

Numerical Schemes and Monte Carlo Method for Black and Scholes Partial Differential Equation: A Comparative Note

Sharif Mozumder*, ABM Shahadat Hossain, Sadia Tasnim, Arafatur Rahman

Department of Mathematics, University of Dhaka, Bangladesh

Copyright © 2015 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract This paper comparatively investigates some iterative methods and Monte Carlo simulation technique for the dynamics underlying the celebrated Black and Scholes (BS) model. In particular we attempt to answer the question: ‘How many Monte Carlo replications can yield prices, for plain vanilla type European derivatives on a stock, which are similar to those obtained by solving the BS PDE using iterative numerical schemes?’ We confine to three frequently referred iterative schemes such as Successive over Relaxation (SOR), Gauss-Seidel (GS) and Jacobi (JC). This information together with the information of ‘differences in time requirements’ will help to guess the similar trade-offs for complex derivatives(exotic) pricing for which there are no analytic pricing formulas.

Keywords Black and Scholes PDE, Iterative Solutions, Monte Carlo Simulation, Option Pricing

1. Introduction

Modern option pricing techniques are often considered among the most mathematically complex area among all applied areas of finance. A vanilla option (normal call or put option) is a financial instrument that gives the holder the right, but not the obligation, to buy or sell an underlying asset at a predetermined price, within a given time frame and it is generally traded on an exchange such as the Chicago Board Options Exchange. In this paper we deal with two approaches in derivative pricing under Black and Schole’s world. One approach is preferred by people in computational mathematics area and another approach falls under the concern of people from stochastic finance area. The first approach concerns iterative methods to solve the Black-Scholes (BS) type partial differential equation in order to price different financial products traded on some underlying, which is often an index or individual stock. The second approach concerns the so called Monte Carlo (MC)

simulation to pricing various financial products. This approach replicates random phenomena which are hypothesized to capture the reality. While the numerical methods are algorithmic and can be applied to most of the PDE’s, the Monte Carlo methods are time consuming and depends upon the number of replications considered in order to price some particular financial product under consideration. We shed some light on the trade-offs between the approaches for basic vanilla type derivative pricing.

We revisit different iterative methods used in numerically solving BS PDE in order to obtain the fair price of the vanilla options. We consider Jacobi (JC), Gauss-Seidel (GS) and Successive-Over-Relaxation (SOR) algorithms with backward substitution to price plain vanilla put options. The iterative methods are studied in great details by Burden[3], Isaacson[7], Atkinson [1] ; and their stochastic versions are explored by Mao [9], Kloeden [10].

We observe that except some differences in time requirement, the prices obtained are very similar. We revisit the application of Monte Carlo (MC) simulation technique to generate asset price paths governed by BS stochastic differential equations (SDE). We admit that MC simulation and its applications in finance and insurance is a huge area. We just get some flavor by delving as little as is required to simulate paths for BS SDE. We price options using these simulated paths and provide reasonable evidence to answer the fundamental question ‘how many paths in MC pricing can provide option prices similar to those obtained by iteratively solving BS PDE under different numerical scheme?’ We discuss the trade-offs associated with both the approaches. While we try different number of replications under MC method we overall observe that except little more time requirement MC method is much more powerful than the iterative numerical schemes. Other useful applications of Monte Carlo methods in Finance are well documented in the literature; see e.g. Jackel [8], Dowd[4] , Glasserman [11] .We figure out about number of replications which can price a particular type of option ensuring that prices are

always better than those obtained by iterative schemes. Comparisons for illustrative options are presented through tables showing the trade-offs between the approaches.

The paper is structured as follows: section 2 briefly revisits the BS PDE and BS pricing. Section 3 is about the option pricing under MC simulation. We briefly discuss the iterative methods under consideration in section 4. Section 5 shows how one can price options using the iterative methods. We conduct numerical implementation and discuss our findings in section 6. Section 7 concludes the paper.

2. BS PDE

The Black–Scholes equation is a partial differential equation which describes the evolution of an option price over time. The key idea behind the equation is that one can perfectly hedge the option by buying and selling the underlying asset and consequently “eliminate risk”. This hedge, in turn, implies that there is only one right price for the option, as returned by the Black–Scholes formula. We consider the BS PDE formula for analytic pricing as we will compare the analytic prices both with numerical prices(coming from solving the BS PDE using iterative methods) as well as prices obtained by applying MC simulation to BS dynamics.

For European style option with pay-off ‘V’ the celebrated BS equation is (see Black and Schole[2]) :

$$\frac{\delta V}{\delta t} + \frac{1}{2} \sigma^2 S^2 \frac{\delta^2 V}{\delta S^2} + rS \frac{\delta V}{\delta S} - rV = 0.$$

The values of call option, $C(S, t)$ and put option $P(S, t)$ for a non-dividend paying underlying stock in terms of the Black–Scholes paradigm are:

$$\left. \begin{aligned} C(S, t) &= N(d_1)S - N(d_2)Ke^{-r(T-t)} && , for\ Call\ option \\ P(S, t) &= N(-d_2)Ke^{-r(T-t)} - N(-d_1)S && , for\ Put\ option \end{aligned} \right\} \quad (1)$$

Here the price put option is based on put-call parity.
In(1):

$$\left. \begin{aligned} d_1 &= \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} \\ d_2 &= \frac{\ln\left(\frac{S}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} \end{aligned} \right\} = d_1 - \sigma\sqrt{T-t} \quad (2)$$

For both the option pricing formulas as in (1), $N(\cdot)$ is the cumulative distribution function of the standard normal distribution, $(T - t)$ is the time to maturity, S is the stock price of the underlying asset, K is the strike price, σ is the volatility of returns of the underlying asset and ‘r’ is the annual risk free rate.

3. Option Pricing Using Monte Carlo Simulation

Monte Carlo Simulation is used in finance to value and analyze instruments, portfolios and investments by simulating the various sources of uncertainty which affect their values, and then determining their average values over the range of resultant outcomes. This is usually done by stochastically simulating the price patterns of the underlying on which the derivatives are traded. For option pricing, the technique involved the following operations, see Hull [6], Higham [5], Jackel[8]:

(1) To generate several thousand possible (but random) price paths for the underlying via simulation.

(2) To calculate the associated exercise value (i.e. "payoff") of the option for each path.

(3) To average these pay offs. This result is the value of the option.

(4) To discount this value to the present value.

Usually a large number of asset path are generated that evolve according to the equation:

$$dS = \sigma S dX + rS dt \quad (3)$$

Where S is the asset value, dS and dt are incremental changes in asset value and time respectively, σ is the volatility, r is the rate of return and dX is a Wiener process. Because of the dX term, S itself is also a random variable. Fig 1 shows the generated random paths for the Monte Carlo simulation.

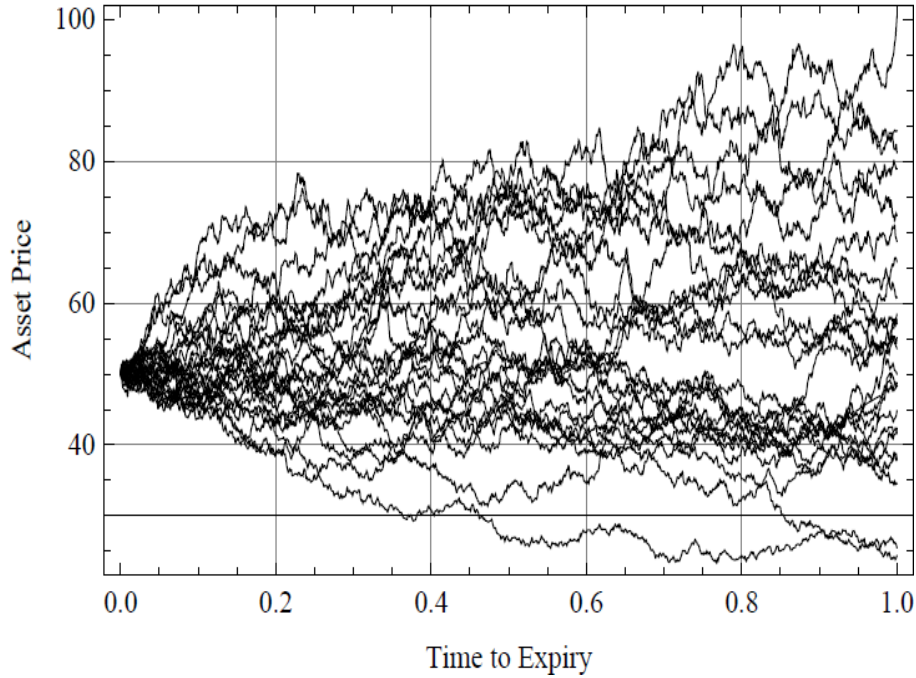


Figure 1. Multiple Monte Carlo generated price paths

4. Iterative Methods

An iterative method is a mathematical procedure that generates a sequence of improved approximation to the solutions for a class of problems. In the process of finding a solution of a system of equations, an iterative method uses an initial guess to generate successive approximations to a solution. We show application of three frequently referred iterative techniques namely: Jacobi (JC), Gauss-Seidel (GS) and Successive-Over-Relaxation (SOR) which produce the numerical solutions for BS PDE. In general we solve a system $\underline{x} = \underline{b}$; A being a matrix and \underline{x} and \underline{b} are column vectors. See Atkinson [1], Isaacson[7].

Jacobi (JC) method is the simplest of all three iterative methods we consider. It uses the k^{th} value of the approximation to determine the $(k + 1)^{th}$ approximation. The GS(or the method of successive displacement) is an iterative method, used to solve a linear system of equations, which is simple modification of the Jacobi method. Though it can be applied to any matrix with non-zero elements on the diagonals, convergence is only guaranteed if the matrix is either diagonally dominant or symmetric and positive definite. GS iterative method uses the recent approximations, instead of k^{th} value, to determine the $(k + 1)^{th}$ value. See Atkinson (1978), Isaacson (1994). The last iterative method of our consideration SOR is a variant of the GS method for solving a linear system of equations. SOR iterative method is the modification of GS iterative method. SOR iterative method uses the $(k + 1)^{th}$ approximation with a relaxation parameter ω as a modification of GS method. Among the three iterative methods SOR gives faster result, as it converges faster than other two iterative methods. An algorithmic view of three iterative methods looks like as below:

for $k = 1, 2, \dots, n$ do

for $i = 1, 2, \dots, n$ do

Jacobi method:

$$z_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^k - \sum_{j=i+1}^n a_{ij} x_j^k \right)$$

$$x_i^{k+1} = x_i^k + z_i^{k+1}$$

Gauss-Seidel method:

$$z_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^n a_{ij} x_j^k \right)$$

$$x_i^{k+1} = x_i^k + z_i^{k+1}$$

SOR method:

$$z_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^n a_{ij} x_j^k \right)$$

$$x_i^{k+1} = \omega z_i^{k+1} + (1 - \omega) x_i^k$$

end i;

end k;

5. Pricing Options Using Iterative Methods

We use an implicit Euler characterization of finite difference method for our illustration. For our bivariate, parabolic PDE, we start by establishing a rectangular solution domain in the two variables, S and t ; and then apply finite difference approximations to each of the derivative terms in the PDE. Consider the domain discretisation in S and t :

Asset Value

Discretisation: $\{0, \delta S, 2\delta S, 3\delta S, \dots, M\delta S\}$ where $M\delta S = S_{max}$

Time Discretisation: $\{0, \delta t, 2\delta t, 3\delta t, \dots, M\delta t\}$ where $N\delta t = T$ and $V_{i,j} = V(i\delta S, j\delta t) \quad i = 0, 1, \dots, M \quad j = 0, 1, \dots, N$

S_{max} is a maximum value for S that we must choose sufficiently large (of course, a discretization cannot allow S to assume infinity!). Following boundary conditions are for European Put options:

$$V(S, dT) = \max(K - S, 0) \quad V(0, t) = Ke^{-r(T-t)} \quad V(S_{max}, t) = 0 \dots \dots \dots \quad (4)$$

In mesh-grid notations:

$$V_{i,N} = \max(K - (i\delta S), 0) \quad i = 0, 1, \dots, M$$

$$V_{0,j} = Ke^{-r(N-j)\delta t} \quad j = 0, 1, \dots, N$$

$$V_{M,j} = 0 \quad j = 0, 1, \dots, N$$

For a given payoff at expiry, our problem is to solve the Black-Scholes PDE backwards in time from expiry to the present time ($t = 0$). If all three derivative terms are approximated with Implicit Euler finite difference the discretisation is given by:

$$\frac{V_{i,j} - V_{i,j-1}}{\delta t} + \frac{1}{2}(i\delta S)^2 \frac{V_{i+1,j-1} - 2V_{i,j-1} + V_{i-1,j-1}}{\delta S^2} + r(i\delta S) \frac{V_{i+1,j-1} - V_{i-1,j-1}}{\delta S} - rV_{i,j-1} = 0 \quad (5)$$

This uses a backward difference to approximate the time derivative (since we are solving backward in time from expiry) and a central difference to approximate the first derivative in asset value. The second derivative term is approximated by the usual second difference.

Upon simplification, we can re-write (5) as:

$$V_{i,j} = A_i V_{i-1,j-1} + B_i V_{i,j-1} + C_i V_{i+1,j-1};$$

where

$$A_i = \frac{1}{2} \delta t (ri - \sigma^2 i^2); \quad B_i = 1 + (\sigma^2 i^2 + r) \delta t; \quad C_i = -\frac{1}{2} \delta t (ri + \sigma^2 i^2)$$

We can now apply iterative methods using these formulas to price options.

6. Numerical Implementation

We produce the results for put options and one can have the similar analysis for call options using put-call parity. We replicate various numbers of paths to price each option and found that MC method is more powerful than numerical schemes even with relatively less number of replication to price each option. We compare the Monte Carlo simulation with the three iterative methods using the illustrative parameter values as value of the underlying, $S = 5$, the volatility $\sigma = 0.3^1$, risk free interest rate $r = .06$, time to expiry $T=1$, number of time and asset mesh points $N=500$. Of course in particular context these parameter sets could be quite different and some parameters might have some effect on numerical schemes and/or MC method as well. However we ignored that issue in this research. We compare each of the numerical schemes with different number of replications in MC pricing. We then choose 2^4 and 2^{10} replications to draw the graphs showing the comparison of iterative methods with MC method. Figure 2 shows the trade-offs between number of replications and corresponding price differences for each of the numerical schemes and MC method.

Table1. Price comparison of Monte Carlo simulation and iterative numerical methods.

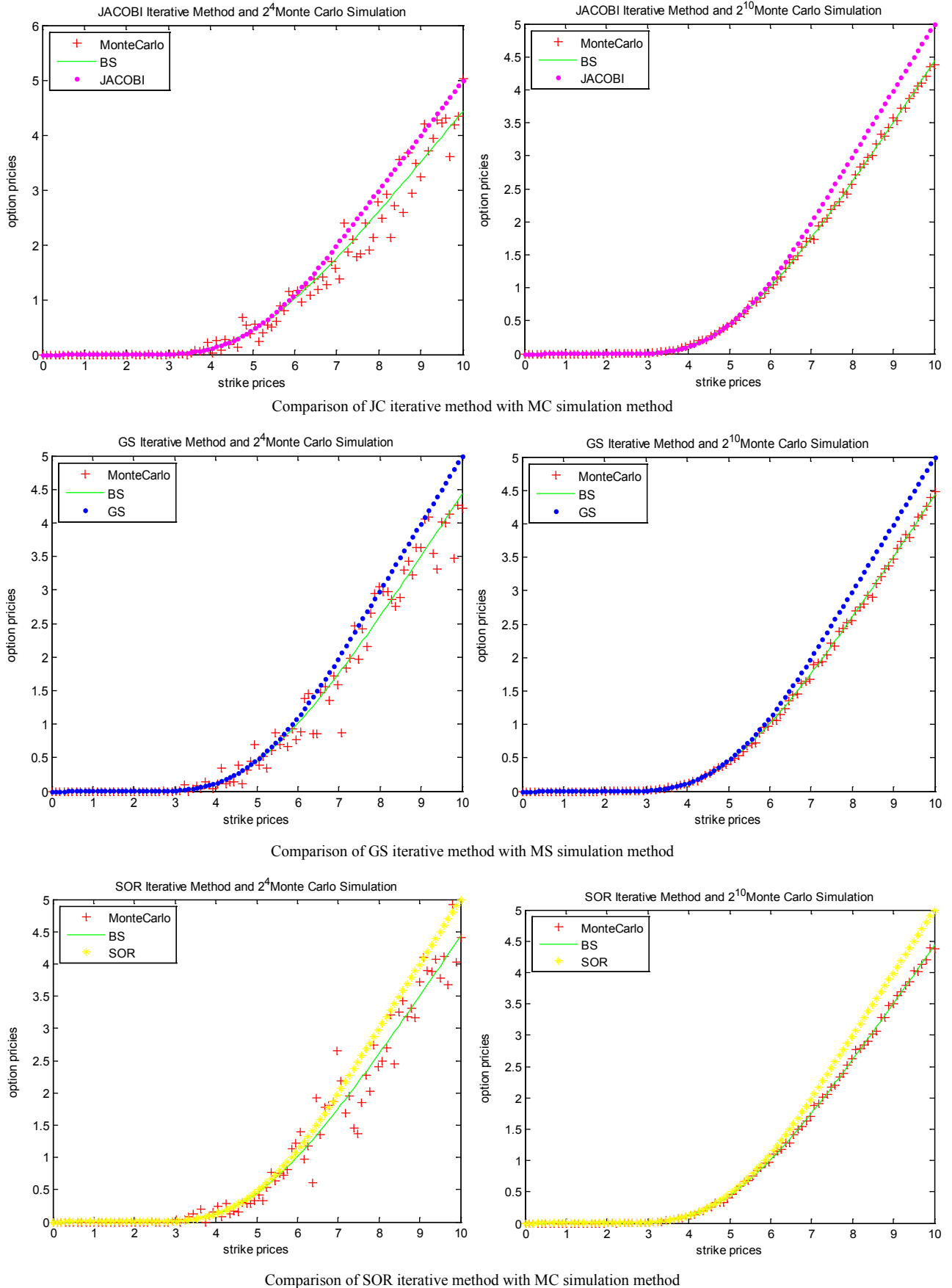
No of Monte Carlo Paths	Prices			
	Jacobi	Gauss Seidel	SOR	Monte Carlo
2^4	0.4526	0.4663	0.4723	0.7620
2^{10}				0.4514

Table2. Time comparison of Monte Carlo simulation and iterative numerical methods.

No of Monte Carlo Paths	Elapsed Time(sec) for prices			
	Jacobi	Gauss Seidel	SOR	Monte Carlo
2^4	16.4629	13.3019	12.1938	3.1839
2^{10}				143.2897

From the results in table 1 we can say that there is hardly any difference in SOR, GS & JC iterative methods for European option pricing. We get little bit better prices for SOR method compared to GS and JC methods; however the difference is really negligible. Nonetheless little preference can go with SOR among the numerical methods. Our main concern is how many paths can ensure that MC prices are in the close proximity of numerical prices. We found that for number of paths gradually increasing from 2^4 to 2^{10} the MC prices become gradually closer to numerical prices. It is not impossible to get MC prices with as little as 2^4 replications which are well comparable with the prices obtained by with any of the numerical schemes. However, more importantly we observe that for number of replications being near 2^{10} it is almost always the case that

¹ For very small σ values there is concern on obtaining the convergence in numerical schemes; see e.g. [12], [13].



Comparison of JC iterative method with MC simulation method

Comparison of GS iterative method with MS simulation method

Comparison of SOR iterative method with MC simulation method

Figure 2. Comparison of iterative and MC methods.

MC prices are in very close proximity to the prices coming from any of the numerical schemes. Furthermore comparing with analytic BS prices it is clear that with more number of replications in MC prices can be made increasingly closer to those obtained with analytic BS formula, however only at the expense of additional computational time. With respect to equally comparable pricing, however, numerical iterative prices are much faster than MC prices; as is evident from table 1 and table 2. Though only reported for an illustrative option type these observations are found true for other maturity options as well as other types (e.g. European Call).

7. Conclusions

In this paper we studied iterative methods and the Monte Carlo simulation method for the dynamics underlying the celebrated Black and Scholes (BS) model in order to see the trade-offs between the approaches. We tried various numbers of replications under MC method in order to see how simulation based pricing fares with iterative pricing. Our observation is that if one is comfortable with little bit more time requirementⁱ then MC prices are much better than the iterative prices. In particular we observe that number of replications around 2^{10} can often ensure that MC prices are better than iterative prices; though one can obtain comparative prices with far less number of replications. We observe these for both European Put option and Call option. In a future work we plan to investigate other derivatives and the effects of other determinates in such trade-offs.

REFERENCES

ⁱ If we consider 'adaptive method' instead of 'constant co-efficient' method similar accuracy can be obtained with low computational cost; see [12], [13].

- [1] Atkinson, A.K.(1978). An Introduction to Numerical Analysis. Wiley.
- [2] Black, F., Scholes, M.(1973) *The pricing of options and corporate liabilities*, Journal of Political Economy, 81 , pp. 637-659.
- [3] Burden, R.L., Faires, J.D.(2010). *Numerical Analysis*, Cengage Learning.
- [4] Dowd, K.(2005). Measuring Market Risk. Wiley.
- [5] Higham, D.J.(2004) *Black-Scholes Option Valuation for Scientific Computing Students*, Department of Mathematics, University of Strathclyde, Glasgow, Scotland, January 2004.
- [6] Hull, J.C.(2009) *Options, Futures, & Other Derivatives*, Prentice Hall, New Jersey, 6th ed..
- [7] Isaacson, E. , Keller, H.B.(1994). *Analysis of Numerical Methods*. Dover.
- [8] Jackel, P.(2002). *Monte Carlo Methods in Finance*. Wiley Finance.
- [9] Mao,X (1997). *Stochastic Differential Equation and their Applications*, Harwood.
- [10] Kloeden, P.,E.(1992). *Numerical Solution of Stochastic Differential Equations*. Springer.
- [11] Paul, Glasserman.(2003). *Monte Carlo Methods in Financial Engineering*, Springer.
- [12] P. Das and V. Mehrmann (2015) *Numerical solution of singularly perturbed convection-diffusion- reaction problems with two small parameters*, BIT Numerical Mathematics, pp 1-26.
- [13] P. Das(2015) *Comparison of a priori and a posteriori meshes for singularly perturbed nonlinear parameterized problems*, Journal of Computational and Applied Mathematics, 290, pp 16-25.