \pm \mathbf{i} \qquad |\alpha_1| = |\alpha_2| = \sqrt{2}$$
$$C_{2,1} = C_{2,2} \approx 2.41$$
$$C_1 \approx -2.54 \qquad C_2 = 8$$
$$C_3 = -16 \qquad C_4 = 12$$

$p(x) = x^3 + x^2 + x + 2$:

$$\alpha_1 \approx -1.35 \qquad \alpha_{2,\,3} \approx 0.18 \pm \mathbf{i} \cdot 1.20$$
$$C_{2,1} \approx 2.83 \qquad C_{2,2} = C_{2,3} \approx 4.64$$
$$C_1 \approx -7.85 \qquad C_2 = 13$$

$$C_3 = -31 \qquad C_4 = 27$$

$p(x) = x^4 + x^3 + x^2 + x + 2$:

$$C_{2,1} = C_{2,2} \approx 3.97 \quad C_{2,3} = C_{2,4} \approx 7.72$$
$$C_1 \approx -16.77 \qquad C_2 = 21$$
$$C_3 = -46 \qquad C_4 = 32$$

In the last two cases, we detailed only some significant steps of the above-mentioned computation.

Related to the previously computed constants, we did some experiments. In Table 2, we collected the results of how the sizes of the squares changed after squaring in the considered canonical number systems. The set of four CNSs is extended by the two rational binary number systems with bases 2 and -2.

We consider all the numbers of 20 to 30 digits. For a given length h, the table contains the distances of the minimal and maximal lengths of the squares from the expected $2h$. Additionally, the average lengths of the squares are presented. The last column displays the theoretical bounds for the corresponding values of distances.

Studying the results, one may conjecture that the minimal and maximal lengths of the squares are considerably closer to the expected value of $2h$ than the analytical computations show. Another suspicion is that the average length of squares is close to $2h$, but increasing the degree of the base α increases the averages.

3. Arithmetic in canonical number systems

Let (α, N) be a CNS and $p(x) = a_n x^n + \cdots + a_0$ be the defining polynomial of α according to Theorem 2.4. The usual arithmetic of integers can be generalized to (α, N). The modified carry computation can be derived from p, described below.

Let $\beta \in \mathbb{Z}[\alpha]$ be the result of some arithmetical operation, and $\beta = \sum_{i=0}^{h} b_i \alpha^i$ is the representation without reduction. If for all $0 \le i \le h$, $b_i \in \{0, \ldots, a_0 - 1\}$ then β is represented in (α, N). Assume now that there exists $0 \le i \le h$ such that $b_i \notin \{0, \ldots, a_0 - 1\}$ and let j be the smallest such integer. Let $c \in \mathbb{Z}$ be such that $b_j = c \cdot a_0 + b'_j$ with $0 \le b'_j < a_0$. Since

$$0 = a_n \alpha^n + a_{n-1} \alpha^{n-1} + \cdots + a_0 \ ,$$

thus

$$\begin{aligned} \beta &= \sum_{i=0}^{h} b_i \alpha^i + c\alpha^j \cdot (a_n \alpha^n + a_{n-1} \alpha^{n-1} + \cdots + a_0) \\ &= \sum_{i=0}^{h'} b'_i \alpha^i \ , \end{aligned}$$

where $b_i = b'_i$ if $0 \leq i < j$, and $b'_j \in \{0, \ldots, a_0 - 1\}$.

In this new representation of β, either all coefficients are in $\{0, \ldots, a_0 - 1\}$ or the smallest k such that $b_k \notin \{0, \ldots, a_0 - 1\}$ satisfies $j < k$. It is proven in [7], that this iteration will terminate in finitely many steps, providing the unique, valid digit expansion of β in (α, N).

Based on the above observation, one can create an algorithm for the arithmetic operations in (α, N), similar to the usual carry computation used for rational integers.

By Theorem 2.4, the results of arithmetic operations have finite representation, whence the carry algorithm will always terminate.

Table 2. Lengths of squares

Length of base numbers

Digits	20	21	22	23	24	25	26	27	28	29	30	T
Defining polynomial: $x - 2$												
Decrease	1	1	1	1	1	1	1	1	1	1	1	1
Increase	0	0	0	0	0	0	0	0	0	0	0	0
Average	39.6	41.6	43.6	45.6	47.6	49.6	51.6	53.6	55.6	57.6	59.6	
Defining polynomial: $x + 2$												
Decrease	3	3	3	3	3	3	3	3	3	3	3	4
Increase	1	1	1	1	1	1	1	1	1	1	1	2
Average	38.9	40.9	42.9	44.9	46.9	48.9	50.9	52.9	54.9	56.9	58.9	
Defining polynomial: $x^2 + x + 2$												
Decrease	8	8	8	8	8	8	8	8	8	8	8	12
Increase	5	5	5	5	5	5	5	5	5	5	5	10
Average	39.6	41.6	43.6	45.6	47.6	49.6	51.6	53.6	55.6	57.6	59.6	
Defining polynomial: $x^2 + 2x + 2$												
Decrease	12	12	12	12	12	12	12	12	12	12	12	16
Increase	9	9	9	9	9	9	9	9	9	9	9	12
Average	40.6	42.6	44.6	46.6	48.6	50.6	52.6	54.6	56.6	58.6	60.6	
Defining polynomial: $x^3 + x^2 + x + 2$												
Decrease	11	14	14	14	14	14	14	14	14	14	14	31
Increase	12	12	12	12	12	12	12	12	12	12	12	27
Average	41.6	43.5	45.5	47.5	49.5	51.5	53.5	55.5	57.5	59.5	61.5	
Defining polynomial: $x^4 + x^3 + x^2 + x + 2$												
Decrease	17	18	18	18	18	18	20	20	20	20	20	46
Increase	21	21	21	21	21	21	21	21	21	21	21	32
Average	43.8	45.6	47.6	49.7	51.9	53.9	55.7	57.7	59.7	61.8	63.8	

4. Generalized Middle-Square Method

Using binary CNSs, we may generalize John von Neumann's MSM.

Let $p(x) \in \mathbb{Z}[x]$ be an irreducible polynomial of degree n, and with coefficients $1 = a_n \leq a_{n-1} \leq \cdots \leq a_0 = 2$. The corresponding CNS has only 2 digits: 0 and 1. For the sake of simplicity, we will call the digits bits and the digit representation of algebraic integers in $\mathbb{Z}[\alpha]$ as a binary representation.

In the design of the generator, we use a seed of $m \in \mathbb{N}$ bits. Similarly, as it is done in the original construction, let u be a sequence over $\mathbb{Z}[\alpha]$ defined by the following:

$u_0 \in$ is a random m-bit number;

if $k > 0$, let

$$u_{k-1}^2 = \sum_{i=0}^{h} b_i \alpha^i \ , \text{ with } b_h \neq 0 \ , \ t = \left\lfloor \frac{h-m}{2} \right\rfloor \text{ and}$$

$$u_k = \sum_{i=0}^{m-1} b_{i+t+1} \alpha^i \ .$$

The value of m should be chosen to be large enough, in particular such that $2m + C_3 > m$, i.e. $m > -C_3$, where C_3 is as defined in section 2.

Another approach is if $t = \lfloor \frac{m}{2} \rfloor$, but then $\frac{m}{2} > -C_3$ should hold.

5. Experimental results

This section provides some experimental results related to the Generalized Middle-Square Method (GMSM). We observe the periodicity properties for several base polynomials, particularly those studied in the previous sections.

Furthermore, some statistical tests – the distributions of moving averages, zero-crossing gaps, and frequency classes – are presented for the GMSM generators, where the arithmetics are derived from the polynomials $x^2 + x + 2$ and $x^4 + x^3 + x^2 + x + 2$. Comparison of the data – both optically and numerically – shows that increasing the degree of the polynomials improves the properties of the generated sequences.

Figure 1 displays the distributions of the moving average of the sequences.

We have initialized the sequences with randomly chosen integers. The sizes of the samples are 10^8. The seeds are 63-bit words, and the pseudorandom values are obtained by a reduction to the 14-bit prefixes (the least significant 49 bits are eliminated). The length of the window for the summation is 100.

We have used the following simple formula to compute the sequence of moving averages:

$$a_k = \frac{1}{100} \sum_{i=k}^{k+100} u_i \ ,$$

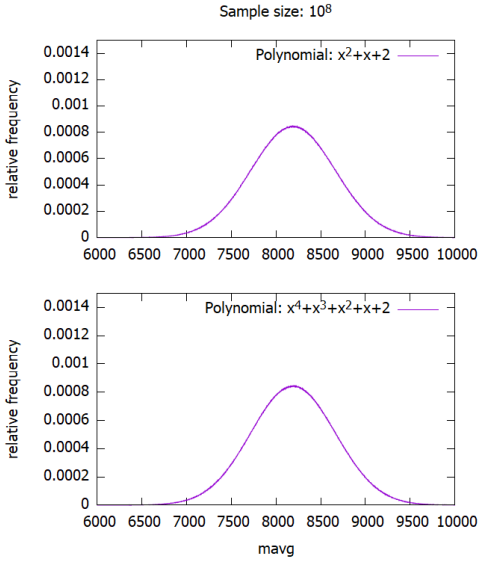

Figure 1. Moving average distribution

where (u_i) is the sequence generated by the GMSM.

Next, we observed the generators' behavior under the random walk test.

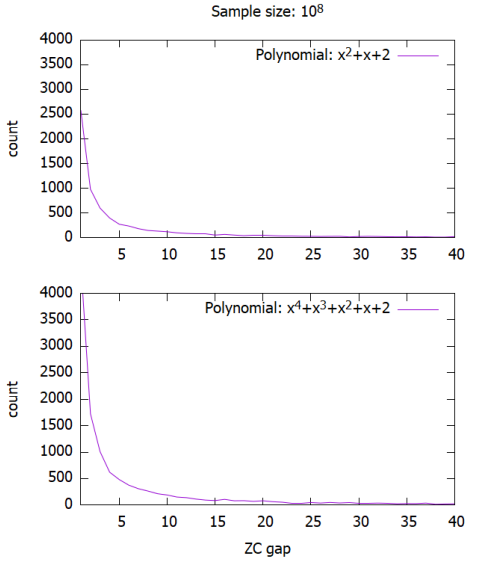

Figure 2. Random walk

The generated sequences are balanced around 0 by a shift with the mean value: $v_k = u_k - E(u)$. Using the new samples, we have computed the cumulative sums:

$$c_k = \sum_{i=0}^{k} v_i \ ,$$

The test calculates the frequency of the lengths of the gaps between consecutive zero crossings of c. The results are presented in Figure 2.

Finally, we have investigated to the distribution of the frequency classes. The values of the sequences are arranged into 2^{14} intervals of equal lengths (again, we reduce the random samples to the 14 most significant bits):

$$\mathrm{U}_i = \left\{ u_k \mid i = \left\lfloor \frac{u_k}{2^{49}} \right\rfloor \right\} \ , \text{ where } i \in \left\{0, \ldots, 2^{14} - 1\right\} \ .$$

Our objective is to describe the probability of the event when the same (reduced) random value appears exactly t times for a given t.

For normalization reasons, the minimum and maximum of the cardinalities are computed:

$$\begin{aligned} \min &= \min\left\{ |\mathrm{U}_i| \mid i = 0..2^{14} - 1 \right\} \text{ and} \\ \max &= \max\left\{ |\mathrm{U}_i| \mid i = 0..2^{14} - 1 \right\} \ . \end{aligned}$$

Figure 3 displays the distributions of the relative frequencies of the cardinalities of U_k.

The horizontal axis is normalized, and the plotted values are calculated according to the following formulas:

$$\begin{aligned} x_t &= \frac{t - \min}{\max - \min} \ , \\ y_t &= \frac{\left|\left\{ i \mid |\mathrm{U}_i| = t, 0 \le i < 2^{14} \right\}\right|}{10^8} \ . \end{aligned}$$

Although the above-presented graphs show good properties of the regarded generators, the investigation of a detailed statistical test provides a more accurate description of the behavior of the sequences. We have tested two of our generators with the NIST Statistical Test Suite (c.f. [10]). The results are summarized in Tables 4 and 5. These two are the MSMs corresponding to the polynomials x^2+x+2 and $x^4 + x^3 + x^2 + x + 2$. We denote them by GMSM1 and GMSM2 in the tables, respectively. In both sequences, we have used a 63-bit seed. The bit sequences for the tests are produced by simply writing the blocks of seeds bit by bit consecutively.

We compared the results with two of the NIST's built-in generators, the LCG and SHA1. The comparison shows that the properties of GMSM sequences are between the two built-in ones.

We used the default parameter adjustments in Table 3.

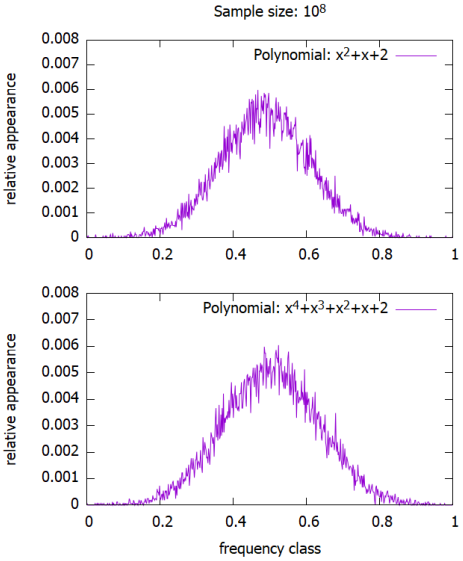

Figure 3. Frequency distribution

Table 3. NIST default settings

Test name	Block length
Block frequency	128
Non-overlapping template	9
Overlapping template	9
Approximate entropy	10
Serial	16
Linear Complexity	500

Both tests have the same arguments: the lengths of the sample sequences are 1000000, and the numbers of independent bitstreams are 1000. The level of acceptance is left to the default 0.01. In Table 4, one can see that both generators have an acceptable uniformity level on average.

Table 5 shows the ratio of the 1000 bitstreams accepted by the tests. Referring to the final report of the NIST test suite, "the minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately 0.981819", while "the minimum pass rate for the random excursion (variant) test is approximately 0.979456". Based on this recommendation, we may say that both generators have passed all tests.

Last but not least, in Table 6, we have collected the periodicity properties of the same GMSM sequences as in Table 2.

Again, one block corresponds to the CNS given by the defining polynomial of

its base. The entries are:
- the number of disjoint cycles;
- the maximal length of the cycles;
- the number of the length-1 cycles

for the different seed sizes. The trivial 0-cycle is excluded from the table.

Table 4. NIST test results: p-values

	p-value	
	GMSM1	GMSM2
Frequency	0.574903	0.142872
Block Frequency	0.936823	0.516113
Cumulative Sums	0.225069	0.484351
Runs	0.818343	0.761719
Longest Run	0.015707	0.674543
Rank	0.807412	0.552383
FFT	0.145326	0.368587
Non-Overlapping Template	0.511596	0.501944
Overlapping Template	0.248014	0.825505
Universal	0.152044	0.655854
Approximate Entropy	0.769527	0.353733
Random Excursions	0.292500	0.341976
Random Excursions Variant	0.480915	0.385875
Serial	0.145441	0.236631
Linear Complexity	0.492436	0.347257

Table 5. NIST test results: proportions

	Proportion	
	GMSM1	GMSM2
Frequency	0.9870	0.9890
Block Frequency	0.9890	0.9950
Cumulative Sums	0.9855	0.9890
Runs	0.9880	0.9890
Longest Run	0.9870	0.9900
Rank	0.9870	0.9860
FFT	0.9930	0.9870
Non-Overlapping Template	0.9905	0.9895
Overlapping Template	0.9860	0.9910
Universal	0.9920	0.9920
Approximate Entropy	0.9880	0.9850
Random Excursions	0.9853	0.9930
Random Excursions Variant	0.9866	0.9912

Table 6. All cycles in GMSM sequences

Nontrivial cycles

Digits (seed)	10	11	12	13	14	15	16	17	18	19	20
Defining polynomial: $x-2$											
Cycles	4	4	6	4	9	12	12	10	11	6	12
Max period length	5	5	10	2	56	70	111	203	197	2	142
Stability points	2	3	3	3	3	3	4	4	6	5	6
Defining polynomial: $x+2$											
Cycles	2	6	7	7	11	12	16	11	13	18	18
Max period length	3	3	2	34	10	27	51	30	2	39	4
Stability points	1	3	4	3	5	4	6	5	8	9	8
Defining polynomial: x^2+x+2											
Cycles	3	4	4	2	4	6	3	3	4	9	7
Max period length	2	2	10	19	10	13	34	21	13	256	476
Stability points	2	3	1	1	1	1	1	1	2	2	2
Defining polynomial: x^2+2x+2											
Cycles	2	4	6	5	5	7	5	4	7	12	13
Max period length	1	2	2	5	5	11	20	2	7	24	117
Stability points	2	3	4	2	2	2	2	3	5	9	8
Defining polynomial: x^3+x^2+x+2											
Cycles	10	13	6	6	3	1	5	6	7	11	5
Max period length	5	5	9	5	1	1	7	67	20	165	57
Stability points	8	10	4	5	3	1	3	3	3	3	1
Defining polynomial: $x^4+x^3+x^2+x+2$											
Cycles	5	8	6	5	5	7	10	11	6	6	8
Max period length	13	19	4	12	83	22	57	54	270	125	258
Stability points	2	1	3	3	3	3	6	7	2	2	3

The first block contains test results in the CNS with base 2, i.e., the simple binary representation of non-negative rational integers.

In the second block, the number system is the extension of the previous to the whole set of integers with base -2.

One must remark that even if they have small period lengths, the sequences can be used for pseudorandom number generators because of the long preperiod. Increasing the size of the seed increases the period length and the length of the longest period, but not in a monotonous way.

References

[1] P. Burcsi, A. Kovács: *Exhaustive search methods for CNS polynomials*, Monatshefte für Mathematik 155 (3) (2008), pp. 421–430.

[2] I. Kátai, B. Kovács: *Canonical number systems in algebraic number fields*, Acta Math. Hung. 37.1-3 (1981), pp. 159–164.

[3] I. Kátai, J. Szabó: *Canonical number-systems for complex integers*, Acta Sci. Math. 37 (1975), pp. 255–260, issn: 0001-6969.

[4] D. E. Knuth: *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, Boston: Addison-Wesley, 1981.

[5] A. Kovács: *Generalized binary number systems*, Annales Univ. Sci. Budapest, Sect. Comp 20 (2001), pp. 195–206.

[6] B. Kovács: *Integral domains with canonical number systems*, Publ. Math. 36.1-4 (1989), pp. 153–156, issn: 0033-3883.

[7] B. Kovács, A. Pethő: *Number systems in integral domains, especially in orders of algebraic number fields*, Acta Sci. Math. 55.3-4 (1991), pp. 287–299, issn: 0001-6969.

[8] B. Kovács, A. Pethő: *On a representation of algebraic integers*, Stud. Sci. Math. Hung. 27.1-2 (1992), pp. 169–172, issn: 0081-6906; 1588-2896/e.

[9] N. Metropolis: *Phase shifts — middle squares — wave equations*, Symposium on Monte Carlo methods, University of Florida (1954), pp. 29–36.

[10] National Institute of Standards and Technology: *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications: NIST SP 800-22*, 2012, url: https://www.nist.gov/publications/statistical-test-suite-random-and-pseudorandom-number-generators-cryptographic (visited on 02/19/2017).

[11] J. von Neumann: *Various Techniques Used in Connection with Random Digits. In: A.S. Householder, G.E. Forsythe, and H.H. Germond, eds., Monte Carlo Method, National Bureau of Standards*, Appl. Math. 12 (1950), pp. 36–38.

[12] V. Padányi, T. Herendi: *Metaanalysis of Pseudorandom Number Generators*, 23rd annual Spring Wind conference 23 (2020), pp. 474–486.