VON NEUMANN'S COMPARISON METHOD FOR RANDOM SAMPLING FROM THE NORMAL AND OTHER DISTRIBUTIONS

BY

GEORGE E. FORSYTHE

STAN-CS-72-254

JANUARY, 1972

COMPUTER SCIENCE DEPARTMENT

School of Humanities and Sciences

STANFORD UNIVERSITY

Von Neumann's Comparison Method for Random Sampling from the Normal and Other Distributions

George E. Forsythe
Computer Science Department
Stanford University

Abstract

The author presents a generalization he worked out in 1950 of von Neumann's method of generating random samples from the exponential distribution by comparisons of uniform random numbers on $(0,1)$. It is shown how to generate samples from any distribution whose probability density function is piecewise both absolutely continuous and monotonic on $(-\infty,\infty)$. A special case delivers normal deviates at an average cost of only 4.036 uniform deviates each. This seems more efficient than the Center-Tail method of Dieter and Ahrens, which uses a related, but different, method of generalizing the von Neumann idea to the normal distribution.

This research was supported in part by the Office of Naval Research under Contracts N-00014-67-A-0112-0057 (NR 044-402) and N-00014-67-A-0112-0029 (NR 044-211), and by the National Science Foundation under Grant GJ- 992.

Von Neumann's Comparison Method for Random Sampling from the Normal and Other Distributions

George E. Forsythe
Computer Science Department
Stanford University

1. Introduction. In the summer of 1949, at the Institute for Numerical Analysis on the campus of the University of California, Los Angeles, John von Neumann [3] lectured on various aspects of generating pseudorandom numbers and variables. At the end he presented an ingenious method for generating a sample from an exponential distribution, based solely on comparisons of uniform deviates. In his last snetence he commented that his "method could be modified to yield a distribution satisfying any first-order differential equation".

In 1949 or 1950 I wrote some notes about what I assumed von Neumann had in mind, but I do not recall ever discussing the matter with him. This belated polishing and publication of those notes is stimulated by papers by Ahrens and Dieter [1, 2] in which several

related algorithms are studied, and by a personal discussion with the authors on how the von Neumann idea can be extended.

In Section 2 the general method is presented, and in Section 3 its efficiency is analyzed. In Sections 4 and 5 it is shown how the exponential and normal distributions show up as special cases. In Section 6 the method for a normal distribution is compared with the Center-Tail method of [1] and [2]. In Section 7 possible generalizations are mentioned.

Although this introduction has emphasized historical matters, the method of Section 6 is a good one, and is competitive with the best known methods for generating normal deviates.

I thank both Professors Ahrens and Dieter for their careful criticism of a first draft of this paper.

2. The general algorithm. Let $f(x) > 0$ be defined for all $x \geq 0$ and satisfy the first-order linear differential equation

$$f'(x) + b(x)\, f(x) = 0 \qquad (0 \leq x < \infty), \tag{1}$$

where $b(x) \geq 0$. Let

$$B(x) = \int_0^x b(t)\, dt\ , \tag{2}$$

and assume that

$$C = \int_0^\infty B(x)\, dx < \infty\ . \tag{3}$$

Then

$$f(x) = C^{-1} e^{-B(x)} \tag{4}$$

is the unique solution of (1) with $\int_0^\infty f(x)\, dx = 1$, and hence f is

the probability density distribution of a nonnegative random variable.

Suppose we have a supply of independent random variables u with a uniform distribution on [0, 1), and that we wish to generate a random variable y with the density distribution f(x). Here is one way to proceed.

We first prepare three tables of constants $\{q_k\}$, $\{r_k\}$, $\{d_k\}$ for k = 0, 1, . . . , K, as follows. (K is defined below.) Let $q_0 = 0$. For each k = 1, 2, . . . , K, pick q_k as large as possible, subject to the two constraints

$$q_k - q_{k-1} \leq 1, \tag{5}$$

$$B(q_k) - B(q_{k-1}) \leq 1 . \tag{6}$$

Next, compute

$$r_k = \int_0^{q_k} f(x)\,dx \qquad (k = 0, 1, \ldots, K). \tag{7}$$

Here K is chosen as the least index such that r_K exceeds the largest representable number less than 1. (K may be chosen smaller, if one sets $r_K = 1$, and if one is willing to truncate the generated variable by reducing any value above q_K to the interval $[q_{K-1}, q_K)$.)

Finally, compute

$$d_k = q_k - q_{k-1} \qquad (k = 1, 2, \ldots, K). \tag{8}$$

For simplicity we define the functions

$$G_k(x) = B(q_{k-1} + x) - B(q_{k-1}) \quad (k = 1, 2, \ldots, K). \tag{9}$$

Now we present the algorithm. Steps 1 to 3 determine which interval $[q_{k-1}, q_k)$ the variable y will belong to. Steps 4 to 11 determine the value of y within that interval.

1. [Begin choice of interval.] Set k $\leftarrow$ 1. Generate a uniform deviate u.
2. [Test.] If $u < r_k$, go to step 4.
3. [Increase interval.] If $u > r_k$, set k $\leftarrow$ k + 1 and go back to step 2.
4. [Begin computation of y in the selected interval.] Generate another uniform deviate u and set w $\leftarrow$ ud_k.
5. Set t $\leftarrow$ $G_k(w)$.
6. Generate another uniform deviate u*.
7. [Test.] If u* > t, go to step 11.
8. [Trial continues.] If u* < t, generate another uniform deviate u.
9. [Test.] If u < u*, set t $\leftarrow$ u and go back to step 6.
10. [Reject the trial.] If u > u*, go back to step 4.
11. [Finish.] Return y $\leftarrow$ $q_{k\ 1}$ + w as the sample variable.

We now show that the above algorithm works as claimed. Since we assume that each u < 1, the test in step 2 must be passed when k = K, if not sooner. Hence an interval $[q_{k-1}, q_k)$ is selected, and the values of r_k were chosen to make the probabilities of choosing the various intervals correct.

Fix k. The remainder of the algorithm can be described as follows: First, a random number w is selected uniformly from the interval $0 < w < d_k$.

Then the algorithm continues to generate independent uniform deviates u_i from $[0, 1)$ until the least n is found with

$$u_2 \geq G_k(w) \qquad (n = 1), \text{ or}$$

(10)

$$u_{n+1} \geq u_n < u_{n-1} < \ldots < u_3 < u_2 < G_k(w) \qquad (n \geq 2).$$

With probability 1 such an n will be found, as will be shown. If n is odd, we return $y \leftarrow q_{k-1} + w$. If n is even, we reject w and all the u, choose a new w, and repeat.

We now determine the probability $P(k, w)$ that one w determined in step 4 will be accepted without returning to step 4. Let $E_1(k, w)$ be the universe of all events. For $n = 2, 3, \ldots$, let $E_n(k, w)$ be the event

$$u_n < u_{n-1} < \ldots < u_3 < u_2 < G_k(w).$$

Then the probability of $E_n(k, w)$ is given by

$$\text{Prob}\{E_n(k, w)\} = \begin{cases} 1 & (n = 1) \\ \displaystyle\int_0^{G_k(w)} dx_2 & (n = 2) \\ \displaystyle\int_0^{G_k(w)} dx_2 \int_0^{x_2} dx_3 \cdots \int_0^{x_{n-1}} dx_n & (n \geq 3) \end{cases}$$

$$= \frac{G_k(w)^{n-1}}{(n-1)!} \qquad (\text{all } n).$$

The occurrence of (10) is the conjunction of $E_n(k, w)$ and not-$E_{n+1}(k, w)$. Since $E_{n+1}(k, w)$ implies $E_n(k, w)$, the probability that (10) occurs for a given n and w is

$$(11)\quad \text{Prob}\{E_n(k, w) \text{ and not-}E_{n+1}(k, w)\} = \frac{G_k(w)^{n-1}}{(n-1)!} - \frac{G_k(w)^n}{n!}.$$

Summing over all odd n, we see that

$$(12)\quad P(k, w) = \sum_{\text{odd } n}\left[\frac{G_k(w)^{n-1}}{(n-1)!} - \frac{G_k(w)^n}{n!}\right] = e^{-G_k(w)}.$$

Since $w \leq d_k$, we have

$$G_k(w) \leq G_k(d_k) = B(q_k) - B(q_{k-1}) \leq 1,$$

whence

$$(13)\quad P(k, w) \geq e^{-1}, \qquad \text{for all } k \text{ and } w.$$

Now $d_k^{-1}d\xi$ is the probability that w is selected in the interval $\xi \leq w \leq \xi + d\xi$. Combining this with (12), we see that the probability that $\xi \leq w \leq \xi + d\xi$ and that w is accepted is given by

$$(14)\quad \text{Prob}\{\xi \leq w \leq \xi + d\xi \text{ and } w \text{ is accepted}\} = e^{-G_k(w)}\frac{d\xi}{d_k}.$$

Corresponding to an accepted w, we return $y = q_{k-1} + w$ as the sample variable. Hence, from (14), the probability that y is in the range $x \leq y \leq x + dx$, for given k, is

$$\frac{1}{d_k} e^{-G_k(x - q_{k-1})} dx$$

$$= \frac{1}{d_k} e^{-B(x) + B(q_{k-1})} dx , \qquad \text{by (9)}$$

$$= \frac{1}{d_k} Ce^{B(q_{k-1})} C^{-1} e^{-B(x)} dx$$

$$= \frac{1}{d_k f(q_{k-1})} f(x)\, dx, \qquad \text{by (4).}$$

That is,

(15) $$\text{Prob}\{ x < y \leq x + dx \text{ and } y \text{ is accepted}\} = \frac{f(x)\, dx}{d_k f(q_{k-1})} .$$

Since this is proportional to $f(x)\, dx$, we see that any accepted y has the desired probability density distribution within the interval $[q_{k-1}, q_k)$. Since, from (13), the probability of an infinite loop back to step 4 is zero, the second half of the algorithm terminates with probability 1. This concludes the demonstration that the algorithm works as claimed.

3. Efficiency of the algorithm. For a general function b , I shall derive a representation for the expected number of uniformly distributed random variables u that must be used to generate one variable y with the probability density proportional to f(x). A similar derivation is given in [2].

The preliminary game to select k -- steps 1 to 3 of the algorithm -- requires one u.

The rest of the algorithm is different for each k, and we shall first determine the expected number N(k) of steps to determine y. To do this, we shall first assume that k is fixed and that w has been picked in the interval $0 \leq w < d_k$. Define $E_n(k, w)$ as in Section 2, and introduce the abbreviations

$$e_n = e_n(k, w) = \text{Prob}\{E_n(k, w)\} \qquad (n = 1, 2, \ldots) \tag{16}$$

and

$$g = G_k(w \ . \tag{17}$$

Then, as in Section 2, we have the following expression for the probability P(k, w) of accepting w without returning to step 4:

$$P(k, w) = (e_1 - e_2) + (e_3 - e_4) + (e_5 - e_6) + \ldots .$$

Moreover, given k and w and given that w is accepted, the expected number of uniform deviates u needed will be

$$\begin{aligned} m_a(k, w) &= P(k, w)^{-1}\left[2(e_1 - e_2) + 4(e_3 - e_4) + 6(e_5 - e_6) + \ldots\right] \\ &= \frac{1}{P(k, w)} \sum_{\text{odd } n} \left[\frac{g^{n-1}}{(n-1)!} - \frac{g^n}{n!}\right] (n+1). \end{aligned} \tag{18}$$

Similarly, the probability 1 - P(k, w) that w is rejected is given by

$$1 - P(k, w) = (e_2 - e_3) + (e_4 - e_5) + (e_6 - e_7) + \ldots .$$

Moreover, given k and w and that w is rejected, the expected number of uniform deviates u needed is

$$m_r(k, w) = [1 - P(k, w)]^{-1} \; [3(e_2-e_3) + 5(e_4-e_5) + 7(e_6-e_7) + \ldots]$$

(19)
$$= \frac{1}{1-P(k, w)} \sum_{\substack{\text{even } n \\ n \geq 2}} \left[\frac{g^{n-1}}{(n-1)!} - \frac{g^n}{n!} \right] (n + 1) .$$

Now, if a w is rejected, the algorithm returns to step 4, a new w is picked, and the process repeats. Let $M(k, w)$ be the expected number of uniform deviates selected until a y is finally selected, given a fixed k and an initially chosen w. Then $N(k)$ is the average of $M(k, w)$ over all w uniformly distributed on $0 \leq w < d_k$.

We have

(20) $$M(k, w) = P(k, w)m_a(k, w) + [1 - P(k, w)] [m_r(k, w) + N(k)] ,$$

since, in case w is rejected, the whole process is repeated. Using the expressions (18) and (19) for $m_a(k, w)$ and $m_r(k, w)$, we get from (20) that

$$M(k, w) = \sum_{n=1}^{\infty} \left[\frac{g^{n-1}}{(n-1)!} - \frac{g^n}{n!} \right] (n+1) + [1 - P(k, w)] N(k)$$

$$= 1 + e^g + [1 - P(k, w)] N(k),$$

or

(21) $$M(k, w) = 1 + e^{G_k(w)} + [1 - P(k, w)] N(k) .$$

Averaging (21) for $0 \leq w \leq d_k$ and using (12), we find that

$$N(k) = 1 + \frac{1}{d_k} \int_0^{d_k} e^{G_k(w)} \, dw + N(k) \left[1 - \frac{1}{d_k} \int_0^{d_k} e^{-G_k(w)} \, dw \right] .$$

Solving for N(k), we get

$$(22) \qquad N(k) = \frac{d_k + \int_0^{d_k} e^{G_k(w)}\, dw}{\int_0^{d_k} e^{-G_k(w)}\, dw} \; .$$

Finally, the expected number $\bar{N}$ of uniform deviates drawn in the main game until a y is returned is the average of N(k) over the intervals, weighted by the probabilities of selecting the various intervals. That is,

$$(23) \qquad \bar{N} = \sum_{k=1}^{\infty} N(k)\,[r_k - r_{k-1}] \; .$$

If we make use of (4), (7), and (9) to express $\bar{N}$ in terms of B(x), we obtain the ugly representation

$$\bar{N} = \sum_{k=1}^{\infty} \frac{d_k + e^{-B(q_{k-1})} \int_{q_{k-1}}^{q_k} e^{B(x)}\, dx}{e^{B(q_{k-1})} \int_{q_{k-1}}^{q_k} e^{-B(x)}\, dx} \times \frac{\int_{q_{k-1}}^{q_k} e^{-B(x)}\, dx}{\int_0^{\infty} e^{-B(x)}\, dx}$$

$$(24) \qquad = \frac{1}{\int_0^{\infty} e^{-B(x)}\, dx} \sum_{k=1}^{\infty} \left[d_k e^{-B(q_{k-1})} + e^{-2B(q_{k-1})} \int_{q_{k-1}}^{q_k} e^{B(x)}\, dx \right]$$

4. Special case: exponential distribution. If $b(x) = 1$ in (1), then $B(x) = x$ and $y(x) = e^{-x}$, corresponding to the exponential distribution treated in [3]. For the algorithm of Section 2 we have $q_k = k$, $d_k = 1$, $r_k = 1 - e^{-k}$, and $G_k(x) = x$, for all k. Since d_k and $G_k(x)$ are independent of k, steps 4 to 10 of the algorithm are the same for all k. They can therefore be carried out independently of steps 1 to 3. By (12), the probability that a chosen w is not accepted is $1 - P(k, w) = 1 - e^{-w}$ (for all k), and the average value of $1 - e^{-w}$ over $0 \leq w < 1$ is e^{-1}.

If the preliminary game of steps 1 to 3 were played, the interval $[k - 1, k)$ would be selected with probability $r_k - r_{k-1} = e^{-k} - e^{-(k+1)} = e^{-k}(1 - e^{-1})$, for $k = 1, 2, \ldots$. Thus the interval $[0, 1)$ would be accepted with probability $1 - e^{-1}$, and rejected with probability e^{-1}. For $k = 1, 2, \ldots$, if $[k-1, k)$ is rejected, then $[k, k+1)$ would be accepted with probability $1 - e^{-1}$, and rejected with probability e^{-1}. Since the rejection ratio for each interval has the same value e^{-1}, which is the a priori probability of rejecting in the main game any w selected in step 4, von Neumann could use the rejection of w as the signal to change the interval from $[k-1, k)$ to $[k, k+1)$. Thus the preliminary game of steps 1-3 is unnecessary for the exponential distribution. This made von Neumann's game very elegant. I know of no comparable trick for general $b(x)$.

From (22) and (23), since $N(k) = \bar{N}$, we see that for the exponential distribution

$$(25) \quad \bar{N} = \frac{1 + (e-1)}{1 - e^{-1}} = \frac{e}{1 - e^{-1}} \doteq 4.30026 ,$$

as stated in [1]. There was an error in [3].

5. Special case: normal distribution. If $b(x) = x$ in (1), then $B(x) = x^2/2$ and $f(x) = \sqrt{2/\pi}\ e^{-x^2/2}$, corresponding to the positive half of the normal distribution. For the algorithm of Section 2 we have

$$q_0 = 0,\ q_1 = 1,\ \ldots,\ q_k = \sqrt{2k - 1} \quad (k > 2).$$

Hence

$$d_1 = 1,\ d_2 = \sqrt{3} - 1,\ \ldots,\ d_k = \sqrt{2k - 1} - \sqrt{2k - 3} \quad (k > 2).$$

Also,

$$G_k(x) = \frac{x^2}{2} - q_{k-1}x \quad (k \geq 1).$$

The values of r_k must be computed from the probability integral. The table below gives 15-decimal values of q_k, d_k, r_k, and $N(k)$ for $k = 1, 2, \ldots, 36$, as computed in Fortran on Stanford's IBM 360/67 computer in double precision.

To generate normal deviates, one selects K and prestores the values of r_k, q_k, and d_k for $k = 1, 2, \ldots, K$. Then set $q_0 \leftarrow 0$ and $d_K \leftarrow 1$. (The limit $K = 12$ permits normal deviates up to + 5.0 to be generated, and the deviates will be truncated less than once in a million trials. A higher limit will decrease the probability of truncation.)

As suggested in [2], one should start the algorithm with a preliminary determination of the sign of the normal deviate. We do this

in steps N1-N3 of the following algorithm. At entry to Step N4, u is a uniform deviate on the interval [0, 1). The rest is the algorithm of Section 2, with the sign appended in the last step.

N1. [Begin choice of sign and interval.] Set k $\leftarrow$ 1. Generate a uniform deviate u on [0, 1). Set u $\leftarrow$ 2u.

N2. [Test for sign.] If u < 1, set s $\leftarrow$ 1 and go to step N4.

N3. If u $\geq$ 1, set s $\leftarrow$ -1, and set u $\leftarrow$ u - 1.

N4. [Test for interval.] If u $\leq r_k$, go to step N6.

N5. [Increase interval.] If u > r_k, set k $\leftarrow$ k + 1 and go back to step N4.

N6. [Begin generation of |y| in the selected interval.] Generate another uniform deviate u on [0, 1) and set w $\leftarrow$ ud_k.

N7. Set t $\leftarrow$ $G_k(w)$.

N8. Generate another uniform deviate u* on [0, 1).

N9. [Test.] If u* $\geq$ t, go to step N13.

N10. [Trial continues.] If u* < t, generate another uniform deviate u on [0, 1).

N11. [Test.] If u < u*, set t $\leftarrow$ u and go back to step N8.

N12. [Reject the trial.] If u $\geq$ u*, go back to step N6.

N13. [Finish.] Return y $\leftarrow$ s (q_{k-1} + w) as the sample normal variable.

As in Section 3, we let N(k) be the expected number of selections of uniform deviates in steps N4-N13, as a function of k. We have from (22):

$$N(1) = \frac{1 + \int_0^1 e^{w^2/2}\,dw}{\int_0^1 e^{-w^2/2}\,dw},$$

$$N(k) = \frac{d_k + e^{3/2-k}\int_{q_{k-1}}^{q_k} e^{w^2/2}\,dw}{e^{k-3/2}\int_{q_{k-1}}^{q_k} e^{-w^2/2}\,dw} \qquad (k \geq 2).$$

Numerical values of N(k) are given in the table. Using the asymptotic formula

$$\int_x^{x+h} e^{\pm t^2/2}\,dt \sim \left[\frac{e^{\pm x^2/2}}{x}\left(1 \pm \frac{1}{x^2}\right)\right]_x^{x+h}, \text{ as } x \longrightarrow \infty,$$

one can show that

$$\lim_{k \longrightarrow \infty} N(k) = \frac{e}{1 - e^{-1}} \doteq 4.30026. \tag{26}$$

Cf. (25). The equality (26) was written to me by U. Dieter.

I have used the same computer to establish that

$$\overline{N} = \sum_{k=1}^{\infty} N(k)\,(r_k - r_{k-1}) \doteq 3.03585,$$

so that the expected number of uniform deviates chosen in order to generate one normal deviate is $1 + \overline{N} \doteq 4.03585$.

The correctness of this algorithm for generating normal deviates, as well as the value of N, have been confirmed in unpublished experiments by A. I. Forsythe and independently by J. H. Ahrens.

6. Comparison with the Center-Tail method of Dieter and Ahrens. In [1], Dieter and Ahrens give a related but different modification of the von Neumann idea for the generation of normal deviates. There are only two intervals, the center and the tail, and the algorithms are quite different for the two. The expected number of uniform deviates needed is near 6.321, and computation of a square root is required in approximately 16 per cent of the cases.

The algorithm of Section 5 above requires no function call, but its main advantage over the Center-Tail method lies in requiring about two-thirds the number of uniform deviates. This should be reflected in a shorter average time of execution.

The Dieter-Ahrens algorithm for the center interval closely resembles my algorithm for each interval, and the proofs are very close to those given above. The big difference is that in [1] all variables u_i have the cumulative distribution function x^2 $(0 \leq x < 1)$, and the comparisons are of the form

$$u_{n+1} \geq u_n < u_{n-1} < u_{n-2} < \ldots < u_3 < u_2 < u_1 .$$

In contrast, in this paper all variables u_i have uniform distributions and the comparisons take the form (for the principal case $k = 1$):

$$u_{n+1} \geq u_n < u_{n-1} < u_{n-2} < \ldots < u_3 < u_2 < \frac{u_1^2}{2} .$$

Changing the distribution function in [1] costs an extra uniform deviate and a comparison for each u_i, whereas forming $u_1^2/2 = G_1(w)$ in Section 5 is done only once for each chain of u_i's. Moreover, the fact that $u_1^2/2$ is usually small means that most of the time $u_2 \geq u_1^2/2$ and hence u_1 is accepted immediately. This contributes to keeping N low in my algorithm. Finally, the use of $G_k(w)$ makes it possible to use the von Neumann technique in any interval in which $G_k(w)$ can be evaluated.

In a more recent manuscript [2] Dieter and Ahrens have improved their Center-Tail method so that the comparisons are simpler and the expected number of uniform deviates needed is reduced to near 5.236. According to the authors, the improved Center-Tail method is still somewhat slower than my algorithm.

7. Further generalizations. Let $f(x)$ $(-\infty < x < \infty)$ be the probability density function of a random variable F. Under what conditions on f could the von Neumann idea be applied to pick a sample from F? It is sufficient that the interval $(-\infty, \infty)$ be the union of a set of abutting intervals $I_k = [q_{k-1}, q_k]$ $(k = \ldots, -2, -1, 0, 1, 2, \ldots)$ such that in each closed interval I_k either $f(x) \equiv 0$ or the following three conditions all hold: $f(x) > 0$, f is absolutely continuous, and f is monotonic.

Then a preliminary game can be played to select an interval I_k. If $b(x) = -f'(x)/f(x) \geq 0$ in I_k, the algorithm of Section 2 can be adapted to select a value in I_k with a density distribution proportional to $f(x)$. (It may be necessary to subdivide I_k so that (5) and (6) hold.)

If $b(x) < 0$ in I_k, change x to $-x$ and follow an analogous algorithm.

The main practical difficulties of the algorithm are these:

(a) One must evaluate various integrals like $\int_a^x f(t)\,dt$, in order to determine the parameters needed to pick the intervals I_k during execution, and to evaluate the needed r_k, q_k, and d_k. These computations have to be done only once in designing the algorithm.

(b) One must evaluate $G_k(w)$ for arbitrary w in $[0, d_k]$ during each execution of the algorithm. Note that

$$G_k(w) = B(q_{k-1} + w) - B(q_{k-1})$$

$$= \int_{q_{k-1}}^{q_{k-1}+w} b(t)\,dt = -\int_{q_{k-1}}^{q_{k-1}+w} \frac{f'(t)}{f(t)}\,dt$$

$$= \ell n \left[\frac{f(q_{k-1})}{f(q_{k-1} + w)}\right] .$$

Since only (b) is done on-line, the success of an algorithm would seem to depend only on the ability to evaluate $\ell n\, f(x)$ rapidly. We thus see that having $f(x) = C \exp(\varphi(x))$ (and hence a solution of an equation of type (1)) is of great practical advantage, but it is not essential in principle to the use of von Neumann's idea.

References.

1. J. H. Ahrens and U. Dieter, "Computer methods for sampling from the exponential and normal distributions," Comm. Assoc. Computing Mach., vol. 15 (1972), pp. 000-000.

2. U. Dieter and J. Ahrens, "A combinatorial method for the generation of normally distributed random numbers," to appear.

3. John von Neumann, "Various techniques used in connection with random digits" (summary written by George E. Forsythe), pp. 36-38 of Monte Carlo Method, [U. S.] National Bureau of Standards, Applied Mathematics Series, vol. 12 (1951). Reprinted in John von Neumann, Collected Works, vol. 5, pp. 768-770, Pergamon Press, 1963.

February 9, 1972.

K	D(K)	C(K)	P(K)	N(K)
1	1.000000000000000	1.000000000000000	0.682689492137085	2.565328528159702
2	1.732050807568877	0.732050807568877	0.916735483364494	4.003318474519820
3	2.236067977499789	0.504017169930912	0.974652681322531	4.156694307617552
4	2.645751311064590	0.409683333564800	0.991849028406497	4.204828245234114
5	3.000000000000000	0.354248689354094	0.997300203936739	4.228693206176426
6	3.316624790355399	0.316624790355399	0.999088881122846	4.242985592718185
7	3.605551275463989	0.288926485085894	0.999688509023232	4.252511250648343
8	3.872983346207416	0.267432070743427	0.999892488823270	4.259316655645425
9	4.123105625617660	0.250122279410243	0.999962620181598	4.264422459071955
10	4.358898943540673	0.235793317923013	0.999986928154633	4.268395116240194
11	4.582575694955839	0.223676751415166	0.999995407166288	4.271574385949102
12	4.795831523312719	0.213255828356879	0.999998379986017	4.274176475319811
13	5.000000000000000	0.204168476687280	0.999999426696856	4.276345543165833
14	5.196152422706631	0.196152422706631	0.999999796544538	4.278181414471334
15	5.385164807134504	0.189012384427872	0.999999927621701	4.279755410332964
16	5.567764362830021	0.182599556955175	0.999999974197157	4.281119847417264
17	5.744562646538028	0.176798283708006	0.999999990784112	4.282313973621910
18	5.916079783099616	0.171517136561587	0.999999996702946	4.283367809863551
19	6.082762530298219	0.166682747198603	0.999999998818707	4.284304711573850
20	6.244997998398398	0.162235468100178	0.999999999576194	4.285143121626455
21	6.403124237432848	0.158126239034450	0.999999999847770	4.285897797630336
22	6.557438524302000	0.154314286869152	0.999999999945260	4.286580688748215
23	6.708203932499369	0.150765408197368	0.999999999980296	4.287201573514557
24	6.855654600401044	0.147450667901675	0.999999999292901	4.287768531362551
25	7.000000000000000	0.144345399598955	0.999999999974403	4.288288296339975
26	7.141428428542850	0.141428428542850	0.999999999990763	4.288766525985538
27	7.280109889280516	0.138681460737668	0.999999999996664	4.289208008195656
28	7.416198487095662	0.136088597815144	0.999999999998794	4.289616822149568
29	7.549834435270748	0.133635948175086	0.999999999999563	4.289996464772889
30	7.681145747868608	0.131311312597858	0.999999999999842	4.290349951052623
31	7.810249675906652	0.129103928038046	0.999999999999942	4.290679894302099
32	7.937253933193771	0.127004257287117	0.999999999999979	4.290988570900672
33	8.062257748298543	0.125003815104778	0.999999999999992	4.291277972902334
34	8.185352771872449	0.123095023573900	0.999999999999997	4.291549851085148
35	8.306623862918073	0.121271091045625	0.999999999999998	4.291805750408574
36	8.426149773176358	0.119525910258284	0.999999999999998	4.292047039396059

AVERAGE NUMBER = 3.035853343141112