University of Freiburg
JProf. Dr. Diyora Salimova
M.Sc. Ilkhom Mukhammadiev

Numerical Approximation of SDEs

Winter 2024/25

Practical Sheet 2

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

1. **a)** Write a MATLAB function `RecRule(a,b,d,n,f)` with input $a \in \mathbb{R}$, $b \in (a, \infty)$, $d, n \in \mathbb{N}$, $f \colon [a,b]^d \to \mathbb{R} \in \mathcal{L}^1(\mathcal{B}_{[a,b]^d}; |\cdot|_{\mathbb{R}})$ and output $R^n_{[a,b]^d}[f]$.

Hint: Implement the evaluation of f in the rectangle rule *recursively*. To this end, write a subroutine `RecRuleRecursion(f,...,d)`. Specify the remaining input parameters of the recursion, and use the following (rough) structure for your code:

---

procedure `RecRuleRecursion(f,...,d)`
  if $d > 1$ **then**
    for $i_d \in \{0, \dots, n-1\}$ **do**
      Fix the d-th coordinate $x_{d,i_d} := a + \frac{i_d}{n}(b-a)$.
      Evaluate the $(d-1)$-dimensional integral with respect
      to the domain $[a,b]^{(d-1)} \times \{x_{d,i_d}\}$ by calling
      `RecRuleRecursion(f, ..., d-1)` and add the result
      to the overall approximation.
    end for
  else
    Use the one-dimensional rectangle rule and add the result
    to the overall approximation.
  end if
end procedure

---

b) Let $a = 0$, $b = 1$ and let $f = [0,1]^d \ni (x_1, \dots, x_d) \mapsto x_1 \in \mathbb{R}$. Test your implementation by computing

$$\left| R^n_{[a,b]^d}[f] - \int_{[a,b]^d} f(x)\, dx \right|_{\mathbb{R}} \tag{1}$$

for $n \in \{2^4, 2^5, \dots, 2^{10}\}$, $d \in \{1, 2, 3\}$, and measure the execution time for each d and n. For each d, plot the error in Eq. (1) against the execution time. Plot all three error curves in one diagram with logarithmic scale and times on the x-axis.

Hint: Use the built-in function `loglog` to generate a logarithmic plot.

2. Let $d \in \mathbb{N}$, $a \in \mathbb{R}$, $b \in (a, \infty)$, $f \in \mathcal{L}^1(\mathcal{B}_{[a,b]^d}; |\cdot|_{\mathbb{R}})$, let $(\Omega, \mathcal{F}, P)$ be a probability space, and let $X_j \colon \Omega \to \mathbb{R}^d$, $j \in \mathbb{N}$, be a sequence of independent $\mathcal{U}_{[a,b]^d}$-distributed random variables on $(\Omega, \mathcal{F}, P)$. For all $N \in \mathbb{N}$ define the functions

$$I_N := \frac{(b-a)^d}{N} \left[\sum_{j=1}^{N} f(X_j) \right]. \tag{2}$$

The function I_N as in Eq. (2) is the *Monte Carlo estimator* of the integral

$$\int_{[a,b]^d} f(x)\, dx = (b-a)^d \mathbb{E}_P[f(X_1)]. \tag{3}$$

a) Write a MATLAB function `intMC(a,b,d,N,f)` with input $a \in \mathbb{R}$, $b \in (a, \infty)$, $d \in \mathbb{N}$, $f \in \mathcal{L}^1(B_{[a,b]^d}; |\cdot|_{\mathbb{R}})$, $N \in \mathbb{N}$ that outputs a realization of I_N.

b) Test your MATLAB function `intMC(a,b,d,f,N)` by repeating the experiment from Problem 1, b) above: Let $a = 0$, $b = 1$ and let $f = [0,1]^d \ni (x_1, \ldots, x_d) \mapsto x_1 \in \mathbb{R}$. Test your implementation by computing

$$\left| I_N - \int_{[a,b]^d} f(x)\, dx \right|_{\mathbb{R}} \tag{4}$$

for $N \in \{2^{14}, 2^{15}, \ldots, 2^{20}\}$, $d \in \{1, 2, 3\}$, and measure the execution time for each d and n. Plot again the error for any dimension d against the execution time and compare the results to the rectangular rule from Problem 1.

3. Approximative realizations of a one-dimensional standard Brownian motion:
Let A be the set given by

$$A = \cup_{n=1}^{\infty} \{\mathbf{t} = (t_1, \ldots, t_n) \in [0, \infty)^n \colon \#_{\mathbb{R}}(\{t_1, \ldots, t_n\}) = n\}, \tag{5}$$

let length: $A \to \mathbb{N}$ be the function which satisfies for all $n \in \mathbb{N}$, $\mathbf{t} = (t_1, \ldots, t_n) \in [0, \infty)^n \cap A$ that

$$\text{length}(\mathbf{t}) = n, \tag{6}$$

and let $Q \colon A \to (\cup_{n=1}^{\infty} \mathbb{R}^{n \times n})$ be the function which satisfies for all $n \in \mathbb{N}$, $\mathbf{t} = (t_1, \ldots, t_n) \in [0, \infty)^n \cap A$ that

$$Q(\mathbf{t}) = (\min\{t_i, t_j\})_{(i,j) \in \{1, \ldots, n\}^2}. \tag{7}$$

Write a MATLAB function `StandardBrownianMotion(t)` with input $\mathbf{t} \in A$ and output a realization of an $\mathcal{N}_{0, Q(\mathbf{t})}$-distributed random variable. The MATLAB function `StandardBrownianMotion(t)` may use at most length($\mathbf{t}$) realizations of an $\mathcal{N}_{0, I_{\mathbb{R}}}$-distributed random variable. Call the MATLAB commands

```
rng('default');
N=10^3;
preimage = (0:1/N:1);
X=StandardBrownianMotion(preimage);
plot(preimage,X);
```

```
hold on
X=StandardBrownianMotion(preimage);
plot(preimage,X,'r');
X=StandardBrownianMotion(preimage);
plot(preimage,X,'g');
```

to test your implementation.

Due: Friday, 15.11.2024.
Webpage: `https://aam.uni-freiburg.de/agsa/lehre/ws24/numsde/index.html`