Vectorization techniques for probability distribution functions using VecCore

Chaparro Amaro Oscar Roberto[1] · Martínez Castro Jesús Alberto[1] · Soon Yung Jun[2]

[1]Instituto Politécnico Nacional, Centro de Investigación en Computación, México.
[2]Fermi National Accelerator Laboratory, USA.

Centro de Investigación en Computación
Instituto Politécnico Nacional

Introduction

Efficient generation of parallel random variates under certain distributions is often a very challenging task. VecMath provides vectorized math utilities and pseudo random number generators that use VecCore library [1]. General strategies on how to vectorize several PDFs (Probability Distribution Functions) are presented.

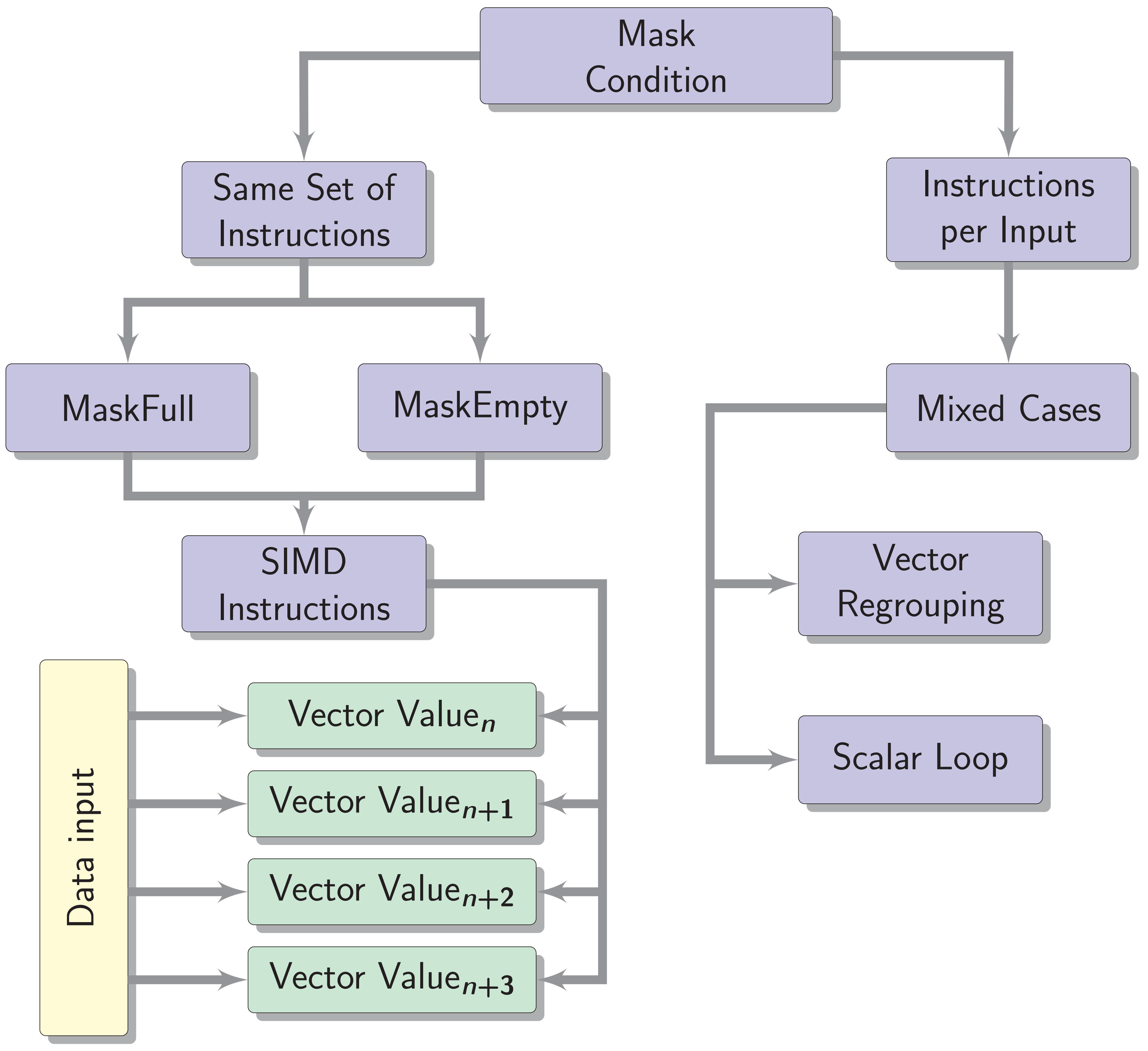
Algorithms without Branches

These cases involve 2 different approaches for PDFs:

- Direct transformation through inverse functions as Exponential distribution).
- Vector instructions without branches as Gaussian or Normal distribution with the Box-Müller algorithm [2]).

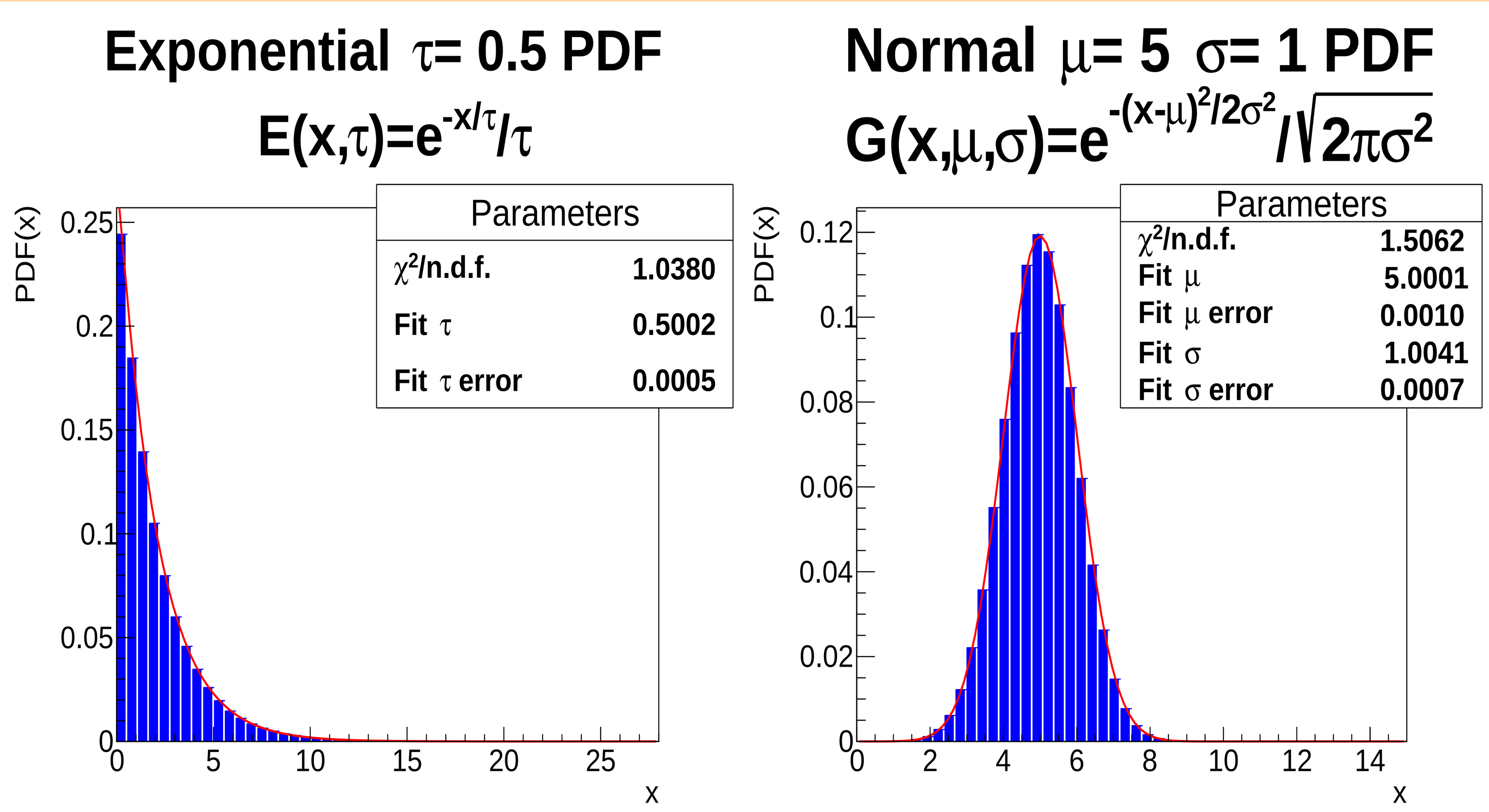
Algorithms with Branches

Cases as Gamma and discrete Poisson distributions need input and parameters that could be rejected or accepted. Using some VecCore APIs, several random conditionals are managed efficiently following: SIMD structure, the combination of branches,regrouping the data and increasing acceptance rate.



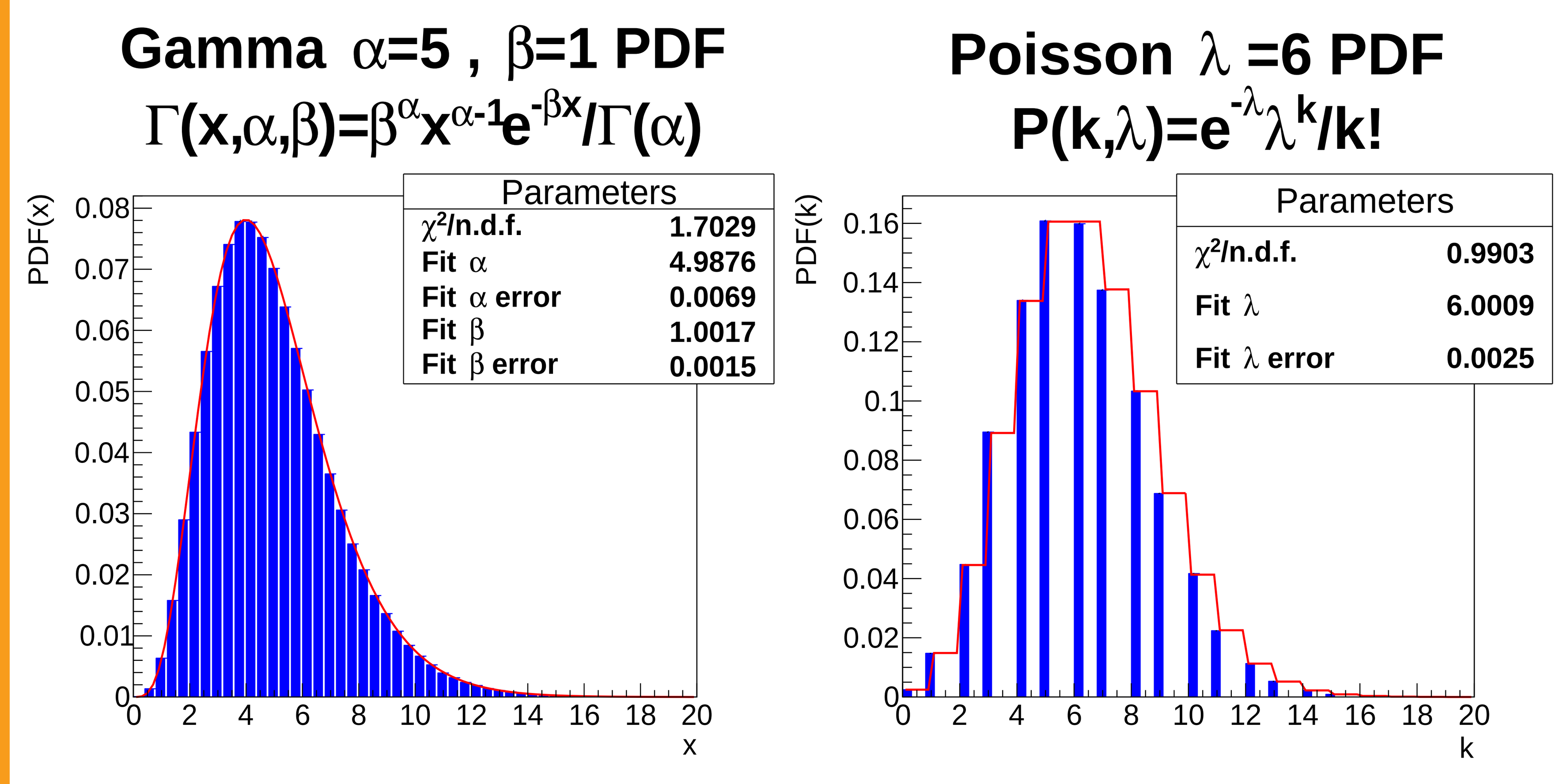
Mask condition allows to apply the same operation at same time at a pool of values inside a vector following SIMD structure if those are in the same threshold (MaskFull or MaskEmpty). Acceptance-rejection methods usually use inside do-while loop structure with random input, so if a group of data has different thresholds, a mixed case is presented. However is possible to increase acceptance rate through regrouping data.

Probability Density Functions



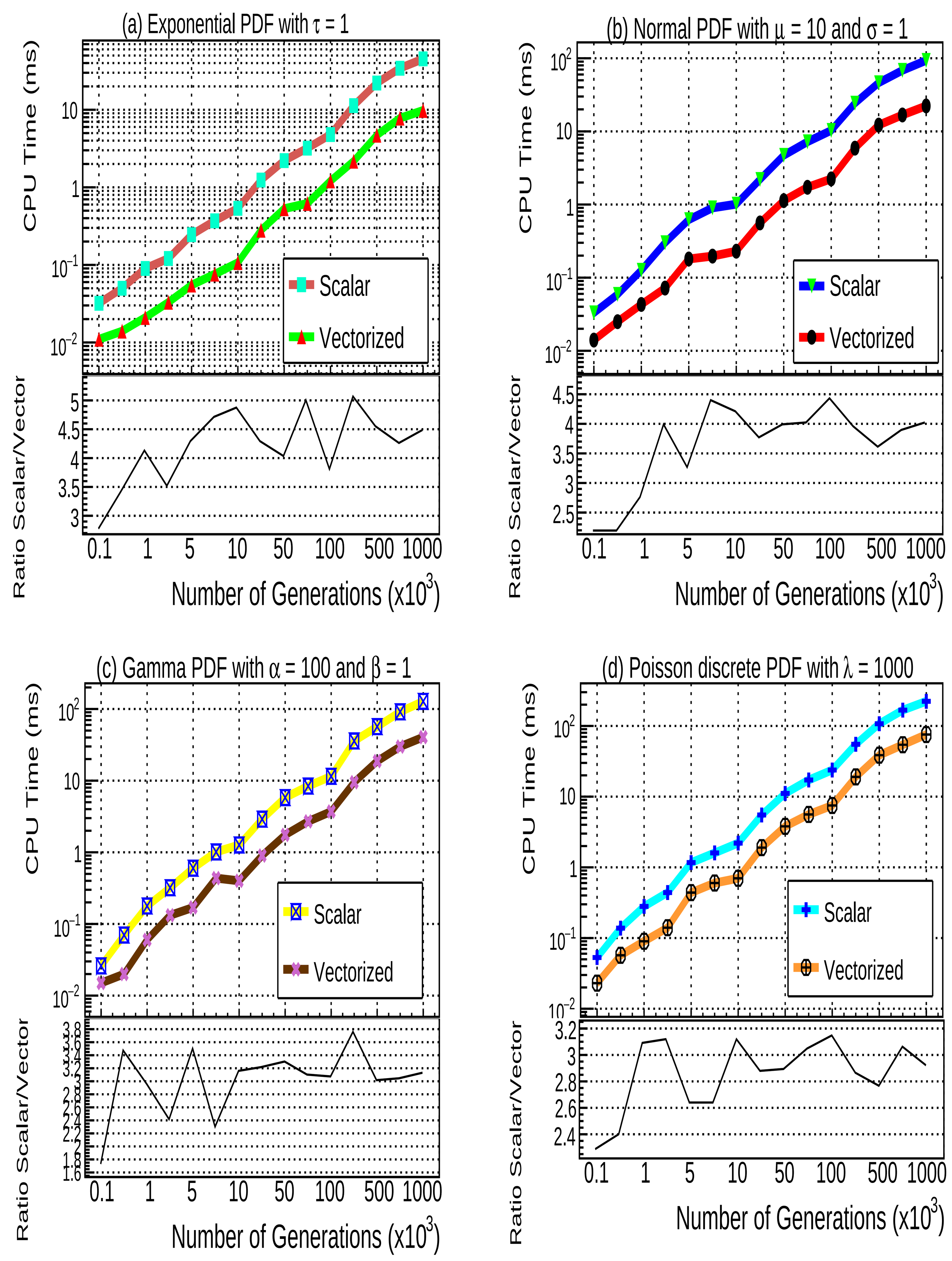
Normalized histograms of Exponential [3] (left) and Normal [2] (right) density functions generated by the vectorized algorithms.

Probability Density Functions



Normalized histograms of Gamma [4] (left) and discrete Poisson [5] (right) density functions generated by the vectorized algorithms.

Scalar and Vectorized Performance



Performance gains by vectorization are roughly **3** ∼ **5** times with test environment CPU Intel®Core^{TM}i7-4510U 2.00GHz using AVX2 set architecture with 1,000,000 samples.

Future Work

Vectorize more complex PDFs algorithms and include into Vecmath APIs.

References

[1]Amadio G. et al. Speeding up software with VecCore. 2018. Journal of Physics: Conference Series. 1085 032034
[2]Murison M. A. A Method for directly generating a Gaussian Distribution with nonunit variance and nonzero mean from Uniform random deviates. 2000. Astronomical Applications Department U.S. Navy observatory.1 5
[3]Micula S. and Pop I. D. Simulations of continuous random variables and Monte Carlo methods. 2016 Journal of Information Systems and Operations Management. 1 435
[4]Khrystyna B. and Lyudmyla S. Compositions of Poisson and Gamma processes. 2017. Modern Stochastics: Theory and Applications. 4 27
[5]Atkinson A. C. The Computer Generation of Poisson Random Variables. 1979. Journal of the Royal Statistical Society. 28 29