Project on Numerical Methods for Option Pricing

1 Pricing a Vertical Spread (50% of project credit)

1.1 Overview

A vertical spread is strategy whereby the option trader purchases a certain number of options at strike price K_1 and simultaneously sells an equal number of options of the same underlying asset and same expiration date, but at a different strike price K_2. We will consider a bull call spread for which the trader buys and sells call options with $K_1 < K_2$.

The aim of this part of the project is to price such a bull call spread by a variety of methods and to compare and contrast these methods.

We consider that the underlying asset $S(t)$ follows geometric Brownian motion

$$dS(t) = rS(t)dt + \sigma S(t)dW,$$

where the risk-free interest rate r and volatility σ are constants. The initial condition is $S(t = 0)$. Time t is measured in years.

The payoff for the bull call spread is

$$\max(S(T) - K_1, 0) - \max(S(T) - K_2, 0), \tag{1}$$

where $T > 0$ is the maturity of the option.

Such options can be valued by solving the Black-Scholes PDE, by Monte Carlo simulation of the underlying process, or simply by appropriately using Black-Scholes formulae.

1.2 Particulars

Use the following parameters

1. The strike prices are $K_1 = £90$ and $K_2 = £120$.
2. Time to maturity (or expiry), T, is 6 months.
3. The interest rate is $r = 0.03$.
4. The volatility is $\sigma = 0.25$.

You will want to price the bull call spread over a range current asset prices from $S = £1$ to $S = £200$.

1.3 Tasks

- For solving the Black-Scholes PDE you must work on a finite interval of asset price: $S \in [S_{\min}, S_{\max}]$. The boundary conditions for this finite interval are

$$V(S_{\min}, t) = 0 \quad V(S_{\max}, t) = (K_2 - K_1)e^{-r(T-t)}. \tag{2}$$

  Implement a transformation between the Black-Scholes equation and the heat equation and from this derive the initial condition and boundary conditions for the heat equation corresponding to the payoff (1) and to boundary conditions (2). You will needs these in the next step.

  Hint: There are two strike prices and so you have freedom in defining the transformation.

  Note: we will only be interested in pricing the bull call spread from $S = £1$ to $S = £200$. However, you will need to use $S_{\max} = £500$ in solving the Black-Scholes PDE.

- Using the information in the previous step, implement the Crank-Nicolson method on the Black-Scholes PDE for pricing the bull call spread. Use a direct solver for solving the linear system of equations.

- For $S_{\min} = £1$ and $S_{\max} = £500$, investigate the accuracy of your solution as you refine the number of time steps N and the number of grid points J. From this study of refinement, choose value of N and the number of grid points J such that maximum absolute error of your numerical solution is less than $£0.05$. This means that $\|V_{\text{PDE}} - V_{\text{exact}}\|_\infty < 0.05$, where V_{PDE} is the numerical solution from the Black-Scholes PDE and V_{exact} is the solution from the Black-Scholes formula.

- For the N and J found in the previous item, determine the CPU time required to solve the Black-Scholes PDE. (The CPU times may be very short and you will have to work out a way to get accurate times. See Matlab documentation on measuring performance of your program.)

- Compute the delta of the bull call spread. Specifically, use a finite-difference representation for the first derivative to numerically differentiate your solution of the Black-Scholes PDE.

- Implement the Monte Carlo method for pricing the bull call spread:
  - Without any variance reduction (naive method);
  - With antithetic variance reduction;
  - With control variates;
  - With importance sampling.

- Run naive Monte Carlo simulations (no variance reduction) for asset prices from $S = £1$ to $S = £200$ in steps of $£1$, (i.e. in Matlab S = 1:200). From the variances of these simulations, determine a sample size such that the absolute accuracy of the Monte Carlo price is $£0.05$, with a 95% confidence level, over the full range of asset prices $1 \le S \le 200$. For that sample size, price the bull call spread for $S = £1$ to $S = £200$ and determine the CPU time needed for the computation.

  Note: the variance depends on the value of S. Do not choose different sample sizes for each S. Choose a single sample size that will keep the absolute error below $£0.05$ over the full range of S.

- Repeat what was done in the previous case for each of the three types of variance reduction.

- Using a method of your choice, implement an accurate computation of the delta for any one of the Monte Carlo methods that uses variance reduction.

2 Pricing a Barrier Option (50% of project credit)

2.1 Overview

A barrier option is a path-dependent option where the payoff depends on whether the underlying asset $S(t)$ hits or does not hit a specified barrier before expiry. Here we consider a down-and-out put option. This is put option that becomes worthless if the asset price falls below the barrier S_b at any time between the present and expiry T. Hence the payoff is:

$$\begin{cases} \max(K - S(T), 0), & \text{if } \min_t S(t) > S_b \\ 0, & \text{if } \min_t S(t) \leq S_b \end{cases} \tag{3}$$

where K is the strike price.

For the underlying process we will use the geometric Brownian motion

$$dS(t) = r(t)S(t)dt + \sigma(S(t), t)S(t)dW, \tag{4}$$

allowing for the possibility that the interest rate is time dependent and the volatility can depend on time and current value of underlying asset S. Below we refer to (4) as the local volatility model.

The aim of this part of the project is to price the down-and-out put option by Monte Carlo methods.

2.2 Particulars

Use the following parameters for pricing options:

1. The strike price is $K = £50$.
2. Time to maturity is 1 year.
3. The interest rate is given by the formula

   $$r(t) = r_0 \exp(r_1 t)$$

   where $r_0 = 0.05$ and $r_1 = 0.5$. Time t is in years.
4. The local volatility is given by the function

   $$\sigma(S, t) = \sigma_0(1 + \sigma_1 \cos(2\pi t))(1 + \sigma_2 \exp(-S/100)).$$

   where $\sigma_0 = 0.3$, $\sigma_1 = 0.12$ and $\sigma_2 = 0.6$. Time t is in years.
5. Assume there are 260 (working) days in a year.

2.3 Tasks

- Implement the local volatility model Eq. (4) into the naive Monte Carlo method for pricing a European put option. Use Euler time stepping with a time step of one day.
- Implement the following variance reduction methods into the Monte Carlo method for pricing a European put option with local volatility:
  - antithetic variance reduction;
  - control variates.
- Determine sample size such that the absolute accuracy of the Monte Carlo price is $£0.05$, with a 95% confidence level, over the range of asset prices $1 \leq S \leq 100$. Do this for each of the three methods implementing the local volatility model (naive method, antithetic variance reduction, control variates).

  For those sample sizes, price the put option for $S = £1$ to $S = £100$ and determine the CPU time needed for the computation.

  Note: while the variance depends on the value of S, do not choose different sample sizes for each S.
- Determine the additional computational costs of the local volatility model compared with constant volatility and interest rate. For this, use the naive Monte Carlo method for pricing a vanilla put option with constant volatility σ_0 and constant interest rate r_0. Determine CPU time needed to price the option for $S = £1$ to $S = £100$ with absolute accuracy of $£0.05$ at 95% confidence level. You can compare with the CPU for naive Monte Carlo simulations of the local volatility model.
- Implement Monte Carlo methods for pricing a down-and-out put option with local volatility:
  - without variance reduction (naive method);
  - antithetic variance reduction;
  - control variates.
- For $S_b = 30$, determine sample size such that the absolute accuracy of the Monte Carlo price is $£0.05$, with a 95% confidence level, over the range of asset prices $S_b < S \leq 100$. Determine the CPU time needed by each of the three methods.
- Consider a few representative values of $S_b < K$. For one of the Monte Carlo methods, compute the price of the down-and-out put option over a range of S. Also determine the price of European put option (equivalent to $S_b = 0$).

  Note: You do not need to re-determine the sample sizes for each value of S_b. Just use the sample size you determined in the previous item.
- Using a method of your choice, implement an accurate computation of the delta for the down-and-out put option in one of the Monte Carlo methods that uses variance reduction. Compute the deltas for the same values of S_b used in the previous item.

3 Project report

The project report should focus on explaining what you did and what results you found. You should not explain general background material taught in the Financial Maths courses. Be aware that clear and concise explanations and presentation of the data will be rewarded with higher marks. In particular,

- **Do Not** include numerous non-illustrative plots;
- **Do Not** explain the irrelevant background financial material taught in the course;
- **Do** included essential information necessary to verify your solutions.

As guidance, well-written reports are normally be between 12 and 16 pages. Reports with a few, well-chosen plots that clearly show the necessary information receive significantly higher marks. Data tables are preferable to graphs on some occasions.

It is suggested that you write in *passive voice* or used the *editorial we* (as in "We are therefore led also to ...").

Students are free to use Word, Latex or any other software they wish to typeset the report.

3.1 Contents

The report should contain the content listed below. You may include a small amount of other material if you feel it would benefit your report. There should be some brief introduction to the full report, or separately brief introductions to each part, but these can short statements of one or two paragraphs addressing what the project contains. Since all of the methods used have been covered in the module lectures, it is not necessary to include any citations in your report. However, if you use methods or information not covered in the module, then you need to cite the source.

Part I

- Give details of the transformation between the Black-Scholes PDE and the heat equation that you used. State specifically what initial condition and boundary conditions you used for the heat equation.
- Explain the need to use $S_{\max} = \pounds 500$ even though we only wanted to price the option up to $S_{\max} = \pounds 200$. You may refer to some Matlab simulations if you wish, but it is not required.
- Discuss your study of time steps N and grid points J and how you arrived at values to ensure that your solution is accurate to $\pounds 0.05$.
- Report the CPU time for obtaining the result and how it was obtained.
- Plot of your solution for the option price. (You may show on the same graph the value obtained from Black-Scholes formulae if you wish, but it is not required.) Discuss why the solution is sensible from a financial perspective.
- Plot of your solution for the delta. Explain and justify how the delta was computed. Discuss why the delta is sensible from a financial perspective.
- Explain, briefly, how you implemented the control variate and importance sampling methods for variance reduction. (It is not necessary that you explain Monte Carlo without variance reduction and Monte Carlo with antithetic variance reduction, but you may briefly if you wish.)

- State the sample sizes used for each of the four different Monte Carlo simulations to obtain the required accuracy. Explain how you determine the sample sizes. Report the CPU time for pricing the bull call spread to the to required accuracy, and state how the CPU time was obtained.
- Plot of your solutions for the option price from the Monte Carlo simulations. You should plot them on a single graph using different symbols. You may also show on the same graph the value obtained from Black-Scholes formulae if you wish, but it is not required.
- Plot of your solution for the delta from the Monte Carlo simulations. Explain what method you used for the computations and why.
- Finally, discuss and compare the above results. For the Monte Carlo approach, discuss any improved efficiency from variance reduction. Compare the PDE and Monte Carlo approaches for this particular problem. Specifically compare the CPU times needed to achieve the given accuracy. Discuss briefly any other pros and cons of the different approaches.

Part II

- Explain briefly how you implemented the local volatility model into the naive Monte Carlo methods for pricing a European put option. You can refer directly to your Matlab code. It might be simplest to include snippets of a key lines from your Matlab.
- Explain how you implemented the antithetic and control variates variance reduction. Again, you can refer directly to your Matlab code.
- For each of the three different Monte Carlo simulations, state the sample sizes used to obtain the required accuracy. Explain how you determine sample sizes. Report the CPU times for pricing the European put option to the to required accuracy, and state how the CPU time was obtained.
- Plot the option price as a function of S for each of the methods. You should plot them on a single graph using different symbols.
- Report and discuss the additional computational costs (CPU time) of simulating the local volatility model compared with Monte Carlo simulations in the case of constant volatility and interest rate. Explain the sources of the additional cost.
- Explain how you implemented the Monte Carlo methods for pricing the down-and-out put option. You do not need to repeat things already explained for the European put option. Focus on what is different for the down-and-out put option. You can refer directly to your Matlab code.
- For each of the three different Monte Carlo simulations, state the sample sizes used to obtain the required accuracy for the down-and-out option with $S_b = 30$. Report the CPU time for pricing the option. (Assuming that you used the same method as above, you do not need to state how the CPU time was obtained.) Discuss any changes in the performance of the variance reduction methods for the down-and-out put option compared with the European put option.
- For $S_b = 30$, plot the option price as a function of S for each of the methods. One graph using different symbols is sufficient.
- Plot option price as a function of S for different values of $S_b < K$. Include also the European put option. This should be a single graph. Discuss the meaning of this graph from a financial perspective.
- Plot of your solutions for the deltas. Explain and justify how the delta was computed. Discuss why the delta is sensible from a financial perspective.

3.2 Matlab

Your Matlab scripts and functions will be submitted along with the written report and must be reference within the written report as follows.

- There should be a section (or sections) in the written report describing what each submitted Matlab file is for and what Matlab commands should be entered in the Matlab command window to replicate the results presented in the report.
- All graphs and tables produced **must** be labelled and include a caption explaining what the graph/table shows, the parameters used to produce the data, and legend (if applicable) to provide clear understanding of what the graph represents. Moreover the caption should include the name of m-file containing a Matlab script that produced the graph. The marker of the project should be able to start Matlab, go to the student's code folder, run the script and get the corresponding graph or table. If the script just reads some data from a different data file, a comment should be given which m-file produces this data.

  It is permissible to have one script produce more than one graph/table.

  In the case of tables, the outputted data may include more information than is included in the report. Also tables in the report may be formatted differently from the output of the Matlab script, but it must be clear that the Matlab output corresponds to the data in the report.
- Graphs or tables produced not adhering to the rule above will receive no marks!!

3.3 Marks

Marks will be awarded for the project as follows.

- Matlab Codes: (40%): The codes should produce correct results; they should implement proper methods; they should be efficient; they should be readable with proper indentation and clear comments. It is not enough that your codes produce correct results. Marks will be deducted for confusing and unnecessarily complex programming.
- Written report: (60%): Correctness of methods and results reported; clarity, accuracy and quality of the presentation, especially the clarity and quality of graphs and tables, and their captions.

4 Project Submission

- The project must be submitted electronically through **mywbs** :
  - The written report should be in single PDF document in the root folder.
  - In the root folder create a sub-folder "Code", and in that sub-folder create two further sub-folders: "Part_I" and "Part_II". Put all of your Matlab scripts and functions from Part I and Part II of the project into these sub-folders.
  - It is the students' responsibility to ensure that if they submit a zip file, then it is not corrupt.

4.1 Rules and Regulations

This project is to be completed by individuals only and is not a group exercise. Plagiarism is taken extremely seriously and any student found to be guilty of plagiarism of fellow students will be severely punished.

4.1.1 Plagiarism

Please ensure that any work submitted by you for assessment has been correctly referenced as WBS expects all students to demonstrate the highest standards of academic integrity at all times and treats all cases of poor academic practice and suspected plagiarism very seriously. You can find information on these matters on my.wbs, in your student handbook and on the library pages here.

It is important to note that it is not permissible to reuse work which has already been submitted for credit either at WBS or at another institution (unless explicitly told that you can do so). This would be considered self-plagiarism and could result in significant mark reductions.

Upon submission of your assignment you will be asked to sign a plagiarism declaration.