STUDIA UNIV. "BABEŞ–BOLYAI", MATHEMATICA, Volume **LII**, Number 3, September 2007

ON SOME IMPLICITE SCHEME IN MATHEMATICAL FINANCE

IOANA CHIOREAN

Dedicated to Professor Petru Blaga at his 60^{th} *anniversary*

Abstract. The aim of this paper is to give a parallel approach for the Crank-Nicsloson method applied to the discretized form of the Black-Scholes equation.

1. Introduction in the value of an option

One of the key problems in Mathematical Finance is the determining the value of an option.

According to [8] the simplest financial option, a European call option, is a contract with the following properties:

- at a prescribed time in the future, known as the *expiry date or expiration date* (denoted by T), the holder of the option may

- purchase a prescribed asset, known as the *underlying asset* (denoted by S), for a
- prescribed amount of money, known as the *exercise price* (denoted by E).

Note 1.: The word "may" in this description implies that for the holder of an option, this contract is a "right" and not an "obligation".

The other part, who is known as the *writer,* has a potential obligation: he "must" sell the asset if the holder chooses to buy it. Since the option confers on its holder a right with no obligation, it has some value. Moreover, it must be paid for

Received by the editors: 04.04.2007.

2000 *Mathematics Subject Classification.* 65H05, 65N06, 91B28.

Key words and phrases. parallel calculus, finite difference, financial option.

This work is supported by the research contract 2CEEX-06-11-96.

at the time of opening the contract. Conversely, the writer of the option must be compensated for the obligation he has assumed. So the following questions arise:

- How much would one pay for this right, i.e. what is the value of an option?
- How can the writer minimize the risk associated with his obligation?

Note 2.: There are Call Options (which means the options *to buy* assets) and Put Options (which means *to sell* assets). Whereas the holder of a call option wants the asset prince to rise - the higher the asset price at expiry, the greater the profit - the holder of a put option wants the asset price to fall as low as possible.

2. **The mathematical model**

The problem of determining the value of an option is mathematically modeled by the well known *Black-Scholes equation:*

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 \cdot s^2 \cdot \frac{\partial^2 V}{\partial s^2} + r \cdot s \cdot \frac{\partial V}{\partial s} - r \cdot V = 0 \tag{1}$$

where the following notations are used:

- V- the value of an option, where $V_s = V(S,t)$, with t-the time and S - the underlying asset. If we have a call option, V will be replaced by C, and if we have a put option, it will be replaced by P.
- σ - the volatility of the underlying asset
- r - the interest rate

Note 3.: For a Call option, e.g., the boundary conditions are:

$$V(S,T) = \max(S - E, 0) \tag{2}$$
$$V(0,t) = 0$$

where we denoted by

- E - the exercise price
- T - the expiry.

The exact solution of equation (1) with boundary condition (2) can be determined, but in practice it is difficult to handle. This is the reason for which a numerical approach is preferred.

3. **Solving numericaly the Black-Scholes equation**

In order to obtain the numerical solution of equation (1), one has to discretize it.

The most common way to do this, is by using finite - difference methods.

In the literature, there exist many results in this direction. So, in [6], [7] and [8] one may find the basic tools for numerical option pricing. In [5], a backward differentiations formula is used and in [2], some results are obtained by using an explicit technique.

In what follows, we recall another technique, known as the Crank-Nicolson method.

3.1. **The Crank-Nicolson method.**

As is presented in [8], the Black-Scholes equation can be reduced to a diffusion equation, where the numerical solutions are easier to determine. Then, by a change of variable, these are converted into numerical solutions of the Black-Scholes equation.

So, let us consider the general form of the transformed Black-Scholes model for the value of a European option,

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2} \tag{3}$$

with the boundary conditions

$$u(x, \tau) \backsim u_{-\infty}(x, \tau), u(x, \tau) = u_{\infty}(x, \tau) \text{ as } x \to \pm\infty \tag{4}$$
$$u(x, 0) = u_0(x)$$

Using grid points with the x-axis divided into equally spaced nodes at distance δx apart, and the τ-axis into equally spaced nodes at a distance $\delta\tau$ apart, the grid points have the form $(n\delta x,\ m\delta\tau)$. We denote by

$$u_n^m = u(n\delta x, m\delta\tau) \tag{5}$$

the value of $u(x,\tau)$ at the grid point $(n\delta x,\ m\delta\tau)$. Considering that, on the grid,

$$N^-\delta x \leq x \leq N^+\delta x,\ 0 \leq t \leq M\delta\tau$$

where N^-, N^+ and M are large positive integers, we may write equation (3) with the boundary conditions (4), in the following manner:

$$\frac{u_n^{m+1} - u_n^m}{\delta\tau} + 0(\delta\tau) = \frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\delta x)^2} + 0\left((\delta x)^2\right) \tag{6}$$

by using an explicit formula and

$$\frac{u_n^{m+1} - u_n^m}{\delta\tau} + 0(\delta\tau) = \frac{u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}}{(\delta x)^2} + 0\left((\delta x)^2\right) \tag{7}$$

by using an implicit formula, where the discretized boundary conditions are:

$$u_{N^-}^m = u_{-\infty}(N^-\delta x, m\delta\tau),\ 0 < m \leq M$$
$$u_{N^+}^m = u_{\infty}(N^+\delta x, m\delta\tau),\ 0 < m \leq M.$$

Making the average of (6) and (7), we obtain the Crank-Nicolson formula which, ignoring the error terms, is the following:

$$\begin{aligned} &u_n^{m+1} - \frac{1}{2}\alpha\left(u_{n-1}^{m+1} - 2u_n^{m+1} + u_{n+1}^{m-1}\right) = \\ &u_n^m + \frac{1}{2}\alpha\left(u_{n-1}^m - 2u_n^m + u_{n+1}^m\right) \end{aligned} \tag{8}$$

where

$$\alpha = \frac{\delta\tau}{(\delta x)^2}.$$

In a matriceal form, (8) can be written as follows:

$$A \cdot u^{m+1} = B \cdot u^m \tag{9}$$

where

$$A = \begin{bmatrix} 1+\alpha & -\frac{1}{2}\alpha & 0 & \cdots & 0 \\ -\frac{1}{2}\alpha & 1+\alpha & -\frac{1}{2}\alpha & \cdots & 0 \\ 0 & -\frac{1}{2}\alpha & 1+\alpha & \cdots & \\ \vdots & \vdots & \vdots & & -\frac{1}{2}\alpha \\ 0 & \cdots & 0 & -\frac{1}{2}\alpha & 1+\alpha \end{bmatrix}$$

$$B = \begin{bmatrix} 1-\alpha & \frac{1}{2}\alpha & 0 & \cdots & 0 & \\ \frac{1}{2}\alpha & 1-\alpha & \frac{1}{2}\alpha & & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \frac{1}{2}\alpha \\ 0 & 0 & \cdots & & \frac{1}{2}\alpha & 1-\alpha \end{bmatrix}$$

By successive replacing (9) becomes:

$$u^{m+1} = A^{-1} \cdot B \cdot u^m = (A^{-1} \cdot B)^2 \cdot u^{m+1} = \cdots = (A^{-1} \cdot B)^m \cdot u^0 \qquad (10)$$

where u^0 contains the option values at the initial moment.

In (10) we have to compute the $m^{th}-$power of a matrix product. The complexity of this computation, performed in a usual manner (it means with a serial computer) is $O(n^{3m})$, where n is the dimension of the matrix. In order to improve this complexity, it means to reduce the effort of computation, one way is to use parallel calculus.

4. **Parallel approaches**

Parallel calculus implies the execution of the corresponding algorithm by means of several processors. For more details about parallel computation, in general, see [4]. Many authors use more than one processor to reduce the execution time, for different types of algorithms. Connected with the numerical methods for the Black-Scholes formula, in [1] a parallel approach is proposed which generates an effort of computation of order $O(\log n)$, where n is the dimension of the problem. Also, in [3], by using another parallel technique, a similar result is given.

4.1. **Using the recursive doubling technique.** One possibility to gain speed is to apply the recursive doubling technique (see [4]) to evaluate the matriceal product in (10).

As presented in [4], having enough processors (let's say p, with $p \geq m$) the matriceal product can be performed on a binary tree network: every leaf processor memorizes a pair $A^{-1}B$, and exactly in $[\log_2 m]$ steps, the final product will be obtained in the root processor. The computation effort at every level is of order $O(n^3)$. So the total computational effort will be of order $O\left(n^3 \cdot [\log_2 m]\right)$.

4.2. **Using a parallel matriceal product.** Another possibility to gain speed is to use the p processors (with $p \geq n^3$, this time) to compute in parallel one matriceal product $A^{-1} \cdot B$. According with some technique presented in [4], this can be done exactly in the time needed to perform one single scalar multiplication. So, the total time involved (the computational effort), will be of order $O(m*$ complexity of a scalar multiplication).

5. **Conclusions**

The previous parallel approaches presented above reduce the computational effort and can be used if there are enough processors in the system. Otherwise, the matrices can be divided into blocks, and then some block parallel techniques may be applied.

References

[1] Chiorean, I., *Parallel Algorithms for Solving the Black-Scholes Equation,* Kragujevac J.Math., 27 (2005), pp.39-48.

[2] Chiorean, I., *On some Numerical Methods for Solving the Black-Scholes Formula,* Creative Math. J., vol.13, 2004, pp.31-36.

[3] Chiorean, I., *A cyclic odd-ever reduction technique applied to a parallel evaluation of an explicite scheme in Mathematical Finance,* Studia Univ. Babeş-Bolyai, Mathematica, vol. LI, nr.4, 2006, pp.65-69.

[4] Chiorean, I., *Calcul paralel,* Ed. Microinformatica, 1995.

[5] Hairer, E., Norsett, S.P., Wanner, G., *Solving ordinary differential equations.* I-Nonstiff problems, Springer Veilag, Berlin, 1993.

[6] Seydel, K., *Tools for Computational Finance,* Springer Verlag, Berlin, 2003.

[7] Tavella, D., Randall, C., *Pricing Financial Instruments - The Finite Difference Method,* John Wiley & Sons, Inc. 2000.

[8] Wilmot, P., Howison, S., Dewynne, J., *The Mathematics of Financial Derivatives,* Cambridge Univ. Press, 1995.

Babeş-Bolyai University,
Faculty of Mathematics and Computer Science,
Str. Kogălniceanu Nr. 1, RO-400084 Cluj-Napoca, Romania
E-mail address: ioana@cs.ubbcluj.ro