2 Coursework Description

If you use a random number generator for any of the problems below, seed the generator so that the results are reproducible.

Problem 1. *Consider the two integrals*

$$I_1 = \int_0^2 \frac{1}{1+x^3}\,dx;$$
$$I_2 = \int_0^\infty \frac{1}{\sqrt{x}} e^{-3x}\,dx.$$

(a) Describe a numerical method to approximate the value of I_1 such that the approximation error is guaranteed to be bounded by 1/1000. Implement this method in Python and provide the value of the approximation.

(b) Explain how one can approximate the integral I_2 using a Monte Carlo estimator. Implement the Monte Carlo estimator in Python and provide a figure that plots a Monte Carlo estimate against the number of samples (as we done have in the lectures and programming sessions). Describe a variance reduction technique that can be applied here and discuss how well this technique works in this example.

Problem 2. *Consider the function $f : \mathbb{R} \to \mathbb{R}$ given by*

$$f(x) = \begin{cases} \alpha(1+x)^2, & \text{if } x \in (0,2), \\ 0, & \text{if } x \notin (0,2), \end{cases}$$

where α is a constant.

(a) Determine α such that f is a probability density function.

(b) Let α be the constant such that f is a probability density function (i.e., the one computed in part (a)). Suppose you would like to generate a sample from f using von Neumann's acceptance-rejection algorithm. Specify a probability density function $g \neq f$ that can be used for this purpose and describe in detail how you can obtain a sample from f by sampling from g using von Neumann's acceptance-rejection algorithm. For your choice of g what is the best possible proportion of numbers that your algorithm accepts? Implement von Neumann's acceptance-rejection algorithm in Python to obtain 10000 samples from f and plot a histogram of the samples.

(c) An alternative to von Neumann's acceptance-rejection algorithm from part (b) for sampling from f would be to use the inverse transform method. Implement it in Python and draw again a histogram of the samples. Which of these two methods do you think is more suitable for generating a sample from f and why?

Problem 3. *We consider the standard Black-Scholes financial market consisting of two assets. The riskless asset has time-t price $B_t = e^{rt}$, where $r \geq 0$ is the constant interest rate and the stock has time-t price*

$$S_t = S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right) t + \sigma W_t\right), \tag{1}$$

where $S_0 > 0$ is the initial stock price, $\sigma > 0$ is the volatility, $(W_t)_{t \geq 0}$ is a standard one-dimensional Brownian motion under the risk-neutral measure.

Fix some constants a, b, K with $b > a > 0$ and $K > 0$ and consider a European option with payoff H at the maturity date $T > 0$ given by

$$H = \begin{cases} K, & \text{if } S_T \in (a, b), \\ 0, & \text{if } S_T \notin (a, b). \end{cases} \tag{2}$$

(a) *Derive an analytical formula for the time-0 price of this option.*

(b) (b.1) *Write down a Monte Carlo estimator for the time-0 price of this option. Justify your answer.*

(b.2) *Explain in detail how one can generate the random variables that are used in your Monte Carlo estimator in part (b.1).*

(b.3) *Compute the variance of the Monte Carlo estimator for the time-0 price of the option analytically.*

(b.4) *Write down a 95% and a 99% asymptotic confidence interval for the time-0 price of the option using your Monte Carlo estimator.*

(c) *Write Python code*

(c.1) *that computes the time-0 price of the option using the analytical formula and*

(c.2) *that computes the approximation of the time-0 price of the option using a Monte Carlo estimator together with an asymptotic confidence interval.*

(d) *Use your Python code to compute the time-0 price of the option for the model parameters $S_0 = 10$, $a = 10$, $b = 12$, $K = 50$, $r = 0.01$, $\sigma = 0.2$, $T = 1$ using both the analytical formula and the Monte Carlo estimator. Provide a 95%-asymptotic confidence interval for the time-0 price of the option. Discuss your results.*

(e) *Describe a variance reduction method to approximate the time-0 price of the option and implement it in Python. Compare the results based on the variance reduction method of your choice to the standard Monte Carlo estimator and to the analytical solution. Discuss your findings.*

Problem 4. *This question continues the previous one, but now we leave the world of the Black-Scholes model. We still assume that the time-t price of the riskless asset is given by* $B_t = e^{rt}$ *with* $r \geq 0$. *However, we now assume that the dynamics under the risk-neutral measure of the risky asset are given by*

$$(3) \quad dS_t = rS_t dt + \sigma\sqrt{S_t} dW_t,$$

where $(W_t)_{t\geq 0}$ *is again Brownian motion under the risk-neutral measure and* $\sigma > 0$. *As before we assume* $S_0 > 0$.

(a) *Explain how you can generate a sample path of* $S = (S_t)_{t\geq 0}$ *given in (3) on the discrete time grid* $0 < h < 2h < \ldots < nh$ *for* $h > 0$, $n \in \mathbb{N}$. *Write Python code that provides samples of* S_T. *Create a plot with ten sample paths of* (S_t) *for* $S_0 = 10$, $r = 0.01$, $\sigma = 0.2$, *and* $T = 1$ *with* $h = 1/250$.
<u>*Additional detail:*</u> *It might happen that your approximation of* S_{ih} *for some* $i \in \{1, 2, \ldots, 2\}$ *becomes negative. Whenever this happens you should replace your approximation by zero.*

(b) *Fix some constants* a, b, K *with* $b > a > 0$ *and* $K > 0$ *and consider a European option with payoff* H *at the maturity date* $T > 0$ *given by*

$$(4) \quad H = \begin{cases} K, & \text{if } S_T \in (a, b), \\ 0, & \text{if } S_T \notin (a, b). \end{cases}$$

Write down a Monte Carlo estimator for the price of this option if the stock price is given by the dynamics in (3). *Justify your answer.*

(c) *Write Python code that computes the approximation of the time-0 price of the option using a Monte Carlo estimator together with an asymptotic confidence interval. Use your Python code to compute the time-0 price of the option for the model parameters* $S_0 = 10$, $a = 10$, $b = 12$, $K = 50$, $r = 0.01$, $\sigma = 0.2$, $T = 1$ *using the Monte Carlo estimator. Provide a 95%-asymptotic confidence interval for the time-0 price of the option. Discuss your results.*

(d) *Specify two control variate estimators, that differ in the choice of the random variable used as control, for approximating the time-0 price of the option with payoff* (4) *under the dynamics of (3) and implement them in Python. Compare the results based on these two control variate estimators and the standard Monte Carlo estimator. Discuss your findings.*