Practical Optimization in Finance, Fall 2001

Suggestion for Project 3: Simulation; the Black-Scholes Formula, & Hedging

Make a program that can simulate paths of the stock-price in the Black-Scholes model. (Just focus on the $Q$-measure version; but in fact the last bit will work for any measure, i.e. $r$ substituted by any other constant in the $S$-definition.) For this you need to simulate the outcome of a $N(0,1)$ variable. In Excel \texttt{RAND()} produces (independent) $U(0,1)$-variables (say $U_1$, $U_2$, ...). There are several ways of transforming such variables in $N(0,1)$'s. A quick & dirty one is $X = \sum_{i=1}^{12} U_i - 6$. A slow & correct one is $X = \text{NORMSINV} (\text{RAND()}))$. A seemingly mysterious one is the Box-Muller transform $(X_1, X_2) = (\sqrt{-2 \ln U_1 \cos(2\pi U_2)}, \sqrt{-2 \ln U_1 \sin(2\pi U_2)})$, which produces two independent $N(0,1)$ variables.

Build into the same program the Black-Scholes (call-option) formula.

Choose some parameters for for model. Simulate (many) outcomes of $S(T)$ and find an estimate of the call-option price from the discounted sample average. How many simulations do you need to get a relative accuracy of 1%? 0.1%? (In terms of, say, 95% confidence intervals. Basic statistics here, folks!)

We now want to investigate hedging in the Black-Scholes model. (To refresh you memory regarding the workings of a hedging strategy, you may want to look at notes from the lectures on October 24.) In the binomial model we had to hold

$$\Delta(t) = \frac{\Delta C}{\Delta S}$$

units of the stock in order to replicate. One can also show that in order to replicate the in the Black-Scholes model one has to hold

$$\Delta(t) = \frac{\partial C}{\partial S} = \Phi \left( \frac{\ln(S(t)/K) + (r + \sigma^2/2)(T-t)}{\sigma \sqrt{T-t}} \right) \exp \left( -d(S(t), t) \right)$$

1
where $\Phi$ is the distribution function of the $N(0,1)$-distribution (i.e. $\Phi(x) = P(X \leq x)$ for $X \sim N(0,1)$; $\Phi$ is called NORMSDIST in Excel).

The aim now is to validate this by simulation. To do this consider the following algorithm:

- Choose some parameters in a Black-Scholes model. Chop the time interval between now (time 0) and expiry of the call-option (time $T$) into $N$ pieces; denote the discretization points $t_i$.

- Simulate a stock-price path $(S^{\text{sim}}(t_i))^N_0$ (i.e. values at the $t_i$’s).

- Suppose that at time 0 somebody gives you an amount of money exactly equal to the Black-Scholes call-price. You use this to buy $\Phi(d_1(S(0), 0))$ units of the stock (whatever extra money you need, you borrow in the bank).

- At time $t_i$ you adjust your portfolio such that you now hold $\Phi(d_1(S^{\text{sim}}(t_1), t_1))$ units of the stock. You do this in such a way that extra funds needed (+/-) are borrowed at the bank.

- Do that $t_2, t_3, t_{N-1}$ & keep track of the value of the portfolio, say $V(t_i)$. (Recall that money in the bank draws interest.)

- Compare $V(t_N)$ to $(S^{\text{sim}}(t_N) - K)^+$; call the difference $\epsilon_N^{\text{sim}}$

- Repeat over many simulated paths. Plot $V(t_N)$ and $(S^{\text{sim}}(t_N) - K)^+$ against each other. Estimate mean & standard deviation of $\epsilon_N$.

- Repeat with larger $N$; how does mean & standard deviation of $\epsilon_N$ depend on $N$?

(Hint: In H&P there’s a VBA-function that does most of this.)