AD-A060 348 SOUTHERN METHODIST UNIV DALLAS TX DEPT OF OPERATIONS--ETC F/G 12/1
GENERATION OF VARIATES FROM DISTRIBUTION TAILS.(U)
JUL 78 B W SCHMEISER N00014-77-C-0425
UNCLASSIFIED OREM-78008 NL

1 OF 1
AD
A060348

END
DATE
FILMED
12-78
DDC

Technical Report OREM-78008

GENERATION OF VARIATES FROM DISTRIBUTION TAILS

Bruce W. Schmeiser

Department of Operations Research and Engineering Management
Southern Methodist University
Dallas, Texas 75275

12p.

July 1978

This research was supported by Office of Naval Research
Contract N00014-77-C-0425

ABSTRACT

The general acceptance/rejection algorithm for generating random values on a computer is specialized for distribution tails, using an exponential majorizing function and a linear minorizing function. Specific algorithms are given for the normal, gamma, Weibull and beta distributions. While the algorithms can be used alone, it is anticipated that their major value will be to serve as components of algorithms for complete distributions.

Key Words and Phrases: Random number generation

CR Catagories: 5.5

1. INTRODUCTION

Distribution tails are treated differently than the rest of the distribution in many random variate generation algorithms. For example, Kinderman and Ramage [5] use Marsaglia's [6] algorithm, as modified by Ahrens and Dieter [1], for the tail of the normal distribution. In the context of the general acceptance/rejection algorithm, Ahrens and Dieter [2] use a normal majorizing function for the body of the gamma distribution and an exponential majorizing function for the tails. Schmeiser and Shalaby [7] use a triangular majorizing function for the tails of the beta distribution.

In addition to use as part of other algorithms, tail generators are sometimes used directly. The most straightforward method of generating from the tail of a distribution is to generate a value x from the complete distribution and use x only if x lies in the desired tail, but this method is inefficient for small tail areas. A second method is available when the inverse cumulative distribution function transformation is used; i.e. $x = F^{-1}(u)$ where $u \sim U(0,1)$. If values are restricted to $x \geq a$, the transformation $x = F^{-1}(u^* + u(1 - u^*))$ is used, where $u^* = F(a)$. Likewise if values are restricted to $x \leq a$, the transformation $x = F^{-1}(uu^*)$ is used, where again $u^* = F(a)$. Unfortunately, such common distributions as the normal, gamma, and beta do not have closed form inverse cumulative distribution functions.

A family of algorithms is developed below to generate variates from distribution tails. The algorithms may be used independently or as part of a complete distribution generator. In Section 2, the general acceptance/rejection algorithm is specialized to use an exponential majorizing function to generate values from the tails of distributions. In Section 3, the normal, gamma, Weibull and beta algorithms are given. The algorithms are discussed in Section 4.

2. THE EXPONENTIAL TAIL ALGORITHM

Let $f(x)$ be the density function of the distribution from which tail values are to be generated. If $t(x)$ is a majorizing function of $f(x)$ ($t(x) \geq f(x)$ for all x), values from $f(x)$ may be generated by the general acceptance/rejection algorithm:

1. Generate x having density $r(x)$ proportional to $t(x)$.

2. Generate $v \sim U(0,1)$.

3. If $v \leq f(x)/t(x)$, deliver x. Otherwise go to step 1.

In this section, this algorithm is specialized to generate only tail values.

Given an appropriate density $f(x)$, the figure illustrates the algorithm developed here for the right tail. The left tail algorithm, discussed later, is quite similar.

Figure About Here

To ensure the algorithm is valid, some assumptions on $f(x)$ are necessary. Assume that $f(x)$ is convex for $x > a$, which is true of "bell-shaped" distributions if the point a is outside the points of inflection. Further assume the tails of $f(x)$ are lighter than the exponential tail. This is true for the normal, beta, gamma, and Weibull distributions, but is not true for the lognormal or Cauchy.

An exponential majorizing function $t(x) = f(a) \exp[-\lambda(x - a)]$ is made tangent to $f(x)$ at the point a by setting $\lambda = -f'(a)/f(a)$. Since $t(x)$ is proportional to the exponential density, $r(x) = \lambda e^{-\lambda(x-a)}$, x may be generated in step 1 using the inverse transformation

$$x = a - \ell n(u)/\lambda$$

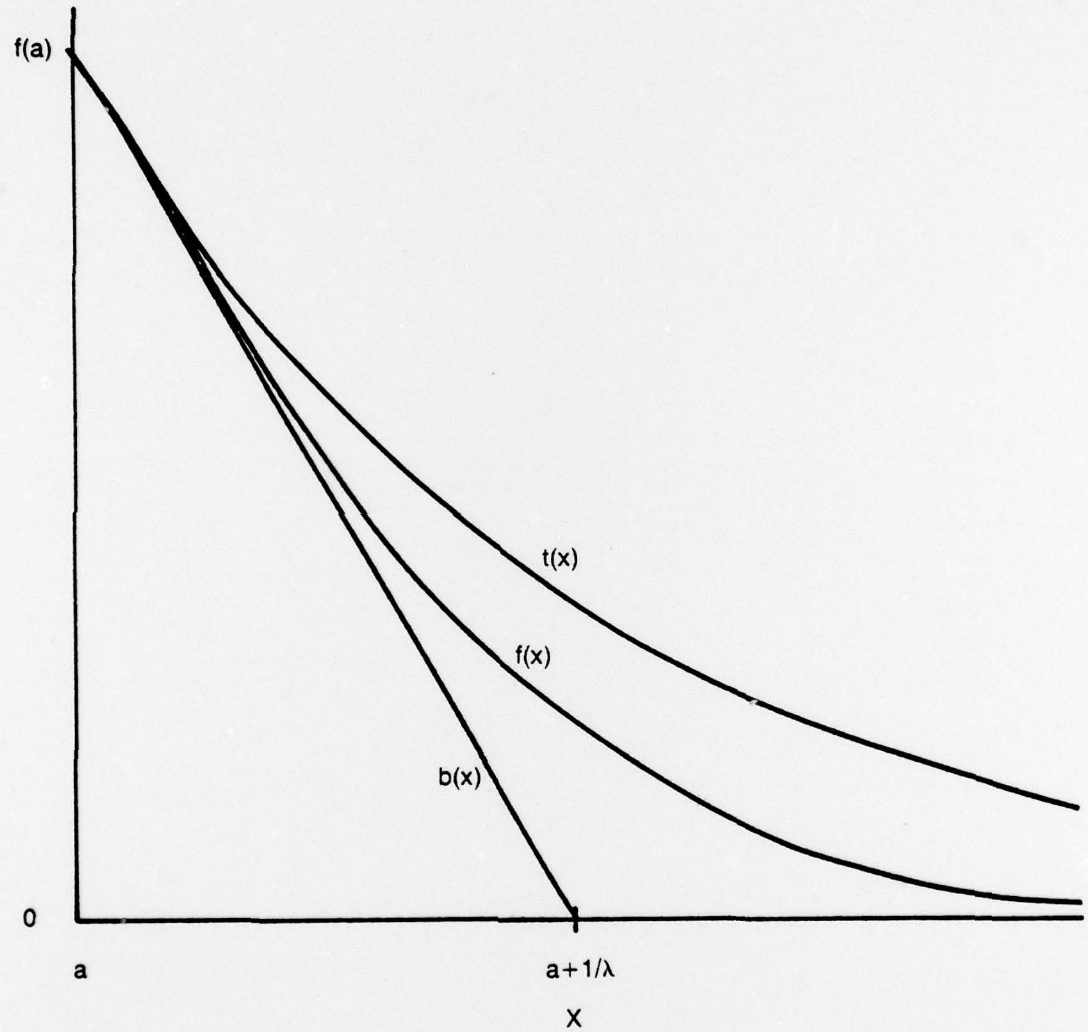

Figure. The relationship of the majorizing function $t(x)$ and the minorizing function $b(x)$ to the density function $f(x)$.

Commonly step 3 is implemented using logarithms, so here step 3 becomes

$$\text{If } \ln v \leq \ln[f(x)/t(x)]$$

or

$$\ln v \leq \ln(f(x)/f(a)) + \lambda(x - a).$$

For many distributions $f(x)/f(a)$ simplifies nicely. However step 3 still requires at least one, and commonly more, exponential level operations. Therefore step 3 is modified to include a preliminary comparison of v with $b(x)/t(x)$ before comparing v with $f(x)/t(x)$, where $b(x) = f(a)[\lambda(a-x)+1]$ is the linear tangent to $t(x)$ and $f(x)$ at a. Since both $t(x)$ and $f(x)$ are convex for $x \geq a$, $b(x) \leq t(x)$ and $b(x) \leq f(x)$ for all $x \geq a$. After cancelling $f(a)$, the preliminary comparison is

$$v \leq (\lambda[a - x] + 1)/\exp[-\lambda(x - a)], \qquad (1)$$

which simplifies to

$$v \leq [\lambda(a - x) + 1]/u$$

after substituting $x = a - \ln(u)/\lambda$ into equation (1).

Allowing the steps to be renumbered, the exponential tail algorithm for a convex density $f(x)$ over the interval $x \geq a$ is as follows:

1. Generate $u, v \sim U(0,1)$
2. $x = a - \ln(u)/\lambda$
3. If $v \leq [\lambda(a - x) + 1]/u$, deliver x.
4. If $\ln v \leq \ln[f(x)/f(a)] + \lambda(x - a)$, deliver x. Otherwise go to step 1.

The algorithm coding is unchanged for generating values from the left tail. Assume $x \leq a$ and $f(x)$ is convex over this region. Then $\lambda = -f'(a)/f(a)$ will now be negative. This change in sign for λ correctly changes each step of the algorithm. Thus negative values of λ yield the left tail and positive values of λ yield the right tail automatically.

A problem does arise when a negative value of x is generated in step 1 for distributions where $f(x)$ is not defined for negative x. In this case step 1b can be added:

1b. $u = e^{a\lambda} + u(1 - e^{a\lambda})$

where $e^{a\lambda}$ is a constant evaluated only one time. Equivalently, step 2b may be added:

2b. If $x < 0$, go to step 1.

For either the left or right tail, the area under $t(x)$ is of interest when generating values from the entire distribution. The probability of needing a tail value for a particular variate is the ratio of the area under the tail majorizing function to the area under the majorizing function for the entire distribution. For the exponential tail algorithm discussed here, tail area is $|f(a)/\lambda|$ for either tail.

3. SPECIALIZATION TO SOME COMMON DISTRIBUTIONS

Specialization to the normal, gamma, Weibull, and beta distributions is now straightforward. Let $K(x)$ denote $\ell n[f(x)/f(a)]$. Scaling constants in $f(x)$ are of no consequence here and are ignored.

<u>Normal</u> $\quad f(x) = \exp(-x^2/2) \qquad -\infty < x < \infty$

$|a| \geq 1$

$\lambda = a$

$K(x) = -(x - a)^2/a.$

<u>Gamma</u> $\quad f(x) = x^{\alpha-1}\exp(-x) \qquad 0 \leq x,\ a \geq 1$

If $\alpha \leq 2$, there is no left tail. Otherwise

$0 < a \leq (\alpha - 1) - (\alpha - 1)^{1/2}$ or $(\alpha - 1) + (\alpha - 1)^{1/2} \leq a$

$\lambda = 1 - [(\alpha - 1)/a]$

$K(x) = (\alpha - 1)\ell n(x/a) + a - x$

<u>Weibull</u> $\quad f(x) = x^{c-1}\exp[-x^c] \qquad 0 \leq x,\ c \geq 1$

If $c \leq 2$ there is no left tail. Otherwise

$0 < a < \{[(c - 1) - (c - 1)^{1/2}]/c\}^{1/c}$ or

$\{[(c - 1) + (c - 1)^{1/2}]/c\}^{1/c} < a$

$\lambda = ca^{c-1} - (c - 1)/a$

$K(x) = (c - 1)[\ell n(x) - \ell n(a)] + a^c - \exp[c\ \ell n(x)]$

<u>Beta</u> $\quad f(x) = x^P(1 - x)^Q \qquad 0 \leq x \leq 1,\ P > 0,\ Q > 0$

If $P + Q \leq 1$, there are no tails.

If $P + Q > 1$, then let $D = (PQ/(P + Q - 1))^{1/2}$

If $P - D \leq 0$, there is no left tail, and if $Q - D \leq 0$, there is no right tail.

Otherwise, $0 < a \leq (P - D)/(P + Q)$ or $(P + D)/(P + Q) \leq a < 1$.

$\lambda = Q/(1 - a) - P/a$

$K(x) = P\ \ell n(x/a) + Q\ \ell n((1 - x)/(1 - a))$

4. ANALYSIS OF THE EXPONENTIAL TAIL ALGORITHM

A result which simplifies the analysis of the algorithms is that the linear comparison of step 3 will be true with probability .5 for any $f(x)$ and for any a. This is easy to verify, since the area under $b(x)$ is $|f(a)/(2\lambda)|$ and the area under $t(x)$ is $|f(a)/\lambda|$.

Since the exponential level operations dominate the computation times on most computers, crude algorithm comparisons can be made by counting the expected number of such operations. Thus the exponential tail algorithm requires $n = 1 + 1 + (m/2)$ expected logarithms where m is the number of logarithms required to evaluate $K(x)$. The four distributions discussed in Section 3 result in $m = 0, 1, 2$, and 2 respectively. (The two exponential operations in the Weibull algorithm are $\ell n(x)$ and $\exp(\cdot)$). The expected total operations are then $n = 2, 2.5, 3$, and 3, respectively.

The other factor in determining execution time is the expected number of iterations per variate generated, which is $[f(a)/\lambda]/\int_a^\infty f(x)dx$ for the right tail and $[-f(a)/\lambda]/\int_{-\infty}^a f(x)dx$ for the left tail. For the four distributions considered in Section 3 the expected number of iterations is relatively constant for all values of a, due to all four distributions having tails not unlike the exponential.

It is interesting to compare Marsaglia's [6] normal tail generator as modified by Ahrens and Dieter [1] to the algorithm of this paper. Comparison was performed on the SMU CDC Cyber 72 Computer under the KRONOS operating system and using the FTN Fortran compiler with its internal source of U(0,1) values RANF. Both are easy to implement, with Marsaglia's algorithm requiring seven lines of code and the algorithm of this paper requiring six lines.

The algorithms ran with almost identical times, ranging from .45 milliseconds per variate for a = 1 to .33 milliseconds for a = 5.5. The ratio of uniform dev iates method, given by Kinderman and Monahan [4], yields a Normal tail algorithm, described in Kinderman and Monahan [3], which generates variates faster but which requires a one-time initialization of constants.

REFERENCES

1. Ahrens, J. H. and Dieter, U. Computer methods for sampling from the exponential and normal distributions. Communications of the ACM, 15, 10(1972), 873-882.

2. Ahrens, J. H. and Dieter, U. Computer methods for sampling from the gamma, beta, Poisson, and binomial distributions. Computing, 12 (1974), 223-246.

3. Kinderman, A. J. and Monahan, J. F. Generating random variables from the ratio of two uniform deviates. Computer Science and Statistics: Ninth Annual Symposium on the Interface, 1976.

4. Kinderman, A. J. and Monahan, J. F. Computer generation of random variables using the ratio of uniform deviates. ACM Transactions on Mathematical Software, 3, 3(1977), 257-260.

5. Kinderman, A. J. and Ramage, J. G. Computer generation of normal random variables. Journal of the American Statistical Association, 71 (1976), 893-896.

6. Marsaglia, G. Generating a variable from the tail of the normal distribution. Technometrics, 6, 1(1964), 101-102.

7. Schmeiser, B. W. and Shalaby, M. Rejection methods for beta variate generation, Technical Report OREM 77014, Southern Methodist University, Dallas, Texas 75275. (Tentatively accepted for publication in Journal of the American Statistical Association.)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER OREM 78008	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) GENERATION OF VARIATES FROM DISTRIBUTION TAILS		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Bruce W. Schmeiser		8. CONTRACT OR GRANT NUMBER(s) ONR N00014-77-C-0425
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research and Engineering Management Southern Methodist University, Dallas, TX 75275		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 277-236
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Analysis Program Office of Naval Research Arlington, VA 22217		12. REPORT DATE July 1978
		13. NUMBER OF PAGES 11
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Random number generation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The general acceptance/rejection algorithm for generating random values on a computer is specialized for distribution tails, using an exponential majorizing function and a linear minorizing function. Specific algorithms are given for the normal, gamma, Weibull and beta distributions. While the algorithms can be used alone, it is anticipated that their major value will be to serve as components of algorithms for complete distributions.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)