George Forsythe's last paper *

Richard P. Brent

Australian National University
Canberra

In memory of
George and Sandra Forsythe

I have a feeling, however, that it is somehow silly to take a random number and put it elaborately into a power series ⋯

John von Neumann
Collected Works
Vol. 5, pg. 769

Summary

I will describe von Neumann's elegant idea for sampling from the exponential distribution, Forsythe's generalization for sampling from a probability distribution whose density has the form $\exp(-G(x))$, where $G(x)$ is easy to compute (e.g. a polynomial), and my refinement of these ideas to give an efficient algorithm for generating pseudo-random numbers with a normal distribution. Later developments will also be mentioned.

Background

Roger Hockney and I are the only people who were lucky enough to have both George Forsythe and Gene Golub as PhD advisors (see the *Mathematics Genealogy Project*).

In my case this came about because Gene went on sabbatical to the UK, and George took over while he was away. However, I managed to finish before Gene returned to Stanford. That was in the days before email, and there was a mail strike in UK, so communication with Gene was difficult. Perhaps that helped me to finish quickly, because if Gene had been at Stanford he probably would have asked me to do more work on the last chapter!

Most of you here today know Gene, but only the older ones will remember George and Sandra Forsythe, so today I will talk about George Forsythe and an interesting link back to John von Neumann and the early days of computers.

History

In summer 1949 Forsythe attended some lectures at UCLA by John von Neumann on the topic of random number generation. The lectures were part of a Symposium on the (then new) *Monte Carlo method*. It seems that von Neumann never wrote up the lectures, but a fascinating 3-page summary was written by Forsythe and published in 1951.

Forsythe must have continued to think about the topic because, shortly before he died, he wrote a Stanford report *Von Neumann's comparison method for random sampling from the normal and other distributions* (STAN-CS-72-254, dated February 9, 1972).

This expanded on a brief comment by von Neumann that his

> *method* [for the exponential distribution] *can be modified to yield a distribution satisfying any first-order differential equation.*

Collected Works **5**, 770.

Ahrens, Dieter and Knuth

Forsythe intended that his Stanford report would form the basis of a joint paper with J. H. Ahrens and U. Dieter, who had discovered related results independently, and had presented them at Stanford in October 1971.

After Forsythe died in April 1972, Don Knuth submitted the Stanford report to *Mathematics of Computation*, and it was published with only minor changes in the October 1972 issue.

This was Forsythe's last published paper, with the possible exception of a paper by E. H. Lee and Forsythe in *SIAM Review* (submitted in October 1971 and published in January 1973).

Ahrens and Dieter published a follow-up paper in *Math. Comp.* (1973), and I published an implementation GRAND of my improvement of the Forsythe – von Neumann method in *Comm. ACM* (1974).

That was in the days before *TOMS*, when interesting algorithms were still published in *Communications*.

The problem

Suppose we want to sample a probability distribution with density

$$f(x) = e^{-G(x)} ,$$

where $G(x)$ is some simple function, e.g. a polynomial. Von Neumann illustrated his idea for the exponential distribution

$$G(x) = x, \ \ (x \geq 0) ,$$

but it also applies to the normal distribution

$$G(x) = x^2/2 + \ln(2\pi)/2 .$$

The function $f(x)$ satisfies a first-order linear differential equation

$$f' + G'(x)f = 0 ,$$

and conversely. That is why Von Neumann made the remark about first-order differential equations.

Von Neumann's insight

The obvious way to generate a sample from the exponential distribution is to generate a sample $u \in (0,1]$ from the uniform distribution and then take

$$x = -\ln(u) .$$

However, the evaluation of $\ln(u)$ is expensive (relative to the cost of generating u by an efficient uniform random number generator). Also, this method does not generalize well to the normal distribution, where we would need to evaluate the inverse of the normal distribution function

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-t^2/2)\, dt .$$

Von Neumann's insight was that we can generate a random sample using a small number (on average) of samples from a uniform distribution, and evaluation of $G(x)$ at a small number of points. There is no need to compute any expensive special functions!

Probability of a run

Suppose for the moment that $0 \leq u_1 = G \leq 1$. Generate samples $u_2, u_3, \ldots$ from the uniform distribution so long as the numbers are decreasing, and then stop. In other words, find $n \geq 1$ such that

$$G = u_1 > u_2 > u_3 > \cdots > u_n \leq u_{n+1} . \quad (1)$$

The probability that

$$G > u_2 > u_3 > \cdots > u_n > u_{n+1} \ \text{ is}$$

$$\frac{G^n}{n!} = \frac{\text{Prob}(\max(u_2, \ldots, u_{n+1}) < G)}{n!} ,$$

so the probability of (1) is

$$p_n = \frac{G^{n-1}}{(n-1)!} - \frac{G^n}{n!} .$$

Check: $p_1 + p_2 + \cdots = 1$ by telescoping series, so the algorithm terminates with probability 1.

Exercise: The expected value of n is $\exp(G)$.

The power series for $\exp(-G)$

What is the probability that our final n is *odd*? It is just

$$p_1 + p_3 + \cdots = 1 - G + \frac{G^2}{2!} - \frac{G^3}{3!} + \cdots = \exp(-G) \ .$$

This suggests a *rejection method* for generating a sample from the distribution with density $\exp(-G(x))$ on some interval $[a, b]$:

1. Generate uniform $w \in [a, b]$ and set $u_1 \leftarrow G(w)$.

2. Generate uniform $u_2, u_3, \ldots \in [0, 1]$ until condition (1) is satisfied ($u_n \leq u_{n+1}$).

3. If n is even, return to step 1 (i.e. *reject w*).

4. Return w (i.e. *accept w*).

This works because the probability that w is accepted, i.e. the probability that n is odd at step 3, is exactly $\exp(-G(w))$.

9

An important condition

The algorithm only works correctly if $0 \leq G(w) \leq 1$ on the interval $w \in [a, b]$.

To apply the idea to the exponential or normal distributions we have to split the infinite interval $[0, +\infty)$ or $(-\infty, +\infty)$ into a union of finite intervals I_k. Provided the intervals I_k are small enough, we can use the algorithm to generate samples from each I_k.

Thus, first select k with the correct probability

$$\int_{I_k} \exp(-G(x)) \, dx \ ,$$

then use the Forsythe – von Neumann algorithm to get a sample from I_k.

Minor detail: The function $G(x)$ has to be modified by addition of a constant to give the appropriate function $G_k(x)$ on the interval I_k. For example, we could use $(x^2 - a^2)/2$ for the normal distribution on $[a, b]$, where $0 \leq a < b$ and $b^2 - a^2 \leq 2$.

10

Exponential and normal distributions

For the exponential distribution, consider the intervals

$$I_k = [(k-1)\ln 2, \ k\ln 2) \ .$$

For convenience on a binary computer, our sample should lie in I_k with probability 2^{-k}, $k = 1, 2, \ldots$ We can select k by counting the leading zero bits in a uniform random number (giving $k - 1$). Then we can apply a rejection method to get a sample with the correct distribution from I_k.

For the normal distribution, it is convenient to randomly generate the sign, then consider the interval $[0, \infty)$. We subdivide this interval into intervals $I_k = [a_{k-1}, a_k)$ such that $a_0 = 0$ and

$$\sqrt{\frac{2}{\pi}} \int_{a_{k-1}}^{a_k} \exp(-x^2/2) \, dx = 2^{-k}$$

for $k = 1, 2, \ldots, w$. It is easy to precompute a table of the constants a_k. The table is small, since we can neglect probabilities 2^{-k} if k is greater than the wordlength w of the computer.

11

Historical notes

For the exponential distribution, von Neumann took intervals $I_k = [k - 1, k)$ so the probability of sampling from I_k is $(e - 1)/e^k$. He did this because he had a trick for combining the trials with the selection of intervals. However, his trick does not generalize to other distributions.

For the normal distribution, Forsythe used intervals defined by $a_0 = 0$ and

$$a_k = \sqrt{2k - 1} \ \text{ for } \ k \geq 1 \ .$$

Presumably he did this because then

$$a_k^2 - a_{k-1}^2 = 2 \ \text{ for } \ k \geq 2 \ .$$

This choice of a_k is what Sandra Forsythe used in her implemention of the algorithm:

> *The correctness of this algorithm $\cdots$ (has) been confirmed in unpublished experiments by A. I. Forsythe and independently by J. H. Ahrens.*

George Forsythe

12

Comment

It is better to use the intervals that I defined, as used in GRAND, because then we do not need to store a table of probabilities (they are just negative powers of 2). With my choice it can be shown that

$$a_k^2 - a_{k-1}^2 < 2\ln 2 < 1.39 \ \text{ for } \ k \geq 1 \,.$$

As well as reducing the table size, my choice reduces the expected number of calls to the uniform random number generator.

Refinements

The algorithms proposed by Forsythe and von Neumann were inefficient in the sense that they used more uniform samples than necessary to generate one sample from the exponential or normal distribution.

The algorithm implemented by Sandra Forsythe requires (on average) 4.04 uniform samples per normal sample. For von Neumann's algorithm the corresponding constant is 5.88.

Ahrens and Dieter (1973) reduced the constant 4.04 to 2.54 (and even further at the expense of larger tables and more complications).

In my 1974 paper describing GRAND I showed how 4.04 could be reduced to 1.38 by using a better subdivision of the infinite interval $[0, +\infty)$ and by not wasting random bits. For example, after step 2,

$$\frac{u_{n+1} - u_n}{1 - u_n}$$

is uniformly distributed and can be used later.

Further refinements

In principle, by using larger tables, it is possible to reduce the constant to $1 + \varepsilon$ for any $\varepsilon > 0$, but this would not necessarily give a faster algorithm. In practice 1.38 is small enough.

Later developments

The idea of *rejection methods* was developed by many people to give efficient algorithms for sampling from a great variety of distributions – see for example the books by Devroye (now available online) and Knuth (Vol. 2).

Specifically for the normal distribution, Forsythe's method (as improved and implemented in GRAND) is much faster than earlier methods, such as the Box-Muller and Polar methods.

There are now many different algorithms for the normal distribution, but I think it is fair to say that none are *clearly better* than GRAND.

The differences between the best algorithms are small – often there is a tradeoff between space and time, and the relative speeds depend on the machine architecture as well as on the choice of uniform random number generator.

Wallace's method

The only method that is clearly much faster than GRAND is Wallace's method, proposed in 1994 by Chris Wallace. It does not use a uniform random number generator. Instead, a pool of normally distributed numbers is maintained and refreshed by performing orthogonal transformations.

The key observation is that, if x is a vector of n independent, normally distributed numbers, then the probability density of x,

$$(2\pi)^{-n/2} \exp\left(-(x_1^2 + \cdots + x_n^2)/2\right) \ ,$$

is a function of $||x||_2$. i.e. the distribution has spherical symmetry. If follows that, if Q is an $n \times n$ orthogonal matrix, then

$$y = Qx$$

is another vector of normally distributed numbers, because $||y||_2 = ||x||_2$.

Wallace's method is interesting and fast, but suffers from some statistical problems: see my paper in the Wallace memorial, to appear (available as a preprint).

17

References

[1] J. H. Ahrens and U. Dieter, Computer methods for sampling from the exponential and normal distributions, *Comm. ACM* **15** (1972), 873–881.

[2] J. H. Ahrens and U. Dieter, Extensions of Forsythe's method for random sampling from the normal distribution, *Math. Comp.* **27**, 124 (Oct. 1973), 927–937.

[3] G. E. Box and M. E. Muller, A note on the generation of random normal deviates, *Ann. Math. Statistics* **29** (1958), 610–611.

[4] R. P. Brent, Algorithm 488: A Gaussian pseudo-random number generator [G5], *Comm. ACM* **17**, 12 (1974), 704–706.

[5] R. P. Brent, Some comments on C. S. Wallace's random number generators, *Computer Journal*, to appear. `http://wwwmaths.anu.edu.au/~brent/pub/pub213.html`

18

[6] L. Devroye, *Non-Uniform Random Variate Generation*, Springer-Verlag, New York, 1986. Available from `http://cg.scs.carleton.ca/~luc/rnbookindex.html`

[7] G. E. Forsythe, Von Neumann's comparison method for random sampling from the normal and other distributions, Report STAN-CS-72-254, January 1972 [but dated February 9, 1972], 21 pp.

[8] G. E. Forsythe, Von Neumann's comparison method for random sampling from the normal and other distributions, *Math. Comp.* **26**, 120 (Oct. 1972), 817–826. [Submitted posthumously April 27, 1972; Forsythe died April 9, 1972.]

[9] A. J. Kinderman and J. F. Monahan, Computer generation of random variables using the ratio of uniform deviates, *ACM Trans. on Mathematical Software* **3** (1977), 257–260.

19

[10] D. E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms* (third edition), Addison-Wesley, Menlo Park, 1998.

[11] E. H. Lee and G. E. Forsythe, Variational study of nonlinear spline curves, *SIAM Review* **15**, 1 (Jan. 1973), 120–123. [Submitted October 21, 1971; revised March 13, 1972.]

[12] J. L. Leva, A fast normal random number generator, *ACM Trans. on Mathematical Software* **18** (1992), 449–453.

[13] W. D. MacLaren, G. Marsaglia and T. A. Bray, A fast procedure for generating normal random variables, *Comm. ACM* **7** (1964), 4–10.

[14] M. E. Muller, A comparison of methods for generating normal variates on digital computers, *J. ACM* **6** (1959), 376–383.

20

[15] J. von Neumann, Various techniques used in connection with random digits, in *Monte Carlo Method*, Appl. Math. Series **12**, US Nat. Bureau of Standards, 1951, 36–38 (summary written by G. E. Forsythe); reprinted in *John von Neumann Collected Works* (ed. A. H. Taub), **5**, Pergamon Press, New York, 1963, 768–770.

[16] C. S. Wallace, Fast pseudo-random generators for normal and exponential variates, *ACM Trans. on Mathematical Software* **22** (1996), 119–127.

[17] *The Mathematics Genealogy Project*, `http://genealogy.math.ndsu.nodak.edu/`

21