Christian-Albrechts-Universität zu Kiel **SS 2016**
Mathematisches Seminar
Prof. Dr. Jan Kallsen
Mark Feodoria Sheet 06

Computational Finance

Exercises for participants of mathematical programmes

C-Exercise 20 (Barrier option in the CRR model)
In the binomial model from Section 2.1 with parameters $S(0)$, r, σ, $T > 0$ and $M \in \mathbb{N}$, we denote by V the fair price process of an *up-and-out put option* on the stock S with strike $K > 0$ and barrier $B > K$. I.e., its payoff is given by

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

Write a scilab function

```
V0 = UpOutPut_BinMod (S_0, r, sigma, T, K, B, M)
```

that computes and returns the fair value at time $t_0 = 0$ of the up-and-out put option.

Test your function with

$$S(0) = 100,\ r = 0.05,\ \sigma = 0.2,\ T = 1,\ K = 100,\ B = 110,\ M = 1000.$$

C-Exercise 21 (Implied volatility in the CRR model)
In C-Exercise 18 you implemented a function that computes the fair price of an American put option. Fix the parameters $S(0)$, r, T, K and M as in C-Exercise 18. Instead of computing the option price $V(0)$ for given volatility σ, we now look for an algorithm computing the volatility σ that yields a given option price $V(0)$. To this end, write a scilab function

```
sigma = AmPut_BinMod_ImplVol (S_0, r, T, K, M, V0)
```

that computes and returns this so-called *implied volatility*.

Test your function with

$$S(0) = 100,\ r = 0.05,\ T = 1,\ K = 100,\ M = 500,\ V(0) = 6.09$$

Useful scilab commands: `fsolve`, `leastsq`

Please turn over.

C-Exercise 22 (Adaptive step size control for the binomial method)
We extend the algorithm of C-Exercise 18 by choosing the number of periods adaptively in the following way. All other parameters being fixed, we compute the fair price of the American put option for M and $2M$ periods, respectively. The corresponding prices are denoted by V_M and V_{2M}. If

$$\frac{|V_M - V_{2M}|}{V_{2M}} < \varepsilon$$

for some fixed accuracy $\varepsilon > 0$, the algorithm accepts V_{2M} as option price and returns this value. In the other case, we double the number of periods and compute the relative deviation of V_{2M} and V_{4M}. If these values satisfy the above termination condition, the value V_{4M} is returned. If not, the number of periods is doubled until the termination condition is satisfied, and the corresponding value is returned. Implement this algorithm in a scilab function

```
V0 = AmPut_BinMod_Adapt (S_0, r, sigma, T, K, M, epsilon),
```

and test it for

$$S(0) = 100,\ r = 0.05,\ \sigma = 0.2,\ T = 1,\ K = 100,\ M = 100,\ \varepsilon = 0.001.$$

T-Exercise 23 (Down-and-in barrier option in the CRR model)
In the binomial model from Section 2.1 with parameters $S(0)$, r, σ, $T > 0$ and $M \in \mathbb{N}$, we denote by V the fair price process of a *down-and-in put option* on the stock S with strike $K > 0$ and barrier $B < K$ (e.g., $K = 100$ and $B = 90$). I.e., its payoff is given by

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

Outline (e.g., in pseudo code) an algorithm that computes the initial price $V(0)$ of the down-and-in put 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 save your solution of each C-Exercise in a file named `Exercise_##.sce`, where ## denotes the number of the exercise. Please include your name(s) as comment in the beginning of the file.

Submit until: Thursday, 02.06.2016, 08:30
Discussion: in the tutorial on Mon, 06.06.2016