Christian-Albrechts-Universität zu Kiel **SS 2024**
Mathematisches Seminar
Prof. Dr. Mathias Vetter
Henrik Valett, Fan Yu, Ivo Richert, Anton Schellin Sheet 02

Computational Finance

Exercises for all participants

T-Exercise 05 (4 points)
We want to price an American put option with strike price $K = 1.2$ and time to maturity being three years. For this purpose we want to utilize a CRR model with $M = 3$ equally spaced time periods, $S(0) = 1$, $\sigma^2 = 0.3$ and an annual interest rate of 5%.

a) Draw and calculate the corresponding CRR model by hand (of course you can still use a calculator) and write beneath each point the corresponding price of the option (please round on four position after the decimal point after each calculation).

b) Calculate the replicating portfolio $\varphi = (\varphi_0, \varphi_1)$ for all time periods.

C-Exercise 06 (Options in the CRR model) (4 points)

(a) Write a Python function

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

that computes and returns an approximation to the price of a European or an American put option with strike $K > 0$ and maturity $T > 0$ in the CRR model with initial stock price $S(0) > 0$, interest rate $r > 0$ and volatility $\sigma > 0$. The parameter "EU" is 1 if the price of an European put shall be computed or is 0 in the American case. Use the binomial method as presented in the course with $M \in \mathbb{N}$ time steps.

(b) As $M \to \infty$ we would expect convergence of the price in the binomial model towards the price in the Black-Scholes model. To show this implement the BS-Formula for European put options as a Python function:

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

(c) Test your algorithm with

$$S(0) = 100,\ r = 0.05,\ \sigma = 0.3,\ T = 1,\ M = 10, \ldots, 500,\ K = 120.$$

- Plot the price of a European put option (EU = 1) in the binomial model in dependence on the number of steps M.
- Plot the fair price in the BS-model into the same plot using the same parameters.
- Print the price of an American put option (EU = 0) with the same $S(0), r, \sigma, T, K$ as above and $M = 500$ steps in the console.

Useful Python commands: `numpy.maximum`

C-Exercise 07 (Valuation of European options in the Black-Scholes model using Monte-Carlo) (4 points)
Write a Python function

```
Eu_Option_BS_MC (S0, r, sigma, T, K, M, f)
```

that computes the initial price $V(0) = e^{-rT}\mathbb{E}_Q[f(S(T))]$ of a European option with payoff $f(S(T))$ at maturity T for some strike price in the Black-Scholes model and the asymptotic 95%-confidence interval $[c_1, c_2]$ via the Monte-Carlo approach using $M \in \mathbb{N}$ simulations. In the B-S model we have the formula

$$S(T) = S(0)\exp\left((r - \sigma^2/2)T + \sigma\sqrt{T}X\right),$$

where X has law $N(0,1)$ under Q.
Test your function for a call option $f(x) = (x-100)^+$, $S_0 = 110$, $r = 0.04$, $\sigma = 0.2$, $T = 1$ and $M = 10000$ and compare the price with the BS-Formula.

Useful Python command: `numpy.random.normal`
Hint: The Black-Scholes formula for the European Call is given on exercise sheet 02.

T-Exercise 08 (Barrier options in the CRR model) (for math 4 points; for QF 4 bonuspoints)
In the binomial model from Section 2.1 with parameters $S(0), r, \sigma, T > 0$ and $M \in \mathbb{N}$, we denote by

- V the fair price process of a *European call option* on the stock S with strike $K > 0$, i.e. its payoff is given by $V(T) = (S(T) - K)^+$,
- $\tilde{V}$ the fair price process of a *down-and-out call option* on the stock S with strike $K > 0$ and barrier $B < K$, i.e. its payoff is given by

$$\tilde{V}(T) = 1_{\{S(t_i) > B \text{ for all } i=0,\ldots,M\}}(S(T) - K)^+,$$

- $\hat{V}$ the fair price process of a *down-and-in call option* on the stock S with strike $K > 0$ and barrier $B < K$, i.e. its payoff is given by

$$\tilde{V}(T) = 1_{\{S(t_i) \leq B \text{ for one } i=0,\ldots,M\}}(S(T) - K)^+.$$

Outline (e.g. in pseudo code) an algorithm that computes the initial price $\tilde{V}(0), \hat{V}(0)$ of the barrier options in $O(M^2)$ steps, i.e. there is a constant $C > 0$ independent of M such that the algorithm terminates after less than CM^2 operations. Please explain where your algorithm differs from the algorithm for European call options presented in the lecture and why these changes make sense/are needed.

Hint:

(a) Start with the *down-and-out call option*.

(b) Express the *down-and-in call option* in terms of a *down-and-out call option*.

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, 10.05.2024, 12:00