JProf. Dr. Diyora Salimova
M.Sc. Ilkhom Mukhammadiev

Practical Sheet 1

Note that we do not distinguish between pseudo random numbers and actual random numbers.

1. Let $\lambda \in (0, \infty)$ and let X be a Laplace distributed random variable with parameter λ, i.e. $X \sim \text{Laplace}_\lambda$. In this exercise you may use without proofs the results of Problem 3 from Sheet 1 regarding the Laplace distribution.

a) Write a MATLAB function `Laplace(`N`,`λ`)` with input $N \in \mathbb{N}$, $\lambda \in (0, \infty)$ which returns as output N independent realizations of Laplace_λ-distributed random variable generated with the inversion method. The MATLAB function `Laplace(`N`,`λ`)` may use at most N realizations of an $\mathcal{U}_{(0,1)}$-distributed random variable.

b) Write a MATLAB function `LaplacePlot()` which plots 10^5 realizations of an $\text{Laplace}_{0.1}$-distributed random variable generated with your MATLAB function `Laplace(`N`,`λ`)` in a histogram with 1000 bins.

2. Let $A \subseteq \mathbb{R}^2$ be the set given by

$$A = \left\{ (x, y) \in \mathbb{R}^2 \colon \frac{x^2}{8} + y^2 \leq 2 \right\}. \tag{1}$$

(i) Write a MATLAB function `AcceptanceRejection(`N`)` with input $N \in \mathbb{N}$ and output a realization of an $(\mathcal{U}_A)^{\otimes N}$-distributed random variable generated with the acceptance-rejection method. Type `AcceptanceRejection(6)` to test your implementation.

(ii) Write a MATLAB function `AcceptanceRejectionPlot()` which uses your MATLAB function `AcceptanceRejection(`N`)` from Item (i) and the built-in MATLAB function `plot(...)` to plot 10^5 realizations of an $\mathcal{U}_A$-distributed random variable in a coordinate plane.

3. (i) Write a MATLAB function `BoxMuller(`N`)` with input $N \in \mathbb{N}$ and output a realization of an $\mathcal{N}_{0, I_{\mathbb{R}^N}}$-distributed random variable generated with the Box-Muller method. Your MATLAB function `BoxMuller(`N`)` may use at most $N + 1$ realizations of an $\mathcal{U}_{(0,1)}$-distributed random variable. Type `BoxMuller(11)` to test your implementation.

(ii) Write a MATLAB function `BoxMullerPlot()` which plots 10^5 realizations of an $\mathcal{N}_{0,I_{\mathbb{R}}}$-distributed random variable generated with your function `BoxMuller(`N`)` from (i) in a *normalized* histogram with 100 bins and which also plots the density of $\mathcal{N}_{0,I_{\mathbb{R}}}$ in this histogram.

Hint: Use the MATLAB function `histogram(...)` with the `'Normalization'`-option to obtain the normalized histogram with empirical density function of the y-axis. For the plot of the $\mathcal{N}_{0,I_{\mathbb{R}}}$-density, extract the data points of the histogram and use and the built-in command `hold on`.

(iii) Write a MATLAB function `MarsagliaPolar(`N`)` with input $N \in \mathbb{N}$ and output a realization of an $\mathcal{N}_{0,I_{\mathbb{R}^N}}$-distributed random variable generated with the Marsaglia polar method. Type `MarsagliaPolar(11)` to test your implementation.

Due: Friday, 01.11.2024.
Webpage: `https://aam.uni-freiburg.de/agsa/lehre/ws24/numsde/index.html`