Comput Econ (2018) 51:961–972
https://doi.org/10.1007/s10614-017-9653-0

Finite Difference Method for the Black–Scholes Equation Without Boundary Conditions

Darae Jeong[1] · Minhyun Yoo[2] · Junseok Kim[1]

Accepted: 18 January 2017 / Published online: 25 January 2017

Abstract We present an accurate and efficient finite difference method for solving the Black–Scholes (BS) equation without boundary conditions. The BS equation is a backward parabolic partial differential equation for financial option pricing and hedging. When we solve the BS equation numerically, we typically need an artificial far-field boundary condition such as the Dirichlet, Neumann, linearity, or partial differential equation boundary condition. However, in this paper, we propose an explicit finite difference scheme which does not use a far-field boundary condition to solve the BS equation numerically. The main idea of the proposed method is that we reduce one or two computational grid points and only compute the updated numerical solution on that new grid points at each time step. By using this approach, we do not need a boundary condition. This procedure works because option pricing and computation of the Greeks use the values at a couple of grid points neighboring an interesting spot. To demonstrate the efficiency and accuracy of the new algorithm, we perform the numerical experiments such as pricing and computation of the Greeks of the vanilla call, cash-or-nothing, power, and powered options. The computational results show excellent agreement with analytical solutions.

Keywords Black–Scholes equation · Finite difference method · Far field boundary conditions

✉ Junseok Kim
cfdkim@korea.ac.kr
http://math.korea.ac.kr/~cfdkim/

1 Department of Mathematics, Korea University, Seoul 136-713, Republic of Korea

2 Department of Financial Engineering, Korea University, Seoul 136-071, Republic of Korea

1 Introduction

In this paper, we propose an accurate and efficient finite difference method to solve the Black–Scholes (BS) equation:

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

with the final condition $u(x, T) = p(x)$, where $u(x, t)$ is the value of an option, x is the value of the underlying asset, t is the time, T is the expiry date, σ is the volatility of the underlying security, and r is the risk-free interest rate (Black and Scholes 1973; Merton 1973). Change of the variable $\tau = T - t$ transforms Eq. (1) into the initial value problem (Farnoosh et al. 2016):

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

with the initial condition $u(x, 0) = p(x)$. Since we cannot numerically solve the BS equation on an infinite domain $(0, \infty)$, we must truncate the infinite domain into a finite domain and need artificial boundary conditions to solve the BS equation numerically (Tavella and Randall 2000). If one knows asymptotical values of the solution of the BS equation, then we can use the Dirichlet boundary condition (Company et al. 2008; Hajipour and Malek 2015; Reisinger and Wittum 2004; Smith 2000; Tangman et al. 2008; Tavella and Randall 2000; Vazquez 1998; Windcliff et al. 2004). Another simplest Dirichlet boundary condition is simply setting the boundary values to be fixed all the time with the payoff value (Kangro and Nicolaides 2000; Pooley et al. 2003). Neumann boundary condition specifies values for the derivative of the solution at the boundary of the spatial domain. This boundary condition also requires the knowledge of the asymptotical behavior of the derivatives of the solution (Kurpiel and Roncalli 1999). Linearity boundary condition assumes that the second derivative of the option value with respect to the underlying asset price x vanishes to zero for the large value of the asset price (Floc'h 2014; Linde et al. 2009; Lötstedt et al. 2007; Windcliff et al. 2004). In the partial differential equation (PDE) boundary condition, we use one-sided discretizations at the boundary points so that we do not require the values of the ghost points (Tavella and Randall 2000; Windcliff et al. 2004).

The main purpose of this work is to present an accurate and efficient numerical method for solving the Black–Scholes equation without boundary conditions. The principle of the proposed method is to reduce one or two computational grid points and only compute the advanced numerical solution on that new grid points at each time step. This approach does not require a boundary condition.

This paper is organized as follows. In Sect. 2, we discretize the BS equation on a non-uniformly spaced grid. Section 3 provides numerical results for the proposed numerical algorithm. Section 4 concludes with a short summary.

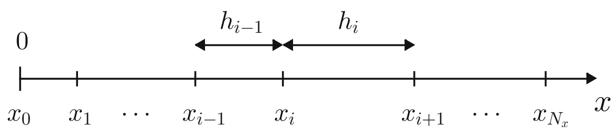

Fig. 1 A non-uniform grid with the grid spacing h_i

2 Numerical Solution

We discretize the BS equation on a grid defined by $x_0 = 0$ and $x_{i+1} = x_i + h_i$ for $i = 0, \ldots, N_x - 1$, where h_i is the grid size and N_x is the number of grid intervals, see Fig. 1.

Let $u_i^n \approx u(x_i, n\Delta\tau)$ be the numerical approximation of the solution, where $\Delta\tau = T/N_\tau$ is the time step size and N_τ is the total number of time steps. By applying an explicit scheme to Eq. (2), we have

$$\frac{u_i^{n+1} - u_i^n}{\Delta\tau} = \frac{\sigma^2 x_i^2}{2}\left(\frac{2u_{i-1}^n}{h_{i-1}(h_{i-1}+h_i)} - \frac{2u_i^n}{h_{i-1}h_i} + \frac{2u_{i+1}^n}{h_i(h_{i-1}+h_i)}\right) + \frac{rx_i(u_{i+1}^n - u_{i-1}^n)}{h_{i-1}+h_i} - ru_i^n. \tag{3}$$

Since the numerical scheme is explicit, to get u_i^{n+1} we only need three values, i.e., u_{i-1}^n, u_i^n, and u_{i+1}^n. At $x = 0$, Eq. (2) becomes $\partial u/\partial\tau = -ru$. Therefore, $u_0^n = u_0^0 e^{-r\tau}$. However, we do not use any boundary conditions for the far-field boundary. Instead, we reduce grid points by one in every time step. In most computational finance applications, we only need a couple of numerical values at limited grid points to calculate the option price itself and its derivatives such as the Greeks.

To derive a stability condition of the proposed scheme, we follow the stability analysis in (Zvan et al. 1998). Rewriting Eq. (3) gives

$$u_i^{n+1} = \frac{\Delta\tau(\sigma^2 x_i^2 - rx_i h_{i-1})}{h_{i-1}(h_{i-1}+h_i)} u_{i-1}^n + \left(1 - r\Delta\tau - \frac{\Delta\tau\sigma^2 x_i^2}{h_{i-1}h_i}\right) u_i^n + \frac{\Delta\tau(\sigma^2 x_i^2 + rx_i h_i)}{h_i(h_{i-1}+h_i)} u_{i+1}^n. \tag{4}$$

In order for all coefficients of u_{i-1}^n, u_i^n, and u_{i+1}^n in Eq. (4) to be positive, the following Peclet condition should be satisfied

$$h_{i-1} < \frac{\sigma^2 x_i}{r}. \tag{5}$$

Note that this condition is automatically satisfied if $\sigma^2/r > 1$; otherwise we can put grid points far away from zero to satisfy the condition. The other condition is

$$\Delta\tau < \frac{h_{i-1}h_i}{rh_{i-1}h_i + \sigma^2 x_i^2}. \tag{6}$$

Under two conditions Eqs. (5) and (6), let $u_i^{n,\max} = \max(u_{i-1}^n, u_i^n, u_{i+1}^n)$, then Eq. (4) can be written as

$$\begin{aligned} u_i^{n+1} \le \frac{\Delta\tau(\sigma^2 x_i^2 - r x_i h_{i-1})}{h_{i-1}(h_{i-1}+h_i)} u_i^{n,\max} + \left(1 - r\Delta\tau - \frac{\Delta\tau\sigma^2 x_i^2}{h_{i-1}h_i}\right) u_i^{n,\max} \\ + \frac{\Delta\tau(\sigma^2 x_i^2 + r x_i h_i)}{h_i(h_{i-1}+h_i)} u_i^{n,\max} \le (1 - r\Delta\tau) u_i^{n,\max} \le u_i^{n,\max}. \end{aligned} \tag{7}$$

Let $u_i^{n,\min} = \min(u_{i-1}^n, u_i^n, u_{i+1}^n)$, then by a similar argument we obtain

$$u_i^{n+1} \ge (1 - r\Delta\tau) u_i^{n,\min}. \tag{8}$$

Finally, from Eqs. (7) and (8) we have

$$u^{n,\min} \le (1 - r\Delta\tau) u_i^{n,\min} \le u_i^{n+1} \le u_i^{n,\max} \le u^{n,\max}, \tag{9}$$

where $u^{n,\min} = \min_i u_i^{n,\min}$ and $u^{n,\max} = \max_i u_i^{n,\max}$. By Eq. (9), the numerical solutions are bounded and stable if the time-step satisfies Eq. (6) and $\Delta\tau \le 1/r$ from Eq. (8).

Next, we describe the basic idea of the proposed algorithm. Let us assume that we are interested in finding numerical solutions at points, $x_{m_1}, \ldots, x_{m_2}$. Given σ, r, and T, we generate a time step $\Delta\tau$ and a non-uniform grid which satisfy Eq. (6). Once the time step and grid are defined, we compute u_i^{n+1} for $i = m_3(n), \ldots, m_4(n)$, where $m_3(n) = 0$ if $n \le N_\tau - m_1 - 2$; otherwise $m_3(n) = n - N_\tau + m_1 + 1$, and $m_4(n) = N_x - n - 1$. Figure 2 shows a schematic illustration of the proposed numerical algorithm for a European call option with a strike price K.

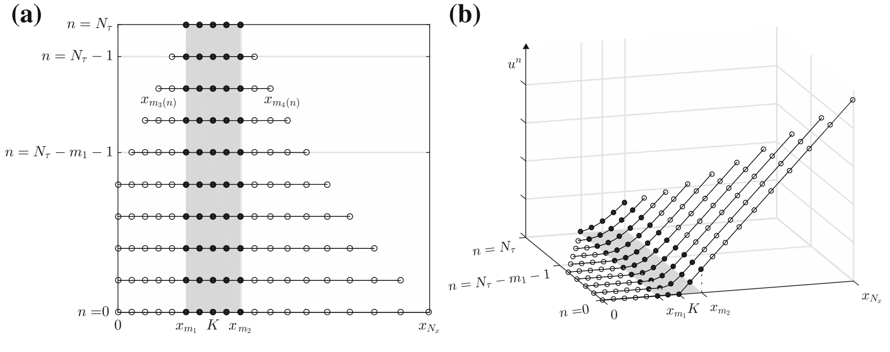

Fig. 2 Schematic illustration of the proposed numerical algorithm for a European call option with a strike price K. **a** A reducing grid with respect to time. **b** Value $u(x, \tau)$ on the reducing grid

3 Computational Results

In order to show the efficiency and accuracy of the proposed algorithm, we present the numerical experiments such as pricing and computation of the Greeks of the vanilla call, cash-or-nothing, power, and powered options. The Greeks of option values are derivatives with respect to market variables or model parameters. Delta (Δ), Gamma (Γ), Theta (Θ), Vega (ν), and Rho (ρ) are defined as follows: $\partial u/\partial x$, $\partial^2 u/\partial x^2$, $\partial u/\partial t = -\partial u/\partial \tau$, $\partial u/\partial \sigma$, and $\partial u/\partial r$, respectively (Haug 1997). Unless otherwise specified, we use the following parameters: $\sigma = 0.3$, $r = 0.03$, and $T = 1$. The proposed model was implemented in MATLAB 9.0 (MathWorks, Inc. 2015) on a computer with Intel(R) Core(TM)i5-6400 CPU@2.70GHZ with 16GB RAM, and the Windows 10 operating system.

3.1 European Call Option

As the first numerical test, we consider a European call option. The initial condition is given as $u(x, 0) = \max(x - K, 0)$ with the strike price $K = 100$. We will approximate the solutions at $x = 100 - h$, 100, $100 + h$ to compute the option pricing and the Greeks, where $h = 1$, 0.5, and 0.25. Let $x_i = hi$ for $i = 0, \ldots, 106/h$. Therefore, $h_i = h$ for $i = 0, \ldots, 106/h - 1$. Let us define a temporary time step,

$$\Delta\tau_{\text{tmp}} = \frac{s h_{106/h-2} h_{106/h-1}}{r h_{106/h-2} h_{106/h-1} + \sigma^2 x_{106/h-1}^2}, \tag{10}$$

where s is a safety factor and we take $s = 0.95$ unless otherwise stated. Let $N_\tau = [T/\Delta\tau_{\text{tmp}}] + 1$, where $[x]$ denotes the largest integer not greater than x. We define the time step as $\Delta\tau = T/N_\tau$. For $i = 106/h, \ldots, 100/h + N_\tau$, we define $h_i = \Delta\tau(\sigma x_i)^2/(s h_{i-1} - \Delta\tau r h_{i-1})$ and $x_{i+1} = x_i + h_i$. Therefore, $N_x = 106/h + N_\tau$, $m_1 = 100/h - 1$, $m_2 = 100/h + 1$, $m_3(n) = 0$ if $n \le N_\tau - m_1 - 2$; otherwise $m_3(n) = n - N_\tau + m_1 + 1$, and $m_4(n) = N_x - n - 1$. For the European call option, the closed-form solution of the BS equation is

$$\begin{aligned} u(x,\tau) &= xN(d_1) - Ke^{-r\tau}N(d_2), \\ d_1 &= \left(\ln(x/K) + \left(r + 0.5\sigma^2\right)\tau\right)/(\sigma\sqrt{\tau}), \quad d_2 = d_1 - \sigma\sqrt{\tau}, \end{aligned}$$

where $N(d) = (1/\sqrt{2\pi})\int_{-\infty}^{d} \exp(-0.5x^2)\,dx$ is the cumulative distribution function for the standard normal distribution (Black and Scholes 1973). The sensitivities of options, called Greeks, are represented as follows (Haug 1997).

$$\begin{aligned} &\Delta = N(d_1), \quad \Gamma = \frac{N'(d_1)}{\sigma x\sqrt{\tau}}, \quad \Theta = -\frac{\sigma x N'(d_1)}{2\sqrt{\tau}} - rKe^{-r\tau}N(d_2), \\ &\nu = x\sqrt{\tau}N'(d_1), \quad \rho = \tau x e^{-r\tau}N(d_2). \end{aligned}$$

Table 1 Convergence of a European call option: absolute errors in the option price and its Greeks at $x = 100$ and $T = 1$

h	N_τ	u (13.283)	Δ (0.599)	Γ (0.013)	Θ (−7.197)	ν (38.667)	ρ (46.587)
1	1050	6.55e−3	2.53e−5	2.83e−6	1.61e−4	1.04e−2	3.21e−3
		[0.09]	[0.09]	[0.09]	[0.09]	[0.19]	[0.18]
$\frac{1}{2}$	4183	1.65e−3	6.33e−6	7.12e−7	3.98e−5	2.61e−3	7.86e−4
		[1.24]	[1.24]	[1.24]	[1.24]	[2.45]	[2.61]
$\frac{1}{4}$	16,717	4.12e−4	1.58e−6	1.78e−7	9.92e−6	6.50e−4	1.73e−4
		[19.25]	[19.25]	[19.25]	[19.25]	[38.47]	[40.73]

The exact values are in parentheses and the CPU times in seconds for each test are in squared parentheses

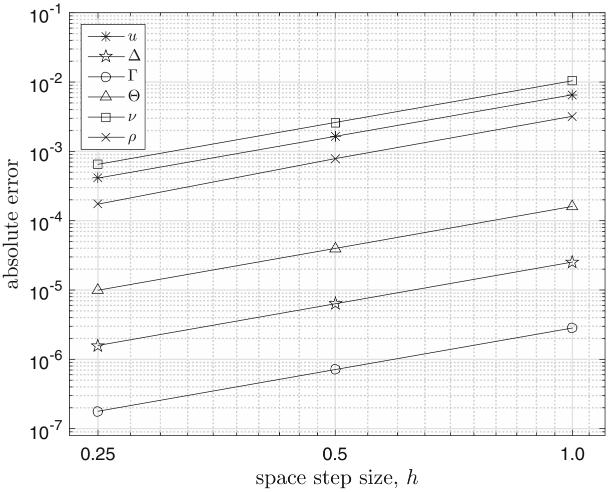

Fig. 3 Convergence of the numerical results for a European call option with various h

Table 1 displays the absolute errors in the option price and its Greeks at $x = 100$ and $T = 1$. The exact values are in parentheses and the CPU times in seconds for each test are listed in squared parentheses. We can see the convergence of the numerical results of a European call option as we refine the grid size.

Figure 3 shows the absolute errors of numerical results for u, Δ, Γ, Θ, ν, and ρ with various spatial step sizes. As we expected, the numerical results converge with second-order accuracy.

3.2 Cash-or-Nothing Option

Next, we consider the cash-or-nothing option which pays an amount C at expiration if the underlying asset is greater than K; otherwise, the payoff is zero. For this test, we use $C = K = 100$. The exact solution and Greeks are written as follows.

Table 2 Convergence of a European cash-or-nothing option: absolute errors in the option price and its Greeks at $x = 100$ and $T = 1$

h	N_τ	u (46.587)	Δ (1.289)	Γ (−0.011)	Θ (2.364)	ν (−32.222)	ρ (82.302)
1	1050	6.93e−4	2.88e−4	1.23e−5	5.19e−4	3.49e−2	7.26e−2
		[0.10]	[0.10]	[0.10]	[0.10]	[0.18]	[0.19]
$\frac{1}{2}$	4183	1.71e−4	7.25e−5	3.08e−6	1.28e−4	8.62e−3	1.83e−2
		[1.26]	[1.26]	[1.26]	[1.26]	[2.50]	[2.45]
$\frac{1}{4}$	16,717	4.26e−5	1.82e−5	7.71e−7	3.19e−5	2.05e−3	4.72e−3
		[18.45]	[18.45]	[18.45]	[18.45]	[38.09]	[37.85]

The exact values are in parentheses and the CPU times in seconds for each test are in squared parentheses

$$u(x,\tau) = Ce^{-r\tau}N(d_2), \quad \Delta = \frac{Ce^{-r\tau}N'(d_2)}{\sigma x\sqrt{\tau}}, \quad \Gamma = -\frac{Cd_1e^{-r\tau}N'(d_2)}{(\sigma x)^2\tau},$$
$$\nu = -Ce^{-r\tau}\frac{d_1}{\sigma}N'(d_2), \quad \Theta = Ce^{-r\tau}\left(rN(d_2) + \left(\frac{d_1}{2\tau} - \frac{r}{\sigma\sqrt{\tau}}\right)N'(d_2)\right),$$
$$\rho = Ce^{-r\tau}\left(-\tau N(d_2) + \frac{\sqrt{\tau}}{\sigma}N'(d_2)\right).$$

The initial condition is given as $u(x, 0) = 100$ if $x > 100$ and zero otherwise. We will approximate the solutions at $x = 100-1.5h, 100-0.5h, 100+0.5h, 100+1.5h$ to compute the option pricing and the Greeks, where $h = 1, 0.5$, and 0.25. Let $x_0 = 0$ and $x_i = h(i-0.5)$ for $i = 1, \ldots, 106/h$. Therefore, $h_i = h$ for $i = 0, \ldots, 106/h-1$. Let the number of time steps be defined as follows:

$$N_\tau = \left[\frac{T(rh_{106/h-2}h_{106/h-1} + \sigma^2 x_{106/h-1}^2)}{sh_{106/h-2}h_{106/h-1}}\right] + 1,$$

then $\Delta\tau = T/N_\tau$. For $i = 106/h, \ldots, 100/h+N_\tau$, we define $h_i = \Delta\tau(\sigma x_i)^2/(sh_{i-1} - \Delta\tau r h_{i-1})$ and $x_{i+1} = x_i + h_i$. Therefore, $N_x = 106/h + N_\tau$, $m_1 = 100/h - 1$, $m_2 = 100/h+1$, $m_3(n) = 0$ if $n \le N_\tau - m_1 - 2$; otherwise $m_3(n) = n - N_\tau + m_1 + 1$, and $m_4(n) = N_x - n - 1$.

Table 2 shows the absolute errors in the option price and its Greeks at $x = 100$ and $T = 1$. The exact values are in parentheses and the CPU times in seconds for each test are listed in squared parentheses. We can see the convergence of the numerical results of a European cash-or-nothing option as we refine the grid size.

Figure 4 shows the absolute errors of numerical results for u, Δ, Γ, Θ, ν, and ρ with various spatial step sizes and demonstrates the second-order convergence of the numerical results.

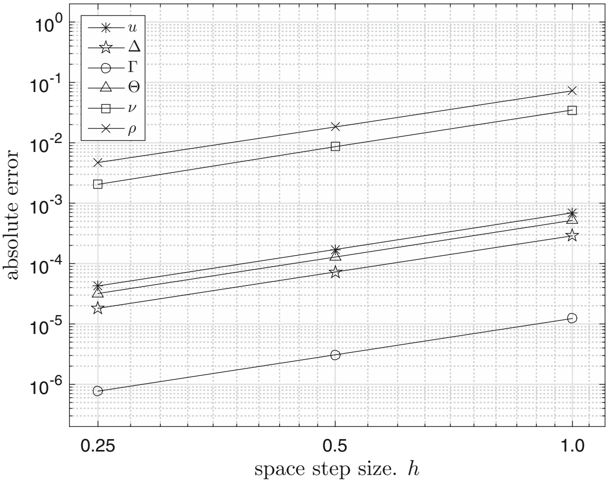

Fig. 4 Convergence of the numerical results for a European cash-or-nothing option with various h

3.3 Standard Call Power Option

Standard call power option has nonlinear payoff at maturity, $u(x, 0) = \max(x^p - K, 0)$, where p is some power. The closed-form solution of this power option is given by (Haug 1997):

$$u(x, \tau) = x^p e^{(p-1)(r+0.5p\sigma^2)\tau} N(d_1) - Ke^{-r\tau} N(d_2), \tag{11}$$

where $d_1 = \left(\ln\left(x/K^{1/p}\right) + \left(r + (p - 0.5)\sigma^2\right)\tau\right)/(\sigma\sqrt{\tau}),\ d_2 = d_1 - p\sigma\sqrt{\tau}$. We consider a standard power option with current asset price of 10, power of 2, and $K = 100$.

We will calculate the solutions at $x = 10 - h$, 10, $10 + h$, to compute the option pricing and the Greeks, where $h = 0.125$, 0.0625, and 0.003125. Let $x_i = hi$ for $i = 1, \ldots, 16/h$. Therefore, $h_i = h$ for $i = 0, \ldots, 16/h - 1$. Let the number of time steps be defined as follows:

$$N_\tau = \left[\frac{T(rh_{16/h-2}h_{16/h-1} + \sigma^2 x_{16/h-1}^2)}{sh_{16/h-2}h_{16/h-1}}\right] + 1, \tag{12}$$

then $\Delta\tau = T/N_\tau$. For $i = 16/h, \ldots, 10/h + N_\tau$, we define $h_i = \Delta\tau(\sigma x_i)^2/(sh_{i-1} - \Delta\tau r h_{i-1})$ and $x_{i+1} = x_i + h_i$. Therefore, $N_x = 16/h + N_\tau$, $m_1 = 10/h - 1$, $m_2 = 10/h + 1$, $m_3(n) = 0$ if $n \leq N_\tau - m_1 - 2$; otherwise $m_3(n) = n - N_\tau + m_1 + 1$, and $m_4(n) = N_x - n - 1$.

Table 3 lists the absolute errors in the option price and its Greeks at $x = 10$ and $T = 1$. The exact values are in parentheses and the CPU times in seconds for each test are listed in squared parentheses. We can observe the convergence of the numerical results of a standard power option as we refine the grid size.

Table 3 Convergence of a standard power option: absolute errors in the option price and its Greeks at $x = 10$ and $T = 1$

h	N_τ	u (33.334)	Δ (15.984)	Γ (4.176)	Θ (22.588)	ν (125.287)	ρ (126.509)
$\frac{1}{8}$	673	3.64e−3	1.71e−4	1.17e−4	9.21e−4	1.66e−2	5.45e−3
		[0.16]	[0.16]	[0.16]	[0.16]	[0.32]	[0.31]
$\frac{1}{16}$	5462	9.10e−4	4.24e−5	2.98e−5	2.29e−4	4.21e−3	1.38e−3
		[2.10]	[2.10]	[2.10]	[2.10]	[4.15]	[4.19]
$\frac{1}{32}$	21832	2.27e−4	1.06e−5	7.49e−6	5.72e−5	1.12e−3	3.57e−4
		[32.26]	[32.26]	[32.26]	[32.26]	[64.04]	[65.15]

The exact values are in parentheses and the CPU times in seconds for each test are in squared parentheses

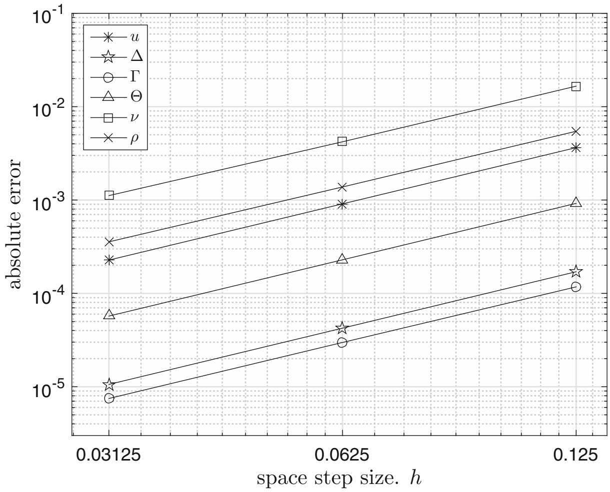

Fig. 5 Convergence of the numerical results for a standard power option with various h

In Fig. 5, we also confirm the second-order convergence of the numerical results for u, Δ, Γ, Θ, ν, and ρ.

3.4 Powered Option

Finally, we consider a powered option whose payoff function at maturity T is $u(x, 0) = \max(x - K, 0)^p$, where p is a power (Haug 1997). The closed-form solution (Esser 2003; Heynen and Kat 1996; Zhang 1998) of the powered option is given by

$$u(x, \tau) = \sum_{q=0}^{p} \frac{p!}{q!(p-q)!} x^{p-q} (-K)^q e^{(p-q-1)(r+0.5(p-q)\sigma^2)\tau} N(d_{p,q}), \qquad (13)$$

Table 4 Convergence of a powered option: absolute errors in the option price and its Greeks at $x = 100$ and $T = 1$

h	N_τ	u (676.758)	Δ (40.102)	Γ (1.598)	Θ (819.296)	ν (4795.291)	ρ (3333.420)
1	1050	1.02e−1	5.20e−3	5.30e−5	7.65e−2	1.07e−0	1.10e−0
		[0.09]	[0.09]	[0.09]	[0.09]	[0.18]	[0.18]
$\frac{1}{2}$	4183	2.54e−2	1.30e−3	1.34e−5	1.92e−2	2.63e−1	2.71e−1
		[1.24]	[1.24]	[1.24]	[1.24]	[2.47]	[2.47]
$\frac{1}{4}$	16717	6.35e−3	3.26e−4	3.34e−6	4.80e−3	5.88e−2	6.41e−2
		[18.92]	[18.92]	[18.92]	[18.92]	[38.37]	[39.06]

The exact values are in parentheses and the CPU times in seconds for each test are in squared parentheses

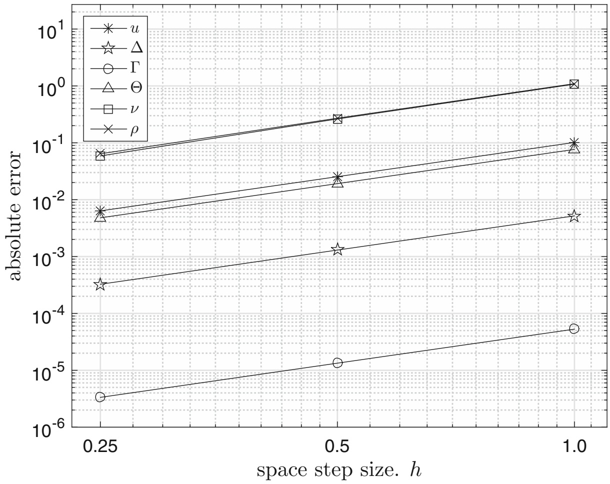

Fig. 6 Convergence of the numerical results for a powered option with various h

where $d_{p,q} = [\ln(x/K) + (r + (p - q - 0.5)\sigma^2)\tau]/(\sigma\sqrt{\tau})$. In this example, we choose $p = 2$ and $K = 100$. We use the same time step and non-uniform grid as used in the European call option test. Table 4 reports the absolute errors in the option price and its Greeks of powered option. From the table, we see that the computed option prices and its Greeks converge toward the exact values as the grid sizes are refined.

Figure 6 shows the second-order convergence of the numerical results for a powered option with various h.

4 Conclusions

We presented an accurate and efficient finite difference method for solving the BS equation without boundary conditions. Unlike most finite difference methods which need an artificial far-field boundary condition such as the Dirichlet, Neumann, linearity, or PDE boundary condition, the proposed explicit finite difference scheme does not use a far-field boundary condition. The basic idea of the method is that we reduce one or two computational grid points and only compute the updated numerical solution on that new grid points at each time step. Therefore, we do not need a boundary condition. This algorithm works because option pricing and computation of the Greeks use the values at a couple of grid points. We demonstrated the efficiency and accuracy of the new proposed algorithm by performing the numerical experiments such as pricing and computation of the Greeks of the vanilla call, cash-or-nothing, power, and powered options. The computational results showed excellent agreement with analytical solutions. In real world financial practices, the sizes of the space step and time step are sufficiently large and small, respectively. Therefore, it is reasonable to use an explicit time marching scheme to advance the numerical solutions. As future research works, it is natural to extend the proposed scheme to multi-asset option pricing problems.

Acknowledgements The authors (D. Jeong and J.S. Kim) thank the Korea Institute for Advanced Study (KIAS) for supporting research on the financial pricing model using artificial intelligence. The corresponding author (J.S. Kim) was supported by a subproject of project Research for Applications of Mathematical Principles (No. C21501) and supported by the National Institute of Mathematics Sciences (NIMS). The authors are grateful to the anonymous referees whose valuable suggestions and comments significantly improved the quality of this paper.

References

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

Company, R., Navarro, E., Pintos, J. R., & Ponsoda, E. (2008). Numerical solution of linear and nonlinear Black–Scholes option pricing equations. *Computers & Mathematics with Applications*, *56*(3), 813–821.

Esser, A. (2003). General valuation principles for arbitrary payoffs and applications to power options under stochastic volatility. *Financial Markets and Portfolio Management*, *17*(3), 351–372.

Farnoosh, R., Rezazadeh, H., Sobhani, A., & Beheshti, M. H. (2016). A Numerical method for discrete single barrier option pricing with time-dependent parameters. *Computational Economics*, *48*(1), 131–145.

Hajipour, M., & Malek, A. (2015). Efficient high-order numerical methods for pricing of options. *Computational Economics*, *45*(1), 31–47.

Haug, E. G. (1997). *The complete guide to option pricing formulas*. New York: McGraw-Hill.

Heynen, R. C., & Kat, H. M. (1996). Pricing and hedging power options. *Financial Engineering and the Japanese Markets*, *3*(3), 253–261.

Kangro, R., & Nicolaides, R. (2000). Far field boundary conditions for Black–Scholes equations. *SIAM Journal on Numerical Analysis*, *38*(4), 1357–1368.

Kurpiel, A., & Roncalli, T. (1999). Hopscotch methods for two-state financial models. *Journal of Computational Finance*, *3*(2), 53–89.

Le Floc'h, F. (2014). TR-BDF2 for fast stable American option pricing. *Journal of Computational Finance*, *17*(3), 31–56.

Linde, G., Persson, J., & von Sydow, L. (2009). A highly accurate adaptive finite difference solver for the Black–Scholes equation. *International Journal of Computer Mathematics*, *86*(12), 2104–2121.

Lötstedt, P., Persson, J., von Sydow, L., & Tysk, J. (2007). Space–time adaptive finite difference method for European multi-asset options. *Computers and Mathematics with Applications*, *53*(8), 1159–1180.

MathWorks, Inc. (2015). *MATLAB: The language of technical computing*. Natick, MA: The MathWorks. http://www.mathworks.com/.

Merton, R. C. (1973). Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, *4*(1), 141–183.

Pooley, D. M., Forsyth, P. A., & Vetzal, K. R. (2003). Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis*, *23*(2), 241–267.

Reisinger, C., & Wittum, G. (2004). On multigrid for anisotropic equations and variational inequalities "Pricing multi-dimensional European and American options". *Computing and Visualization in Science*, *7*(3–4), 189–197.

Smith, R. (2000). On multigrid for anisotropic equations and variational inequalities "Pricing multi-dimensional European and American options". *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *456*(1997), 1019–1028.

Tangman, D. Y., Gopaul, A., & Bhuruth, M. (2008). Numerical pricing of options using high-order compact finite difference schemes. *Journal of Computational and Applied Mathematics*, *218*(2), 270–280.

Tavella, D., & Randall, C. (2000). *Pricing Financial Instruments: The Finite Difference Method*. New York: Wiley.

Vazquez, C. (1998). An upwind numerical approach for an American and European option pricing model. *Applied Mathematics and Computation*, *97*(2), 273–286.

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.

Zhang, P. G. (1998). *A guide to second generation options*. Singapore: World Scientific.

Zvan, R., Forsyth, P. A., & Vetzal, K. R. (1998). Robust numerical methods for PDE models of Asian options. *Journal of Computational Finance*, *1*, 39–78.