Applications of Monte Carlo Methods to Finance and Insurance

Thomas N. Herzog, ASA, Ph.D.

Graham Lord, ASA, Ph.D.

ACTEX Publications, Inc.
Winsted, Connecticut

Cover design by Kathleen Borkowski

ISBN: 978-1-62542-908-7

Preface

Monte Carlo methods are useful in solving a wide range of problems, both stochastic and deterministic, which cannot easily be solved using analytic methods. The genesis of contemporary Monte Carlo methods is the Manhattan Project – the project that developed the atomic bomb in the United States. Most of the early researchers in this field were the physicists and mathematicians working for the Manhattan project.

The present work arose from the confluence of two streams of work.

The first had its genesis in graduate work at the University of Maryland, culminating in the first author's dissertation entitled "Classes of Infinite Binary Sequences and Their Set Relationships." This later led to (1) the short monograph "An Introduction to Stochastic Simulation," written for the Society of Actuaries, (2) research on imputation of missing values in sample surveys, completed under the auspices of the National Academy of Sciences, and (3) an award-winning paper on "Home Equity Conversion Mortgages." Dr. Herzog continues to be involved with applications of Monte Carlo methods to FHA-insured mortgages.

The second author, while at Morgan Stanley, made extensive use of Monte Carlo methods to perform the pricing of financial securities and derivatives and the valuation of insurance products, and to conduct asset-liability studies. He subsequently led a team, under contract to the U.S. Department of Energy, to analyze, by Monte Carlo techniques, the optimal draw-down of the Strategic Petroleum Reserve. He also used Monte Carlo techniques as part of the Andrew Mellon Foundation's study of graduate education in the United States. At Princeton University, Dr.

Lord teaches both undergraduate and graduate courses on Monte Carlo methods and their application to the valuation of financial instruments. Class notes from these courses form the backbone of the present work.

The authors assume that the reader has had at least a one-year undergraduate course in probability and statistics. Knowledge of some concepts of number theory would also be helpful in understanding a small amount of the material in Chapter 2. On the other hand, the reader with no background in number theory should not be at a major disadvantage. Although numerous references to the technical literature are provided, few are necessary for an understanding of the material discussed here. Rather, they are provided for those who would like to consult original sources, enhance their understanding of the topics discussed here, and/or obtain some insight into (1) related areas of interest (such as statistical tests of randomness) or (2) more advanced topics omitted from this introductory work. Quite a number of exercises are provided to help the reader reinforce his or her understanding of the material. Many of these have been taken from past examinations of the Society of Actuaries or the Casualty Actuarial Society, and some require a knowledge of basic life contingencies concepts and notation. Complete solutions to all of the text exercises are available in a companion solutions manual.

The book consists of thirteen chapters. The first chapter is an introduction to the rest of the book. Chapters 2 through 9 discuss the basic methodological approach. Chapter 2 presents a number of different schemes for generating uniform pseudo-random numbers. Chapters 3 and 4 describe schemes for generating pseudo-random numbers from other probability distributions. Chapter 5 describes a number of variance reduction schemes, and Chapter 6 discusses quasi-random numbers and concludes with an example of the simulation of a (Bayesian) predictive distribution. Chapters 7, 8 and 9 deal, respectively, with sample size determination, bootstrapping, and model validation techniques. The emphasis here is on the practical aspects of the techniques discussed. For the most part, lengthy mathematical proofs are omitted; however, if a short, elementary proof exists, then we have tended to include such proof in the text. We have also attempted to give the reader a feel for the historical development of this field of study.

Chapters 10 through 13 each describes a separate application of the Monte Carlo method to a practical problem in insurance and/or finance. In Chapter 10, we demonstrate the generation of future interest rate scenarios via a two-factor mean-reverting model of short-term interest rates. In Chapter 11, we describe a Monte Carlo model used to simulate the experience of a Home Equity Conversion Mortgage (HECM) insurance operation. We consider a number of Monte Carlo applications to Value at Risk issues in Chapter 12. Finally, in Chapter 13, we use simulation techniques to investigate the efficiency of the stock market in the United States.

In addition to the case studies, many of the applied exercises are presented within an insurance or financial setting.

The first author would like to thank his thesis advisor at the University of Maryland, Professor James C. Owings, Jr., for enhancing his understanding of the topic of "randomness." The second author would like to extend his appreciation to Dr. James A. Tilley for providing the stimulating environment at Morgan Stanley within which he honed his understanding of Monte Carlo methods. Many thanks as well to Noreen Goldman for her encouragement and support, and to our children, Ian and Michelle, for their understanding.

In addition to Professor Owings, the authors are indebted to the other members of an academic and practitioner review team for their suggestions that improved the exposition and notation of this work. This group included James G. Bridgeman, FSA, University of Connecticut, Charles Holland, Ph.D., and Walter B. Lowrie, FSA, University of Connecticut (retired).

Finally, the authors would like to extend thanks to Steven and Tracy Herzog for typing portions of an early draft of this work, and to the staff at ACTEX Publications. These include Marilyn J. Baleshiski, for turning our many rough drafts into a polished manuscript, to Kathleen H. Borkowski, for her graphic arts work on the text figures and cover, and to Richard L. (Dick) London, FSA, for his work as mathematics editor.

Thomas N. Herzog, Ph.D., A.S.A
Reston, Virginia

Graham Lord, Ph.D., A.S.A.
Princeton, New Jersey

September, 2002

Table of Contents

♦Chapter 5

Integral Estimation – Variance Reduction Techniques 95

♦Chapter 6

Quasi-Monte Carlo Methods 135

Chapter 1

Introduction

1.1 Overview

Monte Carlo methods are useful in solving a wide range of financial and insurance problems, both stochastic and deterministic, which cannot easily be solved using analytic methods. Monte Carlo methods are often referred to as *stochastic simulation* methods. The term "stochastic" is used to modify simulation in order to emphasize that we are confining our attention to simulation in which numbers are randomly selected from one or more probability distributions. The term "Monte Carlo" was coined during U.S. research work on the development of the hydrogen bomb in the years immediately following World War II. Monte Carlo methods were rarely performed prior to the advent of electronic computers. Because tremendous financial resources were expended on the Manhattan Project to develop the atom bomb and the ensuing work on the development of the hydrogen bomb, these projects were some of the first to have such computers. In fact, this explains why much of the early work on random number generators was performed by the scientists working on these two projects. Today the nearly universal availability of high-speed electronic computers makes Monte Carlo methods a cheap and effective method for solving a wide variety of complex, practical problems.

The actuarial applications of this technique include (1) model offices of life insurance and annuities, (2) analysis of investment and asset allocation strategies (e.g., bond call properties), (3) asset/liability management, (4) product design and pricing studies, (5) dynamic solvency testing of insurance company (or pension fund) solidity and resilience, (6) collective risk models in general, and (7) aggregate loss distributions in particular.

1.2 The Simulation Procedure

The crucial steps of a simulation are the following:

(1) The construction of an appropriate model
(2) The design of the experiment
(3) The repeated generation of *random numbers* from one (or more) probability distributions
(4) The analysis of the results

The focus of the first nine chapters of this work will be on the efficient generation of random numbers, also referred to as output values. The other steps, which are heavily dependent on the specific nature of the problem at hand, are illustrated in the detailed examples, which comprise Chapters 10-13 of this work. Since the generation of random numbers is crucial to any simulation, Chapters 2-6 contain a discussion of schemes (or algorithms) for the computer generation of random numbers from a number of frequently-used probability distributions. Such generation procedures were used in the past because they produce a large number of random numbers in a short period of time and do not require much computer storage space, as would a large table of random numbers permanently stored in the computer's memory.

In the remainder of this chapter, we discuss the historical development of the subject and then briefly describe some applications of the Monte Carlo method to finance and insurance.

1.3 Historical Development

The Monte Carlo method goes back at least as far as 1733 when Buffon [10] considered the following experiment that became known as "Buffon's needle problem." If a thin, straight needle of length b is tossed onto a two-dimensional surface (e.g., a table) having equally-spaced parallel lines a distance $d \geq b$ apart, and the needle lands entirely on the table, Buffon sought the probability (which is $\frac{2b}{\pi d}$) that the needle would not intersect any of the lines on the table.

In 1820, Laplace revived interest in Buffon's problem by suggesting that one could estimate the value of π by using the experimental results of a large number of such tosses in conjunction with results from probability theory.

According to Bennett [6], "The best-known early demonstration of a random sampling experiment was performed by William Sealy Gossett, a research chemist working for the Arthur Guinness Son & Company, Ltd. in Dublin, who was studying the relationship between the quality of Guinness beer and various factors in the beer's production."

Gossett, writing under the name of Student [68], described the Student's *t*-distribution that he used to study the distribution of means in small samples. Because his work at Guinness was highly confidential, he chose a totally unrelated experiment to illustrate his results. His experiment involved the height and left middle finger length for each of 3,000 criminals. This experiment was carried out by transcribing the data on each of these 3,000 individuals on a separate piece of cardboard, thoroughly shuffling the pieces of cardboard, and then drawing 750 samples of four observations each – a reasonably laborious process.

The first reported use of modern computing equipment to perform stochastic simulation was carried out by the scientists in the United States working on the development of the hydrogen bomb. According to Rhodes [60], "At the end of World War II, many of the scientists who had worked on the Manhattan Project left Los Alamos and took jobs at universities or research institutes. In particular, Stanislaw Ulam took an Associate Professorship at the University of Southern California. Shortly after his arrival there, he became seriously ill and spent some time on medical leave from the University. Resting at home during his extended recovery, Ulam amused himself playing solitaire. Sensitivity to patterns was part of his gift. He realized that he could estimate how a game would turn out if he laid down a few trial cards and then noted what proportion of his tries were successful, rather than attempting to work out all the possible combinations in his head. (Here Ulam was thinking about Canfield, or other versions of solitaire where the skill of the player is not important.) 'It occurred to me then,' he remembers, 'that this could be equally true of all processes involving branching of events.' Fission with its exponential spread of reactions was a branching process; so would the propagation of thermonuclear burning be. 'At each stage of the [fission] process, there are many possibilities determining the fate of the neutron. It can scatter at one angle, change its velocity, be absorbed, or produce more neutrons by a fission of the target nucleus, and so on.' Instead of trying to derive the expected outcomes of these processes with complex mathematics, Ulam saw that it should be possible to follow a few thousand individual *sample* particles, selecting a range for each particle's fate at each step of the way by throwing in a random number, and take the outcomes as an approximate answer – a useful result. This iterative process was something a computer could do."

In April of 1946, Ulam was invited to return to work at Los Alamos and subsequently told the eminent mathematician John von Neumann about his solitaire discovery. Ulam and von Neumann then "developed the mathematics together and named the procedure the Monte Carlo method " According to Ulam [70], "[I]t was named for Monte Carlo because of the element of chance, the production of random numbers with which to play the suitable games." Ulam stated that "the name Monte Carlo contributed very much to the popularization of this procedure." Von Neumann had access to the University of Pennsylvania's Electronic Numerical Integrator and Computer (ENIAC), a multipurpose electronic computer completed in 1945. According to Eves [19], "This was the first digital computer controlled by vacuum tubes. The machine required a 30 × 50 foot room, contained 19,000 vacuum tubes, and weighed 30 tons." Von Neumann used the ENIAC to generate random numbers, thereby successfully achieving the first computer application of the Monte Carlo method.

1.4 Examples of the Use of the Monte Carlo Method

There are at least two general types of problems amenable to solution by stochastic simulation. The first are *deterministic* problems that are difficult to solve directly. According to Hammersley and Handscomb [26], "The possibility of applying Monte Carlo methods to deterministic problems was noticed by Fermi, von Neumann, and Ulam, and popularized by them in the immediate post-war years. About 1948, Fermi, Metropolis, and Ulam obtained Monte Carlo estimates for the eigenvalues of the Schrodinger equation."

Another application of the Monte Carlo method to a deterministic problem involves the evaluation of complicated integrals. The method of solution is illustrated in Figure 1.1 below. To find the area under an irregular curve, $f(x)$, we surround the area with a rectangle and instruct the computer to generate a large number of points within the rectangle. We then count the number of points lying under the curve and divide it by the total number of random points generated. This gives us the proportion of points lying in the region of interest. We obtain our desired result by multiplying this proportion by the area of the rectangle. To increase the reliability of our results, we generate a larger number of random points.

The second type of problem is *statistical* in nature and involves a number of random variables that are correlated with one another. These are illustrated in the problems described in the remainder of this section.

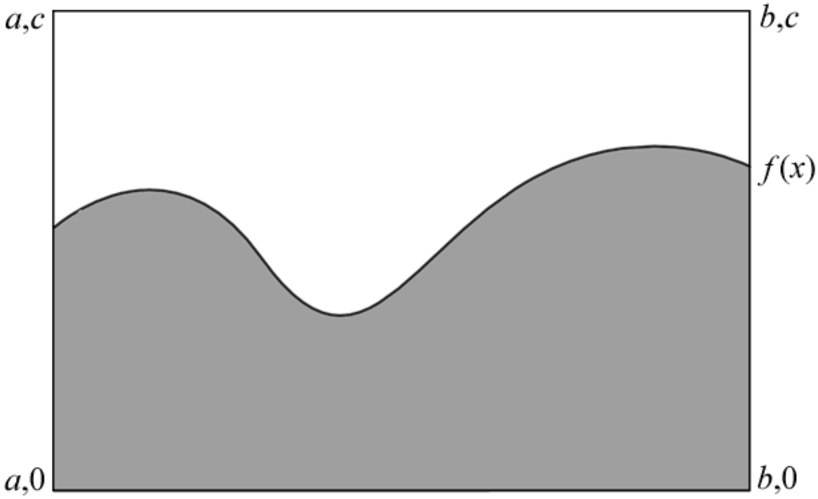

FIGURE 1.1

1.4.1 Estimating Mortgage Prepayment Rates

In the United States, the outstanding balance on single-family mortgages is several *trillion* dollars. The mortgage companies originating such mortgages frequently sell the loans to one of three large entities: the

Government National Mortgage Association (Ginnie Mae), the Federal National Mortgage Association (Fannie Mae), or the Federal Home Loan Mortgage Corporation (Freddie Mac). Ginnie Mae, Fannie Mae, and Freddie Mac then group similar loans together and sell them to investors in the form of so-called *mortgage-backed securities*. In order for investors to price these mortgage-backed securities accurately, investors need to know when the loans constituting the mortgage-backed securities will terminate – either by prepayment or by attainment of the maturity date of the mortgage. This is highly dependent on the course of future interest rates – which is most difficult to predict. One naïve approach to predicting future interest rates is to just assume they will be constant from the date of origination until the date the loan is scheduled to mature. This is an unsatisfactory approach because if mortgage interest rates were to be constant for the entire term of the loan, prepayment rates would be low – well below historical averages. In Chapter 10 we describe a simple, yet effective, procedure for generating a large number of interest rate scenarios.

One way around this is to use stochastic simulation or other techniques to generate a large number of interest rate scenarios over the future lives of such mortgages. This usually results in a more realistic approach – one that produces an entire probability distribution of prepayment rates.

In the mortgage prepayment example just presented, we implicitly assumed a single-decrement situation – i.e., we assumed that all of the mortgage terminations were prepayments. However, if we look at the situation from the perspective of a mortgage guarantee insurance company (which is insuring the mortgages against the risk that the mortgage may be foreclosed on), then we have a double-decrement problem in which the type of termination – foreclosure or prepayment – is usually important. The mortgage guarantee insurance company needs to (1) determine a premium to charge for mortgage guarantee insurance on mortgages insured in the future and (2) estimate the future liabilities of books of business currently in force. As above, one approach to these problems is to use stochastic simulation or other techniques to generate a large number of interest rate scenarios over the future lives of such mortgages. This would, for instance, produce an entire probability distribution for the estimated future liabilities of the mortgage guarantee insurance company.

1.4.2 Reverse Mortgages – A Model Office Approach

A number of years ago, the Chief Actuary of a mortgage insurance company was faced with the problem of predicting the future experience of *reverse mortgages*, a new product which the company had decided to insure. Reverse mortgages allow the elderly homeowner who is house rich but cash poor to borrow money against the equity in her house while continuing to reside in the house until she either decides to sell the house or dies. To better understand the issues, the actuary constructed two two-stage simulation models, one of the appreciation of the price of the house and the other of the continued occupancy of the house. The first stage of the house price appreciation model was a multivariate normal distribution of national house price appreciation rates. The second stage was a model of the annual appreciation rates of 1,000 individual houses given the rate of national appreciation for that year (i.e., the result of the first stage of the model). In the occupancy (or mortality) model, the actuary first simulated the annual mortality rates of female lives beginning with q_{65} in 1990. More generally, estimated values were produced for q_{65+x} In year 1990+x, for $x=1,2,\ldots,44$. Finally, given the simulated q_x values, the experience of the 1,000 insureds was then simulated. Thus, four separate models (two for house price appreciation rates and two for mortality rates) were effectively simulated in order to improve the understanding of the issues involved with reverse mortgages.

This application of the Monte Carlo method is described in more detail in Chapter 11.

1.4.3 Imputation of Missing Data Elements

A life insurance company has a cohort of insured lives consisting of 900 policyholders. The medical records of 840 of these insureds are complete, but the other 60 are, for whatever reason, missing information on the most recent cholesterol test. The results of the cholesterol tests are partitioned into three classes: low, medium, or high. The probability of death during the ensuing 12 months for each policyholder is a function of his or her cholesterol test result, as follows:

Low: 1%
Medium: 2%
High: 5%

Prior to seeing any test results, the chief medical officer of the insurance company feels that one-third of the insureds will fall into each of the three classes. How might we estimate the probability distribution of the frequency of insurance claims from these 900 policies over the next 12 months?

Perhaps the simplest approach is to assign all 60 insureds a score of “medium.” Unfortunately, this is unsatisfactory because it does not even correctly estimate the mean result.

A better approach, perhaps, is to assign 20 insureds to each of the three classes. Although this produces a correct estimate of the (prior) mean of the 60 insureds lacking test results, it leads to an underestimate of the variance of the test results of all 900 policyholders. Essentially this approach assumes that we have valid cholesterol test scores on 900 policyholders when, in fact, we only have 840.

A preferred approach is to use stochastic simulation to impute at least two values for each missing score. This is the methodology first suggested by Rubin [64]. This approach allows the actuary to get reasonable estimates of both the mean and the variance of the distribution without extensive effort.

1.4.4 Other Insurance Products

Stochastic simulation is also highly useful for modeling natural catastrophes such as severe earthquakes or hurricanes, both of which have low frequency of occurrence but high severity of loss. These are of particular interest to reinsurance companies.

Chapter 2

Generating Random Digits and Uniform Random Numbers over [0,1)

This chapter is intended to introduce the reader to various techniques related to the generation of random numbers. It is not meant to be a comprehensive treatment of the subject, nor is it our aim to compete with Knuth [38] who devotes almost 200 pages to the topic of randomness. (Note, however, that an alternative perspective is provided in Chapter 6 of this text.) A highly readable, non-mathematical introduction to the topic is found in Chapter 8 of Bennett [6]. As research continues on this fascinating topic, there is not yet a definitive body of work on this subject. Thus, our aim is simply to make our readers aware of some of the basic issues and to point them in the right direction to the maximum extent possible. In particular, we have almost no discussion of statistical tests of randomness in this book. Again, we refer the interested reader to Chapter 3 of Knuth. For our purposes, random sequences are those without any discernible pattern, or, more precisely, those whose discernible patterns arise and then disappear without any discernible pattern.

In this chapter, we discuss a number of algorithms used to generate sequences of random numbers on a computer. Once such an algorithm has been completely specified, the exact sequence of numbers to be produced is determined. Because such sequences of numbers are produced in a deterministic fashion, they are usually called *pseudo-random* or *quasi-random sequences* in the technical literature. Here, however, we plan to call them simply random sequences and to avoid a philosophical discussion of the concept of randomness.

One of the advantages of such an algorithm is that it facilitates the reproduction of results among researchers, regulators, legal advocates, and/or other interested parties. Some of the desirable properties of such a random generator are the following:

(1) It should be easy to program on a computer.
(2) It should produce a long sequence of numbers before it begins to repeat or recycle the numbers.
(3) It should have acceptable statistical properties for a wide-range of possible uses.

Unfortunately, it is not easy to construct such an ideal random number generator. We omit from this list the goal that the random number generator consume only a small amount of the computer's memory. With recent technological advances this should not be the problem it was in the past.

When sequences of random numbers are used in conjunction with a stochastic model, the sequences may interact with the model in a way that distorts the results – whether means, variances, or correlations. Moreover, such distortions may only become apparent after a careful in-depth analysis of the results of the simulation.

We begin this section with a discussion of the generation of a sequence of random numbers, U_n, from a uniform distribution over the unit interval from 0 to 1. We study the generation of uniform random numbers first because, as described later in this work, we can use such random numbers together with mathematical transformations to generate random numbers from other probability distributions. Since a computer can represent a real number with only finite accuracy, we actually generate integers, X_n, between zero and a positive integer m.

Let U be a random variable with

$$Pr[0 \le U < 1] = 1.$$

Let $U(k)$ denote the k^{th} term in the decimal representation of U. That is,

$$U(k) = \left\lfloor 10^k \cdot U \mod 10 \right\rfloor,$$

where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. Then U is uniformly distributed over [0,1) if and only if

$$Pr[U(k) = j] \quad = \quad \frac{1}{10},$$

for $j = 0,1,2,\ldots,9$, and $k = 0,1,2,\ldots$. This result is established on pages 29-31 of Yakowitz [74].

Thus we have established that random variables which are uniformly distributed over $[0,1)$ can be easily transformed into discrete random variables over the digits 0 through 9, and we can transform random digits into uniform random numbers having as many decimal places as we desire.

2.1 Random Numbers Before Modern Computers

Before the advent of modern electronic computers, researchers employed various creative schemes to obtain random numbers for their research studies and experiments. A table of 41,600 random digits, "taken at random from census reports," was published in 1927 by L.H.C. Tippett. A mere ten years after its publication, Tipett's table was deemed inadequate for very large sampling experiments. In 1938, R.A. Fisher and F. Yates published 15,000 additional random digits, selected from the 15^{th} through 19^{th} decimal places of logarithmic expansions. The digits were obtained through a procedure involving two decks of playing cards.

Since then, a number of devices have been built to generate random numbers mechanically. The first such machine was used in 1939 by M.G. Kendall and B. Babington-Smith to produce a table of 100,000 random digits. Here, the digits were generated randomly by a machine constructed from a rotating disk. The disk was partitioned into ten sectors. As the disk rotated, one of the ten sectors was momentarily illuminated by a flashing neon light.

The Ferranti Mark I computer, first installed in 1951, had a built-in instruction that put 20 random bits into the accumulator using a resistance noise generator; this feature had been suggested by Alan Turing.

In 1955, the Rand Corporation published a widely-used book entitled *A Million Random Numbers and 100,000 Normal Deviates*. The random digits were obtained by re-randomizing a table of digits generated by the random frequency pulses of an electronic roulette wheel.

As larger and larger tables of random numbers were needed to solve both deterministic and probabilistic problems, the storage of random numbers within the early computers began to consume far too much precious memory. Many deemed the solution to be a formula that allowed the computer to generate a random number using the computer's ordinary arithmetical operations at the instant such a number was needed in the computation process.

Amazingly, things came full circle for tables of random numbers 40 years after the Rand Corporation's publication date. As reported in Knuth [38], "Advances in technology made tables [of random numbers] useful again during the 1990s, because a billion well-tested random bytes could be distributed on CDROM. George Marsaglia helped resuscitate tables in 1995 by preparing a demonstration disk that contained 650 random megabytes, generated by combining the output of a noise-diode circuit with deterministically scrambled rock music. (He called it 'white and black noise.')"

2.2 von Neumann's Middle Square Method

In 1951, von Neumann proposed one of the first schemes for generating random numbers on an electronic computer. This arithmetical procedure, called the *middle-square method*, generated random numbers $x_0, x_1, \ldots$, each composed of n or $n+1$ digits. This algorithm starts with a number, x_0, n-digits long, squares it, and then sets x_1 to be the middle n or $n+1$ digits of x_0^2. The algorithm continues on with x_2 being the middle n or $n+1$ digits of x_1^2, and so on. For example, if we choose $x_0 = 157$, a number having three digits, then $x_1 = 464$ because the middle three digits of $(157)^2 = 24{,}649$ are 464. Similarly, because $(464)^2 = 215{,}296$, we obtain $x_2 = 1{,}529$.

Unfortunately, this method has been found to be a poor source of random numbers, flawed by imprudent choices of the starting value x_0. For example, starting with the 4-digit number 3,792 and squaring it, we obtain the number 14,379,264 with the middle four digits being 3,792, the same 4-digit number with which we started.

2.3 Multiplicative Congruential Random Number Generators

Today a frequently-used type of random number generator is known as a *linear congruential generator*, which was introduced by Lehmer [43]. In order to fully specify an individual linear congruential generator we must select the following four integer-valued parameters:

Parameter Name	Symbol	Restrictions
The modulus	m	$m > 0$
The multiplier	a	$0 < a < m$
The increment	c	$0 \le c < m$
The starting value	X_0	$0 < X_0 < m$

The $(n+1)^{st}$ term of the random sequence specified is

$$X_{n+1} \equiv a \cdot X_n + c \mod m.$$

In other words, X_{n+1} is the remainder when $(a \cdot X_n + c)$ is divided by m, so the possible values of X_n are $0, 1, \ldots, m-1$.

2.3.1 Linear Congruential Random Number Generators

To illustrate the use of a linear congruential generator, we set $m = 8$ (so there are eight distinct possible values), $a = 5$, $c = 1$, and $X_0 = 3$. The resulting sequence of numbers is given in the following table.

TABLE 2.1

Values of the Linear Congruential Random Number Generator $X_{n+1} \equiv 5 \cdot X_n + 1 \mod 8$				
n	X_n	$5 \cdot X_n + 1$	$\frac{5 \cdot X_n + 1}{8}$	X_{n+1}
0	3	16	2 + (0/8)	0
1	0	1	(1/8)	1
2	1	6	(6/8)	6
3	6	31	3 + (7/8)	7
4	7	36	4 + (4/8)	4
5	4	21	2 + (5/8)	5
6	5	26	3 + (2/8)	2
7	2	11	1 + (3/8)	3

There is no point in displaying any additional terms of the sequence because at $n = 8$, the sequence begins repeating the numbers in the same order. Algebraically, this means that $X_{n+8} = X_n$ for all non-negative integers n. The portion of the sequence displayed above includes each of the eight possible values exactly once.

The sequence obtained when $m = 10$ and $X_0 = a = c = 7$ is

$$7,6,9,0,7,6,9,0,\ldots .$$

This sequence only makes use of four of the ten possible values before it gets into a loop, or, more formally, begins to *cycle*. These two sequences illustrate the following common property of linear congruential sequences: there is ultimately a cycle of numbers that is endlessly repeated. The appendix to this chapter describes methods for choosing a, m, c, and X_0 so that the maximum cycle length, m, is obtained. For more details, see Knuth [38].

2.3.2 Multiplicative Congruential Random Number Generators

A *multiplicative congruential generator* is the special case of a linear congruential generator which is obtained when $c = 0$. Because the generation process is a little faster when $c = 0$, and most other desirable features are preserved, many practitioners prefer to use multiplicative congruential generators.

2.3.2.1 RANDU

The linear congruential random number generator given by

$$X_{n+1} \equiv 65{,}539 \cdot X_n \mod 2^{31},$$

where X_0 is odd, is known as RANDU. It was developed by IBM and was commonly used on most of the world's computers during the late 1960s. Because RANDU can be rewritten as

$$X_{n+2} \equiv 6 \cdot X_{n+1} - 9 \cdot X_n \mod 2^{31},$$

it fails most three-dimensional criteria for randomness as discussed in Section 2.3.2.3 below. In our opinion, it should never have been used.

2.3.2.2 GGL

IBM eventually replaced RANDU by a random number generator known as GGL, developed by Lewis, Goodman, and Miller [44]. It is a multiplicative congruential generator with values $a = X_0 = 16807 = 7^5$ and $m = 2^{31} - 1$. GGL has a cycle length of $2^{31} - 2 \approx 2$ billion; this is the maximum possible length because, if X_n is ever zero, then all subsequent terms must be zero. Prior to its implementation, this generator successfully passed a wide range of statistical tests as described in Lewis, Goodman, and Miller. GGL is still the random number generator employed as the "?" operator in IBM's version of the APL computer programming language. This generator still works well for many problems as noted on page 189 of Knuth [38].

2.3.2.3 Comparing RANDU and GGL

To test the relative effectiveness of RANDU and GGL, we generate 100,000 random numbers for each generator over the range of 0 to $m-1$. In both cases, we choose 1 as our *seed* (i.e., $X_0 = 1$). Then for $j = 0, \ldots, 99{,}999,$ we define

$$Y_j = \left\lfloor 10 \cdot \frac{X_j}{m} \right\rfloor,$$

where $\lfloor x \rfloor$ denotes the largest integer less than or equal to *x*, and tabulate the frequency of each possible value of Y_j as well as of the triples (Y_j, Y_{j+1}, Y_{j+2}). We summarize the results in Tables 2.2 and 2.3 below.

For the single digits, Y_j, both generators perform well based on the value of their chi-square goodness-of-fit test statistic. However, for the triples generated by RANDU, the value of the chi-square goodness-of-fit test statistic is 1638.3, more than 12 standard deviations above its mean. By contrast, that of GGL is 1021.7, only about half a standard deviation above its mean. As mentioned above, this is an example of a discernible pattern persisting in a discernible way, a fatal weakness of RANDU.

TABLE 2.2

Frequency Count of Selected Digits for Each Random Number Generator

Digit	Random Number Generator	
	RANDU	**GGL**
0	9,862	10,047
1	10,097	10,016
2	9,790	9,863
3	9,900	9,878
4	10,025	10,012
5	10,053	10,285
6	10,094	9,931
7	10,051	9,955
8	9,992	10,118
9	10,136	9,895
Value of Test Statistic	11.6	14.9

TABLE 2.3

Frequency Count of Selected Triples			
Observation	**Frequency Count under RANDU**	**Observation**	**Frequency Count under GGL**
(1,5,4)	143	(3,7,8)	128
(6,0,9)	141	(4,5,0)	128
(5,5,7)	140	(5,6,5)	128
(3,6,9)	139	(1,4,4)	127
(9,9,8)	136	(0,1,8)	126
(2,4,5)	63	(2,9,9)	63
(3,6,3)	71	(0,3,9)	71
(8,9,2)	72	(6,4,9)	72
(0,2,7)	75	(0,4,3)	75
(9,0,4)	76	(3,0,3)	76
Value of Test Statistic	1638.3	Value of Test Statistic	1021.7

2.3.2.4 A Deficiency of Multiplicative Congruential Generators

Marsaglia [46] has identified a serious defect in multiplicative congruential random number generators. This is perhaps best understood by considering the following example constructed by Hoaglin [31].

We consider the generator

$$X_{n+1} \equiv 6 \cdot X_n \mod 17,$$

which, if $X_0 = 1$, produces the sequence

1,6,2,12,4,7,8,14,16,11,15,5,13,10,9,3,… .

If we now plot the overlapping ordered pairs produced by this generator, namely, (1,6), (6,2), (2,12), …, in two-space, we obtain the graph shown in Figure 2.1.