Mathematical Problems of Computer Science 33, 135–143, 2010.

Studies of the Matrix Generator Statistical Features

Sargis Hovasapyan

Yerevan Physics Institute, Yerevan, Armenia
e-mail: ske13@yerphi.am

Abstract

The statistical features for the optimized matrix generator of pseudorandom numbers are studied. The χ^2 and Kolmogorov-Smirnov tests are applied for very long sequences of the tested pseudorandom numbers.

1 Introduction

One of the most important tasks nowadays is the numerical experiments on super computers with the help of modeling of the sophisticated physics system. The Monte Carlo method is used for solving the problems where the high dimensional integration is involved.

To use the Monte Carlo method for analysis of the high energy physics experimental and theoretical problems we have to solve the problem of the quality of the pseudo-random generators, which should have a strong statistical feature, which include a larger period of sequences and a good speed of generating of pseudo-random numbers. The most popular and usable pseudorandom number generators used before have the period 10^9. This period is not enough for solving the integration problem, so we need a super long dimensional period.

The matrix generator based on solid theoretical ground [1] will be adapted to the supercomputer with the usage of the parallel arithmetic which in its turn implements the parallel programming libraries (MPI). The usage of the parallel arithmetic provides more generation speed. And it simultaneously provides generation of arbitrarily long pseudo-random vectors.

In the previous work on this pseudo-random generator [3] there were used some statistical tests and were given some analyses, but in this work a larger period of sequences is investigated and initial values of the vector and matrix are optimized.

2 The Purpose of this Work

The matrix generator initially proposed in [1] is based on solid theoretical ground, which is the Kolmogorov's dynamical K-systems, and has very strong statistical features, as well as super long period [4]. The only disadvantage, which has limited the wide use of such a generator, is relatively slow time of random vectors generation. It is required to build the parallel version of matrix generator which will be much faster than the old version, realized in scalar arithmetic.

At first it is needed to make the optimization of the matrix generator. which consists of the choice for the matrix constant and initial vector, to provide the best statistical features as well as long period of random sequences.

In present work we have checked the statistical features of the optimized matrix generator for very high statistics. To perform the check of statistical features for the sequence of generated numbers we used the χ^2 and the discrepancy D_N criteria [2].

In the future work we are going to perform also the *Spectral* criterion (one of the most powerful statistical criteria [2]). On final stage the matrix generator in parallel version should be tested on the super-cluster and its time characteristics is expected to be compatible with those of the standard random number generator (e. g. rand() - C ++ random number generator).

Let us describe shortly the algorithm of the matrix generator performance. Any n dimensional random vector is obtained by help of product of the $n \times n$ dimensional matrix by initial n dimensional vector. The recursive procedure consists of the input initial vector and the output in every step is the vector which will be the input vector for the next step, the matrix is initialized once.

Since the main calculation operation is the multiplying of every column of the matrix with the vector, to optimize the efficiency of calculation one can use the special matrix form with some zero elements. To provide a good approach an initial vector with researched (tested) values is initialized. After some tests we found that generator is stable for changes of initial matrix values. e.g changing $t = 3141592$ to $t = 1000001$ (see the matrix representation).

3 The Matrix Generator Description

The main idea, on which the matrix generator is based, is as follows: the pseudorandom series P_N is represented by the trajectory of certain unstable dynamical system, whose phase space is a Π^d – hypercube in the space of dimension d. The system must be maximum unstable, for the trajectory to fill this hypercube Π^d uniformly. As it is well known, those are the Kolmogorov's K-systems.

For generating pseudo-random numbers it was proposed to use the authomorphysmes of compact commutative groups, which are defined by the integer matrix $A = \|a_{ij}\|$. $P_N = AP_{N-1}$ *mod* 1, (1) where P is the d dimensional vector, belonging to Π^d (hypercube in the space of dimension d), $P_i = (X_1^{(i)}, X_2^{(i)} ... X_d^{(i)})$ and $detA = \pm 1$ (2). In order to preserve phase space volume Π^d and ensure the automorphism (1) to be the K-system, all its eigenvalues must satisfy the condition: $|\lambda_K| \neq 1, k = 1, ..., d$ (3). Thus, the problem of the pseudorandom sequence $\{P_N\}$ construction reduces to the construction of the matrix A which satisfies the conditions (1), (2) and (3). $P_N = \{A\{A...\{AP_0\}\}...\}$ (4). the points $P_0, P_1. ..., P_N$ form the very trajectory in the hypercube Π^d.

In order to check the statistical features of the sequence (4) we have to calculate χ^2 and the discrepancy D_N. which determine the convergence of the Monte-Carlo sums. Corresponding to the given initial matrix a vector which size is equal to the size of the matrix is generated.

The great advantage of the matrix generator is the possibility to change widely the components of the initial vector $P_0 = (X_1^{(0)}, X_2^{(0)} ... X_d^{(0)})$ and matrix A_d. Also in every step vector of random numbers are generated.

In our researches we use the constants: $P_0 = \{1/\sqrt{3}, 1/\sqrt{5}, 1/\sqrt{7}. 1/\sqrt{11}\}$. (5) for vector

initialization. $y = t + j, j = 0, 4, ...d, t = 3141592$

$$A_0 = \begin{pmatrix} t+y & t+y+1 & & 0 & & \\ t+y-1 & t+y & & 0 & & \\ & & t+y & t+y+1 & \cdots & \\ 0 & 0 & t+y-1 & t+y & & \\ & & & & (...) & \end{pmatrix},$$

or for a more generalized version of this matrix:

$$A_0 = \begin{pmatrix} t+y & t+y+1 & & 0 & & \\ t+y-1 & t+y & & 0 & & \\ & & t+y+i & t+y+i+1 & \cdots & \\ 0 & 0 & t+y+i-1 & t+i+y & & \\ & & & & (...) & \end{pmatrix}$$

for matrix initialization.

3.1 Other Types of the Random Number Generators

There are several types of the random number generators: linear congruent, Fibonacci, multiplicative etc.

$$X_{n+1} = (aX_n + c) \; mod \; m, n \geq 0, \quad (6)$$

(6) called a linear congruent sequence, where
m - the modulus $m > 0$, a - the multiplier $0 \leq a < m$,
c - the increment $0 \leq c < m$, X_0- the starting value $0 \leq X_0 < m$.

A further problem of LCGs is that the lower-order bits of the generated sequence have a far shorter period than the sequence as a whole if m is set to a power of 2. In general, the nth least significant digit in the base b representation of the output sequence, where $b^k = m$ for some integer k, repeats with at most period b^n.

4 Statistical Tests

In a random sequence one can expect that each of the ten decimal digits to occur approximately 1/10 times. Should the radix be 2 (digits of either 0 or 1), each digit should represent approximately 50% of the sequence.

Testing involves differentiating good sources from poor choices. For example, the binary stream ...1111111110000000000... would perform ideally with a simple Frequency test, but fail at advanced tests such as Runs, Longest Runs in a Block, and Cumulative Sums. A counter intuitive point with the presented binary stream is that it is a valid sequence of random numbers. Each stream is as equally likely to occur as any other in an unbiased generator.

Below there are some various tests which should be used when evaluating the effectiveness of a generator.

1. χ^2 - Classic Definition
2. Kolmogorov-Smirnov - Extends the χ^2 test to the set of Real Numbers
3. Gap - detect gaps between a number over a range of numbers in a sequence

4. Poker (Partition) - n groups of five successive integers from the stream, observing the resulting pattern. One pair is aabcd, full house is *aaabb*, etc.
5. Coupon Collector's - Examines the length of the sequence required to observe all numbers in the set 0 to $d-1$.
6. Permutation - number of orderings of grouping the sequence into partitions
7. Collision - Used when the χ^2 test exceeds a certain number of collisions
8. Birthday Spacing - Similar to the Birthday Paradox when selecting two integers in the sequence.

The spectral test determines the hyper-plane separation in congruent generators. To perform it the test applies a Fourier analysis to the full period of the generator using the actual generator equations. This type of test is referred to as a theoretical test and is generally applicable only to a single class of generators. While theoretical in nature, considerable computation is required for any particular parameterization of the generator and is practical only with computer algorithms to complete the calculations.

Another theoretical test applicable to lattice structure generators is the discrepancy test due to Niederreiter. These tests look for the maximum discrepancy from expected counts over sequences of s-dimensional sub regions of the unit hypercube.

In our testing process we used only χ^2 and Kolmogorov-Smirnov tests.

4.1 χ^2 Test

The χ^2 test is perhaps the best known of all statistical tests, and it is a basic method that is used in connection with many other tests. The χ^2 test is used to test if a sample of data comes from a population with a specific distribution.

An attractive feature of the χ^2 goodness-of-fit test is that it can be applied to any univariate distribution for which you can calculate the cumulative distribution function. The χ^2 goodness-of-fit test is applied to binned data (i.e., data put into classes). This is actually not a restriction since for non-binned data you can simply calculate a histogram or frequency table before generating the χ^2 test. However, the values of the χ^2 test statistic are dependent on how the data is binned. Another disadvantage of the χ^2 test is that it requires a sufficient sample size in order the χ^2 approximation be valid.

The χ^2 test is an alternative to the Anderson-Darling and Kolmogorov-Smirnov goodness-of-fit tests. The χ^2 goodness-of-fit test can be applied to discrete distributions such as the binomial and the Poisson. The Kolmogorov-Smirnov and Anderson-Darling tests are restricted to continuous distributions.

This test is sensitive to the choice of bins. There is no optimal choice for the bin width (since the optimal bin width depends on the distribution). Most reasonable choices should produce similar, but not identical, results.

The test statistic follows, approximately, a χ^2 distribution with (k - c) degrees of freedom where k is the number of non-empty cells and c is the number of estimated parameters for the distribution + 1.

Suppose that every observation can fall into one of k categories. We take n independent observations; this means that the outcome of one observation has absolutely no effect on the outcome of any of the others. Let p_s be the probability that each observation falls into category s, and let Y_s be the number of observations that actually do fall into category s. We form the statistic

$$V = \sum_{1 \le s \le k} \frac{(Y_s - np_s)^2}{np_s}$$

the number of *degree of freedom* is $dof = k - 1$, one less than the number of categories. then the value of the reduced χ^2 is defined as: $\chi^2 = V/dof$. This description given above is for one dimensional χ^2. For two dimensional χ^2 the formula is the same but the dof and the statisitcs V is counted in the following way:

$$dof = k^2 - 1, V = \sum_{1 \le i,j \le k} \frac{(Y_{ij} - np_{ij})^2}{np_{ij}}$$

where p_{ij} is the probability that each observation falls into category {i, j}, and Y_{ij} is the number of observations that actually do fall into category {i, j}.

One can see that it is easy to make the generalization of the mentioned above χ^2 definitions for any dimension of space.

4.2 SpecialTests of Matrix Generator in Tespect of Stability.

So far we have investigated the features of pseudo-random numbers given by matrix generator with size 4×4 in respect of possible dependence of 4 component vector combinations used to test the two dimensional χ^2 criterion. To perform this check we used all possible combinations (i.e.$C_4^2 = 6$) of 4 component vectors, the results of this test is shown on Figure 1, one can conclude that any of used combinations satisfied the statistical requirements, then we will use one of them for further studies, i.e. the combination (0, 1), which means the first and second components of 4-vector in case of two dimensional χ^2 test.

Also possible influence of initial vector, as well as the constant into the matrix generator on statistical features of generated numbers was investigated. We checked that change of matrix constant is not essential, but the change of initial vector leads to essential increasing of χ^2/dof values. Based on performed tests one can conclude that the initial vector components should be similar to finite representation of irrational numbers.

Also to compare the empirical χ^2 distribution with the theoretical one, the following very interesting data have been calculated: it was counted 100 of χ^2 tests continuously for 4×4 matrixes with $N_{total} = 10^6$ and for 4 dimensions.

4.3 The Kolmogorov – Smirnov Test

A test for goodness of fit usually involves examining a random sample from some unknown distribution in order to test the null hypothesis that the unknown distribution function is in fact a known, specified function.

We usually use the Kolmogorov-Smirnov test to check the normality assumption in analysis of variance.

A random sample $x_1, x_2, \ldots, x_n$ is drawn from some population and is compared with $F(x)$ in some way to see if it is reasonable to say that $F(x)$ is the true distribution function of the random sample.

One logical way of comparing the random sample with $F(x)$ is by means of the empirical distribution function $S(x)$. The data consist of a random sample $x_1, x_2, \ldots, x_n$ of size n associated with some unknown distribution function, denoted by $F(x)$.

Let $S(x)$ be the empirical distribution function based on the random sample $x_1, x_2, \ldots, x_n$. $F(x)$ be a completely specified hypothesized distribution function and the test statistic T be the greatest (denoted by *sup* for supermom) vertical distance between $S(x)$ and $F(x)$. In symbols we say $T = \sup_x |F(x) - S(x)|$

For a higher dimensional test we used another way for counting the Kolmogorov-Smirnov's formula:

$$d_n = \sum_{i,j=1}^{n} \max_{ij} \left| (\sum_{k=1}^{i} \sum_{p=1}^{j} Y_{kp})/n_{total} - (\mathrm{i}+1)(\mathrm{j}+1)\mathrm{h}^2 \right|$$

where Y_{ij} is the number of observations that actually do fall into category $\{i, j\}$, n_{total} is the count of all observations, n is the matrix size and h is the bin $(1/n)$. And for more dimensional test in formula we should change only the part of sum and multiplication.

The final formula used for this test is:

$$KS = d_n \sqrt{n_{total}}$$

5 Results

5.1 χ^2 Test

For the test results we used two matrixes with 4x4 and 8x8 sizes, and tested it from one to four dimensional tests. Also we gave the total number of observations from 1000 to 10^9. Below some of the results for both matrixes are listed.

Two dimensional χ^2test with the 4x4 matrix: $N_{total} = 10^9, \chi^2 = 1.07174$

Two dimensional χ^2 test with 8x8 matrix $N_{total} = 10^8, \chi^2 = 0.787535$

The results we obtained are compared with the results given by rand() (C++ standard random number generator). In cases when the size of used random sequences is not so big the rand() gave the same results (not equal values, but the equal meaning), but for large used statistics our generator is more stable. The test results are expected to be in range near to unity. In Figure 2 it is showen that more than 90% of the results are closely to the expected unity.

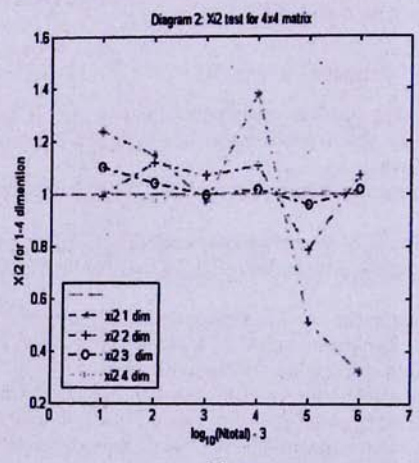

Figure 1.

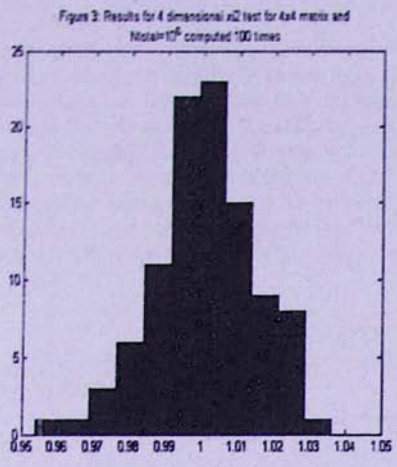

Figure 2.

5.2 χ^2 test for Rand() C++ Random Number Generator

Diagram 3 shows the value of the χ^2 for rand() C++ standard random number generator for 32 bit architecture. As we can see the value of χ^2 starts to grow very fast from $n_{total} = 10^9$. So it shows that the period of the rand() is no more than $n_{total} = 10^9$. For 64 bit architecture this precision us twice more. As it is shown in Figure 5 even in $n_{total} = 10^{10}$ the value of χ^2 is not more than 1.

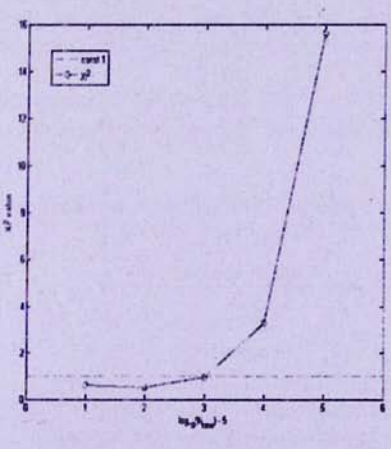

Figure 3.

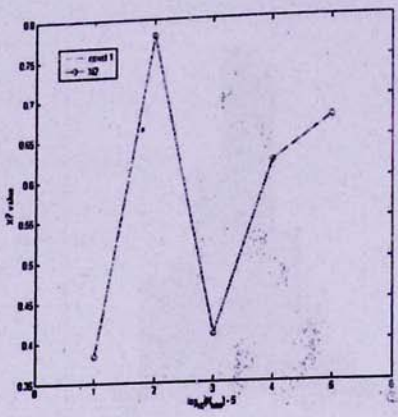

Figure 4.

5.3 Kolmogorov – Smirnov Test

To test the Kolmogorov-Smirnov criterion we again used two matrixes with 4x4 and 8x8 sizes, and tested it from one to four dimensional tests.

For matrix with size 4x4, $N = 10^9$, $ks\_1\,\mathrm{dim} = 0.333873$

For matrix with size 8x8. $N = 10^8$. $ks\_4\,\mathrm{dim} = 1.1615$

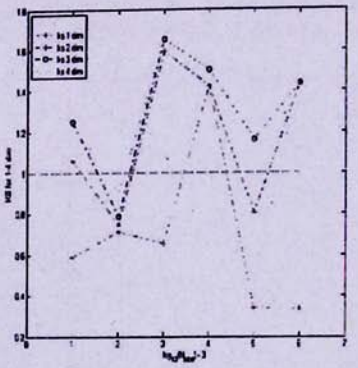

Figure 5.

6 Conclusion

The matrix generator was optimized by help of varying the matrix constant and also the initial vector. The results of performed statistical test check the features of the matrix

generator with very long sequences of pseudorandom numbers (vectors), allow to state that matrix generator has a very strong statistical abilities in respect to the uniformity and randomness of the generated numbers. At the same time the matrix generator has a very long period, which is essential to study the super high dimensional physics problems.

The authors are going to investigate also the statistical features of the optimized matrix generator using the most powerful test, which is the spectral test. On the final stage the matrix generator will be modified in parallel version to be installed on super-cluster. In this case the expected essential reducing of the generation time, will allow to use the matrix generator for the complicated multi-dimensional physics problems solving.

References

[1] N. Z. Akopov, G. K. Savvidy and N.G Ter-Arutyunian, Matrix generator of pseudorandom number, p. 573-579, 1991

[2] D.E Knuth, The Art of the computer programming, Vol. 2 Semi numerical Algorithms, Addison-Wesley, 1981

[3] N. Z. Akopov, G.G. Athanasiu, E.G. Floratos, G.K. Savvidy, Period of K system generator of pseudorandom numbers, Create University preprint CRETE.TH/12/95

[4] B. Jansson, Random Number Generators, Stockholm, 1966.

[5] G.A. Galperin, N.I. Chernow, Billiardi I Khaos, 1967.

[6] F. Gutbrod, New trends in pseudo-random number generation, DESY T-98-01, 1998.

[7] F. James, Monte Carlo theory and practice, CERN Geneva, 1980.

[8] J.H Ahrens and U. Deiter , "Extension of Forsythe's mehod for random sampling from the Normal distriution", *Math.Comp27*, pp 927-937, 1973.

[9] U. Dieter, " Pseudo-random numbers. The exact disribution of pairs", *Math Comp, 25*, pp. 855-883, 1971.

[10] D. Carey and D. Drijard, "Monte Carlo phase space with limited transverse momentum", *J.Comp. Phys*, vol. 28,pp. 327-356, 1978.

[11] P.L'Ecuyer. Efficient and portable random number generators. Com. ACM31: 742, 1988.

[12] I.M. Cobol, Chislennie metodi Monte-Carlo, Nauka, 1973.

[13] F.James, A Review or practical random number generators, October 1988.

Մատրիցային գեներատորի ստատիստիկ հատկանիշների ուսումնասիրություններ

Ս. Հովսեփյան

Ամփոփում

Ուսումնասիրված են պսեվդոդպատահական թվերի օպտիմիզացված մատրիցային գեներատորի ստատիստիկ հատկանիշները: Կիրառվում են χ^2 և Կոլմոգորով-Սմիրնով տեստերը շատ երկար հաջորդականություններ ունեցող պսեուդո պատահական թվերի համար: