THE SQUEEZE METHOD FOR GENERATING GAMMA VARIATES

GEORGE MARISGALIA*
Computer Science Dept., Washington State University, Pullman, Washington 99163, U.S.A.

Communicated by Donald E. Knuth
(Received February 1976, and in revised form April 1977)

Abstract—This paper describes an exact method for computer generation of random variables with a gamma distribution. The method is based on the Wilson-Hilferty transformation and an improvement on the rejection technique. The idea is to "squeeze" a target density between two functions, the top one easy to sample from, the bottom one easy to evaluate.

1. INTRODUCTION

One of the most promising methods for dealing with the gamma distribution, either for numerical work or for generating random variables, is to exploit the remarkable fact that a gamma variate is very nearly the cube of a suitable normal variate. This is the Wilson-Hilferty transformation[10]. It was used by Greenwood[2] to provide a fast and exact method for generating gamma variates. Our group at McGill had independently developed a method based on the WH transformation, but by incorporating it in what we call the squeeze technique. Subsequent comparisons have shown the method to be as simple as and faster than any we have seen published.

Thus the purpose of this paper is twofold: (1) to point out a simple and fast computer program for generating gamma variates, and (2) to describe in some detail the squeeze method, which, although it dates back to about 1961 and can offer significant improvement over the straight rejection technique, is not widely known.

The procedure described here was developed as an appendage to the McGill Random Number Package[9], the body of which provides uniform, exponential and normal variates by means of carefully developed machine-language programs. The important variates that depend on parameters such as gamma, beta and Poisson, are provided via Fortran subroutines.

The program given below is simpler but not quite as fast as one based on what we call the exact-approximation method[8], which uses this idea: Choose a suitable approximation $F^*$ to $F^{-1}$, the inverse of the required distribution, so that $F^{-1}(U)$ is exactly, and $F^*(U)$ is approximately, distributed as $F$. Then, rather than a uniform $U$, one uses a nearly uniform $X$ such that $F^*(X)$ is distributed exactly as $F$. This method, and extensions that use nearly normal rather than nearly uniform variates, can lead to very fast programs. A version for gamma variates runs perhaps 15–20% faster, but is several times larger, than the one below.

The basic input to any (digital) computer generation of random variables is a sequence of independent uniform random variables. In the procedure below we do not directly use uniform, but rather normal and exponential random variables, and thus we assume that the reader has available a procedure for fast and exact generation of normal and exponential variates. See for example [4, 5, 3, p. 105–113], and comments on implementation in Section 3.

2. THE SQUEEZE METHOD

In the ordinary rejection method, one chooses a point uniformly from under one curve then tests to see if it is under another, the target curve. The latter will usually be complicated. In the squeeze method, one fits a third, easy-to-evaluate curve under the target, for a possible quick decision on acceptance of the random point. This method goes back to 1961—see [6, 7 or 3, p. 108].

*This research was supported in part by the National Research Council of Canada. Grant A7001.
We will use the squeeze technique to develop an exact method for generating random variables $W$ with a gamma density $w^{a-1}e^{-w}/\Gamma(a)$. The method will be theoretically exact, but in practice is limited by the precision of the computer. We omit the scale parameter in dealing with the gamma density. Multiples of $W$ will provide variates with the more general density $cw^{a-1}e^{-bw}$. These include the chi-square densities.

Refer to Fig. 1, which shows, for several values of the parameter $a$, three functions $h(x) \leq g(x) \leq f(x)$. The top function, $f$, is the normal density. The method is to choose points $(X, Y)$ with a uniform distribution under $f(x)$ until we get one that also lies under $g(x)$, then exit with $W = u(sX + 1 - s^2)$, where $s = a^{-1/2}$. We may avoid testing under $g$ most of the time by first testing under $h$. The functions $f$ and $h$ are chosen to be close to $g$ and convenient to handle. This is the essence of the squeeze method. ($f$ is not very close to $g$ when $a < 1$, but the procedure is still reasonable for $1/3 < a < 1$ because $h$ is close to $g$. The curves for $a = 0.34$ and $0.5$ are given to indicate limiting behavior; this particular squeeze method is not recommended for $a < 1$.)

We give details of the method in the form of a 3-step algorithm, then amplify and justify by means of a series of remarks and reference to Fig. 1.

Algorithm for generating a gamma variate $W$, density $w^{a-1}e^{-w}/\Gamma(a) \mid w > 0$, for any value of the parameter $a > 1/3$, but recommended for $a > 1$

Step 1. Generate a standard normal random variable $X$. Put $Z = sX + 1 - s^2$, where $s = a^{-1/2}$. If $Z \leq 0$ repeat this step.

Fig. 1. Functions $h \leq g \leq f$. If the point $(X, Y)$ is uniformly distributed under $g$ then $a(sX + 1 - s^2)$ has the gamma distribution. Points uniform under $f$ are easy to generate; those under $h$ are easy to identify.
Step 2. Generate a standard exponential random variable $E$.

Step 3. If

$$\frac{1}{2}x^2 + E > \frac{1}{2}x_0^2 + a(Z^3 - z^3) + (3a - 1)\left(t + \frac{1}{2}t^2 + \frac{1}{3}t^3\right),$$

then exit with $W = az^3$, else if

$$\frac{1}{2}x^2 + E > \frac{1}{2}x_0^2 + a(Z^3 - z^3) + (3a - 1)\log(Z/z_0),$$

then exit with $W = az^3$, else go back to Step 1.

In this algorithm, $x_0 = s - \sqrt{3}$, $z_0 = 1 - s\sqrt{3}$, $t = 1 - z_0/s$.

Figure 1 shows the three functions $h(x) \leq g(x) \leq f(x)$. Except for scale, $f$ is the standard normal density, $g$ is the density of a random variable $X$ on which a simple cubic transformation will produce the required gamma variate, and $h$ is chosen as a function bounded by $g$ but easier to evaluate.

**Remark 1.** If a random point $(X, Y)$ is uniformly distributed in the area under $g$ then $W = a(sX + 1 - s^3)$ has the gamma density $we^{-1/2r(a)}$. This is the Wilson-Hilferty transformation[10]. Again with $z = sx + 1 - s^3$, the formula for $g$ is

$$g(x) = e^{-1/2z^2 - a(z^3 - z_0^3) + (3a - 1)\log(z/z_0)} \text{ for } x > s - 1/s, \text{ else } g = 0.$$

**Remark 2.** A point $(X, Y)$ uniformly distributed under $g$ may be obtained by choosing points $(X, Y)$ uniformly distributed under $f$ until we get one that is under $g$. This is the rejection technique. The efficiency, (area under $g$)/(area under $f$), is 90% for $a = 1$, 98% for $a = 2$ and quickly goes to 1. Even for $a = 1/3$ the efficiency is 49%.

**Remark 3.** If $X$ is a standard normal random variable and $Y = U \cdot f(X)$ with $U$ uniform on $(0, 1)$, independent of $X$, then the point $(X, Y)$ is uniform under $f$; in our case $f(x) = e^{-1/2x^2}$ for $-\infty < x < \infty$.

**Remark 4.** Because of the exponential nature of tests to determine whether the point $(X, Y)$ lies under $g$, it is better to generate the ordinate of the random point $(X, Y)$ in the form $Y = f(X) e^{-E}$, where $E$ is a standard exponential random variable. When $Y$ is so generated the test $Y < g(X)$ becomes the second test in Step 3 of the algorithm.

**Remark 5.** The first test in Step 3 of the algorithm is inserted to avoid, most of the time, the need to calculate $\log(Z/z_0)$. In the graph of Fig. 1, if the point $(X, Y)$ is under $h$ then it must be under $g$. On letting $t = 1 - z_0/s$, the formula for $h$ is

$$h(x) = e^{-1/2z^2 - a(z^3 - z_0^3) + (3a - 1)\log(z/z_0)} \text{ for } x > s - 1/s, \text{ else } h = 0.$$

**Remark 6.** We leave to the reader the task of verifying that $h(x) \leq g(x) \leq f(x)$ for all $x$, and hence that the procedure produces variates $W$ with exact gamma densities; the only compromise is, of course, the limited word size of the computer.

### 3. IMPLEMENTATION

For the convenience of some readers we provide a Fortran function subroutine which will generate a random gamma variate with parameter $A > 1/3$.

```
FUNCTION RGAMA(A)
DATA B/1./
IF(B.EQ.A) GO TO 1
```
B = A
S = .3333333/SQRT(A)
ZO = 1. - 1.732051 * S
CC = A * ZO * * 3 - 0.5 * (S - 1.732051) * * 2
CS = 1. - S * S

1  X = RNOR(0)
Z = S * X + CS
IF(Z.LE.0.) GO TO 1
RGAMA = A * Z * * 3
E = REXP(0)
CD = E + .5 * X * * 2 - RGAMA + CC
T = 1. - ZO/Z
IF(CD + CL * T * (1. + T * (.5 + .333333 * T)) .GT. 0.) RETURN
IF(CD + CL * ALOG(Z/ZO) .LT. 0.) GO TO 1
RETURN
END

Note that this subroutine calls RNOR and REXP. These are the names in the McGill Random Number Package "Super-Duper"[9] for subroutines that produce standard normal and exponential random variables, using what are believed to be the fastest exact methods, described in [3-5] but with improvements developed in the intervening years.

4. COMPARISONS

To provide meaningful comparisons, methods should be tried on the same machine, using the same subroutines for uniform, normal and exponential variates as well as square root, logarithm, and so on. We have seen only two published methods that merit serious attention—the 'optimal algorithm GO' of Ahrens and Dieter[11] and one by Greenwood[2].

Here is a table comparing the speed of those two methods with the above Fortran program. Times are reported in µs.

<table>
<thead>
<tr>
<th>Method</th>
<th>a = 1.000</th>
<th>a = 2.533</th>
<th>a = 4</th>
<th>a = 10</th>
<th>a = 100</th>
<th>a = 1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Squeeze method</td>
<td>210</td>
<td>197</td>
<td>191</td>
<td>187</td>
<td>185</td>
<td>183</td>
</tr>
</tbody>
</table>

Timing was based on 100,000 calls to a Fortran subroutine—the one given above for the squeeze method, the one listed by Greenwood, and a translation to Fortran of the steps in Ahrens and Dieter's Optimized Algorithm GO.

The Greenwood subroutine is also based on the Wilson-Hilferty transformation and rejection, so it appears that the squeeze method saves some 40-50 µs over the straight rejection technique. The Greenwood program provides for all a > 0 as well as handling for the special cases 2a = 1, 2, 3, 4, but small values of the parameter need a different method that requires separate comparison.

The Ahrens-Dieter algorithm only works for a > 2.533 but their article also describes a separate method for small  a.

Acknowledgements—It is a pleasure to acknowledge the support of those who have assisted in the development of the McGill Random Number Package, notably V. S. Alagar, Kasi Ananthanarayanan, Melvin Cohen, Lalit Kumar and N. J. Paul.

REFERENCES


