G. BARBU

Higher Education Institute
Pitești

A New Fast Method for Computer Generation of Gamma and Beta Random Variables by Transformation of Uniform Variables

One of fastest methods for computer generation of random variables, consists of reducing the problem to generating of the most simple random variables, which are uniform random variables.

The generation method is based on a rejection-type procedure [4] *using uniform random vectors over a bounded domain* $D \subset R^2$, *of the form:* $D = \{(u, v); \gamma(u, v) \leqslant 0\}$, *properly choosen for every kind of a random variable.*

In this paper a lemma is fundamental, which connects uniform distribution over a bounded domain, with the random variable which is to be generated by the computer.

Based on this lemma, there are built-up algorithms for computer generation of sampling values for Gamma and Beta random variables.

Finally, results of testing algorithms by the computer are presented and comparisons with other known generation methods are mentioned.

1. A FUNDAMENTAL LEMMA

Lemma. Let X be a random variable whose p.d.f. is

$$f(x) = \frac{1}{K}\, g(x), \quad x \in R,\ g(x) \geqslant 0$$

where

$$K = \int_R g(x)\ \mathrm{d}x$$

Let (U, V) be a random vector, uniformly distributed over the domain $D = \left\{(u, v);\ 0 \leqslant u \leqslant g^{1/2}\left(\frac{u-v}{u}\right)\right\}$.

We assume that the domain D is bounded. Then, the probability density function of $X = \frac{U-V}{U}$ is $f(x)$.

Proof. The probability density function of (U, V) is

$$\mathcal{C}(u, v) = \begin{cases} \dfrac{1}{mes\ D}, & (u, v) \in D \\ 0, & \text{otherwise} \end{cases}$$

where

$$mes\ D = \iint_D \mathrm{d}u\ \mathrm{d}v = \iint\limits_{0 \leqslant u \leqslant g^{1/2}\left(\frac{u-v}{u}\right)} \mathrm{d}u\ \mathrm{d}v$$

Let us use the transform of coordinates:

$$\begin{cases} \dfrac{u-v}{u} = x \\ u = t \end{cases} \quad \text{or} \quad \begin{cases} u = t \\ v = t - tx \end{cases}$$

whose Jaccobian is $-t$ and $\mathrm{d}u\ \mathrm{d}v = t\ \mathrm{d}t\ \mathrm{d}x$.

Then,

$$mes\ D = \iint\limits_{0 \leqslant t \leqslant g^{1/2}(x)} t\ \mathrm{d}t\ \mathrm{d}x = \int_{-\infty}^{+\infty} \int^{g^{1/2}(x)} t\ \mathrm{d}t\ \mathrm{d}x = \frac{1}{2}\int_{-\infty}^{+\infty} g(x)\ \mathrm{d}x = \frac{K}{2}.$$

The cumulative distribution function of $X = \frac{U-V}{U}$ is

$$P(X < x) = P\left(\frac{U-V}{U} < x\right) = \frac{1}{mes\ D} \iint\limits_{\frac{u-v}{u} < x} \mathrm{d}u\ \mathrm{d}x.$$

Using the above transform of coordinates, we have:

$$P(X < x) = \frac{2}{K}\int_{-\infty}^{x} \int_0^{g^{1/2}(x)} t\,\mathrm{d}t\,\mathrm{d}z = \frac{1}{K}\int_{-\infty}^{x} g(z)\,\mathrm{d}z = F(x)$$

where $F(x)$ is the cumulative distribution function corresponding to $f(x)$. The lemma is proved.

2. THE GENERATION ALGORITHM

The previous lemma leads to the following generation algorithm for X.

Step 0. Initialize the RND (algorithm for generating an uniform random number over (0,1)).

Determine a minimal interval (see Fig. 1).

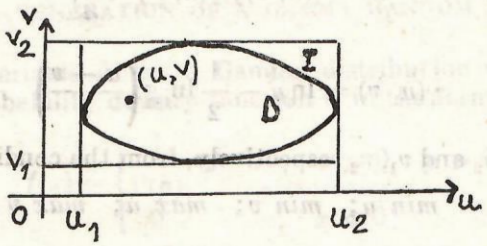


Figure 1

$$I = [u_1,\ u_2] \times [v_1, v_2]$$

such as $D \subset I$, $(u_1 \geqslant 0)$.

Step 1. Generate $(U^*,\ V^*)$ uniformly distributed over I by formula: $U^* = u_1 + (u_2 - u_1) \cdot U'$

$$V^* = v_1 + (v_2 - v_1) \cdot V'$$

where U' and V' are independent random numbers over (0,1).

Step 2. If $(U^*,\ V^*) \notin D$ reject the point $(U^*,\ V^*)$ (which is not on the domain D) and go to Step 1. (Otherwise go to Step 3).

Step 3. Take $X = \dfrac{U^* - V^*}{U^*}$ · STOP.

Remarks.

1. The performance of the algorithm is characterized by the accepting probability (the probability to pass from Step 2 to Step 3) given by:

$$p_a = \frac{mes\ D}{mes\ I} \tag{1}$$

which according to the proof of lemma is

$$p_a = \frac{K}{2(u_2 - u_1) \cdot (v_2 - v_1)} \cdot \tag{1'}$$

The algorithm has a good performance for a large value of p_a (i.e. when p_a is close to one).

2. The points $(U^*,\ V^*)$ are uniformly distributed over the bounded domain D.

3. The problem which must be solved for generating a random variable by the above algorithm is to determine the minimal interval: $I = [u_1, u_2] \times [v_1, v_2]$, $D \subset I$, with I bounded (i.e. $u_2 - u_1 < \infty$, $v_2 - v_1 < \infty$.)

We shall present two equivalent methods to determine the interval I:

First we write the domain D in the form:

The method of Lagrange's multipliers

$$D = \{(u, v);\ \gamma(u, v) \leqslant 0\} \tag{2}$$

where

$$\gamma(u, v) = \ln u - \frac{1}{2} \ln g\left(\frac{u - v}{v}\right) \tag{2'}$$

and calculate u_1, u_2 and v_1, v_2 respectively, from the conditions:

$$min\ u;\quad min\ v;\quad max\ u;\quad max\ v$$

with the constraint $\gamma(u, v) = 0$.

By applying the method of Lagrange's multipliers, it results that u_1, u_2 and v_1, v_2 will be determined respectively from the conditions:

$$\min_D F_1(u, v), \quad \min_D F_2(u, v) \tag{3}$$

$$\max_D F_1(u, v), \quad \max_D F_2(u, v)$$

where

$$F_1(u, v) = u + \lambda\gamma(u, v) \tag{3'}$$

$$F_2(u, v) = v + \lambda\gamma(u, v)$$

with the constraint $\gamma(u, v) = 0$ and the Hessian of F_1 and F_2 must be positive definite.

The method of tangent

We write the domain D in the form:

$$D = \{(u, v);\ \gamma(u, v) \leqslant 0\}$$

and we find the tangent points of the curve $\gamma(u, v) = 0$ parallel to the line $u = \text{const.}$, respectively $v = \text{const.}$

Calculate

$$d\gamma(u, v) = \frac{\partial\gamma(u, v)}{\partial u}\, du + \frac{\partial\gamma(u, v)}{\partial v}\, dv = 0 \tag{4}$$

hence

$$m(u, v) = \frac{dv}{du} = -\frac{\dfrac{\partial\gamma(u, v)}{\partial u}}{\dfrac{\partial\gamma(u, v)}{\partial v}}. \tag{5}$$

Determine u_1, u_2 from the condition $\frac{dv}{du} \to \infty$ and $\gamma(u, v)=0$.

Determine v_1, v_2 from the condition $\frac{dv}{du} \to 0$ and $\gamma(u, v) = 0$.

3. COMPUTER GENERATION OF A GAMMA RANDOM VARIABLE

A random variable X has a Gamma distribution with the parameter $p(p > 0)$ if its probability density function is in the form:

$$f(x) = \begin{cases} \frac{1}{\Gamma(p)} x^{p-1}e^{-x}, & x > 0, \quad p > 0 \\ 0 \quad , & x \leqslant 0. \end{cases}$$

The case when $p > 1$

By applying the lemma, we find:

$$D = \left\{(u, v);\ 0 \leqslant u \leqslant \left(\frac{u-v}{u}\right)^{\left(\frac{p-1}{2}\right)} e^{-\frac{1}{2}\frac{u-v}{u}}\right\} \tag{8}$$

and

$$\gamma(u, v) = \ln u - \frac{p-1}{2} \ln \frac{u-v}{u} + \frac{1}{2} \cdot \frac{u-v}{u}. \tag{9}$$

By applying the method of tangent, we obtain the bounds of the interval I, as:

$$\begin{aligned} u_1 &= 0, \quad u_2 = (p-1)^{\frac{p-1}{2}} \cdot e^{\frac{p-1}{2}} \\ v_1 &= r_1(1-r_1)^{\frac{p-1}{2}} e^{-\frac{1}{2}(1-r_1)} \\ v_2 &= r_2(1-r_2)^{\frac{p-1}{2}} e^{-\frac{1}{2}(1-r_2)} \end{aligned} \tag{10}$$

where r_1, r_2 are the roots of the equation

$$r^2 + pr - 2 = 0.$$

With u_1, u_2 and v_1, v_2 determinated above it results the following algorithm for generating a Gamma random variable:

Step 0. Initialize the RND algorithm for generating uniform random numbers over (0,1).
Input $u_1 = 0$, $p > 1$.
Calculate u_2, v_1, v_2 according to the formulae (10).

Step 1. Generate U by RND and take $U^* = u_2 \cdot U$(U^* is an uniform random number over $(0, u_2)$).
Generate a new U by RND and calculate:
$V^* = v_1 + (v_2 - v_1) \cdot U$ (V^* is an uniform random number over (v_1, v_2)).

Step 2. Calculate $R = \dfrac{U^* - V^*}{U^*}$.

If $\ln U^* - \dfrac{p-1}{2} R + \dfrac{1}{2} R > 0$ go to Step 1, otherwise go to Step 3.

Step 3. $X = R$ (The generated value). STOP.

The accepting probability of this algorithm is:

$$p_a = \frac{(\sqrt{2\pi}(p-1)^{p/2} \cdot 2^{\frac{p-1}{2}} \cdot e^{\frac{\sqrt{p^2+8}-p+2}{2}}}{(\sqrt{p^2+8}-p)(p+2-\sqrt{p^2+8})^{\frac{p-1}{2}} e^{\sqrt{p^2+8}} + (\sqrt{p^2+8}+p)(p+2+\sqrt{p^2+8})^{\frac{p-1}{2}}}. \tag{11}$$

We can see that, $\lim\limits_{p\to\infty} p_a = 0$ which means that the proposed algorithm is performant only for values of p not too large.

The case when $0 < p < 1$.

Le us use the transform $x = y^n$ and the probability density function of Gamma random variable becomes:

$$f(y) = \begin{cases} \dfrac{n}{\Gamma(p)} y^{np-1} e^{-y^n}, & y > 0 \\ 0, & y \leqslant 0. \end{cases} \tag{12}$$

By applying the lemma, we obtain

$$D = \left\{(u, v);\ 0 \leqslant u \leqslant \left(\frac{u-v}{u}\right)^{\frac{np-1}{2}} e^{-\frac{1}{2}\left(\frac{u-v}{u}\right)^n}\right\} \tag{13}$$

and

$$\gamma(u, v) = \ln u - \frac{np-1}{2} \ln \frac{u-v}{u} + \frac{1}{2}\left(\frac{u-v}{u}\right)^n. \tag{14}$$

Below we give one particular version of the algorithm, for $n = 2$ and $p > \dfrac{1}{2}$.

By applying the method of Lagrange's multipliers, we finally obtain the limits of the interval I, namely:

$$u_1 = 0,\ u_2 = \left(\frac{2p-1}{2}\right)^{\frac{2p-1}{4}} \cdot e^{-\frac{2p-1}{4}}$$

$$v_1 = r_1(1 - r_1)^{\frac{2p-1}{2}} \cdot e^{-\frac{1}{2}(1-r_1)^2}$$

$$v_2 = r_2(1 - r_2)^{\frac{2p-1}{2}} \cdot e^{-\frac{1}{2}(1-r_2)^2}$$

where r_1, r_2 are the roots less than one of the equation

$$2r^3 - 4r^2 - (2p - 1)r + 2 = 0$$

because, if we denote by

$$f(r) = 2r^3 - 4r^2 - (2p - 1)r + 2$$

it is easy to see that $f(-1) \cdot f(0) < 0$ and $f(0) \cdot f(1) < 0$.

Finally, we have the following algorithm:

Step 0. Initialize the RND algorithm for generating an uniform random number over (0,1).

Input $u_1 = 0$, $\frac{1}{2} < p < 1$.

Calculate u_2, v_1, v_2 according to the formulae (15).

Step 1. Generate U by RND and calculate $U^* = u_2 \cdot U$ (U^* is an uniform random number over $(0, u_2)$).

Generate a new U by RND and calculate $V^* = v_1 + (v_2 - v_1) \cdot U$ (V^* is an uniform random number over (v_1, v_2)).

Step 2. Calculate $R = \frac{U^* - V^*}{U^*}$

if $\ln U^* - \frac{2p-1}{2} \ln R + \frac{1}{2} R^2 > 0$ go to Step 1. Otherwise go to Step 3.

Step 3. $X = R^2$ (the generated value). STOP.

The accepting probability in the Step 2 is

$$p_a = \frac{1}{2} \cdot \frac{\sqrt{2\pi}\, e^{\frac{r_2^1 + r_2^2 - 2r_1 - 2r_2 + 2}{2}} \cdot \left(\frac{2p-1}{2}\right)^{\frac{2p-1}{2}}}{r_1^{\frac{2p-1}{2}} r_2 (1 - r_1)^{\frac{2p-1}{2}} (1 - r_2)} \tag{16}$$

The accepting probability when $\frac{1}{2} < p < 1$ is large, which means that the proposed algorithm is performant, being very fast.

Computer generation of a Beta random variable

A random variable X has a Beta distribution with $(p, q > 0)$ p and q parameters, if its probability density function is of the form:

$$f(x) = \begin{cases} \dfrac{1}{B(p,\ q)}\, x^{p-1}(1-x)^{q-1}, & x \in (0,1),\ p,\ q > 0 \\ 0, & x \notin (0,1) \end{cases}. \tag{17}$$

The case when $p > 1$, $q > 1$.

By applying the lemma, we obtain:

$$D = \left\{(u, v);\ 0 \leqslant u \leqslant \left(\frac{u-v}{u}\right)^{\frac{p-1}{2}} \left(\frac{v}{u}\right)^{\frac{q-1}{2}}\right\}, \tag{18}$$

and

$$\gamma(u, v) = \ln u - \frac{p-1}{2}\ln\frac{u-v}{u} - \frac{q-1}{2}\ln\frac{v}{u}. \tag{19}$$

After calculations, by one of the two methods presented, we find:

$$u_1 = 0,\ u_2 = \frac{(p-1)^{\frac{p-1}{2}}\,(q-1)^{\frac{q-1}{2}}}{(p+q-2)^{\frac{p+q-2}{2}}}$$

$$v_1 = 0,\ v_2 = \frac{(p-1)^{\frac{p-1}{2}}\,(q+1)^{\frac{q+1}{2}}}{(p+q)^{\frac{p+q}{2}}}. \tag{20}$$

The generation algorithm may be written as:

Step 0. Initialize the RND algorithm for generating uniform random numbers over (0,1).
Input $u_1 = 0$, $v_1 = 0$, $p > 1$, $q > 1$.
Calculate u_2, v_2 according to formulae (20).

Step 1. Generate U by RND and calculate $U^* = u_2 \cdot U$(U^* is an uniform random number over $(0, u_2)$).
Generate a new U by RND and calculate $V^* = v_2 \cdot U$(V^* is an uniform random number over $(0, v_2)$).

Step 2. Calculate $R = \dfrac{U^* - V^*}{U^*}$.

If $\ln U^* - \dfrac{p-1}{2}\ln R - \dfrac{q-1}{2}\ln(1-R) > 0$ go to Step 1.

Otherwise go to Step 3.

Step 3. $X = R$. (The generated value). STOP.
The accepting probability is:

$$p_a = \frac{e\sqrt{2\pi(p+q+1)} \cdot (q-1)^{\frac{q-1}{2}} (p+q)^{\frac{q+1}{2}} (p+q-2)^{p+q-2}}{2(p+q-2)^{p+q}(q+1)^{\frac{q+1}{2}}}. \tag{21}$$

It easy to see that $\lim_{p\to\infty} p_a = 0$, $\lim_{q\to\infty} p_a = 0$, which means that the proposed algorithm is performant when p, q are small numbers.

The case when $0 < p < 1$, $q > 1$.

Let us the transform $X = t^n$ and the probability density function of Beta random variable becomes:

$$f(x) = \begin{cases} \frac{n}{B(p,q)} t^{\frac{np-1}{2}} (1-t^n)^{\frac{q-1}{2}} & , t\in(0,1) \\ & p, q > 0 \\ 0 & , t\notin(0,1). \end{cases} \tag{22}$$

By applying the lemma, we obtain:

$$D = \left\{(u,v);\ 0 \leqslant u \leqslant \left(\frac{u-v}{u}\right)^{\frac{np-1}{2}} \left[1-\left(\frac{u-v}{u}\right)^n\right]^{\frac{q-1}{2}}\right. \tag{23}$$

and

$$\gamma(u,v) = \ln u - \frac{np-1}{2} \ln \frac{u-v}{u} - \frac{q-1}{2} \ln\left[1-\left(\frac{u-v}{u}\right)^n\right]. \tag{24}$$

Below we give one particular version of the algorithm, for $n = 2$ and $p > \frac{1}{2}$, $q > 1$.

In this case, the limits of the interval I are given by:

$$u_1 = 0,\ u_2 = r^{\frac{2p-1}{2}} (1-r^2)^{\frac{q-1}{2}}$$

$$v_1 = 0,\ v_2 = \frac{(2p+2q-s)^2(2p+2q+s-2)(s-1)^{\frac{2p-1}{2}}}{(2p+2q-1)^{\frac{2p+2q-1}{2}}} \tag{25}$$

where

$$r = \frac{\sqrt{4p^2+4pq-8p-2q+3}}{2p+2q-3},\quad s = \sqrt{4p^2+4pq-2q-4p+2}$$

$r,\ s\in R$, because, when $0 < p < 1$ and $q > 1$ we find

$$4p^2 + 4pq - 8p - 2q + 3 = (2p - 1)(2p + 2q + 1) + 4(1 - p) > 0$$

$$4p^2 + 4pq - 2q - 4p + 2 = (2p - 1)(2p + 2q) + 3 - 4p > 0$$

We have the following generation algorithm:

Step 0. Initialize the RND algorithm for generating uniform random numbers over (0,1)

Input $u_1 = 0,\ v_1 = 0, \frac{1}{2} < p < 1,\ q > 1.$

Calculate $u_2,\ v_2$ by formulae (25).

Step 1. Generate U with RND and calculate $U^* = u_2 \cdot U$ (U^* is an uniform random number over $(0,\ u_2)$).

Generate a new U with RND and calculate $V^* = v_2 \cdot U$ (V^* is an uniform random number over $(0,\ v_2)$).

Step 2. Calculate $R = \dfrac{U^* - V^*}{U^*}$.

If $\ln U^* - \dfrac{2p-1}{2} \ln R - \dfrac{q-1}{2} \cdot \ln (1 - R^2) > 0$

go to Step 1. Otherwise go to Step 3.

Step 3. $X = R^2$. (The generated value). STOP.

The accepting probability in the Step 2 is given by:

$$p_a = \frac{(p-1)^{\frac{2p+3}{2}} (q-2)^{\frac{2q+3}{2}} 2p+2q-1)^{\frac{2p+2q-1}{2}} (2p+2q-3)q\cdot(s-1)^{\frac{1-2p}{2}} \sqrt{2\pi}\cdot e}{8(4p^2+4pq-8p-2q+3)^{\frac{2p-1}{4}} \cdot (2q^2+2pq-2p-5p+3)^{\frac{q-1}{2}} (2p+2q-s)^2 (2p+2q+s-2)}. \tag{26}$$

The performance of the algorithm is very good for any values of parameters $p,\ q,\ 0 < p < 1,\ q > 1$.

Testing the algorithms. Comparisons

The algorithms were tested on the computer FELIX C-256 in FORTRAN language. For every algorithm it was written a program which calls three subroutines:

— the subroutine for generating uniform random numbers over (0,1).

— the subroutine which calculate the limits of the interval I.

— the subroutine for generating a sampling value.

The test of each algorithm was made as follows: for every distribution it was calculated the mean value in any theoretical variance σ^2 when the parameters are given; it was generated by each algorithm N sampling values $X_1,\ X_2, \ldots,\ X_N$ and there were estimated the mean value and theoretical variance by the arithmetic mean, respectively sampling variance s_x^2 according to formulae:

$$\overline{X} = \frac{1}{N} \sum_{i=1}^{N} X_i,\ s_x^2 = \frac{1}{N} \sum_{i=1}^{n} X_i^2 - \overline{X}^2. \tag{27}$$

Table 1

The generated variable	The mean value and standard deviation (theoretical)	Sampling values N	The mean value and standard deviation (estimated)	The computing time	The relative errors
Gamma $p = 2$	$m = 2.0000$ $\sigma_X = 1.4142$	5000	$\bar{X} = 2.0037$ $s_X = 1.4038$	19″	0.18 % 0.74 %
Gamma $p = 0.95$	$m = 0.9500$ $\sigma_X = 0.9747$	5000	$\bar{X} = 0.9567$ $s_X = 0.9755$	21″	0.7 % 0.09 %
Beta $p = 10$ $q = 3$	$m = 0.7692$ $\sigma_X = 0.1126$	5000	$\bar{X} = 0.7691$ $s_X = 0.1125$	24″	0.01 % 0.09 %
Beta $p = 0.7$, $q = 2$	$m = 0.2500$ $\sigma_X = 0.2261$	5000	$\bar{X} = 0.2506$ $s_X = 0.2242$	28″	0.04 % 0.8 %

In table 1, we can see that for every proposed algorithm the estimated values are close enough to the theoretical values, which means a correct implementation and the validity of the algorithm.

To have an information about the performance of the previous algorithms they were compared with other known algorithms, for the same value of the parameters, as follows:

1. In order to generate a Gamma random variable with parameter p, $(p > 1)$ we used a composition – rejection type procedure, implemented in [5] by a performant subroutine GANEG 1.

2. In order to generate Gamma random variable with parameter p, $(0 < p > 1)$ it was considered the following algorithm [2]:

The GT algorithm $(n = 2,\ p > 0.5)$

Step 0. $a = p - 0.5,\ b = p/a,\ d = c + 2,\ s = \sqrt{p}$

$h_1 = (0.865 + 0.064/p)/s$

$h_2 = (0.4343 - 0.105/s)/s.$

Step 1. Generate U, U_1 uniform random numbers over (0,1). Calculate $U_2 = U_1 + h_1 U - h_2$.
If $0 < U_2 < 1$ go to Step 2. Otherwise go to Step 1.

Step 2. $W = b(U_1/U_2)^2 \cdot$ If $cU_2 - d + W + W^{-1} \leqslant 0$ go to Step 4.

Step 3. If $c \cdot \log U_2 - \log W + W - 1 \geqslant 0$ go to 1. Otherwise go to Step 4.

Step 4. $X = aW$ (The generated value). STOP.

3. In order to generate a Beta random variable with parameters $p > 1$, $q > 1$ it was considered the BERVJ 4 algorithm described in [4] and programed in [5], based also on a rejection — type procedure.

4. In order to generate a Beta random variable with parameters $0 < p < 1$, $q > 1$, it was considered the BERVJ 3 algorithm described in [4] and programed in [5].

The comparing results related to the performance of the method proposed in this paper and the known methods as they were produced by the computer tests are summarized in table 2.

Table 2

The generated value	The computing time for 5000 sampling values	
	The known method	The proposed method
Gamma $p = 20$	56″	36″
Gamma $p = 0.95$	34″	21″
Beta $p = 30$, $q = 30$	2′12″	1′
Beta $p = 0.7$, $q = 2$	31″	26″

From table 2 and the computer test it results that, the proposed algorithms for generating Gamma and Beta random variables are faster and induce smaller generation errors.

This is to be expected for our methods because the generated values are ratios of uniform variables.

Therefore, we can conclude that the proposed algorithms in this paper are better than other known algorithms and, in addition, they have the advantage of an easy implementation.

REFERENCES

1. BARBU, C . : *The Computer Generation Methods of Gamma, Beta, Logistic and Snedecor Random Variables, as Ratio of Uniform Random Variable.* Economic Computation and Economic Cybernetics Studies and Research, no 3, 1982, p. 61—73.
2. CHENG, R. C. H., FEAST, G. M.: *Gamma Variate Generators with Increased Shape Parameter Range.* Comm. ACM, Vol. 23, No. 7(1980) p. 389—394.
3. JOHNSON, N, KOTZ, S.: *Distributions in Statistics: Continuous Univariate Distributions.* John Wiley and Sons, New York, 1972.

4. VĂDUVA, I.: *Simulation Models for Computer Applications.* Technical Publishers, Bucharest, 1977. (Romanian).
5. VĂDUVA, I.: *RAVACE. A FORTRAN Subroutine Library for Computer Generation of Random Variables and Random Vectors.* G. M. D. Mitteilungen No 39, GMD-Bonn, 1976.
6. VĂDUVA, I.: *Computer Generation of Random Vectors Based on Transformation of Uniformly Distributed Vectors.* Seventh Conference on Probability Theory, Brașov, from August 29 to September 4, 1982, (in print).

UNE NOUVELLE MÉTHODE RAPIDE POUR ENGENDRER A L'AIDE DE L'ORDINATEUR DES VARIABLES ALEATOIRES GAMMA ET BETA PAR DES TRANSFORMATIONS DE VARIABLES UNIFORMES

L'une des méthode les plus rapides pour engendrer à l'aide de l'ordinnateur les variables aléatoires consiste dans la réduction du problème à l'engendrement des plus simples variables aléatoires qui sont les variables aléatoires uniformes. La méthode utilise un procédé [4] du type répulsion — utilisant des vecteurs aléatoires uniformes dans un domaine limité $D \subset R^2$ ayant la forme: $D = \{u, v) ; \lambda(u,v) \leqslant 0\}$, choisi pour chaque genre de variable aléatoire.

Essentielle dans cet ouvrage est une lemme qui relie la répartition uniforme d'un domaine limité à une variable aléatoire à engendrer au moyen de l'ordinateur. Ayant pour base cette lemme on a construit des algorithmes pour engendrer à l'aide de l'ordinateur des variables-échantillion pour les variables aléatoires Gamma et Beta.

НОВЫЙ БЫСТРЫЙ МЕТОД ГЕНЕРИРОВАНИЯ С ПОМОЩЬЮ Э.В.М. СЛУЧАЙННЫХ ПЕРЕМЕННЫХ ГАММА И БЕТА ПУТЕМ ПРЕОБРАЗОВАНИЯ РАВНОМЕРНЫХ ПЕРЕМЕННЫХ

Один из быстрейших методов генерирования с помощью Э.В.М. случайных переменных состоит в приведении задачи к генерированию простейших случайных переменных т.е. равномерных случайных переменных. Метод генерирования основывается на способе [4] типа отказа при использовании случайных равномерных векторов в ограниченной области $D \subset R^2$ формы $D = \{(u,v\gamma) ; (u,v) \leqslant 0\}$, выбранном для каждого вида случайной переменной.

Фундаментальной в этой работе является лемма, связывающая равномерное распределение ограниченной области со случаинной переменной, предстоящей быть генерированной с помощью Э.В.М. На основе этой леммы были разработаны алгоритмы для генерирования с помощью Э.В.М. образцовых переменных для случайных переменных Гамма и Бета.

В конце статьи приводятся результаты, полученные алгоритмами проверки с помощью Э.В.М., а также некоторые сравнения с другими известными методами генерирования.