Comput Econ
DOI 10.1007/s10614-017-9730-4

A Hybrid Monte Carlo and Finite Difference Method for Option Pricing

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

Accepted: 28 August 2017

Abstract We propose an accurate, efficient, and robust hybrid finite difference method, with a Monte Carlo boundary condition, for solving the Black–Scholes equations. The proposed method uses a far-field boundary value obtained from a Monte Carlo simulation, and can be applied to problems with non-linear payoffs at the boundary location. Numerical tests on power, powered, and two-asset European call option pricing problems are presented. Through these numerical simulations, we show that the proposed boundary treatment yields better accuracy and robustness than the most commonly used linear boundary condition. Furthermore, the proposed hybrid method is general, which means it can be applied to other types of option pricing problems. In particular, the proposed Monte Carlo boundary condition algorithm can be implemented easily in the code of the existing finite difference method, with a small modification.

Keywords Black–Scholes equation · Finite difference method · Option pricing · Boundary condition · Monte Carlo simulation

1 Introduction

We present an efficient and accurate hybrid numerical method to solve the following Black–Scholes partial differential equation (Black and Scholes 1973):

✉ Junseok Kim
cfdkim@korea.ac.kr

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

[2] Department of Financial Engineering, Korea University, Seoul 02841, Republic of Korea

Published online: 30 August 2017

$$\frac{\partial u}{\partial t} = -\frac{1}{2}\sum_{i,j=1}^{d} \sigma_i \sigma_j \rho_{ij} S_i S_j \frac{\partial^2 u}{\partial S_i \partial S_j} - r\sum_{i=1}^{d} S_i \frac{\partial u}{\partial S_i} + ru$$
$$\text{for } (\mathbf{S}, t) \in (0, \infty)^d \times [0, T), \tag{1}$$

where $u(\mathbf{S}, t)$ is the value of the derivative security, $\mathbf{S} = (S_1, \ldots, S_d)$ is the d-dimensional multivariate process, t denotes time, T is the expiry date, σ_i is the volatility of the underlying assets, ρ_{ij} is the asset correlation, and r is the risk-free interest rate. Even though there are analytic solutions for pricing derivatives (Capiński 2015; Pun et al. 2015), these are often difficult to derive. As an alternative, numerical solutions can be used (Bandi and Bertsimas 2014). The finite difference method (FDM) is widely used to solve PDEs numerically, and has been used successfully for pricing financial contracts (Jaillet et al. 1990; Tavella and Randall 2000; Windcliff et al. 2004; Duffy 2006; Yang et al. 2007; Tangman et al. 2008; Marroquı and Moreno 2013). To solve the BS equation numerically, we have to truncate the infinite domain $(0, \infty)^d$ to a finite domain $(0, S_{\max})^d$, and then impose a boundary condition at $\mathbf{S} = \mathbf{S}_{\max}$ (Windcliff et al. 2004). There are numerous boundary conditions have been proposed, including Dirichlet (Windcliff et al. 2004; Yang et al. 2007), linear (d'Halluin et al. 2004; Windcliff et al. 2004), Neumann, and PDE (Tavella and Randall 2000; Windcliff et al. 2004) boundary conditions.

The main difficulty with imposing Dirichlet or Neumann conditions is that we need to know the behavior of the solution near the boundary region. The asymptotic behavior of the solution can be determined in some cases, however, in general, this is not a straightforward task (Windcliff et al. 2004). Therefore, the linear boundary condition (LBC) $\partial^2 u/\partial S_i^2 = 0$ for $i = 1, \ldots, d$ is usually used because many of option values are nearly linear with respect to the underlying asset at far from the strike price (Tavella and Randall 2000). However, the payoff may be given as a nonlinear function of the underlying asset (e.g., power or powered options (Zhang 1995; Haug 2007). This implies that using the LBC is not an appropriate choice. In such a case, we can use the PDE boundary condition, which uses one-sided discretization inside the computational domain (Tavella and Randall 2000). Nevertheless, there is a stability restriction on using the PDE boundary condition, because the spatial discretization matrix may not satisfy the stability condition (Windcliff et al. 2004).

The purpose of this work is to develop an accurate, efficient, and robust numerical boundary condition for solving the BS equation. The method is based on the finite difference method, and uses a far-field boundary value derived from a Monte Carlo simulation (MCS). MCSs are commonly used to estimate option prices in the field of applied probability (Hammersley et al. 1964). However, the standard MSC suffers from the disadvantage of slow convergence. Here, the statistical error is proportional to $1/\sqrt{M}$, where M is the number of simulations (Boyle et al. 1997). Even worse, it incurs a very high computational cost when generating accurate hedging parameters, such as the Greeks. On the other hand, an MCS yields far much more accurate far-field boundary values than those of the finite difference method.

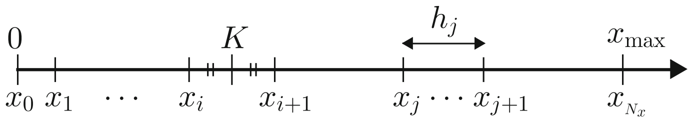

Fig. 1 A non-uniform grid with $N_x + 1$ points and grid-spacing h_i

The remainder of this paper is organized as follows. In Sect. 2, we describe the numerical solution algorithm in detail. In Sect. 3, we provide the numerical results, demonstrating the improved performance of the proposed boundary treatment. Finally, we conclude with a short summary in Sect. 4.

2 Numerical Solution

2.1 One-Dimensional BS Equation

In order to describe the basic idea of the proposed hybrid MC and finite difference method, we consider a one-asset BS equation with the initial condition $u(x, 0)$:

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

where $\tau = T - t$ and x is the value of the underlying security. The BS equation is defined on an infinite spatial domain $(0, \infty)$. However, to solve the equation numerically, we need a finite domain, $\Omega = (0, x_{\max})$, as well as appropriate boundary conditions. The FDM has been used to pricing financial options for many years (Tavella and Randall 2000). For more details on the use of the FDM in computational finance, refer to the books (Wilmott et al. 1993; Topper 2005; Duffy 2006) and to the papers (Brennan and Schwartz 1977, 1978; Schwartz 1977; Itkin and Carr 2012; Kim et al. 2013). The BS equation is discretized on a grid defined by $x_0 = 0$ and $x_{i+1} = x_i + h_i$, for $i = 0, \ldots, N_x - 1$, where N_x is the number of grid intervals and h_i is the grid spacing (see Fig. 1). For a stable calculation, two grid points around K are set to have the same interval from K. We assume that $x_{N_x} = x_{\max}$ and $h_{N_x} = h_{N_x-1}$.

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

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

Then we can rewrite Eq. (3) as

$$\alpha_i u_{i-1}^{n+1} + \beta_i u_i^{n+1} + \gamma_i u_{i+1}^{n+1} = f_i^n, \tag{4}$$

where

$$\alpha_i = \frac{-\sigma^2 x_i^2 + r x_i h_i}{h_{i-1}(h_{i-1} + h_i)}, \quad \beta_i = \frac{\sigma^2 x_i^2 - r x_i (h_i - h_{i-1})}{h_{i-1} h_i} + r + \frac{1}{\Delta\tau},$$
$$\gamma_i = \frac{-\sigma^2 x_i^2 - r x_i h_{i-1}}{h_i (h_{i-1} + h_i)}, \quad f_i^n = \frac{u_i^n}{\Delta\tau}.$$

In this study, we impose the homogeneous Dirichlet boundary condition at $x = 0$ and the MC boundary condition (MCBC) at $x = x_{\max}$. The matrix form of the linear system (4) can be rewritten as $A\mathbf{u} = \mathbf{f}$, where

$$A = \begin{pmatrix} \beta_1 & \gamma_1 & 0 & \dots & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \alpha_{N_x-1} & \beta_{N_x-1} & \gamma_{N_x-1} \\ 0 & \dots & 0 & \alpha_{N_x} & \beta_{N_x} \end{pmatrix},$$

$\mathbf{u} = (u_1^{n+1}, u_2^{n+1}, \dots, u_{N_x-1}^{n+1}, \; u_{N_x}^{n+1})^T$, and $\mathbf{f} = (f_1^n, f_2^n, \dots, f_{N_x-1}^n, f_{N_x}^n - \gamma_{N_x} u_{N_x}^{n+1})^T$. We solve this linear system using the Thomas algorithm (Fausett 2009), which inverts the tri-diagonal matrix directly. Using an MCS, we obtain an option value V at $x = x_{\max}$ and time $\tau = T$. The MCS algorithm (Boyle 1977; Boyle et al. 1997) for a European call option is

$$V = \frac{e^{-rT}}{M} \sum_{m=1}^{M} \Lambda \left(x_{\max} e^{\left(r - \frac{\sigma^2}{2}\right)T + \sigma \phi_m \sqrt{T}} \right), \tag{5}$$

where M is the number of replications, Λ is a payoff function, and ϕ_m is a sample from a standard normal distribution. In this work, unless otherwise specified, we use $M = 10^6$. Then, we define the MCBC as

$$u_{N_x}^n = u_{N_x}^0 \left(\frac{V}{u_{N_x}^0} \right)^{\frac{n\Delta\tau}{T}} \quad \text{for } n = 0, \dots, N_\tau. \tag{6}$$

This is an exponential interpolation between V and the payoff value $u_{N_x}^0$ (see Fig. 2) and we use Eq. (6) as the Dirichlet boundary condition.

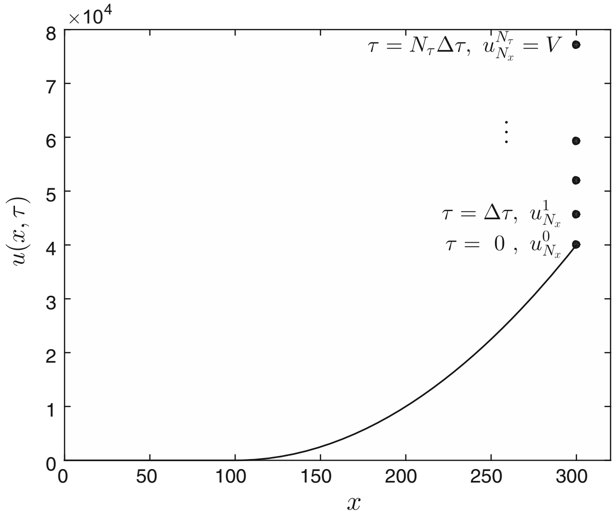

Fig. 2 Schematic illustration of the MCBC for a European powered call option. Here, V is a value derived from an MCS and $u_{N_x}^n$ are from Eq. (6) for $n = 0, \ldots, N_\tau$

2.2 Two-Dimensional BS Equation

Next, we consider the BS equation in a two-dimensional space. If $d = 2$, Eq. (2) becomes

$$\frac{\partial u}{\partial \tau} = \frac{\sigma_x^2 x^2}{2}\frac{\partial^2 u}{\partial x^2} + \frac{\sigma_y^2 y^2}{2}\frac{\partial^2 u}{\partial y^2} + \rho_{xy}\sigma_x\sigma_y xy\frac{\partial^2 u}{\partial x \partial y} + rx\frac{\partial u}{\partial x} + ry\frac{\partial u}{\partial y} - ru$$
$$\text{for } (x, y, \tau) \in (0, \infty)^2 \times (0, T]. \tag{7}$$

In Eq. (7), a finite domain Ω is considered as $(0, x_{\max}) \times (0, y_{\max})$ for the numerical tests. Let $x_0 = y_0 = 0, x_{i+1} = x_i + h_i^x$, and $y_{j+1} = y_j + h_j^y$. Let $u_{ij}^n \approx u(x_i, y_j, n\Delta\tau)$, where $i = 0, \ldots, N_x - 1$, $j = 0, \ldots, N_y - 1$, and $n = 0, \ldots, N_\tau$. Here, N_x and N_y are the numbers of grid intervals in the x- and y-direction, respectively. For an efficient numerical solver of Eq. (7), we use an operator splitting method (Ikonen and Toivanen 2004; Duffy 2006; Kim et al. 2012), which we describe using two fractional time steps, as follows:

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta\tau} = \mathcal{L}_{OS}^x u_{ij}^{n+\frac{1}{2}} + \mathcal{L}_{OS}^y u_{ij}^{n+1}, \tag{8}$$

where

$$\mathcal{L}_{OS}^x u_{ij}^{n+\frac{1}{2}} = \frac{(\sigma_x x_i)^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_{ij}^{n+\frac{1}{2}} + rx_i\left(\frac{\partial u}{\partial x}\right)_{ij}^{n+\frac{1}{2}} + \frac{1}{2}\sigma_x\sigma_y\rho_{xy}x_i y_j\left(\frac{\partial^2 u}{\partial x \partial y}\right)_{ij}^{n}$$
$$-\frac{r}{2}u_{ij}^{n+\frac{1}{2}},$$

$$\mathcal{L}_{OS}^{y}u_{ij}^{n+1} = \frac{\left(\sigma_y y_j\right)^2}{2}\left(\frac{\partial^2 u}{\partial y^2}\right)_{ij}^{n+1} + r y_j\left(\frac{\partial u}{\partial y}\right)_{ij}^{n+1} + \frac{1}{2}\sigma_x\sigma_y\rho_{xy}x_i y_j\left(\frac{\partial^2 u}{\partial x\partial y}\right)_{ij}^{n+\frac{1}{2}}$$
$$-\frac{r}{2}u_{ij}^{n+1}.$$

The spatial differential operators for u_{ij} in Eq. (8) are defined as follows:

$$\left(\frac{\partial u}{\partial x}\right)_{ij} = -\frac{h_i^x u_{i-1,j}}{h_{i-1}^x(h_{i-1}^x + h_i^x)} + \frac{(h_i^x - h_{i-1}^x)u_{ij}}{h_{i-1}^x h_i^x} + \frac{h_{i-1}^x u_{i+1,j}}{h_i^x\left(h_{i-1}^x + h_i^x\right)},$$
$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{ij} = \frac{2u_{i-1,j}}{h_{i-1}^x\left(h_{i-1}^x + h_i^x\right)} - \frac{2u_{ij}}{h_{i-1}^x h_i^x} + \frac{2u_{i+1,j}}{h_i^x\left(h_{i-1}^x + h_i^x\right)},$$
$$\left(\frac{\partial^2 u}{\partial x\partial y}\right)_{ij} = \frac{u_{i+1,j+1} - u_{i-1,j+1} - u_{i+1,j-1} + u_{i-1,j-1}}{h_i^x h_j^y + h_{i-1}^x h_j^y + h_i^x h_{j-1}^y + h_{i-1}^x h_{j-1}^y}.$$

Other derivatives are similarly defined. Then, by the operator splitting method, Eq. (8) leads to two semi-implicit discrete forms:

$$\frac{u_{ij}^{n+\frac{1}{2}} - u_{ij}^{n}}{\Delta\tau} = \mathcal{L}_{OS}^{x}u_{ij}^{n+\frac{1}{2}}, \tag{9}$$
$$\frac{u_{ij}^{n+1} - u_{ij}^{n+\frac{1}{2}}}{\Delta\tau} = \mathcal{L}_{OS}^{y}u_{ij}^{n+1}. \tag{10}$$

The numerical algorithm for Eqs. (9) and (10) is similar to the one-dimensional case. A detailed description can be found in Jeong and Kim (2013). We impose the homogeneous Dirichlet boundary condition at $(x, y) = (0, 0)$, MCBC at $(x, y) = (x_{\max}, y_{\max})$, and the LBC at the other boundaries. For example, $u_{0j} = 2u_{1j} - u_{2j}$, for $j = 1, \ldots, N_y$. Using an MCS, we obtain an option value V at $(x, y) = (x_{\max}, y_{\max})$ and time $\tau = N_t\Delta\tau$. Then, we set

$$u_{N_x,N_y}^{n} = u_{N_x,N_y}^{0}\left(\frac{V}{u_{N_x,N_y}^{0}}\right)^{\frac{n\Delta\tau}{T}} \quad \text{for} \;\; n = 0, \ldots, N_\tau. \tag{11}$$

Similarly, we can define the option value at the end point of the n-dimensional domain Ω, as follows:

$$u_{N_{x_1},N_{x_2},\ldots,N_{x_n}}^{n} = u_{N_{x_1},N_{x_2},\ldots,N_{x_n}}^{0}\left(\frac{V}{u_{N_{x_1},N_{x_2},\ldots,N_{x_n}}^{0}}\right)^{\frac{n\Delta\tau}{T}} \;\text{for} \;\; n = 0, \ldots, N_\tau.$$

3 Computational Results

We perform numerical experiments to test the performance of the proposed MCBC compared to that of the popular LBC. Here, we consider power and powered options which have nonlinear payoffs at maturity. We also present a call option based on the maximum of two assets. In all numerical tests, we use MATLAB (Mathworks Inc., Natick, MA, USA). Closed-form solutions are obtained using *Mathematica* (Wolfram 1999). The programs are executed on an Intel(R) Core(TM) i5-4430 CPU @3.00 GHz desktop PC.

3.1 Convergence Test

In this section, we provide numerical examples that illustrate the performance and convergence of the proposed MCBC. To compare the numerical results, we use the root mean squared error (RMSE). Here, the RMSE (Verkooijen 1996) is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{\#\Omega_e} \sum_{x_i \in \Omega_e} (u_i - u(x_i))^2},$$

where $\#\Omega_e$ is the total number of grid points on $\Omega_e \subset \Omega$, and u_i and $u(x_i)$ are the numerical and exact solutions, respectively.

3.1.1 European Call Option

As a first example, we apply the proposed method to a European call option with payoff function, $\max(x - K, 0)$, where K is the strike price. The closed-form solution (Black and Scholes 1973) of the European call option is given by

$$u(x, \tau) = xN(d_1) - Ke^{-r\tau}N(d_2),$$

where $N(d) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{d} e^{-0.5s^2}\,\mathrm{d}s$ is the cumulative distribution function for the normal distribution, $d_1 = [\ln(x/K) + (r - 0.5)\sigma^2)\tau]/(\sigma\sqrt{\tau})$, and $d_2 = d_1 - \sigma\sqrt{\tau}$. We set the parameter values as $K = 100$, $\sigma = 0.2$, and $r = 0.03$. The computational domain is set as $\Omega = [0,\ 300]$. Table 1a shows the RMSE on $\Omega_e = [0.7\,\mathrm{K}, 1.3\,\mathrm{K}]$ and the spatial convergence rates for the European call option with $T = 1$ and $\Delta\tau = 1/72{,}000$. Here, h denotes the uniform space step size. Table 1b indicates the RMSE and temporal convergence rates for the European call option with $T = 1$ and $h = 0.01$. The results in Table 1 show that the proposed scheme is second-order and first-order accurate in space and time, respectively, as expected from the discretization of the BS equation.

Table 1 RMSE and spatial and temporal convergence rates of the proposed scheme for the BS equation with payoff function, $\max(x-100,0)$

(a)	h	1		0.5		0.25		0.125
	RMSE	1.65e−3		4.17e−4		1.10e−4		3.35e−5
	Order		1.99		1.92		1.72	
(b)	$\Delta\tau$	1/360		1/720		1/1440		1/2880
	RMSE	1.77e−3		8.87e−4		4.43e−4		2.22e−4
	Order		0.99		0.99		0.99	

Table 2 RMSE and spatial and temporal convergence rates of the proposed scheme for the BS equation with payoff function, $\max(x^2-100,0)$

(a)	h	1		0.5		0.25		0.125
	RMSE	0.35968		0.08452		0.02076		0.00529
	Order		2.09		2.03		1.97	
(b)	$\Delta\tau$	1/360		1/720		1/1440		1/2880
	RMSE	2.97e−3		1.50e−3		7.61e−4		3.92e−4
	Order		0.99		0.98		0.96	

3.1.2 Power Option

We consider a power option with a payoff of $\max(x^p - K, 0)$, where $p \in \mathbb{R}^+$ is a power (Haug 2007). The closed-form solution (Heynen and Kat 1996; Zhang 1998; Esser 2003) of the power option is given by

$$u(x,\tau) = x^p e^{[(p-1)(r+p\sigma^2/2)]\tau} N(d_1) - K e^{-r\tau} N(d_2),$$

where $d_1 = [\ln(x/\sqrt[p]{K}) + (r + (p-0.5)\sigma^2)\tau]/(\sigma\sqrt{\tau})$, and $d_2 = d_1 - p\sigma\sqrt{\tau}$. We set the parameter values as $K = 100$, $\sigma = 0.2$, $r = 0.03$, and $p = 2$. The computational domain is set as $\Omega = [0,\ 50]$. Table 2a shows the RMSE on $\Omega_e = [0.7\sqrt[p]{K}, 1.3\sqrt[p]{K}]$ and the spatial convergence rates for a power option with $T = 1$ and $\Delta\tau = 1/3600$. Table 2b indicates the RMSE and temporal convergence rates for a power option with $T = 1$ and $h = 0.01$. The results of Table 2a, b show that the proposed scheme has second-order and first-order convergence rates in space and time, respectively.

3.1.3 Powered Option

Next, we consider a powered call option with the payoff function $\max[(x-K)^p, 0]$, where $p \in \mathbb{N}$. For this powered option (Jarrow and Turnbull 1996; Brockhaus et al. 1999), the exact solution of the BS equation is given as

$$u(x,\tau) = \sum_{j=0}^{p} {}_pC_j\, x^{p-j}(-K)^j e^{(p-j-1)(r+(p-j)\sigma^2/2)\tau} N(d_{p-j}),$$

Table 3 RMSE and spatial and temporal convergence rates of the proposed scheme for the BS equation with payoff function, $\max((x-100)^2, 0)$

(a)	h	1		0.5		0.25		0.125
	RMSE	3.79e−3		1.04e−3		2.73e−4		7.86e−5
	order		1.87		1.93		1.79	
(b)	$\Delta\tau$	1/360		1/720		1/1440		1/2880
	RMSE	0.02400		0.01215		0.00612		0.00307
	order		0.98		0.99		1.00	

Table 4 RMSE with respect to different domain sizes for a power option

$x_{\max}$	15	20	25	30	35
LBC	21.9622	8.8679	4.1777	2.0723	1.0664
MCBC	0.4963	0.1367	0.0296	0.0149	0.0101

where $d_m = [\ln(x/K) + (r + (p - j - 0.5)\sigma^2)\tau]/(\sigma\sqrt{\tau})$. In our example, we set the parameters as $K = 100$, $\sigma = 0.2$, $r = 0.03$, and $p = 2$. The computational domain is set as $\Omega = [0,\ 300]$. Table 3a shows the RMSE on $\Omega_e = [0.7\,\text{K}, 1.3\,\text{K}]$ and the spatial convergence rates for a powered option with $T = 1/360$ and $\Delta\tau = 1/360{,}000$. Table 3b shows the RMSE and temporal convergence rates for a powered option with $T = 1/36$ and $h = 0.01$. From these results, we confirm that the proposed method has second-order and first-order convergence rates in space and time, respectively.

3.2 Superiority of MCBC

To show the superiority of the MCBC over the LBC, we compare the errors obtained using the MC boundary and linear conditions imposed on the artificial boundary $x = x_{\max}$ to those of the power and powered options. For the numerical tests, we assume that $K = 100$, $\sigma = 0.5$, $r = 0.03$, $T = 1$, $h = 0.2$, $\Delta\tau = 1/720$, and $p = 2$. First, we consider the power option with the MCBC and LBC. Table 4 shows the RMSE on $\Omega_e = [0.7\sqrt[p]{K}, 1.3\sqrt[p]{K}]$ with respect to different domain sizes $x_{\max}$. As the size of the computational domain decreases, the RMSEs of both the LBC and MCBC increase rapidly because of the effect of the domain size. That is, the closed-form solutions are obtained from the solutions to the BS equation on an infinite domain. However, the numerical solutions are computed on a truncated finite domain with artificial boundary conditions. Therefore, if the computational domain is small, the boundary values have a greater effect on the numerical solutions. From Table 4, we observe that the RMSE of the MCBC is very small compared to that of the LBC.

The findings are similar for powered options. Table 5 shows the RMSE on $\Omega_e = [0.7\,\text{K}, 1.3\,\text{K}]$ for a powered option with different domain sizes. Here, we use $h = 2$, with the other parameters taking the same values as those of the power option. In this case, we see that the RMSE of the MCBC is smaller than that of the LBC.

Table 5 RMSE for a powered option with respect to different domain sizes

$x_{\max}$	150	200	250	300	500
LBC	1806.200	865.552	415.797	206.219	16.734
MCBC	232.318	100.586	38.709	15.897	0.516

As a result, the MCBC has the advantage of using a small computational domain, which enables calculations that incur small computational costs. In general, an inadequate boundary condition distorts the solution, as shown in the previous tests. However, the proposed MCBC method is useful and incurs a small computational cost, regardless of the types of options.

3.3 Call Option on the Maximum of Two Assets

Here, we consider two-asset European call option problems, which are important to financial engineers as building blocks for constructing more complex derivative products (Haug 2007; Rambeerich et al. 2013). These problems are often used to show the accuracy of a numerical scheme (Buetow and Sochacki 2000). We consider a call option on the maximum of two assets, with a payoff of $u(x, y, 0) = \max[\max(x - K, y - K), 0]$ (see Fig. 3d). Here, $K = 100, \sigma_1 = \sigma_2 = 0.3, r = 0.03, \rho = 0.5, h = 1$, $\Delta\tau = 1/720$, and $T = 1$ are used on $\Omega = (0, x_{\max}) \times (0, y_{\max})$. The closed-form solution (Haug 2007) for this option is

$$\begin{aligned} u(x, y, T) = {} & xM(d_1, d; \rho_1) + yM(d_2, -d + \sigma\sqrt{T}; \rho_2) \\ & -Ke^{-rT}[1 - M(-d_1 + \sigma_1\sqrt{T}, -d_2 + \sigma_2\sqrt{T}; \rho)], \end{aligned}$$

where the cumulative bivariate normal distribution function is defined as

$$M(a, b; \rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\infty}^{b} \int_{-\infty}^{a} \exp\left(-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}\right) dxdy.$$

The other notations are given as follows:

$$d_1 = \frac{\ln\left(\frac{x}{K}\right) + \left(r + \frac{\sigma_1^2}{2}\right)T}{\sigma_1\sqrt{T}}, \ d_2 = \frac{\ln\left(\frac{y}{K}\right) + \left(r + \frac{\sigma_2^2}{2}\right)T}{\sigma_2\sqrt{T}}, \ d = \frac{\ln\left(\frac{x}{y}\right) + \frac{\sigma^2}{2}T}{\sigma\sqrt{T}},$$
$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2}, \ \rho_1 = (\sigma_1 - \rho\sigma_2)/\sigma, \ \rho_2 = (\sigma_2 - \rho\sigma_1)/\sigma.$$

We define the error as $e_{ij} = |u(x_i, y_j, T) - u_{ij}^{N_\tau}|$. To obtain an MCBC value, we select 10^5 discrete sample paths. Figure 3a–c show the results at $T = 1$, with $(x_{\max}, y_{\max}) = (300, 300)$. for the closed-form, LBC, and MCBC solutions, respectively. Figure 3e, f show the errors from the LBC and MCBC, respectively.

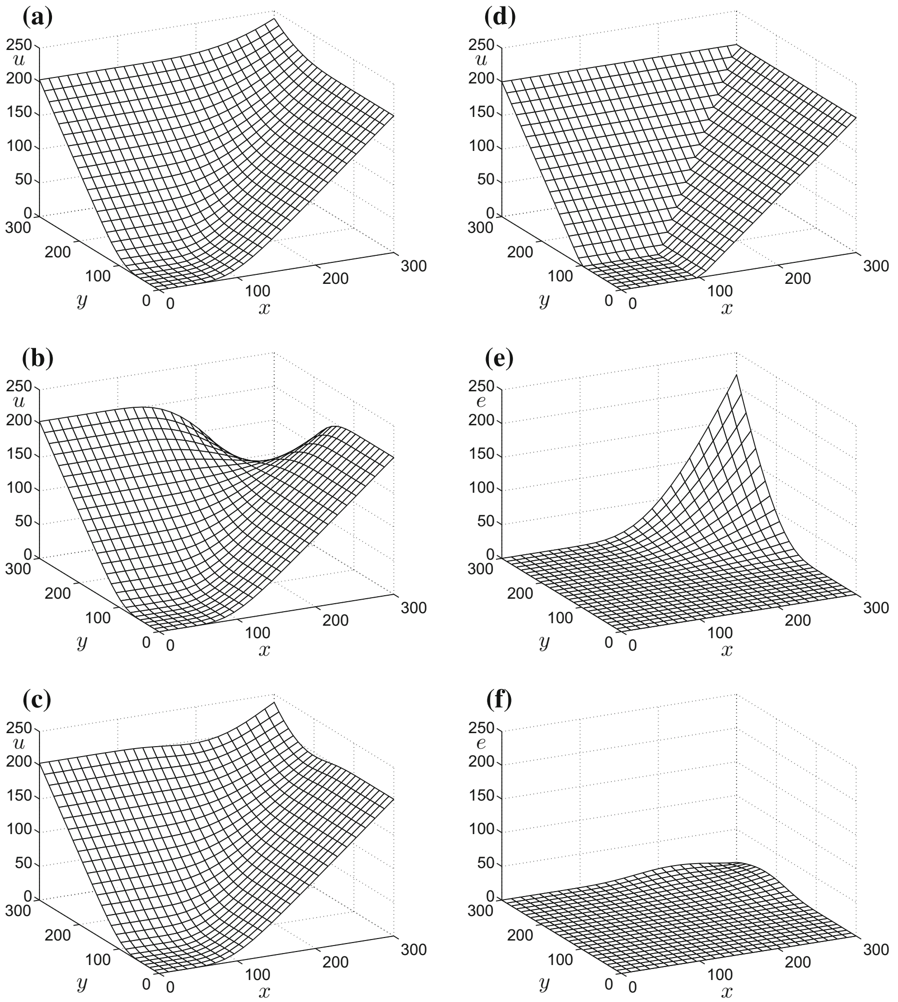

Fig. 3 Two-asset European call option: **a**, **b**, and **c** are the results at $T = 1$ with $(x_{\max}, y_{\max}) = (300, 300)$, for the closed-form, LBC, and MCBC solutions, respectively; **d** is the payoff function; and **e**, **f** are the errors from the LBC and MCBC, respectively

To compare numerical results, we calculate the RMSE as

$$\text{RMSE} = \sqrt{\frac{1}{\#\Omega_e} \sum\nolimits_{(x_i, y_j) \in \Omega_e} \left(u_{ij}^{N_t} - u(x_i, y_j, T)\right)^2}.$$

Here, $\Omega_e = [0.7\,\text{K}, 1.3\,\text{K}] \times [0.7\,\text{K}, 1.3\,\text{K}]$ is the region with the most interest, and $\#\Omega_e$ is the total number of points on Ω_e. Table 6 shows the results at $T = 1$ for different domain sizes.

Table 6 RMSE with different domain sizes L at time $T = 1$. Here, we use $\Omega = (0, L) \times (0, L)$

L	150	200	250	300
LBC	9.18213	0.82833	0.08507	0.08054
MCBC	1.43178	0.18107	0.04728	0.05023

4 Conclusions

In this study, we proposed an accurate, efficient, and robust finite difference method, with a Monte Carlo boundary condition, for solving the Black–Scholes equation. The proposed method uses a far-field boundary value obtained from a Monte Carlo simulation. A similar hybrid method with an MC simulation was presented in a conference paper (Kim et al. 2012). The latter method uses points obtained from an MC simulation as the Dirichlet boundary condition for any time, t. However, our proposed method uses only one value, with an MC simulation at boundary $\mathbf{S} = \mathbf{S}_{\max}$ and expired time $t = T$. As a result, the computational cost of the proposed method is lower than that of the aforementioned method (Kim et al. 2012). Therefore, we can confirm that the proposed method is efficient and fast. To demonstrate its superiority, we performed numerical tests on power, powered, and two-asset European call option pricing problems. Through these numerical simulations, we showed that the proposed boundary treatment yields far better accuracy and robustness than the most commonly used linear boundary condition. Furthermore, the proposed hybrid method is general and, thus, it can be applied to other types of option pricing problems (Farnoosh et al. 2015, 2016, 2017). With a small modification to the existing code, we can easily implement the proposed MCBC algorithm. It is hoped that this result will stimulate further research on MCBCs.

Acknowledgements The author (D. Jeong) was supported by a Korea University Grant. The corresponding author (J. S. Kim) was supported by the Korea Institute for Advanced Study (KIAS) for supporting research on the financial pricing model using artificial intelligence. The authors are grateful to the reviewers whose valuable suggestions and comments significantly improved the quality of this paper.

References

Bandi, C., & Bertsimas, D. (2014). Robust option pricing. *European Journal of Operational Research*, *239*(3), 842–853.

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. Pricing and hedging power options. *The Journal of Political Economy*, *15*, 637–654.

Boyle, P. (1977). Options: A Monte Carlo approach. *Journal of Financial Economics*, *4*(3), 323–338.

Boyle, P., Broadie, M., & Glasserman, P. (1997). Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control*, *21*(8), 1267–1321.

Brennan, M. J., & Schwartz, E. S. (1977). The valuation of American put options. *The Journal of Finance*, *32*(2), 449–462.

Brennan, M., & Schwartz, E. (1978). Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis. *Journal of Financial and Quantitative Analysis*, *13*(3), 461–474.

Brockhaus, O., Ferraris, A., Gallus, C., Long, D., Martin, R., & Overhaus, M. (1999). *Modelling and hedging equity derivatives*. London: Risk Books.

Buetow, G. W., & Sochacki, J. S. (2000). The trade-offs between alternative finite difference techniques used to price derivative securities. *Applied Mathematics and Computation*, *115*(2), 177–190.

Capiński, M. J. (2015). Hedging conditional value at risk with options. *European Journal of Operational Research*, *242*(2), 688–691.
d'Halluin, Y., Forsyth, P. A., & Labahn, G. (2004). A penalty method for American options with jump diffusion processes. *Numerische Mathematik*, *97*(2), 321–352.
Duffy, D. J. (2006). *Finite difference methods in financial engineering: A partial differential equation approach*. New York: Wiley.
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.
Farnoosh, R., Sobhani, A., & Beheshti, M. H. (2017). Efficient and fast numerical method for pricing discrete double barrier option by projection method. *Computers and Mathematics with Applications*, *73*(7), 1539–1545.
Farnoosh, R., Sobhani, A., Rezazadeh, H., & Beheshti, M. H. (2015). Numerical method for discrete double barrier option pricing with time-dependent parameters. *Computers and Mathematics with Applications*, *70*(8), 2006–2013.
Fausett, L. (2009). *Applied numerical analysis using MATLAB* (2nd ed.). Upper Saddle River: Prentice-Hall.
Hammersley, J. M., Handscomb, D. C., & Weiss, G. (1964). *Monte Carlo methods*. London: Springer.
Haug, E. G. (2007). *The complete guide to option pricing formulas*. New York: McGraw-Hill Companies.
Heynen, R. C., & Kat, H. M. (1996). Pricing and hedging power options. *Financial Engineering and the Japanese Markets*, *3*(3), 253–261.
Ikonen, S., & Toivanen, J. (2004). Operator splitting methods for American option pricing. *Applied Mathematics Letters*, *17*(7), 809–814.
Itkin, A., & Carr, P. (2012). Using pseudo-parabolic and fractional equations for option pricing in jump diffusion models. *Computational Economics*, *40*(1), 63–104.
Jaillet, P., Lamberton, D., & Lapeyre, B. (1990). Variational inequalities and the pricing of American options. *Acta Applicandae Mathematicae*, *21*(3), 263–289.
Jarrow, R., & Turnbull, S. (1996). *Derivative securities*. Mason: South-Western College Publishing.
Jeong, D., & Kim, J. (2013). A comparison study of ADI and operator splitting methods on option pricing models. *Journal of Computational and Applied Mathematics*, *247*, 162–171.
Kim, B. J., Ahn, C., & Choe, H. J. (2013). Direct computation for American put option and free boundary using finite difference method. *Japan Journal of Industrial and Applied Mathematics*, *30*(1), 21–37.
Kim, Y., Bae, H.-O., & Jo, T.-C. (2012). A hybrid pricing method for mult-asset options. *Proceedings of the KSIAM*, *7*(1), 309–312.
Marroquı, N., & Moreno, M. (2013). Optimizing bounds on security prices in incomplete markets. Does stochastic volatility specification matter? *European Journal of Operational Research*, *225*(3), 429–442.
Pun, C. S., Chung, S. F., & Wong, H. Y. (2015). Variance swap with mean reversion, multifactor stochastic colatility and jumps. *European Journal of Operational Research*, *245*(2), 571–580.
Rambeerich, N., Tangman, D. Y., Lollchund, M. R., & Bhuruth, M. (2013). High-order computational methods for option valuation under multifactor models. *European Journal of Operational Research*, *224*(1), 219–226.
Schwartz, E. S. (1977). The valuation of warrants: Implementing a new approach. *Journal of Financial Economics*, *4*(1), 79–93.
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.
Topper, J. (2005). *Financial engineering with finite elements*. New York: Wiley.
Verkooijen, W. (1996). A neural network approach to long-run exchange rate prediction. *Computational Economics*, *9*(1), 51–65.
Wilmott, P., Dewynne, J., & Howison, S. (1993). *Option pricing: Mathematical models and computation*. Oxford: Oxford Financial Press.
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*(1), 65–92.
Wolfram, S. (1999). *The mathematica book version 4*. Cambridge: Cambridge University Press.

Yang, X., Liu, Y., & Wang, G. (2007). A study on a new kind of universal difference scheme for solving Black–Scholes equation. *International Journal of Numerical Analysis and Modeling*, *3*(2), 251–260.
Zhang, P. G. (1995). An introduction to exotic options. *European Financial Management*, *1*(1), 87–95.
Zhang, P. G. (1998). *Exotic options: A guide to second generation options* (2nd ed.). Singapore: World Scientific.