SUPER-TIME-STEPPING SCHEME FOR OPTION PRICING

Kedar Nath Upreti *, 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 Chebyshhev 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 Scholes2 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 Chebyshhev polynomial1. 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,

\[ \frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = \xi \],

where, \( V = V(S,t) \), the pay of function, \( S = S(t) \), the stock price, \( t = \text{time with } t \in (0,T) \), \( T \) is the time of maturity, \( r > 0 \) is a constant riskless interest rate, \( \sigma \) 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 on \( 0 \leq S < \infty \), \( 0 \leq t \leq T \). The boundary and terminal conditions are:

\[ V(0,t) = 0 \text{ for } 0 \leq t \leq T, \]
\[ V(S,t) = S - e^{-r(T-t)} \text{ as } S \to \infty, \]
\[ V(S,t) = (S-E)^+. \]

European Put Option

The terminal and boundary conditions for European Put option are:

\[ V(0,t) = e^{-r(T-t)} \text{ for } 0 \leq t \leq T, \]
\[ V(S,t) = 0 \text{ as } S \to \infty, \]
\[ V(S,t) = (E-S)^+. \]

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 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. 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. The equation becomes,
\[
\frac{\partial V}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rS \frac{\partial V}{\partial S} + rV = 0 \tag{4}
\]

**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. We get the following discretized form of equation (4)

\[
\begin{align*}
V_{n+1}^m &= \frac{1}{2}\sigma^2 n^2 (\Delta s^2) V_{n+1}^{m+1} - 2V_n^m + V_{n-1}^m \\
&\quad - r n \Delta s \frac{V_{n+1}^m - V_{n-1}^m}{2\Delta s} + r n V_n^m = 0 
\end{align*}
\tag{5}
\]

for \(n = 1, 2, \ldots, N-1\) and \(m = 0, 1, 2, \ldots, M-1\).

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

From (5), we get

\[
\begin{align*}
V_{n+1}^m &= \frac{1}{2}(\sigma^2 n^2 - r n) \Delta t V_n^m \\
&\quad + (1 - (r + \sigma^2 n^2)) \Delta t V_n^m \\
&\quad + \frac{1}{2}(r n + \sigma^2 n^2) \Delta t V_n^m \\
&\quad + 1 - 2 - 3 \ldots - (N-1)
\end{align*}
\tag{6}
\]

\[
\begin{align*}
V_{n+1}^m &= \frac{1}{2}(\sigma^2 n^2 - \beta n) V_n^m \\
&\quad + (1 - (\sigma^2 n^2 + \beta n)) V_n^m \\
&\quad + \frac{1}{2}(\sigma^2 n^2 + \beta n) V_n^m \\
&\quad + 1 - 2 - 3 \ldots - (N-1)
\end{align*}
\tag{7}
\]

\[
\begin{align*}
V_{n+1}^m &= A V_n^m + X^m 
\end{align*}
\tag{9}
\]

For put option

\[
\begin{align*}
V_0^m &= E e^{-rm} \Delta t \\
V_0^m &= 0
\end{align*}
\]

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

\[
\begin{align*}
V_{n+1}^m &= A V_n^m + X^m 
\end{align*}
\tag{9}
\]

We denote

\[
\begin{align*}
\alpha &= \sigma^2 \Delta t \\
\beta &= r \Delta t
\end{align*}
\]

We get

\[
\begin{align*}
V_{n+1}^m &= \frac{1}{2}(\sigma^2 n^2 - \beta n) V_n^m \\
&\quad + (1 - (\sigma^2 n^2 + \beta n)) V_n^m \\
&\quad + \frac{1}{2}(\sigma^2 n^2 + \beta n) V_n^m \\
&\quad + 1 - 2 - 3 \ldots - (N-1)
\end{align*}
\tag{7}
\]

The matrix equation (9) can be written as

\[
\begin{align*}
V_{n+1}^m &= (I - B \Delta t) V_n^m + X^m 
\end{align*}
\tag{9}
\]

where

\[
\begin{align*}
V_{n+1}^m &= \begin{bmatrix} V_1^m \cr V_2^m \cr \vdots \cr V_{N-1}^m \cr \end{bmatrix} \\
V_n^m &= \begin{bmatrix} c_0 V_0^m \\
0 \\
\vdots \\
0 \\
\end{bmatrix} \\
X^m &= \begin{bmatrix} b_0 V_N^m \\
. \\
. \\
. \\
\end{bmatrix}
\end{align*}
\]

for \(m = 0, 1, 2, \ldots, M-1\) and \(n = 1, 2, \ldots, N-1\) and \(m = 0, 1, 2, \ldots, M-1\). Again, we denote

\[
\begin{align*}
\alpha &= (1 - \alpha n^2 - \beta) n, \quad n = 1, 2, \ldots, N-1 \\
b_n &= \frac{1}{2}(\alpha(n-1)^2 + \beta(n-1)), \quad n = 2, 3, \ldots, N \\
c_n &= \frac{1}{2}(\alpha(n+1)^2 - \beta(n+1)), \quad n = 0, 1, 2, \ldots, N-2 \\
\end{align*}
\]

the equation (8) holds for \(n = 1, 2, \ldots, N-1\). Since \(V_0^m\) 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

\[
\begin{align*}
V_0^m &= 0 \quad \text{for all} \ m
\end{align*}
\]

\[
\begin{align*}
\end{align*}
\]
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 \( 0 \leq S \leq 20, E = 10, r = 0.20, T = 1, \sigma = 0.25 \) for a European call option value and noting the effect of \( \Delta \sigma n^2 = a_i > 1 \).

\[
\text{The equation (8) holds for for}
\]

\[
\text{In this case, the error is of order for}
\]

\[
\text{We get the following discretized form of equation (4)}
\]

\[
\text{We use a forward difference approximation for the time}
\]

\[
\text{The Explicit Method}
\]

\[
\text{from the two boundary conditions on}
\]

\[
\text{order S derivative and a symmetric central difference}
\]

\[
\text{derivative, a central difference approximation for the}
\]

\[
\text{+ 1}
\]

\[
\text{= 1}
\]

\[
\text{= 1}
\]

\[
\text{for all}
\]

\[
\text{2}
\]

\[
\text{...,N}
\]

\[
\text{m}
\]

\[
\text{2}
\]

\[
\text{−}
\]

\[
\text{1}
\]

\[
\text{and}
\]

\[
\text{∆}
\]

\[
\text{t}.
\]

\[
\text{S}
\]

\[
\text{−}
\]

\[
\text{1}
\]

\[
\text{−}
\]

\[
\text{1}
\]

\[
\text{and}
\]

\[
\text{∆}
\]

\[
\text{t}
\]

\[
\text{σ}
\]

\[
\text{+}
\]

\[
\text{X}
\]

\[
\text{1}
\]

\[
\text{E}.
\]

\[
\text{s come}
\]

\[
\text{We}
\]

\[
\text{is maximal}.
\]

\[
\text{. We}
\]

\[
\text{is unstable}.
\]

\[
\text{It can be shown that the system (8) is}^{12}:
\]

\[
\text{• stable if } 0 < a_i \leq 1
\]

\[
\text{• unstable if } a_i > 1.
\]

**Super-Time-Stepping Scheme**

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 \( \Delta t \), 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 \( \Delta T \) consisting of \( N \) timesteps \( \tau_1, \tau_2, \tau_3, ..., \tau_N \). The idea is now to ensure stability over the superstep \( \Delta T \), while trying to maximize its duration

\[
\Delta T = \sum_{k=1}^{N} \tau_k,
\]

**Runge - Kutta Chebychev (RKC) scheme**

In this scheme, we define a super-step \( \Delta T \) and divide it into \( \tau_k \) 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,

\[
V^{n+1} = \prod_{k=1}^{N} (I - B \tau_k) V^{n} \tag{11}
\]

The algorithm is subject to the restrictive stability condition if

\[
\rho(I - B \Delta t) < 1 \Rightarrow (p_B(\lambda)) < 1 \quad \forall \lambda \in [\lambda_{\text{min}}, \lambda_{\text{max}}]
\]

where \( \rho(\cdot) \) denote the spectral radius and

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

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

\[
|p_B(\lambda)| \leq K_0 (0 < K < 1)
\]

and \( |p_B(0)| = \sum_{k=1}^{N} \tau_k \) is maximal.

We can find

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

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

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

which gives

\[
\Delta T \to N^2 \Delta t_{\text{expl}} \tag{14}
\]

as \( \nu \to 0 \).

Here \( N \) explicit steps, each of length \( \Delta t_{\text{expl}} \), cover time \( N \Delta t_{\text{expl}} \). Thus, \( N \) sub-steps of a superstep cover a time interval \( N \) times longer than \( N \) explicit steps when \( \nu \to 0 \). Therefore, super-stepping is \( N \) times faster than the Standard Explicit Scheme at the same cost.

**Implementation**

One determines the explicit time-step \( \Delta t_{\text{expl}} \) in the usual way to satisfy the CFL condition, but instead of executing steps of length \( \Delta t_{\text{expl}} \), one executes supersteps of length \( \Delta T \) as follows: choose \( N, \nu \) and execute the \( N \) sub-steps \( \tau_1, \tau_2, \tau_3, ..., \tau_N \), without outputting until end of each superstep. The only additional expense is trivial computation in \( \tau_k \), while the execution is accelerated by a factor of \( N \).

| Superstep | \( \nu \) | \( N \)-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 \( \Delta t_{\text{expl}} \) are same; if the Supersteps are four, the number of steps in Superstep is four times the number of steps in \( \Delta t_{\text{expl}} \); if the Supersteps are six, the number of steps in Superstep is six times the number of steps in \( \Delta t_{\text{expl}} \), and if the Supersteps are eight, the number of steps in Superstep is eight times the number of steps in \( \Delta t_{\text{expl}} \). Also, as Table1 shows, the number of steps of Superstep depends on the damping factor \( \nu \).
CONCLUSION

In this work, $N$ 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.

ACKNOWLEDGEMENT

The second author gratefully acknowledges University Grants Commission (UGC) Nepal for financial support during my study in Master of Philosophy. Also the authors would like to thank all the referee for their comments and suggestions for the improvement of this work.

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.