Generating Pseudo-Random Numbers by Shuffling a Fibonacci Sequence

By Friedrich Gebhardt

1. Summary. M. D. MacLaren and G. Marsaglia [2] have proposed to mix two pseudo-random number generators in the following way: The first generator is used at the beginning to fill an array with pseudo-random numbers; whenever a random number is needed, the second generator determines which element of the array is to be used and replaced by a new number from the first generator. In this study, only one generator is utilized for both purposes; moreover, the generator chosen (a Fibonacci sequence) is by itself a rather poor one. Nevertheless, the final sequence of pseudo-random numbers passed all statistical tests applied to it, including χ^2-tests of the maximum and minimum of two to ten succeeding numbers and tests applied to sequences immediately following a small number or two almost equal ones.

2. Shuffling a Fibonacci Sequence. A Fibonacci sequence mod 2^k is defined by

$$a_{n+1} \equiv a_n + a_{n-1} \bmod 2^k .$$

For our purpose, one has to assure that at least one of the starting values, a_{-1} and a_0, is odd; otherwise any numbers a_{-1} and a_0 may be chosen, e. g., the sum of part of the computer storage. Under such a sequence, an array of 16 storage locations is initially filled with $a_1, a_2, \cdots, a_{16}$. Thereafter, whenever a random number is needed, two new Fibonacci numbers, a_m and a_{m+1}, say, are computed. Four digits of a_m determine one of the 16 storage locations. The number in that location is used as the random number, and a_{m+1} is stored at that location. This generator is reasonably fast (though not quite as fast as a congruential generator), it may need a few more storage locations than other generators, but it passed all statistical tests to be described now, while with congruential or mixed congruential generators serious dependencies have been observed (M. D. MacLaren and G. Marsaglia [2], M. Greenberger [1]).

The following χ^2-tests have been performed on 12,800 sequences of 10 pseudo-random numbers: uniform distribution of the first number in each sequence by dividing the unit interval into 16 and into 128 parts; joint distribution of the first two numbers by dividing the unit interval into 8 parts; joint distribution of the first three numbers by dividing the unit interval into 4 parts; distribution of the maximum, and of the minimum, of the first 2, 3, $\cdots$, 10 numbers by dividing the unit interval into 16 parts in such a way that each part has probability 1/16 under the null hypothesis of uniform distribution and independency of the original numbers.

All these tests have been repeated nine times. In the first three runs, successive sequences of 10 numbers each have been used. In the runs 4 to 6, pseudo-random

Received November 20, 1966. Revised February 20, 1967.

numbers have been generated until a number less than 0.1 has been found; the next ten numbers have been used for the tests. In the last three runs, two succeeding numbers differing by less than 0.01 have been required for using the following ten numbers. In none of these tests significant departures from the null hypothesis have been detected. Out of all 198 χ^2-values, 11 fall below the lower 5% level (4 of them below the lower 1% level) while 3 exceed the upper 5% level (none of them the upper 1% level). Table 1 gives for some of these tests the probabilities of exceeding the observed χ^2-values. Testing all 198 probabilities on uniform distribution between 0 and 1 by dividing the interval into 10 equal parts yields $\chi_9^2 = 13.2$ which is below the upper 10% level (14.7). A comparison of the three sequences of three runs each reveals no peculiarities.

These results agree with those of M. D. MacLaren and G. Marsaglia who used an array of 128 numbers and two congruential generators, one to determine a location in the array and the other to refill this place.

TABLE 1. *Probability in percent of a χ^2-variable exceeding the observed values. For different specifications of runs* 1–3, 4–6, *and* 7–9, *see text.*

	Run								
Test	1	2	3	4	5	6	7	8	9
Uniformity (128 intervals)	72	40	43	74	12	79	99.4	12	65
Pairs	7	81	35	11	96.6	27	81	73	75
Triples	45	32	9	71	92	69	6	50	1.2
Maximum of 2	32	27	54	34	97.8	77	95.4	61	60
Maximum of 5	87	77	86	23	47	3.9	18	99.1	60
Maximum of 8	77	77	18	41	75	24	14	99.87	90
Minimum of 3	91	75	23	31	60	45	92	67	14
Minimum of 10	20	64	72	78	88	77	18	55	24

Since the period of the pseudo-random number generator will be connected to the period of the Fibonacci sequence, (probably it will be much larger), the following theorem is of importance.

THEOREM. *Any Fibonacci sequence* $a_1, a_2, \cdots$, mod 2^k, *defined by* $a_{n+1} \equiv a_{n-1} + a_n \bmod 2^k$, *satisfying also* $a_1 \equiv 1 \bmod 2$ *has the primitive period* $3 \cdot 2^{k-1}$ (*i.e., there does not exist a shorter period*).

For a proof, see D. D. Wall [3].

3. Acknowledgement. The computations have been carried out on the IBM 7094/I computer of the Deutsches Rechenzentrum. Each run needed about 3 minutes.

Deutsches Rechenzentrum
6100 Darmstadt, Germany

1. M. GREENBERGER, "Method in randomness," *Comm. ACM*, v. 8, 1965, pp. 177–179.

2. M. D. MACLAREN & G. MARSAGLIA, "Uniform random number generators," *J. Assoc. Comput. Mach.*, v. 12, 1965, pp. 83–89. MR **30** #687.

→ D. D. WALL, "Fibonacci series modulo *m*," *Amer. Math. Monthly*, v. 67, 1960, pp. 525–532. MR 22 #10945.