Prof. Dr. Karsten Urban
Mazen Ali

Institute for Numerical Mathematics
Ulm University
Summer Term 2018

Numerical Finance – Sheet 2

(due 30.04.2018)

Refer to the old script online for this exercise.

Exercise 1: Discrepancy

Prove Proposition 2.4.3. In part (b), consider only m = 1.

Exercise 2: Hardy-Krause Variation

Calculate the Hardy-Krause variation $V(f)$ for the function

$$f(x_1, x_2) = \frac{3}{1 + x_1 + 2x_2}.$$

Hint: You will first need to specify $J_k^{(2)}$, $k = 1, 2$ and to calculate the Vitali variations for all index sets $I \in J_k^{(2)}$, $k = 1, 2$.

Programming Exercise 1: Halton Sequence (7 Points)

a) Implement the radical inverse function $\phi_b(i)$, using the idea that

$$i = \sum_{k=0}^{j} d_k b^k = (d_k b^{k-1} + \cdots + d_1)b + d_0.$$

You should not need to use more than one loop in your algorithm.

b) For a d-dimensional Halton sequence

$$x_i = (\phi_{p_1}(i), \ldots, \phi_{p_d}(i)), \qquad i = 1, 2, \ldots,$$

one usually takes $p_1, \ldots, p_d$ to be the first d prime numbers. Generate 1000 points of the 2-dimensional Halton sequence (i.e. $p_1 = 2$, $p_2 = 3$). Also generate 1000 Halton points using $p_1 = 109$, $p_2 = 113$ (the 29th and 30th prime number, respectively). Compare the two point sets by plotting them (Gnuplot: `plot "data"`). What does this tell you about the behaviour of the Halton sequence in higher dimensions? (Hint: the second sequence corresponds to the projection of a 30-dimensional sequence onto the last two coordinates).

Programming Exercise 2: Transformation of Random Variables (17 Points)

Generate a set of 10000 uniformly distributed random variables using your UNIX-LCG from Sheet 1.

a) Transform these variables using

- the simple method $X := \sum_{i=1}^{12} U_i - 6$ into a normal distribution with mean 1 and variance 2,
- the Box-Muller algorithm into a normal distribution with mean 2 and variance 0.5,
- the transformation formula into an exponentially distributed random variable with $\lambda = 1$.

For each distribution, plot the histogram and the probability density function into one figure.

b) Repeat (a) with the van der Corput sequence (use your program from Programming Exercise 1 with $b = 2$) instead of the pseudo random numbers. Do you get the plots you have expected? Why / why not?

c) Generate 10000 2-dimensional independent normally distributed pseudo random variables using the Box-Muller algorithm. Plot them in a scatterplot. Generate 10000 2dimensional normal distributed pseudo random variables with correlation $\rho(X1, X2) = 0.9$. You can generate them by constructing independent pseudo random variables U_1, U_2 with the BoxMuller algorithm and then transform these variables with the formula $X_2 = U_1$, $X_1 = \rho U_1 + \sqrt{1-\rho^2}U_2$. Plot these pseudo random variables in a scatterplot and verify theoretically that the above transformation indeed yields random variables with a correlation of ρ.

Programming Exercise 3: Monte Carlo Simulation of π (5 Points)

The constant π can be approximated using Monte Carlo as follows: Consider the upper right quadrant of the unit circle.

1. Draw uniformly distributed 2D pseudo random variables $X_1, \dots, X_n$ on $[0,1]^2$.
2. Count the number c of random variables that lie within the upper right quadrant of the unit circle as well as the total number of random variables.

Then, the quantity $4\frac{c}{n} \to \pi$ as $n \to \infty$.

a) Verify theoretically that $4\frac{c}{n} \to \pi$ as $n \to \infty$.

b) Implement the MC Simulation and approximate π.

c) How many pseudo random variables do you have to generate (in powers of 10) to obtain an accuracy of 3 significant figures?