Christian-Albrechts-Universität zu Kiel **SS 2022**
Mathematisches Seminar
Prof. Dr. Jan Kallsen
Henrik Valett, Fan Yu, Boy Schultz Sheet 07

Computational Finance

Exercises for all participants

C-Exercise 25 (Sampling from a distribution by the acceptance/rejection method) (4 points)
We want to generate samples of the following density

$$f(x) = \begin{cases} \frac{2\sqrt{1-x^2}}{\pi}, & \text{if } -1 \leq x \leq 1 \\ 0, & \text{otherwise.} \end{cases},$$

Write a Python function

```
Sample_dist_AR (N)
```

that generates and returns $N \in \mathbb{N}$ independent samples from the distribution by means of the acceptance/rejection method. In your algorithm, you may sample only from the uniform distribution on $[0,1]$ using the function `numpy.random.uniform`.

Generate $N = 10000$ samples, and plot them in a histogram. Plot the density $f(x)$ in the same histogram using the right scaling.

Useful Python commands: `plt.hist`, `np.random.uniform`, `while`

C-Exercise 26 (Using antithetic variables to reduce the variance of MC-estimators) (4 points) Write a Python function

```
V0 = BS_EuOption_MC_AV (S0, r, sigma, T, K, M, g)
```

that computes the initial price of a European option with payoff $g(S_T)$ at time T in the Black-Scholes model via the Monte-Carlo approach with $M \in \mathbb{N}$ samples. Use the method of antithetic variables to reduce the variance of the estimator. In addition, the function shall return the radius ε of a confidence interval that contains the true price with a probability of approximately 95% (cf. Section 5.1).
Test your function for

$$S(0) = 110, \quad r = 0.03, \quad \sigma = 0.2, \quad T = 1, \quad K = 100, \quad M = 100000, \quad g(x) = (x-K)^+$$

and compare the result to the exact value (cf. formula below (3.23) in the lecture notes) and the plain Monte-Carlo simulation (cf. C-Exercise 21).
Hint: Modify the function of C-Exercise 21 appropriately.

C-Exercise 27 (Using control variables to reduce the variance of MC-estimators) (4 points)
Write a Python function

```
V0 = BS_EuOption_MC_CV (S0, r, sigma, T, K, M)
```

that computes the initial price of a European self-quanto call, i.e. an option with payoff $(S(T)-K)^+S(T)$ for some strike price K at maturity, in the Black-Scholes model via the Monte-Carlo approach with $M \in \mathbb{N}$ samples. Use a European call option with the same strike price K as control variate to reduce the variance of the estimator. To this end, estimate in a first Monte-Carlo simulation with M samples the optimal value

$$\frac{\mathrm{Cov}((S(T)-K)^+S(T),(S(T)-K)^+)}{\mathrm{Var}((S(T)-K)^+)}.$$

Test your function for the parameters

$$S(0)=110, \quad r=0.03, \quad \sigma=0.2, \quad T=1, \quad K=100, \quad M=100000,$$

and compare the result to the plain Monte-Carlo simulation (cf. C-Exercise 21).

Useful Python commands: `numpy.cov`

T-Exercise 28 (Box-Muller method) (for math only)
Prove that the *Box-Muller method* indeed works. I.e. show that if you have two independent random variables U_1, U_2 which are uniformly distributed on the interval $[0,1]$ then the random variables X_1, X_2 defined via

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2)$$
$$X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

are independent and standard normally distributed.

Hint: Use Theorem 5.1 to find the density of the vector (X_1, X_2).

Please include your name(s) as comment in the beginning of the file.
Do not forget to include comments in your Python-programs.
Submit until: Thu, 02.06.2022, 08:15