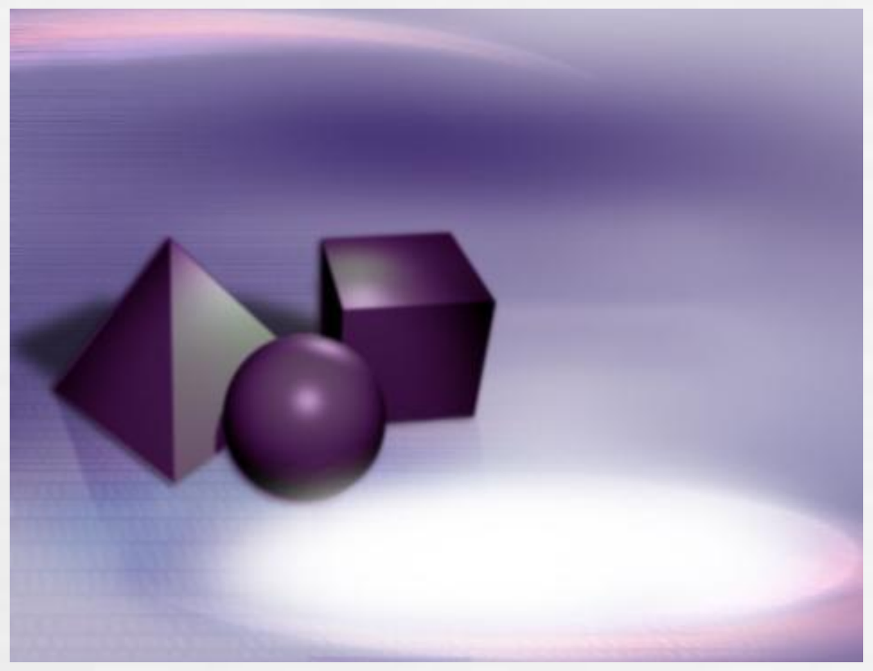


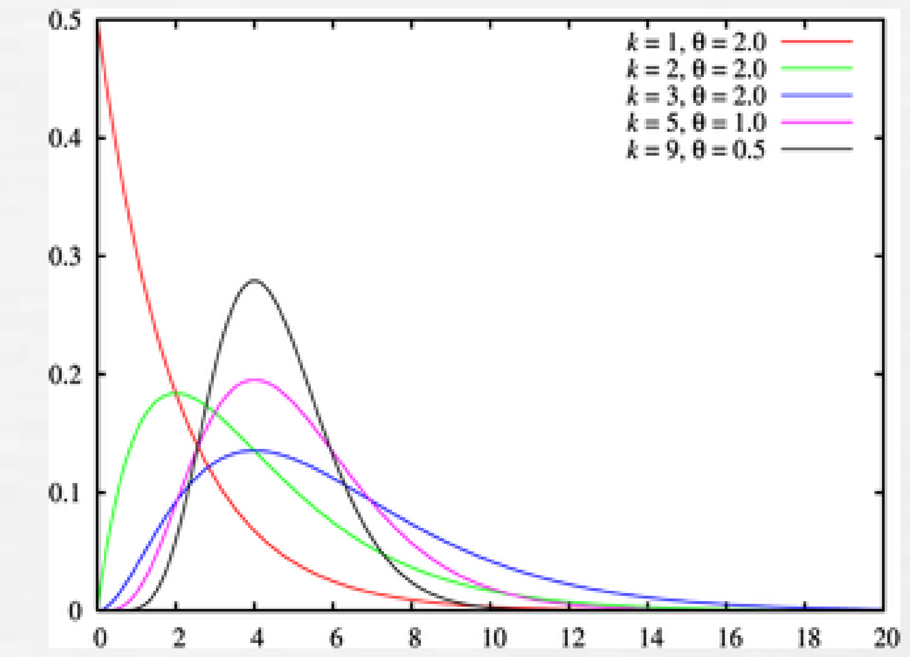
Two New Ratio-Of-Uniforms Gamma Random Number Generators

Kean Ming Tan, Dr. Bowei Xi, Dr. Chuanhai Liu

Kean Ming Tan, Department of Statistics, Purdue University, tank@purdue.edu

Professor Bowei Xi, Department of Statistics, Purdue University, xbw@purdue.edu

Professor Chuanhai Liu, Department of Statistics, Purdue University, chuanhai@purdue.edu



Abstract:

Efficient and high quality random number generators are critical to Monte Carlo simulation which is widely used in different fields of studies. Gamma distribution is one of the fundamental distributions that is used to generate random numbers from other distributions such as Chi-square, Student-t, F, and Beta. The algorithms proposed in the past for gamma random number generation are all limited to certain range of shape parameter values. Two simple algorithms to generate gamma random numbers are proposed in this presentation. Both of the proposed algorithms use the ratio-of-uniforms method. The first algorithm applies to all positive shape parameter value without limitation. The algorithm is very simple and has good performance compared with the existing algorithms. The second algorithm is limited to shape parameter smaller or equal to 1. However it has better performance compared with the first algorithm in that limited shape parameter range.

Introduction:

- ROU1 and ROU2 are proposed using ratio-of-uniform method
- Ratio-of-uniform method: only uniform random generator is used
- Logarithmic transformation are used for both ROU1 and ROU2
- Transformation controls the acceptance rate and range of shape parameter values for ROU1 and ROU2
- ROU1 is not limited to a certain interval
- ROU2 is valid only for α ≤ 1
- ROU1 and ROU2 have reasonable acceptance rate and timing results

Procedures (ROU1):
1 Transformed random variable T=$\sqrt{\alpha}\ln(\frac{X}{\alpha})$
2 Then we have $X = \alpha e^{\frac{T}{\sqrt{\alpha}}}$
3 We have the density $g(t) = \frac{1}{\Gamma(\alpha)}\alpha^{\alpha-\frac{1}{2}}e^{\sqrt{\alpha}t-\alpha e^{\frac{t}{\sqrt{\alpha}}}}$
4 Let $h(t) = e^{\sqrt{\alpha}t-\alpha e^{\frac{t}{\sqrt{\alpha}}}+\alpha}$, then $T = \frac{V}{U}$ has the desired density $g(t)$
5 Finding maximum and minimum, $v_{max}(\alpha) = max_{t>0}t\sqrt{h_\alpha(t)}$, $v_{min}(\alpha) = min_{t<0}t\sqrt{h_\alpha(t)}$, and $u_{max}(\alpha) = max_{t>0}\sqrt{h_\alpha(t)}$
6 Generate V $\in Uniform(v_{min}, v_{max})$, U $\in Uniform(0, u_{max})$, then T=V/U where T is the desired random number

Procedures (ROU2):
1 Transformed random variable T=αlnX
2 Then we have $X = e^{\frac{T}{\alpha}}$
3 We have the density $g(t) = e^{t-e^{\frac{t}{\alpha}}}/(\alpha\Gamma(\alpha))$
4 Let $h(t) = e^{t-e^{\frac{t}{\alpha}}}$, then $T = \frac{V}{U}$ has the desired density $g(t)$
5 Taking first derivative, we obtained $v_{max} = 2\alpha/e/(e-\alpha)$, $v_{min} = -2/e$, and $u_{max} = (\alpha/e)^{\alpha/2}$
6 Generate V $\in Uniform(v_{min}, v_{max})$, U $\in Uniform(0, u_{max})$, then T=V/U where T is the desired random number

α	0.01	0.25	0.5	1	1.25	3	5	10
MT00	20.39	20.31	20.25	13.38	13.29	13.09	13.04	13.02
AD82	N/A	N/A	N/A	16.51	16.00	15.95	11.36	10.98
AD74	9.15	12.55	13.68	N/A	N/A	N/A	N/A	N/A
ROU1	15.44	14.86	14.48	13.71	13.31	12.90	14.28	14.31
ROU2	14.50	12.54	12.61	13.42	N/A	N/A	N/A	N/A

Table 1: The timing results (seconds) of generating 50 million Gamma Random Numbers for each of the algorithm.

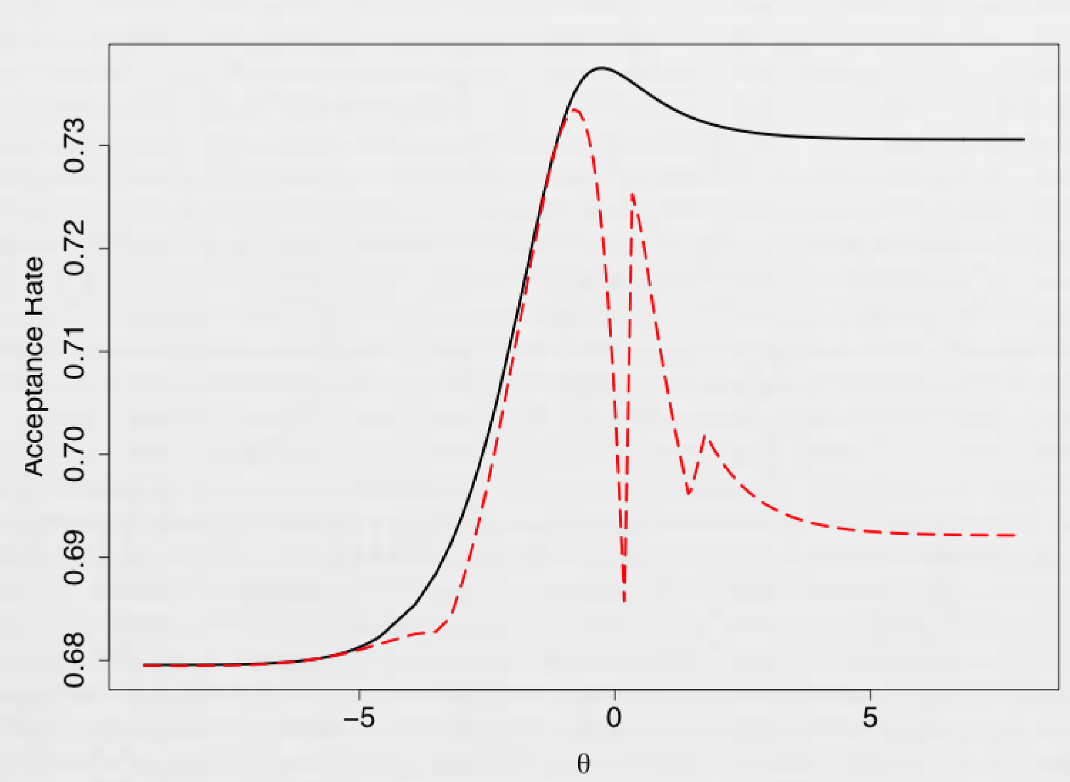


Figure 1: Acceptance rate for ROU1.

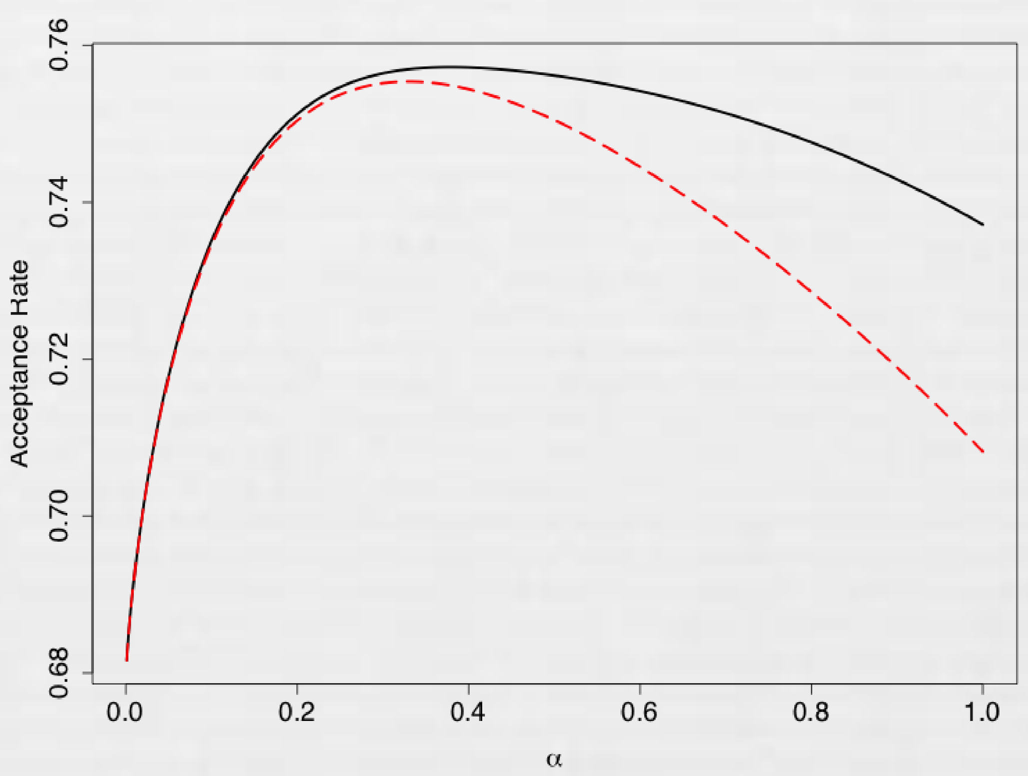


Figure 2: Acceptance rate for ROU2.

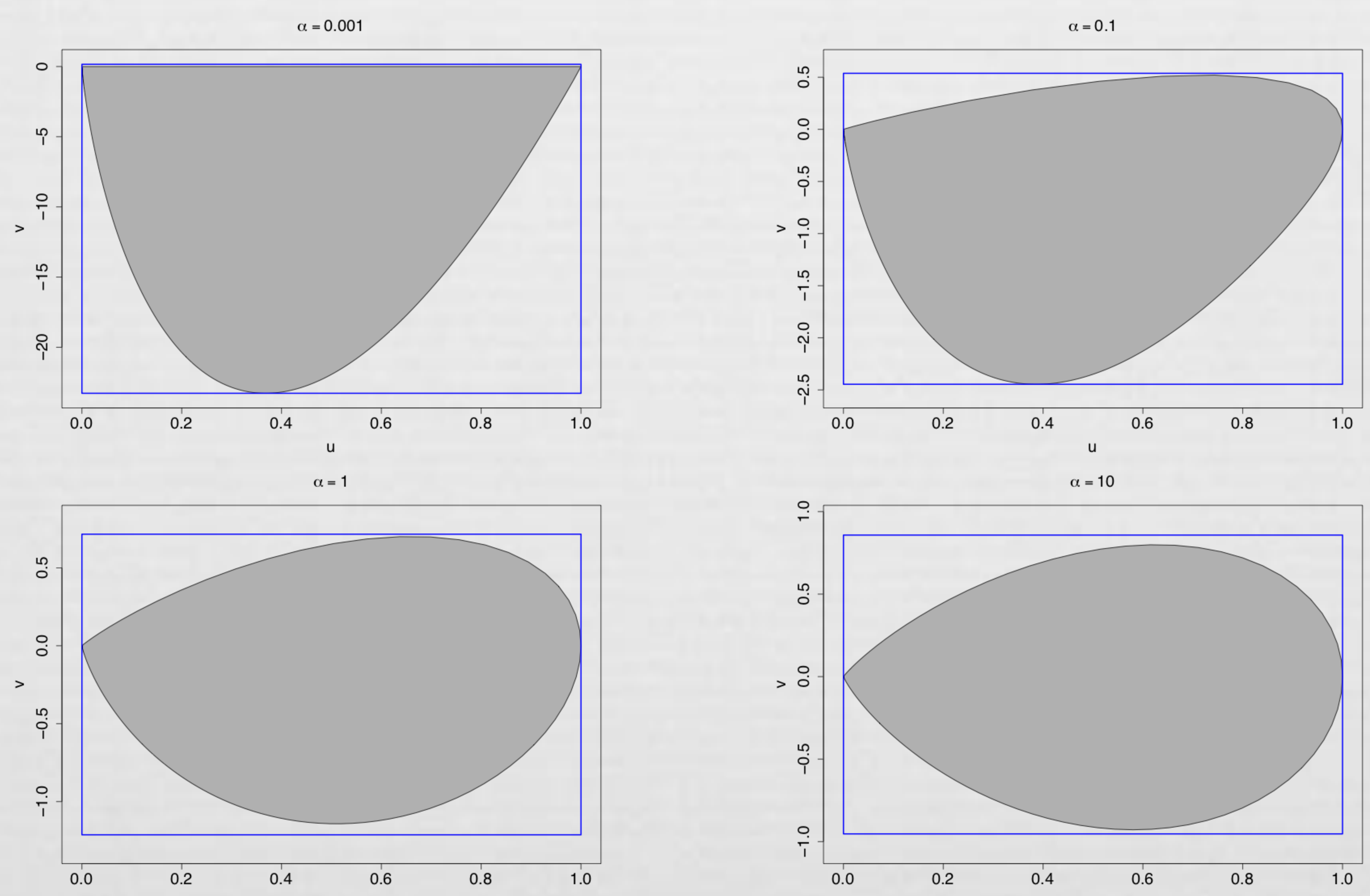


Figure 3: Plot of f(x) vs x*f(x) based on Ratio-of-Uniform Method for ROU1.

Algorithm ROU1 : 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 = Uniform(0,1)
  v= Uniform(0,1)x(B.max-B.min)+B.min
  $t = v/u$ and $t_1 = e^{t/c+\theta}$
  if 2 x ln $(u) \leq \alpha + c$ x t - t_1 **then**
    Deliver t_1 if $\alpha \geq 0.01$; Otherwise deliver $t/c + \theta$.
    $k = k + 1$
  end if
end while

Algorithm ROU2 : Generate n Gamma Random Numbers given alpha <= 1

u.max=$(\alpha/e)^{\alpha/2}$, v.min=$-2/e$, v.max=$2\alpha/e/(e-\alpha)$.
$k = 1$
while $k \leq n$ **do**
  u = u.max x Uniform(0,1)
  t=(Uniform(0,1) x (v.max-v.min)+v.min)/u
  $t_1 = e^{t/\alpha}$
  if 2 x 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

Results/Conclusion:

- Normal and Uniform random numbers generators are required in MT00
- Normal, Uniform, and Exponential random number generators are required for AD82 and AD74
- MT00, AD82 and AD72 are complicated algorithm and difficult to code
- ROU1 and ROU2 only need uniform random number
- ROU1 and ROU2 are faster than MT00 when α<1
- ROU1 has high acceptance rate for α between 1.25 and 3
- ROU1 is faster than AD82 for α < 4
- ROU2 is faster than AD74 for α between 0.25 and 0.8
- ROU1 and ROU2 are very simple to code and speed are comparable to MT00, AD82, and AD74

1

References:

Ahrens JH, Dieter U (1974). "Computer methods for sampling from gamma, beta, Poisson and Binomial Distributions." *Computing*, **12**, 223-246

Ahrens JH, Dieter U (1982). "Generating gamma variates by a modified rejection technique. "*Communications of the ACM*, **25**, 47-52.

Marsaglia G, Tsang WW (2000). "A Simple Method for Generating Gamma Variables." *ACM Transactions on Mathematical Software*, **26**, 363-372.

Marsaglia G, Tsang WW (2000). "The Ziggurat Method for Generating Random Variables." *Journal of Statistical Software*, ***5***, *1-7.*

Tanizaki H (2008). "A Simple Gamma Random Number Generator for Arbitrary Shape Parameters." *Eonomics Bulletin*, **3**, 1-10.