Practical Optimization in Finance, Fall 2001

Project 2 (to be handed in November 15)

Short and sweet: Implement the binomial pricing algorithm presented at the lectures & in the notes using your favorite piece of software.

To get the ball rolling you may want to consider the Visual Basic code in MATLAB; even if you don’t know Visual Basic, you should be able to understand it as “pseudo-code”.

You should allow for arbitrary option expiry dates \( T \); measured in some time-unit, say years) and number of time-periods per year, \( n \), in the binomial lattice (so you can vary these independently). A sensible choice of 1-period interest rate comes from

\[
1 + r_n = \exp(r_c/n),
\]

where the constant \( r_c \) is the continuously compounded interest rate. You can start out by using

\[
u_n = 1/d_n = \exp(\sigma/\sqrt{n}),\]

where \( \sigma \) is a (positive) constant called volatility.

Find call-option prices for some parameter choice of your liking; for instance the one in MATLAB for different values of \( n \), but fixed \( T \) (or in short: “Let \( n \to \infty \”). Plot call-option prices for \( n = 2, 3, \ldots, 100 \) (say). What do you see?

How does the call-option price depend on volatility?

Modify your code so that it can price put-options. (A put-option is the right, but not obligation, to sell the stock at a future date at a pre-specified (strike) price \( K \). In other words the pay-off of a put-option is \((K - S(T))^+\).)

Try using

\[
u_n = \exp(\mu/n + \sigma/\sqrt{n}),\]
\[d_n = \exp(\mu/n - \sigma/\sqrt{n}),\]

where \( \mu \) is a constant, 0.10 for instance. How does \( \mu \) affect future stock prices?

Let \( n \to \infty \), find call-option prices and compare to those you found earlier.

How do prices depend on \( \mu \)?

For a specific call-option with strike \( K \), try using

\[
u_n = \exp(\ln(K/S(0))/(nT) + \sigma/\sqrt{n}),\]
\[d_n = \exp(\ln(K/S(0))/(nT) - \sigma/\sqrt{n}).\]

Plot call-option prices for \( n = 2, 3, \ldots, 100 \) (say). What do you see?

Kindly,

Ralf