

Progress on the generation of random numbers with a parallel algorithm using the gamma function $X \sim \Gamma$.

Soon Yung Jun
Oscar Chaparro

**Centro de Investigación en Computación
Instituto Politécnico Nacional
México**



Gamma Distribution Function

Random Variable X is considered under Gamma Distribution Function

$$X \sim \Gamma(v, \alpha, \beta) \equiv \frac{\beta^\alpha v^{\alpha-1} e^{-\beta v}}{\Gamma(\alpha)} \quad 1$$

$$v, \alpha, \beta > 0$$

α is called shape parameter, β is called scale parameter

¹ Taken from (Pishro-Nik, 2014).

Main assumptions

We assume that a Random variable under Gamma distribution \tilde{X} with parameters $(\alpha, \beta = 1)$ is generated ($\tilde{X} \sim \Gamma(\alpha, 1)$). Then a Random variable under Gamma distribution \tilde{Y} with parameters (α, β) , $\tilde{Y} \sim \Gamma(\alpha, \beta)$ can be generated as:
 $\tilde{Y} = \frac{1}{\beta} \tilde{X}$ (Martino and Luengo, 2013).

The algorithms for the generation of random gamma variables with shape parameter $\alpha > 1$ and shape parameter $\alpha \leq 1$ are different.

Probability Gamma Distribution Plots

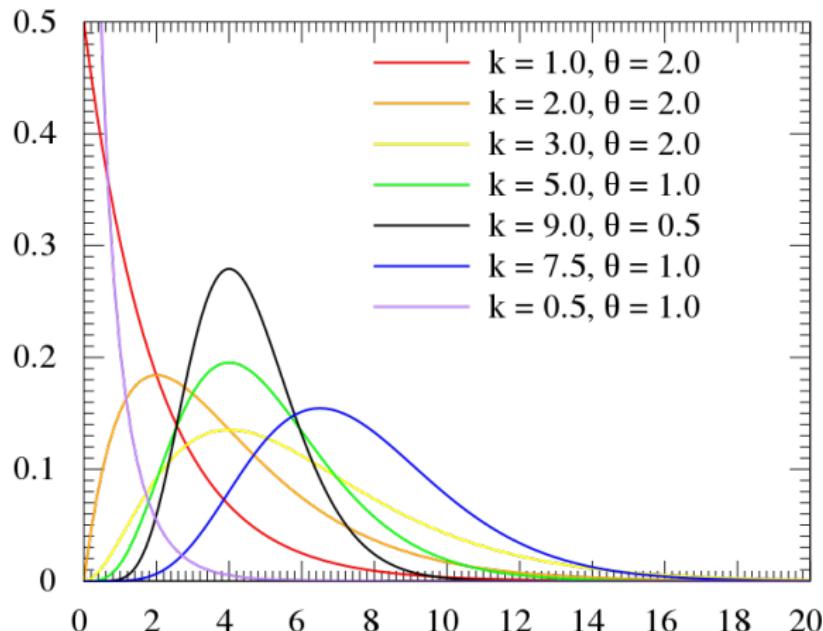
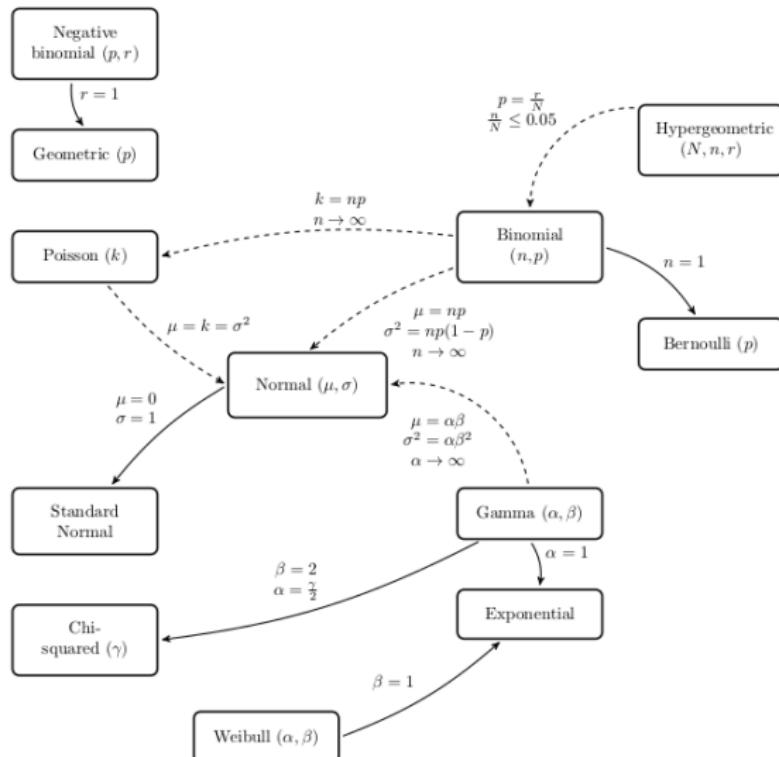


Figure: Gamma Probability Distribution Functions Normalized taken from the Web, where $k = \alpha, \theta = \beta$.

Equivalence between several Probability Distribution Functions (Pishro-Nik, 2014)



We will use

$$\text{Standard Deviation} = \frac{\sqrt{\alpha}}{\beta}$$

$$\text{Variance} = \frac{\alpha}{\beta^2}$$

$$\text{Mean} = \frac{\alpha}{\beta}$$

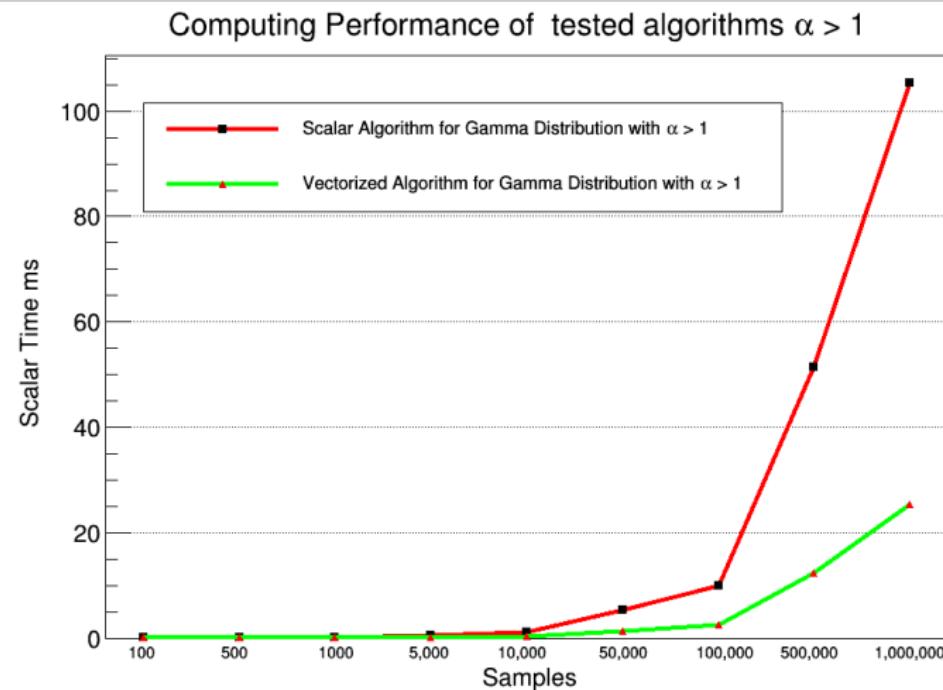
$$\text{Skewness} = \frac{2}{\sqrt{\alpha}}$$

$$\text{Kurtosis} = \frac{6}{\alpha}$$

Generalities of Proposed Algorithms for $\alpha > 1$

- The algorithm implemented is a variant from Marsaglia and Tsang's method (Marsaglia and Tsang, 2000).
- First the algorithm generates one Gauss random number $\sim \tilde{G}$,
 $a = \alpha - 1/3$ and then $b = 1/\sqrt{9 * a}$.
- Then the method applies $\tilde{Y} = (1 + G * b)^3$, where
 $\tilde{Y} \sim \Gamma(\alpha, 1)$.

Computing Performance of $\alpha = 4.0$ and $\beta = 1.0$ Algorithm using processor Intel(R) Core(TM) i7-4510U CPU 2.00GHz with 10 samples



Mean and Standard Deviation of Gamma distribution of the algorithm with $\alpha > 1$ with 1,000,000 samples and their relative errors $\frac{V_{calc} - V_{obt}}{V_{calc}} * 100$

α	β	Mean α/β	Relative error of Mean	Standard Deviation $\frac{\sqrt{\alpha}}{\beta}$	Relative error of Standard Deviation
2	1	2	0.01%	1.414	2.1%
3	1	3	0.06%	1.732	1.9%
5	1	5	0.06%	2.236	1.07%
10	0.5	20	0.05%	6.324	0.63%
8	3	2.666	0.04%	0.943	0.64%
10	1	10	0.02%	3.162	0.63%
12	1	12	0.01%	3.464	0.02%
50	0.5	100	0.02%	14.142	0.13%
100	1	100	0.01%	10	0.1%
200	0.5	400	0.01%	28.284	0.06%

Skewness and Kurtosis of Gamma distribution of the algorithm with $\alpha > 1$ with 1,000,000 samples and their relative errors $\frac{V_{calc} - V_{obt}}{V_{calc}} * 100$

α	β	Skewness $\frac{2}{\sqrt{\alpha}}$	Relative error of Skewness	Kurtosis $\frac{6}{\alpha}$	Relative error of Kurtosis
2	1	1.414	3.09%	3	14%
3	1	1.155	2.3%	2	11%
5	1	0.894	1.9%	1.2	6.5%
10	0.5	0.632	0.95%	0.6	5%
8	3	0.707	2.12%	0.75	4%
10	1	0.632	1.26%	0.6	2.2%
12	1	0.577	1.04%	0.5	8.2%
50	0.5	0.283	0.01%	0.12	15%
100	1	0.2	0.25%	0.06	6.6%
200	0.5	0.141	0.01%	0.03	18%

Examples of the Gamma distribution with $\alpha = 5.0$ and $\beta = 1.0$ with 500,000 samples.

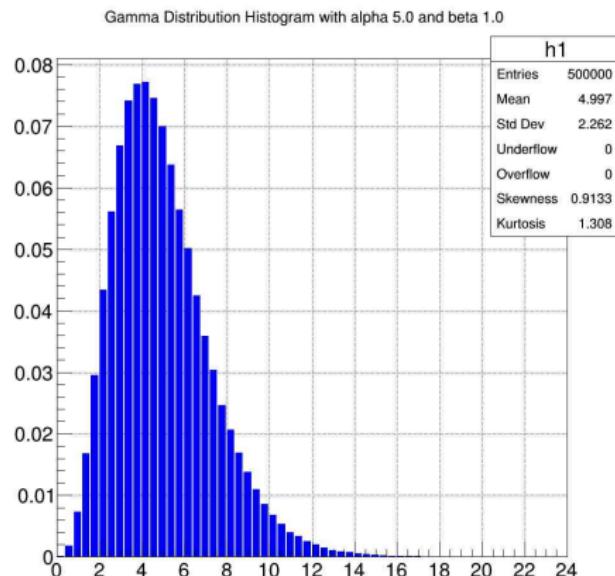
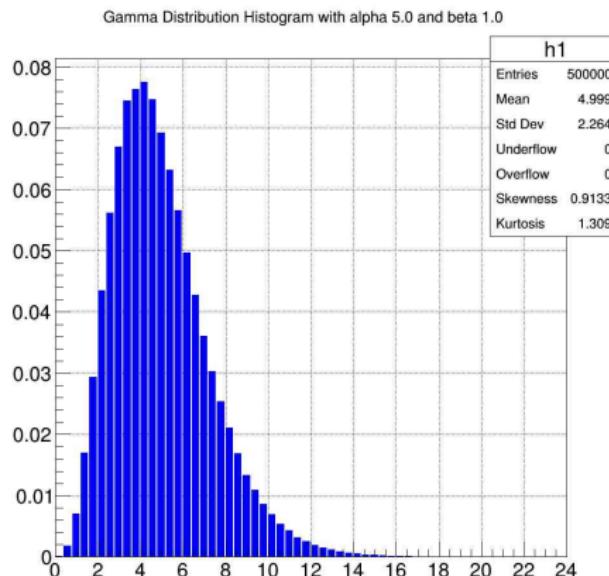


Figure: Scalar.

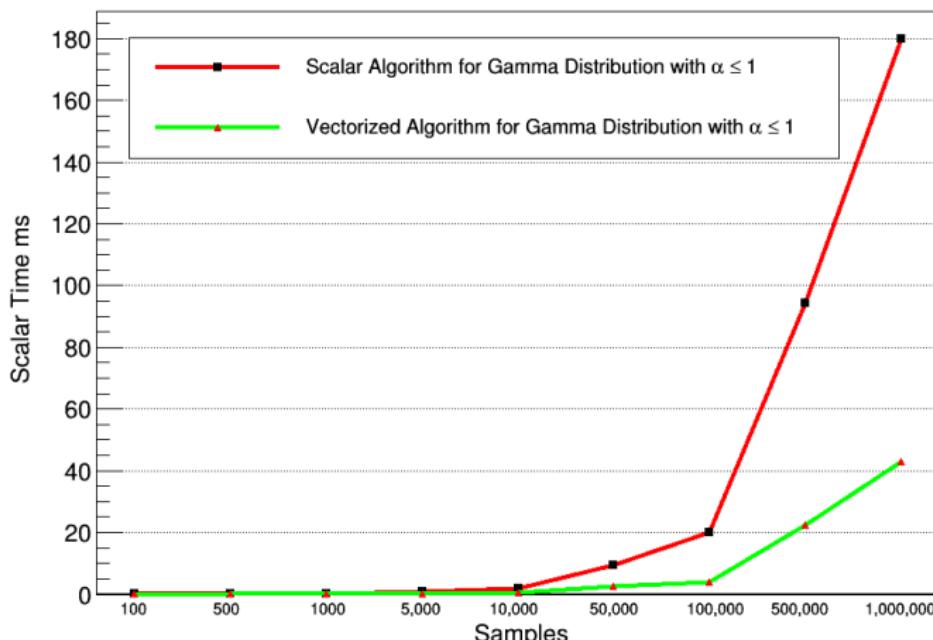
Figure: Vectorized.

Generalities of Proposed Algorithms for $\alpha \leq 1$

- The gamma random value greater than 1 can be extended linearly to values of gamma less than one (Hung et al., 2015).
- It uses one Uniform random value extra U .
- It generates gamma random variable $\tilde{X} \sim \Gamma(\alpha + 1, 1)$ parameters with the previous algorithm and then deliver $\tilde{Y} = XU^{1/\alpha}$.
Where $\tilde{Y} \sim \Gamma(\alpha, 1)$ with $\alpha \leq 1$.

Computing Performance of $\alpha = 0.5$ and $\beta = 0.5$ Algorithms
using processor Intel(R) Core(TM) i7-4510U CPU 2.00GHz
with 10 samples

Computing Performance of tested algorithms $\alpha \leq 1$



Mean and Standard Deviation of Gamma distribution of the algorithm with $\alpha \leq 1$ with 1,000,000 samples and their relative errors $\frac{V_{calc} - V_{obt}}{V_{calc}} * 100$

α	β	Mean α/β	Relative error of Mean	Standard Deviation $\frac{\sqrt{\alpha}}{\beta}$	Relative error of Standard Deviation
0.1	1	0.1	1%	0.316	1.26%
0.2	1	0.2	0.1%	0.447	2.46%
0.3	0.5	0.6	1.66%	1.095	0.01%
0.4	1	0.4	0.22%	0.632	2.85%
0.5	2	0.25	0.24%	0.353	2.83%
0.6	1	0.6	0.03%	0.774	2.71%
0.7	2	0.35	0.01%	0.418	2.39%
0.8	3	0.2666	0.01%	0.298	2.35%
0.9	2	0.45	0.22%	0.474	0.1%
1	1	1	0.01%	1	2%

Skewness and Kurtosis of Gamma distribution of the algorithm with $\alpha \leq 1$ with 1,000,000 samples and their relative errors $\frac{V_{calc} - V_{obt}}{V_{calc}} * 100$

α	β	Skewness $\frac{2}{\sqrt{\alpha}}$	Relative error of Skewness	Kurtosis $\frac{6}{\alpha}$	Relative error of Kurtosis
0.1	1	6.324	5.31%	60	8%
0.2	1	4.472	0.01%	30	4.4%
0.3	0.5	3.651	8.5%	20	25%
0.4	1	3.162	7.15%	15	13.3%
0.5	2	2.828	12.09%	12	16%
0.6	1	2.582	7.6%	10	10%
0.7	2	2.39	6.7%	8.571	9.5%
0.8	3	2.236	16.28%	7.5	9.3%
0.9	2	2.108	10.53%	6.666	11%
1	1	2	6%	6	7%

Examples of the Gamma distribution with $\alpha = 0.7$ and $\beta = 2.0$ with 500,000 samples.

Gamma Distribution Histogram with alpha 0.7 and beta 2.0

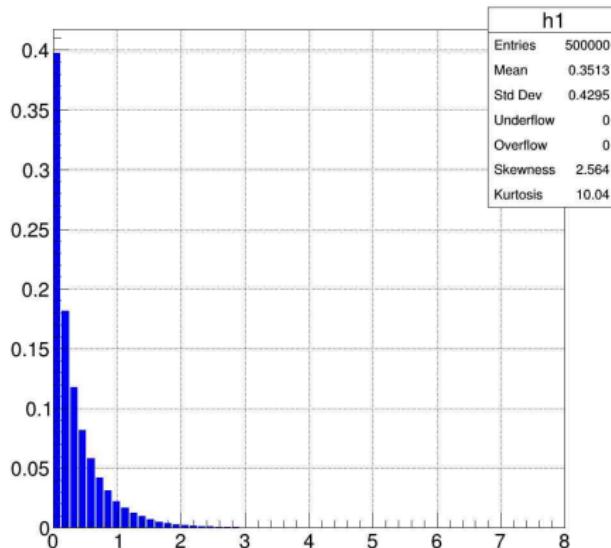


Figure: Scalar.

Gamma Distribution Histogram with alpha 0.7 and beta 2.0

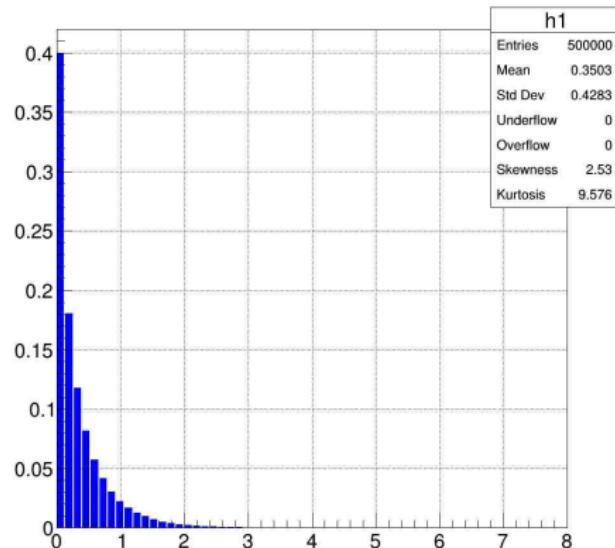
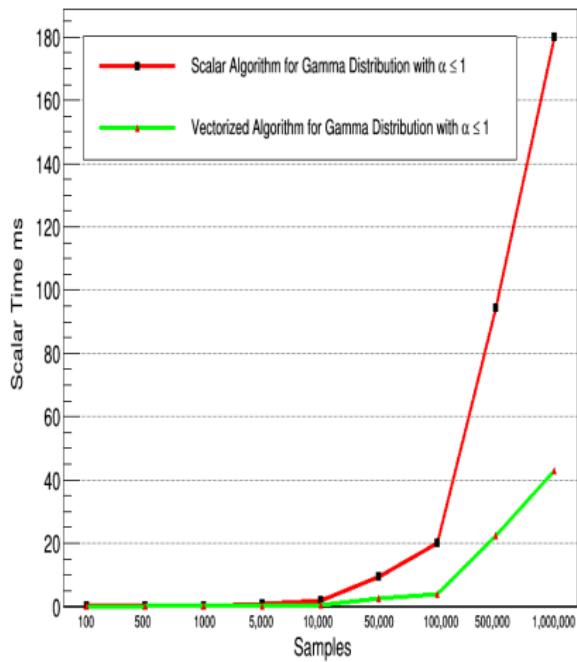


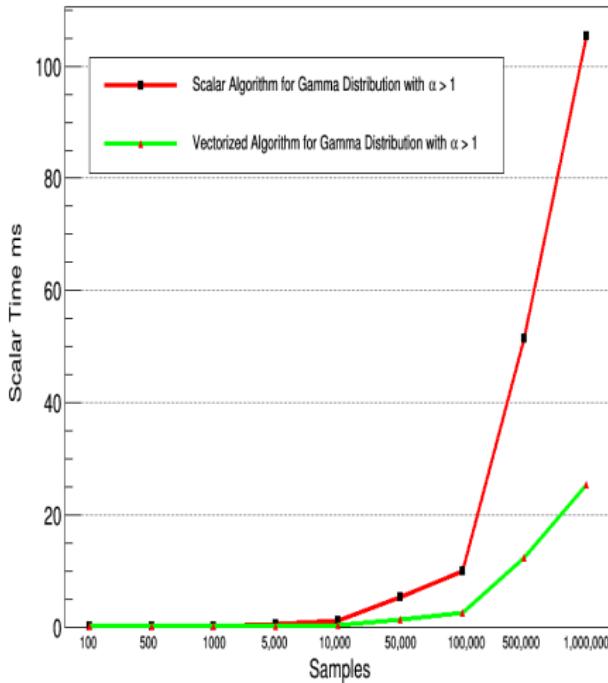
Figure: Vectorized.

Conclusions

Computing Performance of tested algorithms $\alpha \leq 1$



Computing Performance of tested algorithms $\alpha > 1$



Further Work

- Vectorize using CUDA (GTX 1050 Ti).
- Work on the next Probability Distribution Function.

References I

- [1] H. Pishro-Nik. *Introduction to probability, statistics, and random processes*, volume 1.
url<https://www.probabilitycourse.com>, kappa research llc edition, 2014.
- [2] Luca Martino and David Luengo. Extremely efficient generation of gamma random variables for $\alpha \leq 1$. *Cornell University Library.Statistics.Computation*, 3(1):1–10, June 2013.
- [3] George Marsaglia and Wai Wan Tsang. A simple method for generating gamma variables. *ACM Transactions on Mathematical Software (TOMS)*, 26(3):363–372, September 2000.

References II

- [4] Nguyen Van Hung, Ngo Thi Thanh Trang, and Tran QuocChien. An improvement of minh's algorithm for generating gamma variates with any value of shape parameter. *Indian Journal of Computer Science and Engineering (IJCSE)*, 5(6): 199–205, January 2015.