Comp. Appl. Math. (2018) 37:4856–4876
https://doi.org/10.1007/s40314-018-0605-7

An efficient computational algorithm for pricing European, barrier and American options

Somayeh Abdi-Mazraeh[1] · Ali Khani[1]

Received: 14 May 2017 / Revised: 16 February 2018 / Accepted: 8 March 2018 /
Published online: 15 March 2018

Abstract In this work, we use the spectral collocation technique for spatial derivatives and predictor–corrector method for time integration to solve the Black–Scholes (B–S) equation. If the spectral collocation method is worked properly, then we get high accuracy in the numerical solutions. Firstly, theory of application of Chebyshev spectral collocation technique and domain decomposition method on the B–S equation is presented. This method gets a stiff system of differential equations. Secondly, by using the predictor–corrector method with variable step size, we obtain the high accuracy approximate solution in the numerical integration of the stiff system of DEs. We present the order of accuracy for the proposed method. The numerical results show the efficiency and validity of the method.

Keywords Black–Scholes equation · Domain decomposition method · Spectral collocation technique · Predictor–corrector method · Option pricing

Mathematics Subject Classification 65N35 · 35R35 · 60H15

1 Introduction

The B–S equation is one of the popular mathematical models on partial differential equations (PDEs) that was introduced by Black and Scholes in 1973 Black and Scholes (1973). Some of the cases of the B–S equation provide exact closed form solutions Haug (2007), Jiang (2004), but in many cases of this model, the analytical solutions are not available such as nonlinear B–S models Barles and Soner (1998), Company et al. (2008), Sevcovic (2008), nonlocal

Communicated by Jorge Zubelli.

✉ Somayeh Abdi-Mazraeh
s.abdi_m@azaruniv.edu

Ali Khani
khani@azaruniv.edu

1 Department of Sciences, Azarbaijan Shahid Madani University, Tabriz, Iran

volatility models Dupire (1994), etc. These models must be solved using numerical methods. There are many numerical methods for solving these cases. In the numerical solution of these cases, there are many difficulties such as existance of free boundary, and continuous and non-derivative solutions. Nevertheless, a newer approach to solve the B–S equation is given by the finite difference method Hout and Volders (2009), Mashayekhi and Hugger (2015), Wade et al. (2007), finite element method Golbabai et al. (2013), Markolefas (2008), Zhang et al. (2015), radial base functions Bastani et al. (2013), Shcherbakov and Larsson (2016), collocation method Mohammadi (2015) and many other works Chen and Wang (2014), Hurd and Zhou (2010), Khaliq et al. (2008) and Martin-Vaquero et al. (2014).

Mostly, the B–S equation is solved to obtain the price of an option. Two common types of options are the European options and the American options. The European options can be exercised only at expiration date and the American options can be exercised at any time up to the expiration date. For this reason, the problem of American option pricing involves a free boundary condition. Therefore, the American options pricing is more complicated than the European options pricing Haug (2007), Hull (2000) and Jiang (2004).

One way to solve the free boundary value problems is the penalty method. The penalty method converts the free-boundary PDEs to the nonlinear PDEs on a fixed domain by adding a penalty term. Therefore, the use of the penalty method for solving the American option is considered by many researchers Nielsen et al. (2008), Rad et al. (2015), Zhang and Wang (2012) and Zhang et al. (2009). In this paper, we use the penalty method for solving the American option pricing.

Efficient methods to solve PDEs are the spectral collocation methods or the pseudo-spectral methods; for example, see Javidi (2006), Javidi and Golbabai (2009) and Javidi (2011). Pseudo-spectral methods have become increasingly popular for solving PDEs and also very useful in providing highly accurate solutions to PDEs. Using these methods on spatial direction of the PDEs, they get the first-order system of ordinary differential algebraic equations.

One of the most popular methods to solve the first-order differential equation and first-order system of differential equations is the one-step implicit method Izzo and Jackiewicz (2017), Tian et al. (2011) . The implicit methods have high costs because of using Newton's algorithm and calculating Jacobian matrix for systems with higher dimension, then the implicit methods are usually replaced by P–C methods. The main idea in the P–C methods is to use a suitable combination of an explicit and implicit technique to obtain a method with better convergence characteristics. Using the P–C methods for solving the option pricing problem is suggested by many authors Chen et al. (2015), Kalantari et al. (2016), Khaliq et al. (2006) and Zhu and Zhang (2011). One of the most implicit one-step methods that is used for the P–C form is the trapezoidal method. In this method, the order of the predictor is one and the order of the corrector is two. Also, this method is A-stable. Because of using spatial discretization in the B–S equation with the DD method, it leads to a stiff system of DEs; we therefore intend to use the variable step size trapezoidal method. The variable step size method allows a solver to choose the most appropriate step size at a particular point for a given problem. Many researchers have worked to construct the numerical method based on variable step size for solving the initial value problems Amata et al. (2015), Mehrkanoon et al. (2010), Ramos and Singhc (2017) and Thai et al. (2017).

In this paper, we intend to extend the application of the (DD) method based on Chebyshev polynomials to solve the B–S equation in the spatial direction. Our main aim is to get a high accuracy solution of the B–S equation.

In time direction, we intend using the variable step size trapezoidal P–C method to solve the stiff system of first-order DEs obtained by discretizing the B–S equation in spatial direction with the DD method.

The proposed method is A-stable and we get a high accuracy of the order of approximate solution of the B–S equation in spatial and time direction for European option pricing and up-and-out barrier option pricing. The results are in agreement with those obtained by other numerical methods in literature for American option pricing.

This paper is organized as follows: in Sect. 2, a detailed description of the B–S model for European, barrier and American options is provided. In Sect. 3, we describe the DD method based on Chebyshev polynomials. In Sect. 4, implementation of the DD and P–C methods for solving the B–S equation in spatial and time direction is presented. In Sect. 5, we present an analysis of the order of accuracy for the proposed method. In Sect. 6, we report our computational results and demonstrate the accuracy of the proposed numerical scheme by presenting numerical examples. Section 7 ends this paper with a brief conclusion.

2 The B–S model for European, barrier and American options

In the B–S model, the price of an option's underlying asset S follows a geometric Brownian motion that satisfies the following stochastic DE

$$\mathrm{d}S = \mu S \mathrm{d}t + \sigma S \mathrm{d}W, \tag{2.1}$$

where $W(t)$ is a Wiener standard process, and μ and σ are (constant) drift rate and volatility, respectively. Let $V(S, \tau)$ denote the option price on the above asset, T denote the maturity, $t \leq T$ denote a generic instant of time and time to maturity is $\tau = T - t$. Using Itô lemma, it can be shown that $V(S, \tau)$ satisfies Wilmott et al. (1993)

$$\frac{\partial V(S,\tau)}{\partial \tau} - rS\frac{\partial V(S,\tau)}{\partial S} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V(S,\tau)}{\partial S^2} + rV(S,\tau) = 0, \tag{2.2}$$

where r is the interest rate.

2.1 European call option

The B–S model for European call option satisfies the following PDE:

$$\frac{\partial V(S,\tau)}{\partial \tau} = rS\frac{\partial V(S,\tau)}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V(S,\tau)}{\partial S^2} - rV(S,\tau), \quad S \in [0,\infty), \tag{2.3}$$

with initial condition

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

and the boundary conditions

$$V(0,\tau) = 0, \qquad \lim_{S\to+\infty} V(S,\tau) = S - Ee^{-r\tau}, \tag{2.5}$$

where E is the strike price. This option has an exact closed-form solution (see, for example, Hull 2000; Wilmott et al. 1993).

2.2 Up-and-out barrier call option

The following problem

$$\frac{\partial V(S,\tau)}{\partial \tau} = rS\frac{\partial V(S,\tau)}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V(S,\tau)}{\partial S^2} - rV(S,\tau), \quad S \in [0,\infty), \tag{2.6}$$

with inital condition

$$V(S,0) = \max(S-E,0) \tag{2.7}$$

and the boundary conditions

$$V(0,\tau) = 0, \qquad V(B,\tau) = 0, \tag{2.8}$$

is named up-and-out barrier call option with strike price E(constant) and barrier B that has an exact closed-form solution (see, for example, Haug 2007; Jiang 2004).

2.3 B–S model for American put option problem by the penalty method

The B–S model for American put option satisfies the following free boundary PDE:

$$\begin{aligned}
\frac{\partial V(S,\tau)}{\partial \tau} &= rS\frac{\partial V(S,\tau)}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V(S,\tau)}{\partial S^2} - rV(S,\tau), \quad S > B(\tau),\\
V(S,\tau) &= E - S, \quad 0 \le S < B(\tau),\\
\frac{\partial V}{\partial S}(B(\tau),\tau) &= -1,\\
V(B(\tau),\tau) &= E - B(\tau),
\end{aligned} \tag{2.9}$$

with inital condition

$$V(S,0) = \max(E-S,0) \tag{2.10}$$

and the boundary conditions

$$\lim_{S\to+\infty} V(S,\tau) = 0, \tag{2.11}$$

where $B(t)$ represents the free boundary. Note that, since early exercise is permitted, the value $V(S,\tau)$ of the option must satisfy the following formula:

$$V(S,\tau) \ge \max(E-S,0), \quad S \ge 0, \quad 0 \le \tau \le T. \tag{2.12}$$

If Eq. (2.12) were not so, an immediate arbitrage opportunity would ensue.

The above free-boundary PDE does not have an exact closed-form solution; thus, such problems must be priced by numerical techniques Ballestra and Cecere (2016), Zhang et al. (2014) and Zhao et al. (2007).

2.3.1 Penalty method

The penalty method is a simple and efficient method. The main idea of the penalty method is simple. In this method, we replace the free-boundary PDE by the nonlinear PDE on a fixed domain by adding a penalty term.

So, the problem (2.9) is reduced to the following problem (Duffy 2006):

$$
\begin{aligned}
\frac{\partial V(S,\tau)}{\partial \tau} &= rS\frac{\partial V(S,\tau)}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V(S,\tau)}{\partial S^2} - rV(S,\tau) \\
&\quad + \frac{\epsilon C}{V(S,\tau)+\epsilon - E + S}, \quad S \geq 0, \\
V(S,0) &= \max(E-S,0), \\
V(0,\tau) &= E, \\
V(\infty,\tau) &= 0,
\end{aligned} \tag{2.13}
$$

where ϵ is a (small) positive constant and $C \geq rE$ is a constant (both ϵ and C will be chosen in Sect. 6).

3 Analysis of the DD method based on the Chebyshev polynomials

In this section, we first present a brief review of the domain decomposition method based on the Chebyshev polynomials for solving PDEs Javidi (2006), Javidi and Golbabai (2009) and Javidi (2011).

Let $\{T_n(x)\}_{n=0}^{\infty}$ be the Chebyshev polynomials,

$$
\xi_k = \cos\left(\frac{k\pi}{N}\right), \qquad k = 1, \ldots, N-1, \tag{3.1}
$$

and $\xi_0 = \cos(\frac{0\pi}{N}) = 1$ and $\xi_N = \cos(\frac{N\pi}{N}) = -1$ are extreme points of Chebyshev polynomials that recall Chebyshev–Gauss–Lobatto (CGL) points. Now, we define Lagrange polynomials with CGL points as follows:

$$
L_j(x) = \prod_{i=0, i\neq j}^{N} \frac{(x-\xi_i)}{(\xi_j - \xi_i)}, \quad j = 0, \ldots, N. \tag{3.2}
$$

We interpolate function $u(x,t) \in L^2(R^2)$ by the Lagrange polynomials with CGL points of degree at least N of the form

$$
p(u(x,t)) = \sum_{j=0}^{N} u(\xi_j, t) L_j(x) \tag{3.3}
$$

and approximate the polynomial for $u_x(\xi_k, t)$ as follows:

$$
p\left(\frac{\partial}{\partial x} u(\xi_k, t)\right) = \sum_{j=0}^{N} \mathbf{D}_{k,j}^{(1)} u(\xi_j, t), \tag{3.4}
$$

where $\mathbf{D}_{k,j}^{(1)} = \frac{\partial}{\partial x} L_j(x)|_{x=\xi_k}$ and $\mathbf{D}^{(1)} = [\mathbf{D}_{k,j}^{(1)}]$ is named the matrix of the derivative of the first kind Chebyshev polynomials. Also, the approximate polynomial for $u_{xx}(\xi_k, t)$ is as follows:

$$
p\left(\frac{\partial^2}{\partial x^2} u(\xi_k, t)\right) = \sum_{j=0}^{N} \mathbf{D}_{k,j}^{(2)} u(\xi_j, t), \tag{3.5}
$$

where $\mathbf{D}_{k,j}^{(2)} = \frac{\partial^2}{\partial x^2} L_j(x)|_{x=\xi_k}$ and $\mathbf{D}^{(2)} = [\mathbf{D}_{k,j}^{(2)}]$ is named the matrix of the second derivative of the first kind of Chebyshev polynomials Javidi (2011).

4 Implemention of DD and the P–C methods for solving the B–S equation

In this section, first we consider the following PDE:

$$\frac{\partial V(S,\tau)}{\partial \tau} - rS\frac{\partial V(S,\tau)}{\partial S} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V(S,\tau)}{\partial S^2} + rV(S,\tau) = g\big(V(S,\tau)\big), \tag{4.1}$$

with initial and boundary conditions

$$\begin{aligned} V(S,0) &= \psi(S), \quad S \in [0,\infty), \\ V(0,\tau) &= \phi_1(\tau), \\ V(\infty,\tau) &= \phi_2(\tau). \end{aligned} \tag{4.2}$$

To solve this problem, first of all the spatial domain $[0,+\infty)$ is truncated to $[0, S_{\max}]$, where $S_{\max}$ will be chosen large enough such that the error due to the truncation is negligible Golbabai et al. (2013).

Now, by replacing the domain $[0,+\infty)$ with the domain $[0, S_{\max}]$, the initial and the boundary conditions (4.2) are rewritten as follows:

$$\begin{aligned} V(S,0) &= \psi(S), \quad S \in [0,S_{\max}], \\ V(0,\tau) &= \phi_1(\tau), \\ V(S_{\max},\tau) &= \phi_2(\tau). \end{aligned} \tag{4.3}$$

Suppose that domain $\Omega = [0, S_{\max}]$ is divided into m equal subdomains $\Omega_1 = [S_0, S_1]$, $\Omega_2 = [S_1, S_2], \ldots, \Omega_m = [S_{m-1}, S_m]$, where $S_0 = 0$, $S_m = S_{\max}$. The approximate solution to (4.1) equation on Ω is defined by V, and this approximate solution limited to subdomains $\Omega_k = [S_{k-1}, S_k],\ \ k = 1,\ldots,m$ is defined by $V_k,\ \ k = 1,\ldots,m$.

In this case, equation (4.1) is as follows:

$$\begin{aligned} &\frac{\partial V_i(S,\tau)}{\partial \tau} - rS\frac{\partial V_i(S,\tau)}{\partial S} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V_i(S,\tau)}{\partial S^2} + rV_i(S,\tau) = g\big(V_i(S,\tau)\big), \\ &S \in \Omega_i, \quad i = 1,\ldots,m, \end{aligned} \tag{4.4}$$

with initial and boundary conditions

$$\begin{aligned} V_i(S,0) &= \psi(S), \quad S \in \Omega_i, \ \ i = 1,\ldots m, \\ V_1(0,\tau) &= \phi_1(\tau), \\ V_m(S_{\max},\tau) &= \phi_2(\tau). \end{aligned} \tag{4.5}$$

For changing interval $\Omega_i = [S_{i-1}, S_i]$, $i = 1,\ldots,m$ to the interval $[-1,1]$, we use

$$X_i(x) := S = \frac{S_i - S_{i-1}}{2}x + \frac{S_i + S_{i-1}}{2}, \quad S \in \Omega_i,\ x \in [-1,1], \ \ i = 1,\ldots,m. \tag{4.6}$$

By assuming

$$V_i(S,\tau) = u_i(x,\tau), \quad S \in \Omega_i,\ x \in [-1,1],\ i = 1,\ldots,m \tag{4.7}$$

and by applying (4.6) in (4.4), we have

$$\begin{aligned} &\frac{\partial u_i(x,\tau)}{\partial \tau} = R_i^{(1)}(x)\frac{\partial u_i(x,\tau)}{\partial x} + R_i^{(2)}(x)\frac{\partial^2 u_i(x,\tau)}{\partial x^2} - ru_i(x,\tau) + g\big(u_i(x,\tau)\big), \\ &x \in [-1,1], \quad i = 1,\ldots,m, \end{aligned} \tag{4.8}$$

where

$$R_i^{(1)}(x) = rX_i(x)\frac{2}{S_i - S_{i-1}}, \quad i = 1, \ldots, m,$$
$$R_i^{(2)}(x) = \frac{1}{2}\sigma^2\left(X_i(x)\frac{2}{S_i - S_{i-1}}\right)^2, \; i = 1, \ldots, m, \tag{4.9}$$

and initial and boundary conditions are changed to

$$\begin{aligned} u_i(x, 0) &= \psi(X_i(x)), \quad x \in [-1, 1], \; i = 1, \; \ldots, \; m, \\ u_1(-1, \tau) &= \phi_1(\tau), \\ u_m(1, \tau) &= \phi_2(\tau). \end{aligned} \tag{4.10}$$

For matching the approximate solution in the inner nodes, we have

$$\begin{aligned} u_i(1, \tau) &= u_{i+1}(-1, \tau), \quad i = 1, \ldots m - 1, \\ \frac{\partial}{\partial x}u_i(x, \tau)|_{x=1} &= \frac{\partial}{\partial x}u_{i+1}(x, \tau)|_{x=-1}, \quad i = 1, \ldots m - 1. \end{aligned} \tag{4.11}$$

Equation (4.11) guarantees the continuity of u and its derivative in the inner nodes S_i, $i = 1, \ldots, m - 1$. Using Eq. (3.4) in Eq. (4.11) and by assuming

$$\mathrm{d}u_i(\tau) = u_i(1, \tau), \tag{4.12}$$

we have

$$\mathrm{d}u_i(\tau) = \frac{1}{\mathbf{D}_{0,0}^{(1)} - \mathbf{D}_{N,N}^{(1)}}\left[\sum_{j=0}^{N-1}\mathbf{D}_{j,N}^{(1)}u_{i+1}(\xi_j, \tau) - \sum_{j=1}^{N}\mathbf{D}_{j,0}^{(1)}u_i(\xi_j, \tau)\right], \quad i = 1, \; \ldots, \; m - 1. \tag{4.13}$$

By multiplying (4.8) in the Dirac delta function $\delta(x - \xi_j)$ and integrating this in the interval $[-1, 1]$, we have

$$\int_{-1}^{1}\left[\frac{\partial u_i(x, \tau)}{\partial \tau} - R_i^{(1)}(x)\frac{\partial u_i(x, \tau)}{\partial x} - R_i^{(2)}(x)\frac{\partial^2 u_i(x, \tau)}{\partial x^2} + ru_i(x, \tau) + g\big(u_i(x, \tau)\big)\right]\delta(x - \xi_j)\mathrm{d}x = 0, \quad i = 1, \; \ldots, \; m. \tag{4.14}$$

We get

$$\frac{\partial u_i(\xi_j, \tau)}{\partial \tau} = R_i^{(1)}(\xi_j)\frac{\partial u_i(\xi_j, \tau)}{\partial x} + R_i^{(2)}(\xi_j)\frac{\partial^2 u_i(\xi_j, \tau)}{\partial x^2} - ru_i(\xi_j, \tau) + g\big(u_i(\xi_j, \tau)\big), \quad i = 1, \ldots, m. \tag{4.15}$$

By assuming $u_i(\xi_j, \tau) = u_{i,j}(\tau)$, we have

$$\frac{\partial u_{i,j}(\tau)}{\partial \tau} = R_i^{(1)}(\xi_j)\frac{\partial u_{i,j}(\tau)}{\partial x} + R_i^{(2)}(\xi_j)\frac{\partial^2 u_{i,j}(\tau)}{\partial x^2} - ru_{i,j}(\tau) + g\big(u_{i,j}(\tau)\big), \quad i = 1, \ldots, m. \tag{4.16}$$

The initial and boundary conditions change to

$$\begin{aligned} u_i(\xi_j, 0) &= u_{i,j}(0) = \psi(X_i(\xi_j)), \quad i = 1, \ldots, m, \\ u_{1,N}(\tau) &= \phi_1(\tau), \\ u_{m,0}(\tau) &= \phi_2(\tau), \end{aligned} \tag{4.17}$$

and from (4.11), we have

$$
\begin{aligned}
u_{i,0}(\tau) &= u_{i+1,N}(\tau), \quad i = 1, \ldots, m, \\
\frac{\partial}{\partial x} u_{i,0}(\tau) &= \frac{\partial}{\partial x} u_{i+1,N}(\tau), \quad i = 1, \ldots, m.
\end{aligned} \tag{4.18}
$$

Using (3.3)–(3.5) in (4.16), we get

$$
\frac{\partial u_{i,j}(\tau)}{\partial \tau} = \sum_{k=0}^{N} [R_i^{(1)}(\xi_j)\mathbf{D}_{j,k}^{(1)} + R_i^{(2)}(\xi_j)\mathbf{D}_{j,k}^{(2)} - rL_j(\xi_k)] u_{i,k}(\tau) + g\big(u_{i,j}(\tau)\big),
$$

$$
i = 1, \ldots, m. \tag{4.19}
$$

By assuming $Z_{i,j,k} = R_i^{(1)}(\xi_j)\mathbf{D}_{j,k}^{(1)} + R_i^{(2)}\mathbf{D}_{j,k}^{(2)} - rL_j(\xi_k), \ \ i = 1, \ldots, m$, Eq. (4.19) can be written as follows:

$$
\begin{cases}
\dfrac{\partial u_{1,j}(\tau)}{\partial \tau} = \displaystyle\sum_{k=0}^{N} Z_{1,j,k} u_{1,k}(\tau) + g\big(u_{1,j}(\tau)\big), \quad j = 0, \ldots, N-1, \\
\dfrac{\partial u_{i,j}(\tau)}{\partial \tau} = \displaystyle\sum_{k=0}^{N} Z_{i,j,k} u_{i,k}(\tau) + g\big(u_{i,j}(\tau)\big), \quad i = 2, \ldots, m-1, \ \ j = 0, \ldots, N-1, \\
\dfrac{\partial u_{m,j}(\tau)}{\partial \tau} = \displaystyle\sum_{k=0}^{N} Z_{m,j,k} u_{m,k}(\tau) + g\big(u_{m,j}(\tau)\big), \quad j = 1, \ldots, N.
\end{cases} \tag{4.20}
$$

By substituting (4.11)–(4.13) and boundary conditions of (4.17) in (4.20), we have

$$
\begin{cases}
\dfrac{\partial u_{1,j}(\tau)}{\partial \tau} = \displaystyle\sum_{k=0}^{N-1} Z_{1,j,k} u_{1,k}(\tau) + Z_{1,j,N}\phi_1(\tau) + g\big(u_{1,j}(\tau)\big), \quad j = 0, \ldots, N-1, \\
\dfrac{\partial u_{i,j}(\tau)}{\partial \tau} = Z_{i,j,0}\mathrm{d}u_i(\tau) + \displaystyle\sum_{k=1}^{N-1} Z_{i,j,k} u_{i,k}(\tau) + Z_{i,j,N}\mathrm{d}u_{i-1}(\tau) + g\big(u_{i,j}(\tau)\big), \quad i = 2, \ldots, m-1, j = 0, \ldots, N-1, \\
\dfrac{\partial u_{m,j}(\tau)}{\partial \tau} = Z_{m,j,0}\phi_2(\tau) + \displaystyle\sum_{k=1}^{N-1} Z_{m,j,k} u_{m,k}(\tau) + Z_{m,j,N}\mathrm{d}u_{m-1}(\tau) + g\big(u_{m,j}(\tau)\big), \quad j = 1, \ldots, N.
\end{cases} \tag{4.21}
$$

Now we can write Eq. (4.21) as

$$
\begin{cases}
\dfrac{\partial U(\tau)}{\partial \tau} = F(\tau, U(\tau)), \\
U(0) = U_0,
\end{cases} \tag{4.22}
$$

where

$U(\tau) = [u_{1,0}(\tau), \ldots, u_{1,N-1}(\tau), \ldots, u_{i,0}(\tau), \ldots, u_{i,N-1}(\tau), \ldots, u_{m,1}(\tau), \ldots, u_{m,N}(\tau)]$

and

$$
F(\tau, U(\tau)) = [F_{1,0}, \ldots, F_{1,N-1}, \ldots, F_{i,0}, \ldots, F_{i,N-1}, \ldots, F_{m,1}, \ldots, F_{m,N}],
$$

where

$$
\begin{cases}
F_{1,j} = \displaystyle\sum_{k=0}^{N-1} Z_{1,j,k} u_{1,k}(\tau) + Z_{1,j,N}\phi_1(\tau) + g\big(u_{1,j}(\tau)\big), \quad j = 0, \ldots, N-1, \\
F_{i,j} = Z_{i,j,0}\mathrm{d}u_i(\tau) + \displaystyle\sum_{k=1}^{N-1} Z_{i,j,k} u_{i,k}(\tau) + Z_{i,j,N}\mathrm{d}u_{i-1}(\tau) + g\big(u_{i,j}(\tau)\big), \quad i = 2, \ldots, m-1, j = 0, \ldots, N-1, \\
F_{m,j} = Z_{m,j,0}\phi_2(\tau) + \displaystyle\sum_{k=1}^{N-1} Z_{m,j,k} u_{m,k}(\tau) + Z_{m,j,N}\mathrm{d}u_{m-1}(\tau) + g\big(u_{m,j}(\tau)\big), \quad j = 1, \ldots, N,
\end{cases} \tag{4.23}
$$

with the initial vector as follows:

$$U_0 = [u_{1,0}(0), \ldots, u_{1,N-1}(0), \ldots, u_{i,0}(0), \ldots, u_{i,N-1}(0), \ldots, u_{m,1}(0), \ldots, u_{m,N}(0)], \tag{4.24}$$

where $u_{i,j}(0) = \psi(X_i(\xi_j))$. Now, we must resolve the initial value problem (IVP) obtained from (4.22).

Because the IVP (4.22) is the stiff system, this problem must solve by A-stable implicit method. We use the implicit trapezoidal method. Since the implicit methods are costly for systems with higher dimension, we rewrite the implicit trapezoidal method as the P–C method. The solution obtained in the stiff problem can be changed on a timescale that is very small as compared to the interval of integration. Methods that are designed as constant step size must use in time steps small enough to resolve the fastest possible change in intervals where the solution changes slowly. It has high computation cost. So, using a variable step size can be useful for this problem. The main idea of variable step size is that we want the step size for the next step to be optimal. In variable step size, to replace the step size smoothly and alleviate the risk of unstable behavior in the numerical solution, we require to increase or decrease the step size gradually. If the new step size is chosen too big, then a larger error will happen. If the new step size is chosen too small, then it will take more steps to reach $\tau_{\text{end}} = T$ (for more details, see Huang 2005).

The formulae required for the P–C method with variable step size are:

$$\begin{aligned} U_{n+1}^{(1)} &= U_n + F(\tau_n, U_n), \\ U_{n+1}^{(\mu+1)} &= U_n + \frac{h_n F(\tau_n, U_n)}{2}\Big[F(\tau_n, U_n) + F(\tau_{n+1}, U_{n+1}^{(\mu)})\Big], \quad \mu = 1, \ldots, M, \\ \text{ER}_{n+1} &= \|U_{n+1}^{(\mu+1)} - U_{n+1}^{(1)}\|_2, \\ h_{n+1}^* &= h_n \left(\frac{\alpha \text{Tol}}{\text{ER}_{n+1}}\right)^{\frac{1}{(p+1)}}, \end{aligned} \tag{4.25}$$

where $U_{n+1}^{(1)}$ is the predictor of order 1, $U_{n+1}^{(\mu+1)}$ is the corrector of order 2 and $h_n = \tau_{n+1} - \tau_n$ is the step size. Moreover Tol is the tolerance for the method provided by the user, and α is a value between 0 and 1. In this paper, we choose $\alpha = 0.5$ and p is the order of the method. Here, $p = 2$. Also, If $\text{ER}_{n+1} < \text{Tol}$, the step is accepted, otherwise the step is rejected. Also, if a step is rejected, then the step size is halved and the step is repeated. If a step is accepted, then the next step size is as follows:

$$h_{n+1} = \min(h_{n+1}^*, 2). \tag{4.26}$$

Consequently, by assuming

$$K_i(S) = \frac{2S - (S_i + S_{i-1})}{S_i - S_{i-1}}, \quad i = 1, \ldots, m, \; S \in [S_{i-1}, S_i], \tag{4.27}$$

we have a piecewise interpolation solution of (4.1)–(4.2) in $\tau = T$ as follows:

$$\tilde{V}(S, T) = \sum_{i=1}^{m} \left(\sum_{j=0}^{N} u_{i,j}(T) L_j(K_i(S)) \right) \mathbf{1}_{[S_{i-1}, S_i]}, \tag{4.28}$$

where

$$\mathbf{1}_{[S_{i-1}, S_i]} = \begin{cases} 1, & S \in [S_{i-1}, S_i], \\ 0, & \text{otewise.} \end{cases} \tag{4.29}$$

4.1 Implemention for the European call option

If in (4.1), (4.3) we assume

$$
\begin{aligned}
g\big(V(S,\tau)\big) &= 0,\\
\psi(S) &= \max(S-E,0),\\
\phi_1(\tau) &= 0,\\
\phi_2(\tau) &= S - Ee^{-r\tau},
\end{aligned}
\tag{4.30}
$$

we will obtain an approximate solution for the B–S model for the European call option .

4.2 Implemention for the up-and-out barrier call option

If in (4.1), (4.3), we assume

$$
\begin{aligned}
g\big(V(S,\tau)\big) &= 0,\\
\psi(S) &= \max(S-E,0),\\
\phi_1(\tau) &= 0,\\
S_{\max} &= B,\\
\phi_2(\tau) &= 0,
\end{aligned}
\tag{4.31}
$$

we will obtain an approximate solution for the B–S model for the up-and-out barriercall option.

4.3 Implemention for the American put option

If in (4.1), (4.3) we assume

$$
\begin{aligned}
g\big(V(S,\tau)\big) &= \frac{\epsilon C}{V(S,\tau)+\epsilon - E + S},\\
\psi(S) &= \max(E-S,0),\\
\phi_1(\tau) &= E,\\
\phi_2(\tau) &= 0,
\end{aligned}
\tag{4.32}
$$

we will obtain an approximate solution for the B–S model for the American put option.

5 Analysis of the order of accuracy for the proposed method

As in the previous section, we divide the interval $[0, S_{\max}]$ to subinterval $\Delta = \{S_0 = 0 < S_1 \ldots < S_m = S_{\max}\}$. Let $\xi_k = -\cos(\frac{k\pi}{N}),\quad k = 0,\ldots,N$ and $\Delta S_i = S_{i+1} - S_i$. By changing the variable $S_{ik} = S_i + \frac{1}{2}\Delta S_i(1+\xi_k),\ i = 0,\ldots,m-1,\ k = 0,\ldots,N$, we get CGL nodes in $[S_i, S_{i+1}]$ where $S_{i0} = S_i,\ \ S_{iN} = S_{i+1}$.

Now, we approximate $V(S,\tau)$ in the S-direction in the subinterval $[S_i, S_{i+1}]$

$$
\tilde{V}(S,\tau) = \sum_{j=0}^{N} C_j(\tau)T_j\left(2\frac{S-S_i}{\Delta S_i} - 1\right)
\tag{5.1}
$$

and

$$\int_{S_i}^{S_i+\theta\Delta S_i} \tilde{V}(S,\tau)\mathrm{d}S = \Delta S_i \sum_{k=0}^{N} b_k(\theta)\tilde{V}(S_{kj},\tau), \tag{5.2}$$

where

$$b_k(\theta) = \frac{2}{N}\sum_{j=0}^{N}\frac{H_j(\theta)}{e_j e_k}T_j(\xi_k), \tag{5.3}$$

$$H_j(\theta) = \begin{cases} \frac{1}{(1+j)}[T_{j+1}(2\theta-1)+(-1)^j] - \frac{1}{(j-1)}[T_{j-1}(2\theta-1)+(-1)^j], & j\geq 2\\ \frac{1}{8}T_2(2\theta-1)-1, & j=1,\\ \frac{1}{2}T_1(2\theta-1)-1, & j=0, \end{cases} \tag{5.4}$$

and

$$e_j = \begin{cases} 2 & j=0,\ N,\\ 1 & j\neq 0,\ N, \end{cases}. \tag{5.5}$$

Lemma 5.1 (See Khater et al. 2000)

$$\sum_{k=0}^{N} b_k(\theta)\left(\frac{1+\xi_k}{2}\right)^i = \frac{\theta^{i+1}}{i+1},\quad i=0,\ldots,N. \tag{5.6}$$

Theorem 5.1 *Let $V(S,\tau)$ be a sufficient differentiation in the S-direction for $S=S_i$, then the order of accuracy of $\frac{\partial V}{\partial S}$ and $\frac{\partial^2 V}{\partial S^2}$ by using the DD method in $S=S_i$ is as follows:*

$$\mathrm{LTE}\left[\frac{\partial V(S,\tau)}{\partial S}|_{S=S_i}\right] = \mathrm{LTE}\left[\frac{\partial^2 V(S,\tau)}{\partial S^2}|_{S=S_i}\right] = O((\Delta S_i)^{N+2}). \tag{5.7}$$

Proof By assuming $g(s,\tau) = \frac{\partial V(S,\tau)}{\partial S}$, we have

$$V(S,\tau) = V(S_i,\tau) + \int_{S_i}^{S} g(z,\tau)dz. \tag{5.8}$$

By substituting $S = S_i + \Delta S_i\theta$ in (5.8), we get

$$V(S_i+\Delta S_i\theta,\tau) = V(S_i,\tau) + \int_{S_i}^{S_i+\Delta S_i\theta} g(z,\tau)dz. \tag{5.9}$$

Now, by using (5.2) in the right hand side of (5.9) and Taylor approximation g on S_i, we have:

$$\begin{aligned} V(S_i+\Delta S_i\theta,\tau) &= V(S_i,\tau) + \Delta S_i\sum_{k=0}^{N} b_k(\theta) g\left(S_i + \Delta S_i\frac{1+\xi_k}{2},\tau\right) + \mathrm{LTE}[V(S_i,\tau)]\\ &= V(S_i,\tau) + \Delta S_i\sum_{k=0}^{N} b_k(\theta)\sum_{l=0}^{\infty}\frac{[\Delta S_i(1+\xi_k)/2]^l}{l!}\frac{\partial V^{l+1}(S_i,\tau)}{\partial S^{l+1}}\\ &\quad + \mathrm{LTE}[V(S_i,\tau)] \end{aligned}$$

$$= V(S_i,\tau) + \Delta S_i \sum_{l=0}^{N} \frac{(\Delta S_i)^l}{l!} \frac{\partial V^{l+1}(S_i,\tau)}{\partial S^{l+1}} \sum_{k=0}^{N} b_k(\theta) \left(\frac{1+\xi_k}{2}\right)^l$$
$$+ O((\Delta S_i)^{N+2}) + \mathrm{LTE}[V(S_i,\tau)]. \tag{5.10}$$

Using lemma 5.1 in (5.10), we have

$$V(S_i + \Delta S_i \theta, \tau) = V(S_i,\tau) + \Delta S_i \sum_{l=0}^{N} \frac{(\Delta S_i)^l}{l!} \frac{\partial V^{l+1}(S_i,\tau)}{\partial S^{l+1}} \frac{\theta^{l+1}}{l+1}$$
$$+ O((\Delta S_i)^{N+2}) + \mathrm{LTE}[V(S_i,\tau)]. \tag{5.11}$$

Finally, using Taylor expansion in the left hand side of (5.11) on S_i and simplifying them, we obtain

$$\mathrm{LTE}\left[\frac{\partial V(S,\tau)}{\partial S}|_{S=S_i}\right] = O((\Delta S_i)^{N+2}). \tag{5.12}$$

Similarly, we can obtain $\mathrm{LTE}[\frac{\partial^2 V(S,\tau)}{\partial S^2}|_{S=S_i}] = O((\Delta S_i)^{N+2})$. □

According to (4.22), it can be shown that

$$\frac{\partial u_{i,0}(\tau)}{\partial \tau} = F_{i,0}(\tau, u_{i,0}(\tau)) + O((\Delta S_i)^{N+2}), \quad i = 1, \ldots, m, \tag{5.13}$$

where in this problem $\Delta S_i = \frac{1}{m}, \ i = 1, \ldots, m-1$.

Now, by using the P–C method to define in Sect. 4, we approximately obtain the order of accuracy of the problem as follows:

$$V(S_i, \tau_j) = \tilde{V}(S_i, \tau_j) + \max\left\{O\left(\left(\frac{1}{m}\right)^{N+2}\right), O((h^3))\right\},$$
$$i = 0, \ldots, m-1, \ j = 0 : h : T, \tag{5.14}$$

where h is the maximum step size of the P–C method.

6 Numerical experiments

In this section, we give some numerical experiments proposed in Sect. 4 to show the efficiency of the method. The numerical experiments are performed on a computer Intel(R) Core(TM)2 Quad CPU Q8300 @ 2.50 GHz 2.50 GHz 4GB RAM and the software programs are developed and run under Matlab 2016, 64-bit.

Because of non-smoothness of the option's payoff in strike price $S = E$, which is one of the reasons for less order of accuracy in numerical works (not satisfying in Theorem 5.1), we choose to divide interval $[0, S_{\max}]$ to subintervals $[S_0, S_1], \ldots, [S_{m-1}, S_m]$, such that the strike price E coincides in $S_i, \ i = 1, \ldots, m-1$. For this reason, in European and American options by assuming $E = S_k, 0 < k < m, k \in \mathbb{N}$, we have $\frac{k}{m} S_{\max} = E$. So, we get

$$S_{\max} = \frac{mE}{k}. \tag{6.1}$$

On the other hand, by assuming $S_{\max} \geq 2E$ Golbabai et al. (2013), Tagliani et al. (2001), we have $\frac{mE}{k} \geq 2E$ and then $k \leq \frac{m}{2}$. Finally, we get

$$k = \left\lfloor \frac{m}{2} \right\rfloor. \tag{6.2}$$

By substituting (6.2) in (6.1), we obtain $S_{\max} = \frac{mE}{\lfloor \frac{m}{2} \rfloor}$.

For barrier call option, we assume $S_{\max} = B$. Now for coinciding $E \in \{S_i, \quad i = 1, \ldots, m-1\}$, we choose m so that the strike price E is one of the subinterval nodes of the interval $[0, B]$.

6.1 Numerical experiments for European call option and up-and-out barrier call option

For the European call option, the parameters and coefficients are chosen as follows: $\sigma = 0.2$, $r = 0.05$, $T = 0.5$, $E = 10$, and for the up-and-out barrier call option, we set: $\sigma = 0.2$, $r = 0.05$, $T = 0.5$, $E = 100$, $B = 120$.

We want to test the error function in $[0, S_{\max}]$ at $\tau = T$ as follows:

$$\mathrm{Error}(S)_{m,N} = \max_{S} |V(S,T) - \tilde{V}(S,T)|, \tag{6.3}$$

where $V(S,\tau)$ and $\tilde{V}(S,\tau)$ are the exact solution and the approximate solution of the B–S equation, respectively. The convergence rates (ratio) are estimated by dividing the error of the previous coarser spatial discretization by the error of the current discretization. For all tables `Accepted steps` is the number of accepted steps, `Total steps` is the total number of steps used and h_0 is the initial step size.

Tables 1 and 2 show the maximal error for the European call option with $N = 10$ and different values of m, and with $m = 12$ and different values of N, respectively.

Table 3 shows that the maximal error for barrier call option with $N = 7, 9$ and $m = 6,\ 12,\ 24$.

Tables 4 shows that the maximal error decreases by decreasing tolerance Tol.

Table 5 shows the efficiency of the variable step size rather than the constant step size. This table shows the error of the proposed method and the constant step size method, and shows the number of steps and cpu time for constant step size case by $\mathrm{Error}_{\mathrm{fix}}$, $\mathrm{steps}_{\mathrm{fix}}$ and $\mathrm{cpu}_{\mathrm{fix}}$.

Table 1 The $\mathrm{Error}(S)_{m,N}$ for the European call option for fixed $N = 10$ and different values of m nodes

m	$\mathrm{Error}(S)_{m,N}$	Ratio	Tol	h_0	`Accepted steps`	cpu t(s)
4	3.30×10^{-4}	–	10^{-3}	10^{-1}	157	0.13
7	2.37×10^{-6}	139.24	10^{-5}	10^{-3}	629	0.51
12	8.47×10^{-8}	27.98	10^{-7}	10^{-4}	4232	5.07
15	2.51×10^{-9}	33.74	10^{-8}	10^{-4}	13,400	19.82

Table 2 The $\mathrm{Error}(S)_{m,N}$ for the European call option for fixed $m = 12$ and different values of N collocation points

N	$\mathrm{Error}(S)_{m,N}$	Ratio	Tol	h_0	`Accepted steps`	cpu t(s)
5	2.97×10^{-3}	–	10^{-2}	10^{-2}	99	0.17
7	2.51×10^{-5}	118.32	10^{-4}	10^{-2}	385	0.50
9	2.17×10^{-7}	115.66	10^{-6}	10^{-3}	1580	2.03
11	1.04×10^{-8}	20.86	10^{-7}	10^{-4}	4537	5.69

Table 3 The $\mathrm{Error}(S)_{m,N}$ for the barrier option for different values of m and N

N	m	$\mathrm{Error}(S)_{m,N}$	Ratio	Tol	h_0	Accepted steps	cpu t (s)
7	6	3.30×10^{-4}	–	10^{-3}	10^{-2}	469	0.36
7	12	1.13×10^{-5}	29.25	10^{-4}	10^{-3}	1811	2.15
7	24	1.46×10^{-6}	11.84	10^{-4}	10^{-4}	2896	6.52
9	6	9.20×10^{-6}	–	10^{-4}	10^{-4}	1690	1.26
9	12	6.91×10^{-7}	13.31	10^{-5}	10^{-4}	6388	7.94
9	24	1.20×10^{-8}	57.58	10^{-6}	10^{-4}	23,738	56.40

Table 4 The $\mathrm{Error}(S)_{m,N}$ for the European call option for fixed $m = 10$, $N = 9$, $h_0 = 10^{-2}$ and using different tolerances

Tol	$\mathrm{Error}(S)_{m,N}$	Accepted steps	cpu t (s)
10^{-2}	1.22×10^{-3}	643	0.72
10^{-3}	2.20×10^{-4}	648	0.73
10^{-4}	1.61×10^{-5}	673	0.75
10^{-5}	1.36×10^{-6}	789	0.87
10^{-6}	7.79×10^{-7}	1382	1.56
10^{-7}	5.12×10^{-7}	4076	4.29
10^{-8}	5.11×10^{-7}	12,871	13.82

Table 5 Comparison of variable step size and constant step size for the European call option

Variable step size					
m, N	$\mathrm{Error}(S)_{m,N}$	Total steps	cpu t (s)	Tol	h_0
$m = 4, N = 7$	3.35×10^{-3}	38	0.079	10^{-2}	10^{-2}
$m = 7, N = 11$	2.49×10^{-6}	832	0.74	10^{-5}	10^{-3}
$m = 9, N = 12$	1.44×10^{-8}	3297	3.26	10^{-6}	10^{-4}
Constant step size					
m, N	$\mathrm{Error}_{\mathrm{fix}}$	$\mathrm{steps}_{\mathrm{fix}}$	$\mathrm{cpu}_{\mathrm{fix}}$		
$m = 4, N = 7$	3.36×10^{-3}	500	2.84		
$m = 7, N = 11$	2.53×10^{-6}	5000	3.91		
$m = 9, N = 12$	1.83×10^{-8}	10,000	10.79		

Table 6 shows a comparison of the proposed method and the RBF method of the B–S equation for the European call option at $\tau = T$. In Rad et al. (2015) by using various numbers of interpolated points (Np) in the S-direction and using implicit Euler method for the τ-direction, high accuracy was obtained. However in the proposed method, using lower numbers of interpolate points in the S-direction (number of used points $(m \times N - 2)$) and the P–C method with variable step size for the τ-direction, we get a better accuracy than the previous method.

Table 7 shows a comparison of the proposed method and the superconvergence finite element method of the B–S equation for the European call option at $\tau = T$. In Golbabai et al.

Table 6 The $\text{Error}(S)_{m,N}$ of the proposed method and the RBF method (Rad et al. 2015)

m, N	Accepted steps	$\text{Error}(S)_{m,N}$	Np in Rad et al. (2015)	$\text{Error}(S)_{m,N}$ for Rad et al. (2015)
$m = 5, N = 9$	160	1.37×10^{-4}	50	5.08×10^{-3}
$m = 10, N = 9$	1382	7.79×10^{-7}	100	1.17×10^{-4}
$m = 15, N = 9$	4220	6.81×10^{-8}	150	7.83×10^{-5}

Table 7 The $\text{Error}(S)_{m,N}$ of the proposed method and the superconvergence finite element method (Golbabai et al. 2013)

m, N	$\text{Error}(S)_{m,N}$	m, N in Golbabai et al. (2013)	$\text{Error}(S)_{m,N}$ for Golbabai et al. (2013)
$m = 4, N = 11$	2.61×10^{-5}	$m = 16, N = 3$	2.62×10^{-5}
$m = 8, N = 11$	8.84×10^{-8}	$m = 32, N = 3$	3.68×10^{-7}
$m = 14, N = 11$	1.73×10^{-9}	$m = 64, N = 3$	8.10×10^{-9}

(2013), by changing the variable in the S-direction, the interval $[0, S_{\max}]$ changes to $[0, 1]$, and by dividing this interval to m nodes, they approximate these subintervals by polynomials of degree at least 3 (cubic). For finding coefficients of these polynomials, they use numerical integration such as numerical Gaussian integration (the order of superconvergence finite element method is $O((\frac{1}{m})^{2N})$, where N is the degree of polynomials) or numerical Newton–Cotes integration method (the order of accuracy for finite element method is $O((\frac{1}{m})^{N+2})$, where N is the degree of polynomials). In the proposed method, unlike pervious methods, we approximate the solution of the B–S equation in the S-direction by using interpolate polynomials (Lagrange polynomials) and this choice decreases the cost of calculations. So, we can increase the degree of polynomials. However, unlike pervious methods, by subdividing the interval less $[0, S_{max}]$ and using the polynomials of more than 3 and with less cost of calculations, we get high accuracy. In the τ-direction, the pervious methods used the Crank-Nicholson method. But, in this work using the P–C method, we obtain a better accuracy than pervious methods. In τ-direction, more steps have been used in the previous methods, while in the proposed method, we use different step sizes of the P-C method with control of step size, which control error in estimation in each step and we obtain a non-smooth point (strike price) with less steps. After passing through this point, again the step size becomes great. Finally, by using less steps for the τ-direction, we get a high accuracy.

Figure 1 shows the plot of the maximal error with different $m = N = 10, \text{Tol} = 10^{(-6)}, h_0 = 10^{(-3)}$, $\texttt{Accepted steps} = 1616$ using the proposed method for the European call option. Figure 2 shows the plot of approximate solution $\tilde{V}(S, \tau)$ and exact solution $V(S, \tau)$ of the B–S equation for European call option at $\tau = T$, $m = 5$, $N = 8$, $\texttt{Accepted steps} = 360$, $\text{Tol} = 10^{(-5)}$, $h_0 = 10^{(-2)}$. Figure 3 shows the plot of the approximate solution $\tilde{V}(S, \tau)$ and the exact solution $V(S, \tau)$ of the B–S equation for the up-and-out barrier call option $m = 6$, $N = 8$, $\texttt{Accepted steps} = 519$, $\text{Tol} = 10^{(-3)}$, $h_0 = 10^{(-2)}$.

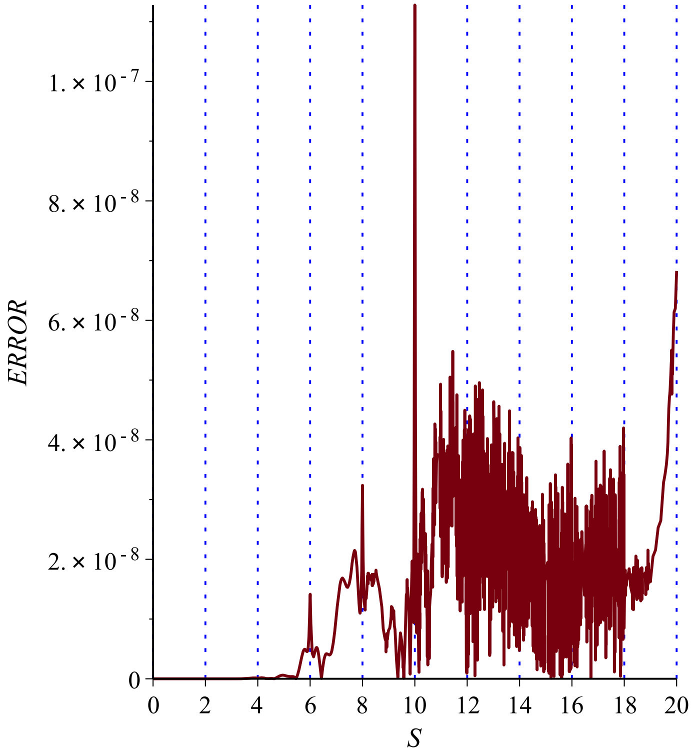

Fig. 1 Plot of Error(S) with $m = N = 10$ for the European call option

6.2 Numerical experiments for the American put option

In this section, we present the results of numerical experiments to show the performance of the proposed method for the American put option. The proposed method is tested against the standard binomial tree method with 1000 time steps. The binomial method was first presented by Cox et al. (1979). This is a useful and very popular technique for pricing an American option, where no closed-form solution exists. This method is a diagram representing different possible paths that might be followed by the stock price over the life of an option (for more details see Hull 2000).

In our numerical experiments, we have used $\sigma = 0.3$, $r = 0.1$, $T = 1$, $E = 1$. Moreover, we choose the parameter $C = rE$ and $\varepsilon = 10^{-4}$ required for the penalty method.

In the American put option if $V(S, t) < E - S$, then one could buy a put option and immediately exercise the option, making a risk-free benefit of $E - S - V(S, t) > 0$. To prevent arbitrage opportunity for the American put option at each step [see Eq. (2.12)], we

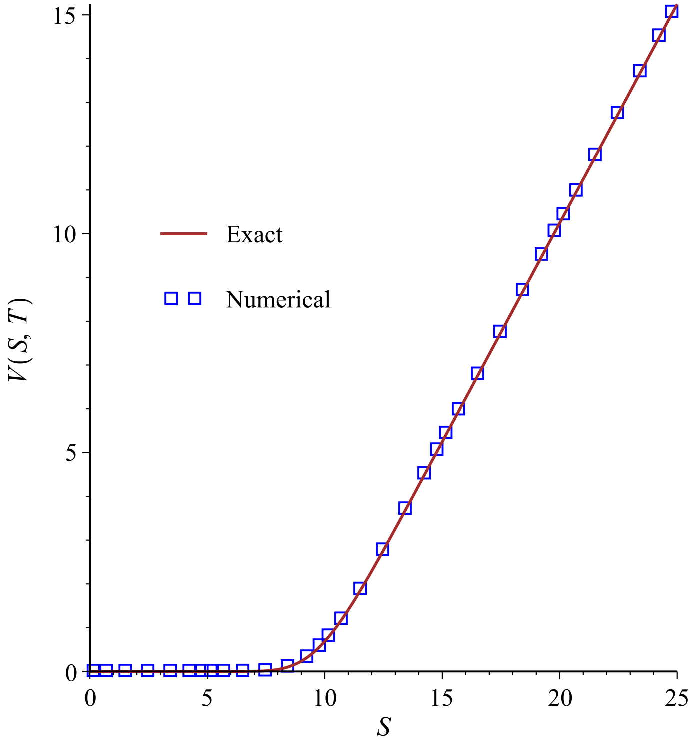

Fig. 2 Plot of $\tilde{V}(S, T)$ and $V(S, T)$ with $m = 5$ and $N = 8$ for the European call option

should update the predictor $U_{n+1}^{(1)}$ and the corrector $U_{n+1}^{(\mu+1)}$, $\mu = 1, \ldots, M$ in Eq. (4.25) as follows:

$$U_{n+1}^{(j)}[i, 1] = \max(U_{n+1}^{(j)}[i, 1], U_0[i, 1]), \quad j = 1, \ldots, M + 1, \quad i = 1, \ldots, m \times N, \tag{6.4}$$

where $U_{n+1}^{(j)}[i, 1]$ is the i-th element of matrix $U_{n+1}^{(j)}$. Also, $U_0[i, 1]$ is the i-th element of matrix U_0.

For all tables `Accepted steps` is the number of accepted steps, `Total steps` is the total number of steps used and h_0 is the initial step size.

Tables 8 shows the maximal error for the American put option with $N = 8$ and different values of m. It is clear that the maximal error is decreased by increasing m.

Tables 9 show the maximal error for the American put option with $m = 16$ and different values of N.

Table 10 shows that in the penalty method, the maximal error decreased by decreasing ϵ.

Table 11 shows the efficiency of the variable step size rather than the constant step size. This table shows the error of the proposed method and constant step size method, and the number of steps and cpu time for constant step size case by $\text{Error}_{\text{fix}}$, $\text{steps}_{\text{fix}}$ and cpu_{fix}.

In Fig. 4, we have plotted the price of the American put option with $m = 7$ and $N = 11$.

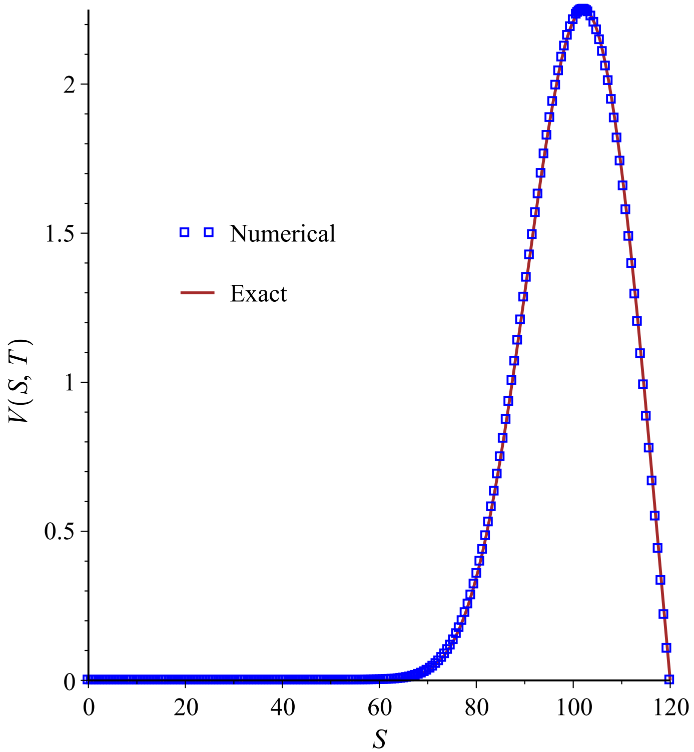

Fig. 3 Plot of $\tilde{V}(S,T)$ and $V(S,T)$ with $m=6$ and $N=8$ for barrier call option

Table 8 The error for the American put option for fixed $N=8$ and different values of m nodes

m	$\mathrm{Error}(S)_{m,N}$	Tol	h_0	Accepted steps	cpu t (s)
3	5.87×10^{-2}	10^{-2}	10^{-2}	71	0.16
4	4.91×10^{-2}	10^{-3}	10^{-2}	148	0.26
5	2.49×10^{-2}	10^{-3}	10^{-3}	294	0.69
7	3.32×10^{-3}	10^{-4}	10^{-3}	742	1.81
8	5.11×10^{-4}	10^{-5}	10^{-3}	913	3.13
10	7.85×10^{-5}	10^{-6}	10^{-4}	1907	4.04

Table 9 The error for the American put option for fixed $m=16$ and different values of N

N	$\mathrm{Error}(S)_{m,N}$	Tol	h_0	Accepted steps	cpu t (s)
2	3.29×10^{-2}	10^{-3}	10^{-2}	32	0.17
4	6.32×10^{-3}	10^{-4}	10^{-2}	356	1.61
6	7.68×10^{-4}	10^{-6}	10^{-3}	1743	3.65
7	7.19×10^{-4}	10^{-5}	10^{-3}	2648	15.79
10	2.43×10^{-5}	10^{-7}	10^{-5}	11,632	25.39

Table 10 The error for the American put option for fixed $m = 8$, $N = 9$, $h_0 = 10^{-3}$, Tol $= 10^{-5}$ and different values of ϵ

ϵ	$\mathrm{Error}(S)_{m,N}$	`Accepted steps`	cpu t (s)
10^{-1}	4.01×10^{-1}	1888	6.42
10^{-2}	1.29×10^{-2}	1655	5.07
10^{-3}	2.72×10^{-3}	1548	4.95
10^{-4}	4.97×10^{-4}	1547	5.01
$10^{-4.2}$	4.67×10^{-4}	1548	4.93

Table 11 Comparison of variable step size and constant step size for the American put option

Variable step size					
m, N	$\mathrm{Error}(S)_{m,N}$	`Total steps`	cpu t (s)	Tol	h_0
$m = 6, N = 8$	1.282×10^{-2}	810	0.99	10^{-5}	10^{-1}
$m = 8, N = 9$	5.069×10^{-4}	2261	2.89	10^{-6}	10^{-6}
Constant step size					
m, N	$\mathrm{Error}_{\mathrm{fix}}$		$\mathrm{steps}_{\mathrm{fix}}$		$\mathrm{cpu}_{\mathrm{fix}}$
$m = 6, N = 8$	1.288×10^{-2}		5000		4.18
$m = 8, N = 9$	5.072×10^{-4}		10,000		11.61

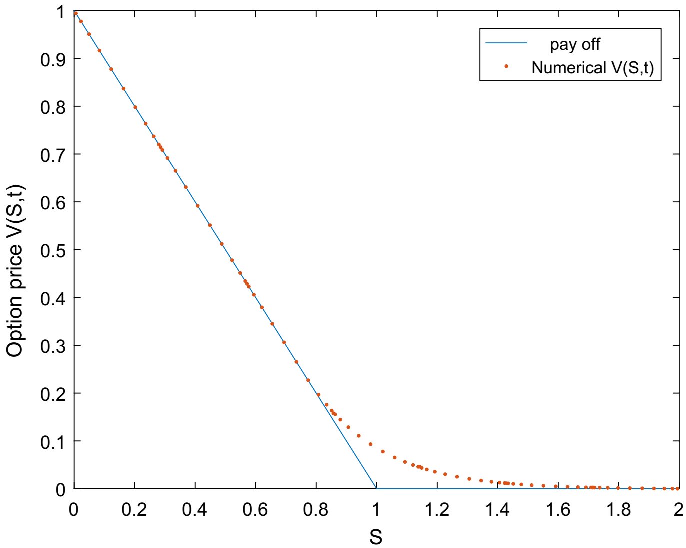

Fig. 4 Plot of $V(S, T)$ with $m = 7$ and $N = 11$ for the American put option

7 Conclusion

In this paper, we presented a numerical scheme for solving the Black–Scholes equation. The method of domain decomposition algorithm based on Chebyshev polynomials in spatial direction was employed. Also, in time direction the predictor–corrector method with variable step size was applied. The obtained results showed that the new approach can solve the problem effectively.

References

Amata S, Legaz MJ, Pedregalc P (2015) A variable step-size implementation of a variational method for stiff differential equations. Math Comput Simul 118:49–57

Ballestra LV, Cecere L (2016) A numerical method to estimate the parameters of the CEV model implied by American option prices: evidence from NYSE. Chaos, Solitons Fractals 88:100–106

Barles G, Soner HM (1998) Option pricing with transaction costs and a nonlinear Black–Scholes equation. Financ Stoch 2:369–397

Bastani A, Ahmadi Z, Damircheli D (2013) A radial basis collocation method for pricing American options under regime-switching jump-diffusion models. Appl Numer Math 65:79–90

Black F, Scholes M (1973) The pricing of options and corporate liabilities. J Polit Econ 81:637–659

Chen W, Wang S (2014) A penalty method for a fractional order parabolic variational inequality governing American put option valuation. Comput Math Appl 67:77–90

Chen W, Xu X, Zhu S (2015) A predictor-corrector approach for pricing American options under the finite moment log-stable model. Appl Numer Math 97:15–29

Company R, Navarro E, Pintos JR, Ponsoda E (2008) Numerical solution of linear and nonlinear Black–Scholes option pricing equations. Comput Math Appl. 56:813–821

Cox J, Ross S, Rubinstein M (1979) Option pricing: a simplified approach. J Financ Econ 7:229–264

Duffy DJ (2006) Finite difference methods in financial engineering a partial differential equation approach. Wiley, Chichester

Dupire B (1994) Pricing with a smile. Risk 7(1):18–20

Golbabai A, Ballestra LV, Ahmadian D (2013) Superconvergence of the finite element solutions of the Black–Scholes equation. Financ Res Lett 10:17–26

Haug EG (2007) The complete guide to option pricing formulas. McGraw-Hill Companies, New York

in't Hout KJ, Volders K (2009) Stability of central finite difference schemes on non-uniform grids for the Black–Scholes equation. Appl Numer Math 59:2593–2609

Huang SJY (2005) Implementation of general linear methods for stiff ordinary differential equations. PhD thesis, Department of Mathematics, Auckland University

Hull JC (2000) Options, futures and other derivatives. Prentice Hall, Englewood Cliffs

Hurd TR, Zhou Z (2010) A fourier transform method for spread option pricing. SIAM J Financ Math 1:142–157

Izzo G, Jackiewicz Z (2017) Highly stable implicit-explicit Runge–Kutta methods. Appl Numer Math 113:71–92

Javidi M (2006) Spectral collocation method for the solution of the generalized Burger–Fisher equation. Appl Math Comput 174(1):345–352

Javidi M, Golbabai A (2009) A new domain decomposition algorithm for generalized Burger's-Huxley equation based on Chebyshev polynomials and preconditioning. Chaos, Solitons Fractals 39:849–857

Javidi M (2011) A modified Chebyshev pseudospectral DD algorithm for the GBH equation. Comput Math Appl 62:3366–3377

Jiang L (2004) Mathematical modeling and methods of option pricing. World Scientific, Tongji University, Shanghai

Kalantari R, Shahmorad S, Ahmadian D (2016) The stability analysis of predictor-corrector method in solving American option pricing model. Comput Econ 47:255–274

Khaliq AQM, Voss DA, Kazmi K (2008) Adaptive θ-methods for pricing American options. J Comput Appl Math 222:210–227

Khaliq AQM, Voss DA, Kazmi K (2006) A linearly implicit predictor-corrector scheme for pricing American options using apenalty method approach. J Bank Financ 30:459–502

Khater AH, Shamardan AB, Callebaut DK, Ibrahim RS (2000) Chebyshev spectral collocation methods for nonlinear isothermal magnetostatic atmospheres. J Comput Appl Math 115:309–329

Markolefas S (2008) Standard galerkin formulation with high order lagrange finite elements for option markets pricing. Appl Math Comput 195:707–720
Martin-Vaquero J, Khaliq AQM, Kleefeld B (2014) Stabilized explicit Runge-Kutta methods for multi-asset American options. Comput Math Appl 67:1293–1308
Mashayekhi S, Hugger J (2015) Finite difference schemes for a nonlinear Black-Scholes model with transaction cost and volatility risk. Acta Math Univ Comenianae 2:255–266
Mehrkanoon S, Majid ZA, Suleiman M (2010) A variable step implicit block multistep method for solving first-order ODEs. J Comput Appl Math 233:2387–2394
Mohammadi R (2015) Quintic B-spline collocation approach for solving generalized Black–Scholes equation governing option pricing. Comput Math Appl 69:777–797
Nielsen B, Skavhaug O, Tveito A (2008) Penalty methods for the numerical solution of American multi-asset option problems. J Comput Appl Math 222:3–16
Rad JA, parand K, ballestra LV (2015) Pricing European American options by radial basis points interpolation. Appl Math Comput 251:363–377
Ramos H, Singhc G (2017) A note on variable step-size formulation of a Simpson's-type second derivative block method for solving stiff systems. Appl Math Lett 64:101–107
Sevcovic D (2008) Transformation methods for evaluating approximations to the optimal exercise boundary for linear and nonlinear Black–Sholes Equations. In: Ehrhard M (ed) Nonlinear models in mathematical finance: new research trends in optimal pricing. Nova Science Publishers, New York, pp 153–198
Shcherbakov V, Larsson E (2016) Radial basis function partition of unity methods for pricing vanilla basket options. Comput Math Appl 71:185–200
Tagliani A, Fusai G, Sanfelici S (2001) Practical problems in the numerical solutions of PDE's in finance. Rendiconti peer gli Studi Economici Quantitativi 2002:105–132
Thai N, Wu X, Na J, Guo Y, Tin NT, Le PhX (2017) Adaptive variable step-size neural controller for nonlinear feedback active noise control systems. Appl Acoust 116:337–347
Tian H, Yu Q, Jin C (2011) Continuous block implicit hybrid one-step methods for ordinary and delay differential equations. Appl Numer Math 61:1289–1300
Wilmott P, Dewynne J, Howison S (1993) Option pricing mathematical models and compuation. Oxford Financial Press, Oxford
Wade BA, Khaliq AQM, Yousuf M, Vigo-Aguiar J, Deininger R (2007) On smoothing of the Crank–Nicolson scheme and higher order schemes for pricing barrier options. J Comput Appl Math 204:144–158
Zhang K, Song H, Li J (2014) Front-fixing FEMs for the pricing of American options based on a PML technique. Appl Anal 94:903–931
Zhang K, Wang S (2012) Pricing American bond options using a penalty method. Automatica 48:472–479
Zhang K, Wang S, Yang XQ, Teo KL (2009) A power penalty approach to numerical solutions of two-asset American options. Numer Math Theory, Methods Appl 2:202–223
Zhang K, Song H, Li J (2015) Front-fixing FEMs for the pricing of American options based on a PML technique. Appl Anal 94:903–931
Zhao J, Davison M, Corless RM (2007) Compact finite difference method for American option pricing. J Comput Appl Math 206:306–321
Zhu SP, Zhang J (2011) A new predictor-corrector scheme for valuating American puts. Appl Math Comput 27:4439–4452