CS 870 Assignment 1

Winter 2005

Instructor: P.A. Forsyth Office: DC3631 e-mail: *paforsyt@uwaterloo.ca*
Classroom: DC3313: Tues-Thurs: 4:30-6:00
Office Hours: Wed: 4:30-5:30
Web Site: http://www.scicom.uwaterloo.ca/~paforsyt/870.html

Assignments 1 and 2 are not to be handed in, but are meant to give you a head start on the course project. If you have any problems, be sure to come and see me. On the course web site is a list of references for more information about Monte Carlo methods.

Monte Carlo Option Pricing

The objective of this assignment is to use Matlab to familiarize yourselves with Monte Carlo option pricing.

As we derived in class, for the purposes of pricing options, we can pretend that the asset price S evolves in the risk-neutral world:

$$dS = rSdt + \sigma SdZ \tag{1}$$

where r is the risk free return, σ is the volatility, and dZ is the increment of a Weiner process.

Let the expiry time of an option be T, and let

$$\begin{aligned} N &= \frac{T}{\Delta t} \\ S(n\Delta t) &= S^n \end{aligned} \tag{2}$$

Then, given and initial price S^0, M realizations of the path of a risky asset are generated using the algorithm

$$S^{n+1} = S^n + S^n(r\Delta t + \sigma\phi\sqrt{\Delta t}) \tag{3}$$

where

$$\phi = \text{normal random variable with mean zero and unit variance}$$

If the $m'th$ value of $S^N = S(T)$ is denoted by $(S^N)^m$, then an approximate value of the option is given by (assuming that $S(0) = S^0$)

$$V(S^0, t=0) = e^{-rT}\frac{\sum_{m=0}^{m=M} Payoff((S^N)^m)}{M} \tag{4}$$

Code up this algorithm in Matlab. Some typical values of the parameters are given in Table 1. Using these parameters, price European puts and calls using the Monte-Carlo algorithm. Carry out convergence studies by reducing Δt and increasing M. How does it compare with theory? For European options, you can obtain the exact solution (see the Matlab function *blsprice*). You might start out using a timestep of $1-2$ days.

Experiment with different values of r, σ, T.

Table 1: Some typical option parameters

σ	.3
r	.06
Time to expiry	0.25 years
Strike Price	\$100
Initial asset price S^0	\$100

For the final project, you will have to do this very efficiently. You should try to avoid loops in Matlab, i.e. at each timestep, generate a vector of many paths at the same time. See the paper by Higham in Siam Review (2001) (full reference below). You should be able to get a simulation with 10,000 paths, each path with 100 timesteps, to run quite quickly.

Note that for the special case of constant coefficients, we can avoid timestepping errors for geometric Brownian motion, since we can integrate equation (1) exactly to get

$$S(t) \quad = \quad S(0) \exp[(r - \sigma^2/2)t + \sigma\phi\sqrt{t}] \tag{5}$$

where equation (5) is exact for any t. Repeat the above experiments and avoid timestepping by using equation (5).

However, this trick only works for simple cases (i.e. constant coefficients), and so, in general, we have to step through time. For example, it is common to use a volatility surface $\sigma(S, t)$ to match observed prices in the market, so the integration can't be done exactly in this case.

Binomial Tree

Code up a binomial tree (lattice) method for valuing a European option. The algorithm for a binomial method is given in the course notes. Note that you should be able to code up a European pricer using only two arrays of length N, where N is the number of timesteps. Compare Monte Carlo with a no-arbitrage binomial tree (lattice) in terms of running time. How would you change this code to handle an American option? Which is faster, the tree method or Monte Carlo?

Implied Volatility

You can get option prices on the Web from various sources. If you google *Mark Broadie* a Professor of finance at Columbia, he has links to many of the major exchanges. Pick your favourite stock, and determine the implied volatility from the quoted price. For calls, if the stock pays no dividends, then it is not optimal to exercise early, so the price should be the *blsprice*.

- Compute the price by taking the midpoint of the bid-ask spread.
- The time to maturity, and exercise price should be listed.
- The value of S_0 is the current stock price.
- For the maturity specified, find the yield for a Government T-bill (with the same maturity). Use this value for r.
- Now, use a Newton iteration method to determine the volatility which reproduces the observed price.

- Try this for various strikes and maturities.

The Newton iteration works as follows. Let the model price for a fixed T, r, S_0 be denoted by $P(\sigma)$. For different values of σ, you will get a different price. The objective is to determine σ which matches the current market price P^*. In order words, we want to find σ which satisfies the equation

$$F(\sigma) \quad = \quad P^* - P(\sigma) = 0 \tag{6}$$

which can be done numerically using Newton iteration. Or, you can try using one of the Matlab zero finders. If you need more help, read "The calculation of implied variances from the Black-Scholes model," J. Finance, vol 37, March, 1982, pages 227-230, by Manaster and Koehler.

Most options traded are American (except for $S\&P500$ index options which are European). Repeat the above tests, only this time use put options (almost all options on stocks in the TSX are American). You won't be able to use the matlab routine *blsprice* in this case. Use your binomial American tree as the pricing engine, and repeat the above steps, using Newton iteration to determine the implied volatility. Note that if it is optimal to exercise, the price becomes independent of the volatility, so be careful.

Some more things to try

1. Read the paper *An algorithmic introduction to numerical solution of stochastic differential equations,* by D. Higham, SIAM Review, vol 43 (2001) 525-546. This article has some good tips on how to write fast Monte Carlo code in Matlab. Note that if you use a lot of *for* loops, your code will run slowly. You should try to take advantage of the vectorization capabilities of Matlab. This will be important for your project. Higham has also written a good introductory text *"An introduction to option valuation,"* D. Higham, Cambridge Press. This book has many tips of efficient use of Matlab.

2. Some tips on implementation of the binomial tree method are given in *Nine ways to implement the binomial tree method in Matlab,* D. Higham, SIAM Review, vol 44 (2002) 661-667.

3. Carry out some Monte Carlo tests to verify that

$$\begin{aligned} \lim_{\Delta t \to 0} \sum_i \Delta Z_i^2 \quad &= \quad t \\ & \Delta Z_i = Z(t_{i+1}) - Z(t_i) \\ & t_{i+1} - t_i = \Delta t \\ & \sum_i \Delta t = t \end{aligned} \tag{7}$$

   where ΔZ_i is the increment of a Weiner process.

4. Read the course notes on strong and weak convergence, and Brownian Bridges. You might want to look up Milstein's method for stochastic integration.

5. You can look at some good C code for Random number generation. On any undergrad machine, copy the README file from the cs870 course account by:

```
cp ~cs870/Random/README
```

This is the Mersenne Twistor algorithm.

6. You might try speeding up convergence of Monte Carlo, using an antithetic variable approach. (This does not always work).

7. Consider an option on two risky assets S_1, S_2, where the payoff is

$$Payoff = \max(\max(S_1, S_2) - K, 0) \tag{8}$$

where K is the strike price. In the risk neutral world, we can consider the two assets to follow:

$$\begin{aligned} dS_1 &= rS_1 dt + \sigma_1 S_1 dZ_1 \\ dS_2 &= rS_2 dt + \sigma_2 S_2 dZ_2 \end{aligned} \tag{9}$$

where $E(dZ_1 dZ_2) = \rho$, where ρ is the correlation between these two Wiener processes. ($-1 \leq \rho \leq +1$). Price a call option on the best of two correlated assets using a Monte Carlo method. Carry out a convergence study.

8. Consider an Asian option, with payoff depending on the average asset price, where the discretely observed average after N observations A_N is defined as

$$A_N = \frac{\sum_{i=1}^{i=N} S_i}{N}$$

where $S_i = S(t_i)$, where t_i are the observation dates. Typically, the interval between observations is one day. Price an Asian fixed strike call option using a Monte Carlo method, where the payoff is

$$payoff = \max(A_N - K, 0)$$

where K is the strike.

9. Try pricing a Barrier option. Suppose the strike is \$100, and there is a barrier at \$120. Price a European barrier call option, where if the asset is observed over the barrier (at the close each day), then the option expires as worthless.