S. S. MAGLIVERAS, B. A. OBERG and A. J. SURKAN

A new random number generator from permutation groups

Estratto dai
« Rendiconti del Seminario Matematico e Fisico di Milano »
Vol. LIV (1984)

TIPOGRAFIA FUSI - PAVIA
6/1987

S. S. MAGLIVERAS, B. A. OBERG and A. J. SURKAN
dell'Università del Nebraska (USA)

A NEW RANDOM NUMBER GENERATOR FROM PERMUTATION GROUPS

(*Conferenza tenuta il 10 dicembre 1984*)
dal Prof. Magliveras

ABSTRACT. — We describe a new random number generator, RPGM, which is based on the cryptographic system PGM invented by Magliveras in 1976 and subsequently studied by Magliveras and Surkan [10]. PGM relies on a certain method of machine representation for permutation groups. This method allows for encryption and decryption algorithms based on a space-efficient data structure which is called a logarithmic signature for the group. The efficacy of RPGM is studied by means of an extensive analysis of generated data of 100,000 numbers using the Mathieu group M_{24} in its 5-transitive representation on 24 points.

I. INTRODUCTION

In 1976, while studying properties of finite simple groups of moderate to large size, S. Magliveras discovered a new symmetric cryptographic system. The system relies on a certain method of machine representation for permutation groups, which can also be extended to other types of representations, for example, to representations by matrices over a finite field. The method allows for encryption and decryption algoritms based on a space-efficient data structure which we call a *logarithmic signature* for the group. Logarithmic signatures were introduced by C. Sims [18], although not by this name, and were considerably refined and used by C. Sims, J. Cannon, J. Leon, and a large number of other researchers in the study of finite groups and their applications. From 1981 to 1984, Magliveras and Surkan studied the cryptosystem which by this time, for lack of a better name, had been christened PGM (Permutation Group Mappings). Although much work has been done to establish the strengths and weaknesses of PGM, the present paper discusses PGM as a random number generator.

A data encryption system, such as PGM, can be thought of as a family $\{(E_k, D_k) : k \in K\}$ of pairs of transformations of the message space $X = Z_n$ indexed by a set of keys K. Here, the size of the message space, n, is the order of the group G, currently being used. E_k and D_k are permutations of X so that E_k is the inverse of D_k. The ciphertext corresponding to a message $x \in X$ is simply $x^* = E_k(x) \in X$. To decrypt x^*, $x = D_k(x^*)$ is computed. To use PGM as a random number generator, we select a key $k \in K$, a seed $x_0 \in X$, and compute the sequence:

$$E_k(x_0),\ E_k(x_0 + 1),\ \ldots,\ E_k(x_0 + r - 1) \tag{1}$$

of r pseudorandom numbers. In the case of PGM, this method is preferable to computing:

$$x_0, x_1, \ldots, x_{r-1} \qquad \text{where} \quad x_i = E_k(x_{i-1}) \tag{2}$$

as (2) possesses cycles of unpredicable and much smaller lenghts than $n = |G|$, the cycle length of (1). In what follows, RPGM is understood to stand for process (1).

To test the feasibility of RPGM as a random number generator, data of 100,000 variates were generated by selecting fifty non-intersecting sequences each containing 2,000 variates. To apply a particular generic test to the data, the test is applied to each of the 50 variates which are expected to be distributed according to a known theoretical distribution. The 50 variates are subjected to two Kolmogorov-Smirnov tests (KS^+, KS^-) to determine how well they fit the theoretical distribution for that generic test. A battery of six in all generic tests were conducted as follows: 1) The « Kolmogorov-Smirnov » test for testing the uniformity of the distribution of the RPGM variates, 2) Three « chi-square » tests of the RPGM variates modulo 11, 13, and 101 respectively to test uniformity, 3) The « run » test to examine the distribution of frequency counts of run-up subsequences of various lengths, 4) The « gap » test used to examine the lengths of gaps between occurrences of x_j in some interval $[a, b]$, 5) The « maximum of t » test, to test the independence of subsequences of the form $x_{i+1}, \ldots, x_{i+t}$, and 6) The serial correlation test to test linear independence for time lags $\lambda = 1, 2, \ldots, 20$.

Like standard random number generators of the congruential type, RPGM creates sequences of numbers which correspond to elements in a particular group. In a congruential generator the abelian

group Z_n is generated sequentially as an orbit under the action of a single affine operator $x \to a \cdot x + b$, $a, b \in Z_n$, $a \neq 0$. The generating process in RPGM is completely different, the underlying group usually non-solvable, with much more irregular and unpredictable generated sequences. From a very short segment of a given congruentially generated sequence $(x_0, x_1, ..., x_s)$, $x_i = a \cdot x_{i-1} + b \pmod{n}$, one can easily determine a, b, n, and hence will be able to precisely predict the generated output. It is much harder to « break » RPGM from a given sequence $(x_0, x_1, ..., x_s)$, even in the most poorly designed cases.

II. The cryptosystem

In this section we describe « logarithmic signatures » of finite groups and exhibit encrypting and decrypting algorithms which have desirable cryptographic properties. These algorithms are subsequently used for constructing our new random number generator.

If G is a finite group and $G = G_0 > G_1 > ... > G_s = 1$ a strictly descending chain of subgroups of G, we define a *logarithmic signature* of G with respect to this chain to be a collection $B = \{B_1, ..., B_s\}$ of subsets of elements of G such that $B_i = \{u(i, j) : 0 \leq j \leq r(i) - 1\}$ is a complete set of right coset representatives of G_i in G_{i-1}. The subsets B_i are called the *blocks* of the logarithmic signature B. If G is a finite permutation group, specified by a set of generators, C. Sims [18] describes an algorithm which generates a logarithmic signature of G. The Sims algorithm relies on a theorem of Schreier's to recursively obtain generators for the subgroups G_i from generators of G_{i-1} and the list $B_i = \{u(i, j) : 0 \leq i \leq r(i) - 1\}$ of coset representatives. The block B_i is itself obtainable from the generators of G_{i-1}. The Sims algorithm can be easily adapted to other modes of machine representation of groups, for example, representations by matrices over a Galois field (see [3]) provided that computationally efficient algorithms exist for determining the coset $G_i u(i, j)$ to which an element x of G_{i-1} belongs.

If $B = \{B_1, ..., B_s\}$, $B_i = \{u(i, j) : 0 \leq j \leq r(i) - 1\}$ is a logarithmic signature of a group G, we call $r = (r(1), ..., r(s))$ the *vector* of B. Note that while $B = \{B_1, ..., B_s\}$ is a logarithmic signature for $G = G_0$, the set of blocks $B(k) = \{B_{k+1}, ..., B_s\}$ is a logarithmic signature for G_k. If the element $u(i, j)$ of $B_i \in B(k)$ is replaced by $h \cdot u(i, j)$ where $h \in G_i$, the resulting collection $B(k)^*$ forms a new

logarithmic signature for G_k. Moreover, any rearrangement of the elements of a block $B_i \in B(k)$ yields a new logarithmic signature for G_k. It follows that there are a total of

$$(3) \qquad \prod_{i=1}^{s} |G_i|^{r(i)} \cdot r(i)! = \prod_{i=1}^{s} \Big(\prod_{j=i+1}^{s} r(j) \Big)^{r(i)} \cdot r(i)!$$

logarithmic signatures for a group G with a vector $r = (r(1), \ldots, r(s))$. This number is of course astronomical. If we were to select for G the Mathieu group M_{24} of order 244,832,040, degree 24, and vector $r = (24, 23, 22, 21, 20, 3, 16)$, corresponding to a chain of stabilizers in its canonical representation, then the number of logarithmic signatures with respect to this chain is of the order of magnitude of 10^{612}.

The above procedure for generating new logarithmic signatures from a given one can be concisely described by considering a certain group action. If $B = \{B_1, \ldots, B_s\}$, $B_i = \{u(i, j) : 0 \leq j \leq r(i) - 1\}$ is a logarithmic signature of G with respect to the chain $G = G_0 > G_1 > \ldots > G_s = 1$ of subgroups, let $\mathcal{M}$ be the group of all matrices of the form:

$$M = \begin{bmatrix} H_1 & 0 & \ldots & 0 \\ 0 & H_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & H_s \end{bmatrix}$$

where H_i is an $r(i) \times r(i)$ monomial matrix with entries in G_i. This means that the H_i can be thought of as $r(i) \times r(i)$ permutation matrices whose unity entries have been replaced by arbitrary elements of G_i. The procedure described above for obtaining new logarithmic signatures of G now corresponds to acting on $(u(1, 0), \ldots, u(1, r(1) - 1); u(2, 0), \ldots, u(2, r(2) - 1); \ldots; u(s, 0), \ldots, u(s, r(s) - 1))$ on the left by some $M \in \mathcal{M}$, i.e.

$$\begin{bmatrix} v(1,0) \\ \vdots \\ v(s, r(s) - 1) \end{bmatrix} = M \begin{bmatrix} u(1,0) \\ \vdots \\ u(s, r(s) - 1) \end{bmatrix}$$

Thus, the totality $L = L(G_0, G_1, \ldots, G_s)$ of logarithmic signatures with respect to $G_0 > G_1 > \ldots > G_s$ is an $\mathcal{M}$ orbit. We observe that

since only the identity of $\mathcal{M}$ fixes a logarithmic signature in L, $\mathcal{M}$ acts regularly on L.

If $x \in G_{i-1}$ then $x \in G_i u(i, j)$ for some $u(i, j) \in B_i$, and therefore, $x = y \cdot u(i, j)$ for some $y \in G_i$. If it is computationally efficient to determine the coset $G_i u(i, j)$ in which x lies, equivalently determine the corresponding coset representative $u(i, j)$, the decomposition $x = y \cdot u(i, j)$ can be achieved by computing $y = x \cdot u(i, j)^{-1}$. It follows that each element $g \in G$ has a unique factorization into elements of the logarithmic signature, one factor per block:

$$g = u(s, P_s) \ldots u(2, P_2) \cdot u(1, P_1) \tag{4}$$

Conversely, if $(P_1, \ldots, P_s)$ is an s-tuple of non-negative integers with $0 \leq P_1 \leq r(i) - 1$ then the element $g = u(s, P_s) \ldots u(1, P_1)$ belongs to G, and as $(P_1, \ldots, P_s)$ ranges over all possible indices $(P_1, \ldots, P_s)$ with $0 \leq P_i \leq r(i) - 1$, g scans G, so that each element of G will be obtained exactly once.

If B is a logarithmic signature for G, with vector $r = (r(1), \ldots, r(s))$ and if $Q = \{(P_1, \ldots, P_s) : P_i \in Z,\ 0 \leq P_i \leq r(i) - 1\}$, then we define mapping $\beta_B : G \to Q$ as follows: If $g \in G$, express g in its unique factorization as $g = u(s, P_s) \ldots u(1, P_1)$, and define $\beta_B(g) = (P_1, \ldots, P_s)$. It is immediate that β_B is a bijection of G onto Q. Both β_B and β_B^{-1} are computable with extreme efficiency in the case of permutation groups (see [4], [18]). In this case G is a permutation group acting on the set $\Omega = \{1, 2, \ldots, n\}$ and G_k is chosen to be the pointwise stabilizer of $\{1, 2, \ldots, k\}$. If $D = \{k = d_1, \ldots, d_{r(k)}\}$ is the orbit under G_{k-1} of the letter $k \in \Omega$ then a complete set of right coset representatives of G_k in G_{k-1} can be chosen by selecting one element $u(k, j) \in G_{k-1}$ for each $d_j \in D$, such that $u(k, j)$ carries $k = d_1$ to d_j. Furthermore, an element $x \in G_{k-1}$ belongs to $G_k u(k, j)$ if and only if $x(k) = [u(k, j)]\ (k) = d_j$, a very fast test. Now, given $r = (r(1), \ldots, r(s))$, let:

$$m_1 = 1,\ m_i = \prod_{j=1}^{i-1} r(i) \qquad \text{for } 2 \leq i \leq s, \tag{5}$$

If X is the set of non-negative integers $\{0, \ldots, |G| - 1\}$, then the function μ: $Q \to X$, defined by

$$\mu(P_1, \ldots, P_s) = \sum_{i=1}^{s} m_i P_i \tag{6}$$

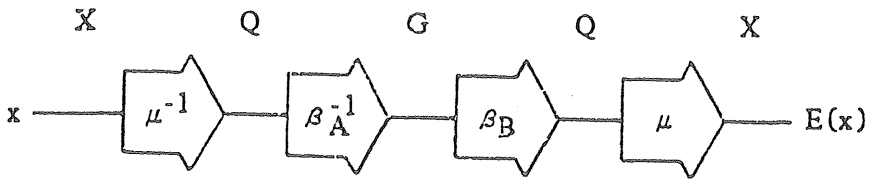
is a bijection, with $\mu^{-1}(m)$ easily and efficiently computable by successive subtractions (knapsack with super-increasing vector).

We are now ready to describe the encryption and decryption algorithms. Select a pair of logarithmic signatures (A, B) for a group G of sufficiently large size to accomodate the message space. For a given $x \in X$, compute the encrypted message $E(x)$ by:

$$[\mu\, \beta_B\, \beta_A^{-1}\, \mu^{-1}]\,(x) \tag{7}$$

Schematically:

Space:



The decryption function is of course $D = \mu\beta_A\, \beta_B^{-1}\mu^{-1}$. Thus, we see that a key in the system consists of an ordered pair of logarithmic signatures of the group and decryption consists of encrypting with the pair of logarithmic signatures interchanged. To emphasize the dependence of E and D on the two logarithmic signatures involved, we denote the two functions by $E_{A,B}$ and $D_{A,B}$ respectively.

A *logarithmic basis* for a group G is a collection $B = \{B_1, ..., B_s\}$ of blocks $B_i = \{u(i, j) : 0 \leq j \leq r(i) - 1\}$ of elements of G such that each element $g \in G$ has a unique representation of the form $g = u(s, P_s) \ldots u(1, P_1)$ with $0 \leq P_i \leq r(i) - 1$. There exist a plethora of logarithmic bases for a group which are not logrithmic signatures with respect to some subgroup chain, and these can be readily used as the first members of pairs (A, B) in encryption functions $E_{A,B}$. However, unless B is a logarithmic signature corresponding to a stabilizer chain of subgroups, the decoding process β_B cannot be effected without an unusual amount of computation. Similarly, if A is a general logarithmic basis, then $D_{A,B}$ is rendered non-feasible since β_A is not easily computable. We may still use the function $E_{A,B}$ with A an arbitrary logarithmic basis for purposes of random number generation, since the decryption process $D_{A,B}$ is not needed. We defer discussing questions about the nature of logarithmic bases to another paper which will deal with the algebraic properties of PGM.

III. THE RPGM GENERATOR

In this section we specify our test RPGM system by means of exhibiting all the parameters necessary for its reconstruction. The underlying group is $G = M_{24}$, in its 5-transitive representation on 24 points. We begin by exhibiting our specific copy of M_{24} in the symmetric group S_{24}. This is done by specifying a set of generators and an initial logarithmic signature A for G which is easily described in terms of these generators. Let:

a = (1 7 22 13 3 16 18 19 9 20 5 11 12 14 10 15 4 23 8 17 6 21 2)

b = (1 24) (2 6) (3 12) (4 16) (8 20) (10 19) (13 17) (18 21)

$c = a^b = b^{-1}ab$

d = (3 10 18 15 14 21 20 22 5 11 7) (4 9 8 12 19 23 17 16 13 24 6)

e = (3 4) (6 7) (9 23) (10 22) (11 21) (12 24) (13 14) (17 18)

f = (4 19 14 16 12 20 11) (5 23 9 18 8 13 7) (6 10 17 24 15 22 21)

g = (4 5 22) (6 15 21) (7 17 23) (8 11 24) (10 16 20) (12 18 13)

h = (5 23 15 9 20) (6 22 14 10 18) (7 24 16 11 17) (8 21 13 12 19)

x = (5 22 8 24) (6 23 7 21) (9 11) (10 12) (13 19 16 17) (14 18 15 20)

y = (1 3 4 2) (6 8 7) (9 16 15 20 10 21 12 11 19 24 13 23 22 18 14)

z = (9 17) (10 20) (11 18) (12 19) (13 21) (14 23) (15 24) (16 22)

The collection $A = \{A_1, ..., A_7\}$, where:

$A_1 = (1, a, a^2, \ldots, a^{22}, b)$;

$A_2 = (1, c, c^2, \ldots, c^{22})$;

$A_3 = (1, d, d^2, \ldots, d^{10}, e, ed, ed^2, \ldots, ed^{10})$;

$A_4 = (1, f, f^2, \ldots, f^6, g, gf, \ldots, gf^6, g^2, g^2f, \ldots, g^2f^6)$;

$A_5 = (1, h, \ldots, h^4, x, xh, \ldots, xh^4, x^2, x^2h, \ldots, x^2h^4, x^3, x^3h, \ldots, x^3h^4)$;

$A^6 = (1, y^5, y^{10})$;

$A^7 = (1, z, z^y = y^{-1}zy, z^{(y^2)}, \ldots, z^{(y^{14})})$;

is easily seen to be a logarithmic signature for G.

In terms of the logarithmic signature A described above, we define two other logarithmic signatures B and C of G by listing their members. To specify an element $g \in G$, we simply list the integer $x = \mu\beta_A(g)$. The reader can recover g from x by computing $[\beta_A^{-1}\ \mu^{-1}]\ (x)$. The two logarithmic signatures B and C are listed in Table 1.

TABLE 1.

Logarithmic Signature B

136864145	209113318	75956997	5195456	57778306	172310255	81475332
234270541	157389943	194499845	149396976	121865139	31898934	176522833
114367900	4812899	226781347	60424912	217821471	133105148	67651386
124545897	222143114	208927766				
69939000	32544744	201633240	1311216	19043544	3462504	121911432
146031504	138491040	109242576	20962752	108358944	19231872	239034072
174650184	38177592	64418280	112652136	16886736	52606992	192436104
203417352	237102480					
168941256	70726656	158604504	235133232	129496440	120385128	55213800
230452824	51082080	104949000	175200936	240387168	171879552	153858960
236373024	93912864	62224752	83220072	35256792	54922896	68830536
43242576						
27785472	89343408	123953808	54550848	83939328	188523456	151897152
145290816	59493456	13662000	48879600	165862752	38702928	42030384
97382736	100479456	132661056	222259488	128131344	38885088	106114272
220340736	66816288	68601456	131592384	159390000	20274480	42333984
191778048	160920144	65541168	179281872	226206288	43354080	143068464
45904320	207334512	35193312	166785696	208864656	185657472	
96909120	163215360	107110080				
30602880	0	168315840	137712960	198918720	107110080	15301440
214220160	61205760	229521600	153014400	45904320	91808640	183617280
122411520	76507200					

Logarithmic Signature C

31244432	183603408	154658650	182802117	127794801	44763260	2549580
63473809	60909760	88541261	187707139	113454567	97236462	233005563
206331114	208892399	219176784	108123445	6327172	9621806	170538035
9390209	78459559	174348002				
95439576	15565152	17574312	161107824	149924376	195271968	139908048
55426368	85551984	45771744	165214920	129196104	216069480	152758512
92664984	134472816	62597376	45139224	97631520	229135344	209603040
45286248	108756072					
242808240	169341456	128947752	227619408	182758920	41100264	29768256
202537632	84354432	45670824	63132240	131181144	160441008	118788744
68673768	184610880	28460568	239188776	16230456	3410808	235155312
241334400						
154228800	217851216	7249968	144938640	191559456	147658896	231003168
89027664	183738720	113874288	234269904	154896720	129952944	41751072
210212640	119824848	10443840	166579248	196380624	181807824	169250928
127001952	74977056	28052640	46924416	142048368	196623504	38763648
68091408	187697664	155309616	195093360	69621552	25757424	32133024
199683792	229266576	102519648	211159872	131337360	142813440	
188717760	71406720	198918720				
15301440	76507200	45904320	0	91808640	61205760	122411520
229521600	107110080	137712960	168315840	30602880	198918720	183617280
214220160	153014400					

To produce a sequence of random numbers we now use the encryption transformation $E_{B,C}: x \to E_{B,C}(x) = [\mu\beta_C\,\beta_B^{-1}\,\mu^{-1}]\,(x)$ and compute the sequence:

$$E_{B,C}(S_i),\ E_{B,C}(S_i+1),\ldots,\ E_{B,C}(S_i+1999)$$

for a given random seed S_i. We obtain fifty sequences of length 2,000 from the seeds listed in Table 2. To make it possible for the reader to reproduce the data and check the system, we list not only the seeds S_i, but also their transforms $E_{B,C}(S_i)$ in Table 2.

TABLE 2.

Seeds

193687836	170924885	4468278	183118541	28039197	219105939	160266898
84681283	93622744	52648559	81107152	4354169	223856560	192299569
96325068	178715303	203581182	214903107	31731082	84077637	230638436
88672092	86191815	19746089	140011012	190077444	206678269	120769115
200096338	157354879	106589028	86253992	85480977	68928027	225019569
153295299	182288871	38569130	190083383	61683948	149844614	209224824
77022538	148960003	30828392	64109493	30855037	49653944	183744271
17370085						

Images

60232788	28102167	184289166	225407319	728260	74451264	52327513
133510109	119119306	103892779	105492754	199003016	107291722	133179616
153447732	240669811	1674510	2989584	40932433	113996980	119932838
5552700	175588676	57643472	27892370	159286548	66564151	177798421
162303001	210610361	190464756	111218422	210183801	65948832	166418097
169837800	95528948	51931587	221538931	207835236	47474706	139661829
169538641	121408181	27567142	162847660	51915031	189337846	55016201
198014311						

IV. THE STATISTICAL TESTS

We proceed to analyze the data generated by the system discussed in the previous section. Only a brief account of the aims and scope of each test is given here. All of the tests used are in Knuth [9], where a comprehensive description can be found for each. The set of seeds $\{S_i : 1 \leq i \leq 50\}$ were chosen so that when $i \neq j$, then $|S_i - S_j| > 2000$. Since $E_{B,C}$ is a permutation, it follows that the sets of images $E_{B,C}[S_i, S_i + 1999]$ form mutually disjoint samples. In whit follows, we denote the data by the matrix $(X_{i,j})$ where $X_{i,j} = E_{B,C}(S_i + j - 1)$, $1 \leq i \leq 50$, $1 \leq j \leq 2000$. Furthermore, for $1 \leq i \leq 50$, X_i denotes the sequence $(X_{i,1}, \ldots, X_{i,2000})$, and Y_i the real sequence $X_i/|M_{24}|$.

The Kolmogorov-Smirnov Test

When we are confronted with the task of deciding whether or not certain data belong to a particular continuous theoretical distri-

bution, the Kolmogorov-Smirnov (KS) test [9] may be used to estimate the degree of closeness of fit. There are two inputs to the KS test:

(i) a collection $\{V_1, ..., V_n\}$ of observed data, and

(ii) a continuous theoretical distribution $F(x) = Pr(V \leq x)$.

The KS test outputs two statistics, KS_n^+ and KS_n^-, which can be dfined in terms of $f(x) =$ [Number of $V_i \leq x$] by:

$$KS_n^+ = \sqrt{n} \max_{1 \leq i \leq n} (f(V_i)/n - F(V_i))$$

(8)

$$KS_n^- = \sqrt{n} \max_{1 \leq i \leq n} (F(V_i) - (f(V_i) - 1)/n)$$

These two statistics, KS_n^+ and KS_n^-, have the same theoretical distribution KS_n, which for large n, say $n > 1000$, is closely approximated by the distribution:

$$KS_\infty(x) = 1 - e^{-2x^2}, \qquad x \geq 0 \tag{9}$$

There in one instance where we will need a closed form for the continuous distribution of KS_n variates, but this is for $n = 2000$, so that we can readily use KS_∞ above. For other instances where we use the KS_n distribution, the probability tables for specific values of the parameter n are incorporated in Table 3.

TABLE 3. — The KS_n Distributon.

n \ Pr	0.01	0.05	0.25	0.50	0.75	0.95	0.99
5	0.02152	0.0947	0.3249	0.5245	0.7674	1.1392	1.4024
10	0.02912	0.1147	0.3297	0.5426	0.7845	1.1658	1.4440
50	0.04968	0.1403	0.3581	0.5675	0.8100	1.1999	1.4891
∞	0.07089	0.1601	0.3793	0.5887	0.8326	1.2239	1.5174

Typically, a test is performed on each of the 50 sequences X_i (or Y_i). This results in 50 variates which, under the hypothesis of randomness, should be distributed according to a known continuous distribution. The 50 variates and the theoretical distribution are

submitted to a KS test, and the values KS_{50}^{+} and KS_{50}^{-} are used as a criterion for determining whether or not the test has « passed ». Very high KS values indicate a bad fit of the 50 variates to their expected theoretical distribution, thus contradicting the hypothesis of randomness of the original data. On the other hand, very low KS values indicate behavior that fits the distribution too closely to be truly random. Thus, the values KS_{50}^{+} and KS_{50}^{-} are favorable to the RPGM generator if they fall somewhere in the middle of the distribution. Our acceptance region is set, in advance, to be the probability interval [0.05, 0.95]. For easy reference, we exhibit the corresponding probabilities along with the computed KS values.

$KS + KS$

Here we test the hypothesis of uniformity of the real RPGM variates $Y_{i,j} = X_{i,j}/|M_{24}|$. Under this hypothesis, each of the fifty sequences $Y_i = (Y_{i,1}, ..., Y_{i,2000})$ is supposed to adhere to the continuous uniform distribution:

$$F(x) = \begin{cases} 1 \text{ if } x \geq 1 \\ x \text{ if } 0 < x < 1 \\ 0 \text{ otherwise} \end{cases} \tag{10}$$

Using each of the fifty sequences Y_i and the continuous distribution $F(x)$, we compute the sets $K^{+} = \{KS_{2000}^{+}(1), ..., KS_{2000}^{+}(50)\}$ and $K^{-} = \{KS_{2000}^{-}(1), ..., KS_{2000}^{-}(50)\}$. These KS statistics should be distributed according to the theoretical distribution $KS_{2000} \simeq KS_{\infty}$. The adherence of the sets K^{+} and K^{-} to KS_{∞} is tested by means of two KS tests yielding the following four KS_{50} statistics: KS^{++}, KS^{+-}, KS^{-+}, and KS^{--}, having obvious meanings (KS^{+-} is the KS^{-} statistic run on the set K^{+}). The four values obtained are:

Variate	*Value*	*Probability*
KS^{++}	0.954	$0.75 < p < 0.95$
KS^{+-}	0.334	$0.15 < p < 0.25$
KS^{-+}	0.464	$0.25 < p < 0.50$
KS^{--}	0.625	$0.50 < p < 0.75$

$\chi^2 + KS$

Given a sample of size n, $\{V_1, ..., V_n\}$ of a discrete random variable V which takes exactly t distinct values $\{a_1, ..., a_t\}$ with corresponding probabilities $\{p_1, ..., p_t\}$, let c_i be the number of variates in $\{V_1, ..., V_n\}$ which take the value a_i. Then, the statistic

$$\chi^2 = \left[\frac{1}{n} \sum_{1 \le i \le t} \left(\frac{c_i^2}{p_i}\right)\right] - n \tag{11}$$

has the χ^2 distribution with $t - 1$ degrees of freedom.

A closed form formula for the cumulative distribution of a χ^2 with d degrees of freedom is not generally known. However, when d is even, the cumulative χ^2 distribution is related to the Poisson distribution (see [13]) by:

$$F(x) = 1 - \sum_{i=0}^{(d/2)-1} \frac{e^{-(x/2)} (x/2)^i}{i!} \tag{12}$$

For each of the samples X_i, $1 \le i \le 50$, we compute the variates $x_i = \{x_{i,j} = X_{i,j} \pmod{11} : 1 \le j \le 2000\}$. Under the hypothesis of uniformity for the sequences X_i, the variates $x_{i,j}$ take the $t = 11$ values in $Z_{11} = \{0, ..., 10\}$ with the associated probabilities $p_l = 1/11$, $0 \le l \le 10$. For each of the fifty samples x_i, we compute the corresponding statistic $\chi^2(i)$, with 10 degrees of freedom, and proceed to subject $\{\chi^2(1), ..., \chi^2(50)\}$ to a KS test using (12) with $d = t - 1 = 10$. The above procedure is repeated with moduli 13 and 101 replacing 11. Thus, in the last case $x_{i,j} = X_{i,j} \pmod{101}$, $p_l = 1/101$, and the corresponding χ^2 statistics have 100 degrees of freedom. The above KS tests on the three sets of fifty χ^2 variates yield the following KS_{50} statistics:

Variate	*Value*	*Probability*
KS^{11+}	2.879	$p > 0.99$
KS^{11-}	0.105	$0.01 < p < 0.05$
KS^{13+}	0.868	$0.75 < p < 0.95$
KS^{13-}	0.278	$0.05 < p < 0.25$
KS^{101+}	0.212	$0.05 < p < 0.25$
KS^{101-}	1.027	$0.75 < p < 0.95$

Run + KS

The « run » test counts the number of monotone increasing runs of lengths $l = 1, 2, ..., t-1$, and $l \geq t$ in a sequence of variates $(V_1, ..., V_n)$. A *run* is defined to be a *maximal* monotone increasing subsequence of $(V_1, ..., V_n)$, and its *length* the number of elements in the subsequence. To insure the independence of runs of various lengths, once a run $(V_j, ..., V_{j+l-1})$, i.e. a value, V_{j+l}, which is less than its pedecessor, V_{j+l-1}, is found, the entry V_{j+l} is deleted and run length counting resumes beginning with the sequence entry V_{j+l-1}.

Under the hypothesis of randomness of the generated data, the expected number of runs of length r is $n \cdot (1/r! - 1/(r+1)!)$, where n is the total number of runs found (see Knuth [9]). A χ^2 test is performed on the t categories of run counts, with the result having $t-1$ degrees of freedom.

For each sequence, X_i, the « run » test was performed with $t = 5$, the largest value which insures that each category will have an expected frequency ≥ 5. This results in fifty discrete variates $R = \{r_1, ..., r_{50}\}$, which under the hypothesis of randomness should adhere to the χ^2 distribution with $t - 1 = 4$ degrees of freedom. Using (12) and the data R, we conduct a KS test, yielding the KS_{50} statistics as follows:

Variate	*Value*	*Probability*
KS^+	0.9694	$0.75 < p < 0.95$
KS^-	0.1282	$0.01 < p < 0.05$

Gap + KS

Let $(V_1, ..., V_n)$ be a sequence of real variates with $V_i \in [0, 1]$, and let $0 \leq a < b \leq 1$. A *gap* of length l is defined to be a subsequence $(V_j, ..., V_{j+l})$ such that $V_{j-1}, V_{j+l} \in [a, b]$, but $V_i \notin [a, b]$ for $j \leq i \leq j + l - 1$. The « gap » test counts the number of gaps of various lengths $l = 1, 2, ..., t-1$, and $l \geq t$ in a sequence of variates. The choice of a and b is arbitrary, but if we let $p = b - a$, then, under the hypothesis that the data is random, the probability of a gap of length l occuring is $p(1-p)^{l-1}$, for $0 \leq l \leq t-1$, and the probability of a gap of length $\geq t$ occuring is $(1-p)^t$. A χ^2 test is

performed on the t categories of gap counts, with the result having $t-1$ degrees of freedom.

For each real sequence, Y_i, the « gap » test is performed with $t=7$, the largest value which insures that each category will have an expected frequency ≥ 5. The test is performed three separate times, once with $a=0$, $b=1/2$ (runs below the mean), again with $a=1/2$, $b=1$ (runs above the mean), and finally with $a=1/4$, $b=3/4$. Each of these tests yields a set of 50 χ^2 variates, which we denote by G^-, G^+, and G^* respectively. Under the hypothesis of randomness for the generated data, G^-, G^+, and G^* should adhere to the χ^2 distribution with $t-1=6$ degrees of freedom. A KS test on G^-, G^+, and G^*, yields the KS_{50} statistics KS^{-+} KS^{--}, KS^{++}, KS^{+-}, KS^{*+}, and KS^{*-} as follows:

Variate	*Value*	*Probability*
KS^{-+}	0.331	$0.05 < p < 0.25$
KS^{--}	0.544	$0.25 < p < 0.50$
KS^{++}	0.629	$0.50 < p < 0.75$
KS^{+-}	0.561	$0.25 < p < 0.50$
KS^{*+}	0.159	$0.05 < p < 0.25$
KS^{*-}	0.835	$0.75 < p < 0.95$

Max of t + KS + KS

Suppose that the time series $(V_1, \ldots, V_{kt})$ comes from a population which is uniformly distributed in $[0, 1]$, and suppose that in each subsequence of t consecutive terms $(V_{(j-1)t+1}, \ldots, V_{jt})$ the entries are mutually independent. Then, the set of variates $W=\{W_i : W = \max(V_{(i-1)t+1}, \ldots, V_{it}),\ 1 \leq i \leq k\}$ is easily seen to have the exponential distribution $F(x)=x^t$.

For $t=5$, we compute $Z=\{Z_i : 1 \leq i \leq 50\}$, where $Z_i=\{Z_{i,j} = \max(Y_{i,(j-1)t+1}, \ldots, Y_{i,jt}),\ 1 \leq j \leq 400\}$. Each of the Z_i should have the exponential distribution $F(x)=x^5$, and we could now compute 50 pairs of variates $(KS^+, KS^-)_i$, one pair for each Z_i, and these, under the hypothesis of independence mentioned above, should adhere to the KS_{400} distribution. A further KS test on these 100

variates could be run if we were able to compute $KS_{400}(x)$ for arbitrary x. Since KS_{400} is not available, but $KS_{2000} \simeq KS_\infty$ is, we merge five Z_i at a time to form 10 sequences of 2000 variates each as follows:

$$ZZ_m = Z_{(m-1)t+1} \cup Z_{(m-1)t+2} \cup \dots \cup Z_{mt} \tag{13}$$

$$1 \leq m \leq 50/t = 10$$

The variates in each ZZ_m still have the exponential distribution $F(x) = x^5$, but now the size of each ZZ_m is 2000, and a KS test can be conducted on each of the 10 sequences to yield KS variates $(KS^+, KS^-)_m$, $1 \leq m \leq 10$. These KS variates should adhere to the KS distribution $KS_{2000} \simeq KS$. We perform a further KS test on the 10 KS variates, producing four KS_{10} values: KS^{5++}, KS^{5+-}, KS^{5-+}, and KS^{5--}. An analogous procedure is performed on the Y_i with $t = 10$ and the Z_i merged into 5 sets, yielding the KS_5 values KS^{10++}, KS^{10+-}, KS^{10-+}, and KS^{10--}. These values are exhibited below.

Variate	*Value*	*Probability*
KS^{5++}	0.955	$0.75 < p < 0.95$
KS^{5+-}	0.317	$0.05 < p < 0.25$
KS^{5-+}	0.672	$0.50 < p < 0.75$
KS^{5--}	0.586	$0.50 < p < 0.75$
KS^{10++}	0.516	$0.25 < p < 0.50$
KS^{10+-}	0.829	$0.75 < p < 0.95$
KS^{10-+}	0.539	$0.50 < p < 0.75$
KS^{10--}	0.073	$0.01 < p < 0.05$

Serial Correlation test

If $V = (V_1, \dots, V_n)$ is a time series, the *autocorrelation* of V for lag λ is defined to be the correlation $r(\lambda) = r(V(t), V(t+\lambda))$, between the variables $V(t)$ and $V(t+\lambda)$. If $r(\lambda)$ is significantly

different from zero for $\lambda > 0$, then we would conclude that there is a high degree of linear dependence between $V(t)$ and $V(t+\lambda)$. Thus, a necessary condition for a good random number generator is that $r(\lambda)$ be very close to 0 for $\lambda \neq 0$. Unfortunately, we do not know what « significantly different from zero » means. This is so because even if the theoretical distribution of V is kown, the distribution of the $r(\lambda)$, for a fixed λ, is not known. For $\lambda = 1$, the mean and standard deviation of the distribution of the $r(1)$ are known and depend only on n:

$$(14) \qquad \mu_n = -1/(n-1),\ \sigma_n = (1/(n-1))\sqrt{n(n-3)/(n+1)}$$

but for a uniformly distributed V, the distribution of $r(1)$ is not known (See Knuth [9]). An empirical condition for $r(1)$ suggested by Knuth is that the random number generator is deemed to have passed the Serial Correlation test if $\mu - 2\sigma \leq r(1) \leq \mu + 2\sigma$.

For each of the fifty sequences $Y_i = (Y_{i,j} : j = 1, ..., 2000)$, we compute the autocorrelations $R_i = \{r_i(\lambda) : \lambda = 1, ..., 20\}$. We now tabulate the number of elements $r_i(1)$ which fall inside the interval $S = [\mu - 2\sigma,\ \mu + 2\sigma]$ and the number which do not. The results obtained are as follows:

[number of $r_i(1)$ inside S] $= 49$ [number of $r_i(1)$ outside S] $= 1$.

In addition to the above information, for each fixed $\lambda = 1, ..., 20$, we compute the mean and standard deviation:

$$(15) \qquad \bar{r}(\lambda) = (1/50)\sum_{i=1}^{50} r_i(\lambda),\ s(\lambda) = \left[(1/49)\sum_{i=1}^{50}(r_i(\lambda) - \bar{r}(\lambda))^2\right]^{1/2}$$

and exhibit the results in Figure 1.

Since no theoretical results are available on the distributions of the $r(\lambda)$, we perform the following comparative test. We generate, by means of the Honeywell-Multics APL resident random number generator, 50 samples of 2000 variates each. We submit these variates to the same autocorrelation computations as for the RPGM variates, and compare the performance of the two generators with respect to autocorrelations. For the APL generator:

[number of $r_i(1)$ inside S] $= 48$ [number of $r_i(1)$ outside S] $= 2$.

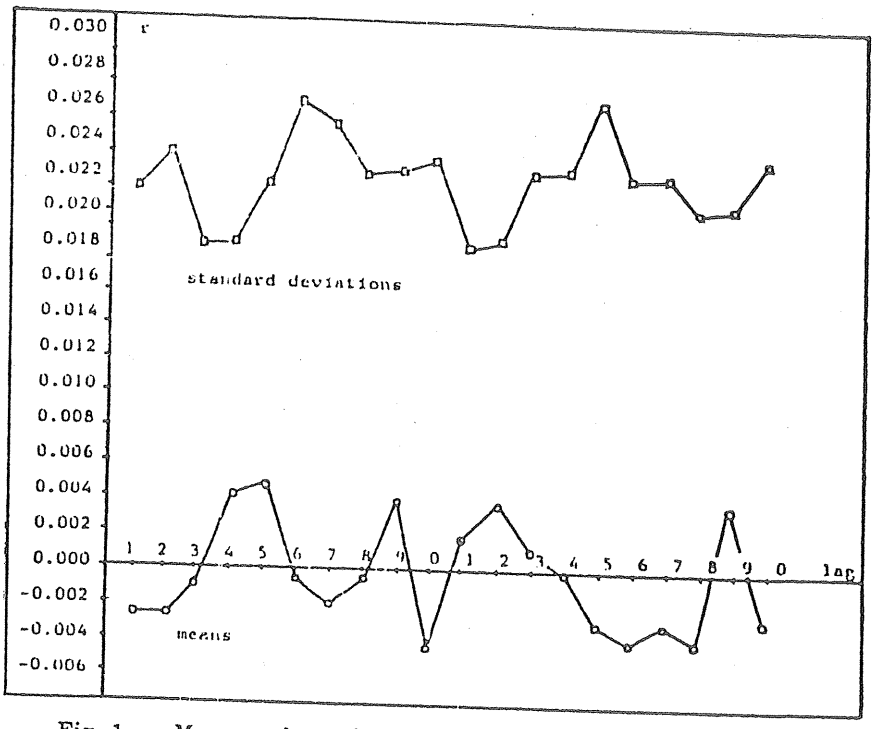


Fig. 1. — Means and standard deviations of autocorrelations r(*).

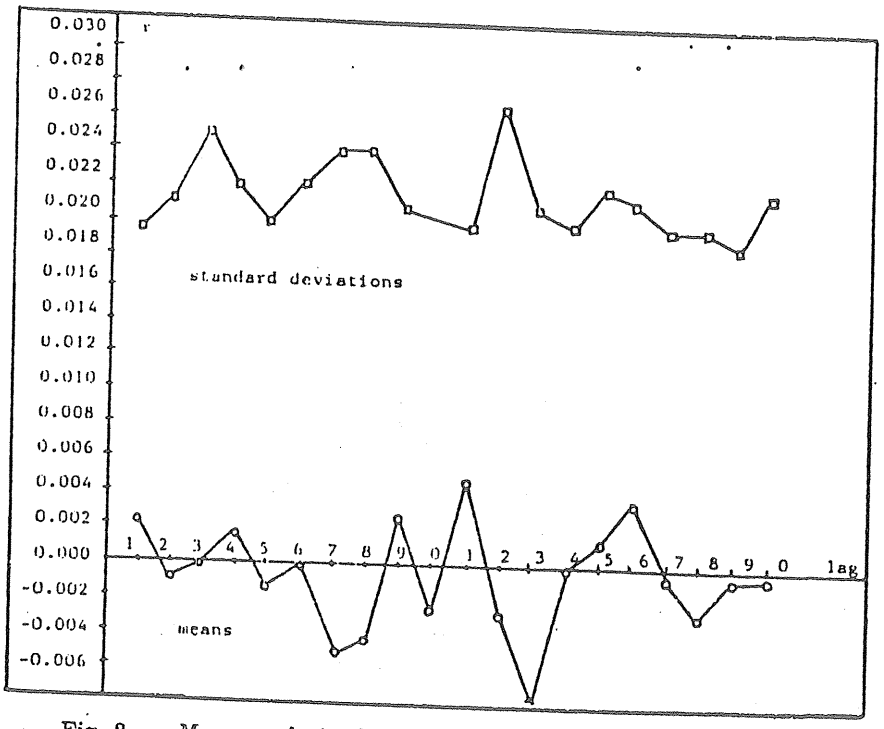


Fig. 2. — Means and standard deviations of autocorrelations r(*)
Multics APL generator.

V. Conclusions

We should mention at the outset that no effort was made to select a pair of logarithmic signatures which would give optimal results. Such a selection should be undertaken, of course, if a version of RPGM were to be put to actual use. Our results were obtained with the first pair of logarithmic singatures produced by our monomial shuffling procedure. The results invoke the following conclusions:

$KS + KS$

The real variates $Y_{i,j}$ were distributed uniformly with the final four KS statistics well in the central region of the KS_{50} distribution. The results here are excellent.

$\chi^2 + KS$

The results obtained in testing for uniformity of the $x_{i,j} = X_{i,j} \pmod m$ were excellent for $m = 13$ and 101, but poor for $m = 11$. The bad results for $m = 11$ were expected as it can be shown that when m is not coprime to the order of the group, certain periodic regularities are likely to occur in the $x_{i,j}$, depending upon seed selection and sample size. The user of RPGM therefore should avoid obtaining discrete variates in Z_m by reducing $X_{i,j}$ modulo m, with $gcd\,(m, |G|) \neq 1$.

$Run + KS$

The results of the run test were fair. The KS^+ value is central, falling nicely within the probability interval [0.75, 0.95]. The KS^- value, however, is marginally low, falling within the probability interval [0.04, 0.05]. This indicates a very tight adherence of the computed χ^2 variates to their expected distribution. While this is cause for suspicion of RPGM in regards to the run test, we prefer a close adherence to the theoretical distribution to extreme non-adherence indicated by high KS values. We should recall that many widely used random number generators tend to have runs which are longer than expected.

Gap + KS

RPGM passed the gap test with flying colors, as can be ascertained from the central nature of all six of the KS_{50} statistics.

Max of t + KS + KS

Out of the eight final *KS* statistics, seven were central and one had a rather low *KS* value, indicating too close adherence to the KS_5 distribution. This occurred for KS^{10--} for which the probability p satisfied $0.01 < p < 0.05$. We would consider these results very good to excellent.

Serial Correlation test

The results obtained under the serial correlation test were excellent for RPGM. Forty nine out of fifty corelations with $\lambda = 1$ fell into the interval $[\mu - 2\sigma, \mu + 2\sigma]$. Moreover, the means $r(\lambda)$ for $\lambda = 2, ..., 20$ and the standard deviations $s(\lambda)$ were very small as can be seen from figure 1. In the comparative test with the Multics-APL congruential generator RPGM was at least as good as the congruential generator, with a slightly better performance in the counts of $r_i(1)$ inside S, with the means and variance of the $r_i(\lambda)$ staying under control for both generators. We could easily argue that the two generators are indistinguishable under the examination of autocorrelations.

We would like to mention that A. Surkan and J. Klopping [19] have undertaken a series of comparative tests between RPGM and other well tuned congruential generators which show RPGM to behave at least as well as the competing generators. Finally, we would like to suggest that « tuning up » RPGM might produce a surprisingly good pseudo-random number generator. An investigation into which pairs of logarithmic bases yield optimal results should be an interesting future project.

SUNTO. — Descriviamo un nuovo generatore di numeri a caso (RPGM), basato sul sistema crittografico PGM inventato da Magliveras nel 1976 e successivamente studiato da Magliveras e Surkan [10]. PGM si fonda su un certo metodo di rappresentazione in un computer di un gruppo di permutazioni. Questo metodo dà luogo ad algoritmi di incrittazione e decrittazione basati su una struttura di dati efficienti, chiamata segnatura logaritmica del gruppo. L'efficacia di RPGM è studiata ricorrendo ad un'ampia analisi dei dati generati di 100.000 numeri, ottenuti usando il gruppo di Mathieu M_{24} nella sua rappresentazione 5-transitiva di grado 24.

IV. BIBLIOGRAPHY

[1] BERLEKAMP E. R., « Algebraic coding theory », McGraw-Hill, New York, 1968.

[2] BRIGHT H. S. and ENISON R. L., « Quasi-Random Number sequences from a Long-Period TLP Generator with Remarks on Application to Cryptography », *ACM Computing Surveys*, Vol. 11, no. 4, December 1979, pp. 358-370.

[3] BUTLER G., « The Schreier Algorithm for Matrix Groups », Symposium on Symbolic and Algebraic Computation, *SYSMAC '76*, 1976, p. 167.

[4] CANNON JOHN J., « On Determining the Order of a Group », *Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation*, Yorktown Heights, New York, 1976. Also: *SIGSAM Bull.*, Vol. 10, No. 3, 1976, p. 5.

[5] FELSCH, V., « Programs for Permutation Groups », Todd-Coxeter, *Defining Reations Survey*, Permutations (Actes Colloq., University Rene-Descartes, Paris, 1972), Gauthier-Villars, Paris, 1974, pp. 241-250.

[6] FRIEDMAN W. F., « Cryptology », *Encyclopedia Britannica*, Vol. 6, 1967, pp. 844-851.

[7] GOLOMB S. W., « Shift Register Sequences », Holden-Day, San Francisco, California, 1967.

[8] HALL M., « The Theory of Groups », MacMillan, 1959.

[9] KNUTH D. E., « The Art of Computer Programming », Vol. 2, *Seminumerical Algorithms*, Second Edition, Addison-Wesely, Reading, Mass., 1981, pp. 38-75.

[10] MAGLIVERAS S. S. and SURKAN A. J., « A Cryptosystem from Logarithmic Signatures of Finite Groups », to appear in the Proceedings of the 29th Midwest Symposium on Circuits and Systems, Elsevier Publ. Co., 1986.

[11] MORRIS R., SLOANE N. J. A. and WYNER A. D., « Assessment of the National Bureau of Standards Proposed Federal Data Encryption Standard », *Cryptologia*, Vol. 1, No. 3, July 1977, pp. 281-284.

[12] NEUBUSER J., « Some Applications of Group Theoretical Programs », *Proceedings of the Second Symposium on Symbolic and Algebraic Manipulations*, L. A., California, 1971, *ACM*, New York, 1971, p. 77.

[13] PEARSON E. S. and HARTLEY H. O., eds., « Biometrika Tables for Statisticians », Vol. 1, Cambridge University Press, 1958, p. 122.

[14] PLESS V., « Encryption Schemes for Computer Confidentiality », *IEEE Trans. Comp.*, C-26, 11, November 1977, pp. 1133-1136.

[15] RABIN M. O., « Probabilistic Algorithms », *Algorithms and Complexity*, J. F. Straub (ed.), Academic Press, New York, 1976, pp. 21-40.

[16] SHANNON C. E., « The Mathematical Theory of Communication », *Bell Syst. J.*, 27, July and October 1948, pp. 379-423 and pp. 623-656.

[17] SHANNON C. E., « Communication Theory of Secrecy Systems », *Bell Syst. J.*, 28, October 1949, pp. 656-715.

[18] SIMS C. C., « Computational Methods in the Study of Permutation Groups », « Computational Problems in Abstract Algebra », *Proc. Conf.*, Oxford, 1964, Pergamon Press, Oxford, 1970, pp. 169-183.

[19] SURKAN A. J. and KLOPPING J., « Comparative Tests for RPGM », unpublished working draft.

[20] WIELANDT H., « Finite Permutation Groups », Academic Press, 1964.