Computational Economics (2026) 67:2263–2276
https://doi.org/10.1007/s10614-025-10945-w

A Convergent Fourth-Order Finite Difference Scheme for the Black–Scholes Equation

Seungyoon Kang[1] · Soobin Kwak[1] · Gyeonggyu Lee[1,2] · Yougjin Hwang[1] · Seokjun Ham[1] · Junseok Kim[1]

Accepted: 30 March 2025 / Published online: 24 April 2025

Abstract
We present a convergent fourth-order finite difference method (FDM) for the Black–Scholes (BS) equation. The proposed numerical scheme is constructed based on an implicit Euler method and a fourth-order accurate FDM. Many options used in the financial market have payoff functions with discontinuous differentials. Because the second and first derivatives in the BS equation cause high errors and low convergence rates, we propose the numerical method to overcome this difficulty by applying a fine mesh at the early stage of the numerical method. For a sufficiently smooth option price function, a coarse mesh is employed. We can achieve a fourth-order convergence rate applying the proposed algorithm. The effectiveness and capability of the proposed algorithm are validated through various computational results using a European call option.

Keywords Black–Scholes equation · Fourth-order scheme · European call option

✉ Junseok Kim
cfdkim@korea.ac.kr

Seungyoon Kang
heroe2401@korea.ac.kr

Soobin Kwak
soobin23@korea.ac.kr

Gyeonggyu Lee
ssoss787@nims.re.kr

Yougjin Hwang
youngjin_hwang@korea.ac.kr

Seokjun Ham
seokjun@korea.ac.kr

[1] Department of Mathematics, Korea University, Seoul 02841, Republic of Korea

[2] Mathematical Modelling, National Institute for Mathematical Sciences, Daejeon 34047, Republic of Korea

1 Introduction

Options are among the most well-known financial derivatives. They convey to the owner the right to buy or sell a specific amount of an underlying asset at a strike price. The right to sell is a put option, while the right to buy is a call option. Depending on when the options can be exercised, an American option can be exercised at any time, whereas a European option can only be exercised at specific expiration. In this study, we focus on the European option.

Beginning with the seminal papers by Black and Scholes (1973) and Merton (1973), the Black–Scholes (BS) equation marked a major breakthrough in option pricing. The classical one-dimensional BS equation is given as follows:

$$\frac{\partial u}{\partial t} = -\frac{1}{2}(\sigma x)^2\frac{\partial^2 u}{\partial x^2} - rx\frac{\partial u}{\partial x} + ru, \text{ for } (x,t) \in [0,\infty) \times [0,T], \tag{1}$$

where t represents time, T is the expiry time, x is the price of the underlying asset, and $u(x, t)$ is the option price. Constants σ and r represent volatility and the risk-free rate, respectively. Here, the risk-free rate is the theoretical rate of return on options with zero risk. We introduce a new time variable: $\tau = T - t$. Equation (1) can thus be rewritten as follows:

$$\frac{\partial u}{\partial \tau} = \frac{1}{2}(\sigma x)^2\frac{\partial^2 u}{\partial x^2} + rx\frac{\partial u}{\partial x} - ru, \ (x,\tau) \in [0,\infty) \times [0,T]. \tag{2}$$

The payoff function, which is the initial condition of the governing equation, is

$$u(x,0) = p(x), \ x \in [0,\infty). \tag{3}$$

The BS equation is among the most extensively used models in option pricing. An example is the work of Gulen et al. (2019), where the authors presented a high-order approach to the BS equation. The proposed method combines a sixth-order FDM in space with a third-order Runge–Kutta method in time. State-of-the-art methods such as artificial neural networks (Eskiizmirliler et al., 2021) are also used for the numerical solution of this model. Equation (2) models a single asset, while the multi-dimensional BS equation governs problems involving multiple assets. Common methods for multi-dimensional partial differential equations, such as the operator splitting method (Cho et al., 2023) can be applied to solve multi-dimensional BS equations. Monte Carlo simulation (Hwang et al., 2023; Kim et al., 2024) is also popular method in financial analysis (Lux, 2022). Time-fractional BS equations have also been proposed to model numerous anomalous processes (She et al., 2021; Zhang & Zheng, 2022; Tian et al., 2021; Rezaei et al., 2021).

The European call option is of great interest in both academic and industrial research, with numerous reports on this option model. Figlewski and Gao (1999) introduced an adaptive mesh model that greatly reduces nonlinearity error. Their option test set included European puts with different conditions. Lötstedt et al. (2007) discretized the multi-dimensional BS equation to numerically solve the European call option. Authors introduced a dynamic adaptive grid method to reduce discretization errors. A stable and fast numerical scheme for the BS equation that does not require a far-field boundary condition was proposed by Lee et al. (2023). Numerical simulations for the European option showed stable results with sufficiently large time steps. Various tests showed the efficiency and stability of the proposed scheme. While the European option model is considered as a popular model to study, numerous researchers have also developed efficient numerical methods for other option models, such as the American model (Kalantari & Shahmorad, 2019; Alrabeei & Yousuf, 2020; Gan et al., 2020).

Numerous studies have proposed fourth-order schemes to solve the BS equation. Deswal and Kumar (2022) used an orthogonal spline and Rannacher time-marching scheme to solve the non-smooth terminal condition problem. Numerical experiments were conducted using the proposed method, and the proposed method showed second-order and fourth-order accuracies in time and space, respectively. Roul and Goura (2020) used the Crank–Nicolson method and a high-order compact finite difference method (HOCFDM) to construct a higher order numerical method. In addition, they numerically analyzed convergence rates. Numerical results also showed consistent convergence rates. Patel and Mehra (2020) developed a fourth-order compact scheme derived from the Riemann–Liouville spatial fractional derivative. They numerically verified the stability and accuracy of the developed system against the advection–diffusion equation.

The main contribution of this study is to present a fast, fourth-order finite difference scheme for the BS equation, supported by a startup procedure that enhances the efficiency of the proposed method. A fourth-order finite difference approximation using Taylor expansion is suggested, where derivatives at each point depend on the corresponding point and four neighboring points. Various barrier options, including the European call option, have payoff functions that are discontinuous or non-differentiable at certain points, which can be a challenging problem for Taylor expansion. Herein, a startup procedure is introduced to overcome errors generated at non-differentiable points of the payoff function. Applying a simple adaptive computational mesh at the beginning of the numerical simulation can enhance the overall efficiency and correct the convergence order. Numerical simulations suggest appropriate duration times and mesh refinement for the startup procedure.

The outline of this paper is as follows. In Sect. 2, we describe the convergent fourth-order finite difference scheme for the BS equation and the startup procedure for the proposed method. Next, in Sect. 3, we consider a benchmark problem and perform various numerical experiments to demonstrate the performance of the proposed method. Lastly, in Sect. 4, concluding remarks are given.

2 Numerical Solution

2.1 Fourth-Order Finite Difference Method

Let $\Omega = [0, S_{max}]$ be the computational domain for the underlying asset in the one-dimensional BS equation and T be the expiry date. To solve this numerically, we apply a finite difference method (Li & Tourin, 2022). We first divide Ω using a uniform discrete mesh. Let h and τ be the spatial step size and time step, respectively. Correspondingly, $N_x = S_{max}/h$ and $N_t = T/\tau$ are the number of grid points in the computational domain and time interval, respectively. Therefore, we have grid points $x_0 = 0$, $x_{i+1} = x_i + h$ for $i = 0, \dots, N_x - 1$, and $x_{N_x} = S_{max}$ for the computational domain. Additional points $x_{-1} = x_0 - h$, $x_{N_x+1} = x_{N_x} + h$ and $x_{N_x+2} = x_{N_x} + 2h$ are assigned to satisfy boundary conditions. See Fig. 1 for a schematic illustration of the discretized computational domain.

In the same way, the time interval is discretized using a uniform grid $t_n = n\Delta\tau$ for $n = 0, \dots, N_t$. Therefore, the numerical solution at price x_i and time $t = n\Delta\tau$ is denoted by $u_i^n \approx u(x_i, n\Delta\tau)$ (Morton & Mayers, 2005). An iterative method will be applied where the solutions at time t_{n+1} are calculated with the former solution t_n. Using the aforementioned notations, we apply an implicit method on Eq. (2):

$$\frac{u_i^{n+1} - u_i^n}{\Delta\tau} = \frac{\sigma^2 x_i^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_i^{n+1} + r x_i \left(\frac{\partial u}{\partial x}\right)_i^{n+1} - r u_i^{n+1}, \tag{4}$$

where the discrete second derivative is defined as

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i^{n+1} = \frac{-u_{i-2}^{n+1} + 16u_{i-1}^{n+1} - 30u_i^{n+1} + 16u_{i+1}^{n+1} - u_{i+2}^{n+1}}{12h^2}. \tag{5}$$

We can see that the next time approximation u^{n+1} is evaluated using the solutions from t^{n+1} and t^n, making an implicit Euler method. Herein, we discuss the convergence of the proposed numerical scheme. The convergence rate is the characterization of how quickly the numerical method approaches the desired solution. In this study, we seek the numerical solution at time T, therefore we discuss the size of error at time T, due to spatial step size h. In an analytical point of view, the order of h in the error represents the order of the proposed method. The numerical calculation of the convergence rate will be introduced in the computational results section. Note that the discrete second derivative in Eq. (5) is fourth-order accurate (Anderson et al., 2013). Similarly, we do the following Taylor series calculation to achieve fourth-order accuracy in the discrete first derivative.

For $i = 1, 2, \dots, N_x$,

$$u_{i-2}^{n+1} = \sum_{k=0}^{\infty} \frac{(-2h)^k}{k!}\left(\frac{\partial^k u}{\partial x^k}\right)_i^{n+1}, \quad u_{i-1}^{n+1} = \sum_{k=0}^{\infty} \frac{(-h)^k}{k!}\left(\frac{\partial^k u}{\partial x^k}\right)_i^{n+1},$$
$$u_{i+1}^{n+1} = \sum_{k=0}^{\infty} \frac{h^k}{k!}\left(\frac{\partial^k u}{\partial x^k}\right)_i^{n+1}, \quad \text{and} \quad u_{i+2}^{n+1} = \sum_{k=0}^{\infty} \frac{(2h)^k}{k!}\left(\frac{\partial^k u}{\partial x^k}\right)_i^{n+1}.$$

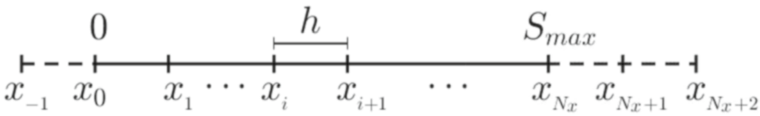

Fig. 1 Uniform grid for the underlying asset with step size h

Subtract u_{i-2}^{n+1} from u_{i+2}^{n+1} and u_{i-1}^{n+1} from u_{i+1}^{n+1}.

$$-u_{i-2}^{n+1} + u_{i+2}^{n+1} = 4h\left(\frac{\partial u}{\partial x}\right)_i^{n+1} + \frac{8h^3}{3}\left(\frac{\partial^3 u}{\partial x^3}\right)_i^{n+1} + \frac{8h^5}{15}\left(\frac{\partial^5 u}{\partial x^5}\right)_i^{n+1} + \cdots, \tag{6}$$

$$-u_{i-1}^{n+1} + u_{i+1}^{n+1} = 2h\left(\frac{\partial u}{\partial x}\right)_i^{n+1} + \frac{h^3}{3}\left(\frac{\partial^3 u}{\partial x^3}\right)_i^{n+1} + \frac{h^5}{60}\left(\frac{\partial^5 u}{\partial x^5}\right)_i^{n+1} + \cdots. \tag{7}$$

Multiply Eq. (7) by 8 and subtract Eq. (6).

$$u_{i-2}^{n+1} - 8u_{i-1}^{n+1} + 8u_{i+1}^{n+1} - u_{i+2}^{n+1} = 12h\left(\frac{\partial u}{\partial x}\right)_i^{n+1} - \frac{2h^5}{5}\left(\frac{\partial^5 u}{\partial x^5}\right)_i^{n+1} + \cdots, \tag{8}$$

$$\left(\frac{\partial u}{\partial x}\right)_i^{n+1} = \frac{-u_{i-2}^{n+1} + 8u_{i-1}^{n+1} - 8u_{i+1}^{n+1} + u_{i+2}^{n+1}}{12h} + h^4\left[\frac{1}{30}\left(\frac{\partial^5 u}{\partial x^5}\right)_i^{n+1} + \cdots \right. \tag{9}$$

$$= \frac{-u_{i-2}^{n+1} + 8u_{i-1}^{n+1} - 8u_{i+1}^{n+1} + u_{i+2}^{n+1}}{12h} + O(h^4). \tag{10}$$

Applying Eqs. (5) and (10) on Eq. (4), we achieve fourth-order accuracy for both the first and second derivatives, with respect to the underlying asset. Therefore, the proposed scheme is fourth-order accurate. Then, we rewrite Eq. (4) in the form of

$$a_i u_{i-2}^{n+1} + b_i u_{i-1}^{n+1} + c_i u_i^{n+1} + d_i u_{i+1}^{n+1} + e_i u_{i+2}^{n+1} = f_i^n, \tag{11}$$

where

$$a_i = \frac{\sigma^2 x_i^2}{24h^2} - \frac{r x_i}{12h}, \quad b_i = -\frac{16\sigma^2 x_i^2}{24h^2} + \frac{8r x_i}{12h}, \quad c_i = \frac{30\sigma^2 x_i^2}{24h^2} + r + \frac{1}{\Delta\tau},$$
$$d_i = -\frac{16\sigma^2 x_i^2}{24h^2} - \frac{8r x_i}{12h}, \quad e_i = \frac{\sigma^2 x_i^2}{24h^2} + \frac{r x_i}{12h}, \quad f_i^n = \frac{u_i^n}{\Delta\tau}.$$

Different boundary conditions are imposed at $x = 0$ and $x = S_{\max}$. A homogeneous Dirichlet boundary condition is used at $x = 0$, whereas a linear boundary condition is used at $x = S_{\max}$ (Persson & von Persson, 2007; Tavella & Randall, 2000; Windcliff et al., 2004). According to the homogeneous Dirichlet boundary condition, $u_0^{n+1} = 0$ and the ghost point u_{-1}^{n+1} satisfies $u_{-1}^{n+1} = 2u_0^{n+1} - u_1^{n+1} = -u_1^{n+1}$ for all n. Therefore, Eq. (11) with $i = 1$ and $i = 2$ is calculated in the following way:

$$\begin{aligned} (c_1 - a_1)u_1^{n+1} + d_1 u_2^{n+1} + e_1 u_3^{n+1} &= f_1^n, \\ b_2 u_1^{n+1} + c_2 u_2^{n+1} + d_2 u_3^{n+1} + e_2 u_4^{n+1} &= f_2^n. \end{aligned} \tag{12}$$

The linear boundary condition is defined as

$$\frac{\partial^2 u}{\partial x^2}(S_{\max}, \tau) = 0, \text{ for } \tau \in [0, T]. \tag{13}$$

From the discretization of Eq. (13) at $i = N_x$, we can obtain as follows:

$$\frac{-u_{N_x-2}^{n+1} + 16u_{N_x-1}^{n+1} - 30u_{N_x}^{n+1} + 16u_{N_x+1}^{n+1} - u_{N_x+2}^{n+1}}{12h^2} = 0.$$

Moreover, we can obtain the relation $u_{N_x+1}^{n+1} = 2u_{N_x}^{n+1} - u_{N_x-1}^{n+1}$ and $u_{N_x+2}^{n+1} = 2u_{N_x}^{n+1} - u_{N_x-2}^{n+1}$ using the boundary condition. Applying these relations, Eq. (11) at $i = N_x$ and $i = N_x - 1$ are given as follows:

$$\begin{aligned}
(a_{N_x} - e_{N_x})u_{N_x-2}^{n+1} + (b_{N_x} - d_{N_x})u_{N_x-1}^{n+1} + (c_{N_x} + 2d_{N_x} + 2e_{N_x})u_{N_x}^{n+1} &= f_{N_x}^n, \\
a_{N_x-1}u_{N_x-3}^{n+1} + b_{N_x-1}u_{N_x-2}^{n+1} + (c_{N_x-1} - e_{N_x-1})u_{N_x-1}^{n+1} + (d_{N_x-1} + 2e_{N_x-1})u_{N_x}^{n+1} &= f_{N_x-1}^n.
\end{aligned}$$

Implementing all the above-mentioned boundary conditions, we rewrite the linear system Eq. (11) in a penta-diagonal matrix form:

$$A\mathbf{u}^{n+1} = \mathbf{f}^n, \tag{14}$$

where

$$A = \begin{pmatrix}
c_1 - a_1 & d_1 & e_1 & 0 & 0 & \dots & 0 \\
b_2 & c_2 & d_2 & e_2 & 0 & \dots & 0 \\
a_3 & b_3 & c_3 & d_3 & e_3 & \dots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \dots & a_{N_x-2} & b_{N_x-2} & c_{N_x-2} & d_{N_x-2} & e_{N_x-2} \\
0 & \dots & 0 & a_{N_x-1} & b_{N_x-1} & c_{N_x-1} - e_{N_x-1} & d_{N_x-1} + 2e_{N_x-1} \\
0 & \dots & 0 & 0 & a_{N_x} - e_{N_x} & b_{N_x} - d_{N_x} & c_{N_x} + 2d_{N_x} + 2e_{N_x}
\end{pmatrix},$$

$$\mathbf{u}^{n+1} = \left(u_1^{n+1} \; u_2^{n+1} \; \cdots \; u_{N_x-1}^{n+1} \; u_{N_x}^{n+1}\right)^T, \quad \text{and} \quad \mathbf{f}^n = \left(f_1^n \; f_2^n \; f_3^n \; \cdots \; f_{N_x-2}^n \; f_{N_x-1}^n \; f_{N_x}^n\right)^T.$$

Solving the linear system of the penta-diagonal matrix, we can numerically solve the one-dimensional BS equation. We note that we can directly solve the penta-diagonal matrix, similar to the tri-diagonal matrix solver (Kwak et al., 2023).

2.2 Startup Procedure

The one-dimensional European call option is considered to measure the convergence rate. Note that the payoff function of the European call option $u(x, 0) = \max(x - K, 0)$ is of class C^0 but not of class C^1. However, the analytical solution (16) is of class C^∞. This implies that the Taylor series expansion is hard to adopt and therefore extra effort must be applied to achieve fourth-order accuracy. We overcome this problem by applying a relatively finer mesh in the early stage of the numerical simulation to obtain smooth convergence. We call this process a startup procedure. The numerical result of the startup procedure is used as the initial condition for the subsequent part of the method.

Let x_i be the original uniform mesh for $i = 0, 1, \dots, N_x$ and let p and q be positive integers representing the numbers of mesh refinements and duration time steps, respectively. During the startup procedure, a p times refined uniform mesh x'_j, for $j = 0, \dots pN_x$ is applied. We assume that h is the mesh size for x, then h/p is the mesh size for x'. After the startup procedure, each x_i corresponds to x'_{pi} for $i = 0, 1, \dots, N_x$. Figure 2 is a schematic illustration of the mesh during and after the startup procedure where the mesh is refined by p times until time $t = q\Delta\tau$. Black dots represent x_i, for $i = 0, 1, \dots, N_x$ and white dots represent the complement of x' with respect to x. Many other financial market problems, such as the cash-or-nothing option also have C^0 or C^1 class payoff function, as well as the European call option (Mehdizadeh Khalsaraei et al., 2021). Hitherto, there has been prior fourth- or high-order finite difference methods for the BS equation concerning the non-differentiable points of the payoff function. One approach is reported by Hu and Gan (2018) where a new coordinate system and coordinate transformation is introduced accompanied by a refined grid that distributes more points near the non-differentiable point. However, this method requires negligible computational cost for coordinate transformation and grid refinement is maintained throughout the entire computation. Another work worth mentioning is done by Voss et al. (2003) where the authors applied Padé approximations to approximate exponential functions. However, due to a lack of effort made at non-differentiable points, the proposed method failed to achieve the optimal fourth-order convergence (Wade et al., 2007). The main contribution of this paper lies in achieving the desired convergence rate through a simple startup procedure applied only at the very early stage of the numerical computation. Therefore, this startup procedure can be applied to resolve accuracy and convergence problems in many other financial option models.

3 Computational Experiments

Now, computational tests are conducted to validate the effectiveness of the proposed method. First, we present the benchmark problem along with its exact solution. Using this benchmark problem, convergence rates with and without the startup procedure are given. Then, results with various p and q values are presented to propose appropriate parameter values.

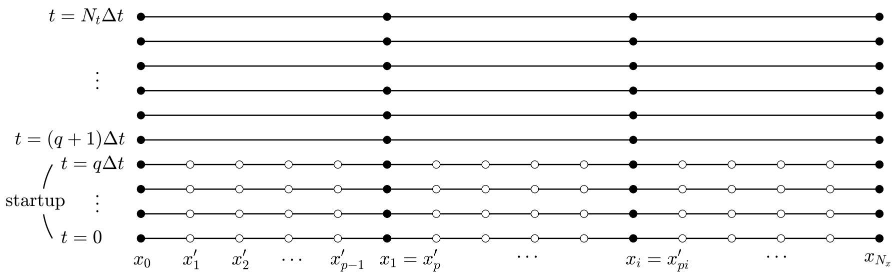

Fig. 2 Schematic illustration of mesh during and after startup procedure

3.1 The Benchmark Problem

The European call option is highly used in the financial market and can be used as a building block for constructing complex option products (Vollert, 2001). Thus, we consider the European call option as a benchmark problem for numerical examples. From a numerical analysis perspective, an analytical solution exists for the European call option. Consequently, it is suitable for numerical experiments. In this study, we concentrate on options with a single asset, which is governed by the one-dimensional BS equation. The payoff function for a single asset European call option is

$$u(x,0) = \max(x-K,0),\ x \in [0, S_{max}], \tag{15}$$

where K is the strike price. When parameter values are given, we obtain the exact analytical solution of the BS equation (Black & Scholes, 1973):

$$\begin{aligned} &u(x,\tau) = xN(d_1) - Ke^{-r\tau}N(d_2),\ (x,\tau) \in [0, S_{max}] \times [0,T], \\ &\text{where } d_1 = \frac{\ln(x/K) + \left(r + \frac{1}{2}\sigma^2\right)\tau}{\sigma\sqrt{\tau}},\ d_2 = d_1 - \sigma\sqrt{\tau}, \\ &N(d) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{d} \exp\left(-\frac{x^2}{2}\right) dx. \end{aligned} \tag{16}$$

Here, $N(d)$ is the cumulative distribution function of the standard normal distribution.

Figure 3(a) illustrates the payoff function for a one-asset European call option when the strike price $K = 100$ and $S_{max} = 300$. With the startup procedure to the previously described payoff function, we obtain a smooth function, as shown in Fig. 3(b).

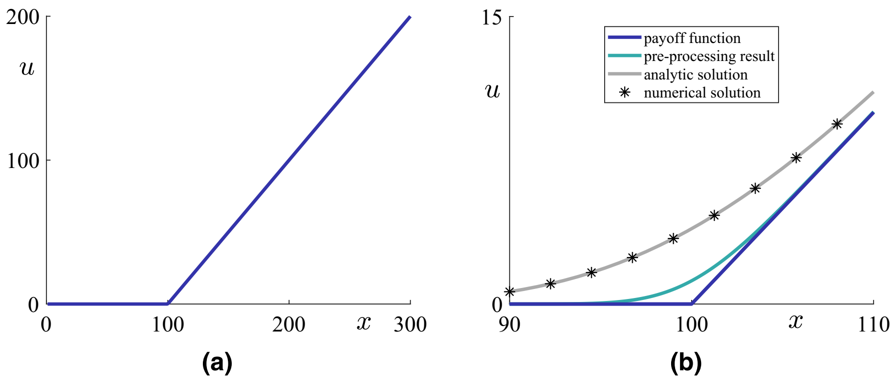

Fig. 3 Example of European call option. (a) payoff function. (b) smooth function with startup procedure

3.2 Convergence Rate with and Without Startup Procedure

We perform a series of numerical experiments to understand the effectiveness of the startup procedure. Tests are conducted using a uniform adaptive mesh. A uniform fine mesh is applied during the startup procedure and a coarse uniform mesh in subsequent calculations. We tested the European call option problem with $S_{max} = 1000,\ \sigma = 0.3,\ r = 0.03,\ K = 100,\ S_0 = 100,\ T = 0.01$ and $\Delta\tau = 10^{-7}$. Recall that p denotes the extent of mesh refinement in the startup procedure and q is the number of time steps for which the startup procedure proceeds. Let h be the grid size, then $p = 6$ times finer grid size $h/6$ is used for $q = 3000$ time steps of the startup procedure. The startup procedure proceeds until $t = q\Delta\tau = 0.003$. The main concern in option pricing is the option price at S_0. Therefore the error is calculated as $e_h = u(x_s) - u^{ex}(x_s)$ where $x_s = S_0$ and u^{ex} is the exact analytical solution. We use three different mesh sizes, $h = 10/2^i$ for $i = 0,\ 1,\ 2$. The convergence rate is calculated as the ratio of the error between refined meshes: $\log_2(||e_h||/||e_{h/2}||)$. Results with and without the startup procedure are shown in Table 1. It can be seen that the error is small and fourth-order convergent with the startup procedure, whereas the error is larger and only second-order convergent without the startup procedure. Additionally, in financial markets, the tick size for option trading is in cents. Therefore, regarding the tabulated errors in Table 1, the proposed method is applicable for calculating option prices in the financial market.

3.3 Efficient Startup Procedure Demonstration

The startup procedure is determined by two main factors: mesh refinement (p) and duration time (q). The European call option is the benchmark problem for this test with $S_{max} = 1000,\ \sigma = 0.3,\ r = 0.03,\ K = 100,\ S_0 = 100,\ T = 0.1,\ N_T = 10^5$ and $\Delta\tau = 10^{-6}$. Assume that the startup procedure proceeds for q time steps with p times refined mesh. Here we used $p = 1,\ 2,\ 4,\ 8,\ 16$ and $q = 1,\ 10,\ 100,\ 1000,\ 10000$. The case where $p = 1$ is when a startup procedure is not applied. We observed the error at final time T on current price S_0. Each test is performed with $h = 10/2^i$ for $i = 0,\ 1,\ 2$ to derive the convergence rate. The convergence rate between $h = 10$ and $h = 5$ is given at Table 2 and the convergence rate between $h = 5$ and $h = 2.5$ is given at Table 3.

Tables 2 and 3 present the startup procedure shows significant results when $q = 1000$ and $q = 10000$, which is $1\%,\ 10\%$ of total time steps. The best choice for p

Table 1 Error and spatial convergence rate with and without startup procedure

	$h = 10$	Rate	$h = 5$	Rate	$h = 2.5$
Without startup procedure	4.6183 E − 11		9.3609E − 2		2.2207E − 2
Rate		2.3027		2.0757	
With startup procedure	2.6345E − 1		1.4954E − 2		8.9139E − 4
Rate		4.1390		4.0683	

Table 2 Convergence rate between $h = 10$ and $h = 5$ with respect to various p and q

	Number of time steps q				
p	1	10	100	1000	**10000**
1	2.3026	2.3026	2.3026	2.3026	2.3026
2	2.3029	2.3056	2.3329	2.5876	3.3815
4	2.3034	2.311	2.3855	3.0328	**4.0645**
8	2.3045	2.3215	2.4748	3.2642	4.5678
16	2.3066	2.3407	2.5564	3.2542	4.7966

Table 3 Convergence rate between $h = 5$ and $h = 2.5$ with respect to various p and q

	Number of time steps q				
p	1	10	100	**1000**	10000
1	2.0754	2.0754	2.0754	2.0754	2.0754
2	2.0767	2.0878	2.1966	2.9616	2.2736
4	2.0789	2.1094	2.4002	**4.1024**	2.5616
8	2.0831	2.1495	2.6171	4.5882	3.1702
16	2.0911	2.2058	2.6269	4.8016	3.6589

is when $p = 4$ for both cases because it best matches with the theoretical convergence rate of 4. Recommended p and q values are highlighted in bold in Tables 2 and 3. Therefore, a relatively short startup procedure with simple mesh refinement can lead to an accurate, fourth-order convergent scheme for the BS equation.

Regarding the tabulated data in Tables 2 and 3, we discuss the applicability of the startup procedure under various conditions and option models. First, we can see that the convergence rate strictly increases with higher mesh refinement p for all cases. This indicates that the proposed method is applicable to achieve the desired convergence rate by modifying the mesh refinement and duration time. Therefore, the proposed method can be adjusted in various situations depending on the discontinuity and non-differentiability of the payoff function.

3.4 Comparison with Previous Methods

We demonstrate the superiority of the proposed scheme over existing methodologies in the numerical analysis field. We start with classic methods from the literature. The main focus of comparison will be accuracy, computational efficiency, robustness, and applicability of the proposed method. We first compare it with an explicit method (Uddin et al., 2013). In order to emphasize the effect of the startup procedure, the derivatives are approximated with fourth-order convergence, as defined by Eqs. (10) and (5). The same initial conditions given in the last section, $p = 4$ and $q = 10000$ are used for comparison with the fourth-order explicit Euler method. Computational results for error and CPU time for three cases $h = 10,\ 5,\ 2.5$ are compared in Table 4.

Table 4 Comparison of error and cpu time between explicit and proposed methods

	$h = 10$		$h = 5$		$h = 2.5$	
Method	Error	cpu time	Error	cpu time	Error	cpu time
Explicit	7.7087E − 1	2.2500E−1	4.0375E − 1	2.3298E−1	3.3256E − 1	2.5625E−1
Proposed	1.6101E − 1	2.0625E−1	9.6233E − 3	3.3500E−1	1.6301E − 3	5.8750E−1

Due to the fact that the explicit Euler method is fundamentally simpler and faster than implicit methods, the fourth-order explicit Euler method shows slightly faster computational times. However, it fails to achieve sufficient computational error, which must be accurate in cent level. Moreover, the convergence rate of the explicit Euler method, presented in Table 4, is calculated as 0.93 and 0.28, which is far from the desired convergence order.

Next, we compare the second-order implicit Euler method with the proposed method (Lee & Lee, 2019). The first and second derivatives are approximated as $\partial u/\partial x = (u_{i+1} - u_{i-1})/2h$ and $\partial^2 u/\partial x^2 = (u_{i+1} - 2u_i + u_{i-1})/h^2$. Similar to the prior comparison, the same initial conditions are applied, and the computational results are illustrated in Table 5.

The second-order implicit Euler method is a tri-diagonal system, which performs slightly faster than the proposed penta-diagonal system. However, it fails to meet the error threshold, which makes it unsuitable for accurate financial market applications. Additionally, the second-order implicit Euler method does not achieve its target convergence rate, which shows convergence rates of 1.09 and 3.88. Therefore, we confirm that the proposed method is superior in both accuracy and applicability while performing well in terms of computational time.

4 Conclusions

In this paper, we presented a convergent fourth-order finite difference scheme for the BS equation. The proposed scheme was shown to achieve fourth-order convergence through Taylor series expansion. Notably, most option models in the financial market are not of class C^1, which poses a significant challenge when deriving the convergence rate from the Taylor series expansion. We resolve this issue by introducing a startup procedure, where a finer mesh is applied to the underlying asset during the initial phase of the method. The effectiveness of this startup procedure

Table 5 Comparison of error and cpu time between implicit and proposed methods

	$h = 10$		$h = 5$		$h = 2.5$	
Method	Error	cpu time	Error	cpu time	Error	cpu time
Implicit	9.6194E − 1	2.0625E−1	4.4948E − 1	2.4437E−1	3.4348E − 1	4.4375E−1
Proposed	1.6101E − 1	2.0625E−1	9.6233E − 3	3.3500E−1	1.6301E − 3	5.8750E−1

was demonstrated through numerical results. Furthermore, different values for p and q were used to explore an efficient model for the proposed method. We conclude that allocating 1–10% of the total time for the startup procedure, along with a four-fold finer mesh, is optimal in terms of both accuracy and efficiency.

The BS equation is very important in valuing options, especially barrier options where discontinuous or non-differentiable payoffs are frequently encountered. These discontinuities and non-differentiabilities can lead to problematic behavior, such as the failure to achieve the desired convergence rate of the numerical method (Dorfleitner et al., 2008). However, the proposed method successfully resolves the issue of non-differentiability in the European call option. Furthermore, Tables 2 and 3 showed a stable increase in convergence order as mesh refinement increases. Precise numerical results in Table 1 confirmed the applicability of the proposed method. Note that the startup procedure is applied only at the early stage of the numerical computation. Although such scenarios are rare, the proposed method may show limitations if an option model exhibits discontinuity or non-differentiability throughout the entire computation. Consequently, our proposed model demonstrates stable and accurate fourth-order convergence for various option models based on the BS equation. For future work, we will apply the proposed fourth-order accurate numerical scheme to compute the local volatility surface (Yoo et al., 2024).

Acknowledgements The corresponding author (J.S. Kim) was supported by the Brain Korea 21 FOUR from the Ministry of Education of Korea. Gyeonggyu Lee was supported by the National Institute for Mathematical Sciences (NIMS) granted by the Korea government (No. NIMS-B24910000). The authors thank the reviewers for the constructive and helpful comments on the revision of this article.

Funding Not applicable.

Data availability The data that support the findings of this study are available from the corresponding author upon request.

Declarations

Conflict of interest The author has no relevant financial or non-financial interests to disclose.

References

Alrabeei, S., & Yousuf, M. (2020). Robust Numerical Scheme for Pricing American Options under Jump Diffusion Models. *International Journal of Mathematical and Computational Sciences, 14*(11), 116–120.

Anderson, J. D., Degrez, G., Dick, E., & Grundmann, R. (2013). Computational fluid dynamics: an introduction. *Springer Science & Business Media*.

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of political economy, 81*(3), 637–654.

Cho, J., Yang, D., Kim, Y., & Lee, S. (2023). An operator splitting method for multi-asset options with the Feynman-Kac formula. *Computers & Mathematics with Applications, 135*, 93–101.

Deswal, K., & Kumar, D. (2022). Rannacher time-marching with orthogonal spline collocation method for retrieving the discontinuous behavior of hedging parameters. *Applied Mathematics and Computation, 427*, 127168.

Dorfleitner, G., Schneider, P., Hawlitschek, K., & Buch, A. (2008). Pricing options with Green's functions when volatility, interest rate and barriers depend on time. *Quantitative Finance, 8*(2), 119–133.

Eskiizmirliler, S., Günel, K., & Polat, R. (2021). On the solution of the black-scholes equation using feed-forward neural networks. *Computational Economics, 58*, 915–941.
Figlewski, S., & Gao, B. (1999). The adaptive mesh model: a new approach to efficient option pricing. *Journal of Financial Economics, 53*(3), 313–351.
Gan, X., Yang, Y., & Zhang, K. (2020). A robust numerical method for pricing American options under Kou's jump-diffusion models based on penalty method. *Journal of Applied Mathematics and Computing, 62*, 1–21.
Gulen, S., Popescu, C., & Sari, M. (2019). A new approach for the black-scholes model with linear and nonlinear volatilities. *Mathematics, 7*(8), 760.
Hu, J., & Gan, S. (2018). High order method for Black-Scholes PDE. *Computers & Mathematics with Applications, 75*(7), 2259–2270.
Hwang, Y., Kim, I., Kwak, S., Ham, S., Kim, S., & Kim, J. (2023). Unconditionally stable monte carlo simulation for solving the multi-dimensional Allen-Cahn equation. *Electronic Research Archive, 31*(8), 5104–5123.
Kim, S., Lyu, J., Lee, W., Park, E., Jang, H., Lee, C., & Kim, J. (2024). A practical monte Carlo method for pricing equity-linked securities with time-dependent volatility and interest rate. *Computational Economics, 63*(5), 2069–2086.
Kalantari, R., & Shahmorad, S. (2019). A stable and convergent finite difference method for fractional Black-Scholes model of American put option pricing. *Computational Economics, 53*, 191–205.
Lee, C., Kwak, S., Hwang, Y., & Kim, J. (2023). Accurate and efficient finite difference method for the Black-Scholes model with no far-field boundary conditions. *Computational Economics, 61*(3), 1207–1224.
Kwak, S., Kang, S., Ham, S., Hwang, Y., Lee, G., & Kim, J. (2023). An unconditionally stable difference scheme for the two-dimensional modified Fisher-Kolmogorov-Petrovsky-Piscounov equation. *Journal of Mathematics, 2023*(1), 5527728.
Lee, S., & Lee, Y. (2019). Stability of numerical methods under the regime-switching jump-diffusion model with variable coefficients. *ESAIM: Mathematical Modelling and Numerical Analysis*, 53(5), 1741–1762.
Li, Z., & Tourin, A. (2022). A finite difference scheme for pairs trading with transaction costs. *Computational Economics, 60*(2), 601–632.
Lötstedt, P., Persson, J., von Sydow, L., & Tysk, J. (2007). Space-time adaptive finite difference method for European multi-asset options. *Computers & Mathematics with Applications, 53*(8), 1159–1180.
Lux, T. (2022). Bayesian estimation of agent-based models via adaptive particle Markov chain Monte Carlo. *Computational Economics, 60*(2), 451–477.
Mehdizadeh Khalsaraei, M., Shokri, A., Mohammadnia, Z., & Sedighi, H. M. (2021). Qualitatively stable nonstandard finite difference scheme for numerical solution of the nonlinear Black-Scholes equation. *Journal of Mathematics, 2021*, 1–12.
Merton, R. C. (1973). Theory of rational option pricing. *The Bell Journal of economics and management science*, 141–183.
Morton, K. W., & Mayers, D. F. (2005). *Numerical solution of partial differential equations: an introduction*. Cambridge University Press.
Patel, K. S., & Mehra, M. (2020). Fourth order compact scheme for space fractional advection-diffusion reaction equations with variable coefficients. *Journal of Computational and Applied Mathematics, 380*, 112963.
Persson, J., & von Persson, L. (2007). Pricing European multi-asset options using a space-time adaptive FD-method. *Computing and Visualization in Science, 10*, 173–183.
Rezaei, M., Yazdanian, A. R., Ashrafi, A., & Mahmoudi, S. M. (2021). Numerical pricing based on fractional Black-Scholes equation with time-dependent parameters under the CEV model: Double barrier options. *Computers & Mathematics with Applications, 90*, 104–111.
Roul, P., & Goura, V. P. (2020). A new higher order compact finite difference method for generalised Black-Scholes partial differential equation: European call option. *Journal of Computational and Applied Mathematics, 363*, 464–484.
She, M., Li, L., Tang, R., & Li, D. (2021). A novel numerical scheme for a time fractional Black-Scholes equation. *Journal of Applied Mathematics and Computing, 66*, 853–870.
Tavella, D., & Randall, C. (2000). *Pricing financial instruments: The finite difference method*. New York: Wiley.
Tian, Z., Zhai, S., Ji, H., & Weng, Z. (2021). A compact quadratic spline collocation method for the time-fractional Black-Scholes model. *Journal of Applied Mathematics and Computing, 66*, 327–350.

Uddin, M. K. S., Ahmed, M., & Bhowmilk, S. K. (2013). A note on numerical solution of a linear black–scholes model. *GANIT: Journal of Bangladesh Mathematical Society*, 33, 103–115.
Vollert, A. (2001). Margrabe's option to exchange in a paretian-stable subordinated market. *Mathematical and Computer Modelling, 34*(9–11), 1185–1197.
Voss, D. A., Khaliq, A. Q. M., Kazmi, S. H. K., & He, H. (2003). A fourth order L-stable method for the Black–Scholes model with barrier options. in: M.L. Gavrilova, V. Kumar, C.J.K. Tan (Eds.), *Lecture Notes in Computer Science*, (2669) Springer, Berlin, 199–207.
Wade, B. A., Khaliq, A. Q., Yousuf, M., Vigo-Aguiar, J., & Deininger, R. (2007). On smoothing of the Crank-Nicolson scheme and higher order schemes for pricing barrier options. *Journal of Computational and Applied Mathematics, 204*(1), 144–158.
Windcliff, H., Forsyth, P. A., & Vetzal, K. R. (2004). Analysis of the stability of the linear boundary condition for the Black-Scholes equation. *Journal of Computational Finance, 8*, 65–92.
Yoo, C., Kwak, S., Hwang, Y., Jang, H., Kim, H., & Kim, J. (2024). Calibration of local volatility surfaces from observed market call and put option prices. *Computational Economics*. https://doi.org/10.1007/s10614-024-10590-9
Zhang, M., & Zheng, X. (2022). Numerical Approximation to a Variable-Order Time-Fractional Black–Scholes Model with Applications in Option Pricing. *Computational Economics*, 1–21.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.