

Stochastic Methods + Lab

Assignment Sheet 4

Due on October 14, 2019

Problem 1 [14 points]

Implement the binomial tree in python via backwards induction. Use only one **for** loop to go from one step to the previous one, and implement all other operations with vectors. In more detail, implement a function

```
binomial_tree(payoff, n, rp, sigma, S, K, T)
```

that returns the price of the option at time $T = 0$. The arguments of the function are:

- **payoff**, a function that takes the stock price S (possibly a vector) and strike price K as arguments and returns the payoff,
- **n**, the number of steps,
- **rp**, the risk-free period interest rate,
- **sigma**, the volatility,
- **S**, the initial stock price,
- **K**, the strike price,
- **T**, the maturity.

Use the calibration of the model that we discussed in class, i.e., use the parameters

$$u = \frac{1}{d} = \exp\left(\sigma\sqrt{\frac{T}{n}}\right).$$

Test your code by pricing a European call option with strike price $K = 0.8$, risk-free interest rate $r = 0.02$, volatility $\sigma = 0.4$, maturity $T = 1$, and initial stock price $S = 1$, using $n = 1000$ steps.

Problem 2 [6 points]

The price of a European Call option with current stock price S , strike price K , annualized volatility σ , annual risk-free interest rate r , and maturity time T can be computed explicitly with the Black-Scholes formula

$$C = S\Phi(x) - Ke^{-rT}\Phi(x - \sigma\sqrt{T}),$$

where

$$x = \frac{\ln(S/K) + (r + \sigma^2/2) T}{\sigma\sqrt{T}},$$

and Φ denotes the cumulative distribution function of the standard normal distribution with mean zero and variance one. Compare your call option prices from the binomial tree model with n steps against those computed by the Black-Scholes formula. Plot the logarithm of the error vs. n . Do you roughly obtain a straight line? If so, with what power of n does the error scale?

Choose parameters $S = 1$, $K = 1.2$, $\sigma = 0.5$, $T = 1$, and $r = 0.03$ (for which the option price is 0.1410).