Logarithmic Transformation-Based Gamma Random Number Generators

Bowei Xi

Purdue University

Kean Ming Tan

University of Washington

Chuanhai Liu

Purdue University

Abstract

Developing efficient gamma variate generators is important for Monte Carlo methods. With a brief review of existing methods for generating gamma random numbers, this article proposes two simple gamma variate generators that are obtained from the ratio-of-uniforms method and based on two logarithmic transformations of the gamma random variable. One transformation allows for the generators to work for all shape parameter values. The other is introduced to have improved efficiency for shape parameters smaller than or equal to one.

Keywords: gamma distribution, random number generation, ratio-of-uniforms.

1. Introduction

The unit gamma distribution, $\text{Gamma}(\alpha)$, with density

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

where $\alpha (> 0)$ is the shape parameter, is fundamental in computational statistics. It has simple relationship with commonly used distributions in statistics, including chi-square, Student- t , F , Beta, and, of course, scaled gamma distributions. In the past many articles proposed different algorithms to generate gamma random numbers; see Devroye (1986) and Tanizaki (2008) for a comprehensive survey. However, the existing algorithms apply to only a limited range of shape parameter values, either for $\alpha \geq 1$ (e.g., Ahrens and Dieter 1982; Marsaglia and Tsang 2000) or for $\alpha < 1$ (e.g., Ahrens and Dieter 1974). Marsaglia and Tsang (2000) proposed a gamma variate generator for the $\alpha \geq 1$ case. They suggested that gamma variates with shape parameter α smaller than one can be generated using the fact that if two random variables X and Y are independent with $X \sim \text{Gamma}(\alpha + 1)$ and $Y \sim \text{Uniform}(0, 1)$, then

$XY^{1/\alpha} \sim \text{Gamma}(\alpha)$. In the same vein, based on the well-known fact that the sum of two independent gamma random variables with shape parameters n and δ is $\text{Gamma}(n + \delta)$, [Ahrens and Dieter \(1974\)](#) mentioned that a gamma variate with shape parameter larger than one can be generated using a generator for integer shape parameters and another generator for $\alpha < 1$. Since they need to draw one more random number, such random number generators are inefficient when a high quality uniform random number generator with a very long period is used.

The ratio-of-uniforms (ROU) method of [Kinderman and Monahan \(1977\)](#), a special case of acceptance-rejection sampling method, can be used to create gamma random number generators. Let $h(t)$ be a non-negative function with finite integral M_h . That is, the normalized function $h(t)/M_h$ is a probability density function. If the random point (U, V) is uniformly distributed over what we call the ROU region

$$\mathcal{C} = \left\{ (u, v) : 0 \leq u \leq \sqrt{h(v/u)} \right\},$$

then $T = V/U$ has density $h(t)/M_h$. Hence to generate random numbers from the distribution with density $h(t)/M_h$, we generate a random point (U, V) uniformly over the ROU region $\mathcal{C}$, often using rejection sampling, and then compute the ratio $T = V/U$ as the output. [Cheng and Feast \(1980\)](#) used the ROU method and a power transformation of the gamma random variable. To generate gamma random numbers with shape parameter greater than or equal to $1/n$, where n is an integer, they considered the power transformation $Y = X^{1/n}$ and provided two illustrative algorithms for the special cases $n = 2$ and $n = 4$.

In this article, we propose two algorithms to generate gamma random numbers using the ROU method. Both algorithms are based on logarithmic transformations of gamma random variables. One algorithm applies to all positive shape parameter values without any limitation. The second algorithm is limited to shape parameters smaller or equal to one, but it has better performance in that range. In terms of transformation, the proposed algorithms can be considered as the limiting case of [Cheng and Feast \(1980\)](#) i.e., $1/n \rightarrow 0$. In addition to extending the method of [Cheng and Feast \(1980\)](#) for all shape parameters by using the logarithmic transformation, our methods also take into account centering and rescaling for improved efficiency (see, e.g., [Wakefield, Gelfand, and Smith 1991](#)).

The remaining of the article is organized as follows. Section 2 presents the algorithm that applies to all positive shape parameter values. Section 3 presents the algorithm that applies to shape parameters smaller than or equal to one. Finally, Section 4 concludes the article.

2. The algorithm for all shape parameters

2.1. The “standardized” logarithmic transformation

Suppose that $X \sim \text{Gamma}(\alpha)$. We consider the transformed random variable

$$T = \sqrt{\alpha} \ln \left(\frac{X}{\alpha} \right) \tag{2}$$

for generating random numbers from $\text{Gamma}(\alpha)$. That is, we first generate T using the ROU method, then obtain $X = \alpha e^{T/\sqrt{\alpha}}$.

Figure 1: ROU region $\mathcal{C}_\alpha$ for $\alpha = 0.001, 0.1, 1$, and 10 respectively. The horizontal red dotted lines show the range $v_{\min}(\alpha) \leq v \leq v_{\max}(\alpha)$. The vertical solid blue lines show the range $0 \leq u \leq 1$. The horizontal solid blue lines show the range $-\exp(b_w(\theta)) \leq v \leq \exp(b_s(\theta))$, where $b_s(\theta)$ and $b_w(\theta)$ are specified in Tables 1 and 2. For $\alpha = 0.001$ and 0.1 , the horizontal solid blue lines and the red dotted lines are almost indistinguishable, suggesting that the bounds are very close to the actual maximum and minimum values.

The density of T is

$$f_T(t) = \frac{\alpha^{\alpha-1/2}}{\Gamma(\alpha)} \exp(\sqrt{\alpha}t - \alpha e^{t/\sqrt{\alpha}}) \quad (-\infty < t < \infty). \quad (3)$$

Applying Stirling's formula for the gamma function and taking the Taylor expansion of $e^{t/\sqrt{\alpha}}$, one can show that, as $\alpha \rightarrow \infty$, $f_T(t)$ approaches the density of the standard normal distribution. For this reason, we call this transformation the “standardized” logarithmic transformation.

2.2. Theoretical results on the ROU region

To apply the ROU method to the density function $f_T(t)$ in Equation 3, we take

$$h_\alpha(t) = \exp(\sqrt{\alpha}t - \alpha e^{t/\sqrt{\alpha}} + \alpha). \quad (4)$$

Hence the ROU region is

$$\mathcal{C}_\alpha = \left\{ (u, v) : 0 \leq u \leq \sqrt{h_\alpha(t)}, \quad -\infty < t < \infty \right\}, \quad (5)$$

where $t = v/u$. Routine algebraic operations lead to the area of the ROU region $\mathcal{C}_\alpha$:

$$|\mathcal{C}_\alpha| = \frac{\Gamma(\alpha)e^\alpha}{2\alpha^{\alpha-\frac{1}{2}}}.$$

As shown in Figure 1, both the area and the shape of the ROU region $\mathcal{C}_\alpha$ depend on α . The gray regions in Figure 1 are the ROU regions for $\alpha = 0.001, 0.1, 1$, and 10 respectively.

Since $\max_t \sqrt{h_\alpha(t)} = 1$, $\forall \alpha > 0$, the range of u is given by $0 \leq u \leq 1$. This is shown in Figure 1 by the vertical solid blue lines. The boundary curve of the ROU region $\mathcal{C}_\alpha$ is defined by $v = t\sqrt{h_\alpha(t)}$, $t \in (-\infty, \infty)$. The range of v is determined by

$$v_{\max}(\alpha) = \max_{t>0} t\sqrt{h_\alpha(t)}$$

and

$$v_{\min}(\alpha) = \min_{t<0} t\sqrt{h_\alpha(t)}$$

and changes with α . The range $v_{\min}(\alpha) \leq v \leq v_{\max}(\alpha)$ is shown in Figure 1 by the horizontal dotted red lines.

To generate T using the ROU method given an α , we uniformly sample (U, V) from the ROU region $\mathcal{C}_\alpha$, then take the ratio V/U . This is achieved by rejection sampling: We use a rectangular region to cover $\mathcal{C}_\alpha$, uniformly sample from the rectangular region, and accept the point if it is in the ROU region $\mathcal{C}_\alpha$. Maximization and minimization of $t\sqrt{h_\alpha(t)}$ do not have closed-form solutions. Hence we cannot directly use $0 \leq u \leq 1$ and $v_{\min}(\alpha) \leq v \leq v_{\max}(\alpha)$ to cover $\mathcal{C}_\alpha$. Here we investigate the theoretical properties of the natural logarithm of $v_{\max}(\alpha)$ and $|v_{\min}(\alpha)|$. We re-parameterize by

$$\theta = \ln \alpha.$$

Both notations α and θ are used in this article later. Let $s(\theta) = \ln v_{\max}(\alpha)$ and $w(\theta) = \ln |v_{\min}(\alpha)|$. Here we investigate their properties, which are summarized in the following two theorems, with their proofs given in Appendix A.

Theorem 1 *Let*

$$s(\theta) = \max_{t>0} \ln(t\sqrt{h_\alpha(t)}).$$

Then, as a function of $\theta \in (-\infty, \infty)$, $s(\theta)$ is strictly increasing and concave. Furthermore,

$$\lim_{\theta \rightarrow -\infty} \frac{s(\theta)}{\theta} = \frac{1}{2}, \quad (6)$$

$$\lim_{\theta \rightarrow \infty} s(\theta) = \ln \sqrt{\frac{2}{e}} = \frac{1}{2}(\ln 2 - 1). \quad (7)$$

Figure 2: $s(\theta)$ and $w(\theta)$ are shown as black solid curves. Their upper bounds $b_s(\theta)$ and $b_w(\theta)$ are shown as red dashed curves. $b_s(\theta)$ and $b_w(\theta)$ are specified in Tables 1 and 2.

Theorem 2 *Let*

$$w(\theta) = \max_{t < 0} \ln(|t|\sqrt{h_\alpha(t)}).$$

Then, as a function of $\theta \in (-\infty, \infty)$, $w(\theta)$ is strictly decreasing and convex. Furthermore,

$$\lim_{\theta \rightarrow -\infty} \frac{w(\theta)}{\theta} = -\frac{1}{2}, \quad (8)$$

$$\lim_{\theta \rightarrow \infty} w(\theta) = \ln \sqrt{\frac{2}{e}} = \frac{1}{2}(\ln 2 - 1). \quad (9)$$

The functions $s(\theta)$ and $w(\theta)$ are shown as the solid black curves in Figure 2. For any given α , we need to run numerical optimization algorithms to obtain the values of $s(\theta)$ and $w(\theta)$. It is infeasible to directly use $s(\theta)$ and $w(\theta)$ in a gamma random number generator, since the initial parameter setup step would take too long. Hence we construct simple upper bounds of $s(\theta)$ and $w(\theta)$, denoted by $b_s(\theta)$ and $b_w(\theta)$ respectively. Hence

$$v_{\max}(\alpha) = \exp(s(\theta)) \leq \exp(b_s(\theta)) \quad \text{and} \quad v_{\min}(\alpha) = -\exp(w(\theta)) \geq -\exp(b_w(\theta)).$$

As a result, the rectangle given by $0 \leq u \leq 1$ and $-\exp(b_w(\theta)) \leq v \leq \exp(b_s(\theta))$ covers $\mathcal{C}_\alpha$. The resulting acceptance rate of the corresponding acceptance-rejection method for simulating desired (U, V) is smaller than that using the exact range of v .

The choice of $b_s(\theta)$ and $b_w(\theta)$ is not unique. We use a piece-wise linear curve based on connecting the tangent lines to $s(\theta)$ to construct the upper bound $b_s(\theta)$. To construct $w(\theta)$'s upper bound $b_w(\theta)$, we first find an upper bound of $w(\theta)$ for the smallest θ s. Then we choose a set of points on the curve $w(\theta)$ and use the intervening line segments as an upper bound for the rest of the θ values. The upper bounds are shown as red dashed curves in Figure 2. These are detailed next in Section 2.3.

2.3. Constructing upper and lower bounds of the ROU region

The optimal rectangular region that covers the ROU region is the one using the exact maximum and minimum of v ,

$$\mathcal{B} = \{(u, v) : 0 \leq u \leq 1, -\exp(w(\theta)) \leq v \leq \exp(s(\theta))\}.$$

The rectangular region that we use to cover the ROU region is based on the upper bounds,

$$\mathcal{B}_R = \{(u, v) : 0 \leq u \leq 1, -\exp(b_w(\theta)) \leq v \leq \exp(b_s(\theta))\}.$$

The ratio of the size of the two rectangular regions is

$$r(\theta) = \frac{|\mathcal{B}|}{|\mathcal{B}_R|} = \frac{\exp(s(\theta)) + \exp(w(\theta))}{\exp(b_s(\theta)) + \exp(b_w(\theta))}.$$

Thus, the quantity $1 - r(\theta)$ measures the loss of efficiency. To construct efficient upper bounds, we pre-specify a minimum rate R (for example $R = 0.9$ or 0.99), such that

$$r(\theta) \geq R, \quad \forall \theta \in (-\infty, +\infty).$$

Given the chosen R ($R < 1$), we construct the upper bounds $b_w(\theta)$ and $b_s(\theta)$ as follows.

Constructing $b_w(\theta)$ given R

Let

$$\delta = -\ln R.$$

For a given θ , if both $w(\theta) \leq b_w(\theta) \leq w(\theta) + \delta$ and $s(\theta) \leq b_s(\theta) \leq s(\theta) + \delta$, then we have $r(\theta) \geq R$. Here $\forall \theta \in (-\infty, +\infty)$, we construct a $b_w(\theta)$ such that $w(\theta) \leq b_w(\theta) \leq w(\theta) + \delta$.

It follows from the inequality $\ln x \leq x - 1$ that $\forall t < 0$,

$$\ln |v| \leq \ln(|t|\sqrt{h_\alpha(t)}) \leq \ln(|t|e^{\frac{\sqrt{\alpha}}{2}t + \frac{\alpha}{2}}) \leq \ln\left(\frac{2}{\sqrt{\alpha}}e^{\frac{\alpha}{2}-1}\right) = (\ln 2 - 1) - \frac{\theta}{2} + \frac{e^\theta}{2}.$$

Note that the function $(\ln 2 - 1) - \frac{\theta}{2} + \frac{e^\theta}{2}$ is strictly increasing for $\theta > 0$. As shown in the right panel of Figure 2, the above function of θ (the left most segment of the red dashed curve) can serve as a tight upper bound for the small θ values. However, the factor $\frac{e^\theta}{2} - \frac{\theta}{2}$ goes to ∞ quickly as α increases. This upper bound is inefficient for large θ .

Here we propose an alternative method to construct $b_w(\theta)$. The proposed method consists of two steps. First, find θ_δ^L such that

$$(\ln 2 - 1) - \frac{\theta_\delta^L}{2} + \frac{e^{\theta_\delta^L}}{2} = w(\theta_\delta^L) + \delta. \quad (10)$$

Similar to $(\ln 2 - 1) - \frac{\theta}{2} + \frac{e^\theta}{2}$, the function $(\ln 2 - 1) - \frac{\theta}{2} + \frac{e^\theta}{2} - w(\theta)$ is also strictly increasing. Thus, Equation 10 has a unique solution θ_δ^L . Let

$$b_w(\theta) = (\ln 2 - 1) - \frac{\theta}{2} + \frac{e^\theta}{2} \quad (\theta \leq \theta_\delta^L).$$

We have $w(\theta) \leq b_w(\theta) \leq w(\theta) + \delta$, $\forall \theta \leq \theta_\delta^L$.

Second, for the remaining interval $(\theta_\delta^L, \infty)$ we switch to the intervening line segments connecting a set of points on the curve $w(\theta)$; see the red dashed line in the right panel of Figure 2. The function $w(\theta)$ on the remaining interval $(\theta_\delta^L, \infty)$ decreases from $w(\theta_\delta^L)$ to $w(\infty) = \ln \sqrt{2/e}$. This step simply partitions $(\theta_\delta^L, \infty)$ into a number of sub-intervals over which $b_w(\theta) \leq w(\theta) + \delta$. The number of sub-intervals is determined by

$$M\delta \leq w(\theta_\delta^L) - w(\infty) \leq (M+1)\delta, \quad (11)$$

where M is an integer.

Let $w(\theta) = \ln \sqrt{2/e} + k\delta$, $k = 1, \dots, M$, which specifies a set of points θ_k^L in $(\theta_\delta^L, \infty)$, and let $\theta_{M+1}^L = \theta_\delta^L$. The intervening line segments connecting the points on $w(\theta)$ at $\theta = \theta_{M+1}^L, \dots, \theta_1^L$ form the second part of $b_w(\theta)$. The right most segment is a horizontal line $b_w(\theta) = \ln \sqrt{2/e} + \delta$. It follows that the difference between $w(\theta)$ and $b_w(\theta)$ is at most δ in $(\theta_\delta^L, \infty)$ as well.

To numerically solve for θ_k^L , $k = 1, \dots, M+1$, ($\theta_{M+1}^L < \theta_M^L < \dots < \theta_1^L$), we need to have an explicit expression of $w(\theta)$, though not directly as a function of θ . Following the proof of Theorem 2 in Appendix A, we define y implicitly as a function of θ given by

$$\frac{2}{y(1 - e^{-y})} = e^\theta. \quad (12)$$

The $w(\theta)$ can be written explicitly as a function of y ,

$$w(\theta) = \frac{1 - y - e^{-y}}{y(1 - e^{-y})} + \frac{\ln(2y)}{2} - \frac{\ln(1 - e^{-y})}{2} \quad (13)$$

Write $(\ln 2 - 1) - \frac{\theta}{2} + \frac{e^\theta}{2}$ as a function of y as well. Hence we have

$$(\ln 2 - 1) - \frac{\theta}{2} + \frac{e^\theta}{2} - w(\theta) = \ln(1 - e^{-y}) - 1 + \frac{y + e^{-y}}{y(1 - e^{-y})}.$$

Given a pre-specified rate R ($R < 1$ and $\delta = -\ln R$), the following three steps compute θ_k^L , $k = 1, \dots, M+1$:

Step 1. Solve the equation

$$\ln(1 - e^{-y}) - 1 + \frac{y + e^{-y}}{y(1 - e^{-y})} = \delta$$

for y . Compute θ_δ^L from y based on (12).

Step 2. Compute the integer M in (11). Let $\theta_{M+1}^L = \theta_\delta^L$.

Step 3. Iterate for $k = 1, \dots, M$ to solve the equation

$$\frac{1 - y - e^{-y}}{y(1 - e^{-y})} + \frac{\ln(2y)}{2} - \frac{\ln(1 - e^{-y})}{2} = \ln \sqrt{\frac{2}{e}} + k\delta$$

for y and, thereby, for θ_k^L through (12).

Note there is a unique solution for the equation in Step 1. Because $w(\theta)$ is strictly decreasing, each equation in Step 2 has a unique solution.

To summarize, we have the bound $b_w(\theta)$ given by

$$b_w(\theta) = \begin{cases} \frac{1}{2}(\ln 2 - 1) + \delta, & \forall \theta \geq \theta_1^L; \\ \frac{w(\theta_{k+1}^L) - w(\theta_k^L)}{\theta_{k+1}^L - \theta_k^L} \times \theta + \frac{\theta_k^L w(\theta_{k+1}^L) - \theta_{k+1}^L w(\theta_k^L)}{\theta_k^L - \theta_{k+1}^L}; & \forall \theta_{k+1}^L \leq \theta < \theta_k^L \ (k = 1, \dots, M); \\ (\ln 2 - 1) - \frac{\theta}{2} + \frac{e^\theta}{2}, & \forall \theta < \theta_{M+1}^L. \end{cases}$$

Constructing $b_s(\theta)$ given $\mathbf{R}$

We use a small number of tangent line segments to $s(\theta)$ to form a piece-wise linear function $b_s(\theta)$ that bounds $s(\theta)$ from above; see the red dashed line in the left panel of Figure 2. The basic idea is simple. The proposed algorithm starts from $\theta \equiv +\infty$. The right most segment is a horizontal line segment $b_s(\theta) = \ln \sqrt{2/e}$. We then find a set of change points $\theta_1^U, \dots, \theta_N^U$ ($\theta_1^U > \dots > \theta_N^U$). The first change point θ_1^U is obtained by solving $\ln \sqrt{2/e} = s(\theta_1^U) + \delta$. We use a segment of the tangent line to $s(\theta)$ which starts from $(\theta_1^U, s(\theta_1^U) + \delta)$ as the next piece of $b_s(\theta)$. The segment covers the interval between the first and the second change point (θ_1^U, θ_2^U) . The segment ends at $(\theta_2^U, s(\theta_2^U) + \delta)$. Denote by θ_1^T the point of tangency for the tangent line segment between θ_2^U and θ_1^U . The distance between the tangent line segment and $s(\theta)$ is no more than δ . Together with the computed $b_w(\theta)$, $r(\theta) \geq R$ for $\theta \in (\theta_2^U, \theta_1^U]$ and $\theta \in (\theta_1^U, +\infty)$.

We iterate through the above process to find the subsequent change points and compute the corresponding points of tangency ($\theta_2^T > \dots > \theta_N^T$) until a stopping criterion is satisfied at $\theta = \theta_N^T$. The stopping criterion is designed to directly control $r(\theta) > R$ over the interval $(-\infty, \theta_N^T]$. A stopping criterion can be defined by using the following function

$$c(\theta) = \frac{e^{w(\theta)}}{e^{s(\theta)} + \frac{2}{\sqrt{\alpha(\theta)}} e^{\frac{\alpha(\theta)}{2} - 1}} \quad (\theta \leq \theta_{M+1}^L), \quad (14)$$

where θ_{M+1}^L is defined by the algorithm creating $b_w(\theta)$, $\alpha(\theta) = e^\theta$. The stopping criterion is given by the two constraints $\theta_N^T < \theta_{M+1}^L$ and $c(\theta_N^T) \geq R$. Then we have $r(\theta) > R$, $\forall \theta \leq \theta_N^T$, because

$$r(\theta) \geq \frac{e^{w(\theta)}}{e^{b_s(\theta)} + \frac{2}{\sqrt{\alpha(\theta)}} e^{\frac{\alpha(\theta)}{2} - 1}} \geq \frac{e^{w(\theta)}}{e^{s(\theta_N^T)} + \frac{2}{\sqrt{\alpha(\theta)}} e^{\frac{\alpha(\theta)}{2} - 1}} \geq c(\theta_N^T) \geq R.$$

Next we show how to numerically solve for the change points θ_k^U and the points of tangency θ_k^T . Again we need an explicit expression of $s(\theta)$, though not directly as a function of θ . Following the proof of Theorem 1 in Appendix A, we define y implicitly as a function of θ ,

$$T(y) = \ln 2 - \ln y - \ln(e^y - 1) = \theta. \quad (15)$$

We then write $s(\theta)$ as an explicit function of y ,

$$S(y) = s(\theta) = \frac{1 + y - e^y}{y(e^y - 1)} + \frac{\ln(2y)}{2} - \frac{\ln(e^y - 1)}{2}.$$

We write its derivative $s'(\theta)$ as an explicit function of y ,

$$P(y) = s'(\theta) = \frac{1 + y - e^y}{y(e^y - 1)} + \frac{1}{2}.$$

Given a pre-specified rate R ($R < 1$), the following algorithm computes the change points $\theta_1^U, \dots, \theta_N^U$ ($\theta_1^U > \dots > \theta_N^U$) and the points of tangency $\theta_1^T, \dots, \theta_N^T$ ($\theta_1^T > \dots > \theta_N^T$).

Step 1. Obtain the first change point by solving the equation

$$S(y) = \ln \sqrt{\frac{2}{e}} - \delta$$

for y . Compute θ_1^U from y based on (15).

Step 2. Given the k -th change point θ_k^U , use the tangent line to $s(\theta)$ that passes through $(\theta_k^U, s(\theta_k^U) + \delta)$. Obtain the point of tangency θ_k^T ($\theta_k^T < \theta_k^U$), by solving

$$P(y) \times \theta_k^U + S(y) - P(y) \times T(y) = s(\theta_k^U) + \delta$$

for y . Compute θ_k^T from y based on (15).

If $\theta_k^T < \theta_{M+1}^L$, compute $c(\theta_k^T)$. If $c(\theta_k^T) \geq R$, stop. Otherwise continue.

Step 3. Find the $(k+1)$ -th change point θ_{k+1}^U ($\theta_{k+1}^U < \theta_k^T$) by solving

$$s'(\theta_k^T) \times T(y) + s(\theta_k^T) - s'(\theta_k^T) \times \theta_k^T = S(y) + \delta$$

for y . Compute θ_{k+1}^U from y based on (15).

Go to Step 2.

Since $s(\theta)$ is strictly increasing toward $\ln \sqrt{\frac{2}{e}}$, the equation in Step 1 has a unique solution. In addition $s(\theta)$ is strictly concave. The distance between a tangent line segment and $s(\theta)$ is strictly increasing for θ larger than the point of tangency, and strictly decreasing for θ smaller than the point of tangency. Hence each of the equations in Step 2 and 3 has a unique solution. Assume the above algorithm returns N change points $\theta_1^U, \dots, \theta_N^U$ from right to left, and N points of tangency $\theta_1^T, \dots, \theta_N^T$ from right to left. Let $\theta_{N+1}^U = -\infty$. The computed function $b_s(\theta)$ is given by

$$b_s(\theta) = \begin{cases} \frac{1}{2}(\ln 2 - 1), & \text{if } \theta > \theta_1^U; \\ s'(\theta_k^T) \times \theta + s(\theta_k^T) - s'(\theta_k^T)\theta_k^T, & \text{if } \theta_{k+1}^U < \theta \leq \theta_k^U \text{ } (k = 1, \dots, N). \end{cases}$$

2.4. Acceptance rate

We set $R = 0.9$ and implement the algorithms in Section 2.3 to obtain the corresponding $b_w(\theta)$ and $b_s(\theta)$. The precision of the parameter values is 10^{-8} . The resulting functions $b_w(\theta)$ and $b_s(\theta)$ are shown in Tables 1 and 2. The function $b_s(\theta)$ has 3 segments and $b_w(\theta)$ has 4 segments respectively as shown in Figure 2. Both the best acceptance rate based on the

Range of θ	$b_w(\theta)$
$\theta < 0.20931492$	$\ln \frac{2}{e} - \frac{\theta}{2} + \frac{e^\theta}{2} = -0.30685282 - \frac{\theta}{2} + \frac{\alpha}{2}$
$0.20931492 \leq \theta < 0.52122324$	$-0.13546023 \times \theta + 0.12789964$
$0.52122324 \leq \theta < 1.76421669$	$-0.08476353 \times \theta + 0.10147534$
$\theta \geq 1.76421669$	$1/2(\ln 2 - 1) - \ln R = -0.04806589$

Table 1: $b_w(\theta)$ given $R = 0.9$.

Range of θ	$b_s(\theta)$
$\theta \leq -3.33318991$	$0.30625300 \times \theta + 0.27127295$
$-3.33318991 < \theta \leq 1.44893155$	$0.12465180 \times \theta - 0.33403833$
$\theta > 1.44893155$	$1/2(\ln 2 - 1) = -0.15342641$

Table 2: $b_s(\theta)$ given $R = 0.9$.

Figure 3: Best acceptance rate using the exact maximum and minimum (solid black line) and acceptance rate using the upper bounds (dashed red line) specified by Tables 1 and 2 for the Algorithm 1.

exact maximum and minimum $\exp(s(\theta))$ and $\exp(w(\theta))$ and the acceptance rate based on the upper bounds $\exp(b_s(\theta))$ and $\exp(b_w(\theta))$ in Tables 1 and 2 are shown in Figure 3. The uniform region and the corresponding rectangular cover/envelope given by $\exp(b_s(\theta))$ and $\exp(b_w(\theta))$ for $\alpha = 0.001, 0.1, 1$, and 10 are shown in Figure 1.

It should be noted that more segments are needed for larger R . For example, for $R = 0.95$, the corresponding $b_s(\theta)$ has 4 segments and $b_w(\theta)$ has 7 segments. With larger acceptance rate, more segments in bounds will increase the initial parameter set-up time.

2.5. The algorithm

Using the bounds $b_w(\theta)$ and $b_s(\theta)$ in Tables 1 and 2 for $R = 0.9$, Algorithm 1 demonstrates

Algorithm 1 Generate n gamma random numbers.

```

 $\theta = \ln(\alpha)$  and  $c = \sqrt{\alpha}$ 
Compute  $b_s(\theta)$  and  $B.\max = \exp\{b_s(\theta)\}$ 
Compute  $b_w(\theta)$  and  $B.\min = -\exp\{b_w(\theta)\}$ 
 $k = 1$ 
while  $k \leq n$  do
   $u = \text{Uniform}(0, 1)$ 
   $v = \text{Uniform}(0, 1) \times (B.\max - B.\min) + B.\min$ 
   $t = v/u$  and  $t_1 = \exp(t/c + \theta)$ 
  if  $2 \ln(u) \leq \alpha + c \times t - t_1$  then
    Deliver  $t_1$  if  $\alpha \geq 0.01$ ; Otherwise deliver  $t/c + \theta$ .
     $k = k + 1$ 
  end if
end while

```

how to generate gamma random numbers using the proposed approach. When $\alpha < 0.01$, the algorithm returns the generated random numbers on logarithmic scale.

3. The algorithm for shape parameter smaller or equal to 1

For $\alpha \leq 1$, we consider the logarithmic transformation:

$$T^* = \alpha \ln X$$

or, equivalently $X = \exp(T^*/\alpha)$. The density function of T^* is

$$g(t) = \frac{\exp(t - e^{t/\alpha})}{\alpha \Gamma(\alpha)} \quad (-\infty < t < \infty).$$

Here we propose another ratio of uniforms algorithm based on T^* . It is limited to $\alpha \leq 1$, but it is simpler and faster than Algorithm 1 for $\alpha \leq 1$. This second ROU method is outlined below. Let

$$h_\alpha^*(t) = \exp(t - e^{t/\alpha}).$$

The ROU region is

$$\mathcal{C}_\alpha^* = \{(u, v) : 0 \leq u \leq \sqrt{h^*(t)}, -\infty < t = \frac{v}{u} < \infty\}.$$

The size of the ROU region $\mathcal{C}_\alpha^*$ is

$$|\mathcal{C}_\alpha^*| = \frac{\Gamma(\alpha)\alpha}{2}.$$

Because $\max \sqrt{h^*(t)} = (\frac{\alpha}{e})^{\alpha/2}$, we have

$$0 \leq u \leq \left(\frac{\alpha}{e}\right)^{\alpha/2}. \quad (16)$$

Using the inequality $\ln x \leq x - 1$, we have $\forall t > 0$,

$$t\sqrt{h^*(t)} = te^{\frac{t}{2} - \frac{1}{2}e^{t/\alpha}} \leq te^{\frac{t}{2} - \frac{et}{2\alpha}} \leq \frac{2\alpha}{e(e - \alpha)}.$$

Figure 4: Best acceptance rate based on the exact maximum and minimum of u and v for each α (solid black line) and acceptance rate for Algorithm 2 using the intervals specified by Equations 16 and 17 (dashed red line).

For all $t < 0$, we have

$$t\sqrt{h^*(t)} \geq te^{\frac{t}{2}} \geq -\frac{2}{e}.$$

Therefore we have the range for v :

$$-\frac{2}{e} \leq v \leq \frac{2\alpha}{e(e-\alpha)}. \quad (17)$$

Equations 16 and 17 define a rectangle that covers the ROU region $\mathcal{C}_\alpha^*$. The uniform region $\mathcal{C}_\alpha^*$ and its cover are similar to those shown in Figure 1. We have a simple algorithm (Algorithm 2) to generate gamma random numbers for $\alpha \leq 1$. Again when $\alpha < 0.01$ the algorithm returns the generated random numbers on logarithmic scale. The resulting acceptance rate is shown in

Algorithm 2 Generate n gamma random numbers given $\alpha \leq 1$.

$u.\text{max} = (\alpha/e)^{\alpha/2}$, $v.\text{min} = -2/e$, $v.\text{max} = 2\alpha/e/(e-\alpha)$.

$k = 1$

while $k \leq n$ **do**

$u = u.\text{max} \times \text{Uniform}(0, 1)$

$t = (\text{Uniform}(0, 1) \times (v.\text{max} - v.\text{min}) + v.\text{min})/u$

$t_1 = e^{t/\alpha}$

if $2 \ln(u) \leq t - t_1$ **then**

 Deliver t_1 if $\alpha \geq 0.01$; Otherwise deliver t/α .

$k = k + 1$

end if

end while

Figure 4. Algorithm 2 (used as an example in [Liang, Liu, and Carroll 2010](#)) reaches maximum acceptance rate 0.7554 when $\alpha = 0.33$.

4. Conclusion

In this article we proposed two algorithms using the ratio-of-uniforms method. Both algorithms only require uniform random numbers. One algorithm covers the entire positive shape parameter value range. The second algorithm has better performance than the first one though it is limited to shape parameter smaller or equal to 1.

For modern large scale simulation, the random numbers need a very long period so the same sequence will not repeat itself. The Mersenne Twister algorithm for uniform random number generation has a period $2^{19937} - 1$ and is a popular high quality and efficient uniform random number generator. On the other hand it is slower than a uniform random number generator with a very short period, such as a multiplicative congruential generator used by older versions of MATLAB, which has period $2^{31} - 1$ ([Moler 2004](#)). The adoption of the Mersenne Twister algorithm for software packages such as R ([R Core Team 2013](#)) and MATLAB ([The MathWorks, Inc. 2011](#)) leads to slower uniform random number generation, and consequently slows down the generation of random numbers from other distributions, such as normal and exponential. While our proposed algorithms only require uniform random numbers, some existing gamma random number generation algorithms, such as [Marsaglia and Tsang \(2000\)](#) and [Ahrens and Dieter \(1982\)](#), require normal and exponential random numbers to achieve high acceptance rates. Furthermore, previously proposed gamma random number generators are limited to a certain range of shape parameter values. Additional steps are needed to cover the rest of the shape parameter value range. For example, [Marsaglia and Tsang \(2000\)](#) needs one more uniform random number for $\alpha < 1$.

We also conducted a simulation study to compare our methods to existing methods¹. Since they may not be reproducible, due to different machines and operating systems used and their performance, the detailed timing results are not reported here. The results showed improvements of 25–35% in the $\alpha < 1$ and 0–20% in the $\alpha \geq 1$ case from the existing methods.

Acknowledgments

The authors would like to thank the reviewers and the associate editor for their very helpful comments. This work is partially supported by NSF DMS-0904548, NSF DMS-1007678, NSF DMS-1208841, NSF BCS-1244708, NSF DMS-1228348, ARO W911NF-12-1-0558, and DoD MURI W911NF-08-1-0238.

¹ In the simulation study we generated 50 million gamma random numbers for each of a set of selected shape parameter values 0.01, 0.25, 0.5, 0.8, 1, 1.25, 3, 5, 10, and 100. We compared our proposed algorithms with CF80GT of [Cheng and Feast \(1980\)](#) with $n = 2$, CF80GBH of [Cheng and Feast \(1980\)](#) with $n = 4$, AD82 of [Ahrens and Dieter \(1982\)](#) for $\alpha \geq 1$, AD74 of [Ahrens and Dieter \(1974\)](#) for $\alpha < 1$, and MT00 of [Marsaglia and Tsang \(2000\)](#), where gamma random numbers for $\alpha < 1$ are generated as $\text{Gamma}(\alpha) = \text{Gamma}(\alpha + 1) \times \text{Uniform}(0, 1)^{1/\alpha}$. The computer used in the simulation is a 2.4 GHz Intel Core 2 Duo processor MacBook. The computer runs Mac Os/X Snow Leopard with 3MB L2 cache, 4GB memory, and 800MHz frontside bus. All the algorithms were coded in C++. The GNU g++ compiler was used. The initial parameter values were saved for repeated draws of gamma random numbers.

References

- Ahrens JH, Dieter U (1974). “Computer Methods for Sampling from Gamma, Beta, Poisson and Binomial Distributions.” *Computing*, **12**(3), 223–246.
- Ahrens JH, Dieter U (1982). “Generating Gamma Variates by a Modified Rejection Technique.” *Communications of the ACM*, **25**(1), 47–54.
- Cheng RCH, Feast GM (1980). “Gamma Variate Generators with Increased Shape Parameter Range.” *Communications of the ACM*, **23**(7), 389–395.
- Devroye L (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag, New York.
- Kinderman AJ, Monahan JF (1977). “Computer Generation of Random Variables Using the Ratio of Uniform Deviates.” *ACM Transactions on Mathematical Software*, **3**(3), 257–260.
- Liang F, Liu C, Carroll RJ (2010). *Advanced Markov Chain Monte Carlo Methods*. John Wiley & Sons, New York.
- Marsaglia G, Tsang WW (2000). “A Simple Method for Generating Gamma Variables.” *ACM Transactions on Mathematical Software*, **26**(3), 363–372.
- Moler C (2004). *Numerical Computing with MATLAB*. Society for Industrial and Applied Mathematics (SIAM).
- R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
- Tanizaki H (2008). “A Simple Gamma Random Number Generator for Arbitrary Shape Parameters.” *Economics Bulletin*, **3**(7), 1–10.
- The MathWorks, Inc (2011). *MATLAB – The Language of Technical Computing, Version R2011b*. The MathWorks, Inc., Natick, Massachusetts. URL <http://www.mathworks.com/products/matlab/>.
- Wakefield JC, Gelfand AE, Smith AFM (1991). “Efficient Generation of Random Variates via the Ratio-of-Uniforms Method.” *Statistics and Computing*, **1**(2), 129–133.

A. Proofs

A.1. The proof of Theorem 1

Theorem 1. *Let*

$$s(\theta) = \max_{t>0} \ln(t\sqrt{h_\alpha(t)}).$$

Then, as a function of $\theta \in (-\infty, \infty)$, $s(\theta)$ is strictly increasing and concave. Furthermore,

$$\lim_{\theta \rightarrow -\infty} \frac{s(\theta)}{\theta} = \frac{1}{2},$$

$$\lim_{\theta \rightarrow \infty} s(\theta) = \ln \sqrt{\frac{2}{e}} = \frac{1}{2}(\ln 2 - 1).$$

Proof: Let $y = t/\sqrt{\alpha}$ and write

$$k(y, \alpha) = 2 \ln(t\sqrt{h_\alpha(t)}) = \alpha + \ln \alpha + 2 \ln y + \alpha y - \alpha e^y \quad (y > 0).$$

Then

$$s(\theta) = \frac{1}{2} \max_{y>0} k(y, \alpha),$$

where $\theta = \ln \alpha \in \mathcal{R}$. It follows from

$$\frac{\partial k(y, \alpha)}{\partial y} = \frac{2}{y} + \alpha(1 - e^y) \quad (y > 0; \alpha > 0)$$

that the function $s(\theta)$ is given by

$$2s(\theta) = e^\theta + \theta + 2 \ln y + e^\theta(y - e^y) \quad (\theta \in \mathcal{R}) \quad (18)$$

where y is a function of θ defined implicitly by

$$e^\theta = \frac{2}{y(e^y - 1)} \quad (19)$$

a one-to-one mapping: $\theta \mapsto y$ from $\mathcal{R}$ to $\{y : y > 0\}$.

Routine algebraic operations lead to the first-order derivative of $2s(\theta)$ with respect to θ :

$$\begin{aligned} \frac{\partial 2s(\theta)}{\partial \theta} &= 1 + e^\theta(1 + y - e^y) + \left[\frac{2}{y} + e^\theta(1 - e^y) \right] \frac{1}{\frac{\partial \theta}{\partial y}} \\ &\stackrel{(19)}{=} 1 + e^\theta(1 + y - e^y), \end{aligned} \quad (20)$$

where

$$\frac{\partial \theta}{\partial y} = -\frac{1}{y} - \frac{e^y}{e^y - 1}. \quad (21)$$

To show that (20) is positive for all $y > 0$ (or all $\theta \in \mathcal{R}$), write (20) in terms of y based on (19)

$$1 - \frac{2}{y} + \frac{2}{e^y - 1} = \frac{ye^y - 2e^y + 2 + y}{y(e^y - 1)}. \quad (22)$$

The numerator of (22) has the first-order derivative

$$ye^y - e^y + 1,$$

which is strictly increasing in y because its first-order derivative ye^y is positive for all $y > 0$. Note that the numerator of (22) tends to zero as y goes to zero. This implies that the numerator of (22) is positive for all y . The denominator of (22) is also positive for all $y > 0$, due to the fact that $e^y > 1$. As a result, (22) or equivalently (20) is positive for all $y > 0$. That is, $2s(\theta)$ and, thereby, $s(\theta)$ is increasing.

The concavity of $s(\theta)$ or $2s(\theta)$ is to be proved by showing that its second-order derivative or the first-order derivative of (20) or (22) with respect to θ is negative. The first-order derivative of (22) with respect to θ is given by

$$2 \left[\frac{1}{y^2} - \frac{e^y}{(e^y - 1)^2} \right] \frac{1}{\frac{\partial \theta}{\partial y}}.$$

Since $\frac{\partial \theta}{\partial y} < 0$, the concavity result follows if the bracketed term in the above expression is positive. Note that the bracketed term

$$\frac{1}{y^2} - \frac{e^y}{(e^y - 1)^2} = \left[\frac{1}{y} + \frac{e^{y/2}}{(e^y - 1)} \right] \left[\frac{1}{y} - \frac{e^{y/2}}{(e^y - 1)} \right]$$

has the same sign as the second factor

$$\frac{1}{y} - \frac{e^{y/2}}{(e^y - 1)} = \frac{e^y - 1 - ye^{y/2}}{y(e^y - 1)}. \quad (23)$$

The numerator of (23) is positive because (i) its first-order derivative,

$$e^{y/2} [e^{y/2} - 1 - y/2],$$

is positive for all $y > 0$, and (ii) it approaches zero as $y \rightarrow 0$. Note again that $y(e^y - 1)$, the denominator of (23), is positive. Thus, the function $s(\theta)$ is strictly concave.

The identities (7) and (6) can be easily established by applying standard algebraic operations to (18) and (19).

A.2. The proof of Theorem 2

Theorem 2. *Let*

$$w(\theta) = \max_{t < 0} \ln(|t| \sqrt{h_\alpha(t)}).$$

Then, as a function of $\theta \in (-\infty, \infty)$, $w(\theta)$ is strictly decreasing and convex. Furthermore,

$$\lim_{\theta \rightarrow -\infty} \frac{w(\theta)}{\theta} = -\frac{1}{2},$$

$$\lim_{\theta \rightarrow \infty} w(\theta) = \ln \sqrt{\frac{2}{e}} = \frac{1}{2}(\ln 2 - 1).$$

Proof: The proof is similar to that of Theorem 1. Let $y = -t/\sqrt{\alpha}$ and write

$$k(y, \alpha) = 2 \ln(|t| \sqrt{h_\alpha(t)}) = \alpha + \ln \alpha + 2 \ln y - \alpha y - \alpha e^{-y} \quad (y > 0).$$

Then

$$w(\theta) = \frac{1}{2} \max_{y>0} k(y, \alpha)$$

where $\theta = \ln \alpha \in \mathcal{R}$. It follows from

$$\frac{\partial k(y, \alpha)}{\partial y} = \frac{2}{y} - \alpha(1 - e^{-y}) \quad (y > 0; \alpha > 0)$$

that the function $w(\theta)$ is given by

$$2w(\theta) = e^\theta + \theta + 2 \ln y - e^\theta(y + e^{-y}) \quad (\theta \in \mathcal{R}), \quad (24)$$

where y is a function of θ defined implicitly by

$$e^\theta = \frac{2}{y(1 - e^{-y})} \quad (25)$$

a one-to-one mapping: $\theta \mapsto y$ from $\mathcal{R}$ to $\{y : y > 0\}$.

The first-order derivative of $2w(\theta)$ is negative and the second-order derivative of $2w(\theta)$ is positive $\forall y > 0$. Therefore it is strictly decreasing and convex. The identities (9) and (8) can be easily established by applying standard algebraic operations to (24) and (25).

Affiliation:

Bowei Xi, Chuanhai Liu
 Department of Statistics
 Purdue University
 250 N. University Street
 West Lafayette, IN 47907, United States of America
 E-mail: xbw@purdue.edu, chuanhai@purdue.edu

Kean Ming Tan
 Department of Biostatistics
 University of Washington
 Seattle, WA 98195-7232, United States of America
 E-mail: keanming@uw.edu