E C O N O M I C S   B U L L E T I N

A Simple Gamma Random Number Generator for Arbitrary Shape Parameters

Hisashi Tanizaki
Graduate School of Economics, Kobe University

Abstract

This paper proposes an improved gamma random generator. In the past, a lot of gamma random number generators have been proposed, and depending on a shape parameter (say, alpha) they are roughly classified into two cases: (i) alpha lies on the interval (0,1) and (ii) alpha is greater than 1, where alpha=1 can be included in either case. In addition, Cheng and Feast (1980) extended the gamma random number generator in the case where alpha is greater than 1/n, where n denotes an arbitrary positive number. Taking n as a decreasing function of alpha, in this paper we propose a simple gamma random number generator with shape parameter alpha greater than zero. The proposed algorithm is very simple and shows quite good performance.

This research is partially supported by Japan Society for the romotion of Science, Grant-in-Aid for Scientific Research (C) #18530158, 2006-2009, and Grant-in-Aid for the 21st Century COE Program.
Citation: Tanizaki, Hisashi, (2008) "A Simple Gamma Random Number Generator for Arbitrary Shape Parameters." *Economics Bulletin,* Vol. 3, No. 7 pp. 1-10
Submitted: December 24, 2007. **Accepted:** January 27, 2008.
URL: http://economicsbulletin.vanderbilt.edu/2008/volume3/EB-07C10012A.pdf

1 Introduction

This paper proposes an improved gamma random generator. In the past, a lot of gamma random number generators have been proposed. Generally, the gamma random number generators are classified depending on the shape parameter, say α. Johnk (1964), Berman (1971), Ahrens and Dieter (1974) and Best (1983) proposed the random number generators in the case of $0 < \alpha < 1$. The generator that Ahrens and Dieter (1974) originally proposed using rejection sampling and Best (1983) modified is included in the IMSL library (see `http://www.vni.com/products/imsl/index.html`).

Fishman (1976, 1996), Cheng (1977), Best (1978), Tadikamalla (1978), Cheng and Feast (1979), Schmeiser and Lal (1980), Ahrens and Dieter (1982), Minh (1988) and Marsaglia and Tsang (2001) proposed gamma random number generators provided that $\alpha > 1$. Especially, Schmeiser and Lal (1980) used squeeze method with mixture distribution, which is included in the IMSL library for $\alpha > 1$ but it is complicated in programming (i.e., a lot of lines or steps have to be required in programming). Marsaglia and Tsang (2001) proposed the simple and fast algorithm of gamma random number generation which utilizes a standard normal random number generator, but it is slower than the algorithm in Schmeiser and Lal (1980). In Tadikamalla and Johnson (1981), various gamma random number generators are surveyed in the cases of both $0 < \alpha < 1$ and $\alpha > 1$ (also, see Devroye (1986, pp.401–428) for the gamma density). Thus, depending on α the generators are roughly classified into two cases: $0 < \alpha < 1$ and $\alpha > 1$ ($\alpha = 1$ might be included in either of the two cases).

As mentioned above, Cheng and Feast (1979) proposed a gamma random number generator in the case of $\alpha > 1$, where the ratio-of-uniforms method is utilized for random number generation. Moreover, Cheng and Feast (1980) extended it to the case of the shape parameter greater than $1/n$ for any positive n, and especially they showed two algorithms in the cases of $n = 2, 4$. As an extension of Cheng and Feast (1980), taking n as a decreasing function of α, in this paper we propose a simple and fast algorithm on gamma random number generation for all $\alpha > 0$. The ratio-of-uniforms method is utilized as a sampling method, where the acceptance rate of the proposed generator is almost constant (i.e., about 0.76) for a wide range of α and it takes a minimum value (i.e., 0.5) at $\alpha = 0$. The proposed algorithm is very simple and shows quite a good performance.

2 Gamma Random Number Generator with $\alpha > 0$

The ratio-of-uniforms method, which is a random number generation method, is as follows. Suppose that a bivariate random variable (U_1, U_2) is uniformly distributed over the region determined by the inequality: $0 \le U_1 \le \sqrt{h(U_2/U_1)}$ for any non-negative function $h(x)$, which has to be a bounded region. Then, $X = U_2/U_1$ has a density function $f(x) = h(x)/\int h(x)\,\mathrm{d}x$. Typically, $U_1 \sim U(0, a)$ and $U_2 \sim U(b, c)$ might be taken, where $a = \sup_x \sqrt{h(x)}$, $b = \inf_x x\sqrt{h(x)}$, and $c = \sup_x x\sqrt{h(x)}$. Note that $U(\cdot, \cdot)$ denotes the uniform distribution between two arguments. In particular, we have $b = 0$ when $h(x)$ has positive support. It is easily verified that the acceptance probability is given by $\int h(x)\,\mathrm{d}x/(2a(c - b))$. See Kinderman and Monahan (1977) for the ratio-of-uniforms method.

Consider the following density function of X, denoted by $f(\cdot)$:

$$f(x) = \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x}, \qquad x > 0,$$

which is a gamma distribution with shape parameter α, denoted by $G(\alpha, 1)$. When $X \sim G(\alpha, 1)$ and $W = \beta X$, we have $W \sim G(\alpha, \beta)$, where β is called the scale parameter. Therefore, we focus on random number generation from $G(\alpha, 1)$. Note that $a = \sup_x \sqrt{h(x)} < \infty$ holds when $\alpha \geq 1$, where $h(x) = x^{\alpha-1}e^{-x} \propto f(x)$ in the case of $G(\alpha, 1)$.

When X is a gamma random variate with shape parameter α and $X = Y^n$ is defined, we consider generating a random variate of Y. The density function of Y, denoted by $f_y(\cdot)$, is given by:

$$f_y(x) = \frac{n}{\Gamma(\alpha)} x^{n\alpha-1} \exp(-x^n), \qquad x > 0.$$

Applying the ratio-of-uniforms method, the acceptance region is determined by the following inequality:

$$U_1^2 \leq (U_2/U_1)^{n\alpha-1} \exp(-(U_2/U_1)^n), \tag{1}$$

where $0 < U_1 < a$ and $b < U_2 < c$. We obtain $a = ((\alpha - 1/n)/e)^{(\alpha-1/n)/2}$ when $\alpha > 1/n$ and $a = 1$ when $\alpha = 1/n$, $b = 0$, and $c = ((\alpha + 1/n)/e)^{(\alpha+1/n)/2}$. Note that (U_1, U_2) satisfying (1) is a bounded set when $\alpha \geq 1/n$, i.e., a, b and c exist for $\alpha \geq 1/n$, and accordingly the ratio-of-uniforms method can be applied in this case. See Cheng and Feast (1980). Define $U_1 = aV_1$ and $U_2 = cV_2$, where $V_1 \sim U(0, 1)$ and $V_2 \sim U(0, 1)$. When the acceptance region is determined by the following inequality:

$$\left(\frac{cV_2}{aV_1}\right)^n \leq -n(\alpha + 1/n)\log aV_1 + n(\alpha - 1/n)\log cV_2, \tag{2}$$

then $X = (cV_2/aV_1)^n$ is taken as a gamma random variate with shape parameter $\alpha \geq 1/n$.

Taking into account $h(x) = x^{n\alpha-1}\exp(-x^n)$ in this case, the acceptance rate $AP(\alpha)$ is given by:

$$AP(\alpha) \equiv \frac{\int h(x)\,\mathrm{d}x}{2a(c - b)} = \frac{e^{\alpha}\Gamma(\alpha)}{2n(\alpha + 1/n)^{(\alpha+1/n)/2}(\alpha - 1/n)^{(\alpha-1/n)/2}},$$

Figure 1: The acceptance rate $AP(\alpha)$ given n

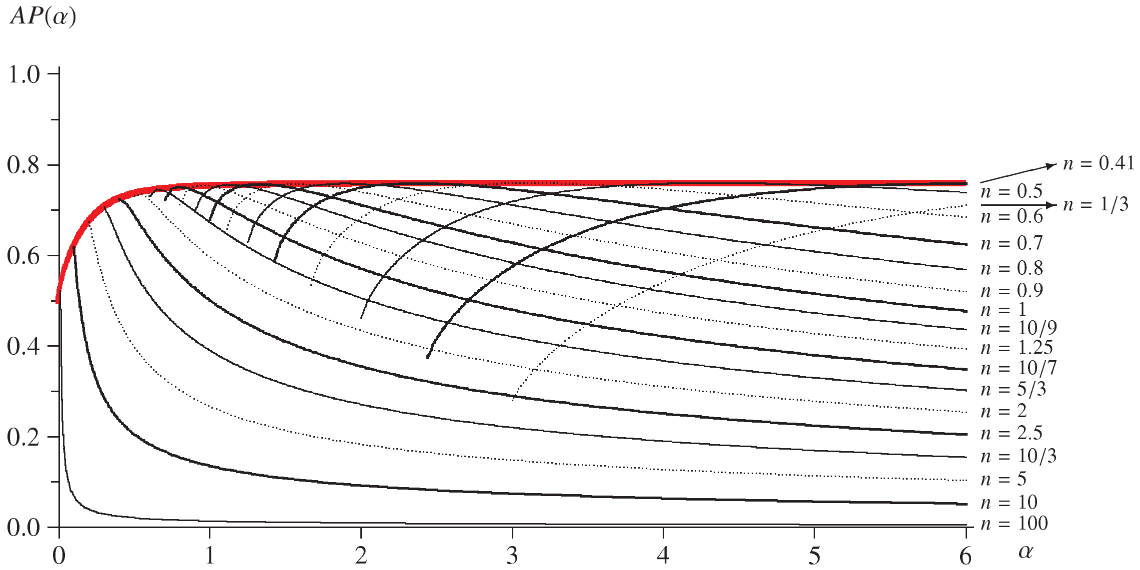


Figure 2: The relationship between n and α which maximizes $AP(\alpha)$

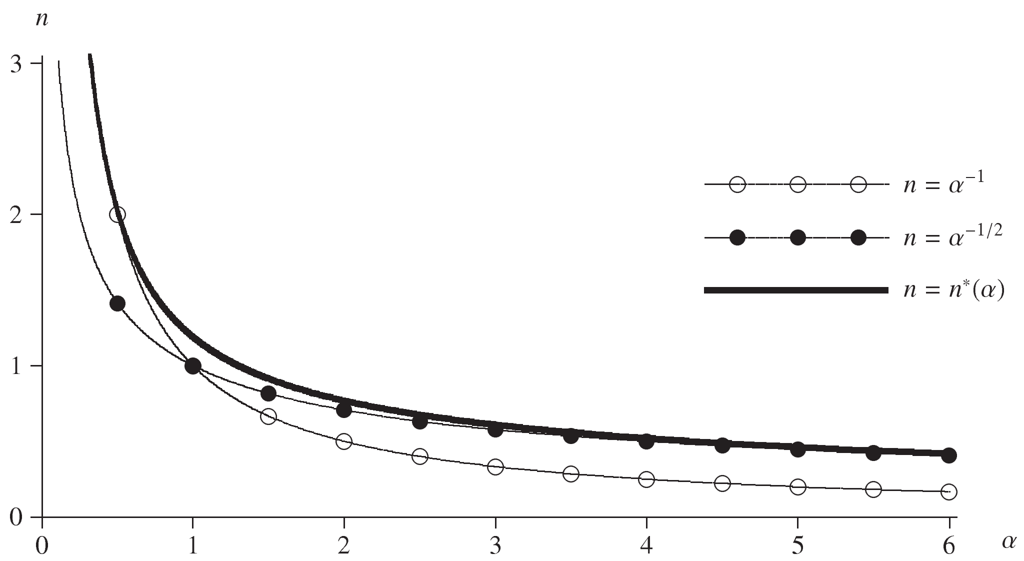


Table 1: Comparison between n^* and n^+

α	α^{-1}	$\alpha^{-1/2}$	n^*	n^+	$AP^*(\alpha)$	$AP^+(\alpha)$	$n^* - n^+$	$AP^*(\alpha)$ $-AP^+(\alpha)$
0.001	1000.0	31.6228	1000.000	1000.000	0.5033	0.5033	0.0000	0.0000
0.1	10.000	3.16228	10.00000	10.00000	0.6175	0.6175	0.0000	0.0000
0.2	5.000	2.23607	5.00045	5.00000	0.6735	0.6735	0.0005	0.0000
0.3	3.333	1.82574	3.34169	3.33333	0.7063	0.7060	0.0084	0.0003
0.4	2.500	1.58114	2.53181	2.50000	0.7255	0.7236	0.0318	0.0019
0.5	2.000	1.41421	2.06534	2.05556	0.7368	0.7367	0.0098	0.0001
1.0	1.000	1.00000	1.19969	1.16667	0.7544	0.7534	0.0330	0.0010
2.0	0.500	0.70711	0.77170	0.72222	0.7589	0.7545	0.0495	0.0044
3.0	0.333	0.57735	0.61145	0.57407	0.7596	0.7560	0.0374	0.0036
4.0	0.250	0.50000	0.52181	0.50000	0.7599	0.7583	0.0218	0.0016
5.0	0.200	0.44721	0.46268	0.44721	0.7600	0.7590	0.0155	0.0010
8.0	0.125	0.35355	0.36109	0.35355	0.7601	0.7597	0.0075	0.0004

which depends on both α and n.

In Figure 1, $AP(\alpha)$ is plotted against α, where the cases of $n = 1/3$, 0.41, 0.5, 0.6, $\cdots$, 100 are drawn. In the cases of $n \geq 2$, $AP(\alpha)$ seems to be maximized when α is close to $1/n$. We consider taking the envelope, which is drawn by the thickest solid line in Figure 1 and obtained as follows:

$$\max_n \; AP(\alpha).$$

Note that given α the acceptance rate $AP(\alpha)$ is numerically maximized with respect to n. Let n^* be the optimal value of n. n^* is given by a function of α, i.e., $n^* = n^*(\alpha)$. Thus, we have the optimal n corresponding to n. In Figure 2, $n^*(\alpha)$ is displayed by the thick solid line. To examine the relationship between α and n, we draw the three curves, i.e., $n = \alpha^{-1}$, $n = \alpha^{-1/2}$ and $n = n^*(\alpha)$, in Figure 2. The reasons why the three curves are compared are as follows. $\alpha \geq 1/n$ has to be satisfied in order to perform the ratio-of-uniforms, i.e., α^{-1} indicates a lower bound of n. Accordingly, the boundary line $n = \alpha^{-1}$ is in Figure 2. Furthermore, it is shown that $n = \alpha^{-1/2}$ for large α, which is verified as follows. Differentiating $\log AP(\alpha)$ with respect to n, we obtain the following equation:

$$2n = \log(\alpha + \frac{1}{n}) - \log(\alpha - \frac{1}{n}). \tag{3}$$

Suppose that $n\alpha \longrightarrow \infty$ as $\alpha \longrightarrow \infty$. Then, using $\lim_{n\alpha\to\infty} \log((1 + (n\alpha)^{-1})^{n\alpha}) = 1$, $n^2\alpha \longrightarrow 1$ is obtained from (3), which implies that $n = \alpha^{-1/2}$ is an optimal solution for large α (note that $n = \alpha^{-1/2}$ satisfies the condition: $n\alpha \longrightarrow \infty$ as $\alpha \longrightarrow \infty$).

Therefore, $n = \alpha^{-1}$, n^* and $n = \alpha^{-1/2}$ are compared in Figure 2. Thus, it is shown from Figure 2 that n^* is close to α^{-1} for small α and it is close to $\alpha^{-1/2}$ for large α. n^* has to be larger than or equal to the α^{-1} line, because of $\alpha - 1/n \geq 0$. n^* can be numerically obtained, but it is time-consuming to compute n^* every time we generate gamma random draws for fixed α. Therefore, we consider approximating $n^* = n^*(\alpha)$. Let n^+ be an approximation of n^*. As shown in Figure 2, it seems that n^* is close to α^{-1} for $0 < \alpha \leq 0.4$ and $\alpha^{-1/2}$ for $\alpha > 4$. As for $0.4 < \alpha \leq 4.0$, we take $n^+ = \alpha^{-1} + \alpha^{-1}(\alpha - 0.4)/3.6$, where the relationship between n^+ and α represents a line passing through two points $(n^+, \alpha) = (0.4^{-1}, 0.4)$, $(2^{-1}, 4)$. Thus, n is represented as a continuous function of α for any $\alpha > 0$, where the condition $\alpha \geq 1/n$ is satisfied.

Table 1 shows how close it is between n^* and n^+. $AP^*(\alpha)$ and $AP^+(\alpha)$ are denoted by $AP(\alpha)$ evaluated at $n = n^*$ and $n = n^+$, respectively. In the table, $AP^*(\alpha)$ is almost equal to $AP^+(\alpha)$. More precisely, the maximum value of $n^* - n^+$ is given by 0.0502 at $\alpha = 1.78$, while that of $AP^*(\alpha) - AP^+(\alpha)$ is 0.00448 at $\alpha = 2.23$. Thus, n^+ gives us a good approximation of n^*.

Practically, it might be safe to take the logarithms in order to avoid the computational problem in which we have an overflow in $(V_2/V_1)^n$ especially when V_1 is close to zero, V_2 is close to one and n is large. Summarizing, the algorithm for generating gamma random variates when the shape parameter is greater than zero is shown as follows.

(i) Given α, set n, b_1, b_2, c_1 and c_2 as follows:

$$n = \begin{cases} \alpha^{-1}, & \text{if } 0.0 < \alpha \leq 0.4, \\ \alpha^{-1} + \alpha^{-1}(\alpha - 0.4)/3.6, & \text{if } 0.4 < \alpha \leq 4.0, \\ \alpha^{-1/2}, & \text{if } 4.0 < \alpha, \end{cases} \tag{4}$$

$$b_1 = \alpha - 1/n, \qquad b_2 = \alpha + 1/n,$$

$$c_1 = \begin{cases} 0, & \text{if } 0.0 < \alpha \leq 0.4, \\ b_1(\log b_1 - 1)/2, & \text{if } 0.4 < \alpha, \end{cases}$$

$$c_2 = b_2(\log b_2 - 1)/2.$$

(ii) Generate v_1 and v_2 independently from $U(0,1)$. Set $w_1 = c_1 + \log v_1$, $w_2 = c_2 + \log v_2$, and $y = n(b_1 w_2 - b_2 w_1)$.

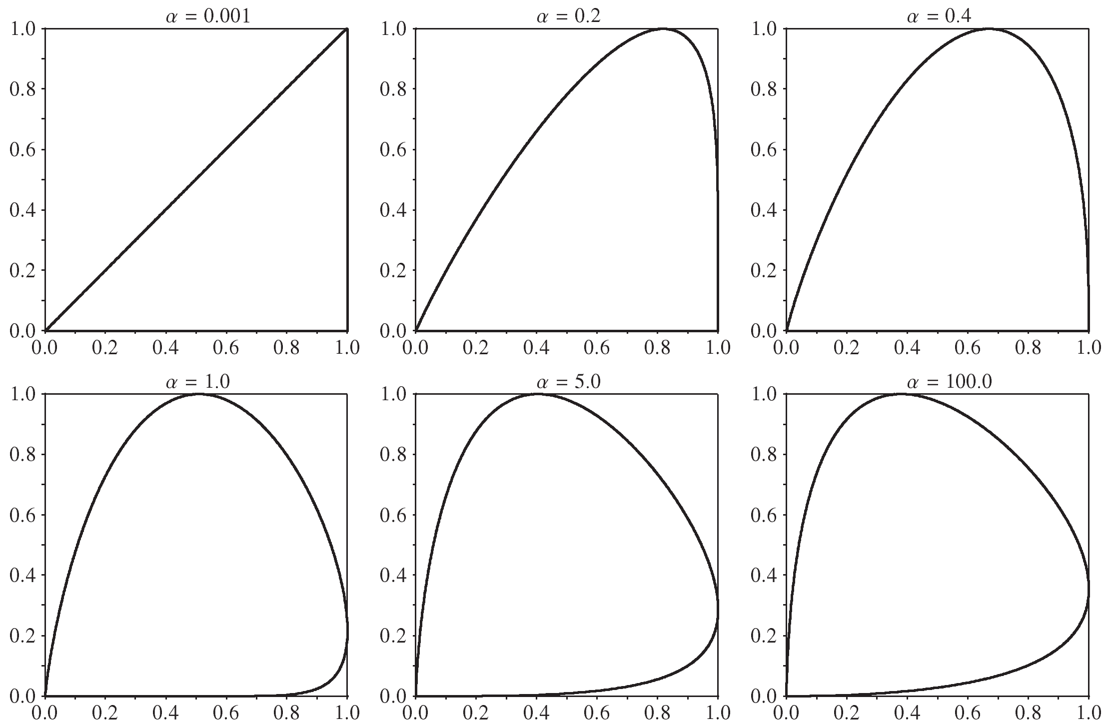
(iii) Go to (ii) if $y < 0$.

(iv) Set $x = n(w_2 - w_1)$, and take e^x as a gamma random draw with shape parameter α if $\log y \geq x$ and go to (ii) otherwise.

The acceptance regions determined by (2) and (4), which is hereafter called (2)+(4) in this paper, are displayed in Figure 3, where the vertical axis indicates V_1 and the horizontal axis represents V_2. The inside of the closed curve corresponds to the area shown by (2)+(4). As α goes to zero, (2)+(4) approaches the right and isosceles triangle, i.e., the acceptance rate goes to 0.5. (2)+(4) is close to each other for all $\alpha > 1$, which is also shown from the envelope in Figure 1 and $AP^+(\alpha)$ in Table 1. Thus, we can see that (2)+(4) has a broad stable acceptance probability within the domain of α.

Figure 3: The acceptance regions determined by (2) and (4)

— Vertical axis V_1 and horizontal axis V_2 —



The gamma random number generator shown in (i) – (iv) is really simple and it does not have any restriction on a range of the shape parameter. Therefore, it might be very easy and useful in programming.

Cheng and Feast (1979) considered the acceptance region given by (2) in the case of $n = 1$ and they approximated the acceptance region, taking into account computational aspects. Their approximation shows a good performance when α is large, but it is quite poor when α is close to one (for example, see Table 3.7 in Fishman (1996, p.196)). Because the $AP(\alpha)$ evaluated at $n = 1$ is small when α is large (see the case of $n = 1$ in Figure 1), Cheng and Feast (1979) suggest that (V_1, V_2) is generated as:

$$\begin{cases} V_1 \sim U(0,1), \quad V_2 \sim U(0,1), & \text{for } 1 < \alpha < 2.5, \\ V_1 \sim U(0,1), \quad V_2 = V_1 + (1 - U_0(1 + \sqrt{2/e}))/\sqrt{\alpha}, & \text{for } 2.5 \leq \alpha, \end{cases} \tag{5}$$

where $U_0 \sim U(0,1)$. Using $n = 1$, (2) and (5), the acceptance rate $AP(\alpha)$ is given by: $AP(\alpha) \longrightarrow e/4 = 0.6796$ as $\alpha \longrightarrow 1$ and $AP(\alpha) \longrightarrow \sqrt{\pi/2}/(1 + \sqrt{2/e}) = 0.6746$ as $\alpha \longrightarrow \infty$. See Ripley (1987, p.89) and Fishman (1996, p.198). In this paper, under (2) and (4) we have $AP(\alpha) \longrightarrow 0.5$ as $\alpha \longrightarrow 0$ and $AP(\alpha) \longrightarrow (\frac{1}{2}\pi/e)^{1/2} = 0.7602$ as $\alpha \longrightarrow \infty$, where we utilize the followings: $\Gamma(\alpha + 1)/\alpha = \Gamma(\alpha)$ and $n\alpha = 1$ for

Table 2: CPU time (hundreds of seconds)

	(a) w/o Parameter Setup							(b) w/ Parameter Setup						
α	(2)+(4)	B83	CF79	C77	SL80	MT01	$\overline{\text{MT01}}$	(2)+(4)	B83	CF79	C77	SL80	MT01	$\overline{\text{MT01}}$
0.01	63.3	26.3	—	—	—	59.0	48.2	69.5	39.4	—	—	—	61.4	52.7
0.1	54.7	28.2	—	—	—	57.6	47.2	61.0	40.8	—	—	—	60.6	52.2
0.2	50.8	31.0	—	—	—	57.4	47.0	57.0	43.9	—	—	—	60.3	51.7
0.3	48.8	41.8	—	—	—	66.6	56.2	55.0	55.4	—	—	—	68.3	59.4
0.4	47.8	43.4	—	—	—	66.6	56.2	54.0	53.0	—	—	—	68.2	59.4
0.6	46.9	43.5	—	—	—	66.6	56.2	59.1	53.1	—	—	—	68.1	59.3
0.8	46.6	39.1	—	—	—	66.6	56.2	58.7	48.7	—	—	—	68.0	59.2
0.99	46.5	25.9	—	—	—	66.6	56.2	58.6	39.8	—	—	—	67.9	59.2
1.01	46.5	—	49.4	48.5	27.0	41.3	30.8	58.6	—	63.7	58.2	60.4	43.3	33.4
1.4	46.4	—	46.0	43.6	26.6	40.6	30.3	58.6	—	59.7	53.3	59.7	42.6	32.9
1.8	46.4	—	47.2	41.5	27.7	40.2	30.0	58.5	—	61.1	51.2	58.1	42.3	32.7
2.2	46.1	—	49.2	40.3	27.5	40.0	29.8	58.3	—	63.4	50.0	82.0	42.1	32.5
2.6	46.0	—	57.8	39.6	26.8	39.9	29.8	60.7	—	73.8	49.3	84.4	41.9	32.4
3	45.9	—	56.9	39.0	26.5	39.8	29.7	58.0	—	72.7	48.8	81.0	41.9	32.4
4	45.7	—	55.6	38.2	26.1	39.7	29.6	57.9	—	71.2	47.9	83.9	41.7	32.3
5	45.7	—	54.9	37.8	25.9	39.6	29.5	57.6	—	70.3	47.5	83.8	41.7	32.2
10	45.6	—	53.8	37.0	25.8	39.5	29.4	57.5	—	68.9	46.7	80.4	41.5	32.1
20	45.7	—	53.4	37.0	25.8	39.4	29.4	57.6	—	68.5	46.6	83.4	41.5	32.0
50	45.1	—	52.8	37.6	26.1	39.4	29.3	56.8	—	67.7	47.2	83.7	41.4	32.0
100	44.9	—	52.8	38.2	26.2	39.4	29.4	56.6	—	67.2	47.7	89.1	41.4	32.0
400	44.7	—	53.0	38.8	26.4	39.7	29.6	56.1	—	67.8	48.3	97.6	41.6	32.2

Compiled by: `wfl386 /Ox [Fortran file]`.

small α, and $\Gamma(\alpha) \approx e^{-\alpha}\alpha^{\alpha-1/2}\sqrt{2\pi}$ and $n = \alpha^{-1/2}$ for large α. Thus, for all α, the $AP(\alpha)$ in the case of (4) is larger than the $AP(\alpha)$ in the case of $n = 1$ and (5). The generator proposed in this paper is better than the Cheng and Feast (1979) generator with respect to the acceptance rate.

3 Simulation Studies

In Table 2, 10^{10} gamma random draws are generated and computational time (hundreds of seconds) is compared for seven generators. Dual Xeon 3.6GHz CPU (Socket 604, FSB 800MHz, L2 2MB) Personal Computer, Microsoft Windows XP Professional Version SP2 Operating System, and Open Watcom F77/32 Compiler (Version 1.5, downloaded from `http://www.openwatcom.com`) are utilized. In the table, (2)+(4), B83, CF79, C77, SL80, MT01 and $\overline{\text{MT01}}$ represent as follows:

- (2)+(4): The generator proposed in this paper, where the acceptance region is constructed by (2) and (4).
- B83: The Best (1983) generator.
- CF79: The Cheng and Feast (1979) generator, where the acceptance region is given by the case of $n = 1$ in (2) and (5).
- C77: The Cheng (1977) generator.

- SL80: The Schmeiser and Lal (1980) generator.
- MT01: The Marsaglia and Tsang (2001) generator using the Box-Muller transformation for standard normal random number generation. From the fact that $XU^{1/\alpha} \sim G(\alpha, 1)$ when $X \sim G(\alpha + 1, 1)$ and $U \sim U(0, 1)$ (for example, see Devroye (1986, p.420)), Marsaglia and Tsang (2001) is extended to the case of $\alpha > 0$.
- $\overline{\text{MT01}}$: This is equivalent to MT01 except for use of the standard normal random number generation method suggested by Hörmann and Derflinger (1990).

In the table, (a) indicates CPU times in the case where parameters are initially setup and 10^{10} random draws are generated, while (b) represents CPU times in the case where parameters are setup every time one random draw is generated. Taking an example of the algorithm (i) – (iv) in Section 2, when 10^{10} random draws are generated, the former repeats (ii) – (iv) 10^{10} times and the latter repeats (i) – (iv) 10^{10} times. Sometimes the latter is more practical than the former in programming. Because SL80 requires a lot of parameters to be set up, it might be expected that (a) performs much better than (b). The uniform random number generator utilized in this paper is given by L'Ecuyer (1988, Figure 3 on p.747), where for 32-bit computers the period of the uniform random number generator is of the order of 2.31×10^{18}.

As shown in Figure 1 and Table 1, (2)+(4) becomes stable very quickly for a wide range of α although it takes a larger value for small α. As for B83, we have a large rejection rate around the middle between zero and one, and accordingly the cases of $\alpha = 0.3, 0.4, 0.6$ take larger values for both (a) and (b). For (a), C77 takes a long time when α is close to one, compared with large α. MT01 and $\overline{\text{MT01}}$ take similar values for all $\alpha > 1$, i.e., the marginal generation times for MT01 and $\overline{\text{MT01}}$ are about 40 and 30 for all $\alpha > 1$, respectively, but the cases of $0 < \alpha < 1$ are much larger than those of $\alpha > 1$, because the extra computation in the case of $0 < \alpha < 1$ takes a quite long time. MT01 is better than C77 when α is close to one. $\overline{\text{MT01}}$ is much better than MT01. In Table 2(a) C77 and MT01 are not too different from (2)+(4) as α is large. However, SL80 and $\overline{\text{MT01}}$ are much better than (2)+(4) for $\alpha > 1$, although they are complicated in programming (note that SL80 needs almost five times longer steps than (2)+(4) and that the Hörmann-Derflinger standard normal random number generator in $\overline{\text{MT01}}$ is more than ten times longer than the Box-Muller generator in MT01). Moreover, as it is expected, (2)+(4) is superior to CF79 for all $\alpha > 1$. Note that CF79 has discontinuity at $\alpha = 2.5$ in computational time because of (5). As for (a), simple and fast generators are C77 and MT01, but not too different from the proposed generator (2)+(4).

In the case where both parameter initialization and random number generation are included in programming at the same time, i.e., in the case of Table 2(b), SL80 performs much worse than the others. For both MT01 and $\overline{\text{MT01}}$ in Table 2, there is not too much difference between (a) and (b), which implies that it does not take too much extra time to initialize parameters.

4 Summary

In this paper, we have proposed the simple gamma random number generator shown in the algorithm (i) – (iv), which has no restriction on the shape parameter. Schmeiser and Lal (1980) proposed a fast generator in the case of $\alpha > 1$, which is modified by Sarkar (1996). Law and Kelton (2000, p.464) mentioned that the generator proposed by Schmeiser and Lal (1980) is roughly twice as fast as the one presented by Cheng (1977), but the algorithm of Schmeiser and Lal (1980) is much more complicated and requires additional time to set up the necessary constants for a given value of $\alpha > 1$ (also, see Tadikamalla and Johnson, 1981). In the case of $0 < \alpha < 1$, the algorithm of Best (1983) is known as the best gamma random number generator. Thus, different algorithms are conventionally used for $0 < \alpha < 1$ and $\alpha > 1$, i.e., there are not any gamma random number generators which utilize the same algorithm without depending on the value of α. Cheng and Feast (1980) discussed the cases of $\alpha > 1/n$. Based on Cheng and Feast (1980), we have proposed the gamma random number generator for $\alpha > 0$ by taking n as a decreasing function of α, and the proposed generator is really simple and practically useful in programming although it is not the best generator in a sense of speed.

References

Ahrens, J.H. and Dieter, U., 1974, "Computer Methods for Sampling from Gamma, Beta, Poisson and Binomial Distributions," *Computing*, Vol.12, pp.223 – 246.

Ahrens, J.H. and Dieter, U., 1982, "Generating gamma variates by a modified rejection technique," *Communications of the ACM*, Vol.25, pp.47 – 54.

Berman, M.B., 1971, "Generating gamma distributed variates for computer simulation models," Technical Report R-641-PR, Rand Corporation.

Best, D.J., 1978, "Letter to the Editors," *Applied Statistics*, Vol.27, p.181.

Best, D.J., 1983, "A Note on Gamma Variate Generators with Shape Parameter Less Than Unity," *Computing*, Vol.30, pp.185 – 188.

Cheng, R.C.H., 1977, "The Generation of Gamma Variables with Non-Integral Shape Parameter," *Applied Statistics*, Vol.26, No.1, pp.71 – 75.

Cheng, R.C.H. and Feast, G.M., 1979, "Some Simple Gamma Variate Generators," *Applied Statistics*, Vol.28, No.3, pp.290 – 295.

Cheng, R.C.H. and Feast, G.M., 1980, "Gamma Variate Generators with Increased Shape Parameter Range," *Communications of the ACM*, Vol.23, No.7, pp.389 – 394.

Devroye, L., 1986, *Non-Uniform Random Variate Generation* Springer-Verlag.

Fishman, G.S., 1976, "Sampling from the Gamma Distribution on a Computer," *Communications of the ACM*, Vol.19, No.7, pp.407 – 409.

Fishman, G.S., 1996, *Monte Carlo: Concepts, Algorithms and Applications* (Springer Series in Operation Research), Springer-Verlag.

Hörmann, W. and Derflinger, G., 1990, "The ACR method for generating normal random variables," *OR Spektrum*, Vol.12, pp.181 – 185.

Johnk, M.D., 1964, "Erzeugung von Betaverteilten und Gammaverteilten Zufallazahlen," *Metrika*, Vol.8, pp.5 – 15.

Kinderman, A.J. and Monahan, J.F., 1977, "Computer Generation of Random Variables Using the Ratio of Uniform Deviates," *ACM Transactions on Mathematical Software*, Vol.3, No.3, pp.257 – 260.

L'Ecuyer, P., 1988, "Efficient and Portable Combined Random Number Generators," *Communications of the ACM*, Vol.31, No.6, pp.742 – 749.

Law, A.M. and Kelton, W.D., 2000, *Simulation Modeling and Analysis (3rd Ed.)*, McGraw-Hill.

Marsaglia, G. and Tsang, W.W., 2001, "A Simple Method for Generating Gamma Variables," *ACM Transactions on Mathematical Software*, Vol.26, No.3, pp.363 – 372.

Minh, D.L., 1988, "Generating Gamma Variates," *ACM Transactions on Mathematical Software*, Vol.14, No.3, pp.261 – 266.

Ripley, B.D., 1987, *Stochastic Simulation*, John Wiley & Sons.

Sarkar, T.K., 1996, "A Composition-Alias Method for Generating Gamma Variates with Shape Parameter Greater Than 1," *ACM Transactions on Mathematical Software*, Vol.22, No.4, pp. 484 – 492.

Schmeiser, B. and Lal, R., 1980, "Squeeze Methods for Generating Gamma Variates," *Journal of the American Statistical Association*, Vol.75, pp.679 – 682.

Tadikamalla, P.R., 1978, "Computer Generation of Gamma Random Variates – II," *Communications of the ACM*, Vol.21, No.11, pp.925 – 928.

Tadikamalla, P.R. and Johnson, M.E., 1981, "A Complete Guide to Gamma Variate Generation," *American Journal of Mathematical and Management Sciences*, Vol.1, No.3, pp.213 – 236.