CS 774 Project: Fall 2009

Version: November 27, 2009

Instructors:
Peter Forsyth, *paforsyt@uwaterloo.ca* Office: DC3631
Office Hours: Tues: 4:00-5:00; Thurs: 11:00-12:00
Lectures:MWF 3:30-4:20 MC2036

CS 774 Web Site: www.student.cs.uwaterloo.ca/∼cs774/

Due Date, Project Format

This project is due at noon, December 21, in my office. Incompletes will not be given out except in the case of medical problems. Hand in what you have done by the due date.

- Hard copy must be handed in. Do not email me a pdf file.
- The entire project must no more than 25 typeset pages, including figures and tables. At least 10pt font must be used. This does not include hard copy of your code.
- In addition to the above, submit hard copies of your code. Use some formatting tool for your code. On Linux, use something like

```
enscript --fancy-header --pretty-print --landscape --borders  --columns=2
```

Project components

This project consists of two parts: a set of assigned tasks where you simulate the expected returns from discrete delta hedging, and a *research* section where you can carry out an in-depth study of an area of computational finance. I will list a series of suggested topics for the research section, but these are suggestions. You are free to come up with other ideas. Before starting on the research project, you should have handed in the project proposal and received feedback from me.

Marking

About 50% of the marks for this project will be allocated for the assigned tasks, and 50% for the research section. The project should be typeset, and figures and graphs carefully thought out to present the data and your conclusions in an effective and clear manner. Poor presentation of your work will result in a poor mark. I will put a marking scheme on the Web site for the assigned tasks in a few weeks.

In all cases, I expect you to explain your algorithms, and describe what you see in detail. You should also submit hard copies of your code, along with some documentation.

Even for the assigned tasks, some of the questions are open-ended, so I will give more marks to answers which demonstrate a thorough investigation of the issues.

I expect an in-depth discussion of all the numerical issues, not just simply doing a few tests and presenting the results. I am going to be very particular about this this year. Some of the projects that I have received in past years were very minimal in terms of discussion. This will not go unnoticed this year.

Matlab has good plotting facilities. Create figures with Matlab to include in your document.

Assigned tasks: simulated hedging

The basic idea of these tasks is to simulate a discrete hedging strategy. Assuming that

$$dS = \mu S dt + \sigma S dZ \tag{1}$$

where μ is the drift rate, S is the asset price, σ is the volatility, and dZ is the increment of a Wiener process, then the no-arbitrage price of an American option is given by

$$\begin{aligned} -\frac{\partial V}{\partial \tau} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV &\leq 0 \qquad (2) \\ V^* &\leq V \qquad (3) \end{aligned}$$

where r is the risk-free rate of interest, and where one of (2-3) holds with equality. V^* is the payoff condition.

1. First, show that you have validated your code for pricing American options using a PDE method (from Assignment 2). Show convergence as the grid/timestep is refined, number of penalty iterations, etc.

   Note that for the next task, you will need to keep track of whether or not it is optimal to exercise at any given point.

2. Using the data given in Table 1, determine the expected hedging error using discrete delta hedging. First, solve for the value of an American put, backwards in time, and, for each discrete time, and value of the asset, store the value of the option, as well as the option delta, in a large table. Use C-N timestepping, and enforce the early exercise constraint implicitly. The delta can be approximated by finite differences.

   Discrete delta hedging consists of a portfolio $P(t)$ with components:

   - A short option position $-V(t)$.
   - Long $\alpha(t)S(t)$ shares
   - An amount in a risk-free bank account $B(t)$.

   Intially, we have

$$\begin{aligned} P(0) = 0 &= -V(0) + \alpha(0)S(0) + B(0) \\ &\quad \alpha = V_S \\ &\quad B(0) = V(0) - \alpha(0)S(0) \end{aligned}$$

   The hedge is rebalanced at discrete times t_i. Defining

$$\begin{aligned} \alpha_i &= V_S(S_i, t_i) \\ V_i &= V(S_i, t_i) \end{aligned}$$

   then, we have to update the hedge by purchasing $\alpha_i - \alpha_{i-1}$ shares at $t = t_i$, so that updating our share position requires

$$S(t_i)(\alpha_i - \alpha_{i-1})$$

in cash, which we borrow from the bank. If $\Delta t = t_i - t_{i-1}$, then the bank account balance is updated by

$$B_i = e^{r\Delta t} B_{i-1} - S_i(\alpha_i - \alpha_{i-1})$$

At the instant after the rebalancing time t_i, the value of the portfolio is

$$P(t_i) = -V(t_i) + \alpha(t_i)S(t_i) + B(t_i)$$

Now, simulate the path of an asset price forward in time using a Monte Carlo method. At each discrete hedging interval, update your portfolio, using the current asset price, and the corresponding value of delta (you will have to interpolate these values from your stored table). Note that for some values of the asset price, it will be optimal to exercise the option, so your simulation will stop at that point. For each simulated path, determine the discounted relative hedging error

$$error = \frac{e^{-rT^*} P(T^*)}{V(S_0, t = 0)} \qquad (4)$$

where T^* is expiry time of the option or the exercise time (if the option was exercised before expiry). Do not update your hedge at $t = T^*$, i.e.

$$P(T^*) = -V(T^*) + \alpha(T^* - \Delta t)S(T^*) + B(T^* - \Delta t)e^{r\Delta t}$$

NOTE: The hedging interval Δt is not necessarily the same as the timestep in your PDE or Monte Carlo algorithms.

Plot some representative histograms of the relative hedging error, i.e. the number of Monte Carlo trials giving hedging errors between E and $E + \Delta E$. Compute the standard deviation, VAR (95%) and CVAR (95%) for a series of tests with different sizes of hedging interval. The standard deviation of the hedging error (relative profit and loss) should be $O((\Delta t_{hedge})^\beta)$ as $\Delta t_{hedge} \to 0$. Estimate β (a log-log plot would be useful).

Note: you should be using *at least* 10,000 simulations to get a reasonable histogram. If you are using Matlab, refer to hints in Assignment 1 to get this to run quickly.

You will need to construct tables which have a range of S values for your hedging simulations. A reasonable range can be estimated from

$$\begin{aligned} X_{max} &= (\mu - \frac{\sigma^2}{2})T + \kappa\sigma\sqrt{T} \\ X_{min} &= (\mu - \frac{\sigma^2}{2})T - \kappa\sigma\sqrt{T} \\ S_{max} &= S_0 e^{X_{max}} \\ S_{min} &= S_0 e^{X_{min}} \end{aligned} \qquad (5)$$

where $\kappa = 3$ is a good estimate. A reasonable number of entries in the range$[S_{min}, S_{max}]$ would be $\simeq 100$. Generate this data at each time point you will need in the hedging simulation.

3. Consider the jump diffusion process

$$\frac{dS(t)}{S(t^-)} = \mu dt + \sigma dZ + (\eta - 1)dq \qquad (6)$$

where

μ is the drift rate,

dq is the independent Poisson process, $= \begin{cases} 0 & \text{with probability } 1-\lambda dt \\ 1 & \text{with probability } \lambda dt, \end{cases}$

λ is the mean arrival time of the Poisson process,

$\eta - 1$ is an impulse function producing a jump from S to $S\eta$,

σ is the volatility,

dZ is an increment of the standard Gauss-Wiener process.

Note that the increments in dS depend on the value of S just before a potential jump at t, not on the value just after the jump.

Assume that the jump size η is lognormal, i.e. the density for $x = \log \eta$ is given by $g(x)$, $x \in [-\infty, +\infty]$.

$$g(x) = \frac{e^{\left(-\frac{(x-\mu_J)^2}{2\gamma^2}\right)}}{\sqrt{2\pi}\gamma}. \tag{7}$$

This model is known as the Merton jump diffusion model. There is a closed form expression for the price of a European put/call under the Merton jump diffusion model. You can find this in many textbooks, or you can read the orginal paper *Option pricing when underlying stock returns are discontinuous*, R.C. Merton, Journal of Financial Economics, (1976) Vol 3:125-144. There is an M-file for this pricing model on the course Web site.

Notation note: μ is commonly used for the drift rate and the mean of the Merton jump size. This is confusing in the literature. Note μ is the drift rate and μ_J is the mean of the Merton jump size in this project description.

Now, develop a Monte Carlo function in Matlab, whereby you directly simulate the SDE (6). Use forward Euler timestepping, and change variables to $X = \log S$, i.e. your timestepping algorithm should be written like

$$X(t + \Delta t) \quad = \quad X(t) + \ldots \tag{8}$$

where $S(T) = \exp(X(T))$. You can assume $\lambda \Delta t \ll 1$. Glasserman's book is an excellent reference on this topic. Note that the risk neutral drift rate for pricing is $\mu = r - \lambda\kappa$ where

$$\kappa = E[\eta - 1] = \exp[\mu_J + \gamma^2/2] - 1 \tag{9}$$

since, in the risk neutral world $E[dS/S] = r\ dt$.

Check your Monte code by comparing with the analytic solution.

Now, carry out a study of Delta hedging, assuming the the real world follows a jump diffusion. You will need to construct tables of delta values for the option. Precompute these using the analytic solution (Assume that the target option is European). Then, simulate the delta hedge using Monte Carlo simulation.

Plot histograms of the hedging error (as in Task 2), for various hedging intervals. Use the data in the file *jump_hedge_data.m* on the course Web site.

Table 1: A typical test case

σ	.25
r	.05
Time to expiry	0.25 years
Strike Price	\$100
Initial asset price S^0	\$100
Drift rate	.07

Research tasks

This part of the project is much more open ended. Your mark for this section will be based on a thorough, creative job, not just the minimal suggestion listed below! Be sure to read some additional papers. Do not just do some implementation. You will need to make some interesting comments.

Here are some suggestions (to get started). Come and see me if you want some more ideas.

- There are many interesting problems in finance which give rise to nonlinear Hamilton Jacobi Bellman (HJB) PDEs. To get started, you can read the paper *Numerical Methods for Controlled HJB equations in finance* on my Website. There are also many other papers on this topic on my Website.

- Use a low-discrepancy sequence to carry out the simulations for the hedging strategies described above. You will need to use a high dimensional Sobol (or some other) sequence generator. Experiment with an LDS timestepping method for pricing an Asian or Parisian option.

- Develop PDE code for pricing a discretely observed Parisian or Asian option (a set of 1-d PDE solves). Can you extend this to an American- Asian option? How about Parisian-Asian options? Lookbacks?

- Develop a PDE code for pricing convertible bonds (see notes).

- Investigate semi-Lagrangian timestepping as an alternative to upwinding for drift dominated problems. See the paper on Asian options on my Website.

- Look at the embedded option in hedge fund fees.

- Recently, jump diffusion models have been suggested as a way to improve the fit of observed stock prices, better than simple geometric Brownian motion. Can you develop a PDE algorithm for pricing American options with jump diffusion?

- Develop code for pricing zero coupon bonds, assuming the CIR single factor interest rate model. Extend your algorithm to handle coupons. Discuss how you handle the boundary conditions. There is a paper on my Web site about callable bonds which will help you get started. Can you handle a finite notice period?

- Investigate SDE timestepping methods for the CIR model. Some references are in Assignment 1.

- Develop code for modelling uncertain volatility or transaction cost models. There is a paper on my web site about this. Validate the transaction cost model through MC simulation.
- Implement a two factor lattice method. On my website, download the paper *Negative coefficients in two factor option pricing models*. This paper describes several two factor lattice methods. Compare with Monte Carlo. Carry out a study of discrete two factor delta hedging.
- Carry out a study of methods for computing the Greeks assuming you are valuing options using MC. Read the references on the Web site.
- Look at FFT timestepping methods. (Jackson et al, Journal of Computational Finance Vol 12 (Winter), 2009, pages 1-29; Lord et al, SIAM J. Sci. Comp. Vol 30 (2009), pages 1678-1705).
- Robust portfolio optimization.