Christian-Albrechts-Universität zu Kiel **SS 2024**
Mathematisches Seminar
Prof. Dr. Mathias Vetter
Henrik Valett, Fan Yu, Ivo Richert, Anton Schellin Sheet 03

Computational Finance

Exercises for all participants

C-Exercise 09 (Sampling from a distribution by the acceptance/rejection method) (4 points)
We want to generate samples of the truncated normal distribution with parameters $a < b$ and $\mu \in \mathbb{R}, \sigma > 0$ which has the following density

$$f(x) = \begin{cases} \frac{\phi(\frac{x-\mu}{\sigma})}{\sigma\left(\Phi(\frac{b-\mu}{\sigma})-\Phi(\frac{a-\mu}{\sigma})\right)}, & \text{if } a \le x \le b \\ 0, & \text{otherwise} \end{cases},$$

where $\phi(x)$ is the density of a standard normal distribution and $\Phi(x)$ is the cdf of a standard normal distribution. Write a Python function

`Sample_TruncNormal_AR (a, b, mu, sigma, N)`

that generates and returns $N \in \mathbb{N}$ independent samples from the truncated normal distribution with parameters $a < b$ and $\mu \in \mathbb{R}, \sigma > 0$ by means of the acceptance/rejection method from the lecture notes. In your algorithm, you may sample only from the uniform distribution on $[0,1]$ using the function `numpy.random.uniform`. Do not use `scipy.stats.truncnorm`!

For $a = 0$, $b = 2$, $\mu = 0.5$, $\sigma = 1$ 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: `scipy.stats.norm.pdf`, `scipy.stats.norm.cdf`, `plt.hist`, `np.random.uniform`, `while`

C-Exercise 10 (Using control variates 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 from the lecture notes (including the variance reduction via control variates) 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

$$S(0) = 100, \quad r = 0.05, \quad \sigma = 0.3, \quad T = 1, \quad K = 110, \quad M = 100000,$$

and compare the result to the plain Monte-Carlo simulation (cf. C-Exercise 07).

Useful Python commands: `numpy.cov`

C-Exercise 11 (Pricing a deep out-of-the-money European call option by Monte-Carlo with importance sampling) (4 points)
Consider a Black-Scholes model with parameters $S(0)$, r, $\sigma > 0$. The goal is to approximate the fair price $V(0)$ of an European call option on the stock with strike $K \gg S(0)$ at maturity T.

Write a Python function

```
[V0, CIl, CIr] = BS_EuCall_MC_IS (S0, r, sigma, K, T, mu, N,
                                   alpha)
```

that approximates the price of the European call option via the Monte-Carlo approach from the lecture notes (including importance sampling to reduce the variance) based on $N \in \mathbb{N}$ samples and additionally returns the left and right boundary of an asymptotic α-level confidence interval. Use a new random variable $Y \sim N(\mu, 1)$ for the importance sampling method.

Test your function for $S(0) = 100$, $r = 0.05$, $\sigma = 0.3$, $K = 220$, $T = 1$, $N = 10000$, $\alpha = 0.95$ and plot your estimator for $V(0)$ in dependence on μ against the true value.

Hint: Experiment with the range of μ such that you can see visible changes in the variance of your estimator. For the true value use the Black-Scholes formula provided on sheet 01.

T-Exercise 12 (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 2.1 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: Fri, 17.05.2024, 12:00