G. BARBU
Higher Education Institute
Pitești

On Computer Generation of Random Variables as Ratio of Uniform Random Variables

In building-up simulation models, as well as in various numerical Monte-Carlo method calculations [2] it is necessary to use the algorithms for computer generations of diferent types of random variables.

The aim of this paper is to built-up the algorithms for computer generation of some types of random variables such as: Cauchy, lognormal, Student, Fisher and Burr by applying a certain property of uniform random vectors over a bounded domain $D \subset R^2$, which is not generally an interval.

1. DESCRIPTION OF THE METHOD

Let (U, V) be a random vector defined on the domain $D \subset R^2$. This vector has an uniform distribution if its probability density function (p.d.f.) is

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

(we assume that the domain D is bounded, therefore mes $D < \infty$).

The following lemma [1] is fundamental:

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

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

where

$$K = \int_R g(x)\, dx.$$

Let (U, V) be a random vector, uniformly distributed over the domain:

$$D = \left\{ (u, v);\quad 0 \leqslant u \leqslant g^{1/2}\left(\frac{v}{u}\right) \right\} \tag{1}$$

(we assume that the domain D is bounded and mes $D < \infty$). Then, the probability density function of $X = \frac{V}{U}$ conditioned by $(U, V) \in D$ is $f(x)$.

The proof of this lemma is given in [3].

The generation algorithm of the random variable X, whose probability density function is $f(x)$, $x \in R$ is the following:

Step 0. Initialize the uniform random number algorithm *R N D* and do other necessary initializations.

Step 1. Generate the random vector (U, V) uniform over the domain

$$D = \left\{ (u, v);\quad 0 \leqslant u \leqslant g^{1/2}\left(\frac{v}{u}\right) \right\}.$$

Step 2. $X = \frac{V}{U}$. (X is the generated value).

The generation of (U, V) uniform over the domain D is made (see [2] p. 138) according to the following algorithm (rejection algorithm):

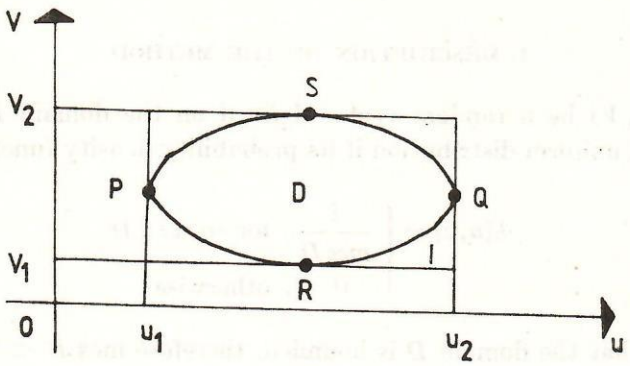


Fig. 1

ALGRES

Step 0. Determine a minimal interval (see Fig. 1) $I = [u_1, u_2] \times [v_1, v_2]$ such as $D \subset I \cdot (u_1 \geqslant 0)$.

Step 1. Generate (U^*, V^*) uniform over I by formulas:

$$U^* = u_1 + (u_2 - u_1) \cdot U'$$
$$V^* = v_1 + (v_2 - v_1) \cdot V'$$

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

Step 2. If $U^* \geqslant g^{1/2}\left(\frac{V^*}{U^*}\right)$ reject the point (U^*, V^*) (which is not on the domain D) and go to Step 1. (Otherwise go to Step 3).

Step 3. $U = U^*$, $V = V^*$ (the point (U, V) will be an uniform point over D) and the generated value is $X = \frac{V}{U}$.

The accepting probability corresponding to the rejection algorithm (i.e. the probability to pass from Step 2 to Step 3) is:

$$p_a = \frac{mes\ D}{mes\ I} = \frac{K}{2(u_2 - u_1)\,(v_2 - v_1)} \tag{2}$$

$\left(\text{because mes } D = \frac{K}{2}\right)$ and the rejection probability is

$$p_r = 1 - p_a.$$

The performance of this algorithm is characterized by the accepting probability p_a; the algorithm has a good performance for a large value of p_a (i.e. when p_a is close to one).

The problem which must be solved when we apply the above lemma for generating a particular random variable is the following:

To determine the minimal interval $I = [u_1, u_2] \times [v_1, v_2]$ such as $D \subset I$ and to see if I is bounded (i.e. $u_2 - u_1 < \infty$, $v_2 - v_1 < \infty$).

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

The first method.

First we write the domain D from (1) as:

$$D = \{(u, v);\ \gamma(u, v) \leqslant 0\} \tag{1'}$$

where

$$\gamma(u, v) = ln\ u - \frac{1}{2} l_n g\left(\frac{v}{u}\right). \tag{3}$$

Then the bounds u_1, u_2 will be determined respectively from the conditions:

$$min\ u \quad \text{and} \quad \max\ u \tag{4}$$

with the condition $\gamma(u, v) = 0$. By applying Lagrange multipliers method it results that u_1, u_2 will be determined respectively from the conditions:

$$F_1(u, v) = min$$
$$F_1(u, v) = max \tag{5}$$

where

$$F_1(u, v) = u + \lambda \gamma(u, v) \tag{5'}$$

with the condition

$$\gamma(u, v) = 0. \tag{5''}$$

Therefore, we determine first the solutions of the system:

$$\frac{\partial F_1(u, v)}{\partial u} = 0, \quad \frac{\partial F_2(u, v)}{\partial v} = 0 \tag{6}$$

obtaining $u = u(\lambda)$, $v = v(\lambda)$ and from (5'') we shall determine the solution λ_i^* $(i \geqslant 1)$ such that $\gamma(u(\lambda^*), v(\lambda^*)) = 0$.

Now, we find $u_1 = \min\limits_i u(\lambda_i^*)$, $u_2 = \max\limits_i u(\lambda_i^*)$

If i=1 the domain degenerates to a curve (*mes* $D=0$) and the lemma cannot be applied.

If $i > 1$ and $u_2 \leqslant 0$ the domain D is also degenerated and the lemma cannot be applied.

If $u_2 = +\infty$, the domain D is unbounded and the lemma again cannot be applied.

In the same manner we determine v_1, v_2 from the conditions:

$$\begin{aligned} F_2(u), v &= min \\ F_2(u, v) &= max \end{aligned} \tag{7}$$

where

$$F_2(u, v) = v + \lambda\gamma(u, v) \tag{7'}$$

with the restriction

$$\gamma(u, v) = 0. \tag{7''}$$

The lemma can be also applied if we have:

$$-\infty < v_1 < v_2 < +\infty.$$

In the above formulas (5'), (7'), λ is the Lagrange's multiplier.

It is necessary to note that we must have $u_1 \geqslant 0$.

The second method.

If we write the domain D in the form (1') with $\gamma(u, v)$ given by (3) and we assume that the curve $\gamma(u, v) = 0$ has the parallel tangents by the coordinates axis, then determining the interval I consists in the following:

— to find the tangent points $P(u_1, v^*)$, $Q(u, v^{**})$ of the curve $\gamma(u, v) = 0$ parallel to the lines $u = \text{const.}$;

— to find the tangent points of curve $\gamma(u, v) = 0$, $R(u^*, v_1)$, $S(u^{**}, v_2)$ parallel to the lines $v = \text{const.}$

Let

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

hence

$$m(u, v) = \frac{dv}{du} = -\frac{\dfrac{\partial\gamma(u, v)}{\partial u}}{\dfrac{\partial\gamma(u, v)}{\partial v.}}. \tag{8}$$

The points P and Q will be determined respectively from the system of equations:

$$\begin{gathered}\lim_{\substack{u \to u_1 \\ v \to v^*}} m(u, v) = \infty \\ \gamma(u_1, v^*) = 0\end{gathered} \tag{9}$$

and

$$\begin{gathered}\lim_{\substack{u \to u_2 \\ v \to v^{**}}} m(u, v) = \infty \\ \gamma(u_2, v^{**}) = 0\end{gathered} \tag{9'}$$

while the points R and S will be determined respectively from the systems:

$$\begin{gathered}\lim_{\substack{u \to u^* \\ v \to v_1}} m(u, v) \\ \gamma(u^*, v_1) = 0\end{gathered} \tag{10}$$

and

$$\begin{gathered}\lim_{\substack{u \to u^{**} \\ v \to v_2}} m(u, v) \\ \gamma(u^{**}, v_2) = 0.\end{gathered} \tag{10'}$$

In the case of the second method we must have also $0 < u_2 < +\infty$, $-\infty < v_1 < v_2 < +\infty$ such as the lemma to be applied.

Taking into consideration the above results, the lemma may be applied for generating the random variable X if the following proposition is satisfied:

Proposition. The domain D is bounded and nondegenerated, if the optimum solutions of problems from the first method or the coordonates u_1, u_2 and v_1, v_2 obtained by the second method satisfy the conditions:

$$0 < u_2 < +\infty$$

$$-\infty < v_1 < v_2 < +\infty.$$

Finally, we note that any described method can be applied to determine the interval I, while the lemma can be applied in the conditions of the above proposition.

2. THE APPLICATIONS OF THE METHOD PRESENTED IN § 1.

Now, we apply the method presented in § 1, to generate various types of random variables which were less studied in the literature.

2.1. *Computer generation of a Cauchy random variable.*

The random variable X has Cauchy distribution if its probability density function is

$$f(x) = \frac{1}{\pi} \cdot \frac{1}{1+x^2}, \quad x \in R. \tag{11}$$

In this case we have:

$$K = \pi, \quad g(x) = \frac{1}{1+x^2}, \quad x \in R \tag{12}$$

and after suitable calculations, we get

$$\gamma(u, v) = \frac{1}{2} ln\ (u^2 + v^2). \tag{13}$$

Now, by applying the first method and the above proposition, we obtain after some calculations:

$$\begin{aligned} u_1 &= 0, \quad u_2 = 1. \\ v_1 &= -1, \ v_2 = 1 \end{aligned} \tag{14}$$

Finally, it results the following algorithm for computer generation of a Cauchy random variable:

THE CAUCHY ALGORITHM

Step 0. Initialize the $R\,N\,D$ (the algorithm for generating uniform random numbers over (0, 1)).
Input: $u_1 = 0$, $u_2 = 1$, $v_1 = -1$, $v_2 = 1$ (in accordance to formulas (14).

Step 1. Generate U with $R\,N\,D$ and take $U^* = U$(U^* is an uniform random number over (0, 1)).
Generate a new U with $R\,N\,D$ and calculate $V^* = v_1 + (v_2 - v_1) \cdot U$($V^*$ is an uniform random number over (−1,1)).

Step 2. If $\frac{1}{2} ln\ (U^{*2} + V^{*2}) > 0$ go to Step 1.

Step 3. $X = \frac{V^*}{U^*}$. (The generated value). S T O P.

The accepting probability in the Step 2 (see [2]) is:

$$p_a = \frac{\pi}{4} = 0{,}7854481$$

$$p_r = 1 - p_a = 0{,}2145519$$

which illustrate a good theoretical performance of the algorithm.

2.2. *Computere generation of a lognormal random variable.*

A random variable is lognormal if $Y = ln\ X$ is a normal random variable, i.e. its probability density function is in the form:

$$f(y) = \frac{1}{\sigma_y \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y - \mu_y}{\sigma_y}\right)^2}$$

where

$$\mu_y = E[Y], \quad \sigma_y^2 = Var\ (Y).$$

The probability density function of X is

$$f(x) = \frac{1}{x\sigma_y \sqrt{2\pi}} e^{-\frac{2}{2\sigma_y^2}(lnx - \mu_y)^2}, \quad x > 0. \tag{15}$$

with the mean value and the variance given by formulae

$$E[X] = e^{\mu_y + \frac{\sigma_y^2}{2}} = \mu_x$$

$$\text{Var}\ (X) = (E[X])^2 (e^{\sigma_y^2} - 1) = \sigma_x^2$$

and we obtain:

$$\begin{aligned} \sigma_y^2 &= ln\left[\frac{\sigma_x^2}{\mu_x^2} + 1\right] \\ \mu_y &= ln\ \mu_x - \frac{1}{2} ln\left[\frac{\sigma_x}{\mu_x^2} + 1\right]. \end{aligned} \tag{15'}$$

Therefore, in accordance with the lemma we have:

$$K = \sigma_y \sqrt{2\pi},\ g(x) = \frac{1}{x} e^{-\frac{1}{2\sigma_y^2}(lnx - \mu_y)^2} \tag{16}$$

and after computations we find the function $\gamma(u, v)$ in the form:

$$\gamma(u, v) = ln\ u + \frac{1}{2} ln\frac{v}{u} + \frac{1}{4\sigma_y^2}\left(ln\frac{v}{u} - \mu_y\right)^2. \tag{17}$$

With the aim to apply the second method we find:

$$d\gamma(u, v) = \frac{1}{u}\, du + \frac{1}{2q}\, \frac{u\, dv - v\, du}{u^2} + \frac{1}{2q\sigma_y^2}\, (ln\ q - \mu_y)\, \frac{u\, dv - v\, du}{u^2}$$

$$q = \frac{v}{u}$$

and the equation $d\gamma = 0$ becomes:

$$1 + \frac{1}{2q}\, \frac{\sigma_y^2 + ln\ q - \mu_y}{\sigma_y^2} \left(\frac{dv}{du} - q \right) = 0. \tag{18}$$

By applying the second method, we finally obtain the limits of the interval I, namely:

$$\begin{aligned} u_1 &= 0, \quad u_2 = e^{\frac{\sigma_y^2 - 2\mu_y}{4}} \\ v_1 &= 0, \quad v_2 = e^{\frac{\sigma_y^2 + \mu_y}{4}} \end{aligned} \tag{18'}$$

With u_1, u_2 and v_1, v_2 determinated above the algorithm for generating a lognormal random variable is the following:

THE LOGNORM ALGORITHM

Step 0. Initialize the *R N D* (algorithm for generating uniform random numbers over (0,1)).
Input μ_x, σ_x, $u_1 = 0$, $v_1 = 0$.
Calculate σ_y^2 and μ_y by formulae (15') and u_2, v_2 by formulae (18').

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

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

$$\text{If } ln\ U^* + \frac{1}{2}\, ln\ R + \frac{(ln\ R - \mu_y)^2}{4\sigma_y^2} > 0 \text{ go to Step 1.}$$

Step 3. $X = R$ (the generated value). S T O P.

The accepting probability in the Step 2 is

$$p_a = \frac{mes\ D}{mes\ I} = \frac{K}{2\, u_2 v_2}$$

$$p_a = \frac{\sigma_y \sqrt{2\pi}}{2\, e^{\frac{\sigma_y^2}{2}}} = \frac{\sqrt{\frac{\pi}{2}}\,\sigma_y}{e^{\frac{\sigma_y^2}{2}}}$$

and we note that if $\sigma_y \to \infty$, then $p_a \to 0$ and, in this case the algorithm is not recomended (But this is not a practical case).

2.3. *Computer generation of a Student random variable.*

Let X be a Student random variable with n degrees of freedom, whose probability density function is given by

$$f(x) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\,\Gamma\left(\frac{n}{2}\right)}\left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}}, \quad x \in R. \tag{20}$$

With the notations of lemma, we have in this case:

$$K = \frac{\sqrt{n\pi}\,\Gamma\left(\frac{n}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right)}, g(x) = \left(1 + \frac{x^2}{n}\right), \quad x \in R \tag{20'}$$

and the function $\gamma(u, v)$ is

$$\gamma(u, v) = ln\, u + \frac{n+1}{4} ln\left[1 + \frac{1}{n} \cdot \left(\frac{v}{u}\right)^2\right]. \tag{20''}$$

By applying the first method and the proposition from § 1, we finally obtain.

$$u_1 = 0, \quad u_2 = 1$$

$$v_1 = -\frac{\sqrt{2\pi}(n-1)^{\frac{n-1}{4}}}{(n+1)^{\frac{n+1}{4}}}, \quad v_2 = \frac{\sqrt{2n}\,(n-1)^{\frac{n-1}{4}}}{(n+1)^{\frac{n+1}{4}}} \tag{21}$$

Therefore, we have the following algorithm for generating a Student random variable:

THE STUDENT ALGORITHM

Step 0. Initialize the R N D algorithm for generating uniform random numbers over (0,1).
Input $n > 1$, $u_1 = 0$, $u_2 = 1$.
Calculate v_1 and v_2 by formulae (21).

Step 1. Generate U with $R\,N\,D$ and take $U^* = U$, (U^* is an uniform random number over (0,1)).
Generate a new U with $R\,N\,D$ and calculate $V^* = v_1 + (v_2 - v_1)\,U$($V^*$ is an uniform random number over (v_1, v_2)).

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

If $ln\,U^* + \dfrac{n+1}{4}\,ln\left[1 + \dfrac{R^2}{n}\right] > 0$ go to Step 1.

Otherwise go to Step 3.

Step 3. $X = R$ (the generated value). S T O P.

The performance of the algorithm is characterized by the accepting probability in Step 2, i.e.

$$p_a = \frac{\sqrt{2\pi}\,(n+1)^{\frac{n+1}{4}}}{8\Gamma\left(\dfrac{n+1}{4}\right)(n-1)^{\frac{n-1}{4}}}. \tag{22}$$

By applying Stirling's formula:

$$(p-1)! = \Gamma(p) = (p-1)^{p-1}\,e^{-(p-1)}\sqrt{2\pi(p-1)}$$

we find that for large values of n the probability p_a becomes

$$p_a = \frac{2^{\frac{n-6}{2}} \cdot e^{\frac{n-1}{2}} \cdot (n+1)^{\frac{n+1}{4}}}{(n-1)^{\frac{3n-1}{4}}} \tag{22'}$$

therefore

$$\lim_{n \to \infty} p_a = 0.$$

Thus, the performance of the algorithm is reduced for a large n. In this case it is known that the Student random variable tends to normal random variable $N(0, 1)$ which may be generated by other good methods.

2.4. *Computer generation of a Fisher random variable.*

A random variable X has a Fischer distribution with n_1, n_2 parameters (n_1, n_2 integers and n_1, $n_2 > 0$) if its probability density function is in the form:

$$f(x) = 2\left(\frac{n_2}{n_1}\right)^{\frac{n_1}{2}} \frac{\Gamma\left(\dfrac{n_1+n_2}{2}\right)}{\Gamma\left(\dfrac{n_1}{2}\right)\Gamma\left(\dfrac{n_2}{2}\right)}\,e^{n_1 x}\left(1 + \frac{n_1}{n_2}\,e^{2x}\right)^{-\frac{n_1+n_2}{2}}. \tag{23}$$

With the notations of lemma, $x \in R$ we have in this case

$$K = \frac{1}{2}\left(\frac{n_2}{n_1}\right)^{\frac{n_1}{2}} \frac{\Gamma\left(\frac{n_1}{2}\right)\Gamma\left(\frac{n_2}{2}\right)}{\Gamma\left(\frac{n_1+n_2}{2}\right)}$$

$$g(x) = e^{n_1 x}\left(1 + \frac{n_1}{n_2} e^{2x}\right)^{-\frac{n_1+n_2}{2}}, \quad x \in R$$

and the function $\gamma(u, v)$ has the form:

$$\gamma(u, v) = ln\, u - \frac{n_1}{2}\frac{v}{u} + \frac{n_1+n_2}{2} \cdot ln\left(1 + \frac{n_1}{n_2} e^{2\frac{v}{u}}\right). \tag{24}$$

By applying the second method (and denoting by $q = \frac{v}{u}$) we first find:

$$m(u, v) = \frac{dv}{du} = \frac{2\, n_2 + 2\, n_1\, e^{2q} + q\, n_1 n_2 - q\, n_1 n_2\, e^{2q}}{n_1 n_2 (1 - e^{2q})} \tag{25}$$

and we finally obtain the bounds of the interval I as

$$u_1 = 0, \; u_2 = \left(1 + \frac{n_1}{n_2}\right)^{-\frac{n_1+n_2}{4}}$$

$$v_1 = \alpha_1\left(1 + \frac{n_1}{n_2} \cdot e^{2\alpha_1}\right)^{-\frac{n_1+n_2}{4}} e^{\frac{n_1}{2}\alpha_1} \tag{26}$$

$$v_2 = \alpha_2\left(1 + \frac{n_1}{n_2} \cdot e^{2\alpha_2}\right)^{-\frac{n_1+n_2}{4}} e^{\frac{n_1}{2}\alpha_2}$$

where α_1 and α_2 are the solutions of the following equation in q:

$$2\, n_2 + 2\, n_1\, e^{2q} + q\, n_2 n_1 - q\, n_1 n_2\, e^{2q} = 0 \tag{27}$$
$$n_1^2 (1 - e^{2q}) > 0.$$

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

THE FISHER ALGORITHM

Step 0. Initialize the R N D algorithm for generating uniform random numbers over (0,1).
Input n_1, n_2, $u_1 = 0$
Calculate: u_2, v_1, v_2 by formulae (26).

Step 1. Generate U with $R\,N\,D$ and calculate $U^* = u_2\,U$(U^* is an uniform random number over $(0, u_2)$).
Generate a new U with $R\,N\,D$ and calculate $V^* = v_1 + + (v_2 - v_1)$. U(V^* is an uniform random number over (v_1, v_2)).

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

If $ln\ \ U^* + \dfrac{n_1+n_2}{4}\,ln\left(1 + \dfrac{n_1}{n_2}\ ^{2R}\right) - \dfrac{n_1}{2}\,R > 0$ go to Step 1.
Otherwise go to Step 3.

Step 3. $X = R$ (the generated value) S T O P.

The accepting probability in the Step 2 is given by

$$p_a = \frac{(n_1+n_2)^{\frac{n_1+n_2}{4}}\,(n_2+n_1\,e^{\alpha_1})^{\frac{n_1+n_2}{4}}\,(n_2+n_1\,e^{2\alpha_2})^{\frac{n_1+n_2}{4}}}{4n_1^{\frac{n_1}{2}}\,n_2^{\frac{n_1+3n_2}{4}}\,e^{\frac{n_1}{2}(\alpha_1+\alpha_2)}\,\Gamma\left(\dfrac{n_1+n_2}{2}\right)}$$

and for n_1, n_2 finite numbers, the algorithm is reasonable good; in the case n_1, $n_2 \to \infty$ (the case which is not frequently met in practice), $p_a \to 0$, which illustrates a reduced performance of the algorithm.

2.5. *Computer generation of a Burr random variable.*

The random variable X has a Burr distribution if its probability density function is

$$f(x) = \frac{k\,c\,x^{c-1}}{(1+x)^{k+1}},\quad x > 0;\quad k,\ c \in R_+. \tag{28}$$

By applying the lemma, we have:

$$K = \frac{1}{kc},\quad g(x) = \frac{x^{c-1}}{(1+x^c)^{k+1}},\quad x > 0 \tag{28'}$$

and the function $\gamma\ (u, v)$ is given by the relation:

$$\gamma(u, v) = ln\ u + \frac{k+1}{2}\,ln\left[1 + \left(\frac{v}{u}\right)^c\right] - \frac{c-1}{2}\,ln\,\frac{v}{u}\cdot \tag{29}$$

With the aim to apply the second method we calculate

$$d\gamma(u, v) = \frac{du}{u} + \frac{k+1}{2}\cdot\frac{cq^{c-1}\,\dfrac{u\,dv - v\,du}{u^2}}{1+q^c} - \frac{c-1}{2}\cdot\frac{\dfrac{u\,dv - vdu}{u^2}}{q} = 0 \tag{29'}$$

where we denoted $\dfrac{v}{u} = q$, and the equation $d\gamma = 0$ may be written in the form:

$$1 + \frac{c(k+1)}{2}\cdot\frac{\dfrac{dv}{du} - q}{1+q^c} - \frac{c-1}{2}\cdot\frac{\dfrac{dv}{du} - q}{q} = 0. \tag{29''}$$

After performing calculation required by the second method we obtain the bounds of the interval I:

$$u_1 = 0, \quad u_2 = \frac{\left(\frac{c-1}{1+kc}\right)^{\frac{c-1}{2c}}}{\left(1+\frac{c-1}{1+kc}\right)^{\frac{k+1}{2}}} \tag{30}$$

$$v_1 = 0, \quad v_2 = \frac{(1+c)^{\frac{c+1}{2c}}\,(kc-1)^{\frac{kc-1}{2c}}}{(kc+c)^{\frac{k+1}{2}}} - \cdot$$

The interval I being determined, we can built-up the following algorithm for generating a Burr random variable:

THE BURR ALGORITHM

Step 0. Initialize the $R\,N\,D$ algorithm for generating uniform random numbers over (0,1).
Input k, c $(c > 1,\ kc > 1)$, $u_1 = 0$, $v_1 = 0$.
Calculate u_2, v_2 by the formulae (30).

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

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

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

Step 3. $X = R$. STOP.

The accepting probability in the Step 2 is given by

$$p_a = \frac{(kc+c)^{k+1}}{2\,kc(c-1)^{\frac{c-1}{2c}}\,(c+1)^{\frac{c+1}{2c}}\,(kc-1)^{\frac{kc-1}{2c}}\,(kc+1)^{\frac{kc+1}{2}}}. \tag{31}$$

We can see that $\lim\limits_{c\to\infty} p_a = 0$, $\lim\limits_{k\to\infty} p_a = 0$, $\lim\limits_{kc\to\infty} p_a = 0$ (these cases are not frequently met in practice) which means a reduced performance of the algorithm for large values of c and k.

3. TESTING THE ALGORITHMS. COMPARISONS

3.1. The algorithms were tested on the computer F E L I X C-256 in F O R T R A N language. For every algorithm it was written a preparation subroutine which (depending of the distribution parameters) determines the particular constants in the corresponding generation algorithm (for instance the bounds u_2, v_1, v_2 of the interval I and the constants which appears in the expression of function $\gamma(u, v)$).

The first test of each algorithm was made as follows: for every distribution it was calculated (excepting Cauchy random variable) the mean value m and theoretical variance σ^2 depending on the distribution parameters (assumed to be known).

Then, there were generated by each algorithm N sampling values $X_1, X_2, ..., X_n$ and there were estimated the mean value and the variance by the arithmetic mean, respectively sampling variance s_x^2 according to formulae:

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

For the four types of random variables (excepting Cauchy random variables, which does not have moments) there were obtained the results in the table 1:

Table 1

No	The generated variable.	The mean value and standard deviation (theoretical)	Sampling values N	The mean value and standard deviation (sample).	The computing time	The relative errors
1.	Lognormal	$m = \mu_x = 1$ $\tau_x = 1$	5000	$\overline{X} = 1.00714$ $s_x = 0.94313$	24″	0.7% 5.7%
2.	Student $n = 3$	$m = 0$ $\sigma = 1.7321$	5000	$\overline{X} = 0.0088$ $s_x = 1.7363$	22″	0.88% 0.24%
3.	Fisher $n_1 = 1,\ n_2 = 3$	$m = -0.4507$ $\sigma = 1.2114$	5000	$\overline{X} = -0.4617$ $s_x = 1.2100$	28″	2.4% 0.11%
4.	Burr $k=2,\ c=2$	$m = 0.7854$ $\sigma = 0.7071$	5000	$\overline{X} = 0.7890$ $s_x = 0.6724$	31″	0.45% 4.8%

Therefore, for every tested algorithm there were obtained estimated values close enough to theoretical values, which means a correct implementation and the validity of the algorithm.

For the Cauchy random variable we used a simple test; it was calculated the probability

$$p = P(-1 \leqslant X \leqslant 1) = \int_{-1}^{1} f(x)\mathrm{d}x = 0.5.$$

By generating N sampling values and denoting by α the values which "fall" in the interval $[-1,1]$ it was estimated p by $\bar{p} = \frac{\alpha}{n}$.

The testing figures are $N = 5000$, $\bar{p} = 0.49540$ which correspond to a relative error of 0.9%.

Therefore the Cauchy algorithm is validated and correctly implementated.

3.2. To have an informations about the algorithms performance, comparing with other known generation methods we made the tests as follows:

For every random variable we considered one good previous known generation method which was separately programed and tested by the computer. These methods were the following:

1°. In order to generate a Cauchy random variable it was considered the inverse method on the following algorithms [2]:

THE CAUCHYI ALGORITHM (CAUCHY-INVERSE).

Step 0. Initialize *R N D*. Input.
Step 1. Generate U uniform over (0.1).
Step 2. $X = \mathrm{tg}\ \pi(U - 0.5)$. S T O P.

2°. In order to generate the lognormal random variable of parameters $\mu_x = E[X]$ and $\sigma_x = \mathrm{Var}\ (X)$ it was considered the following algorithm [2]:

THE LOGNORD ALGORITHM (LOGNORMAL-DIRECT)

Step 0. Initialize *R N D*
Input μ_x, σ_x. Calculate μ_y, σ_y.
Step 1. Generate the variable Z normal $N(0,1)$ (the variable Z is generated with the method based on the central limit theorem, i.e. $Z = U_1 + U_2 + \ldots + U_{12} - 6$, where U_i are independent uniform random variables over (0,1)).
Step 2. Calculate $Y = \mu_y + Z\sigma_y$ (Y is $N(\mu_y, \sigma_y)$).
Step 3. $X = e^Y$. S T O P.

3°. In order to generate the Student random variable with n degrees of freedom we used the method based on its definition, namely [2]:

THE STUDENTD ALGORITHM (STUDENT-DEFINITION)

Step 0. Initialize *R N D*. Input $n>0$.
Step 1. Generate Z normal N (0,1).

Step 2. Generate χ^2 (hi-square with n degrees of freedom) independent of Z. (The generation is made as follows: generate Z_i, $1 \leq i \leq n$ normal $N(0,1)$ independent random variables and independent from Z and calculate $\chi^2_n = \sum_{i=1}^{n} Z_i$)

Step 3. Calculate $X = Z / \sqrt{\frac{\chi^2_n}{n}}$. (By definition X has a Student distribution with n degrees of freedom).

4°. In order to generate a Fisher random variable with n_1, n_2 degrees of freedom we used the method based on the definition of this random variable namely:

THE FISHERD ALGORITHM (FISHER-ALGORITHM)

Step 0. Initialize *R N D*. Input n_1, $n_2 > 0$.

Step 1. Generate $\chi^2_{n_1}$, $\chi^2_{n_2}$ independent χ^2 random variables.

Step 2. Calculate $X = \frac{1}{2} ln \frac{\chi^2_{n_1} \cdot n_2}{\chi^2_{n_2} \cdot n_1}$. STOP.

5°. In order to generate a Burr random variable we used the inverse method, namely:

THE BURRI ALGORITHM (BURR-INVERSE)

Step 0. Initialize *R N D*. Input c, $k > 0$.

Step 1. Generate U uniform over (0,1).

Step 2. Calculate $X = ((1/(1-U))^{1/k} - 1)$. STOP.

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

Table 2

No.	The variable	The computing time for 5000 sampling values	
		The known method	The proposed method
1.	Cauchy	18″	19″
2.	Lognormal $\mu_x = 1 = \tau_x$	36″	24″
3.	Student $n = 3$	1′ 43″	22″
4.	Fisher $n_1 = 1$, $n_2 = 3$	1′ 44″	28″
5.	Burr $c = 2$, $k = 2$	23″	31″

It can be pointed out that the proposed method is sensitively faster for the most part of considered distributions and it is at least as good as the known methods for the Cauchy and Burr random variables.

Furthermore, for any of the considered distributions, the proposed method induces smaller generation errors, because the generated value is the only ratio of uniform variates.

Therefore, we can conclude that the method proposed in this paper is better than other known methods and, in addition, it has the advantage of an easy implementation.

REFERENCES

1. R.C.H. CHENG, G. M. FEAST: *Gamma Variate Generators with Increased Shape Parameter Range.* Comm. ACM, Vol. 23, Nr. 7 (1980) p. 389 — 394.
2. I. VĂDUVA: *Simulation models for Computer applications.* Technical Publishers, Bucharest, 1977.
3. GH. BARBU: *On computer generation of Weibull random variables with parameter greater than one.* The Annals University of Bucharest, 1982 (in print).

SUR L'ENGENDREMENT A L'AIDE DE L'ORDINATEUR DE CERTAINES VARIABLES ALEATOAIRES EN TANT QUE RAPPORTS DE VARIABLES UNIFORMES

Dans la construction des modèles de simulation, ainsi que dans les divers calculs numériques ayant pour base la méthode Monte Carlo [2] il est nécessaire d'utiliser des algorithmes pour engendrer à l'aide de l'ordinateur les différents types de variables aléatoires.

Le présent ouvrage a pour but la construction de certains algorithmes pour engendrer à l'aide de l'ordinateur de certaines variables aléatoires Cauchy, lognormales, Student, Fisher Burr, utilisant une certaine propriété de quelques vecteurs uniformes sur un domaine $D \subset R^2$, domaine qui en général, n'est pas un intervalle.

К ВОПРОСУ ГЕНЕРИРОВАНИЯ С ПОМОЩЬЮ ЭВМ НЕКОТОРЫХ АЛЛЕАТОРНЫХ ПЕРЕМЕННЫХ КАК ОТНОШЕНИЙ РАВНОМЕРНЫХ ПЕРЕМЕННЫХ

При фактическом построении симуляционных моделей, как при различных числовых расчётах, основанных на методе Монте-Карло 2, необходимо использовать алгоритмы для генерирования с помощью ЭВМ различных типов алеаторных переменных.

Цель настоящей работы состоит в построении алгоритмов для генерирования с помощью ЭВМ алеаторных переменных Коши, логнормальных, Студент, Фишер, Бур, используя определённое свойство равномерных алеаторных векторов на множестве $D \subset R^2$, вообще звляющемся интервалом.

21. 12. 1981