20. Pseudo-randomnumber. Explain how to obtain normally distributed pseudo-random numbers.

Numerical Analysis E2021

Institute of Mathematics
Aalborg University

AALBORG UNIVERSITY
DENMARK

Motivation

We have examined several pseudo-random number generators

- IBM Randu 1960
- Old Matlab algorithm
- Marsaglia's algorithm

These all generate uniformly distributed numbers.

How do we transform these into normally distributed numbers?

Polar algorithm

Generates two values at a time. Finds a random point in unit circle, by generating uniformly random points in the square $[-1,1]\times[-1,1]$, and then rejects all points outside the circle. The accepted points are then transformed using the polar transformation:

$$v_i = \sqrt{-2\frac{\log(r)}{r}}\cdot u_i, \quad i = 1,2$$

This produces two independent normally distributed elements.

Quite expensive as we will have to reject roughly 21% of the elements generated, and we have to compute square roots and logarithms.

MATLAB demo of Polar algorithm

Ziggurat algorithm

Consider the density function

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right)$$

We uniformly generate pairs $(x, y) \in \mathbb{R} \times [0, \infty)$. If $y > f(x)$ then we reject (x, y). All the remaining x will then be normally distributed.

Ziggurat algorithm

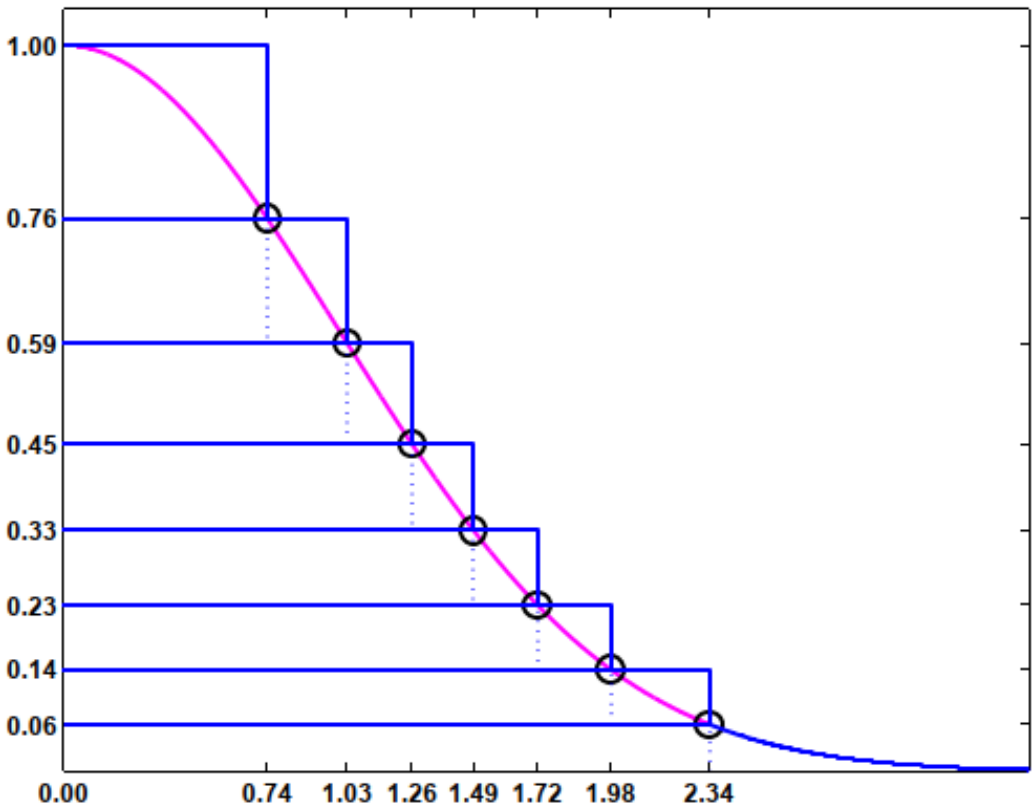

Figure: The ziggurat algorithm.