SUPER-TIME-STEPPING SCHEME FOR OPTION PRICING

Kedar Nath Uprety [*], **Harihar Khanal** [**] and **Ananta Upreti** [*]

[*]Central Department of Mathematics, Tribhuvan University, Kirtipur, Kathmandu, Nepal

[**]Department of Mathematics, Embry-Riddle Aeronautical University, USA

Abstract: We solve the Black - Scholes equation for option pricing numerically using an Explicit finite difference method. To overcome the stability restriction of the explicit scheme for parabolic partial differential equations in the time step size Courant-Friedrichs-Lewy (CFL) condition, we employ a Super Time Stepping (STS) strategy based on modified Chebyshev polynomial. The numerical results show that the STS scheme boasts of large efficiency gains compared to the standard explicit Euler method.

Keywords: Black Scholes Equation; Explicit Scheme; Super Time Stepping Scheme; Modified Chebyshev Polynomial.

INTRODUCTION

Options have been considered to be the most dynamic segments of the security markets since the inception of the Chicago Board Options Exchange (CBOE) in April 1973. With one million contracts per day, CBOE is the largest option exchange in the world. After that, several other options exchanges described in [4].

The Black Scholes Model is one of the most important concepts in modern financial theory. It was developed in 1973 by Fisher Black, Robert Merton and Myron Scholes[2] and is still widely used till today. It is regarded as one of the best ways of determining fair prices of options based on six variables such as volatility, type of option, underlying stock price, time, strike price, and risk-free rate [2].

Explicit methods are simple and accurate to implement, and are convenient for parallelization but suffer severely with stability restriction on the time step size. To overcome the stability restriction of the explicit method, we apply first - order Super Time Stepping (STS) scheme based on modified Chebyshev polynomial[1]. STS is a technique which can be used to accelerate explicit scheme for parabolic problems.

Our main objectives in this paper are: (a) to solve the Black- Scholes Equation for option pricing numerically using an Explicit finite difference method, and (b) to overcome the stability restriction of the Explicit scheme in order to meet these objectives, we employ a STS strategy based on modified Chebyshev polynomial.

The Model Equation

The Linear Black-Scholes Equation (LBS)[2], as developed by Fisher Black and Myron Scholes in 1973 is,

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \qquad (1)$$

where, $V = V(S,t)$, the pay of function, $S = S(t)$, the stock price, t = time with $t \in (0,T]$, T is the time of maturity, $r > 0$ is a constant riskless interest rate, σ is constant volatility for the asset. To solve the equation (1), we need to state the initial and boundary conditions for both the European Call and Put option.

European Call Option

The solution to the Black-Scholes Equation (1) is the value $V(S,t)$ of the European Call option[2] on $0 \leq S < \infty$; $0 \leq t \leq T$. The boundary and terminal conditions are:

$V(0,t) = 0$ for $0 \leq t \leq T$,

$V(S,t) = S - Ee^{-r(T-t)}$ as $S \to \infty$, $\qquad (2)$

$V(S,t) = (S - E)^+$.

European Put Option

The terminal and boundary conditions for European Put option are:

$$V(0,t) = Ee^{-r(T-t)} \text{ for } 0 \leq t \leq T,$$
$$V(S,t) = 0 \text{ as } S \to \infty, \qquad (3)$$
$$V(S,t) = (E - S)^+.$$

It is difficult to find the analytic solutions to (1) for European option. Therefore, we need to use a numerical approximation. To find an approximation of the option value, one can compute a solution to the Black-Scholes Equation (1) using a finite difference method.

Finite Difference Method

The finite difference methods[3] attempt to solve Black Scholes partial differential equation by approximating the differential equation over the area of integration by a system of algebraic equations. The most common finite difference methods for solving the Black Scholes partial differential equation are the Explicit method, the fully Implicit method and the Crank-Nicolson method. These are closely related but differ in stability, accuracy and execution speed[6]. In this paper, we use Explicit method to solve the equation (1).

Transformation

The Black Scholes Equation (1) is of backward type, so considering the Black Scholes Equation and applying the change of variables,

$$\tau = T - t$$

to transform the parabolic backward equation into a forward equation[12]. The equation becomes,

---
Author for Correspondence : Ananta Upreti, Tri-Chandra College, Department of Mathematics, Ghantaghar, Kathmandu, Nepal.
Email: upretimath@yahoo.com

$$\frac{\partial V}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rS\frac{\partial V}{\partial S} + rV = 0 \qquad (4)$$

The Explicit Method

We use a forward difference approximation for the time derivative, a central difference approximation for the first order S derivative and a symmetric central difference approximation of the second order S derivative

$$\frac{\partial V}{\partial t}(n\Delta s, m\Delta t) = \frac{V_n^{m+1} - V_n^m}{\Delta t} + O(\Delta t).$$

$$\frac{\partial V}{\partial s}(n\Delta s, m\Delta t) = \frac{V_{n+1}^m - V_{n-1}^m}{2\Delta s} + O((\Delta s)^2).$$

$$\frac{\partial^2 V}{\partial s^2}(n\Delta s, m\Delta t) = \frac{V_{n+1}^m - 2V_n^m + V_{n-1}^m}{\Delta s^2} + O((\Delta s)^2).$$

We get the following discretized form of equation (4)

$$\frac{V_n^{m+1} - V_n^m}{\Delta t}$$
$$-\frac{1}{2}\sigma^2 n^2 (\Delta s^2)\frac{V_{n+1}^m - 2V_n^m + V_{n-1}^m}{\Delta s^2}$$
$$-rn\Delta s\frac{V_{n+1}^m - V_{n-1}^m}{2\Delta s} + rnV_n^m = 0 \qquad (5)$$

for $n = 1,2,...,N-1$ and $m = 0,1,2,...,M-1$.

In this case , the error is of order $O(\Delta t + (\Delta s^2))$.

From (5), we get

$$V_n^{m+1} = \frac{1}{2}(\sigma^2 n^2 - rn)\Delta t V_{n-1}^m$$
$$+(1 - (r + \sigma^2 n^2)\Delta t V_n^m \qquad (6)$$
$$+\frac{1}{2}(rn + \sigma^2 n^2)\Delta t V_{n+1}^m$$

for $n = 1,2,...,N-1$ and $m = 0,1,2,...,M-1$.

We denote

$$\alpha = \sigma^2 \Delta t$$
$$\beta = r\Delta t.$$

We get

$$V_n^{m+1} = \frac{1}{2}(\alpha n^2 - \beta n)V_{n-1}^m$$
$$+(1 - (\alpha n^2 + \beta))V_n^m \qquad (7)$$
$$+\frac{1}{2}(\alpha n^2 + \beta n)V_{n+1}^m$$

for $n = 1,2,...,N-1$ and $m = 0,1,2,...,M-1$.

Again, we denote

$$a_n = (1 - \alpha n^2 - \beta), n = 1,2,...,N-1$$

$$b_n = \frac{1}{2}(\alpha(n-1)^2 + \beta(n-1)), n = 2,3,...,N$$

$$c_n = \frac{1}{2}(\alpha(n+1)^2 - \beta(n+1)), n = 0,1,2,...,N-2.$$

$$\therefore V_n^{m+1} = c_n V_{n-1}^m + a_n V_n^m + b_n V_{n+1}^m \qquad (8)$$

the equation (8) holds for $n = 1,2,...,N-1$. Since V_{-1}^m and V_{N+1}^m are not defined, so there are $N-1$ equations for $N+1$ unknowns. The remaining two equations come from the two boundary conditions on $n = 0$ and $n = N$.

For call option

$$V_0^m = 0 \text{ for all } m$$

$$VNm = N\Delta s - Ee^{-rm\Delta t}$$

For put option

$$V_0^m = Ee^{-rm\Delta t} \text{for all } m$$

$$V_N^m = 0$$

The equation (8) can be written as in the matrix equation

$$V^{m+1} = AV^m + X^m \qquad (9)$$

where

$$V^{m+1} = \begin{bmatrix} V_1^{m+1} \\ V_2^{m+1} \\ \vdots \\ V_{N-1}^{m+1} \end{bmatrix},$$

$$V^m = \begin{bmatrix} V_1^m \\ V_2^m \\ \vdots \\ V_{N-1}^m \end{bmatrix},$$

$$X^m = \begin{bmatrix} c_0 V_0^m \\ 0 \\ \vdots \\ 0 \\ b_0 V_N^m \end{bmatrix}$$

$$m = 0, 1, 2, \ldots, M-1 \text{ and}$$

$$A = \begin{bmatrix} a_1 & b_2 & 0 & \cdots & & 0 \\ c_1 & a_2 & b_3 & & & \vdots \\ 0 & \ddots & & & & \\ & & & \ddots & & 0 \\ \vdots & & \ddots & \ddots & & b_{N-1} \\ 0 & & \cdots & & c_{N-2} & a_{N-1} \end{bmatrix}$$

The matrix equation (9) can be written as

$$V^{m+1} = (I - B\Delta t)V^m + X^m$$

where $B = tridiag(-c_n, e_n, -b_n)$,

$e_n = r + \sigma^2 n^2, n = 1,2,...,N-1$.

Example 1. [5] If the parameters are $0 \le S \le 20, E = 10, r = 0.20, T = 1yr, \sigma = 0.25$ for a European call option value and noting the effect of CFL condition $\Delta t\sigma^2 n^2 = \alpha_1 \le 1$.
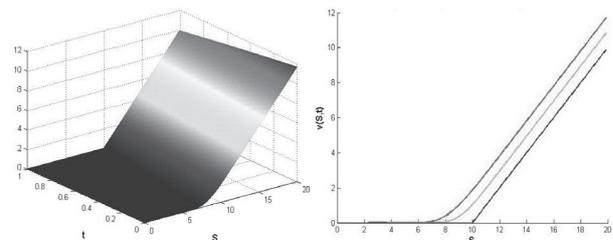
Figure 1: Call option values at expiry, half year before and one year before

The example 1 gives the values of call option for different expiry. The values of the option are obtained by using Explicit finite difference method as shown in figure 1.

When stock price increases from the exercise price $E = 10$ then the holder of the option will exercise at the maturity date.

Example 2. If the parameters are $0 \leq S \leq 20, E = 10, r = 0.20, T = 1yr, \sigma = 0.25$ for a European call option value and noting the effect of $\Delta t \sigma^2 n^2 = \alpha_1 > 1$.
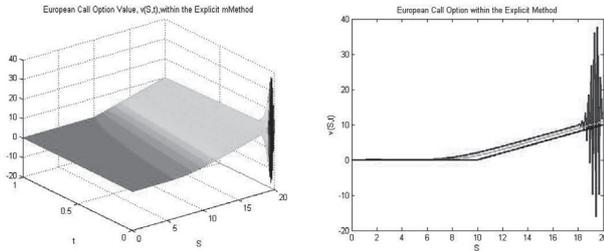
Figure 2: Call option values at different expiry

Figure 2 shows that the rounding errors are growing in magnitude at each iteration of the solution. Therefore, the solution is unstable. The stability problem arises because we are using the finite precision computer arithmetic to solve the difference equation[12]. This introduces rounding error into the numerical solution to (8). The system is stable if these rounding errors are not magnified at each iteration. If the rounding errors do grow in magnitude at each iteration of the solution procedure, then the system (8) is unstable.

It can be shown that the system (8) is[12]:

- stable if $0 < \alpha_1 \leq 1$
- unstable if $\alpha_1 > 1$.

Super-Time-Stepping Scheme[1]

To overcome the stability restriction of the Explicit method, we apply first order STS scheme [1] based on the modified Chebyshev polynomial. In order to relax the CFL condition, we do not require stability at the end of each time step Δt, but rather at the end of a cycle of N timesteps, thus leading to a Runge - Kutta - like method with stages N. We introduce a superstep ΔT consisting of N timesteps $\tau_1, \tau_2, \tau_3, ..., \tau_N$. The idea is now to ensure stability over the superstep ΔT, while trying to maximize its duration

$\Delta T = \sum_{k=1}^{N} \tau_k.$

Runge - Kutta Chebychev (RKC) scheme[1]

In this scheme , we define a super-step ΔT and divide it into τ_k steps with $k = 1, 2, ..., N$, so that the duration of one super-step $\Delta T = \sum_{k=1}^{N} \tau_k.$

Then the Explicit Scheme (8) can be written as,

$$V^{m+1} = \prod_{k=1}^{N} (I - B\tau_k)V^m \qquad (11)$$

The algorithm is subjected to the restrictive stabilty condition if

$$\rho(I - B\Delta t) < 1 \Rightarrow (p_N(\lambda)) < 1 \qquad \forall \lambda \in [\lambda_{min}, \lambda_{max}] \qquad (12)$$

where $\rho(.)$ denote the spectral radius and

$$p_N(\lambda) = \prod_{k=1}^{N} (I - \tau_k \lambda)$$

and λ are the eigenvalues of B. From the modified Chebyshev polynomial, the partitions $\tau_1, \tau_2, \tau_3, ..., \tau_N$ are chosen subject to

$$|p_N(\lambda)| \leq K, (0 < K < 1)$$

and $|p'_N(0)| = \sum_{k=1}^{N} \tau_k$ is maximal.

We can find

$$\tau_k = \Delta t_{expl}\left((-1+\nu)\cos\left(\frac{2k-1}{N}\frac{\pi}{2}\right) + 1 + \nu\right)^{-1}$$

Where ν is damping coefficient and $0 < \nu < \frac{\lambda_{min}}{\lambda_{max}}$. We can show the relation

$$\Delta T = \sum_{k=1}^{N} \tau_k = \Delta t_{expl}\frac{N}{2\sqrt{\nu}}\left(\frac{(1+\sqrt{\nu})^{2N} - (1-\sqrt{\nu})^{2N}}{(1+\sqrt{\nu})^{2N} + (1-\sqrt{\nu})^{2N}}\right) \qquad (13)$$

which gives

$$\Delta T \rightarrow N^2 \Delta t_{expl.} \qquad (14)$$

as $\nu \rightarrow 0$.

Here N explicit steps, each of length Δt_{expl}, cover time $N\Delta t_{expl}$ .Thus, N substeps of a superstep cover a time interval N times longer than N explicit steps when $\nu \rightarrow 0$. Therefore, supersteping is N times faster than the Standard Explicit Scheme at the same cost.

Implementation

One determines the explicit time-step Δt_{expl} in the usual way to satisfy the CFL condition, but instead of executing steps of length Δt_{expl} , one executes supersteps of length ΔT as follows: choose N, ν and execute the N sub-steps $\tau_1, \tau_2, \tau_3, ..., \tau_N$, without outputing until end of each super-step. The only additional expense is trivial computation in τ_k, while the execution is accelerated by a factor of N.

Superstep	ν	N-steps	t-steps
1	0	20480	20480
4	0.20	1024	4096
4	0.15	1088	4352
6	0.10	512	3072
6	0.15	484	2904
8	0.10	288	2304
8	0.15	272	2176

Table 1: Performance of STS

Table 1 shows if the Superstep is one, then the number of steps in Superstep and Δt_{expl} are same; if the Supersteps are four, the number of steps in Superstep is four times the number of steps in Δt_{expl}; if the Supersteps are six, the number of steps in Superstep is six times the number of steps in Δt_{expl} , and if the Supersteps are 8, the number of steps in Superstep is 8 times the number of steps in Δt_{expl}. Also, as Table1 shows, the number of steps of Superstep depends on the damping factor ν.

CONCLUSION

In this work, N substeps of a Superstep cover a time interval N times longer than N explicit steps when $v \rightarrow 0$. Therefore, Supersteping is N times faster than the Standard Explicit Scheme at the same cost.

ACKNOWLEDGEMENT

REFERENCES

1. Alexiades, V., Amiez, V. and Gremaud, P. A. 1996. SuperTime-Stepping acceleration of explicit schemes for parabolic problems. *Communications in Numerical Methods in Engineering*. **12**: 31-42.

2. Black, F. and Scholes, M. 1973. The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy*. **81.3**: 637-654.

3. Duffy, D. J. 2006. *Finite Difference Methods in Financial Engineering*. John Wiley and Sons Ltd, Chichester, United Kingdom.

4. Emmanuel, F.S., Helen, A.F. and Helen , A.O. 2012. On the stability and Accuracy of finite Difference Methods for Options Pricing. *'Mathematical Theory and Modeling*. **6**: 101108.

5. http://www.ehu.es/aitor

6. Wilmott, P. Howison, S. and Dewinne, J., E. 2008. *The Mathematics of financial Derivatives*. Second edition, Cambridge University Press, New York.

■