Christian-Albrechts-Universität zu Kiel **SS 2024**
Mathematisches Seminar
Prof. Dr. Mathias Vetter
Henrik Valett, Fan Yu, Ivo Richert, Anton Schellin Sheet 01

Computational Finance

Exercises for participants of all programs

C-Exercise 01 (4 Points)

(a) Write a Python function

```
S = CRR_stock(S_0, r, sigma, T, M)
```

that returns the stock price matrix $S \in \mathbb{R}^{(M+1)\times(M+1)}$ in the CRR-model with initial stock price $S(0) > 0$, interest rate $r > 0$ and volatility $\sigma > 0$ to a time horizon $T > 0$ with $M \in \mathbb{N}$ time steps.
Reminder: In the stock price matrix S of the CRR-model $S_{ji} = S(0)u^j d^{i-j}$ denotes the stock price at time t_i after j upward and hence $i - j$ downward movements (see Section 2.4).

(b) Now we want to price call options in the CRR-model. Write a Python function

```
V_0 = CRR_EuCall (S_0, r, sigma, T, M, K)
```

that computes and returns an approximation to the price of an European call option with strike $K > 0$ and maturity $T > 0$ in the Black-Scholes model with initial stock price $S(0) > 0$, interest rate $r > 0$ and volatility $\sigma > 0$ using the CRR-model as presented in the course and $M \in \mathbb{N}$ time steps.

(c) We want to compare the CRR model to the true price in the BS-model. To this end implement the BS-Formula for European call options as a Python function:

```
V_0 = BlackScholes_EuCall (t, S_t, r, sigma, T, K)
```

(d) Compare the results by plotting the error of the CRR model against the BS-price in a common graph. Use the following parameters

$$S(0) = 100,\ r = 0.03,\ \sigma = 0.3,\ T = 1,\ M = 100,\ K = 70, \ldots, 200.$$

Black-Scholes formula
The fair price of a European Call option with strike $K > 0$ in the Black-Scholes model at time $0 \leq t \leq T$ with stock price $S(t)$ is given by:

$$C(t, S(t), r, \sigma, T, K) = S(t)\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2).$$

Here, Φ denotes the cumulative distribution function of the standard normal distribution and

$$d_1 := \frac{\log\left(\frac{S(t)}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}, \quad d_2 := d_1 - \sigma\sqrt{T-t}.$$

C-Exercise 02 (4 points)
Let $s_1, \ldots, s_N$ denote a time series of, e.g., stock prices on days $1, \ldots, N$. The *logarithmic return* (log-return) of the stock on day $n \in \{2, \ldots, N\}$ is given by

$$l_n := \log\left(\frac{s_n}{s_{n-1}}\right) = \log(s_n) - \log(s_{n-1}).$$

Assuming 250 trading days per year, the *annualized empirical mean* of log-returns is given by

$$\hat{\mu} = \frac{250}{N-1} \sum_{k=2}^{N} l_k$$

and the *annualized empirical standard deviation* of log-returns is given by

$$\hat{\sigma} = \sqrt{\frac{250}{N-2} \sum_{k=2}^{N} \left(l_k - \frac{\hat{\mu}}{250}\right)^2}.$$

(a) Write a Python function

```
log_returns(data)
```

that computes and returns the time series of log-returns for the time series given in `data`.

(b) In the Material folder you find the file `time_series_dax_2024.csv` containing a time series of daily DAX data. Import this time series and test your function with it. This includes to

- apply the function to the imported time series,
- visualize the time series of log-returns in a plot,
- compute and display the annualized empirical mean and standard deviation of the log-returns.

(c) Simulate a time series of log-returns with the assumption that these are normal distributed. Use your result of the empirical mean and standard deviation as parameters. Plot this simulated log-returns in the same figure as the log-returns from the data (in different colors).

(d) What differences do you observe between the two time series? What do you conclude? (Write a short answer as comment at the end of your program.)

Hint: Use the data from the column 'Close' and pay attention to correct symbols indicating the seperator between columns and the decimal point.

Useful Python commands: `numpy.diff`, `numpy.log`, `numpy.genfromtxt`, `numpy.mean`, `numpy.var`, `numpy.random.normal`, `print`, `str`

T-Exercise 03 (4 Points)
In the course we fixed the relation $u = \frac{1}{d}$ in the specification of the binomial model that is used as an approximation to the Black-Scholes model. For even $M \in \mathbb{N}$, this implies $S(0)u^{\frac{M}{2}}d^{\frac{M}{2}} = S(0)$. Replace the condition (1.3), i.e. $u = \frac{1}{d}$, by

(a) $q = 0.5$,

(b) $S(0)u^{\frac{M}{2}}d^{\frac{M}{2}} = S(0)e^{rT}$,

(c) $S(0)u^{\frac{M}{2}}d^{\frac{M}{2}} = K$.

and compute the value of the parameters u, d and q for each case (*maths students*) resp. only for (a) *(QF students)*. Discuss the potential benefit of the alternative conditions and in what scenario they may be useful.

T-Exercise 04 (Convergence of the binomial model to the Black-Scholes model) (for math only)
For $M \in \mathbb{N}$, denote by $\left(S_t^M\right)_{t\in\{t_0,\ldots,t_M\}}$ the stock price process in a binomial model with M timesteps that approximates the Black-Scholes model with parameters $r > 0$, $S_0 > 0$, $\mu \in \mathbb{R}$, $\sigma > 0$ and time horizon $T > 0$.

(a) Show that $\log\left(S_{t_M}^M\right) \xrightarrow[M\to\infty]{} Z$ in law, where the random variable Z is normally distributed with mean $\log(S_0) + \left(\mu - \frac{1}{2}\sigma^2\right)T$ and variance $\sigma^2 T$.

(b) Conclude that the initial prices of European put and call options with maturity T and strike $K > 0$ in the binomial model with M timesteps approximating the risk-neutral dynamics of the Black-Scholes model converge to the corresponding Black-Scholes prices as $M \to \infty$.

Without proof, you can use

Slutsky's Theorem: Let $(A_n)_{n\in\mathbb{N}}$, $(B_n)_{n\in\mathbb{N}}$ and $(X_n)_{n\in\mathbb{N}}$ be sequences of random variables such that $A_n \xrightarrow[n\to\infty]{} A$, $B_n \xrightarrow[n\to\infty]{} B$ in probability and $X_n \xrightarrow[n\to\infty]{} X$ in law. Then $A_nX_n + B_n \xrightarrow[n\to\infty]{} AX + B$ in law.

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, 02.05.2024, 12:00