Expressing the randomity of events – An analysis of random number generation with given distributions

Carl Zhou

Abstract In cases where it is necessary to generate random numbers that obey specific distributions, some of those distributions can be expressed as mathematical functions while others cannot. This is especially the case for epidemiological, medical, and pharmaceutical investigations, where more accurate methods, utilising actual distribution (from survey and experimental data) to generate random numbers may be required. In this study, three methods are analyzed to demonstrate simple computation examples. These methods include: inverse transform, acceptance-rejection, and Monte-Carlo simulations. Their applications are explored from a data analysis point of view. Additionally, this article discusses a flexible and practical approach of statistical measures optimization, which approximates the solution by fitting the statistical measures.

Keywords: Pseudo-random; Random number generator; Specified distribution; Monte Carlo simulation

Résumé Dans les cas où il est nécessaire de générer des nombres aléatoires qui obéissent à des distributions spécifiques, certaines distributions peuvent être exprimées sous forme de fonctions mathématiques, alors que d'autres ne peuvent pas l'être. Ceci est le cas pour les enquêtes épidémiologiques, médicales et pharmaceutiques, où on exige parfois des méthodes plus précises qui utilisent la distribution actuelle (à partir des données expérimentales et d'enquêtes) afin de générer des nombres aléatoires. Cette étude explore trois méthodes afin de démontrer des exemples de calculs simples. Ces méthodes comprennent la transformée inverse, la méthode du rejet et les simulations de Monte-Carlo. Leurs applications sont explorées à partir des analyses de base de données. En outre, cet article traite d'une approche flexible et pratique l'optimisation des mesures statistiques, qui estime la solution en ajustant les mesures statistiques.

Mots Clés: Pseudo-aléatoire; Générateur de nombres aléatoires; Distribution spécifiée; Simulation Monte-Carlo

1 Introduction

Random numbers, or stochastic numbers, are often necessary for experimentation, that is, to describe stochasticity in studies by generating random numbers instead of performing mass data collection. Generally, random numbers have three uses: (i) Sampling. e.g. sampling a population representatively, or randomly distributing laboratory animals into experimental groups, for example; (ii) Optimizing model parameters. e.g. Monte-Carlo method (*1*); (iii) Simulating population or events. e.g. simulating the initial genetic status of a population, in which the selection, genetic cross, and mutations are all random. It can also be applied to queuing and resource storage scenarios, which often require cost estimations by performing simulations based on generated random numbers.

Correspondence: czhou027@uottawa.ca
Interdisciplinary School of Health Sciences, University of Ottawa, 25 University Private, K1N 7K4, Ottawa, Canada

Since the 1950s, a number of studies have presented methodologies of generations based on computation mathematics and related technology. Because a mass of random numbers is only made by computers, discussions of the methodology can be reviewed in two stages. In the computer's early years, Marsaglia, Tausworthe, Fishman, and Lewis systematically presented the theoretical analyses, which indicated the technical feasibility and laid the theoretical foundations for generating random numbers in the 1960s and 1970s (*2–5*). Since then, the use of statistical simulations has accelerated due to the development of computer technology and its wide applications (*6*). Many researchers discussed and developed realistic approaches for various experiments (e.g. Rubenstein, Devroye, Ripley, Dagpunar, Law, and Kelton) (*7–11*). Most studies introduce approaches in detail for generating random numbers on several popular distributions (e.g. Aiello et al., Deng and Lin, Ferguson et al., L'Ecuyer, Luby, Marsaglia and Zaman, Niederreiter

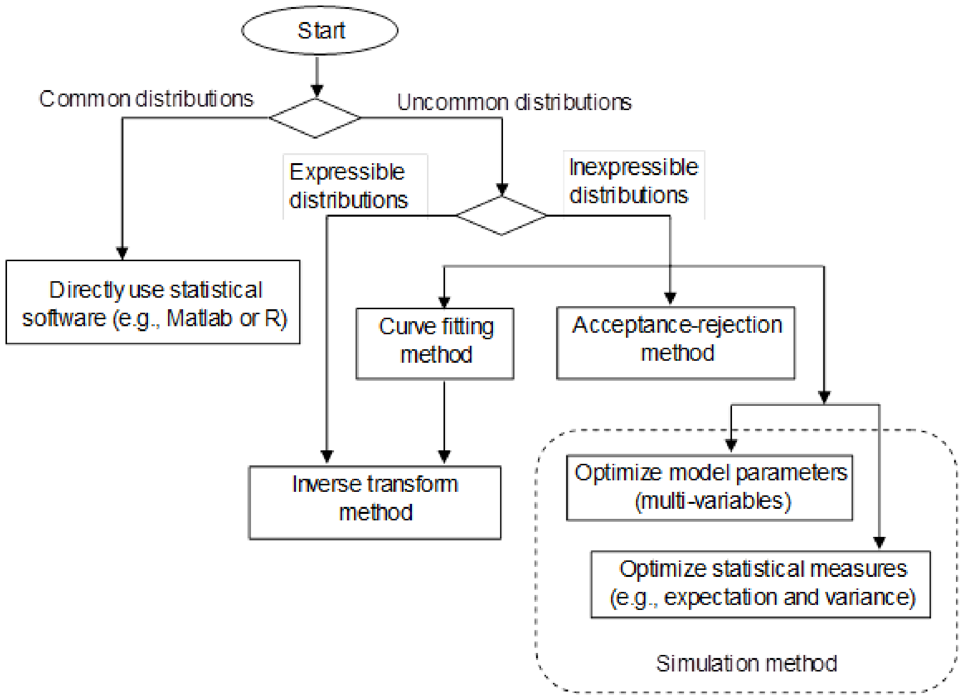
Start
Common distributions
Uncommon distributions
Expressible distributions
Inexpressible distributions
Directly use statistical software (e.g., Matlab or R)
Curve fitting method
Acceptance-rejection method
Inverse transform method
Optimize model parameters (multi-variables)
Optimize statistical measures (e.g., expectation and variance)
Simulation method

Figure 1: Method selection flow chart for constructing random number generators that obey different distributions. Diamonds denote logical judgement, rectangles indicate methods or processes, dotted boxes mean type of method and arrow lines illustrate the selection of methods.

and Shparlinski, Ross) (*12–19*). To describe various practical problems, however, not all distributions can be easily expressed by mathematical analytic formulas. When performing data analysis, there is the frequently asked question of how to generate random numbers with the actual distributions. Especially for epidemiological, medical, and pharmaceutical experiments, the need for precision might require the use of actual data distributions collected from surveys, investigations, and experiments to generate random numbers.

With regards to the aforementioned needs for realising stochasticity, an analysis of different approaches to generate random numbers is presented. These approaches include inverse transform, acceptance-rejection, and Monte-Carlo simulations. As a powerful tool for the application of simulation methods, the sampling technique is introduced and illustrated for solving the problems that have no analytical solutions. In other words, this article derivates algorithms, which generate random numbers based on several different calculations (Figure 1). The study objectives are to: (1) provide an overview of three principal approaches for generating random numbers that describe the stochasticity of events based on practical observations, and (2) analyze the random number generator based on inexpressible distributions and its accuracy and efficiency for applications.

2 Existing Distributions Provided by Statistical Software

Theoretically, random numbers are generated through physical phenomena, such as coin tossing, dice rolling, roulette, etc. These random numbers are called true random numbers and they are relatively difficult to create via computer. However, in practical application, the use of pseudo-random numbers is often sufficient. These number series appear to be random numbers, but they are actually generated through a repeatable computation using the computer (*20–22*). Utilizing pseudo-random numbers with uniform distribution, many statistical software packages can generate random numbers subjected to various distributions. For example, Matlab can generate random numbers subjected to the distributions of normal (randn), Beta (betarnd), binomial (binornd), chi-square (chi2rnd), exponential (exprnd), F (frnd), and numerous other function forms. If a desired distribution cannot be constructed using a statistical software, further calculations or transformations are required to generate these random numbers. The following sections summarize some approaches that can be used. These methods all use the most basic pseudo-random number generator.

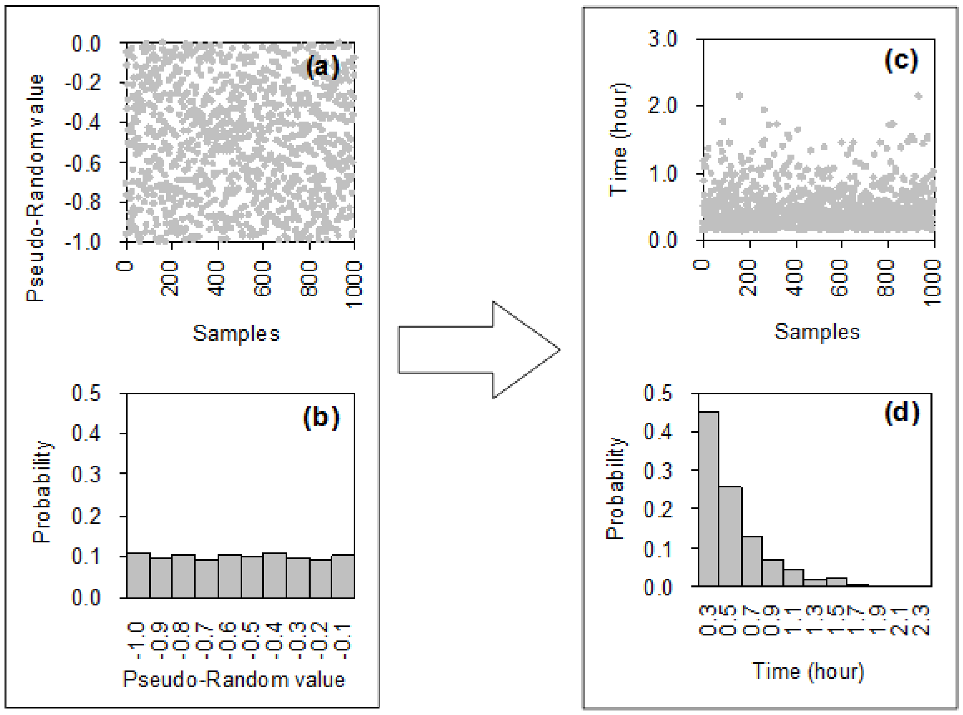

Figure 2: Diagram of the transformation from uniform distribution x to exponential distribution y. (a) and (b) illustrate all 1000 sample values x generated by uniform distribution; (c) and (d) show the converted random data complying with y.

3 Random Generator Based on Expressible Distributions

In some cases, the required distribution can be expressed with a formula. This allows for easy mathematical processing. A commonly used approach is called the inverse transform method, which includes two steps that are briefly outlined as follows.

Step 1: Assume $X[0,1]$ obeys uniform distribution $u(x)$ and its cumulative probability distribution function $U(x)$. $Y[ay,y]$ obeys the desired distribution $h(y)$ and its cumulative probability distribution function $H(y)$. Then we solve $U(x)$ and $H(y)$, and let $U(x) = H(y)$, i.e. $\int_{ax}^{x} u(t)dt = \int_{ay}^{y} h(t)dt$, where x and y are two random variables, and ax and ay are initial values. Note that there is a one-to-one correspondence between x and y. Because the left side of the equation is equal to x, we obtain $x = H(y)$.

Step 2: Solve the inverse function of x to y based on $U(x) = H(y)$. The inverse function is $y = H - 1(x)$. Therefore, we can generate random number y using x.

The following is a calculation example. Survey data suggests that three newborns are delivered at a hospital per hour on average. In order to simulate (not only to calculate) the number of newborns delivered in the next hour, a set of dummy data is required to represent the delivery of a newborn after time T. Let us assume that the dummy data meets the exponential distribution

$$e^{-kt} \text{ or } ra^{-kt} \tag{1}$$

Here we will use the latter to demonstrate the approach (let the parameters be $r = 2.0$, $a = 2.7$, $k = 3.0$).

According to the example scenario, set: Uniform random number x's probability density function is

$$p(x) = \begin{cases} 1, & if 0, \leq x \leq 1 \\ 0, & otherwise. \end{cases} \tag{2}$$

Its cumulative probability is $U(x) = \int_0^x p(x)dx$. The desired probability density function is

$$p(y) = \begin{cases} ra^{-ky}, & y \geq 0 \\ 0, & y < 0 \end{cases} \tag{3}$$

where y denotes the time in hours when no babies are born. Its cumulative probability is $H(y) = \int_0^y p(y)dy$. Since $p(x)$ and $p(y)$'s cumulative probabilities are equal, we have

$$\int_0^x dx = \int_0^y ra^{-ky}dy \tag{4}$$

Table 1: Observation of a doctor's time spent with patients.

Time (minute)	Number of Patients	Probability
10	26	0.250
11	24	0.231
12	20	0.192
13	17	0.164
14	10	0.096
15	6	0.058
16	1	0.010
Total	104	1.0

The number of patients (observed data) is 104 ($n = 104$).

Solving it, we get $x = -(ra^{-ky})/(k \log a)$. Its inverse function can then be derived as

$$y = -[\log(-kx) + \log(\log a) - \log r]/(k \log a)(1) \quad (5)$$

This is the random number generator that meets the distribution (ra^{-kt}). Figure 2 displays the transformation from uniform distribution x to exponential distribution y. As demonstrated for this simulation, when T is 0.5 hours, the probability of no newborns delivered is 0.273. In other words, the probability of at least one newborn delivered is 0.727 in this time.

4 Random Generator Based on Inexpressible Distributions

4.1 Observation-based distributions

4.1.1 Curve fitting

There will often be some need for curve fitting in many practical applications. We hope to generate random numbers according to the probability distribution of the actual sample. These distributions derived from observed samples vary greatly, and they often cannot be expressed using existing function forms. In these cases, a suitable function can be chosen, such as a polynomial function, to fit the actual distribution. Then, we can use the inverse transform method to derive the distribution curve. It is important to note that since the inverse function of second- and higher-order polynomials is very complex, it may be difficult to find a solution. Therefore, solving these inverse functions require optimization by computer. In the following paragraph, the example of queuing for a walk-in clinic will be used to demonstrate how to solve such problems.

Example: According to survey data, the distribution representing a doctor's time spent with patients is neither Poisson nor negative exponential distribution, as shown in Table 1 and Figure 3. Time cost calculation depends on this distribution; we need to generate random numbers to simulate the possible visitation times of a number of patients (e.g. 300 patients). The derivation of the distribution is presented as follows by fitting the curve and solving its inverse function.

Let uniform random number x's probability density function be

$$p(x) = \begin{cases} 1, & 0 < x \leq 1 \\ 0, & otherwise \end{cases} \quad (6)$$

Its cumulative probability is $U(x) = \int_0^\infty p(x)dx$. The observation data probability function is

$$p(y) = -0.00263y^2 + 0.02690y + 0.24863 \ (10 \leq y \leq 16) \quad (7)$$

where y denotes doctor's time for each patient (Figure 3). Its cumulative probability is

$$H(y) = \int_0^y p(y)dy \quad (8)$$

where $H(y)$ is equal to 1. Since $\int_0^\infty dx$ equals 1, let $\int_0^\infty dx = \int_0^y p(y)dy$. We have

$$x = -0.000876667y^3 + 0.01345y^2 + 0.24863y = 2.95463 \quad (9)$$

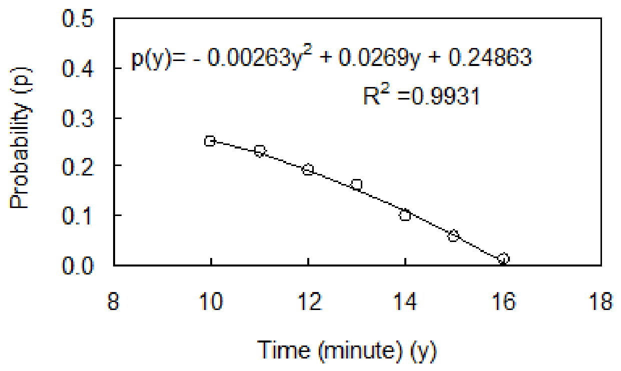

Figure 3: The distribution representing a doctor's time spent with patients based on observed data. The circles denote the time samples ($n = 7$), and the solid line displays the fitting curve based on the regression function $p(y)$.

The inverse function of $x = f(y)$ is hard to express in analytical form. We can solve it through programmed optimization algorithms. The idea is to first generate 300 random numbers x ($0 \leq x \leq 1$), then find a y for every x that is $10 \leq y \leq 16$, until the precision of $x - x \leq 0.001$ (or $x - x = minimum$) is satisfied, where x is the x that

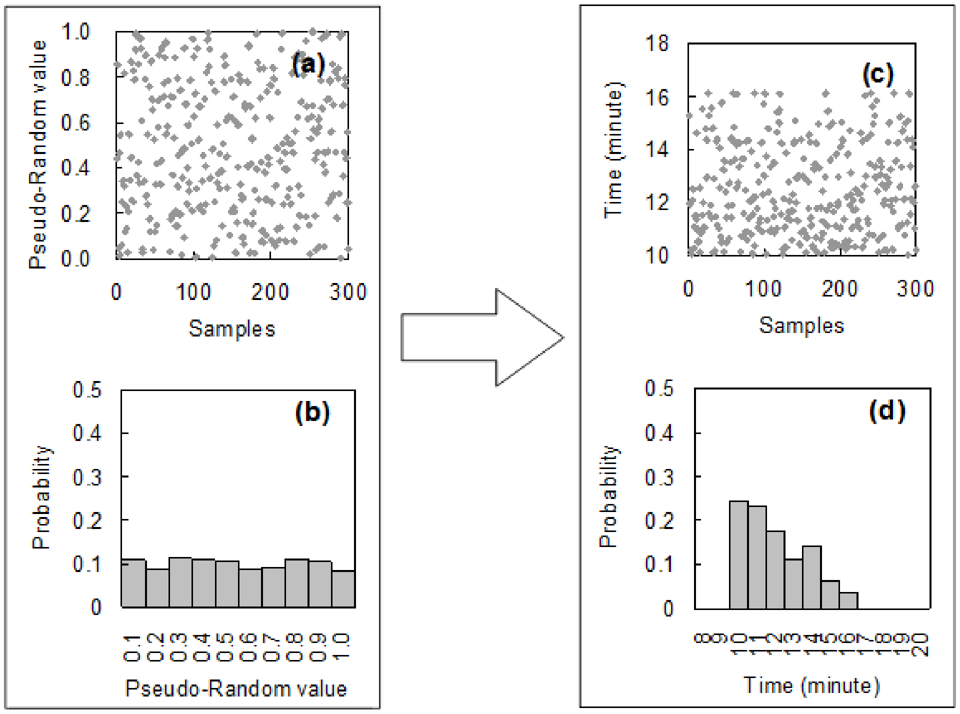

Figure 4: An example of random number generation under a specified probability density function. (a) and (b) illustrate all 300 sample values generated by uniform distribution; (c) and (d) denote the converted random data complying with the specified distribution shown in Figure 3.

has met the accuracy of the polynomial. After 300 x's have been fitted, we obtain all y values corresponding to x (Figure 4).

The solution presented above shows that the resultant random numbers distribution is close to the survey data distribution (Figure 4). Clearly, good curve-fitting increases the accuracy of the distribution, while poor fitting will not yield a satisfactory estimate. If higher precision is required to match the two distributions (survey data and random numbers), the following method (the acceptance-rejection approach) is a very practical option.

4.1.2 The acceptance-rejection technique

The acceptance-rejection technique is simple to understand. It can be expressed visually by a two-dimensional area composed of x and y covered with randomly distributed dots. If the envelope of a given distribution curve or probability density function is placed in this area, and the dots above the line are rejected while the remaining dots are accepted, then they will become the random numbers we need (Figure 5). Ross suggested a judgement approach, which may be summarised as follows (*23*):

Step 1. Specify P to include enough random numbers p, having uniform probability density function.

Step 2. Specify Q to include enough random numbers q, having uniform probability density function.

Step 3. Generate p and q concurrently as a random pair. p is the x coordinate, while q is y coordinate.

Step 4. Specify a probability density function $f(x)$.

Step 5. Compare the q of each point with the function $f(x)$.

Step 6. Reject the points if $q > f(x)$. Accept the remaining points.

For example, we have the sampling results of a survey for the heights of men with different education levels in a city. The probability density of the survey results cannot be expressed by a function. How can we produce a large random number set (500) that complies with this survey probability? The survey distribution curve obtained is the solid black line in Figure 5; this example generated 3000 random numbers. The grey crosses above the black line are rejected random numbers (2500), and the circles below and on the line are the accepted random numbers (500).

4.2 Assumption-based multiple variables and distributions

In some cases, the required distribution does not come from observational results, but from our subjective judgment. This is often the case in parameter optimization, especially for parameterizing complex models, such as process-based models. If optimization of multiple parameters is needed, then it is necessary to generate random numbers in a multidimensional space. In other words, the

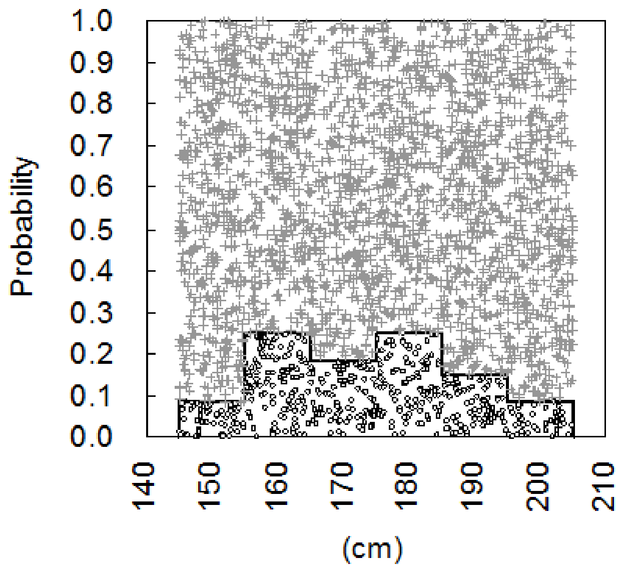

Figure 5: An example for generating random numbers using the acceptance-rejection method. The circles and crosses are accepted and rejected random data, respectively.

acceptance-rejection method is still used, but when the number of dimensions is greater than two, the time cost will be huge. In the interest of saving time, it may be necessary to use an effective algorithm, e.g. the Metropolis-Hastings algorithm (*24,25*), to search for optimal parameters and to achieve rapid convergence. This is the application of Markov Chain Monte Carlo (MCMC) method (*26*). Here we employ exhaustion to illustrate this random number generator.

For example, the empirical equation for predicting male children's height, weight, and lung capacity in a specific city is

$$y = 0.01(b \times w - a \times h) - 0.1352 \qquad (10)$$

where y is lung capacity, h is height, w is weight. a and b are undetermined parameters. Observed data is shown in Table 2. Now we parameterize this equation using statistic simulation instead of multiple regressions. Assume a ranges from 0.0 to 1.0, and b ranges from 0 to 10.0. Assume the distribution of these two parameters is uniform (Figure 6).

The algorithm can be summarized as follows:

Step 1. Generate a pair of random numbers (a, b) within a predetermined range. Substitute them into Equation 7.

Step 2. Obtain y based on observed h and w.

Step 3. Compare calculated y to measured y and go to Step 1 until residual error sum becomes minimal.

Table 2: Observation of male children's height (h), weight (w), and lung capacity (y).

No.	h	w	y	No.	h	w	y
1	145.8	34.5	2.3	16	155.4	39.1	2.5
2	159.6	48.0	3.0	17	144.8	33.0	2.5
3	137.7	29.9	2.0	18	154.4	42.6	2.3
4	152.2	44.3	2.8	19	157.1	38.5	2.5
5	160.4	40.2	2.8	20	147.7	31.4	1.5
6	168.4	49.1	2.9	21	157.8	37.2	2.0
7	153.2	32.0	1.7	22	154.1	35.2	2.0
8	153.6	41.5	2.7	23	154.4	32.4	1.7
9	150.0	39.9	2.0	24	147.6	34.2	2.5
10	133.4	32.5	1.8	25	171.1	41.8	2.7
11	161.9	45.9	2.7	26	134.9	27.3	1.3
12	144.9	32.0	1.8	27	152.1	35.7	1.8
13	156.8	37.5	2.7	28	146.8	37.8	2.2
14	161.1	37.7	2.0	29	155.1	32.4	1.7
15	150.7	33.6	2.2	30	154.1	36.2	2.0

The number of children (observed data) is 30 ($n = 30$). The units are cm (h), kg (w), and litre (y).

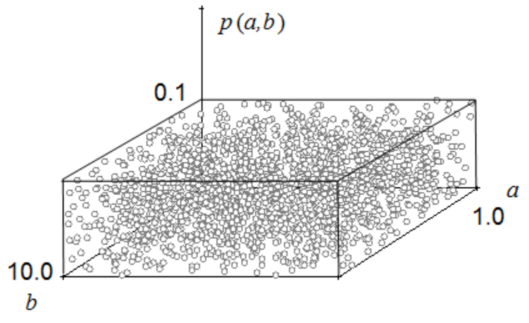

Figure 6: Two-dimensional distribution of parameters a and b in Equation 7. The circles represent all random pairs that include coordinate values a and b. Random numbers were generated to consist of 2000 pairs in this example. A point ($a = 0.4558$, $b = 8.0766$) was found within the distribution as an optimal parameter a and b.

This is an example of generating random numbers in a two-dimensional distribution (Figure 6). In this way, some parameter pairs can be solved. One of the pairs is solved as $a = 0.4558$, $b = 8.0766$. They are close to the optimal parameters. As the sample increases, the accuracy can be improved.

4.3 Parameter-based distributions

In model or parameter optimization, we may be required to obtain several random numbers. The function form of the distribution may not be required, but the sta-

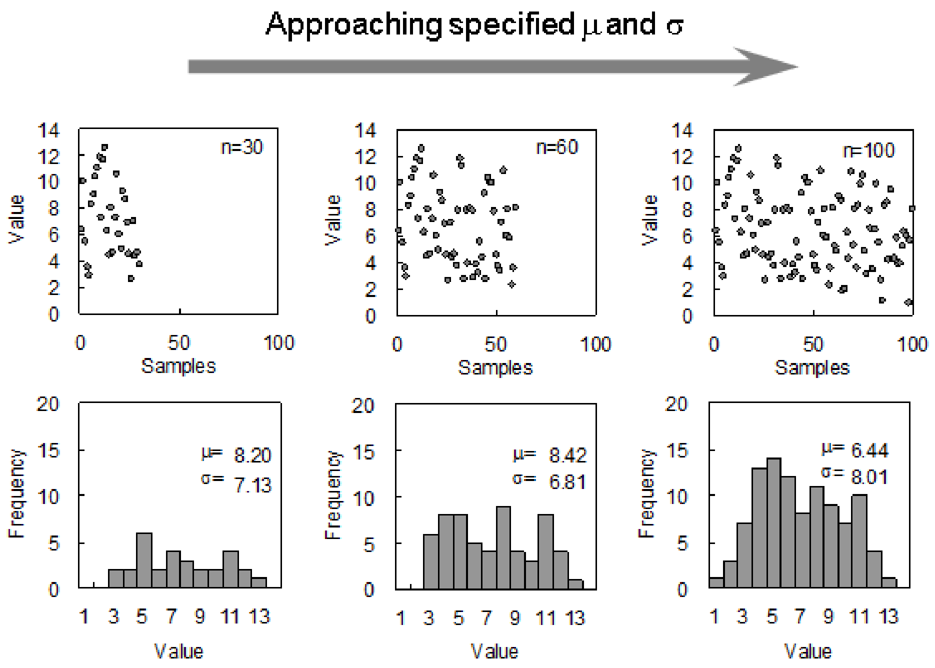

Figure 7: A generation of random data on a specified normal distribution ($\mu = 65$, $\sigma = 80$). The relative differences are less than 1.0% for both specified mathematical expectation (μ) and variance (σ). The dots denote generated numbers.

tistical measures of distribution must be specified, e.g. at least the mathematical expectation (μ) and variance (σ^2) or standard deviation (σ). Optimization of statistical measures is needed in these cases. The algorithm can be designed to generate random numbers by the Monte-Carlo simulation method. The simulation progresses dynamically step by step. While samples are generated adequately, the statistical measures of distribution approach the specified values (*27*). When the residual error between the random number and specified value reaches an acceptable low level, the simulation can end.

For example, the steps below demonstrate how to fit the two statistical measures by generating 100 random number sets ($\mu = 6.5$, $\sigma = 8$; Figure 7).

Step 1. Generate two random numbers near μ.

Step 2. Generate the next random number, calculate the μ and σ of the three random numbers, and compare them with the specified value.

Step 3. If the error does not meet the acceptable level, e.g. 2.0 for μ and 1.5 for σ, then abandon this random number. Generate the next random number, repeat Step 3.

Step 4. If the error is accepted and meets the acceptable level, accept this random number. Put the accepted random numbers in the set. Generate the next random number and repeat Step 3 until we have 100 accepted random numbers ($n = 100$).

Figure 7 illustrates the state of simulation when n is 30, 60, and 100. μ and σ are constantly approaching the specified value, finally reaching the relative error of less than 1.1% ($\mu = 6.43$, $\sigma = 8.01$). As more random numbers are generated, the accuracy of the resulting distribution will increase.

5 Summary

Three main methods for constructing random number generators, i.e. inverse transform, acceptance-rejection, and Monte-Carlo simulations, have been analyzed in this study by demonstrating applicable and simple computation experiments. As Section 3 explained, a random generator is easily realized for widely used distributions in study experiments. In practice, however, the distributions from observed data may not be close to the common forms. To generate random numbers that obey these practical distributions, the approach application depends on different cases. When selecting an appropriate approach, the issues of accuracy and efficiency are often considered. For the curve fitting method, polynomials can theoretically fit any continuous function of probability density, but their inverse functions may not be derived easily, especially for the inverse functions of fourth order polynomials. The approximate solutions of an inverse function are available by utilizing the numerical method with consideration to both accuracy and efficiency. Similarly, the accuracy and efficiency should still be balanced in the acceptance-rejection method. If the dimension is more than three, searching for an optimal solution becomes a challenge. In that case, coding experiments using Matlab or R would be necessary to finding an appropriate polynomial to decrease computational time. Briefly, random number generation under inexpressible distributions is an issue that we encounter in the applications of statistical simulation. This analysis suggests that this issue has flexible and diverse solutions, and always requires a balance between accuracy and efficiency. The selection of methods is illustrated in Figure 1. If the distribution density can be regressed with an expression, the approaches of curve fitting and inverse transform are suitable for building a random number generator. The merit of this method is convenience, and the demerit is that the use is limited by the form of distribution functions. In the cases of distributions that show irregular forms, the acceptance-rejection approach is a realistic option. Additionally, this approach generates random numbers strictly according to the given distribution curve (Figure 5); therefore, the accuracy is higher than the curve fitting method. As for some special purpose experiments such as the construction of assumption distributions or parameter-based distributions, the simulation methods can be flexibly utilized for various applications.

Competing interests

The author declares that he has no competing interests.

Acknowledgements

I am grateful to the two anonymous reviewers and the corresponding editor for their insightful comments and very helpful suggestions.

References

1. L. Kelvin, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **2**, 1 (1901).
2. G. Marsaglia, *Commun. ACM* **6**, 37 (1963).
3. R. C. Tausworthe, *Mathematics of Computation* **19**, 201 (1965).
4. G. S. Fishman, *Principles of Discrete Event Simulation* (John Wiley & Sons, Inc., New York, NY, USA, 1978).
5. P. A. W. Lewis, G. S. Shedler, *Naval Research Logistics Quarterly* **26**, 403 (1979).
6. B. Efron, R. Tibshirani, *Science* **253**, 390 (1991).
7. R. Y. Rubinstein, *Simulation and the Monte Carlo Method* (John Wiley & Sons, Inc., New York, NY, USA, 1981), first edn.
8. L. Devroye, *Non-Uniform Random Variate Generation(originally published with* (Springer-Verlag, 1986).
9. B. D. Ripley, *Stochastic Simulation* (John Wiley & Sons, Inc., New York, NY, USA, 1987).
10. J. Dagpunar, *Principles of random variate generation*, Oxford science publications (Clarendon Press, 1988).
11. A. M. Law, D. M. Kelton, *Simulation Modeling and Analysis* (McGraw-Hill Higher Education, 1999), third edn.
12. W. Aiello, S. Rajagopalan, R. Venkatesan, *Journal of Algorithms* **29**, 358 (1998).
13. L.-Y. Deng, D. K. J. Lin, *The American Statistician* **54**, 145 (2000).
14. N. Ferguson, B. Schneier, T. Kohno, *Cryptography Engineering: Design Principles and Practical Applications* (Wiley Publishing, 2010).
15. P. L'Ecuyer, *Annals of Operations Research* **53**, 77 (1994).
16. M. G. Luby, L. Michael, *Pseudorandomness and Cryptographic Applications* (Princeton University Press, Princeton, NJ, USA, 1994).
17. G. Marsaglia, A. Zaman, *Ann. Appl. Probab.* **1**, 462 (1991).
18. H. Niederreiter, I. E. Shparlinski, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, K.-T. Fang, H. Niederreiter, F. J. Hickernell, eds. (Springer Berlin Heidelberg, Berlin, Heidelberg, 2002), pp. 86–102.
19. S. Ross, *Introduction to Probability Models* (Elsevier Science, 2006).
20. D. H. Lehmer, *Proceedings of the Second Symposium on Large Scale Digital Computing Machinery* (Harvard University Press, Cambridge, United Kingdom, 1951), pp. 141–146.
21. M. Greenberger, *J. ACM* **8**, 163 (1961).
22. W. H. Payne, J. R. Rabung, T. P. Bogyo, *Commun. ACM* **12**, 85 (1969).
23. S. M. Ross, *Simulation, Fourth Edition* (Academic Press, Inc., Orlando, FL, USA, 2006).
24. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, *The Journal of Chemical Physics* **21**, 1087 (1953).
25. W. K. Hastings, *Biometrika* **57**, 97 (1970).
26. S. Asmussen, P. Glynn, *Stochastic Simulation: Algorithms and Analysis*, Stochastic Modelling and Applied Probability (Springer New York, 2007).
27. G. Givens, J. Hoeting, *Computational statistics*, Wiley series in probability and statistics (Wiley-Interscience, 2005).