QF4102 Financial Modelling and Computation: Assignment 3 Report

G04: Dong Shaocong, He Xinyi

November 19, 2017

Abstract

This document is the report for QF4102 Financial Modelling and Computation, assignment 3. In this report, we firstly examined a transformed black-scholes PDE model. We wrote and tested the functions for implementing the fully implicit schemes. Next we looked at the digital options and Monte Carlo methods to estimate their option value. In the end, we implemented the control variate version and tested the effects of it.

1 A 3.1 Transformed Black-Scholes PDE Model

Description of work done

In this question, we examined the transformed black-scholes PDE model given in the question. We firstly derived a fully $O[\Delta t + (\Delta x)^2]$ implicit scheme corresponding to the PDE. We represented this implicit scheme using both system of linear equations as well as its matrix form. Next, we implemented a MatLab function for this fully implicit scheme for European vanilla call option. Using different precision of grids, we plotted a graph of different estimates. In addition, we implemented an American vanilla call option's fully implicit scheme while at the same time also incorporating the **PSOR** algorithm. We then tested your implementation with an American vanilla call option with the same option and grid parameters. Below is the detailed documentation of the first question.

1.1 Fully implicit scheme for a European vanilla call option

In order to have a fully implicit scheme for European vanilla call option, we discretize the PDE at (x_j, t_n) where $t_n = n\Delta t$, $x_n^j = j\Delta x$ and replace the partial derivatives with the given finite difference approximations as follows:

$$\text{Forward time finite difference formula: } \frac{\partial u}{\partial t}|_{(x_n^j, t_n)} = \frac{u_{n+1}^j - u_n^j}{\Delta t} + O(\Delta t)$$

$$\text{Central space first derivative finite difference formula: } \frac{\partial u}{\partial x}|_{(x_n^j, t_n)} = \frac{u_n^{j+1} - u_n^{j-1}}{2\Delta x} + O(\Delta x^2)$$

$$\text{Central space second derivative finite difference formula: } \frac{\partial^2 u}{\partial x^2}|_{(x_n^j, t_n)} = \frac{u_n^{j+1} - 2u_n^j + u_n^{j-1}}{\Delta x^2} + O(\Delta x^2)$$

With the following finite difference approximation

$$\frac{u_{n+1}^j - u_n^j}{\Delta t} + \frac{\sigma^2}{2}\frac{u_n^{j-1} - 2u_n^j + u_n^{j+1}}{\Delta x^2} + (r - q - \frac{\sigma^2}{2})\frac{u_n^{j+1} - u_n^j}{\Delta x} - ru_n^j + O(\Delta t) + O(\Delta x^2) = 0$$

We drop the truncation error terms and re-arrange with the unknowns on the left hand side gives the fully implicit finite difference equation below

$$U_{n+1}^j = aU_n^{j-1} + bU_n^j + cU_n^{j+1}$$

$$\text{where } \alpha = \frac{\sigma^2 \Delta t}{(\Delta x)^2} \text{ and } \beta = \frac{\Delta t(r - q - \frac{\sigma^2}{2})}{2\Delta x}$$

$$a = \beta - \frac{\alpha}{2},\ b = 1 + \alpha + r\Delta t,\ c = -\frac{\alpha}{2} - \beta$$

$$x \in (-\infty, \infty) \text{ is truncated to } x \in [-x_{max}, x_{max}] \text{ where } x_{max} = Idx.$$

$$j \in [-x_{max}, x_{max}],\ n \in [0, N] \text{ where } N = \frac{T}{\Delta t}$$

We can write it in matrix form: $AU_n = U_{n+1} + F_n$, where

$$\text{A} = \begin{bmatrix} b_{-I+1} & c_{-I+1} & & & \\ a_{-I+2} & b_{-I+2} & c_{-I+2} & & \\ & \ddots & \ddots & \ddots & \\ & & a_{I-2} & b_{I-2} & c_{I-2} \\ & & & a_{I-1} & b_{I-1} \end{bmatrix},\ U_n = \begin{bmatrix} U_n^{-I+1} \\ U_n^{-I+2} \\ \vdots \\ U_n^{I-2} \\ U_n^{I-1} \end{bmatrix}$$

$$F_n = \begin{bmatrix} -a_{-I}U_n^{-I} \\ 0 \\ \vdots \\ 0 \\ -c_I U_n^I \end{bmatrix}$$

The boundary conditions are given as follows:

$$U_n^{-I} = 0,\ U_n^I = e^{x_{max} - q(T - n\Delta t)} - Xe^{-r(T - n\Delta t)} \text{ for } n \in [0, N]$$

The terminal conditions are given as:

$$U_N^i = (e^{i\Delta x} - X)^+, i \in [-I, I]$$

1.2 European Call: Matlab function fully implicit scheme

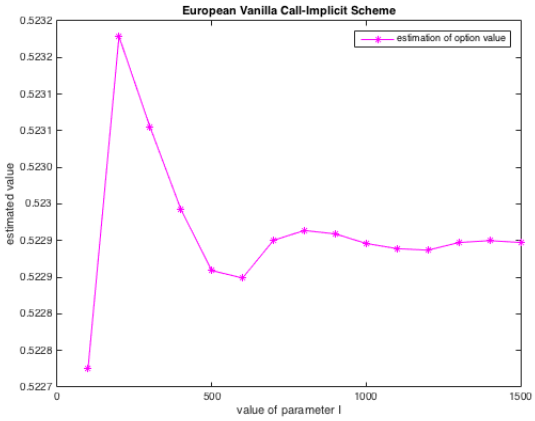

Figure 1: European Call Estimated Value using fully implicit scheme against parameter I

The option value estimates oscillate with increasing value of parameter I, and it converges quickly to a stable value of around 0.5229 which is very closed to the actual value calculated by closed form formula. As grid size decreases, the accuracy of the estimation increases. As the grid size decreases to smaller than $\frac{5}{1000} = 0.005$, the change in value becomes very small thus becoming insignificant. With the increasing I, the accuracy of the estimate increases. This is true from the error term $O[\Delta t + (\Delta x)^2]$ and as I increases, Δx becomes smaller, thus the error becomes smaller. Therefore, we can estimate this European vanilla call option to be the estimated value we obtained in the final step with $I = 1500$. From the lecture on the necessary condition of convergence (monotonicity condition), the following must be satisfied for the matrix A:

$$a_{ij} \leqslant 0 \text{ if } i \neq j\ ,\ a_{ii} > \sum_{i \neq j} |a_{ij}|$$

From our calculation, the above conditions are always satisfied for the different values of I we used in the calculation, thus we can be sure this fully implicit scheme is convergent.

1.3 American Call: Matlab function fully implicit scheme with PSOR

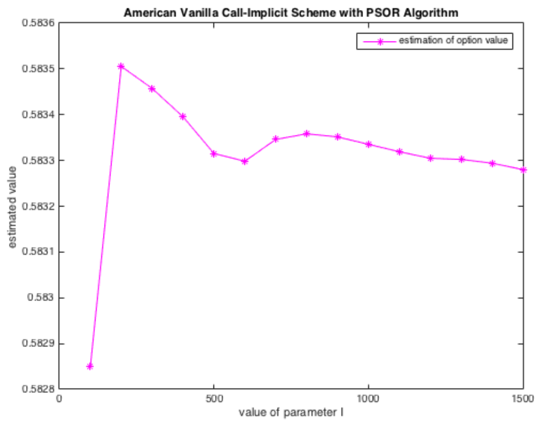

Figure 2: American Call Estimated Value with projected SOR iterative methods against parameter I

As we can see from the plot, the change of option value estimation exhibits the same pattern as the European one. As the grid size becomes smaller, the option value's oscillation becomes smaller, and the difference between each estimate becomes smaller. Given the same set of parameters, American option has an estimation of 0.5833, which is higher than the European option's 0.5229.

2 A 3.2 Digital Option with Monte Carlo Methods

Description of work done

This question is about Monte Carlo method and digital option. To solve this problem, we firstly implement a function for a digital option with return of value of 1 if $S_T > X$. Next, we considered a 3-asset digital call option with given parameters and defined correlation coefficients using Monte-Carlo estimation. We also tried with different strike price and price-path bundles. In the end, we modified the function to incorporate a control variate (a basket of three single-asset digital calls) for the purpose of variance reduction. The results are tabulated and shown below.

2.1 Without Control Variate: Option value estimates and standard variations

strike Price/Price-path Bundle	100	1000	10000	100000
8.5	0.88774	0.87837	0.87812	0.87803
9.5	0.72111	0.7126	0.71407	0.71404
10.5	0.49958	0.50552	0.50455	0.50369

Table 1: Option Value estimates without control variate

From the option values, we can see for different strike price, the estimates of the option price will be different. The change in option value estimates and the number of price-path bundle is not monotonous. As the number of price-path bundle becomes bigger, the change of option value estimates become smaller.

strike Price/Price-path Bundle	100	1000	10000	100000
8.5	0.027135	0.0096018	0.0024944	0.00085757
9.5	0.047976	0.010657	0.0047262	0.0013899
10.5	0.05394	0.012538	0.0046253	0.0013663

Table 2: Option Value standard deviations without control variate

From this table about the option value standard deviation, As the number of price-path bundle becomes bigger, the order of magnitude of standard deviation changes from 0.01 to 0.0001 or even 0.00001. As the price-path bundle becomes bigger, the option value estimates thus become more precise and stable.

In addition, as shown in lecture, standard deviation is proportional to the reciprocal of square root of p. We can see from this table that increasing the price-path bundles 100 times decreases the standard deviation for about 10 times smaller. It's thus expensive to use more price-path bundles to control the standard error, and various reduction methods are needed to achieve accurate result.

2.2 With Control Variate: Option value estimates and standard variations

strike Price/Price-path Bundle	100	1000	10000	100000
8.5	0.87194	0.87914	0.87814	0.87801
9.5	0.71452	0.71244	0.71495	0.71434
10.5	0.5059	0.50407	0.50464	0.50418

Table 3: Option Value estimates with control variate

The option value estimates have a similar trend and very close values as the Monte Carlo estimation without control variate. With more and more price-path bundles, we can expect a more accurate estimation and we can see there is a trend of convergence as well for all three strike price cases.

strike Price/Price-path Bundle	100	1000	10000	100000
8.5	0.021339	0.0074483	0.0019174	0.00067129
9.5	0.019028	0.0082898	0.0025408	0.00082191
10.5	0.021273	0.0060285	0.0025041	0.00087121

Table 4: Option Value standard deviations with control variate

From this tabulation of results, we can see the similar trend for standard deviation again, as the price-path bundles increase, the standard deviation becomes increasingly smaller. We can also notice smaller standard errors as compared to the Monte Carlo simulation without control variate.

A clearer comparison is given in the plot below.

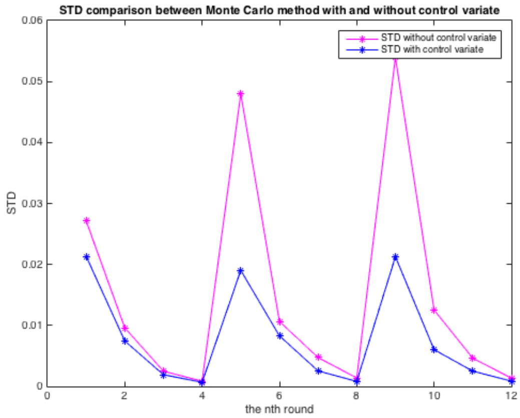

Figure 3: STD Comparison between Monte Carlo Method with and without control variate

From this plot, we can see using control variate significantly improves the standard error. For the Monte Carlo simulation without control variate, if the number of price-path bundles is kept constant, the standard deviation increases with increasing strike price. However, for Monte Carlo simulation with control variate, we can have similar standard error for different strike price keeping number of price-path bundle as constant. This demonstrates the control variate works effectively in controlling the variation for options with different strike price.