Technische Universiteit Delft
Fac. Elektrotechniek, Wiskunde en Informatica
Computational Finance (WI4154)

Resit of Assignment 1

Submission deadline: **7th May 2025, 23:59**

Please type your answers in Word or Latex and submit your answer sheet in pdf format. Handwriting won't be graded.

For the programming questions, you are required to submit your Python codes in .py files, instead of Jupyter notebooks.

Please do not zip your files when uploading to BrightSpace. Zipped package won't be graded.

Group discussion is not allowed.

1. **Derivatives pricing (6 out of 20 points)**

   (i) Consider a discretely monitored Asian option with the payoff at maturity T being defined as follows:

$$\Lambda(T) = \max\left(\left(\prod_{n=1}^{N} S(t_n)\right)^{1/N} - K, 0\right),$$

   where $\{t_n\}_n^N$ are equally spaced with $t_n = n \cdot \Delta t$ and $N \cdot \Delta t = T$.

   Note that

$$\prod_{i=1}^{n} S(t_i) = \frac{S(t_n)}{S(t_{n-1})}\left(\frac{S(t_{n-1})}{S(t_{n-2})}\right)^2\left(\frac{S(t_{n-2})}{S(t_{n-3})}\right)^3 \cdots \left(\frac{S(t_1)}{S(t_0)}\right)^n S_0^n.$$

   We further assume a log-normal model for the asset price, i.e.,

$$S(t_{n+1}) = S(t_n)e^{\left(\mu-\frac{1}{2}\sigma^2\right)\Delta t+\sigma\sqrt{\Delta t}Z_n},$$

   for i.i.d. $Z_n \sim N(0,1)$.

   Find out the distribution of $\ln\left[(\prod_{n=1}^{N} S(t_n))^{\frac{1}{N}}/S_0\right]$. (**2 points**)

   (ii) Now consider a continuously monitored Asian option with the payoff at maturity T being defined as $\Lambda(T) = \max(A_T - K, 0)$, with A_T defined as the time$-T$ value of the following process:

$$A_t = \frac{1}{T}\int_0^t S_u du.$$

We price the option using the Black-Scholes model under the risk neutral probability measure, i.e.,

$$dS_t = rS_t dt + \sigma S_t dB_t,$$

where

- S_t is the stock price;
- r is the constant risk free rate;
- σ is the constant volatility;
- B_t is a Brownian motion.

Use C_t to denote the price of the continuously monitored Asian call at time t. **Derive the Partial Differential Equation (PDE)** that C_t satisfies. Further, specify the boundary conditions of PDE. (**2 points**)

(iii) To simplify the calculation let's assume now $S_t = e^{B_t}$, and $A_T = \int_0^T S_t dt$. Consider $\varphi(\omega) = \mathbb{E}\left[e^{i\omega A_T}\right]$. Use the Feynman-Kac theorem to formulate $\varphi(\omega)$ as a solution to **a partial differential equation (PDE)** , together with the boundary and terminal conditions. (**2 points**)

2. Calibration of the Heston model: (8 out of 20 points)

Let us use the excel file enclosed, whereby some European call and put option quotes are provided. (Recall that the column "Strike" stores the strike prices, the column "Midpoint" stores the quoted option prices, the underlying asset is SPX index and the as-of-date price of SPX is in cell R2, the as-of-date is in cell R1 and the maturity date is in cell R5.) Assume the risk-free interest rate is 0.11% and use Act-365 day count convention (i.e. the year fraction between two dates is calculated as the number of days in between divided by 365).

(i) Code the characteristic function of the Heston model (given in the original COS paper, after equation (34)) in python (**1 point**).

(ii) Calibrate the model parameters for the Heston model using both the call and put option prices provided in the excel file **(4 points)**.

(iii) Plot the risk neutral probability density of the logarithm of the underlying asset price at maturity that you obtained from your calibrated Heston model. Use the risk neutral valuation theory to explain that your calibrated model is indeed arbitrage free. **(1 point)**.

(iv) Make a plot to compare model-implied prices, i.e. option prices using the calibrated Heston model parameters from (ii), and the market prices **(1 point).** Summarize what you observe **(1 point).**

3. The COS method: (6 out of 20 points)

Follow the steps in section 2.1 "Inverse Fourier integral via cosine expansion" of the original COS paper (which is available on BrightSpace under lecture nr. 7), but

(i) Replace equation (5) and (6) by the Fourier-sine series expansion instead, to derive a sine version of equation (11). (**1 point**)

(ii) Program your sine version of equation (11) in Python and reproduce the errors in Table 1 of the paper (copied below). Note that only the errors are needed, not the CPU times. (**2 points**)

TABLE 1
Maximum error when recovering $f(x)$ from $\phi(\omega)$ by Fourier-cosine expansion.

N	4	8	16	32	64
Error	0.25	0.11	0.0072	4.04e-07	3.33e-16
CPU time (msec.)	0.046	0.061	0.088	0.16	0.29
Diff. in CPU (msec.)	–	0.015	0.027	0.072	0.13

(iii) Derive the sine version of the pricing formula for **European put options**, i.e. equation (19) and (25). Hint: the characteristic function of the logarithm of the underlying asset price is usually known, and thus, in the paper the payoff function of a European option is expressed as a function of the log-asset-price. Please follow the same convention. (**1 point**)

(iv) Implement your formula from (iii) in python to price an **European put option**, assuming the underlying stock price follows a log-normal model, i.e.

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

where S_t is the underlying stock price with $S_0 = 3$, $\sigma = 0.3$, the maturity of the option $T = 1$ and the strike price $K = 4$. The risk-free interest rate $r = 0.03$. (**2 points**).