A strong nonrandom pattern in Matlab default random number generator

Petr Savicky
Institute of Computer Science
Academy of Sciences of CR
Czech Republic
savicky@cs.cas.cz

September 16, 2006

Abstract

The default random number generator in Matlab versions between 5 and at least 7.3 (R2006b) has a strong dependence between the numbers $z_{i+1}, z_{i+16}, z_{i+28}$ in the generated sequence. In particular, there is no index i such that the inequalities $z_{i+1} < 1/4$, $1/4 \leq z_{i+16} < 1/2$, and $1/2 \leq z_{i+28}$ are satisfied simultaneously. This fact is proved as a consequence of the recurrence relation defining the generator. A random sequence satisfies the inequalities with probability 1/32. Another example demonstrating the dependence is a simple function f with values -1 and 1, such that the correlation between $f(z_{i+1}, z_{i+16})$ and $\operatorname{sign}(z_{i+28} - 1/2)$ is at least 0.416, while it should be zero.

A simple distribution on three variables that closely approximates the joint distribution of $z_{i+1}, z_{i+16}, z_{i+28}$ is described. The region of zero density in the approximating distribution has volume 4/21 in the three dimensional unit cube. For every integer $1 \leq k \leq 10$, there is a parallelepiped with edges $1/2^{k+1}$, $1/2^k$ and $1/2^{k+1}$, where the density of the distribution is 2^k. Numerical simulation confirms that the distribution of the original generator matches the approximation within small random error corresponding to the sample size.

1 Introduction

Matlab contains three generators of pseudorandom numbers, which are implemented as different methods used by function rand. According to the documentation of this function, the user may choose between the different methods using commands

```
rand('state',s)
rand('seed',s)
rand('twister',s)
```

where s is the seed. If none of these commands is used in a session, rand('state',0) is assumed. This applies to all Matlab versions between 5 and at least 7.3 (R2006b), which is the current version at the time of writing the paper.

It appears to be quite useful not to rely on the default and set 'twister', which implements Mersenne Twister generator, see [10]. The following Matlab code demonstrates the problem if the user does not set any of the methods explicitly or sets it to 'state':

```
Z = rand(28,100000);
condition = Z(1,:) < 1/4;
scatter(Z(16,condition),Z(28,condition),'.');
```

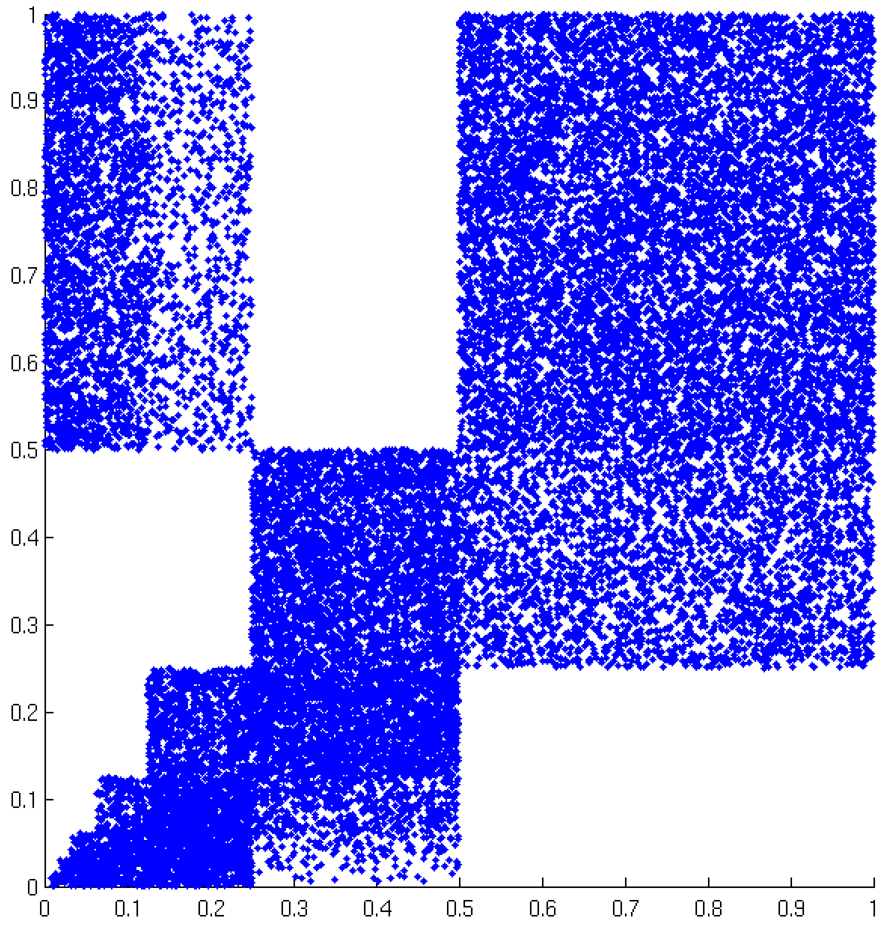

Figure 1: Scatter plot of points $Z(16, \text{condition}), Z(28, \text{condition})$.

The code constructs a random matrix of dimension 28 times 100000 and displays a scatter plot of the conditional distribution of pairs of the values from rows 16 and 28 under the condition that the corresponding value in row 1 is less than 1/4. The problem is that this code produces approximately the output presented at Figure 1. For a correct generator, the points should be uniformly distributed in the unit square, since different entries of a random matrix constructed in the above way simulate independent variables.

It is important to note that another experimental demonstration of nonuniformity of the generator is published at [13] among Technical Solutions for Matlab as Solution Number 1-10HYAS, where the problem is demonstrated using a sequence of $5 \cdot 10^7$ consecutive random numbers obtained using the default setting of rand. A reference to it was given to the author of the current paper in a reply to posting the problem at the news group of Matlab user community. See Section 3.5 for more detail. The basic documentation of rand does not contain any information on the problem.

The current paper demonstrates that the dependence is much stronger than implied by [13]. It is shown that the generator may produce stable and highly biased results already using the two most significant bits of 390 consecutive random numbers. Moreover, the reason for the dependence is determined by the analysis of the source code of the generator. In particular, the indices of the dependent numbers in the sequence are identified.

If we denote the sequence of the numbers generated by rand as z_i, then the rows 1, 16 and 28 of matrix Z used in the plot above are formed by the numbers $z_{28\,j+1}$, $z_{28\,j+16}$ and $z_{28\,j+28}$ for consecutive values of j starting at 0. The observed pattern is due to the fact that there is a strong dependence between these three numbers. Note that these triples do not overlap. The dependence remains the same, even if we consider overlapping triples $z_{i+1}, z_{i+16}, z_{i+28}$, for all i. This may be verified for example by the following code

```
Z   = rand(1,100028);
Z1  = Z(1, 1:100001);
Z16 = Z(1,16:100016);
Z28 = Z(1,28:100028);
scatter(Z16(Z1 < 1/4),Z28(Z1 < 1/4),'.');
```

The generated plot shows the same overall pattern.

In Figure 1, there are five squares of edge 1/4, which do not contain any of the points (z_{i+16}, z_{i+28}) selected using z_{i+1}. Consider the two of these squares, whose projection to the x-axis is the interval [1/4, 1/2]. Combining the condition that (z_{i+16}, z_{i+28}) belongs to their union with the condition on z_{i+1}, which was used to produce the plot, we obtain the following inequalities

$$\begin{array}{rcccl} 0 & \le & z_{i+1} & < & 1/4 \\ 1/4 & \le & z_{i+16} & < & 1/2 \\ 1/2 & \le & z_{i+28} & < & 1\ . \end{array} \qquad (1)$$

Figure 1 suggests that there is no index i such that the numbers generated from rand satisfy these three inequalities simultaneously, although a random sequence satisfies them with probability 1/32. This is indeed true as shown in Section 4.3 as a consequence of the recurrence relation used to define the generator.

The structure of the current paper is as follows. Section 2 presents general remarks on the situation. Section 3 presents several examples of Monte Carlo integrations, whose result is biased due to the discussed dependence. The technical reason for the failure of the default method of rand is discussed in detail in Section 4 using the source code of the generator. A simple distribution closely approximating the dependence is presented in Section 5. Using this approximation, the volume of the region with zero density in the distribution of $(z_{i+1}, z_{i+16}, z_{i+28})$ is shown to be close to 4/21. Moreover, an algorithm, which allows to calculate the density of the approximating distribution in any given point, is presented. Numerical comparison of the density of the approximating distribution and sample estimates of the density of the numbers from the original generator is presented in Section 5.4 using samples of sizes 10^9, 10^{10} and 10^{11}. Remarks and references concerning the complexity theoretic point of view on pseudorandom generators are presented in Section 6.

2 General remarks

The reason for the failure of the default method of rand is not that we are using pseudorandom numbers. On the contrary to the usual perception, is it consistent with the current state of knowledge to assume that efficient pseudorandom number generators with good statistical properties do exist.

There is a trivial argument against generation of pseudorandom numbers using a fixed algorithm. If we are given a sequence of consecutive numbers from the generator, we can search through the space of all possible seeds to find the one that generates exactly the given sequence. Once the seed is found, the next number may be predicted with a high probability of success or even with certainty, if the given sequence is long enough. However, this argument becomes pointless, if we consider also the complexity of the computation. For a good generator, the above scheme of prediction is infeasible due to a large computation time.

If we limit our consideration to numerical applications, whose running time is bounded by a polynomial in the size of the input, then it is consistent with the current state of knowledge in complexity theory to assume that there are generators, which pass all statistical tests computable in polynomial time. See Section 6 for more detail.

On the other hand, there is no hope to give a mathematical proof that a concrete generator passes all statistical tests computable in polynomial time. In particular, such a proof would automatically imply P≠NP, which belongs to the hardest problems in complexity theory. Hence, we have to accept that there is always a chance that a new dependence is found in a used generator. However, it is necessary to distinguish different levels of severity of such dependencies. Two important quantitative characteristics of a dependence are the following:

1. The length of the sequence of random numbers that has to be generated in order to demonstrate a significant difference from truly random numbers.

2. The computation time needed to verify the difference numerically.

Besides these quantitative criteria for measuring known dependencies, it is possible to distinguish how hard it was to find the dependence at the current state of knowledge. If someone finds an unkown dependence in a generator that was considered good at the current state of the knowledge, this may even be a mathematical discovery. It is not possible to avoid such situations. However, there is no reason to use generators containing dependencies that may easily be determined using the description of the generator.

The dependence in the method 'state' of rand is very bad in all three directions. Already a characteristic function of a parallelepiped in three out of 28 consecutive coordinates in precision of roughly two most significant bits may have highly biased expected value. If we like to formulate this in terms of statistical significance, we need more numbers. However, the two most significant bits from 390 consecutive nubers are sufficient to formulate a simple frequency test which rejects randomness of the generator at the level $1.4 \cdot 10^{-9}$ of statistical significance. In order to perform the test, a few hundreads of operations are sufficient. See Section 3.4 for more detail.

Moreover, the dependence is very easy to find. Immediately from the description of the generator, it is clear that there is a component with a known strong dependence in dimension 28. Combining the two components of the generator may eliminate this. However, the used algorithm for combining the components does not belong among the combinations recommended by the theory. It is easy to see from the description of the combining algorithm that there is a risk that the selected type of combination does not eliminate the dependence in the first component. Verifying this requires a few lines of code and a few seconds of CPU time.

Good sources of information on pseudorandom generators for simulations are for example [4], [7] and [12]. References to practical implementations of good linear generators are e.g. in papers [5], [11]. An example of an efficient and reliable nonlinear generator is presented in [3].

3 Failures in Monte Carlo integration

3.1 An event with positive probability never occurs

Let us reformulate the problem discussed in the introduction in terms of a Monte Carlo integration. It is usual to consider the nonoverlapping subsequencies $(z_{dj+1}, \ldots, z_{dj+d})$ for successive values of j as independent realizations of uniformly distributed points in the unit cube of dimension d. The example from the introduction shows that using this with the default setting of rand may lead to very biased results, if $d \geq 28$.

Assume, one likes to determine the probability that a random vector $(Z_1, \ldots, Z_{28})$ chosen uniformly from the unit cube in dimension 28 satisfies the inequalities

$$\begin{array}{ccccc} 0 & \leq & Z_1 & < & 1/4 \\ 1/4 & \leq & Z_{16} & < & 1/2 \\ 1/2 & \leq & Z_{28} & < & 1 \ , \end{array} \tag{2}$$

which are a reformulation of the inequalities (1) from the introduction. The correct answer is obviously $1/32$.

Using Monte Carlo simulation for this purpose, we may choose a large n, generate vectors $(z_{28\,j+1}, \ldots, z_{28\,j+28})$ for $j = 1, \ldots, n$ that simulate independent realizations of $(Z_1, \ldots, Z_{28})$ and count, how many of them satisfy (2). If s is the number of successes, then the resulting estimate is s/n.

If we use a good random number generator, then s/n is $1/32$ plus an error decreasing approximately as $1/\sqrt{32\,n}$. More exactly, the expected value of the result is $1/32$ and the standard deviation is $\sqrt{1/32(1 - 1/32)/n}$. The error has approximately Gaussian distribution due to the central limit theorem. In particular, if $n = 10^6$, the standard deviation is approximately $1/5700$. Hence, the error is less than $1/1000$ with high probability.

If we use the default generator in Matlab, the result is 0. This is not a random error. It is a systematic error, which does not depend on the seed and on the number n of iterations. On the other hand, the error depends on the particular choice of the indices 1, 16 and 28. For other triples of indices, everything is correct and the error decreases with $1/\sqrt{32\,n}$ as implied by the theory.

3.2 Overestimating probability by a large factor

The next example is derived using the result of Section 5.3, however, a numerical verification is possible already here. Let k be any integer satisfying $2 \le k \le 10$ and consider the probability that a vector $(Z_1, \ldots, Z_{28})$ chosen uniformly from the unit cube in dimension 28 satisfies the inequalities

$$\begin{array}{ccccc} 0 & \le & Z_1 & < & 1/2^k \\ 0 & \le & Z_{16} & < & 1/2^k \\ 0 & \le & Z_{28} & < & 1/2^k \;. \end{array} \tag{3}$$

This probability is obviously $1/2^{3k}$. Monte Carlo simulation using method 'state' of rand yields approximately $1/2^{2k+1}$ in average. Hence, the result is overestimated by a factor of 2^{k-1}.

Numerical verification is possible for any k in the specified range. For example, if we choose $k = 8$, the probability of (3) is overestimated approximately by a factor of 128, which means that generating 10^6 nonoverlapping vectors yields 7.6 successes in average instead of the correct 0.06 in average.

According to the results of Section 5, the density of the distribution of the triples Z_1, Z_{16}, Z_{28} is not constant in the region specified by inequalities (3). For any $1 \le k \le 10$ consider inequalities

$$\begin{array}{lcccl} 1/2^{k+1} & \le & Z_1 & < & 1/2^k \\ 1/2^k & \le & Z_{16} & < & 1/2^{k-1} \\ 1/2^{k+1} & \le & Z_{28} & < & 1/2^k \;, \end{array} \tag{4}$$

which determine a parallelepiped with edges $1/2^{k+1}$, $1/2^k$ and $1/2^{k+1}$. By results of Section 5.3, the density of the distribution in the region specified by (4) is constant and approximately equal to 2^k. Simulation using rand overestimates the probability of the event (4) by a factor of 2^k.

3.3 Predicting the magnitude of the next random number

It appears that it is possible to predict with success probability 0.708, whether the next number generated by rand will be at least $1/2$ or not using the history of 27 last values of rand. The prediction may be computed by evaluating 5 linear inequalities using two out of these 27 numbers and combining the results by a Boolean operation.

For a good pseudorandom number generator, the success probability of such a prediction is $1/2$ or very close to $1/2$, since nothing better (and also nothing worse) than a random guess is possible in an affordable amount of computation time. For linear generators, it is sometimes tolerated, if more efficient predictions may be done using linear algebra modulo primes,

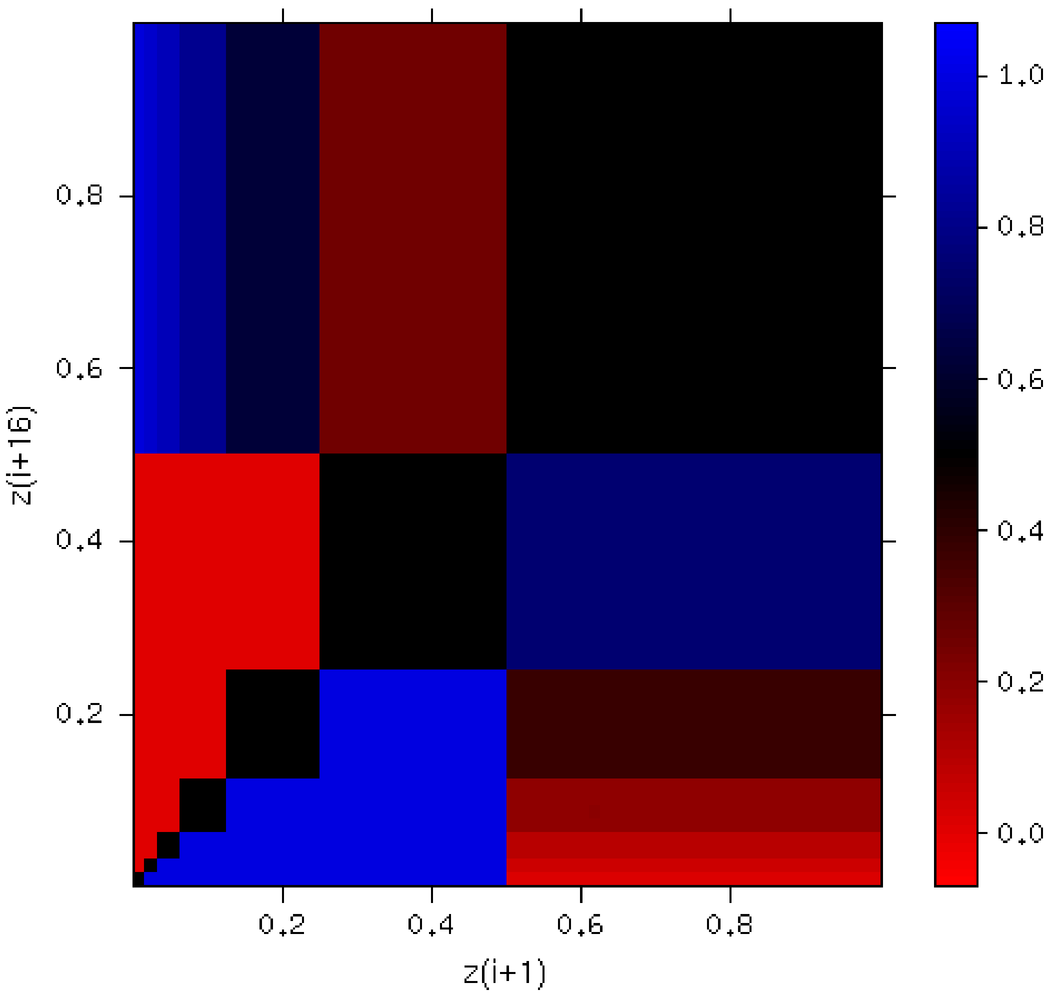

Figure 2: Map of conditional probabilities $\Pr(z_{i+28} \geq 1/2 \mid z_{i+1}, z_{i+16})$ in a grid of 64×64 squares of equal size estimated by numerical simulation.

but such predictions are defined by extremely noncontinuous functions and are unlikely to interfere with a typical numerical problem.

Let us start by reformulating the problem from the introduction in terms of prediction. We already know that for every i such that $0 \leq z_{i+1} < 1/4$ and $1/4 \leq z_{i+16} < 1/2$, we certainly have $z_{i+28} < 1/2$. In order to investigate predictions of this kind systematically, let us consider a map, where the axes correspond to the values of z_{i+1} and z_{i+16} and the colour of a point (z_{i+1}, z_{i+16}) is determined by the conditional probability of the event $z_{i+28} \geq 1/2$ under the condition that z_{i+1} and z_{i+16} have the given values. The map in Figure 2 approximates such a map using the result of simulating 10^9 nonoverlapping triples $(z_{28\,j+1}, z_{28\,j+16}, z_{28\,j+28})$ and plotting the estimated average of the required conditional probability separately in each cell of a 64 times 64 regular grid of squares, each with edge 1/64. Black colour corresponds to the conditional probability 1/2, which is the correct value. Bright blue corresponds to conditional probability 0 and bright red corresponds to conditional probability 1. The conditional probabilities between 0 and 1/2 and between 1/2 and 1 are linearly mapped to the RGB colours between blue and black and between black and red, respectively.

In particular, the square

$$\begin{array}{ccccc} 0 & \le & z_{i+1} & < & 1/4 \\ 1/4 & \le & z_{i+16} & < & 1/2 \end{array}$$

is bright blue due to the considerations above. The map, however, shows also other areas, where the truth value of $z_{i+28} \ge 1/2$ may be predicted (positively or negatively) with certainty. There are also regions, where a prediction may not be done with certainty, but a prediction with a strong correlation with $z_{i+28} \ge 1/2$ is possible. In the regions with nonzero degree of blue, it is better to predict 0. In the regions with nonzero degree of red, it is better to predict 1. In the black regions, there is not a significant difference between predicting 0 or 1. Using the map, it is easy to determine simple polygonal boundaries, which separate the regions with prediction 0 from regions with prediction 1. For example, one may use the following prediction rule. In order to simplify the calculation of correlation, the prediction of the truth value of $z_{i+28} \ge 1/2$ is transformed into a prediction of the value of $\mathrm{sign}(z_{i+28} - 1/2)$. Logical values are used in arithmetic expressions as if they are the real numbers 0 and 1.

```
if $z_{i+1} > z_{i+16}$
        prediction $= 1 - 2\,(z_{i+1} > 1/2 \;\&\; z_{i+16} < 1/4)$
else
        prediction $= 2\,(z_{i+16} > 1/2 \;\&\; z_{i+1} < 1/4) - 1$
end
```

The prediction obtained in this way is equal to $\mathrm{sign}(z_{i+28} - 1/2)$ with probability at least 0.708. An equivalent formulation is that the mean of

$$\text{prediction} \cdot \mathrm{sign}(z_{i+28} - 1/2) \tag{5}$$

is at least 0.416. The correct value of this mean is zero, since the two variables that are multiplied have zero mean and simulate independent variables. Hence, we again have a failure in a Monte Carlo integration. In this case it is calculating the mean of a function with values -1 and 1, which is constant on 8 simple nonconvex polygonal regions. The regions are defined by splitting the space with 6 planes in three coordinates and putting some of the adjacent pieces together. The first 5 planes are those used in the prediction and the last plane corresponds to the condition $1/2 \le z_{i+28}$. Then, the value of the function is specified separately on each of the resulting pieces.

Each of the two random variables prediction and $\mathrm{sign}(z_{i+28} - 1/2)$ individually has a distribution very close to the uniform one. It is not possible to distinguish their expected values from 0 using frequencies over runs shorter than the period. Hence, both these quantities may be considered as random variables with zero mean and unit standard deviation. In other words, the expected value of (5) is simultaneously the correlation of the two variables, which is at least 0.416, although it should be 0.

The numerical values may be verified using predict.m presented at [15].

3.4 An estimate of the level of statistical significance

The example of failing Monte Carlo integration from Section 3.1 uses two cubes of edge 1/4, which have zero density in the joint distribution of $z_{i+1}, z_{i+16}, z_{i+28}$. In order to identify several further such cubes, consider constructing a sequence of integers $a_i = \lfloor 4\, z_i \rfloor$. Since the joint distribution has zero density in the cube

$$\begin{array}{ccccc} 0 & \le & z_{i+1} & < & 1/4 \\ 1/4 & \le & z_{i+16} & < & 1/2 \\ 1/2 & \le & z_{i+28} & < & 3/4 \end{array}$$

there is no index i such that $(a_{i+1}, a_{i+16}, a_{i+28}) = (0, 1, 2)$, although the expected number of such indices in a really random sequence of length n is $(n - 27)/64$. Besides the missing combination $(0,1,2)$, there are 9 other missing combinations. Using this, one can design a statistical test, which rejects randomness of the default generator rand at the level $1.4 \cdot 10^{-9}$ of statistical significance using only 390 consecutive numbers. The test uses only the two most significant bits in each number and is described in the rest of the section.

Assume, we have a sequence of 390 random integers between 0 and 3 and split them into 10 vectors of length 39. Denote the coordinates of any of these vectors $(a_1, \dots, a_{39})$. Consider the 12 triples $(a_{i+1}, a_{i+16}, a_{i+28})$ for $i = 28, \dots, 39$ in each of the 10 vectors, i.e. the triples

$$(a_1, a_{16}, a_{28}), \dots, (a_{12}, a_{27}, a_{39}).$$

It is easy to verify that these triples are disjoint. Considering all the 10 vectors together, we have 120 disjoint triples. Each of the triples has $4^3 = 64$ possible combinations of the values. Let A be the event that none of these 120 triples contains a combination that belongs to the following list of 10 combinations

$$\begin{array}{c}(0,0,1),(0,1,2),(0,1,3),(0,2,0),(0,3,0),\\(1,0,0),(1,0,1),(1,1,1),(2,1,0),(3,1,0).\end{array} \tag{6}$$

The probability of the event A under the null hypothesis, which is the assumption that the input sequence is a random sequence of independent and uniformly distributed numbers, is $(54/64)^{120} < 1.4 \cdot 10^{-9}$. Hence, a test of uniformity of a sequence of pseudorandom numbers may be done as follows: Consider 390 consecutive numbers from the tested generator and calculate the corresponding numbers $a_i = \lfloor 4\, z_i \rfloor$. Then, determine for the obtained sequence of 390 integers, whether A occurs or not. If it occurs, reject the generator on the level of statistical significance $1.4 \cdot 10^{-9}$.

If we generate the 390 numbers from the default setting of rand, the event A will always occur, independently on the seed and the number of repetitions of the experiment. This statement is based on a large amount of simulations and is proved in Section 4.3 using the definition of the generator. In this situation, it is not neccessary to present the result of a concrete run of the test as usual in other situations. It is possible to make a strong statement that whenever the test is performed, it always rejects the generator. A single repetition of the test rejects the uniformity of the generator on the level $1.4 \cdot 10^{-9}$ of statistical significance. Repeating the test makes, of course, the rejection even stronger, but needs a longer sequence of random numbers.

It is possible to formulate other tests, which use the results of Section 5, for example the region of zero density of volume 4/21 or the approximated density. Such tests allow to reject the generator on the same level of significance as above using a sequence of consecutive random numbers of length smaller than 390. These tests are not presented, since their description is more complex than the above one.

3.5 Related problem

A related problem, caused by the same property of the default setting of rand, is described at web pages of MathWorks among Technical Solutions as Solution Number 1-10HYAS, see [13]. The problem is demonstrated using a sequence of $5 \cdot 10^7$ consecutive numbers generated from rand. This sequence is split into runs consisting of consecutive numbers which are at least 1/20 and ended by one number less than 1/20. In the histogram of the frequencies of different lengths of these runs, the point corresponding to length 27 is visibly below the curve formed by the points for other lengths. The reason is that length 27 corresponds to the event B defined as

$$\begin{array}{rcll} z_{i+1} & < & 1/20 & \\ z_{i+j} & \geq & 1/20 & \text{for all } j = 2, \dots, 27 \\ z_{i+28} & < & 1/20 & \end{array}$$

and the probability of B is smaller than it should be. In order to demonstrate that this is caused by the dependence between the numbers z_{i+1}, z_{i+16} and z_{i+28} discussed in the current paper, let us consider the approximation of the conditional probability

event	type of quantity	numerical value
B	expected probability	$q \cdot (19/20) \cdot (35/64)$
	observed probability	$3.7936 \cdot 10^{-4}$
	ratio obs./exp.	1.000319
C_{16}	expected probability	$q \cdot (19/20) \cdot (1 - 35/64)$
	observed probability	$3.1428 \cdot 10^{-4}$
	ratio obs./exp.	1.000169
C_k, $k \neq 16$	expected probability	$q \cdot (1/20) \cdot (35/64)$
	minimum observed	$1.9937 \cdot 10^{-5}$
	maximum observed	$1.9983 \cdot 10^{-5}$
	ratio obs./exp.	1 ± 0.0012

Figure 3: Numerical simulation using sample of size 10^{11}.

$\Pr(z_{i+16} \geq 1/20 \mid z_{i+1} < 1/20,\ z_{i+28} < 1/20)$ implied by the analysis in Section 5.1. By considering the probability of the intersection of the regions $R(k_1, k_{16}, k_{28})$ with the event B for all $1 \leq k_1, k_{16}, k_{28} \leq 5$, it may be verified that in the distribution defined by (14), we have

$$\Pr(Z_{16} \geq 1/20 \mid Z_1 < 1/20,\ Z_{28} < 1/20) = 35/64.$$

Using this, the probability of B may be approximated by $q \cdot (19/20) \cdot (35/64)$, where $q = (1/20)^2 \cdot (19/20)^{24}$. The result of a numerical simulation using sample of size 10^{11} is consistent with this assumption, see the table in Figure 3.

In order to formulate a stronger test, let us consider the events C_k for $k = 2, \ldots, 27$ defined as

$$\begin{array}{lll} z_{i+j} & < & 1/20 \quad \text{for } j = 1, k, 28 \\ z_{i+j} & \geq & 1/20 \quad \text{otherwise .} \end{array}$$

Note that these events are disjoint with B and they are also mutually disjoint. Using the assumption that the dependence between $(z_{i+1}, z_{i+16}, z_{i+28})$ is the only dependence between the numbers $(z_{i+1}, z_{i+2}, \ldots, z_{i+28})$ and using the results of Section 5.1, we obtain the approximation

$$\Pr(C_k) = \begin{cases} q \cdot (19/20) \cdot (1 - 35/64) & k = 16 \\ q \cdot (1/20) \cdot (35/64) & k \neq 16 \end{cases}$$

The result of the numerical simulation using sample of size 10^{11} is presented in the table in Figure 3. For the events C_k, $k \neq 16$, the table contains only the minimum and maximum of the observed probabilities. The obtained numerical results are consistent with the prediction.

4 Explanation of the problem

4.1 A brief description

The default method 'state' used by rand is a combination of two components: Marsaglia's subtract with borrow generator and a 32-bit linear generator modulo 2 (using xor operation). Subtract with borrow generators were suggested in [9] and are described e.g. in Section 3.6 Linear Recurrences with Carry of [7].

The numbers generated by the first component, the subtract with borrow generator, have a strong dependence between x_{i+1}, x_{i+16} and x_{i+28}, where x_i is the i-th element of the sequence. This dependence is a direct consequence of the formula defining the sequence. A dependence of this kind may be eliminated, if the output z_i of the whole generator is constructed by combining x_i with the output of the second component.

The standard way of combining two generators is to use some group operation. The two simplest group operations, which may be used, are addition of real numbers modulo 1 (inside the half closed interval $[0, 1)$) and the xor operation (addition of the bits modulo 2) applied to the bits of the two numbers at the same positions in a fixed point representation. Such a combination may provably eliminate a dependence in a generator, if it is combined with a generator that does not have this particular dependence, see [6].

Unfortunately, the combination used in rand is not a group operation. It is done by applying xor operation to the bits in floating point representation and not in fixed point. Floating point representation uses an exponent to shift the digits so that the first nonzero digit is always at the same position. Moreover, the first nonzero digit is not explicitly contained in the machine representation. The combination used in rand constructs the output z_i by a xor operation, which adds the bits of the second component y_i to those bits of the mantissa (significand) of x_i, which are present in the machine representation. The result of this operation is the mantissa of the z_i. The exponent of x_i is preserved and used as the exponent of z_i. Hence, the first nonzero digit of x_i and the possible zero digits before it are not altered. As a consequence, the position of the first nonzero digit in the output z_i is the same as in the subtract with borrow component x_i. This similarity in magnitude between x_i and z_i allows the strong dependence in x_i to be transformed into a slightly weaker, but still very strong dependence in the output z_i.

It is important to note that the selected type of combination guarantees that the mantissa of the numbers generated by rand consists of 53 random looking bits, even in the cases, when the generated number is small. Hence, the numbers look random up to the full precision of double. This would not be true, if the combination of x_i and y_i is done in fixed point arithmetic, since then the bits of the order 2^{-54} and below would always be zero. However, the advantage of this is questionable. For example, the same is not true for $1 -$ rand, so the user has to be carefull to use this. Moreover, if the user has a specific reason to use full precision floating point random numbers, then the same effect may be achieved by combining two consecutive numbers in the sequence appropriately. Hence, the advantage of having full precision random numbers does not seem to overweight the spectacular loss in uniformity of the output in dimensions 28 and more.

4.2 An exact description

MathWorks provides an exact description of the default generator among Technical Solutions as "Solution Number: 1-169M3", see [14]. At this web page, one may find an m-file randtx.m, which is, of course, slower than rand, but exactly reproduces the default numerical behaviour of rand. The statements concerning the structure of the generator, which are formulated in the previous section, are based on an analysis of this file.

The subtract with borrow generator used in rand is defined by the recurrence

$$x_i = (x_{i-12} - x_{i-27} - b_{i-1}) \bmod 1 \tag{7}$$

$$b_i = \begin{cases} 0 & \text{if } x_{i-12} - x_{i-27} - b_{i-1} \geq 0 \\ 2^{-53} & \text{otherwise} \end{cases} , \tag{8}$$

where x_i are double precision numbers between 0 and $1 - 2^{-53}$ of the form $m2^{-53}$, where m is an integer, $0 \leq m \leq 2^{53} - 1$. The mod 1 operator means that its argument is mapped to $[0, 1)$ by adding an integer. In the recurrence above, its argument is always in $[-1, 1)$, so only adding 1 or 0 may occur.

For the purpose of the current paper, the exact definition of b_i is not important and the fact that for each i either $b_i = 0$ or $b_i = 2^{-53}$ is sufficient. This implies that x_{i+28} differs from either $x_{i+16} - x_{i+1}$ or from $x_{i+16} - x_{i+1} + 1$ by at most 2^{-53}. This is a strong dependence.

The second component of the generator used in rand is a linear generator modulo 2 generating nonzero unsigned 32-bit integers with the period $2^{32} - 1$. In each step of the generator, two elements of this sequence are computed and combined together into a 52-bit integer, which we denote y_i.

The exact combination used in rand may be found at the last 4 lines in randtx.m and is as follows. The resulting number z_i is constructed using x_i and y_i. Let us denote the binary digits after the decimal point in x_i as $a_1, a_2, \ldots$ and the binary digits in y_i as $b_1, b_2, \ldots$ Hence, we have

$$\begin{aligned} x_i &= 0.\ a_1\ a_2\ a_3\ \ldots\ a_{53}\ , \\ y_i &= b_1\ b_2\ b_3\ \ldots\ b_{52}\ . \end{aligned}$$

If $a_1 = 1$, then

$$\begin{aligned} x_i &= 0.\ 1\ a_2\ a_3\ \ldots\ a_{53}\ , \\ z_i &= 0.\ 1\ (a_2 \oplus b_1)\ (a_3 \oplus b_2)\ \ldots\ (a_{53} \oplus b_{52})\ . \end{aligned}$$

where $\oplus$ denotes addition modulo 2, i.e. the xor operation. If $a_1 = 0$ and $a_2 = 1$, then

$$\begin{aligned} x_i &= 0.\ 0\ 1\ a_3\ \ldots\ a_{53}\ , \\ z_i &= 0.\ 0\ 1\ (a_3 \oplus b_1)\ \ldots\ (a_{53} \oplus b_{51})\ b_{52}\ . \end{aligned}$$

In general, the bits of y_i are used to modify 52 bits in x_i starting at the bit after the first nonzero bit of x_i. Consequently, the positions of the first nonzero bit in z_i and x_i are the same. Note that this implies that for every integer k, we have

$$1/2^k \le x_i < 2/2^k \tag{9}$$

if and only if

$$1/2^k \le z_i < 2/2^k\ . \tag{10}$$

This similarity between z_i and x_i implies that the dependence between x_{i+1}, x_{i+16} and x_{i+28} transforms into a dependence between z_{i+1}, z_{i+16} and z_{i+28}. This remains true, even if y_i are perfect random numbers.

4.3 Examples of cubes with zero density

In this section, we provide proofs of two statements promissed earlier. The first theorem implies that the inequalities (1) are never satisfied. This is the example of a nonuniform property of the investigated generator discussed in the introduction and Section 3.1. The second theorem implies that the statistical test described in Section 3.4 always rejects the generator, independently of the seed. Although the second theorem implies the first one, we present both, since the proof of the first one is simpler.

Let us consider the following consequence of (7). For every i, there is a number $b \in \{0, 2^{-53}\}$, such that

$$x_{i+28} = (x_{i+16} - x_{i+1} - b) \bmod 1\ , \tag{11}$$

Theorem 1 *There is no index i such that the sequence z_i of the generator satisfies the inequalities (1).*

Proof. Let us use the notation x_i, y_i for the two components of the generator as in Section 4.2. Due to the equivalence between (9) and (10), it is sufficient to prove that the inequalities

$$\begin{array}{ccccc} 0 & \le & x_{i+1} & < & 1/4 \\ 1/4 & \le & x_{i+16} & < & 1/2 \\ 1/2 & \le & x_{i+28} & < & 1\ . \end{array} \tag{12}$$

are never satisfied. By subtracting the first two of them, we obtain

$$0 < x_{i+16} - x_{i+1} < 1/2\ .$$

Since the numbers x_i are multiples of 2^{-53}, we have

$$2^{-53} \le x_{i+16} - x_{i+1} < 1/2\ .$$

This implies

$$0 \le x_{i+16} - x_{i+1} - b < 1/2\ ,$$

where b is the number from (11). Hence, we have

$$0 \le x_{i+28} < 1/2 \ .$$

This is inconsistent with the last of the inequalities (12) and, hence, the inequalities are never satisfied simultaneously. □

For every i, let $a_i = \lfloor 4z_i \rfloor$. Note that the above theorem implies the following. There is no index i such that the triple $(a_{i+1}, a_{i+16}, a_{i+28})$ is equal to $(0, 1, 2)$ or $(0, 1, 3)$, which are the second and third of the triples (6) considered in Section 3.4. The following theorem extends this to all 10 triples (6). Hence, the statistical test described in Section 3.4 indeed always rejects the generator, independently of the seed.

Theorem 2 *Then there is no index i such that the triple $(a_{i+1}, a_{i+16}, a_{i+28})$ is equal to some of the triples (6).*

Proof. We need to work with intervals of values of z_i, whose bounds are 0 or a power of 2. Hence, we consider the values $a_i = 2$ and $a_i = 3$ together, and use the following correspondence of values a_i and intervals

a_i	z_i
0	$[0, 1/4)$
1	$[1/4, 1/2)$
2, 3	$[1/2, 1)$.

Using this, the 10 triples (6) correspond to 7 Cartesian products of intervals of values of $(z_{i+1}, z_{i+16}, z_{i+28})$ as follows. The 4 triples consisting only of 0,1 correspond directly to products of appropriately selected intervals $[0, 1/4)$ and $[1/4, 1/2)$. For example, the triple $(1, 0, 1)$ corresponds to $[1/4, 1/2) \times [0, 1/4) \times [1/4, 1/2)$. The remaining 6 triples containing also 2 or 3 are combined into pairs that correspond to products of intervals as follows

$(a_{i+1}, a_{i+16}, a_{i+28})$	$(z_{i+1}, z_{i+16}, z_{i+28})$
$(0, 1, 2), (0, 1, 3)$	$[0, 1/4) \times [1/4, 1/2) \times [1/2, 1)$
$(0, 2, 0), (0, 3, 0)$	$[0, 1/4) \times [1/2, 1) \times [0, 1/4)$
$(2, 1, 0), (3, 1, 0)$	$[1/2, 1) \times [1/4, 1/2) \times [0, 1/4)$.

Let $[r_1, s_1) \times [r_{16}, s_{16}) \times [r_{28}, s_{28})$ be any of the obtained 7 products of intervals. The triple $(z_{i+1}, z_{i+16}, z_{i+28})$ belongs to it iff

$$\begin{array}{ccccc} r_1 & \le & z_{i+1} & < & s_1 \\ r_{16} & \le & z_{i+16} & < & s_{16} \\ r_{28} & \le & z_{i+28} & < & s_{28} \ . \end{array}$$

Due to the equivalence between (9) and (10), this system of inequalities has a solution if and only if the system

$$\begin{array}{ccccc} r_1 & \le & x_{i+1} & < & s_1 \\ r_{16} & \le & x_{i+16} & < & s_{16} \\ r_{28} & \le & x_{i+28} & < & s_{28} \end{array} \qquad (13)$$

has a solution. By a calculation similar to the one in the proof of Theorem 1 the first two inequalities imply

$$r_{16} - s_1 \le x_{16} - x_1 - b < s_{16} - r_1 \ ,$$

where b is the number from (11). Hence, we have

$$x_{i+28} \in [r_{16} - s_1, s_{16} - r_1) \cup [r_{16} - s_1 + 1, s_{16} - r_1 + 1) \ .$$

It is possible to verify that this is inconsistent with $x_{i+28} \in [r_{28}, s_{28})$ for each of the 7 considered products $[r_1, s_1) \times [r_{16}, s_{16}) \times [r_{28}, s_{28})$. This completes the proof. □

5 Density of the distribution

5.1 Approximating distribution

The analysis of the combination of x_i and y_i presented above may be used to describe a simple distribution on three numbers (Z_1, Z_{16}, Z_{28}), which is a good approximation of the joint distribution of $(z_{i+1}, z_{i+16}, z_{i+28})$. The approximation is obtained by the following changes in the recurrence relations defining the generator:

- the distribution of x_{i+1}, x_{i+16} and $y_{i+1}, y_{i+16}, y_{i+28}$ is assumed to be uniform of $[0, 1)$ and these numbers are assumed to be independent.
- the number x_{i+28} is determined by (7), where we assume $b_{i+27} = 0$.

Besides this, the combination of $x_{i+1}, x_{i+16}, x_{i+28}$ and $y_{i+1}, y_{i+16}, y_{i+28}$ to obtain $z_{i+1}, z_{i+16}, z_{i+28}$ is done in the same way as in rand, which is described in Section 4.2.

For convenience of the reader, let us describe the approximating distribution in a self-contained manner. Let (X_1, X_{16}, X_{28}) be a random vector with the uniform distribution on the intersection of the unit cube with the two planes defined by

$$\begin{aligned} X_{28} &= X_{16} - X_1 \\ X_{28} &= X_{16} - X_1 + 1 \ . \end{aligned}$$

A random point (X_1, X_{16}, X_{28}) distributed uniformly over this intersection may be generated by taking (X_1, X_{16}) from the uniform distribution on the unit square and setting $X_{28} = (X_{16} - X_1) \bmod 1$.

Let (Y_1, Y_{16}, Y_{28}) be chosen from the uniform distribution on $[0, 1)^3$. The numbers are used as infinite sequencies of random bits. If the first 52 of these bits in each of the three numbers are used as a binary representation of an integer, then we have a distribution that has the same range of values as the numbers $(y_{i+1}, y_{i+16}, y_{i+28})$, but is uniform on all triples of 52 bit numbers. The original numbers are not uniform simply because we have only $2^{32} - 1$ combinations, which is much less than the needed 2^{156}. However, nonuniformity of the triples $(y_{i+1}, y_{i+16}, y_{i+28})$ is not the reason for the discussed problem, so we neglect it.

Combination of (X_1, X_{16}, X_{28}) and (Y_1, Y_{16}, Y_{28}), which generates the resulting vector (Z_1, Z_{16}, Z_{28}) is done by adding modulo 2 the digits of the number Y_i startig at the first position after the decimal point to the digits of X_i starting after the first nonzero digit. Since we assume that Y_1, Y_{16}, Y_{28} are independent and uniformly distributed, we may describe the combination as an application of the same randomized rule to each of the three numbers of (X_1, X_{16}, X_{28}) independently. The rule is described in the next paragraph.

Let X_i for $i = 1, 16, 28$ be the original number and let k_i be the integer such that X_i belongs to the half open interval $[1/2^{k_i}, 2/2^{k_i})$. Since we assume that Y_i is uniformly distributed and independent from other random variables used in the construction, the digits of X_i that are after the first nonzero digit have no influence on the distribution of the result and may be discarded. Hence, the resulting number Z_i belongs to the same interval $[1/2^{k_i}, 2/2^{k_i})$ as X_i, but is uniformly distributed in this interval. Note that k_i is the common position of the first nonzero digit in X_i and Z_i. Hence, the resulting combination of X_i and Y_i has the same distribution as $2^{k_i}(1 + U_i) = 2^{\lfloor \log_2(X_i) \rfloor}(1 + U_i)$, where U_i is uniformly distributed in $[0, 1)$ and independent from other numbers in the construction.

Let us summarize the construction of the approximation. The input are five uniformly distributed independent numbers $X_1, X_{16}, U_1, U_{16}, U_{28}$ from the interval $[0, 1)$. Then

$$\begin{aligned} X_{28} &= (X_{16} - X_1) \bmod 1 \\ Z_1 &= 2^{\lfloor \log_2(X_1) \rfloor}(1 + U_1) \\ Z_{16} &= 2^{\lfloor \log_2(X_{16}) \rfloor}(1 + U_{16}) \\ Z_{28} &= 2^{\lfloor \log_2(X_{28}) \rfloor}(1 + U_{28}) \ . \end{aligned} \tag{14}$$

The distribution of the resulting triple (Z_1, Z_{16}, Z_{28}) is a good approximation of the distribution of $(z_{i+1}, z_{i+16}, z_{i+28})$. An empirical verification of the closeness of the two distributions is presented in Section 5.4.

The density of the approximating distribution may be calculated in any given point in the unite cube. The rest of the section is devoted to a description of this calculation. It follows from (14) that for any natural numbers $k_1, k_{16}, k_{28} \geq 1$, the distribution of the point (Z_1, Z_{16}, Z_{28}) has constant density on each of the regions $R(k_1, k_{16}, k_{28})$, where $R(k_1, k_{16}, k_{28})$ is the set of points satisfying the inequalities

$$\begin{array}{ccccc} 1/2^{k_1} & \leq & Z_1 & < & 2/2^{k_1} \\ 1/2^{k_{16}} & \leq & Z_{16} & < & 2/2^{k_{16}} \\ 1/2^{k_{28}} & \leq & Z_{28} & < & 2/2^{k_{28}} \ . \end{array}$$

In order to match the real generator, using numbers k_1, k_{16}, k_{28} above 53 is not meaningful. However, for the sake of simplicity of the description of the approximation, we work with a model, where k_1, k_{16}, k_{28} are not bounded from above.

For every $k_1, k_{16}, k_{28} \geq 1$, let

$$p(k_1, k_{16}, k_{28}) = \Pr[(Z_1, Z_{16}, Z_{28}) \in R(k_1, k_{16}, k_{28})] \ .$$

Since the randomization rule using U_i preserves the regions $R(k_1, k_{16}, k_{28})$, we have

$$p(k_1, k_{16}, k_{28}) = \Pr[(X_1, X_{16}, X_{28}) \in R(k_1, k_{16}, k_{28})] \ .$$

For every $r \in \{0, 1\}$, let $D(k_1, k_{16}, k_{28}, r)$ be the event

$$\begin{array}{ccccc} 1/2^{k_1} & \leq & X_1 & < & 2/2^{k_1} \\ 1/2^{k_{16}} & \leq & X_{16} & < & 2/2^{k_{16}} \\ 1/2^{k_{28}} & \leq & X_{16} - X_1 + r & < & 2/2^{k_{28}} \ . \end{array} \tag{15}$$

Moreover, let

$$p_r(k_1, k_{16}, k_{28}) = \Pr[D(k_1, k_{16}, k_{28}, r)] \ . \tag{16}$$

Note that $D(k_1, k_{16}, k_{28}, r)$ is a predicate depending only on X_1, X_{16}. The predicate is satisfied inside a simple convex polygonal region in the unit square, which is the set of values of X_1, X_{16} satisfying (15). Hence, the probability of the event $D(k_1, k_{16}, k_{28}, r)$ may easily be computed using elementary analytical geometry.

Using the definition of X_{28}, we obtain

$$(X_1, X_{16}, X_{28}) \in R(k_1, k_{16}, k_{28}) \iff D(k_1, k_{16}, k_{28}, 0) \vee D(k_1, k_{16}, k_{28}, 1) \ .$$

Moreover, the events $D(k_1, k_{16}, k_{28}, 0)$ and $D(k_1, k_{16}, k_{28}, 1)$ cannot occur simultaneously, since otherwise $1/2^{k_{28}} + 1 < 2/2^{k_{28}}$, which is not possible, since $k_{28} \geq 1$. Hence, we have

$$p(k_1, k_{16}, k_{28}) = p_0(k_1, k_{16}, k_{28}) + p_1(k_1, k_{16}, k_{28}) \ . \tag{17}$$

The density of the distribution defined by (14) is constant within the region $R(k_1, k_{16}, k_{28})$. Hence, this density is $2^{k_1 + k_{16} + k_{28}} p(k_1, k_{16}, k_{28})$, which is the ratio of the probability and the volume of $R(k_1, k_{16}, k_{28})$.

Using (15), (16) and (17), it is possible to calculate the density in any region $R(k_1, k_{16}, k_{28})$ for given numbers k_1, k_{16}, k_{28}. We do not present more explicit formulas, however, we present examples of the calculated densities. Table in Fig. 4 contains densities in the regions $R(k_1, k_{16}, k_{28})$ for $1 \leq k_1, k_{16}, k_{28} \leq 3$. These regions cover the points Z_1, Z_{16}, Z_{28} in the unite cube, where $\min\{Z_1, Z_{16}, Z_{28}\} \geq 1/4$. For completeness of the table, it contains also average densities over several further areas covering the remaining parts of the unite cube. These are infinite unions of $R(k_1, k_{16}, k_{28})$, where one or more of the indices k_i ranges over all numbers $k_i \geq 4$. The densities in these additional regions are not constant. However, since each of these additional regions R' is a union of $R(k_1, k_{16}, k_{28})$ for an appropriate set of indices, we again have

$$\Pr[(Z_1, Z_{16}, Z_{28}) \in R'] = \Pr[(X_1, X_{16}, X_{28}) \in R']$$

as in the case of individual regions $R(k_1, k_{16}, k_{28})$. This allows to calculate these probabilities using a finite calculation by generalizing the approach above. The details are not presented.

k_1	k_{28}	$k_{16}=1$	$k_{16}=2$	$k_{16}=3$	$k_{16}\geq 4$
1	1	1	1.5	0.75	0.25
2	1	0.5	1	2	2
1	2	0.5	1	2	2
3	1	1.25	0	1	2
2	2	2	0	0	0
1	3	1.25	0	1	2
3	2	1.5	1	0	0
2	3	1.5	1	0	0
3	3	0	4	0	0
≥ 4	1	1.75	0	0	1
1	≥ 4	1.75	0	0	1
≥ 4	2	0.5	3	0	0
2	≥ 4	0.5	3	0	0
≥ 4	3	0	2	4	0
3	≥ 4	0	2	4	0
≥ 4	≥ 4	0	0	4	4

Figure 4: Density in some of the regions $R(k_1, k_{16}, k_{28})$ and average density in some of their infinite unions.

5.2 Zero density

Using (15), (16) and (17), it is possible to identify the region of zero density in the distribution defined by (14). The volume of this region is 4/21. The rest of this section is devoted to the proof of this fact.

The range of $X_{16} - X_1 + r$, if X_1, X_{16} satisfy the first two conditions in (15), is the interval

$$\left(\frac{1}{2^{k_{16}}} - \frac{2}{2^{k_1}} + r, \frac{2}{2^{k_{16}}} - \frac{1}{2^{k_1}} + r\right) .$$

Hence, the event (15) is satisfied by at least one pair of values of X_1, X_{16}, if and only if

$$\begin{aligned} \frac{1}{2^{k_1}} - \frac{2}{2^{k_{16}}} + \frac{1}{2^{k_{28}}} &< r \\ \frac{2}{2^{k_1}} - \frac{1}{2^{k_{16}}} + \frac{2}{2^{k_{28}}} &> r . \end{aligned} \tag{18}$$

Note that under these conditions, the probability of (15) is positive. Using this, we obtain the following.

Lemma 1 *For every $k_1, k_{16}, k_{28} \geq 1$, $p(k_1, k_{16}, k_{28})$ is positive iff at least one of the two following conditions is satisfied:*

$$\frac{1}{2} < 2^{k_{16}-k_1} + 2^{k_{16}-k_{28}} < 2 , \tag{19}$$

$$2^{k_{16}-k_1} + 2^{k_{16}-k_{28}} > 2^{k_{16}-1} + \frac{1}{2} . \tag{20}$$

Moreover, these two conditions are never satisfied simultaneously.

Proof. If $r = 0$, then (18) is equivalent to (19). If $r = 1$, then (18) is equivalent to (20). If both (19) and (20) are satisfied, then $2^{k_{16}-1} + \frac{1}{2} < 2$, which implies $k_{16} = 1$. Then $2 > 2^{1-k_1} + 2^{1-k_{28}} > \frac{3}{2}$, which is not possible, since k_1 and k_{28} are integers. □

Theorem 3 *The sum of the volumes of areas $R(k_1, k_{16}, k_{28})$ with zero density is* 4/21.

Proof. We will prove that the sum of volumes of areas $R(k_1, k_{16}, k_{28})$ satisfying (19) is 11/28 and satisfying (20) is 5/12. Hence, zero density is in volume $1 - 11/28 - 5/12 = 4/21$.

Let us prove the above statements. The situations, where condition (19) is satisfied, consist of the five mutually exclusive cases listed in the following table.

case	k_1	k_{16}	k_{28}	volume
$k_1 = k_{28} \wedge k_{16} = k_1 - 1$	$i+1$	i	$i+1$	1/28
$k_1 < k_{28} \wedge k_{16} = k_1$	i	i	$i+j$	1/7
$k_1 < k_{28} \wedge k_{16} = k_1 - 1$	$i+1$	i	$i+j+1$	1/28
$k_1 > k_{28} \wedge k_{16} = k_{28}$	$i+j$	i	i	1/7
$k_1 > k_{28} \wedge k_{16} = k_{28} - 1$	$i+j+1$	i	$i+1$	1/28

In each row, a parametrization of all k_1, k_{16}, k_{28} satisfying the corresponding condition is given using integer parameters $i, j \geq 1$. Using this parametrization, the sum of $1/2^{k_1+k_{16}+k_{28}}$ over the triples in the given row may be calculated using one of the identities

$$\sum_{i,j\geq 1} \frac{1}{2^{\alpha i+j+\beta}} = \sum_{i\geq 1} \frac{1}{2^{\alpha i+\beta}} = \frac{1}{(2^\alpha - 1)2^\beta} \,.$$

For example, in the first row, we have $k_1 + k_{16} + k_{28} = \alpha i + \beta = 3i + 2$, which yields volume 1/28. In the second row, we have $k_1 + k_{16} + k_{28} = \alpha i + j + \beta = 3i + j$, which yields volume 1/7.

Similarly, for condition (20), we obtain the following three cases

case	k_1	k_{16}	k_{28}	volume
$k_1 = 1 \wedge k_{28} = 1$	1	i	1	1/4
$k_1 = 1 \wedge k_{28} \geq 2 \wedge k_{16} \geq k_{28}$	1	$i+j$	$i+1$	1/12
$k_1 \geq 2 \wedge k_{28} = 1 \wedge k_{16} \geq k_1$	$i+1$	$i+j$	1	1/12

Summarizing the results finishes the proof. □

5.3 High density

Let $k \geq 1$ be an integer. Using (14), we obtain that $Z_1 < 1/2^k, Z_{16} < 1/2^k$ iff $X_1 < 1/2^k, X_{16} < 1/2^k$. Hence, the conditional probability

$$\Pr(Z_{28} < 1/2^k \mid Z_1 < 1/2^k, Z_{16} < 1/2^k)$$

is equal to

$$\Pr(Z_{28} < 1/2^k \mid X_1 < 1/2^k, X_{16} < 1/2^k)\,.$$

If we extend the condition by the assumption $X_1 < X_{16}$, then the condition implies $0 < X_{28} < 1/2^k$ and, hence, $0 < Z_{28} < 1/2^k$ with certainty. On the other hand, if $X_1 > X_{16}$, then $0 < Z_{28} < 1/2^k$ is never satisfied. Hence,

$$\Pr(Z_{28} < 1/2^k \mid Z_1 < 1/2^k, Z_{16} < 1/2^k) = \frac{1}{2}\,.$$

This implies that

$$\Pr(Z_1 < 1/2^k,\ Z_{16} < 1/2^k,\ Z_{28} < 1/2^k) = \frac{1}{2^{2k+1}}\,.$$

This proves the statement concerning the event (3) in Section 3.2. Under the uniform distribution, the probability of the same event is $1/2^{3k}$. Hence, the probability in the model (14) is larger by a factor of 2^{k-1} than in the uniform distribution.

sample size	χ^2	p-val (χ^2)	D$^+$	D$^-$
10^9	211196.7	0.944323	0.002864007	0.003117393
10^9	212890.8	0.1563487	0.002058352	0.004033333
10^9	211706.2	0.7904987	0.002284093	0.004099419
10^9	212135.4	0.5591362	0.00223734	0.002803002
10^{10}	211043.7	0.9662099	0.001746556	0.001839517
10^{10}	212714.4	0.2298325	0.001603772	0.001746639
10^{11}	212018.3	0.6288094	0.001368394	0.001015529
10^{11}	213030.5	0.1105557	0.001527936	0.001700189

Figure 5: Results of numerical comparison of the predicted density with the density of numbers generated from rand.

The density in the region used above is not constant. Consider the event (4) from Section 3.2. By (17), its probability may be expressed as $p(k+1,k,k+1) = p_0(k+1,k,k+1) + p_1(k+1,k,k+1)$. The system of inequalities (15) corresponding to $p_0(k+1,k,k+1)$ is

$$\begin{array}{lcccl} 1/2^{k+1} & \le & X_1 & < & 1/2^k \\ 1/2^k & \le & X_{16} & < & 1/2^{k-1} \\ 1/2^{k+1} & \le & X_{16} - X_1 & < & 1/2^k \ . \end{array}$$

Note that the second inequality is the sum of the first and the third one. Hence, the second inequality may be omitted. The solution of the remaining two inequalities may be parametrized as

$$\begin{array}{rcl} X_1 & = & t \\ X_{16} & = & s + t \end{array}$$

where $t, s \in [1/2^{k+1}, 1/2^k)$. Hence, the area of the set of solutions is $1/2^{2k+2}$. The system of inequalities (15) corresponding to $p_1(k+1,k,k+1)$ is not satisfiable. Hence, the probability of (4) is $1/2^{2k+2}$, while the correct value is $1/2^{3k+2}$. Since the density in this region is constant, both the probability and the density are larger by a factor of 2^k than in the uniform distribution.

5.4 Numerical simulation

Since the model (14) is only an approximation, the density of the distribution defined by (14) has been compared with estimates of the density of the original generator based on numerical simulation. The default method of rand was used to generate samples of nonoverlapping triples (Z_1, Z_{16}, Z_{28}) of sizes 10^9, 10^{10} and 10^{11} in Matlab. The unit cube was partitioned into a grid of 64^3 disjoint cubes of edge 1/64. Then, the number of the generated triples that fall into each of these cubes was counted. The obtained frequencies were compared to the expected value derived from (14).

There are $n_0 = 49910$ cells among the 64^3 cells described above, which have zero probability in the model (14). These n_0 cells are exactly the cells, where the actual observed frequency was 0. Only cells with positive probability in the model were used for further tests.

We used two types of tests. First, the standard χ^2 test with $N = 64^3 - n_0$ classes was used. This test did not detect a significant difference between expected and observed frequencies. The p-values are presented in the table in Fig. 5.

Using the usual notation, the χ^2 statistic is calculated as

$$\sum_{i=1}^{N} \frac{(O_i - E_i)^2}{E_i}$$

Under the null hypothesis, the numbers

$$\frac{O_i - E_i}{\sqrt{E_i}} \tag{21}$$

should be distributed approximately as in the normal distribution $\mathrm{N}(0,1)$. Since the number of classes N is quite large, we may afford to test, whether the numerical results correspond to this assumption. For this purpose, we calculated the KS test statistics for the numbers (21) obtained for several samples of triples of different sizes. The statistics are denoted D^+ and D^- and are the maximum positive and the maximum negative difference between the sample distribution function of the numbers (21) and the distribution function of $\mathrm{N}(0,1)$. These statistics are presented in the table in Figure 5.

The p-value of the standard KS test may not be computed in this case, since the numbers (21) contain a lot of ties. The reason for these ties is that many cells have the same expected number of hits and if this expected number is small, then some part of these cells gets also the same observed frequency. This implies the same value of (21). However, the differences between the two distribution functions are quite small and this confirmes a good correspondence between the prediction and the simulation results.

6 Random number generators and computational complexity

From the point of view of computational complexity, a generator is good, if its output is not distinguishable from a random sequence in polynomial time. Such a generator is called polynomially secure and existence of such generators is consistent with the current knowledge of complexity theory. On the other hand, proving that polynomially secure generators do exist is beyond the reach of the known methods, since existence of such a generator implies P≠NP and even more.

Let us present a slightly simplified definition of a polynomially secure pseudorandom generator used e.g. in monographs [1] Chapter 3, [2], [8].

A generator of pseudorandom bits is a function $g : \{0,1\}^s \to \{0,1\}^n$, where s is the length of the seed. It is assumed that the seed x is a truly random sequence with uniform distribution on $\{0,1\}^s$ and $g(x)$ is the resulting pseudorandom vector of length n. Since increasing lengths n of the generated sequence are considered, it is assumed that $s = s(n)$ may depend on n.

In order to formalize the notion of a statistical test of randomness, assume a function $T : \{0,1\}^n \to \{0,1\}$ computable in polynomial time and call it a test. The generator g passes the test T, if for every $c > 0$ and $n \to \infty$, we have

$$\left| \Pr_x[T(g(x)) = 1] - \Pr_y[T(y) = 1] \right| = O\left(\frac{1}{n^c}\right) ,$$

where x is chosen from the uniform distribution on $\{0,1\}^s$ and y is chosen from the uniform distribution on $\{0,1\}^n$.

The generator g is called a polynomially secure pseudorandom generator, if it passes every polynomial time computable test.

Acknowledgement. The author would like to thank to Martin Dietzfelbinger, Martin Holena, Jan Klaschka, Pavel Pudlak and Jiri Sgall for helpful comments concerning preliminary versions of the paper.

References

[1] O. Goldreich, Modern Cryptography, Probabilistic Proofs and Pseudorandomness. Springer-Verlag, Algorithms and Combinatorics, Vol 17, 1998.

[2] O. Goldreich, Lecture Notes on Pseudorandomness - Part I (polynomial-time generators), 2000. Online version available at
`http://www.wisdom.weizmann.ac.il/~oded/c-indist.html`

[3] P. Hellekalek, S. Wegenkittl, Empirical evidence concerning AES. ACM Transactions on Modeling and Computer Simulation (TOMACS), 13, 4 (2003), pp. 322–333.

[4] D. E. Knuth, The Art of Computer Programming, Volume 2: Seminumerical Algorithms, 3rd Edition, Addison-Wesley, Reading, Mass., 1998.

[5] P. L'Ecuyer, Good Parameter Sets for Combined Multiple Recursive Random Number Generators, Shorter version in Operations Research, 47, 1 (1999), 159–164.

[6] P. L'Ecuyer and J. Granger-Pich. Combined Generators with Components from Different Families, Mathematics and Computers in Simulation, 62 (2003), pp. 395–404.

[7] P. L'Ecuyer. Random Number Generation, Chapter 3 of Elsevier Handbooks in Operations Research and Management Science: Simulation, S. G. Henderson and B. L. Nelson, eds., Elsevier Science, to appear in 2006.

[8] M. Luby. Pseudorandomness and Cryptographic Applications. Princeton University Press, 1996.

[9] G. Marsaglia, A. Zaman, A new class of random number generators. The Annals of Applied Probability, 1, pp. 462–480.

[10] M. Matsumoto and T. Nishimura. Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. ACM Trans. on Modeling and Computer Simulation Vol. 8, No. 1, January pp.3-30 (1998).

[11] F. Panneton, P. L'Ecuyer, and M. Matsumoto, Improved Long-Period Generators Based on Linear Recurrences Modulo 2, ACM Transactions on Mathematical Software, 32, 1 (2006), pp. 1-16.

[12] The pLab project. Random Number Generators.
`http://random.mat.sbg.ac.at`

[13] Technical Solutions for Matlab, Solution Number: 1-10HYAS:
`http://www.mathworks.com/support/solutions/data/1-10HYAS.html?solution= 1-10HYAS`

[14] Technical Solutions for Matlab, Solution Number: 1-169M3:
`http://www.mathworks.com/support/solutions/data/1-169M3.html?solution= 1-169M3`

[15] Additional information to the current paper.
`http://www.cs.cas.cz/~savicky/prng`