Neural, Parallel, and Scientific Computations 22 (2014) 521-538

COMPARISON OF SOME NUMERICAL METHODS FOR OPTION PRICING PROBLEMS

MOHMED H. M. KHABIR AND KAILASH C. PATIDAR

Department of Mathematics and Applied Mathematics, University of the Western Cape, Private Bag X17, Bellville 7535, South Africa

ABSTRACT. We propose two numerical methods for pricing European option pricing problem which is represented by a time-dependent parabolic partial differential equation. The first method is based on the semi-discretization by the Method of Lines and then using a finite difference approximation in space whereas a number of MATLAB ode solvers are experimented to perform the time integration. The second one is based on the temporal semi-discretization by implicit Euler and a cubic spline discretization in space. After several numerical comparisons, we found that in terms of applicability, the approach based on splines is more flexible than the one based on the method of line.

AMS (MOS) Subject Classification. 39A05, 65M06, 65M12, 91G60.

1. Introduction

Financial mathematics is a branch of mathematics that assesses the risk and value of various financial instruments. Banks, companies, and other institutions mitigate their risk through financial instruments known as derivatives, that derive their value from some underlying asset. These derivatives are often represented by differential equations. However, equations that arise from pricing and modeling can be very complex, and thus leading to the necessity of numerical methods.

The specific derivatives that we are interested to discuss in this paper are options. An option is a security giving its holder the right to buy or sell an asset, subject to certain conditions, within a specified period of time. If the option is for buying the asset, it is called a *Call option* whereas if it is for selling the asset, then it is called a *Put option*. These options are mainly classified as standard and non-standard options. From these classes, we choose a standard option, namely, European put options for our study in this paper. From the definition of the European option, which states that, a European option can be exercised only on the expiration date, we see that the holder of option has the right without obligation to transact, so the option has some positive value.

Received October 1, 2014 1061-5369 $15.00 ©Dynamic Publishers, Inc.

Numerical methods in option valuation have been investigated by many researchers. The numerical approaches vary from finite element discretizations [7, 12] to finite difference approximations [17]. A finite-difference scheme often employed is the Crank-Nicolson (CN) scheme (see [17]). The CN scheme employs a classical trapezoidal formula for time integration and second-order central difference formulas for discretization of asset derivatives.

Brennan and Schwartz [1] were the first to explore the use of finite-difference methods for pricing options. Geske and Shastri [8] compared the efficiency of various finite-difference and other numerical methods for option pricing. Vázquez [16] presented an upwind scheme for solving the backward parabolic partial differential equation problem arising in the case of European options.

Second-order L-stabilized time integration schemes have been proposed by Chawla *et al.* [3]. Chawla *et al.* [4] presented high-accuracy finite-difference methods for the Black-Scholes equation in which they employed the fourth-order L-stable time integration schemes (LSIMP) developed in Chawla *et al.* [5] and the well-known Numerov method for discretization in the asset direction. They compared the computational efficiency of their LSIMP-NUM schemes with the CN and Douglas schemes by considering valuation of European options and American options via the linear complementarity approach.

Company *et al.* [6] constructed and analyzed a finite difference scheme for solving a nonlinear Black-Scholes partial differential equation modelling stock option prices in the realistic case when transaction costs arising in the hedging of portfolios are taken into account.

The method of lines is an interesting numerical method for solving partial differential equations. The idea is to semi-discretize the PDE into a system of continuous and interdependent ODEs, which can then be solved by using efficient time integration schemes. However, this method is suitable only for certain classes of partial differential equations, namely initial value problems (IVPs). Fortunately, the pricing of the European options meets this criteria. An example of an unsuitable partial differential equation would be the standard Laplace equation which does not have any such initial conditions. The resulting IVPs in our case are solved by using the MATLAB ode suite [13].

After we study the method of lines, we discuss another class of numerical methods, namely, a cubic spline interpolation. In terms of applicability, the approach based on splines is more flexible than the one based on method of line.

The rest of the paper is organized as follows. In Section 2, we describe an option pricing problem and show how to reduce it to a simple parabolic problem. The numerical methods are constructed in sections 3 and 4. Comparative numerical results are presented in Section 5 whereas in Section 6 we summarize the main outcomes.

2. **Problem description**

The value of a European put option satisfies the Black-Scholes equation with appropriately specified final and boundary conditions [14, 17]:

$$\frac{\partial V}{\partial t}+\frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2}+rS\frac{\partial V}{\partial S}-rV=0, \quad 0<S<\infty\,,0<t\leq T. \tag{2.1}$$

The parabolic equation (2.1) satisfies the boundary conditions

$$V(0,t)=V_0(t),\ V(\infty,t)=V_T(S), \tag{2.2}$$

and a final payoff condition

$$V(S,T)=V_T(S), \tag{2.3}$$

for given $V_0(t)$, $V(\infty,t)$ and $V_T(S)$.

In the above, $V=V(S,t)$, S is the value of the underlying asset at time t, σ is the volatility of the underlying asset; E is the exercise price; r is the interest rate and T is the expiry time.

Note that Black and Scholes had proposed the backwards parabolic equation model (2.1) for the valuation of European options with the following final condition at $t=T$:

$$V(S,T)=\max(E-S,0). \tag{2.4}$$

The boundary condition at $S=0$ satisfies

$$V(S,t)=Ee^{-r(T-t)}-S, \tag{2.5}$$

and the boundary condition at $S=+\infty$ satisfies

$$V(S,t)=0. \tag{2.6}$$

Using the log transformation, we transform the Black-Scholes equation (2.1) to a standard diffusion equation as follows:

$$S=Ee^x,\ t=T-\frac{2\tau}{\sigma^2},\ V(S,t)=E\exp\left[-\frac{1}{2}(k-1)x-\frac{1}{4}(k+1)^2\tau\right]u(x,\tau), \tag{2.7}$$

and setting $k=2r/\sigma^2$, we obtain

$$\frac{\partial u}{\partial \tau}=\frac{\partial^2 u}{\partial x^2}, \quad -\infty<x<\infty, \quad 0<\tau\leq\frac{1}{2}\sigma^2 T. \tag{2.8}$$

The final condition (2.4) is transformed to the initial condition

$$u(x,0)=f(x)=\max\left(\exp\left[\frac{1}{2}(k-1)x\right]-\exp\left[\frac{1}{2}(k+1)x\right],0\right) \tag{2.9}$$

and the boundary conditions (2.5) and (2.6) are transformed to

$$u_{-\infty}(\tau) = \exp\left[\frac{1}{2}(k-1)x_{-\infty} + \frac{1}{4}(k+1)^2\tau\right]\exp\left(-\frac{2r\tau}{\sigma^2}\right), \tag{2.10}$$

and

$$u_{\infty}(\tau) = 0. \tag{2.11}$$

In next two sections, we explain two different numerical approaches to solve the above reduced problem.

3. Solving option pricing problem by method of lines

The method of lines (MOL) is used to solve diffusion equations by reducing the problem to an IVP. This is done by introducing approximations for the $x-$derivatives, and using initial value methods to solve the resulting problem. The basic idea behind the MOL is to replace the spatial (boundary-value) derivatives in the PDE with algebraic approximations. Once this is done, the spatial derivatives are no longer stated explicitly in terms of the spatial independent variables. Thus, in effect, only the initial-value variable, typically time in a physical problem, remains. In other words, with only one remaining independent variable, we have a system of ODEs that approximate the original PDE. Once formulating the approximating system of ODEs is done, we can apply any integration algorithm for initial-value ODEs to compute an approximate numerical solution to the PDE. Thus, one of the salient features of the MOL is the use of existing, and generally well-established, numerical methods for IVPs for ODEs.

To proceed with, first we discretize the domain. The infinite interval $-\infty < x < \infty$ is replaced by a finite interval $x_{-\infty} \le x \le x_{\infty}$. The end values $x_{-\infty} = x_{\min} < 0$ and $x_{\infty} = x_{\max} > 0$ should be chosen in such a way that for $S_{\min} = Ee^{x_{-\infty}}$, $S_{\max} = Ee^{x_{\infty}}$ and the interval $S_{\min} \le S \le S_{\max}$, a sufficiently smooth approximation can be obtained. Then for a suitable integer n, the step length in x-direction is defined by $\Delta x = h = (x_{\infty} - x_{-\infty})/n$.

To illustrate the procedure, we carry out the following steps (see [9] for further details) for the diffusion equation (2.8).

The first step is to evaluate the equation at $x = x_i$. This gives

$$u_{\tau}(x_i, \tau) = u_{xx}(x_i, \tau), \qquad 0 \le \tau \le \frac{1}{2}\sigma^2 T. \tag{3.1}$$

Introducing the central difference approximation for the spatial derivative, we obtain

$$u_{\tau}(x_i, \tau) = \frac{u(x_{i+1}, \tau) - 2u(x_i) + u(x_{i-1}, \tau)}{h^2} + \mathcal{O}(h^2). \tag{3.2}$$

Dropping the truncation error term, we obtain

$$\frac{d}{d\tau}u_i(\tau) = \frac{u_{i+1}(\tau) - 2u_i(\tau) + u_{i-1}(\tau)}{h^2}, \quad 1 \le i \le n-1, \tag{3.3}$$

where $u_i(\tau)$ is the resulting approximation for $u(x_i, \tau)$.

Combining all the above steps, we see that the solution to $u_i(\tau)$ is the solution to the following IVP:

$$u^0(x) = \max\left(\exp\left[\frac{1}{2}(k-1)x\right] - \exp\left[\frac{1}{2}(k+1)x\right], 0\right), \quad \text{(initial value)}, \tag{3.4a}$$

$$\begin{cases} u_0 & = u_{-\infty}(\tau), \qquad \text{(value at the left boundary)}, \\ \left(\frac{\mathrm{d}u}{\mathrm{d}\tau}\right)_1 & = \frac{1}{h^2}\left(u_2 - 2u_1 + u_0\right), \\ \left(\frac{\mathrm{d}u}{\mathrm{d}\tau}\right)_2 & = \frac{1}{h^2}\left(u_3 - 2u_2 + u_1\right), \\ \vdots & \vdots \qquad \vdots \\ \left(\frac{\mathrm{d}u}{\mathrm{d}\tau}\right)_{n-2} & = \frac{1}{h^2}\left(u_{n-1} - 2u_{n-2} + u_{n-3}\right), \\ \left(\frac{\mathrm{d}u}{\mathrm{d}\tau}\right)_{n-1} & = \frac{1}{h^2}\left(u_n - 2u_{n-1} + u_{n-2}\right), \\ u_n & = u_{\infty}(\tau) \qquad \text{(the value at right boundary)}. \end{cases} \tag{3.4b}$$

Solving the above problem, we obtain the approximation for $u(x_i, \tau)$.

Collecting the u_i's together (excluding the left and the right boundary values), equation (3.4b) can be written in a vector form as

$$\frac{d}{dt}\mathbf{u}(t) = \mathbf{C}\mathbf{u}. \tag{3.5}$$

where

$$\mathbf{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_{n-1}(t) \end{pmatrix} \tag{3.6}$$

and

$$\mathbf{C} = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & 0 & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix}. \tag{3.7}$$

The initial condition $u(x,0) = f(x)$ now takes the form

$$\mathbf{u}(0) = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \end{pmatrix}. \tag{3.8}$$

Equations (3.5)–(3.8) represents a standard IVP. Furthermore, we can see that the system is strictly diagonally dominant and hence non-singular. This guarantee the uniqueness of the solution. We can now use a variety of IVP solvers to solve the system for $\mathbf{u}$ and recover the solution by using the transformation (2.7). To this end, in this work, we have used MATLAB solvers *ode45*, *ode15s* and *ode23s*.

4. **Solving option pricing problem by cubic splines**

In this section, we present a numerical method which is based on implicit Euler for temporal semi-discretization and then the use of a cubic spline for the discretization in space.

We consider a two-dimensional grid as follows: Let $\Delta\tau$ and Δx, be the mesh step-sizes in the τ and x-directions. The step-size in τ-direction is given by $\Delta\tau = \tau_{\max}/m$ with $\tau_{\max} = \sigma^2 T/2$ where m is an integer. The calculation of the step-size for the x-discretization is done as in the previous section where the method of lines was applied. Note that the equidistant grid is defined in terms of x and τ, and not for S and t. Transforming the (x,τ)-grid via the transformation in (2.7) back to the (S,t)-plane, leads to a nonuniform grid with unequal distances of the grid lines $S = S_i = Ee^{x_i}$.

Time semi-discretization. Now for temporal discretization, we use finite difference technique with uniform step-size $\Delta\tau$, for discretizing equation (2.8) and obtain the following system of linear ordinary differential equations:

$$u^0 = f(x), \qquad -\infty < x < \infty, \tag{4.1a}$$

$$\frac{u^{m+1} - u^m}{\Delta\tau} = u_{xx}^{m+1}, \qquad -\infty < x < \infty, \quad \tau > 0, \tag{4.1b}$$

with the boundary conditions,

$$u^{m+1}(x_{-\infty}) = u_{-\infty}(\tau^{m+1}), \qquad u^{m+1}(x_\infty) = u_\infty(\tau^{m+1}), \tag{4.1c}$$

where u^{m+1} is the solution of Eq. (4.1) at $(m+1)^{th}$ time level. Here $u^m = u(x,\tau^m)$, $\Delta\tau$ is the time step-size and the superscript m denotes m^{th} time level, i.e., $\tau^m = m\Delta\tau$.

At time level $m = 0$, we can rewrite Eq.(4.1) as

$$u^0 = f(x), \qquad -\infty < x < \infty, \tag{4.2a}$$

$$\delta u_{xx}^1 + u^1 = u^0, \qquad -\infty < x < \infty, \quad \tau > 0 \quad , \tag{4.2b}$$

with the boundary conditions,

$$u^1(x_{-\infty}) = u_{-\infty}(\tau^1), \qquad u^1(x_{\infty}) = u_{\infty}(\tau^1), \tag{4.2c}$$

where $\delta = -\Delta\tau$.

The same can be done at all levels. Then at each of these levels, we will use cubic spline approximations to solve the problem in spatial direction. This is explained below.

Spatial discretization. In this section, we describe the derivation of the cubic spline, in general, as well as in context of our problems.

Cubic spline in general. Suppose we have $n+1$ points $x_0, x_1, \ldots, x_n$ in the segment $[a, b]$ which satisfy a grid $a = x_0 < x_1 < \cdots < x_n = b$. These points are called knots. The points x_0 and x_n are called end (boundary) knots. The grid above is called uniform if a distance between every two neighboring knots is the same [15].

A function $S(x)$ given on segment $[a, b]$ is called a spline of type $p+1$ (degree p) if this function consists of piecewise polynomial which are $p-1$ times continuously differentiable on every segment $\triangle_j = [x_j, x_{j+1}]$, $j = 0, 1, \ldots, n-1$, that is, we can write $S(x)$ in the form

$$S(x) = S_j(x) = \sum_{k=0}^{p} a_k^{(j)}(x - x_j)^k, \quad j = 0, 1, \ldots, n-1, \tag{4.3}$$

where $S(x) \in C^{p-1}[a, b]$. The condition $S(x) \in C^{p-1}[a, b]$ means that the function $S(x)$ and its derivatives $S'(x), S''(x), \ldots, S^{p-1}(x)$ at the points $x_1, x_2, \ldots, x_{n-1}$ are continuously differentiable. There is a separate cubic polynomial for each interval:

$$S_j(x) = a_0^{(j)} + a_1^{(j)}(x - x_j) + a_2^{(j)}(x - x_j)^2 + a_3^{(j)}(x - x_j)^3. \tag{4.4}$$

Note that the index (j) of coefficient $a_k^{(j)}$ indicates a system of numbers of the function $S(x)$, see, e.g., [15], for every partial segment $\triangle_j$.

Given a function $y(x)$ defined on $[a, b]$ and a set of knots $a = x_0 < x_1 < \cdots < x_n = b$, a cubic spline interpolant, S, for $y(x)$ is a function that satisfies the following conditions [2]:

(a) S is a cubic polynomial denoted by S_j on the subinterval $[x_j, x_{j+1}]$ for $j = 0, 1, \ldots, n-1$,
(b) $S(x_j) = y(x_j)$ for $j = 0, 1, \ldots, n$,
(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for $j = 0, 1, \ldots, n-2$,
(d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ for $j = 0, 1, \ldots, n-2$,
(e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ for $j = 0, 1, \ldots, n-2$,
(f) one of the following set of end (boundary) conditions is satisfied
    1. $S''(x_0) = S''(x_n) = 0$, (free or natural boundary),

2. $S^{'}(x_0) = y^{'}(x_0)$ and $S^{'}(x_n) = y^{'}(x_n)$, (clamped boundary).

When the free boundary conditions occur, the spline is called a natural spline, and it approximately takes the shape of a long elastic rod if forced to go through the data points. In general clamped splines are more accurate approximations since they include more information about the function.

Why do we need the end conditions? In each interval we need to find 4 coefficients to specify the cubic polynomials, and we have n intervals. We therefore have a total of $4n$ unknowns to find. The conditions (b) give $n+1$ independent equations, and the conditions (c), (d) and (e) give $3 \times (n-1)$ independent equations. So we have $4n$ unknowns and $4n-2$ equations. There are two missing equations, and that is why the end (boundary) conditions (f) are required. The conditions (b) are called the interpolation conditions, and the conditions (c), (d) and (e) are called the continuity conditions.

Now we drive the equation for $S_j(x)$ on the interval $[x_j, x_{j+1}]$. First we define the numbers $z_j = S^{''}(x_j)$. These z_j exist for $0 \leq j \leq n$ and satisfy

$$\lim_{x \to x_j^-} S^{''}(x) = z_j = \lim_{x \to x_j^+} S^{''}(x), \qquad (1 \leq j \leq n-1), \tag{4.5}$$

because $S^{''}(x)$ is continuous at each interior knots [11].

Since $S_j(x)$ is a cubic polynomial on $[x_j, x_{j+1}]$, $S^{''}(x)$ is a linear function satisfying $S_j^{''}(x_j) = z_j$ and $S_j^{''}(x_{j+1}) = z_{j+1}$ and therefore it is given by the straight line between z_j and z_{j+1}, i.e.,

$$S_j^{''}(x) = \frac{z_j}{h_j}(x_{j+1} - x) + \frac{z_{j+1}}{h_j}(x - x_j), \tag{4.6}$$

where $h_j = x_{j+1} - x_j$. Integrating twice, we obtain

$$S_j(x) = \frac{z_j}{6h_j}(x_{j+1} - x)^3 + \frac{z_{j+1}}{6h_j}(x - x_j)^3 + C(x - x_j) + D(x_{j+1} - x), \tag{4.7}$$

where C and D are the integration constants. The interpolation conditions $S_j(x_j) = y_j$ and $S_j(x_{j+1}) = y_{j+1}$ can be imposed on S_j to determine C and D; where we use the notation $y(x_j) = y_j$. Further simplification leads to

$$\begin{aligned} S_j(x) &= \frac{z_j}{6h_j}(x_{j+1} - x)^3 + \frac{z_{j+1}}{6h_j}(x - x_j)^3 + \left(\frac{y_{j+1}}{h_j} - \frac{z_{j+1}h_j}{6}\right)(x - x_j) \\ &\quad + \left(\frac{y_j}{h_j} - \frac{z_j h_j}{6}\right)(x - x_j). \end{aligned} \tag{4.8}$$

To determine $z_1, z_2, \ldots, z_{n-1}$, we use the continuity conditions for $S^{'}$. At the interior knots x_j, we should have $S_{j-1}^{'}(x_j) = S_j^{'}(x_j)$. Equation (4.8) at $x = x_j$ gives

$$S_j^{'}(x_j) = -\frac{h_j}{3}z_j - \frac{h_j}{6}z_{j+1} - \frac{y_j}{h_j} + \frac{y_{j+1}}{h_j}, \tag{4.9}$$

and

$$S'_{j-1}(x_j) = \frac{h_{j-1}}{6} z_{j-1} + \frac{h_{j-1}}{3} z_j - \frac{y_{j-1}}{h_{j-1}} + \frac{y_j}{h_{j-1}}. \tag{4.10}$$

The continuity condition therefore implies

$$h_{j-1}z_{j-1} + 2(h_j + h_{j-1})z_j + h_j z_{j+1} = \frac{6}{h_j}(y_{j+1} - y_j) - \frac{6}{h_{j-1}}(y_j - y_{j-1}), \tag{4.11}$$

where $1 \le j \le n-1$. It then gives a system of $n-1$ linear equations for the $n+1$ unknowns $z_0, z_1, \ldots, z_n$. Also $z_0 = 0$ and $z_n = 0$ corresponds to placing simple supports at the end [1], and then we solve the resulting system of equations to obtain $z_1, z_2, \ldots, z_{n-1}$. The resulting spline function is called a natural cubic spline [11]. The linear system of equations (4.11) with $z_0 = 0$ and $z_n = 0$ is symmetric, tridiagonal, diagonally dominant, and of the form

$$\begin{bmatrix} u_1 & h_1 & & & & \\ h_1 & u_2 & h_2 & & & \\ & h_2 & u_3 & h_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & & h_{n-2} & u_{n-1} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{bmatrix} \tag{4.12}$$

where

$$h_j = x_{j+1} - x_j, u_j = 2(h_j + h_{j-1}), b_j = \frac{6}{h_j}(y_{j+1} - y_j), v_j = b_j - b_{j-1}.$$

Application of cubic spline to option pricing problem. The approximate solution of problem (4.2) is given in the form of a cubic spline $S(x)$, which is denoted by $S_j(x)$ on each subinterval $[x_j, x_{j+1}]$ for $j = 0, 1, \ldots, n-1$, and satisfies the equation

$$\begin{cases} \delta S''(x_j) + S(x_j) = f_j, \;\; x_{-\infty} \leqslant x_j \leqslant x_\infty \\ S(x_{-\infty}) = u_{-\infty}(\tau), \; S(x_{+\infty}) = u_\infty(\tau), \end{cases} \tag{4.13}$$

where $f_j = f(x_j)$. Then we have

$$z_j = S''_j(x_j) = \frac{1}{\delta}\left[f_j - S_j(x_j)\right] = \frac{1}{\delta}\left[f_j - u_j\right], \tag{4.14}$$

where $S \approx u$. We substitute z_j in equations (4.11) and obtain

$$\frac{1}{\delta}h_{j-1}\left[f_{j-1} - u_{j-1}\right] + \frac{2}{\delta}(h_j + h_{j-1})\left[f_j - u_j\right] + \frac{1}{\delta}h_j\left[f_{j+1} - u_{j+1}\right]$$

$$= \frac{6}{h_j}u_{j+1} - \frac{6}{h_j}u_j - \frac{6}{h_{j-1}}u_j + \frac{6}{h_{j-1}}u_{j-1}, \tag{4.15}$$

which upon simplifications leads to

$$\left[\frac{-h_{j-1}}{\delta} - \frac{6}{h_{j-1}}\right] u_{j-1} + \left[\frac{-2(h_j + h_{j-1})}{\delta} + \frac{6}{h_j} + \frac{6}{h_{j-1}}\right] u_j + \left[\frac{-h_j}{\delta} - \frac{6}{h_j}\right] u_{j+1}$$

$$= -\frac{h_{j-1}}{\delta} f_{j-1} - \frac{-2(h_j + h_{j-1})}{\delta} f_j - \frac{h_j}{\delta} f_{j+1}.$$

Multiplying by $-\delta$, we have for $1 \le j \le n-1$:

$$\left[h_{j-1} + \frac{6\delta}{h_{j-1}}\right] u_{j-1} + \left[2(h_j + h_{j-1}) - \frac{6\delta}{h_j} - \frac{6\delta}{h_{j-1}}\right] u_j + \left[h_j + \frac{6\delta}{h_j}\right] u_{j+1}$$

$$= h_{j-1} f_{j-1} + 2(h_j + h_{j-1}) f_j + h_j f_{j+1}. \tag{4.16}$$

By choosing a uniform mesh spacing h, equation (4.16) becomes

$$\left[h + \frac{6\delta}{h}\right] u_{j-1} + \left[4h - \frac{12\delta}{h}\right] u_j + \left[h + \frac{6\delta}{h}\right] u_{j+1}$$

$$= h f_{j-1} + 4h f_j + h f_{j+1}, \tag{4.17}$$

or

$$\gamma_j^- y_{j-1} + \gamma_j^c y_j + \gamma_j^+ y_{j+1} = q_j^- f_{j-1} + q_j^c f_j + q_j^+ f_{j+1}, \tag{4.18}$$

where

$$\gamma_j^- = h + \frac{6\delta}{h}, \gamma_j^c = 4h - \frac{12\delta}{h}, \gamma_j^+ = h + \frac{6\delta}{h}; q_j^- = h, q_j^c = 4h, q_j^+ = h.$$

Equation (4.18) gives a system of $n-1$ linear equations for the unknowns $u_1, u_2, \ldots, u_{n-1}$ with $u_0 = u_{-\infty}(\tau)$ and $u_n = u_\infty(\tau)$ of the form

$$A\mathbf{u} = \mathbf{q}, \tag{4.19}$$

where

$$A = \begin{bmatrix} \gamma_1^c & \gamma_1^+ & & & & \\ \gamma_2^- & \gamma_2^c & \gamma_2^+ & & & \\ & \gamma_3^- & \gamma_3^c & \gamma_3^+ & & \\ & & \ddots & \ddots & \ddots & \\ & & & \gamma_{n-2}^- & \gamma_{n-2}^c & \gamma_{n-2}^+ \\ & & & & \gamma_{n-1}^- & \gamma_{n-1}^c \end{bmatrix}, \tag{4.20}$$

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{bmatrix}, \tag{4.21}$$

and

$$\mathbf{q} = \begin{bmatrix} q_1^- f_0 + q_1^c f_1 + q_1^+ f_2 - \gamma_1^- u_0 \\ q_2^- f_1 + q_2^c f_2 + q_2^+ f_3 \\ q_3^- f_2 + q_3^c f_3 + q_3^+ f_4 \\ \vdots \\ q_{n-2}^- f_{n-3} + q_{n-2}^c f_{n-2} + q_{n-2}^+ f_{n-1} \\ q_{n-1}^- f_{n-2} + q_{n-1}^c f_{n-1} + q_{n-1}^+ f_n - \gamma_{n-1}^+ u_n \end{bmatrix}. \tag{4.22}$$

We can see that the system is strictly diagonally dominant and hence non-singular. Hence this method applied to the problem above using a basis of cubic splines has a unique solution. Note that at each time level we solve the system (4.19) to get the solution of equation (2.8).

5. Numerical simulations and results

In this section, we present some numerical results for the solution of Black-Scholes equation for pricing a European put option. The values $V(S,t)$ can be interpreted as a piece of surface over the subset $S > 0, \quad 0 \le t \le T$ of the (S,t)-plane. We use the following parameters that are used for numerical simulations:

Expiration date	$T = 0.5$ (year)
Exercise price	$E = 10.0$
Risk free interest rate	$r = 0.05$
Volatility	$\sigma = 0.2$
Number of equations	$m = 100$

Figure 1 illustrates the surface for the European put option (obtained by using MOL) for the fixed values of E, T, r and σ. Through Figure 2, we explain that the European put option (obtained by using MOL) can take values above the lower bound $Ee^{-r(T-t)} - S$. For small values of S, the value V approaches its lower bound. The similar observation is made when we used cubic spline and the results are presented in Figures 3 and 4, respectively.

In Table 1, we have tabulated some comparative results. It contains the exact, Quasi-RBFs and MOL solutions for the European put option.

In Table 2, we have tabulated the exact solution, B-spline solution and the solution obtained by method of lines along with MATLAB solver *ode45* for a European put option. We compute results using B-splines with the parameters given above along with $\Delta t = 10^{-5}$ and $\Delta x = 0.005$.

In Table 3 we have tabulated the exact solution and those obtained by using method of lines along with MATLAB solvers *ode45*, *ode15s* and *ode23s*.

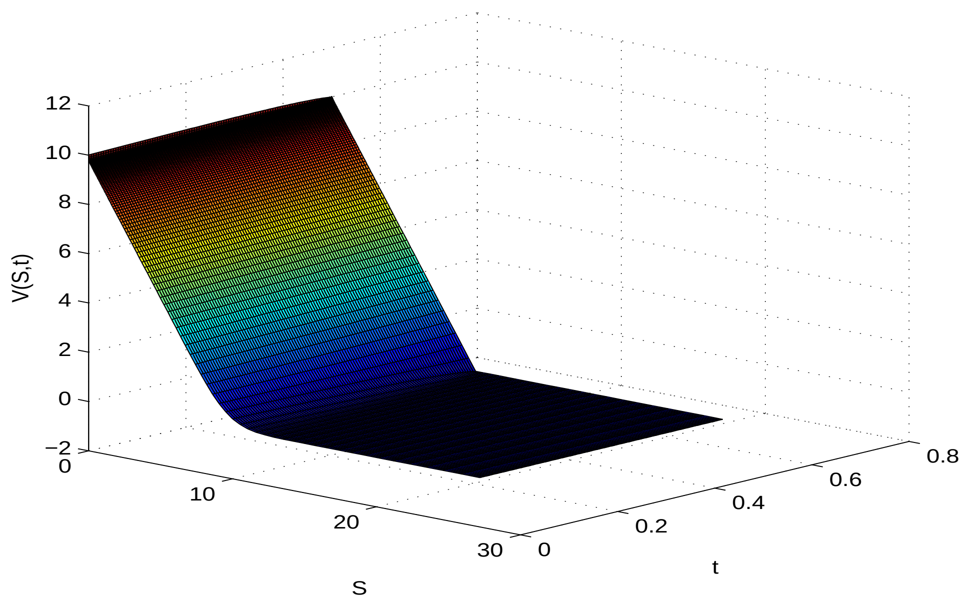

FIGURE 1. Values of European put option obtained by using method of lines for $T = 6/12,\ E = 10,\ r = 0.05,\ \sigma = 0.20$ with $\Delta x = 0.05$, $x \in (-10, 1)$.

Table 4 contains the exact, B-spline and cubic spline solutions for the European put option for $E = 10$, $r = 0.05$, $T = 0.5$, and $\sigma = 0.20$, with $\Delta t = 10^{-5}$ and $\Delta x = 0.005$. Note that the results obtained by cubic spline and B-spline are exactly the same. In Table 5, we have tabulated the exact, B-spline and cubic spline solutions for the European put option for $E = 10$, $r = 0.05$, $T = 0.5$, and $\sigma = 0.20$, with $\Delta t = 10^{-5}$ and $\Delta x = 0.008$. Once again the results obtained by cubic spline and B-spline are exactly the same.

6. **Summary and scope for future research**

In this paper we studied two classes of numerical methods for a European option pricing problem which is represented by a time-dependent parabolic partial differential equation. The first method is based on the semi-discretization by the Method of Lines and then using a finite difference approximation in space where several MATLAB ode solvers are used to perform the time integration. The second one is based on

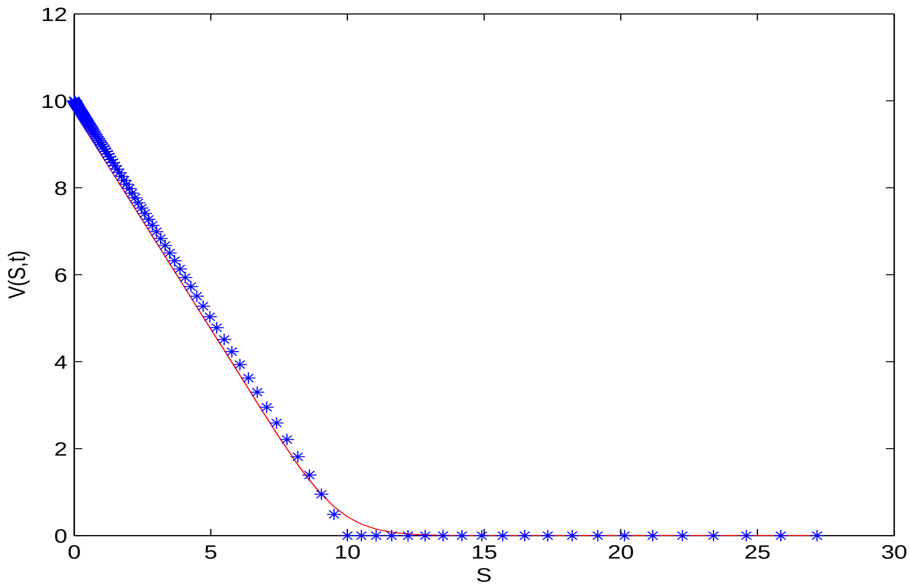

FIGURE 2. Values of European put option at $t = 0$ using method of lines for $T = 6/12$, $r = 0.05$, $\sigma = 0.20$ with $\Delta x = 0.05$. The curve with '*' shows payoff whereas the solid curve represents the value of the option.

TABLE 1. Comparison between the exact solution, Quasi-RBF solution [10] and solution obtained by method of lines along with MATLAB solver *ode45* for a European put option for two different space step-sizes.

S	Exact solution	Quasi-RBF solution [10]	MOL solutions	
			$\Delta x = 0.01$	$\Delta x = 0.005$
2.00	7.7531	7.7531	7.7531	7.7531
4.00	5.7531	5.7531	5.7531	5.7531
6.00	3.7532	3.7532	3.7532	3.7532
7.00	2.7568	2.7568	2.7569	2.7568
8.00	1.7987	1.7988	1.7988	1.7987
9.00	0.9880	0.9881	0.9881	0.9880
10.00	0.4420	0.4420	0.4416	0.4419
11.00	0.1606	0.1606	0.1607	0.1606
12.00	0.0483	0.0483	0.0484	0.0484
13.00	0.0124	0.0124	0.0124	0.0124
14.00	0.0028	0.0028	0.0028	0.0028
15.00	0.0006	0.0006	0.0006	0.0006
16.00	0.0001	0.0001	0.0001	0.0001

TABLE 2. Comparison between the exact solution, B-spline solution and solution obtained by method of lines along with MATLAB solver *ode45* for a European put option for two different space step-sizes.

S	Exact solution	B-spline solution	MOL solutions	
			$\Delta x = 0.01$	$\Delta x = 0.005$
2.00	7.7531	7.7531	7.7531	7.7531
4.00	5.7531	5.7531	5.7531	5.7531
6.00	3.7532	3.7532	3.7532	3.7532
7.00	2.7568	2.7568	2.7569	2.7568
8.00	1.7987	1.7987	1.7988	1.7987
9.00	0.9880	0.9880	0.9881	0.9880
10.00	0.4420	0.4419	0.4416	0.4419
11.00	0.1606	0.1606	0.1607	0.1606
12.00	0.0483	0.0484	0.0484	0.0484
13.00	0.0124	0.0124	0.0124	0.0124
14.00	0.0028	0.0028	0.0028	0.0028
15.00	0.0006	0.0006	0.0006	0.0006
16.00	0.0001	0.0001	0.0001	0.0001

TABLE 3. Comparison between the exact solution and solution obtained by method of lines along with different MATLAB solvers for the European put option.

S	Exact solution	MOL solutions with $\Delta x = 10^{-3}$		
		ode45	ode15s	ode23s
2.00	7.7531	7.7531	7.7531	7.7531
4.00	5.7531	5.7531	5.7531	5.7531
6.00	3.7532	3.7532	3.7532	3.7532
7.00	2.7568	2.7568	2.7568	2.7569
8.00	1.7987	1.7987	1.7987	1.7987
9.00	0.9880	0.9880	0.9880	0.9880
10.00	0.4420	0.4419	0.4419	0.4419
11.00	0.1606	0.1606	0.1606	0.1606
12.00	0.0483	0.0484	0.0484	0.0483
13.00	0.0124	0.0124	0.0124	0.0124
14.00	0.0028	0.0028	0.0028	0.0028
15.00	0.0006	0.0006	0.0006	0.0006
16.00	0.0001	0.0001	0.0001	0.0001

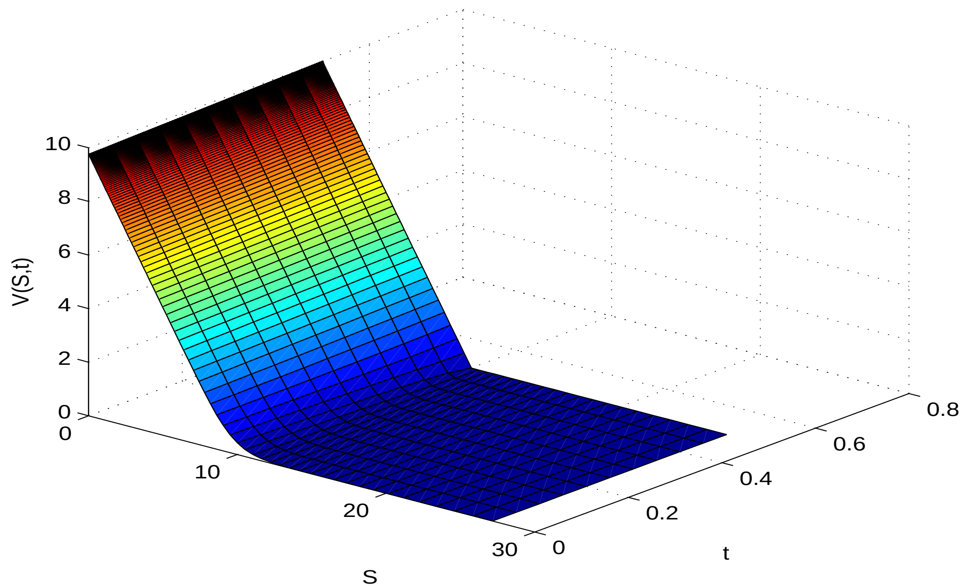

FIGURE 3. Values of European put option obtained by using cubic spline for $T = 6/12,\ E = 10,\ r = 0.05,\ \sigma = 0.20$ with $\Delta\tau = 0.001$, and $\Delta x = 0.05,\ x \in (-10, 1)$.

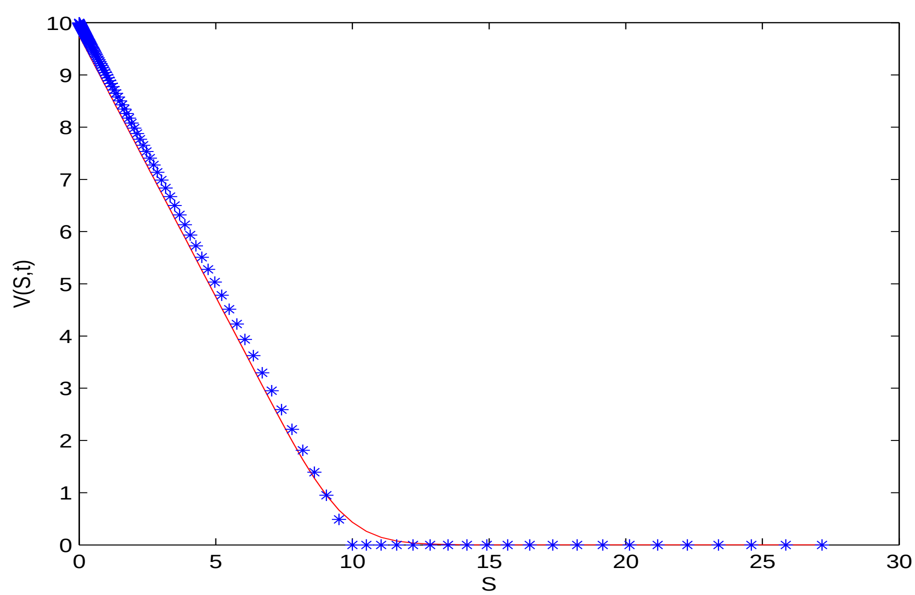

FIGURE 4. Values of European put option at $t = 0$ using cubic spline for $T = 6/12,\ r = 0.05,\ \sigma = 0.20$ with $\Delta\tau = 0.001$, and $\Delta x = 0.05$. The curve with '*' shows payoff whereas the solid curve represents the value of the option.

the temporal semi-discretization by implicit Euler and a cubic spline discretization in

TABLE 4. Comparison between the exact, B-spline and the cubic spline solutions for the European put option for $E = 10$, $r = 0.05$, $T = 0.5$, and $\sigma = 0.2$. With $\Delta x = 0.005$ and $\Delta t = 10^{-5}$.

S	Exact solution	B-spline solution	Cubic spline solution
2.00	7.7531	7.7531	7.7531
4.00	5.7531	5.7531	5.7531
6.00	3.7532	3.7532	3.7532
7.00	2.7568	2.7568	2.7568
8.00	1.7987	1.7987	1.7987
9.00	0.9880	0.9880	0.9880
10.00	0.4420	0.4419	0.4419
11.00	0.1606	0.1606	0.1606
12.00	0.0483	0.0484	0.0484
13.00	0.0124	0.0124	0.0124
14.00	0.0028	0.0028	0.0028
15.00	0.0006	0.0006	0.0006
16.00	0.0001	0.0001	0.0001

TABLE 5. Comparison between the exact, B-spline and the cubic spline solutions for the European put option for $E = 10$, $r = 0.05$, $T = 0.5$, and $\sigma = 0.2$. With $\Delta x = 0.008$ and $\Delta t = 10^{-5}$.

S	Exact solution	B-spline solution	Cubic spline solution
2.00	7.7531	7.7531	7.7531
4.00	5.7531	5.7531	5.7531
6.00	3.7532	3.7532	3.7532
7.00	2.7568	2.7568	2.7568
8.00	1.7987	1.7987	1.7987
9.00	0.9880	0.9880	0.9880
10.00	0.4420	0.4418	0.4418
11.00	0.1606	0.1606	0.1606
12.00	0.0483	0.0483	0.0483
13.00	0.0124	0.0124	0.0124
14.00	0.0028	0.0028	0.0028
15.00	0.0006	0.0006	0.0006
16.00	0.0001	0.0001	0.0001

space (asset) direction. As is seen from the tabular results, for each case, we obtained the results that are comparable with those seen in the literature.

Currently, we are investigative whether we can extend the two approaches proposed in this paper to solve the nonlinear Black-Scholes partial differential equation modelling European and American option pricing problems for multi assets.

ACKNOWLEDGMENTS. MHMK acknowledges the financial support from the Sudan University of Science and Technology (SUST), Sudan. KCP's research was supported by the South African National Research Foundation.

REFERENCES

[1] M. J. Brennan and E. S. Schwartz, Finite difference methods and jump processes arising in the pricing of contingent claims: a synthesis, *J. Fin. Quant. Anal.*, 13:461–474, 1978.

[2] R. L. Burden and J. D. Faires, *Numerical Analysis-5th ed*, PWS-KENT Publishing Company, USA, 1993.

[3] M. M. Chawla, M. A. Al-Zanaidi and D. J. Evans, Generalized trapezoidal formulas for the Black-Scholes equation of option pricing, *Int. J. Comp. Math.*, 80:1521–1526, 2003.

[4] M. M. Chawla and D. J. Evans, High-accuracy finite-difference methods for the valuation of options, *Int. J. Comp. Math.*, 82:1157–1165, 2005.

[5] M. M. Chawla, A. A. Karaballi and M. S. Al-Sahhar, Extended double-stride L-stable methods for the numerical solution of ODEs, *Comp. Math. Appl.*, 31:1–6, 1996.

[6] R. Company, L. Jódar and J.-R. Pintos, A numerical method for European Option Pricing with transaction costs nonlinear equation, *Math. Comp. Model.*, 50:910–920, 2009.

[7] P. A. Forsyth, K. R. Vetzal and R. Zvan, A finite element approach to the pricing of discrete lookbacks with stochastic volatility, *Appl. Math. Fin.*, 6:87–106, 1999.

[8] R. Geske and K. Shastri, Valuation by approximation: a comparison of alternative option valuation techniques, *J. Fin. Quant. Anal.*, 20:45–71, 1985.

[9] M. H. Holmes, *Introduction to Numerical Methods in Differential Equations*, Springer, New York, 2007.

[10] Y. C. Hon, A quasi-radial basis functions method for American options pricing, *Comp. Math. Appl.*, 43:513–524, 2002.

[11] D. R. Kincaid and E. W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, Brooks/Cole Publishing Company, California, 1991.

[12] O. Pironneau and F. Hecht, Mesh adaptation for the black and scholes equations, *East-West J. Numer. Math.*, 8:25–35, 2000.

[13] L. F. Shampine and M. W. Reichelt, The MATLAB ODE suite, *SIAM J. Sci. Comp.*, 18(1):1–22, 1997.

[14] W. T. Shaw, *Modelling Financial Derivatives with Mathematica*, Cambridge University Press, Cambridge, 1998.

[15] E. V. Shikin and A. I. Plis, *Handbook on Splines for the User*, CRC Press, Florida, 1995.

[16] C. Vázquez, An upwind numerical approach for an American and European option pricing model, *Appl. Math. Comput.*, 97:273–286, 1998.

[17] P. Wilmott, S. Howison and J. Dewynne, *The Mathematics of Financial Derivatives*, Cambridge University Press, Cambridge, 1995.