Random Number Generators: Metrics and Tests for Uniformity and Randomness

E. A. Yfantis and J. B. Pedersen
Image Processing, Computer Vision and Machine Intelligence Lab
School of Computer Science
College of Engineering
University of Nevada, Las Vegas
4505 Maryland Parkway
Las Vegas, NV, 89154-4019
`yfantis@cs.unlv.edu, matt@cs.unlv.edu`

Abstract

Random number generators are a small part of any computer simulation project. Yet they are the heart and the engine that drives the project. Often times software houses fail to understand the complexity involved in building a random number generator that will satisfy the project requirements and will be able to produce realistic results. Building a random number generator with a desirable periodicity, that is uniform, that produces **all** the random permutations with equal probability, and at random, is not an easy task. In this paper we provide tests and metrics for testing random number generators for uniformity and randomness. These tests are in addition to the already existing tests for uniformity and randomness, which we modify by running each test a large number of times on sub-sequences of random numbers, each of length n. The test result obtained each time is used to determine the probability distribution function. This eliminates the random number generator misclassification error. We also provide new tests for uniformity and randomness, the new tests for uniformity test the skewness of each one of the subgroups as well as the kurtosis. The tests for randomness, which include the Fourier spectrum, the phase spectrum, the discrete cosine transform spectrum, and the orthogonal wavelet domain, test for patterns not detected in the row data space. Finally we provide visual and acoustic tests.

1 Introduction

Sequences generated in a deterministic way via an algorithm which is programmed in the computer, are usually called pseudo random numbers. Of course by the nature of generation of pseudo random numbers, it is difficult to argue that the produced numbers are random. The question is not if the algorithm used is deterministic or not, but if the sequence produced has random behavior or not. Knowing the current number can one predict the next number or not, or even if one can predict if the next number has some properties such as greater than the current number with high probability, etc. The first attempt to generating random numbers was the John Von Neumann "middle square method" in the early 1950s which was proven to be poor source of random numbers with short periodicity [5]. Metropolis [11] working with 38-bits managed to obtain a sequence of 750,000 before it degenerated. The Metropolis 38-bit algorithm passed satisfactorily a number of statistical tests for uniformity and randomness, but for today's many applications, it does not have large enough periodicity. One of the most popular random number generators in use today are special cases of the linear congruential method introduced by D.H. Lehmer [6], the algorithm is of the form: $X_{n+1} = (aX_n + c) \mod m$, where $n \geq 0$, and m is the maximum number stored by the computer. The coefficients a and c, and the seed X_0 of the random number generator ($0 \leq a, c, X_0 < m$) all have to be chosen in a way the period is maximum that means the period is m and the generator produces uniform random numbers. The number m is chosen to be $m = 2^b$, where b is the maximum number of bits we want to use to generate the appropriate random numbers. Usually b is equal to the integer register size of the computer. If we choose $a = 1$, $c = 1$, $X_0 = 0$, and $m = 2^b$, then the numbers the generator $X_{n+1} = (X_n+1) \mod m$ with $X_0 = 0$ will generate, have a maximum periodicity of m, but the sequence generated is $0, 1, 2, 3, 4, \ldots, m-1$, which is not random at all.

The selection of a, and c, that guarantees to generate a sequence of period m, if $m = 2^b$, where b is the maximum number of bits we want to use to generate the random sequence, is as follows:

> c is relatively prime to 2, and $a = 4k + 1$, for $k = 1, 2, \ldots$. If $c = 0$ the maximum period is obtained if a is a primitive element modulo m and X_0 is relatively prime to m.

Maximum periodicity of course does not guarantee randomness. Random number generators have been investigated over the years by many researchers; examples include [1, 2, 3, 4, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18]. They have many applications. They are the heart of random games (electronic slot machines, electronic bingo, electronic poker, etc.), for simulating industrial processes, queuing theory related applications, flight simulation, and Monte Carlo methods for solving mathematical problems.

According to Dr. I. J. Matrix [5], although mathematicians consider the decimal expansion of π as a random series. To modern numerologists however π is full of interesting patterns. Dr. Matrix has pointed out that the first repeated two-digit number in π's expansion is 26:

$$\pi = 3.14159\mathbf{26}5358979323846\mathbf{26}4338327950\ldots.$$

According to Dr. Matrix, the digits of π, if correctly interpreted, convey the entire history of the human race. Statistics and signal processing provide the tools to create tests for uniformity, and randomness. If a series behaves randomly with respect to a battery of k-tests administered to the sequence, an additional test administered to the sequence might result in rejecting the randomness of the sequence. It is very difficult to assess that a sequence of numbers is random but for each test for randomness it passes the confidence on the randomness of the generator increases.

In this paper we revisit some of the popular tests for uniformity, and randomness, but instead of administering them on one sequence using one estimate of the test-statistic, the way it is recommended by the literature, we administer them to several sub-sequences and use the test-statistic results obtained each time to create a probability distribution function for the test-statistic. The probability distribution obtained this way has to adhere to the theoretical distribution of the test-statistic. Due to the law of large numbers and the small variance of the estimated parameters, the estimated parameters of the probability distribution converge to the true parameters. The proposed changes are computationally intense, and also require new software development. They do however reduce the probability of misclassifying the generator. In addition to the revised traditional tests we present a number of new tests for uniformity and randomness. Some of these tests pertain to the discrete uniform distribution which is very important for electronic casino games.

This paper is organized as follows: In section 2 we present the revised χ^2 test and Kolmogorov-Smirnov tests for uniformity. We also revisit the tests for randomness presented in the literature but we revise them. The revised versions increase the potency of the randomness tests. In section 3 we present the new tests for randomness and uniformity. We also organize the random numbers into a high definition (R, G, B) still images. Thus we use three consecutive random numbers quantized between 0 and 255 to create pixels of a color image. In this color image we can apply the theorems of section 3 to tests for randomness, and we also perform a visual inspection for patterns. We also quantize the random numbers between -2^{15} and $2^{15}-1$, and we organize the acoustic signals into a wave file to detect any non-random acoustic patterns.

2 A Revision of the Existing Tests

The way the tests described below are administered is as follows: a sample is obtained from the random sequence and the test is applied on the sample; if the sample passes the test, the random number generator passes the test, otherwise the random number generator does not pass the test.

In our revision, every test described here is applied to may sub-sequences from the random number generator (typically over 300), and the length of every sub-sequence is over 1,000 numbers. From all of these computations we form a random distribution which should be the same as the one described by the theory related to that test.

In order for us to show that the two distributions, namely the theoretical and the distribution obtained by the multiple samples are the same we test, the hypothesis is that the parameters of the estimated distribution are not significantly different than those of the theoretical distributions. Variance tests result in a χ^2 with appropriate degrees of freedom, and tests for the means result in a t-distribution with appropriate degrees of freedom. If the parameters of two distributions (belonging to the same family) are the same, then the distributions are the same. Here due to the law of large numbers, the experimental parameters should converge to the theoretical parameters, if the random generator is uniform, and/or random, depending on the test administered. Of course another way is to use Kolmogorov-Smirnov, or even a χ^2 test to show that the expected number of data in each segment of the probability function is equal to the observed number of data. This revision increases the reliability of testing the random numbers. The suggested modifications require writing a great deal of software. Furthermore using them requires time consuming computations that could outweigh the benefits for some applications.

The Serial Test
The Serial test examines the frequency of the possible pairs (q, r), $0 \leq q, r \leq d$, where $0, 1, \ldots, d$, are all the possible outcomes of the random game and the random number generator simulating the random game. There are $(d+1)^2$ possible pairs, and each possible pair appears with probability $1/(d+1)^2$, thus in a sequence of N random numbers, the expected number of times a particular pair occurs is $N/(d+1)^2$, the sum of the ratios of the observed minus the expected squared over the expected has a χ^2 distribution with

$d^2 + 2d$ degrees of freedom. Instead of performing this test on a sub-sequence of N random numbers, we select k sub-sequences, each of size N and perform the test on each one of the sub-sequences. Each time we perform the test, we obtain a χ^2 value. The histogram of the k χ^2 values should be an estimate of a χ^2 distribution with $d^2 + 2d$ degrees of freedom.

The Gap Test
If $0 < n < k < d$, we consider the lengths of consecutive sub-sequences $r_i, r_{i+1}, r_{i+2}, \ldots, r_{i_{m-1}}, r_{i+m}$ in which r_{i+m} is between n and k and the other ones are not. In other words, we consider the inter-arrival time of numbers which are between n and k.

The Poker Test
The poker test considers n groups of five successive integers and observes which of the following patterns is matched by each quintuple:

- All different.
- One pair and three different.
- Two pairs and one different.
- Three of a kind and two different.
- Full house: three the same and two the same.
- Four of a kind.
- Five of a kind.

The Coupon Collectors Test
The length required to get a complete set of integers from 0 to $d - 1$.

The Permutation Test
Divide the input sequence into n groups of k elements each (r_{ik}, r_{ik+1}, $\ldots, r_{(i+1)k-1}$), where $0 \leq i \leq n$. If n is sufficiently large with respect to k, the element in each group could have $k!$ possible relative orderings. The number of times each ordering appears is counted and a χ^2test is applied with probability $1/k!$ for each ordering to occur. This test we implement on large sub-sequences, and the algorithm we use is different than the one in the literature. We demonstrate our algorithm with an example of 4 elements ($k = 4$), and $n = 2,400$. So there are 24 permutations for each vector of four elements, and we expect each permutation to be repeated 100 times. The χ^2 of course is the sum of observed minus expected the quantity squared over expected, and it has 23 degrees of freedom. For example, consider the following four tuples generated $(30, 24, 516, 2)$, $(44, 532, 29, 1, 63, 892)$, $(54, 89, 031, 102, 532, 108, 378)$, $(395, 289, 1, 029, 107)$, $(489, 391, 33, 763)$, $\ldots$.

According to our algorithm, in each of the above vector of 4 numbers, we find the position of the largest and we record that, then we consider the remaining 3 and we find the position of the larger among the remaining three, finally we find the position of the larger among the 2. These permutation vectors associated to the above numbers are: $(2, 0, 0)$, because the largest number between the 4 numbers $(30, 24, 516, 2)$, is at position 2, next we ignore 516, and the largest number among the remaining 3, $(30, 24, 2)$ is at position 0, now we ignore 30, and the largest number among the reaming 2, $(24, 2)$, is at position 0. Thus the vector is $(2, 0, 0)$. The permutation vectors associated with the above vectors therefore are $(2, 0, 0), (3, 0, 0), (3, 2, 1), (2, 0, 0), (3, 0, 0), \ldots$. The implementation of the algorithm is described by the following pseudo-code:

```
**int** $i$, $j$, $k$, $n$, $m$, $max$, $a[4][3][2]$;
**for** ($i = 0$; $i < 4$; $i$++)
    **for** ($j = 0$; $j < 3$; $j$++)
        **for** ($k = 0$; $k < 2$; $k$++)
            $a[i][j][k] = 0$;
$n = 2400$; // the number of random vectors of size four to be generated;

**for**($m$=0; $m < n$; $m$++) {
    Generate the first random number;
    Generate the second random number;
    Generate the third random number;
    Generate the fourth random number;
    Find the $max$ of the four random numbers;
    Set $i$=$max$;
    Find the $max$ of the remaining three;
    Set $j$=$max$ of the remaining 3;
    Find the $max$ of the remaining two;
    Set $k$=$max$ of the remaining 2;
    $a[i][j][k]$++; //This array keeps count of each permutation
}
```

At the end the array $a[0][0][0]$, to $a[3][2][1]$, will hold all the possible permutations generated. Theoretically for this example every element $a[i][j][k]$ of the array should be equal to 100. In practice we have fluctuations from the theoretical number the χ^2 distribution decides if these fluctuations are random, or not.

Run-Test
A sequence may be tested for "runs up" and "runs down". In this test we

examine the lengths of monotone sub-sequences of the original sequence.

Maximum of k test
We consider sub-sequences of length k, and we find the maximum, that is, $S_i = max(r_{ik}, r_{ik+1}, \ldots, r_{ik+(k-1)})$ then we apply the Kolmogorov-Smirnov test to the sequence $S_0, S_1, \ldots, S_{n-1}$.

3 New Tests for Uniformity and Randomness

The following five theorems pertain to testing the uniformity and randomness of random generators. Specifically theorem 1 pertains to the uniformity. Since the number N of random numbers generated is very large, due to the law of large numbers, the mean, variance, skewness, and kurtosis, should approach very close to the ones given by the theorem. The z-test is a good distance formula, specifying the maximum allowable distance of the sample statistics from the theoretical values. Theorems 2, 3, 4, and 5 all pertain to randomness. Thus if the random generator does not produce truly random data, then the produced sequence is auto-correlated and therefore multivariate analysis, the theory of random processes, and signal processing, are appropriate theories to detect if there are any dependencies between the numbers of the sequence generated by the generator.

Theorem 1 *Given M random numbers from the discrete uniform distribution function with possible values $1, 2, 3, 4, \ldots, M$, then if X is a random variable with probability*

$$P(X = i) = \frac{1}{M}, \quad i = 1, 2, \ldots, M$$

Then we get the following results:

$$\begin{aligned}
E(X) &= \frac{M+1}{2} \\
E(X^2) &= \frac{(M+1)(2M+1)}{6} \\
E(X^3) &= \frac{M(M+1)^2}{4} \\
E(X^4) &= \frac{(M+1)(6M^3+9M^2+M-1)}{30} \\
\sigma_X^2 &= \frac{M^2-1}{12} \\
E(X-\mu)^3 &= 0 \\
E(X-\mu)^4 &= \frac{(M-1)(M+1)(48M^3+75M^2+8M-15)}{240}
\end{aligned}$$

also the skewness

$$\gamma_1 \quad = \quad \frac{E(X-\mu)^3}{\sigma_X^3} = 0$$

and the kurtosis is:

$$\gamma_2 \quad = \quad \frac{E(X-\mu)^4}{\sigma_X^4} = \frac{(M-1)(M+1)(48M^3+75M^2+8M-15)}{240\sigma_X^4}$$

Proof:

$$E(X) = \mu = \frac{1}{M}\sum_{n=1}^{M} n = \frac{M+1}{2} \tag{1}$$

also we get by the binomial theorem

$$\begin{aligned}
1^3 &= (0+1)^3 = 1 \\
2^3 &= (1+1)^3 = 1^3 + 3*1^2 + 3*1 + 1 \\
3^3 &= (2+1)^3 = 2^3 + 3*2^2 + 3*2 + 1 \\
4^3 &= (3+1)^3 = 3^3 + 3*3^2 + 3*3 + 1 \\
\vdots & \quad \vdots \\
(M+1)^3 &= M^3 + 3*M^2 + 3*M + 1
\end{aligned}$$

So to determine the sum

$$\sum_{i=1}^{M+1} i^3$$

we sum each side of the equal sign, after canceling out similar terms, arrive at

$$(M+1)^3 = 1 + 3(1^2+2^2+3^2+\ldots+M^2) + 3\frac{M(M+1)}{2} + M$$

which implies

$$\sum_{i=1}^{M} i^2 = \frac{M(M+1)(2M+1)}{6}$$

and hence

$$E(X^2) = \frac{(M+1)(2M+1)}{6} \tag{2}$$

and

$$\sigma_X^2 = \frac{(M+1)(M-1)}{12} \tag{3}$$

The third moment $E(X^3)$ can be found in a similar manner using the binomial theorem and the technique applied above:

$$\begin{aligned}(M+1)^4 &= M^4 + 4*M^3 + 6*M^2 + 4*M + 1\\ &= 1 + 4(1^3 + 2^3 + 3^3 + \ldots + M^3)\\ &\quad +M(M+1)(2M+1) + 2M(M+1) + M\end{aligned}$$

From the above we obtain

$$\sum_{i=1}^{M} i^3 = \frac{M^2(M+1)^2}{4}.$$

Thus

$$E(X^3) = \frac{M(M+1)^2}{4}. \tag{4}$$

Furthermore

$$\begin{aligned}E(X-\mu)^3 &= E(X^3) - 3E(X^2)\mu + 3E(X)\mu^2 - \mu^3\\ &= \frac{M(M+1)^2 - (M+1)^2(2M+1) - (M+1)^3}{4}\\ &= 0\end{aligned}$$

and hence the skewness is

$$\gamma_1 = \frac{E(x-\mu)^3}{\sigma_X^3} = 0 \tag{5}$$

which implies that the probability function is symmetric about the mean.

The fourth moment $E(X^4)$ can be found as follows:

$$\begin{aligned}(M+1)^5 &= M^5 + 5*M^4 + 10*M^3 + 10*M^2 + 5*M + 1\\ &= 1 + 5(1^4 + 2^4 + 3^4 + \ldots + M^4) + \frac{5M^2(M+1)^2}{2} +\\ &\quad \frac{5M(M+1)(2M+1)}{3} + \frac{5M(M+1)}{2} + M\end{aligned}$$

From the above we have

$$\sum_{i=1}^{M} i^4 = \frac{M(M+1)(2M+1)(3M^2+3M-1)}{30},$$

which implies that the fourth moment is

$$E(X^4) = \frac{(M+1)(2M+1)(3M^2+3M-1)}{30}. \tag{6}$$

In addition

$$\begin{aligned} E(X-\mu)^4 &= E(X^4) - 4E(X^3)\mu + 6E(X^2)\mu^2 - 4E(X)\mu^3 + \mu^4 \\ &= \frac{(M+1)(2M+1)(3M^2+3M-1)}{30} \\ &\quad -M(M+1)^2\mu + (M+1)(2M+1)\mu^2 - 3\mu^4 \\ &= \frac{(M-1)(M+1)(48M^3+75M^2+8M-15)}{240} \qquad (7) \end{aligned}$$

Thus the kurtosis becomes

$$\gamma_2 = \frac{(M-1)(M+1)(48M^3+75M^2+8M-15)}{240\sigma_X^4} \qquad (8)$$

□

Tests for skewness and kurtosis are very important. The way we administer these tests is by estimating the skewness based on a large number of random numbers. Then, due to the law of large numbers, the difference of the theoretical skewness and the sample skewness is a normal distribution with mean zero. The z-test is used to test the skewness. In a similar way we test the kurtosis. Random number generators with skewness and/or kurtosis significantly different than the theoretical values are not good random number generators.

Theorem 2 *If $\underline{X} = [X_1, X_2, X_3, \ldots, X_k]$ is a random vector with each X_i $(i = 1, 2, 3, \ldots, k)$ being a discrete uniform distribution with random values between 1 and M as well as independent, then*

$$E[\underline{X}] = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_k \end{bmatrix} = \begin{bmatrix} \frac{M+1}{2} \\ \frac{M+1}{2} \\ \vdots \\ \frac{M+1}{2} \end{bmatrix},$$

$$\Sigma = \begin{bmatrix} \frac{M^2-1}{12} & 0 & 0 & \ldots & 0 \\ 0 & \frac{M^2-1}{12} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \frac{M^2-1}{12} \end{bmatrix}.$$

Furthermore, The principal components of the vector $\underline{X}$ are the X_i $(i = 1, 2, 3, \ldots, k)$ and the eigenvalues are all equal to $(M^2-1)/12$.

Proof:

Since each one of the random variables X_i $(i = 1, 2, 3, \ldots, k)$ has the uniform distribution, we get:

$$\mu_i = E[X_i] = \frac{M+1}{2} \tag{9}$$

and the variance

$$\sigma^2_{X_i} = \frac{(M-1)(M+1)}{12} = \frac{M^2-1}{12},\ i = 1, 2, 3, \ldots, k. \tag{10}$$

Also $E[(X_i - \mu_i)(X_j - \mu_j)] = 0$ because X_i and X_j are independent (for $i \neq j$). Hence, from the above we infer that the mean vector, and the variance covariance matrices are the ones stated by the theorem. $\square$

This is a very important test which tests for auto-correlation. We form vectors of size n using n consecutive random numbers. Thus the first n-consecutive random numbers from 1 to n form one vector. The second n-consecutive random numbers from $n+1$, to $2n$, form a second vector of n-numbers, and so on. For the large sample of random vectors we compute the variance covariance matrix . The computed matrix will not be exactly as the theoretical matrix, but should not be significantly different than the theoretical matrix either. To test the significance we test each diagonal of the computed matrix against the diagonal of the theoretical matrix by using the χ^2 distribution, for each pair of experimental-theoretical diagonal entry. Subsequently, we use the Wishart distribution to test the experimental variance covariance matrix if it is significantly different than the theoretical variance covariance matrix. For an acceptable generator the two matrices should not be significantly different. This test is also an auto-correlation test up to lag n, and as such it is also a randomness tests, which many generators fail.

Theorem 3 *Let $x_0, x_1, x_2, \ldots, x_{N-1}$ be a sequence of random number from the discrete uniform distribution $U[1, M]$. The discrete cosine transform of the sequence is*

$$X_m = c_m \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi m(2n+1)}{2N}\right), m = 0, 1, 2, \ldots, N-1,$$

where

$$\begin{aligned} c_0 &= \frac{1}{\sqrt{2}} \\ c_m &= 1,\ m = 1, 2, 3, \ldots, N-1. \end{aligned}$$

Then

$$\begin{aligned} E(X_0) &= \frac{\sqrt{N}(M+1)}{2} \\ \sigma^2_{X_0} &= \frac{(M+1)(M-1)}{12} \\ E(X_m) &= 0,\ m = 1,2,3,\ldots,N-1. \end{aligned}$$

Proof:

Since

$$X_0 = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n$$

we get

$$E[X_0] = \frac{\sqrt{N}(M+1)}{2} \tag{11}$$

and

$$\begin{aligned} E[X_0^2] &= \frac{1}{N} E\left(\sum_{n=0}^{N-1} x_n\right)^2 \\ &= \frac{1}{N}\sum_{n=0}^{N-1} E(x_n^2) \\ &\quad + \frac{2}{N}\left((N-1)\frac{(M+1)^2}{4} + (N-2)\frac{(M+1)^2}{4} + \cdots + \frac{(M+1)^2}{4}\right) \\ &= \frac{1}{N}\sum_{n=0}^{N-1} E(x_n^2) + \frac{2}{N}\sum_{i=1}^{N-1} i\frac{(M+1)^2}{4} \\ &= \frac{(M+1)(2M+1)}{6} + \frac{(N-1)(M+1)^2}{4} \end{aligned}$$

which implies

$$\begin{aligned} \sigma^2_{X_0} &= \frac{(M+1)(2M+1)}{6} + \frac{(N-1)(M+1)^2}{4} - \frac{N(M+1)^2}{4} \\ &= \frac{(M+1)(M-1)}{12} \end{aligned} \tag{12}$$

$\square$

This test is best to be administered on 64 consecutive random numbers, organized in an 8×8 square and subtracting the mean value from each of the numbers. Then all components would have mean zero, and the expected number of positive should be equal to the expected number of negative values, and each one should be about 32.

Theorem 4 *Let $x_0, x_1, x_2, \ldots, x_{N-1}$ be a sequence of random numbers from the discrete uniform distribution $U[1, M]$. The finite Fourier transform of the sequence is*

$$X_m = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi mn}{N}} = U_m + iV_m, \; m = 0, 1, 2, \ldots, N-1.$$

The spectral density is

$$S_m = U_m^2 + V_m^2$$

and the phase spectrum

$$\varphi_m = \arctan\left(\frac{V_m}{U_m}\right)$$

has a uniform circular distribution between 0 and 2π.
Furthermore

$$\begin{aligned} E(X_0) &= E(U_0) = \frac{\sqrt{N}(M+1)}{2} \\ E(V_0) &= 0 \\ \sigma_{X_0}^2 &= \frac{M^2-1}{12} \\ E(X_m) &= 0, \; m = 1, 2, 3, \ldots, N-1 \end{aligned}$$

Proof:

$$X_0 = \sqrt{N}\bar{x} = \sqrt{N}\frac{\sum_{i=0}^{N-1} x_i}{N}$$

hence

$$E(X_0) = \frac{\sqrt{N}(M+1)}{2} \tag{13}$$

and

$$\sigma_{X_0}^2 = E(X_0^2) - E^2(X_0) = \frac{M^2-1}{12}. \tag{14}$$

So now

$$E(X_m) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} E(x_n) e^{-i\frac{2\pi mn}{N}} = \frac{1}{\sqrt{N}} \frac{M+1}{2} \sum_{n=0}^{N-1} e^{-i\frac{\pi mn}{N}} = 0. \quad (15)$$

□

The power spectrum of the random sequence should not be significantly different than the square of the mean of the component at the zero frequency; for all other frequencies the power spectrum should be uniformly distributed up to the Nyquest frequency. The phase spectrum should have a uniform distribution between $[0, 2\pi]$. The spectrum test and the phase spectrum test are both tests for randomness. Random number generators with spectral properties different than the theoretical spectral properties are auto-correlated and therefore not random.

Theorem 5 *Let $x_0, x_1, x_2, \ldots, x_{N-1}$ be a sequence of random numbers from the discrete uniform distribution $U[1, M]$, and let $\underline{X} = [x_0, x_1, x_2, \ldots, x_{N-1}]$ be the corresponding N-dimensional vector. Consider the seven tap wavelet quadrature mirror filter given by this orthogonal matrix:*

$$W = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & 0 & a_3 & a_2 & a_1 \\ -a_1 & a_0 & -a_1 & a_2 & -a_3 & 0 & -a_3 & a_2 \\ a_2 & a_1 & a_0 & a_1 & a_2 & a_3 & 0 & a_3 \\ -a_3 & a_2 & -a_1 & a_0 & -a_1 & a_2 & -a_3 & 0 \\ 0 & a_3 & a_2 & a_1 & a_0 & a_1 & a_2 & a_3 \\ -a_3 & 0 & -a_3 & a_2 & -a_1 & a_0 & -a_1 & a_2 \\ a_2 & a_3 & 0 & a_3 & a_2 & a_1 & a_0 & a_1 \\ -a_1 & a_2 & -a_3 & 0 & -a_3 & a_2 & -a_1 & a_0 \end{bmatrix}$$

Now let

$$\begin{aligned} y_0 &= a_0x_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_3x_5 + a_2x_6 + a_1x_7 \\ z_0 &= -a_1x_0 + a_0x_1 - a_1x_2 + a_2x_3 - a_3x_4 - a_3x_6 + a_2x_7. \end{aligned}$$

y_0 ($y_0 = [W \cdot \underline{X}]_0$) is the result of applying the low pass filter in the process and z_0 ($z_0 = [W \cdot \underline{X}])_1$) is the result of applying the high pass filter in the process. Then the expected value and the variance of y_0 and z_0 are

$$\begin{aligned} E(y_0) &= \sqrt{2}\mu_{x_0} \\ \sigma^2_{y_0} &= \sigma^2_{x_0} \\ E(z_0) &= 0 \\ \sigma^2_{z_0} &= \sigma^2_{x_0} \end{aligned}$$

Proof:

Since this is an orthogonal quadrature mirror filter, the coefficients a_0, a_1, a_2, a_3 satisfy the following equations:

$$\begin{aligned}
a_0 + 2a_1 + 2a_2 + 2a_3 &= \sqrt{2} \\
a_0 + 2a_2 &= \frac{\sqrt{2}}{2} \\
2a_1 + 2a_3 &= \frac{\sqrt{2}}{2} \\
a_0^2 + 2a_1^2 + 2a_2^2 + 2a_3^2 &= 1 \\
2a_0a_2 + 2a_1a_3 + a_1^2 + a_3^2 &= 0 \\
4a_1a_3 + 2a_2^2 &= 0
\end{aligned}$$

Solving the above system of equations, we obtain

$$\begin{aligned}
a_0 &= 0.8534972 \\
a_1 &= 0.3610506 \\
a_2 &= -0.0731952 \\
a_3 &= -0.0074972.
\end{aligned}$$

Now if we consider the low pass filter y_0 then

$$\begin{aligned}
E(y_0) &= E(a_0x_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_3x_5 + a_2x_6 + a_1x_7) \\
&= (a_0 + 2a_1 + 2a_2 + 2a_3)\mu_{x_0} \\
&= \sqrt{2}\mu_{x_0} \qquad (16) \\
E(y_0^2) &= E(a_0x_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_3x_5 + a_2x_6 + a_1x_7)^2 \\
&= (a_0^2 + 2a_1^2 + 2a_2^2 + 2a_3^2)E(x_0^2) \\
&\quad + (4a_0a_1 + 4a_0a_2 + 4a_0a_3 + 4a_1a_2 + 4a_1a_3 + 2a_1^2 + \\
&\quad + 4a_2a_3 + 2a_2^2 + 2a_1a_2 + 2a_3^2 + 4a_1a_3 + 4a_2a_3 + 2a_1a_2)\mu_{x_0}^2
\end{aligned}$$

From the above equations we obtain the following:

$$\begin{aligned}
E(y_0^2) &= E(x_0^2) + 4(a_1 + a_3)(a_0 + 2a_2)\mu_{x_0}^2 \\
&= E(x_0^2) + \mu_{x_0}^2 \\
&= \sigma_{x_0}^2 + 2\mu_{x_0}^2
\end{aligned}$$

And thus

$$\sigma_{y_0}^2 = E(y_0^2) - E(y_0)^2$$

$$\begin{aligned} &= \sigma_{x_0}^2 + 2\mu_{x_0}^2 - E(y_0)^2 \\ &= \sigma_{x_0}^2 + 2\mu_{x_0}^2 - (\sqrt{2}\mu_{x_0})^2 \\ &= \sigma_{x_0}^2 \qquad (17) \end{aligned}$$

The expected value of the high pass filter z_0 is

$$\begin{aligned} E(z_0) &= E(-a_1x_0 + a_0x_1 - a_1x_2 + a_2x_3 - a_3x_4 - a_3x_6 + a_2x_7) \\ &= (-2a_1 + a_0 + 2a_2 - 2a_3)\mu_{x_0} \\ &= 0 \qquad (18) \end{aligned}$$

To compute $\sigma_{z_0}^2$ we proceed with

$$\begin{aligned} E(z_0^2) &= E\left((-a_1x_0 + a_0x_1 - a_1x_2 + a_2x_3 - a_3x_4 - a_3x_6 + a_2x_7)^2\right) \\ &= (a_0^2 + 2a_1^2 + 2a_2^2 + 2a_3^2)E(x_0^2) \\ &\quad + 2(-a_1a_0 + a_1^2 - a_1a_2 + a_1a_3 + a_1a_3 - a_1a_0 - a_1a_2 \\ &\quad + a_0a_2 - a_0a_3 - a_0a_3 + a_0a_2 - a_1a_2 + a_1a_3 + a_1a_3 \\ &\quad + a_1a_2 - a_2a_3 - a_2a_3 + a_2^2 - a_2a_3 - a_2a_3)\mu_{x_0}^2 \\ &= E(x_0^2) - \mu_{x_0}^2 \\ &= \sigma_{x_0}^2 \end{aligned}$$

And since $E(z_0) = 0$ we get

$$\sigma_{z_0}^2 = E(z_0^2) - E(z_0)^2 = \sigma_{x_0}^2. \qquad (19)$$

□

The way this test applied is as follows: We apply the low pass filter to the random sequence and we obtain a signal half the size of the original signal which is the low pass transform signal (the low pass coefficients). Then we apply to the original sequence the high pass wavelet filter, and we obtain a signal half the size of the original signal which is the high pas filter. From the above theorem both low pass, and high pass signals should have exactly the same energy. If that does not happen the generator is not random. This is a test that the majority of the generators fail. In non random data sequences the low pass component of the wavelet transformed data has significantly higher energy than the high pass component of the wavelet transformed data. Due to Parsevals theorem, the total energy of the original data is equal to the total energy of the transformed data.

4 Visual and acoustic tests for random number generators

In addition to the tests described in sections 2 and 3, a number of visual and acoustic tests can be provided. A simple and yet very affective visual test for assessing if the random number generator is capable of producing all the possible pairs of integer numbers within a certain range $1, 2, \dots, C$, with equal probability, is to run the generator until the last number in the sequence is generated. Every consecutive pair of random numbers (r_i, r_{i+1}) in the sequence represents a coordinate in a bitmap type image with resolution $C * R$, where C is the number of columns and R is the number of rows of the image. The pixel having the coordinate (r_i, r_{i+1}) is painted black. Before the process starts all pixels are initialized to white. If the cycle of the generator is large enough relative to the resolution of the bitmap image, then the generator should generate all the possible coordinates of the image with equal probability. Thus the image should be totally black after all the random numbers are generated. Furthermore all coordinates should be generated equal amount of times. This is not always the case as we see in the bmp file with resolution 800×800 pixels generated by `rand()`, a very popular random number generator. Not all pairs are generated with equal probability, and as it shown in Fig. 1, some pairs are never generated by the end of the cycle of the generator, while some other pairs are generated multiple times.

In Fig. 1 consecutive random numbers were paired. The first random number of the pair represents the x-coordinate, and the second the y-coordinate. For each pair (x, y) a black pixel is drawn on white screen. So white pixels represent (x, y) coordinates never reached by any of the generated random pairs, while the black pixels represent (x, y) coordinates reached at least once by the end of the generated sequence. The generator used was `rand()` which is part of the library functions of many compilers. Ideally the picture should be all black, and every (x, y) coordinate should be visited equal number of times by the end of the generated sequence.

A number of additional visualization tests can be administered with triplets (three consecutive random numbers, the first two being the (x, y) coordinates and the third being the gray scale intensity), or with five-tuples, the first two being the (x, y) coordinates and the other 3 being the three color channel intensities red, green, and blue (R, G, B). In addition to that one could compute the frequency distribution of each five tuple. Acoustic tests using the random number generators can also be created by organizing the sequence into a wav file mono, stereo, surround sound, or just a voice resolution channel, and play it to detect patterns, and zero crossings. We

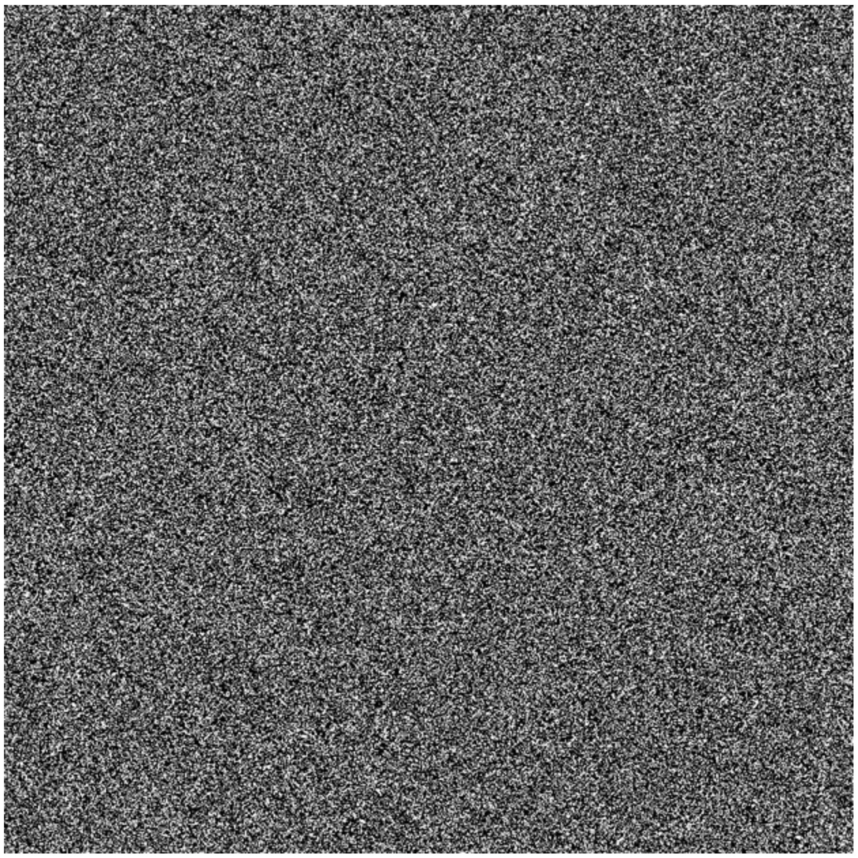

Figure 1: 800×800 bitmap result of `rand()` testing.

quantize the random numbers in the range -2^{15} and $2^{15}-1$ before we organize them into an acoustic wave file. Acoustic inspection of the wave file could reveal patterns in the random number generator.

5 Conclusions

Random number generators are very important for scientific applications as well as commercial applications such as the casinos. Here we provide a number of tests, which includes tests already known and revised in this paper, new tests for uniformity and randomness as well as visual and acoustic tests. The new tests for uniformity test the skewness of each one of the subgroups, and the kurtosis. The tests for randomness, which include the Fourier spectrum, the phase spectrum, the discrete cosine transform spectrum, and the orthogonal wavelet frequency domain, test for patterns not detected in the row data space. Finally we provide visual and acoustic tests. In the acoustic tests we quantize the random numbers between -2^{15} and $2^{15} - 1$ and then we organize them into an acoustic wave file. Acoustic inspection of the wave file could reveal patterns in the random number generator as well.

References

[1] S. L. Anderson. Random Number Generators on Vector Supercomputers and other Advanced Architectures. *SIAM Review*, 32(2):221–251, 1990.

[2] A. C. Arvillias and D. G. Maritsas. Partitioning the period of a class of m-sequences and applications to pseudorandom generation. *JACM*, 25:675–686., 1978.

[3] P. H. Bardell. Analysis of Cellular Automata Using a Pseudorandom Pattern Generators. *IEEE International Test Conference*, pages 762–768, 1990.

[4] M. Fushimi. Statistical Tests of Pseudorandom Bunbers Generated by a Linear Recurrence Modulo Two. *Asian Pacific Operations Research88*, pages 185–199, 1990.

[5] D. E. Knuth. *The Art of Computer Programming*, volume 2: Seminumerical Algorithms 2nd ed. Addison Wesley, Reading, Mass., 1981.

[6] D. H. Lehmer. Random Number Generators. In *Proc. 2nd Symp. On Large-Scale Digital Calculating Machinery*, pages 141–146. Harvard University Press, 1951.

[7] G. Marsaglia. Random Numbers Fall Mainly in the Plains. *Proc. National Academy of Science U.S.A.*, 61:25–28, 1968.

[8] G. Marsaglia. A Current View of Random Number Generators. In *Keynote Address: Proceddings, Computer Science and Statistics: 16th Symposium on the Interface*. Elsevier, 1985.

[9] G. Marsaglia and L. H. Tsay. Matrices and the Structure of Random Number Sequences. *Linear Algebra and its Applications*, 67:147–156, 1985.

[10] G. Marsaglia and A. Zaman. A new Class of Random Number Generators. *Annals of Applied Probability*, 1(3):323–346, 1991.

[11] N. Metropolis, A. Rosenbluth, M. Teller, and E.J. Teller. Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[12] H. Niederreiter. Random Number Generation and Quasi-Monte Carlo Methods. *CBMS*, 63:161–216, 1992.

[13] S. Tezuka. On the Discrepancy of GFSR Pseudorandom Numbers. *JACM*, 34(4):939–949, 1987.

[14] S. Tezuka and M. Fushimi. Calculation of Fibonacci Polynomials for GFSR Sequences with Low Disccrepancies. *Mathematics of Computation*, 60(202):763–770, 1993.

[15] E. J. Watson. Primitive Polynomials (Mod 2). *Math. Comp.*, 16:368–369, 1992.

[16] E. A. Yfantis. Random Number Generator fo Electronic Applications. U.S. Patent No. 5,871,400, February 16. 1999.

[17] E. A. Yfantis, S. Baker, and G. M. Gallitano. An Image Compression Algorithm and its Implementation. *Finite Fields, Coding Theory and Advances in Communications and Computing*, 141:417–427, 1992.

[18] N. Zieler and J. Brillhart. On Primitive Trinomials (mod 2). *Information and Control*, 14:556–569, 1969.