Commun. Korean Math. Soc. **30** (2015), No. 3, pp. 297–311
http://dx.doi.org/10.4134/CKMS.2015.30.3.297

ROBUST AND ACCURATE METHOD FOR THE BLACK–SCHOLES EQUATIONS WITH PAYOFF-CONSISTENT EXTRAPOLATION

Yongho Choi, Darae Jeong, Junseok Kim, Young Rock Kim,
Seunggyu Lee, Seungsuk Seo, and Minhyun Yoo

Reprinted from the
Communications of the Korean Mathematical Society
Vol. 30, No. 3, July 2015

Commun. Korean Math. Soc. **30** (2015), No. 3, pp. 297–311
http://dx.doi.org/10.4134/CKMS.2015.30.3.297

ROBUST AND ACCURATE METHOD FOR THE BLACK–SCHOLES EQUATIONS WITH PAYOFF-CONSISTENT EXTRAPOLATION

Yongho Choi, Darae Jeong, Junseok Kim, Young Rock Kim,
Seunggyu Lee, Seungsuk Seo, and Minhyun Yoo

Abstract. We present a robust and accurate boundary condition for pricing financial options that is a hybrid combination of the payoff-consistent extrapolation and the Dirichlet boundary conditions. The payoff-consistent extrapolation is an extrapolation which is based on the payoff profile. We apply the new hybrid boundary condition to the multi-dimensional Black–Scholes equations with a high correlation. Correlation terms in mixed derivatives make it more difficult to get stable numerical solutions. However, the proposed new boundary treatments guarantee the stability of the numerical solution with high correlation. To verify the excellence of the new boundary condition, we have several numerical tests such as higher dimensional problem and exotic option with nonlinear payoff. The numerical results demonstrate the robustness and accuracy of the proposed numerical scheme.

1. Introduction

Let $s_i(t) \in \mathbb{R}_+$, $i = 1, 2, \ldots, d$ denote the value of the i-th underlying asset at time t and $u(\mathbf{s}, t)$ denote the price of an option, where $\mathbf{s} = (s_1, s_2, \ldots, s_d)$. Then the option price follows the following generalized Black–Scholes (BS) partial differential equation (PDE): For $(\mathbf{s}, t) \in \mathbb{R}_+^d \times (0, T)$,

$$(1)\qquad \frac{\partial u(\mathbf{s},t)}{\partial t} + \sum_{i=1}^{d} r s_i \frac{\partial u(\mathbf{s},t)}{\partial s_i} + \frac{1}{2}\sum_{i,j=1}^{d} \rho_{ij}\sigma_i\sigma_j s_i s_j \frac{\partial^2 u(\mathbf{s},t)}{\partial s_i \partial s_j} - r u(\mathbf{s},t) = 0$$

Received February 10, 2015; Revised June 1, 2015.

2010 *Mathematics Subject Classification.* Primary 91G60, 65N06.

Key words and phrases. multi-dimensional Black–Scholes equations, operator splitting method, extrapolation, linear boundary condition, high correlation.

The author (D. Jeong) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2014R1A6A3A01009812). This research was supported by the research fund of Hankuk University of Foreign Studies. The authors (S. Seo and J. S. Kim) were supported by The Small and Medium Business Administration.

with final condition $u(\mathbf{s}, T) = \Lambda(\mathbf{s})$, where r is the constant riskless interest rate, σ_i are volatilities of s_i, and ρ_{ij} are the asset correlations between s_i and s_j. $\Lambda(\mathbf{s})$ is the payoff function at maturity T [12]. There are three classical techniques such as the finite difference method (FDM) [1, 13], the finite element method [4, 8, 9], and the finite volume method [16] for the numerical solution of the BS PDE. In this paper, we will focus on the FDM. When we solve the multi-dimensional BS PDE, we typically treat the mixed derivative terms having correlation coefficients explicitly because of the difficulty to solve them implicitly. The value of the mixed derivative term $\rho_{ij}\sigma_i\sigma_j s_i s_j u_{s_i s_j}$ in Eq. (1) is dependent on the values of correlation, volatilities, and underlying assets. In order to achieve the stability of the numerical schemes, it is important to treat the mixed derivative term appropriately because the large value of the mixed derivative term severely restricts the numerical time step size and the numerical solution undergoes oscillation or blow-up unless we use small enough time steps. The main purpose of this article is to propose a new robust numerical boundary condition that is combination of the payoff-consistent extrapolation and the Dirichlet boundary conditions. We apply the new boundary condition to solving the multi-dimensional BS equations with high correlation values.

The contents of this paper are as follows. In Section 2, we describe the solution algorithm, which is the operator splitting method. In Section 2.2, we propose a new numerical method which is based on a payoff-consistent extrapolation and the Dirichlet boundary conditions. We present several numerical experiments to show the robustness and accuracy of the proposed numerical method in Section 3. Finally, conclusions are drawn in Section 4.

2. Numerical solution

In this section, we describe the discretization, the operator splitting scheme, and the payoff-consistent extrapolation and the Dirichlet boundary condition.

2.1. Discretization

The two-dimensional version of Eq. (1) can be written as

$$\frac{\partial u}{\partial \tau} = \mathcal{L}_{BS} u \quad \text{for } (x, y, \tau) \in \Omega \times (0, T], \tag{2}$$

where $\tau = T - t$ and $\mathcal{L}_{BS} u = 0.5\sigma_1^2 x^2 u_{xx} + 0.5\sigma_2^2 y^2 u_{yy} + \rho\sigma_1\sigma_2 xy u_{xy} + rxu_x + ryu_y - ru$. Equation (2) is defined in $\Omega \times (0, T] = \{(x, y, t) \mid x > 0,\ y > 0,\ \tau \in (0, T]\}$. However, we need a finite domain to solve Eq. (2) by using a FDM. Let us discretize the computational domain $\Omega = (0, L) \times (0, M)$ with a uniform space step $h = L/N_x = M/N_y$ and a time step $\Delta\tau = T/N_\tau$. Here, N_x, N_y, and N_τ are the number of grid points in the x-, y-, and τ-direction, respectively. Let us introduce the notation $u_{ij}^n \equiv u(x_i, y_j, \tau^n) = u(ih, jh, n\Delta\tau)$, where $i = 0, \ldots, N_x$, $j = 0, \ldots, N_y$, and $n = 0, \ldots, N_\tau$. In Fig. 1, the marked dots denote the boundary points.

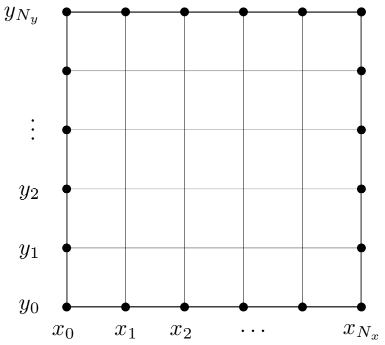

FIGURE 1. Discrete computational domain with marked boundary points.

We use the operator splitting (OS) method [2, 7, 15] to solve Eq. (2):

$$\frac{u_{ij}^{n+\frac{1}{2}} - u_{ij}^n}{\Delta\tau} = \mathcal{L}_{OS}^x u_{ij}^{n+\frac{1}{2}}, \tag{3}$$

$$\frac{u_{ij}^{n+1} - u_{ij}^{n+\frac{1}{2}}}{\Delta\tau} = \mathcal{L}_{OS}^y u_{ij}^{n+1}, \tag{4}$$

where the discrete difference operators $\mathcal{L}_{OS}^x$ and $\mathcal{L}_{OS}^y$ are defined by

$$\begin{aligned}\mathcal{L}_{OS}^x u_{ij}^{n+\frac{1}{2}} = {} & \frac{(\sigma_1 x_i)^2}{2} \frac{u_{i-1,j}^{n+\frac{1}{2}} - 2u_{ij}^{n+\frac{1}{2}} + u_{i+1,j}^{n+\frac{1}{2}}}{h^2} + rx_i \frac{u_{i+1,j}^{n+\frac{1}{2}} - u_{i-1,j}^{n+\frac{1}{2}}}{2h} - \frac{r}{2} u_{ij}^{n+\frac{1}{2}} \\ & + \frac{1}{2}\sigma_1\sigma_2\rho x_i y_j \frac{u_{i+1,j+1}^n + u_{i-1,j-1}^n - u_{i-1,j+1}^n - u_{i+1,j-1}^n}{4h^2},\end{aligned} \tag{5}$$

$$\begin{aligned}\mathcal{L}_{OS}^y u_{ij}^{n+1} = {} & \frac{(\sigma_2 y_j)^2}{2} \frac{u_{i,j-1}^{n+1} - 2u_{ij}^{n+1} + u_{i,j+1}^{n+1}}{h^2} + ry_j \frac{u_{i,j+1}^{n+1} - u_{i,j-1}^{n+1}}{2h} - \frac{r}{2} u_{ij}^{n+1} \\ & + \frac{1}{2}\sigma_1\sigma_2\rho x_i y_j \frac{u_{i+1,j+1}^{n+\frac{1}{2}} + u_{i-1,j-1}^{n+\frac{1}{2}} - u_{i-1,j+1}^{n+\frac{1}{2}} - u_{i+1,j-1}^{n+\frac{1}{2}}}{4h^2}.\end{aligned} \tag{6}$$

If we add Eqs. (3) and (4), then we have

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta\tau} = \mathcal{L}_{OS}^x u_{ij}^{n+\frac{1}{2}} + \mathcal{L}_{OS}^y u_{ij}^{n+1}. \tag{7}$$

The solution algorithm using the OS method is as follows: First, we rewrite Eq. (3) as $\alpha_i u_{i-1j}^{n+\frac{1}{2}} + \beta_i u_{ij}^{n+\frac{1}{2}} + \gamma_i u_{i+1j}^{n+\frac{1}{2}} = f_{ij}$, where

$$\alpha_i = -\frac{\sigma_1^2 x_i^2}{2h^2} + \frac{rx_i}{2h},\ \beta_i = \frac{1}{\Delta\tau} + \frac{\sigma_1^2 x_i^2}{h^2} + \frac{r}{2},\ \gamma_i = -\frac{\sigma_1^2 x_i^2}{2h^2} - \frac{rx_i}{2h},$$
$$f_{ij} = \frac{1}{2}\rho\sigma_1\sigma_2 x_i y_j \frac{u_{i+1,j+1}^n - u_{i+1,j-1}^n - u_{i-1,j+1}^n + u_{i-1,j-1}^n}{4h^2} + \frac{u_{ij}^n}{\Delta\tau}.$$

Here the boundary values are obtained from a payoff-consistent extrapolation, which will be explained later. Then the system of discrete equation can be rewritten as

$$\begin{pmatrix} \beta_1 & \gamma_1 & 0 & \dots & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \alpha_{N_x-2} & \beta_{N_x-2} & \gamma_{N_x-2} \\ 0 & \dots & 0 & \alpha_{N_x-1} & \beta_{N_x-1} \end{pmatrix} \begin{pmatrix} u_{1,j}^{n+\frac{1}{2}} \\ u_{2,j}^{n+\frac{1}{2}} \\ \vdots \\ u_{N_x-2,j}^{n+\frac{1}{2}} \\ u_{N_x-1,j}^{n+\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} f_{1,j}^* \\ f_{2,j} \\ \vdots \\ f_{N_x-2,j} \\ f_{N_x-1,j}^* \end{pmatrix},$$

where $f_{1,j}^*{=}f_{1,j} - \alpha_1 u_{0,j}^n$ and $f_{N_x-1,j}^*{=}f_{N_x-1,j} - \gamma_{N_x-1} u_{N_x,j}^n$. We solve the discrete system of equations by using the Thomas algorithm. Next, we rewrite Eq. (4) as $\alpha_j u_{ij-1}^{n+1} + \beta_j u_{ij}^{n+1} + \gamma_j u_{ij+1}^{n+1} = g_{ij}$, where

$$\alpha_j = -\frac{\sigma_2^2 y_j^2}{2h^2} + \frac{r y_j}{2h},\ \beta_j = \frac{1}{\Delta\tau} + \frac{\sigma_2^2 y_j^2}{h^2} + \frac{r}{2},\ \gamma_j = -\frac{\sigma_2^2 y_j^2}{2h^2} - \frac{r y_j}{h},$$

$$g_{ij} = \frac{1}{2}\sigma_1\sigma_2\rho x_i y_j \frac{u_{i+1,j+1}^{n+\frac{1}{2}} + u_{i-1,j-1}^{n+\frac{1}{2}} - u_{i-1,j+1}^{n+\frac{1}{2}} - u_{i+1,j-1}^{n+\frac{1}{2}}}{4h^2} + \frac{u_{ij}^{n+\frac{1}{2}}}{\Delta\tau}.$$

We have

$$\begin{pmatrix} \beta_1 & \gamma_1 & 0 & \dots & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \alpha_{N_y-2} & \beta_{N_y-2} & \gamma_{N_y-2} \\ 0 & \dots & 0 & \alpha_{N_y-1} & \beta_{N_y-1} \end{pmatrix} \begin{pmatrix} u_{i,1}^{n+1} \\ u_{i,2}^{n+1} \\ \vdots \\ u_{i,N_y-2}^{n+1} \\ u_{i,N_y-1}^{n+1} \end{pmatrix} = \begin{pmatrix} g_{i,1}^* \\ g_{i,2} \\ \vdots \\ g_{i,N_y-2} \\ g_{i,N_y-1}^* \end{pmatrix},$$

where $g_{i,1}^*{=}g_{i,1} - \alpha_1 u_{i,0}^{n+\frac{1}{2}}$ and $g_{i,N_y-1}^*{=}g_{i,N_y-1} - \gamma_{N_y-1} u_{i,N_y}^{n+\frac{1}{2}}$.

2.2. Payoff-consistent and Dirichlet boundary condition

In this section, we describe the proposed new hybrid boundary condition in detail. For easy explanation, we consider a vanilla call option. As shown in Fig. 2, the payoff of the option on $\Omega = (0, L) \times (0, L)$ is given as

$$\Lambda(x, y) = \max\{x - K, y - K, 0\}. \tag{8}$$

First, we consider the conventional linear boundary condition which is mostly used in financial engineering. Linear boundary condition is defined that the second derivative of the solution across a boundary is zero [10]. For example, at $x = L$ we have $u_{N_x,j}^n = 2u_{N_x-1,j}^n - u_{N_x-2,j}^n$ for $j = 1, \dots, N_y - 1$. Next, we consider a new hybrid boundary condition that is based on the payoff-consistent extrapolation and the Dirichlet boundary conditions. For example, at $x = L$ we use the common linear boundary condition, i.e., $u_{N_x,j}^n = 2u_{N_x-1,j}^n - u_{N_x-2,j}^n$ for $j = 0, \dots, N_y - 2$. However, if we apply the linear extrapolation at $j = N_y - 1$, then we have $u_{N_x,N_y-1}^0 = 2u_{N_x-1,N_y-1}^0 - u_{N_x-2,N_y-1}^0 = u_{N_x-1,N_y-1}^0 = L -$

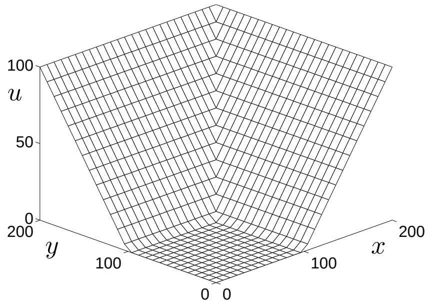

FIGURE 2. European call option payoff on the maximum of two assets. Here, $K = 100$ and $L = 200$.

$K - h \neq \Lambda(L, L-h) = L - K$. To correct this discrepancy, we propose the payoff-consistent extrapolation as the following stencil of the extrapolation: $u^n_{N_x,N_y-1} = 2u^n_{N_x-1,N_y-2} - u^n_{N_x-2,N_y-3}$, which is consistent with the payoff function when $n = 0$, i.e., $u^0_{N_x,N_y-1} = 2u^0_{N_x-1,N_y-2} - u^0_{N_x-2,N_y-3} = L - K = \Lambda(L, L-h)$. Similarly, we apply the payoff-consistent extrapolation at $u^n_{N_x,N_y}$ and $u^n_{N_x-1,N_y}$. When we solve Eq. (3), we use the boundary values from the payoff-consistent extrapolation at the time level n. Likewise, for Eq. (4), we use the boundary values from the time level $n+\frac{1}{2}$. To highlight the effect of the proposed payoff-consistent extrapolation, let us consider the following mixed derivative term in the BS equation:

$$\rho\sigma_1\sigma_2 xy\frac{\partial^2 u}{\partial x\partial y}. \tag{9}$$

Let us compute the discrete approximations of Eq. (9) at (x_{N_x-1}, y_{N_y-1}) using the conventional linear extrapolation and the proposed stencils as shown in Figs. 3(a) and (b), respectively.

Using the linear boundary extrapolation, we have

$$\begin{aligned}\left(\frac{\partial^2 u}{\partial x\partial y}\right)^0_{N_x-1,N_y-1} &= \frac{u^0_{N_x,N_y} + u^0_{N_x-2,N_y-2} - u^0_{N_x,N_y-2} - u^0_{N_x-2,N_y}}{4h^2}\\ &= \frac{(L-K-2h) + (L-K-2h) - (L-K) - (L-K)}{4h^2}\\ &= -\frac{1}{h}.\end{aligned}$$

On the other hand, using the proposed stencil, we have

$$\begin{aligned}\left(\frac{\partial^2 u}{\partial x\partial y}\right)^0_{N_x-1,N_y-1} &= \frac{u^0_{N_x,N_y} + u^0_{N_x-2,N_y-2} - u^0_{N_x,N_y-2} - u^0_{N_x-2,N_y}}{4h^2}\\ &= \frac{(L-K) + (L-K-2h) - (L-K) - (L-K)}{4h^2}\end{aligned}$$

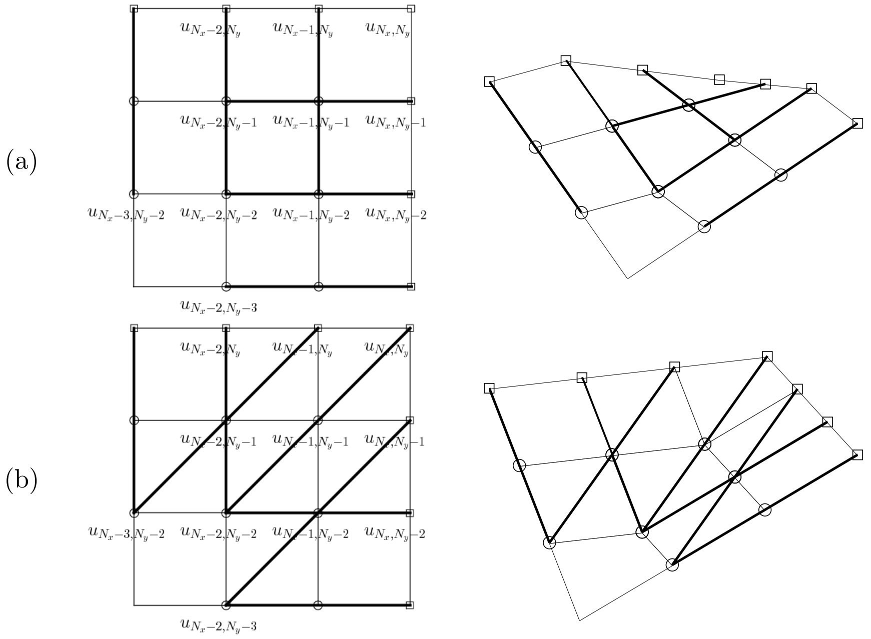

FIGURE 3. Schematic of (a) conventional linear and (b) new hybrid extrapolations for finding boundary points.

$$= -\frac{1}{2h},$$

which is consistent with the value obtained from using the payoff function at the boundary points. This consistent value is important because the coefficient of the mixed derivative term is large, i.e., $\rho\sigma_1\sigma_2 L^2$.

2.3. Extension to a three-dimensional problem

In this section, we consider a three-dimensional discretized Black–Scholes problem:

$$\frac{u_{ijk}^{n+1} - u_{ijk}^{n}}{\Delta\tau} = (\mathcal{L}_{BS}^{x} u)_{ijk}^{n+\frac{1}{3}} + (\mathcal{L}_{BS}^{y} u)_{ijk}^{n+\frac{2}{3}} + (\mathcal{L}_{BS}^{z} u)_{ijk}^{n+1}, \tag{10}$$

where the discrete difference operators $\mathcal{L}_{BS}^{x}$, $\mathcal{L}_{BS}^{y}$, and $\mathcal{L}_{BS}^{z}$ are defined by

$$\begin{aligned}
(\mathcal{L}_{BS}^{x} u)_{ijk}^{n+\frac{1}{3}} &= \frac{(\sigma_x x_i)^2}{2} D_{xx} u_{ijk}^{n+\frac{1}{3}} + r x_i D_x u_{ijk}^{n+\frac{1}{3}} + \frac{1}{3}\sigma_x\sigma_y\rho_{xy} x_i y_j D_{xy} u_{ijk}^{n} \\
&\quad + \frac{1}{3}\sigma_y\sigma_z\rho_{yz} y_j z_k D_{yz} u_{ijk}^{n} + \frac{1}{3}\sigma_z\sigma_x\rho_{zx} z_k x_i D_{zx} u_{ijk}^{n} - \frac{1}{3} r u_{ijk}^{n+\frac{1}{3}}, \\
(\mathcal{L}_{BS}^{y} u)_{ijk}^{n+\frac{2}{3}} &= \frac{(\sigma_y y_j)^2}{2} D_{yy} u_{ijk}^{n+\frac{2}{3}} + r y_j D_y u_{ijk}^{n+\frac{2}{3}} + \frac{1}{3}\sigma_x\sigma_y\rho_{xy} x_i y_j D_{xy} u_{ijk}^{n+\frac{1}{3}}
\end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{3}\sigma_y\sigma_z\rho_{yz}y_jz_kD_{yz}u_{ijk}^{n+\frac{1}{3}}+\frac{1}{3}\sigma_z\sigma_x\rho_{zx}z_kx_iD_{zx}u_{ijk}^{n+\frac{1}{3}}-\frac{1}{3}ru_{ijk}^{n+\frac{2}{3}},\\
(\mathcal{L}_{BS}^z u)_{ijk}^{n+1} = {}&\frac{(\sigma_z z_k)^2}{2}D_{zz}u_{ijk}^{n+1}+rz_kD_zu_{ijk}^{n+1}+\frac{1}{3}\sigma_x\sigma_y\rho_{xy}x_iy_jD_{xy}u_{ijk}^{n+\frac{2}{3}}\\
&+\frac{1}{3}\sigma_y\sigma_z\rho_{yz}y_jz_kD_{yz}u_{ijk}^{n+\frac{2}{3}}+\frac{1}{3}\sigma_z\sigma_x\rho_{zx}z_kx_iD_{zx}u_{ijk}^{n+\frac{2}{3}}-\frac{1}{3}ru_{ijk}^{n+1}.
\end{aligned}
$$

For the discretization of the space variables in Eq. (10), we employ the following difference equations:

$$
\begin{aligned}
D_xu_{ijk} &= \frac{u_{i+1,jk}-u_{i-1,jk}}{2h}, \quad D_{xx}u_{ijk} = \frac{u_{i-1,jk}-2u_{ijk}+u_{i+1,jk}}{h^2},\\
D_{xy}u_{ijk} &= \frac{u_{i+1,j+1,k}-u_{i-1,j+1,k}-u_{i+1,j-1,k}+u_{i-1,j-1,k}}{4h^2}.
\end{aligned}
$$

Then, OS method consists of the following three discrete equations

$$\frac{u_{ijk}^{n+\frac{1}{3}}-u_{ijk}^{n}}{\Delta\tau} = (\mathcal{L}_{BS}^x u)_{ijk}^{n+\frac{1}{3}}, \tag{11}$$

$$\frac{u_{ijk}^{n+\frac{2}{3}}-u_{ijk}^{n+\frac{1}{3}}}{\Delta\tau} = (\mathcal{L}_{BS}^y u)_{ijk}^{n+\frac{2}{3}}, \tag{12}$$

$$\frac{u_{ijk}^{n+1}-u_{ijk}^{n+\frac{2}{3}}}{\Delta\tau} = (\mathcal{L}_{BS}^z u)_{ijk}^{n+1}. \tag{13}$$

To condense the discussion, we only describe the numerical solution algorithm to solve Eq. (11). Discrete Eqs. (12) and (13) are similarly solved. Given u_{ijk}^n, Eq. (11) is rewritten as follows:

$$\alpha_i u_{i-1,jk}^{n+\frac{1}{3}}+\beta_i u_{ijk}^{n+\frac{1}{3}}+\gamma_i u_{i+1,jk}^{n+\frac{1}{3}} = f_{ijk} \quad \text{for } i=1,\dots,N_x-1, \tag{14}$$

where

$$
\begin{aligned}
\alpha_i &= -\frac{(\sigma_x x_i)^2}{2h^2}+\frac{rx_i}{2h},\ \beta_i = \frac{(\sigma_x x_i)^2}{h^2}+\frac{r}{3}+\frac{1}{\Delta\tau},\ \gamma_i = -\frac{(\sigma_x x_i)^2}{2h^2}-\frac{rx_i}{2h},\\
f_{ijk} &= \frac{1}{3}\sigma_x\sigma_y\rho_{xy}x_iy_jD_{xy}u_{ijk}^n+\frac{1}{3}\sigma_y\sigma_z\rho_{yz}y_jz_kD_{yz}u_{ijk}^n\\
&\quad+\frac{1}{3}\sigma_x\sigma_z\rho_{zx}x_iz_kD_{zx}u_{ijk}^n-\frac{1}{\Delta\tau}u_{ijk}^n.
\end{aligned}
$$

For fixed index j and k, we solve the following tridiagonal system

$$
\begin{pmatrix}
\beta_1 & \gamma_1 & 0 & \dots & 0\\
\alpha_2 & \beta_2 & \gamma_2 & \dots & 0\\
\vdots & \ddots & \ddots & \ddots & \vdots\\
0 & \dots & \alpha_{N_x-2} & \beta_{N_x-2} & \gamma_{N_x-2}\\
0 & \dots & 0 & \alpha_{N_x-1} & \beta_{N_x-1}
\end{pmatrix}
\begin{pmatrix}
u_{1,jk}^{n+\frac{1}{3}}\\
u_{2,jk}^{n+\frac{1}{3}}\\
\vdots\\
u_{N_x-2,jk}^{n+\frac{1}{3}}\\
u_{N_x-1,jk}^{n+\frac{1}{3}}
\end{pmatrix}
=
\begin{pmatrix}
f_{1,jk}^*\\
f_{2,jk}\\
\vdots\\
f_{N_x-2,jk}\\
f_{N_x-1,jk}^*
\end{pmatrix},
$$

where $f^*_{1,jk}$=$f_{1,jk} - \alpha_1 u^n_{0,jk}$ and $f^*_{N_x-1,jk}$=$f_{N_x-1,jk} - \gamma_{N_x-1} u^n_{N_x,jk}$. We solve the discrete system of equations by using the Thomas algorithm. By the new hybrid boundary method, we impose the following condition at boundary:

$$\begin{aligned}
&u_{0,jk} = 2u_{1,jk} - u_{2,jk}, u_{N_x,jk} = 2u_{N_x-1,jk} - u_{N_x-2,jk},\\
&u_{N_x,j,N_z} = 2u_{N_x-1,j,N_z-1} - u_{N_x-2,j,N_z-2},\\
&u_{N_x-1,j,N_z} = 2u_{N_x-2,j,N_z-1} - u_{N_x-3,j,N_z-2},\\
&u_{N_x,j,N_z-1} = 2u_{N_x-1,j,N_z-2} - u_{N_x-2,j,N_z-3}.
\end{aligned}$$

The other boundaries are similarly defined.

3. Numerical experiments

In this section, we present the numerical results to compare the performances using two different boundary conditions. All computations were performed using MATLAB version 7.6 [11]. The error of the numerical solution is defined as $e_{ij} = u^e_{ij} - u_{ij}$, where u^e_{ij} is the exact solution and u_{ij} is the numerical solution. To compare the errors of different methods, we compute the root mean square error (RMSE) on an interested region. The RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{(x_i,y_j)\in\Omega_0}^{N} e^2_{ij}},$$

where N is the number of points on the gray region as shown in Fig. 4 and the region indicates a neighborhood, $\Omega_0 = [0.7X_1, 1.3X_1] \times [0.7X_2, 1.3X_2]$, of the exercise prices X_1 and X_2.

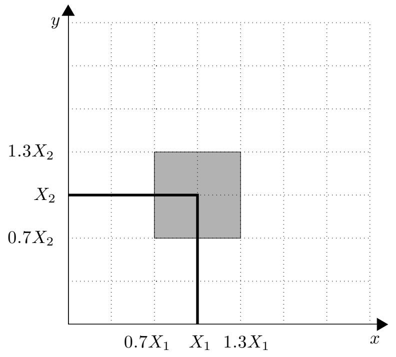

FIGURE 4. RMSE is calculated on the gray region.

In all numerical tests, unless otherwise specified, we use the following parameters: $r = 0.03$, $\sigma_1 = \sigma_2 = 0.3$, and $T = 1$ on the computational domain

$\Omega = [0, L] \times [0, L]$. The closed-form solution [5] for option value with the payoff Eq. (8) is given by

$$\begin{aligned} u(x, y, T) &= xe^{-rT} M(d_1, d; \rho_1) + ye^{-rT} M(d_2, -d + \sigma\sqrt{T}; \rho_2) \\ &\quad - Xe^{-rT}[1 - M(-d_1 + \sigma_1\sqrt{T}, d_2 + \sigma_2\sqrt{T}; \rho)], \end{aligned} \tag{15}$$

where

$$d = \frac{\ln(x/y) + \sigma^2 T/2}{\sigma\sqrt{T}},\ d_1 = \frac{\ln(x/X) + \sigma_1^2 T/2}{\sigma_1\sqrt{T}},\ d_2 = \frac{\ln(y/X) + \sigma_2^2 T/2}{\sigma_2\sqrt{T}},$$

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2},\ \rho_1 = (\sigma_1 - \rho\sigma_2)/\sigma, \text{ and } \rho_2 = (\sigma_2 - \rho\sigma_1)/\sigma.$$

In Eq. (15), $M(a, b; \rho)$ is the standard cumulative normal distribution function as

$$M(a, b; \rho) = \frac{1}{2\pi\sqrt{1 - \rho^2}} \int_{-\infty}^{a} \int_{-\infty}^{b} \exp\left[-\frac{x^2 - 2\rho xy + y^2}{2(1 - \rho^2)}\right] dx\ dy. \tag{16}$$

The following is a MATLAB code for the closed-form solution (15):

```
% Call option on the maximum of two assets
clear;sigma1=0.3;sigma2=0.3;r=0.03;rho=0.5;T=0.5;X=100;L=300;M=300;Nx=31;Ny=31;
sig=sqrt(sigma1^2+sigma2^2-2*rho*sigma1*sigma2);rho1=(sigma1-rho*sigma2)/sig;
rho2=(sigma2-rho*sigma1)/sig;x=linspace(0,L,Nx+1);y=linspace(0,M,Ny+1);
for i=1:Nx+1
    for j=1:Ny+1
      d=(log(x(i)/y(j))+0.5*sig^2*T)/(sig*sqrt(T));
      d1=(log(x(i)/X)+0.5*sigma1^2*T)/(sigma1*sqrt(T));
      d2=(log(y(j)/X)+0.5*sigma2^2*T)/(sigma2*sqrt(T));
      V(i,j)=x(i)*mvncdf([d1 d],[0 0],[1 rho1; rho1 1])  ...
            +y(j)*mvncdf([d2 -d+sig*sqrt(T)],[0 0],[1 rho2; rho2 1])  ...
            -X*exp(-r*T)*(1-mvncdf([-d1+sigma1*sqrt(T)  ...
            -d2+sigma2*sqrt(T)],[0 0],[1 rho; rho 1]));
    end
end V(1,1)=0; surf(x,y,V)
```

Table 1 shows RMSE on $\Omega_0 = [0.7X_1, 1.3X_1] \times [0.7X_2, 1.3X_2]$ with various values of L, ρ, $\Delta\tau$, and h. As shown in Table 1, RMSEs with the conventional linear boundary condition are relatively bigger than those with the new hybrid boundary condition. Especially, with our proposed hybrid scheme we have the significantly smaller RMSEs compared to the conventional linear boundary condition under the high correlation and smaller domain size (see the column of $L = 160$ and $\rho = 0.8$).

Table 2 shows option values at the point $(x, y) = (100, 100)$ with various values of L, ρ, $\Delta\tau$, and h. The values inside the parenthesis are the closed-form solutions. We can observe that the hybrid scheme is more accurate when correlation is high and domain size is small (see the column of $L = 160$ and $\rho = 0.8$).

Figure 5 shows numerical results at time $T = 1$ with (a) $\rho = 0.2$, (b) $\rho = 0.5$, and (c) $\rho = 0.8$. The first, second, and third rows are from closed-form solution, conventional linear boundary, and new hybrid boundary conditions

Table 1. RMSE on $\Omega_0 = [0.7X_1, 1.3X_1] \times [0.7X_2, 1.3X_2]$ with various values of L, ρ, $\Delta\tau$, and h.

		$L = 160$			$L = 300$		
		$\rho = 0.2$	$\rho = 0.5$	$\rho = 0.8$	$\rho = 0.2$	$\rho = 0.5$	$\rho = 0.8$
		Conventional linear boundary condition					
$\Delta\tau$	h						
1/180	2	2.3930545	5.8624586	15.3996467	0.0422851	0.0409698	0.0612859
1/360	1	2.2371132	5.9692506	18.3127686	0.0501792	0.0477070	0.0763009
1/720	0.5	2.1712868	6.2847185	22.6034383	0.0530512	0.0501025	0.1011170
		New hybrid boundary condition					
$\Delta\tau$	h						
1/180	2	1.2460393	1.2238070	0.9422444	0.0413479	0.0406781	0.0420729
1/360	1	1.2936015	1.2443000	0.9134185	0.0491920	0.0478605	0.0474969
1/720	0.5	1.3589632	1.3003881	0.9505281	0.0520435	0.0505581	0.0495902

Table 2. Option values at the point $(x, y) = (100, 100)$ with various values of L, ρ, $\Delta\tau$, and h.

		$L = 160$			$L = 300$		
		$\rho = 0.2$	$\rho = 0.5$	$\rho = 0.8$	$\rho = 0.2$	$\rho = 0.5$	$\rho = 0.8$
		(22.08152)	(20.28972)	(17.73092)	(22.08152)	(20.28972)	(17.73092)
		Conventional linear boundary condition					
$\Delta\tau$	h						
1/180	2	21.37668	17.94798	10.18170	22.12432	20.33081	17.77331
1/360	1	21.43003	17.87647	8.55308	22.13635	20.34217	17.77218
1/720	0.5	21.45117	17.72268	6.21900	22.14078	20.34659	17.76964
		New hybrid boundary condition					
$\Delta\tau$	h						
1/180	2	21.66787	19.80599	17.34698	22.12405	20.33106	17.78158
1/360	1	21.65211	19.80128	17.35894	22.13609	20.34248	17.78379
1/720	0.5	21.62693	19.77727	17.33704	22.14052	20.34700	17.78652

with $\Delta\tau = 1/360, h = 1$, respectively. Throughout these results, we can see that the results by the new hybrid extrapolation at boundary are more robust than the conventional linear extrapolation.

3.1. Three-dimensional call option on max

In this section, we compare numerical results of the three-dimensional vanilla call option problem with two different boundary conditions. Note that the approach of new hybrid boundary is mentioned in Sec. 2.3. For numerical test, we use the following parameters $\sigma_1 = \sigma_2 = \sigma_3 = 0.3$, $\rho_{12} = \rho_{23} = \rho_{31} = \rho$, $r = 0.03$, $T = 1/6$, $X_1 = X_2 = X_3 = 100$, $L = 200$ on $\Omega = [0, L]^3$. Table 3 shows option values at the position $(x, y, z) = (100, 100, 100)$ with different values of h, $\Delta\tau$, and ρ. Error, which is defined as the absolute value of difference between numerical and exact solutions, is represented in parentheses.

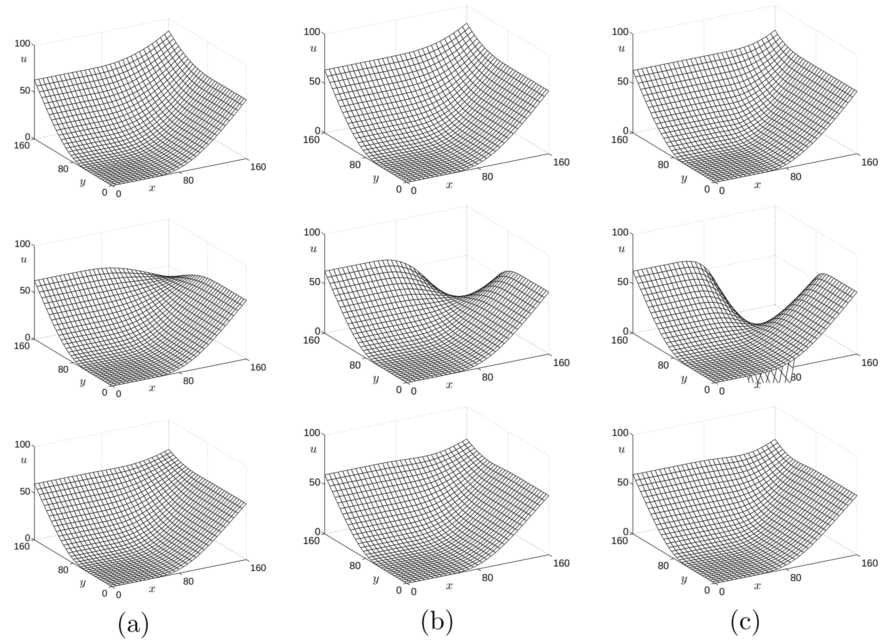

Figure 5. Numerical results at time $T = 1$ with (a) $\rho = 0.2$, (b) $\rho = 0.5$, and (c) $\rho = 0.8$. The first, second, and third rows are from closed-form solution, conventional linear boundary, and new hybrid boundary conditions with $\Delta\tau = 1/360, h = 1$, respectively.

Table 3. Three-dimensional call option values and errors at the point $(x, y, z) = (100, 100, 100)$ with various values of h, $\Delta\tau$, and ρ. Here, error is represented in parentheses.

		Conventional linear BC	New hybrid BC	Exact solution
		$(\rho = 0.2)$		
$\Delta\tau$	h			
1/180	2	27.847160 (0.7649)	27.861416 (0.7506)	
1/360	1	28.184661 (0.4273)	28.188749 (0.4233)	28.612055
		$(\rho = 0.5)$		
$\Delta\tau$	h			
1/180	2	24.326417 (0.7856)	24.230621 (0.8814)	
1/360	1	24.903289 (0.2088)	24.599627 (0.5124)	25.112060
		$(\rho = 0.8)$		
$\Delta\tau$	h			
1/180	2	27.432632 (6.9115)	19.353496 (1.1677)	
1/360	1	38.654425 (18.1332)	19.849670 (0.6715)	20.521182

As shown in Table 3, the agreement between the numerical results with hybrid boundary and the analytic exact solutions appears good. However, the results with conventional linear boundary are not good as ρ is large. Therefore, we can confirm that the results with new hybrid boundary condition are more accurate than those with conventional linear boundary condition.

3.2. Powered call option

Finally, we consider a powered option whose price follows the following single asset BS PDE: For $(x, \tau) \in \mathbb{R}_+ \times (0, T)$,

$$\frac{\partial u(x,\tau)}{\partial \tau} = rx\frac{\partial u(x,\tau)}{\partial x} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 u(x,\tau)}{\partial x^2} - ru(x,\tau). \tag{17}$$

The payoff function $\Lambda(x)$ at maturity T is $\Lambda(x) = \max\{x - K, 0\}^p$, where $p \in \mathbb{R}_+$ is a power [5]. The closed-form solution [3, 6, 14] of the powered option is given by

$$u(x,\tau) = \sum_{j=0}^{p} \frac{p!}{j!(p-j)!} x^{p-j}(-K)^j e^{(p-j-1)(r+(p-j)\sigma^2/2)\tau} N(d_{p,j}), \tag{18}$$

where $N(d_{p,j}) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{d_{p,j}} e^{-0.5s^2}\, \mathrm{d}s$ is the cumulative distribution function for the normal distribution, $d_{p,j} = [\ln(x/K) + (r + (p - j - 0.5)\sigma^2)\tau]/(\sigma\sqrt{\tau})$. In this example, we choose $p = 2$ and then for the boundary condition, we take $u_{Nx} = u_{Nx-3} - 3u_{Nx-2} + 3u_{Nx-1}$, which comes from the quadratic polynomial approximation at the boundary. This is the payoff-consistent extrapolation, i.e., $\Lambda(x_{N_x}) = u_{Nx-3} - 3u_{Nx-2} + 3u_{Nx-1}$.

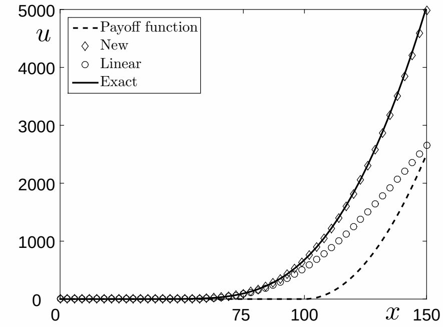

FIGURE 6. Numerical results with conventional linear and new hybrid boundary condition for a powered call option.

Figure 6 shows numerical results with conventional linear and hybrid boundary conditions with $h = 1, \Delta\tau = 1/3600, \sigma = 0.3, r = 0.03, L = 150$, and $K = 100$. In this test, we obtain two numerical values from conventional linear and new hybrid boundary conditions at $x = 100$ as $u_{\text{linear}}(100) = 534.590510$ and $u_{\text{hybrid}}(100) = 675.032810$, respectively. For comparison, we compute the analytic solution $u_{\text{exact}}(100) = 676.758118$ from Eq. (18). The relative percent error of u_{linear} is $|u_{\text{linear}}(100) - u_{\text{exact}}(100)|/u_{\text{exact}}(100) \times 100\% = 21.007\%$. And the relative percent error of u_{hybrid} is $|u_{\text{hybrid}}(100) - u_{\text{exact}}(100)|/u_{\text{exact}}(100) \times 100\% = 0.255\%$. By these results, we can confirm that the numerical results from new hybrid boundary condition are more accurate than those from conventional linear boundary condition. Also, these results indicate that the consideration of payoff is important to determine proper boundary conditions.

4. Conclusion

In this paper, we proposed a new concept for numerical treatments of boundary condition, i.e., a payoff-consistent extrapolation, which is an extrapolation consistent with the given payoff function. The new method is a hybrid combination of payoff-consistent extrapolation and the Dirichlet boundary conditions. The most popular conventional linear boundary condition has drawback that generates the bad results at boundary when the high correlation in multi-dimensional problem or option problem with a nonlinear payoff. However, the proposed new hybrid boundary condition is efficient to treat the these problems because this method takes into account of the given payoff function profile. To show the superiority of the hybrid boundary condition, we have several numerical tests such as two-, three-dimensional call options on max and one-dimensional powered option. The numerical results demonstrated that the proposed numerical scheme is more accurate and robust than the conventional linear boundary condition. As the future work, we suggest the applications of the proposed hybrid boundary condition using nonuniform grids.

References

[1] Z. Cen, A. Le, and A. Xu, *Finite difference scheme with a moving mesh for pricing Asian options*, Appl. Math. Comput. **219** (2013), no. 16, 8667–8675.

[2] D. J. Duffy, *Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach*, John Wiley and Sons, 2006.

[3] A. Esser, *General valuation principles for arbitrary payoffs and applications to power options under stochastic volatility*, Financ. Markets and Portfolio Manage. **17** (2003), no. 3, 351–372.

[4] A. Golbabai, L. V. Ballestra, and D. Ahmadian, *A Highly Accurate Finite Element Method to Price Discrete Double Barrier Options*, Comput. Econ. (2013), 1–21.

[5] E. G. Haug, *The Complete Guide to Option Pricing Formulas*, McGraw-Hill, New York, 1998.

[6] R. C. Heynen and H. M. Kat, *Pricing and hedging power options*, Financ. Eng. JPN. Markets **3** (1996), no. 3, 253–261.

[7] S. Ikonen and J. Toivanen, *Operator splitting methods for American option pricing*, Appl. Math. Lett. **17** (2004), no. 7, 809–814.

[8] K. J. In't Hout and S. Foulon, *ADI finite difference schemes for option pricing in the Heston model with correlation*, Int. J. Numer. Anal. Model **7** (2010), no. 2, 303–320.

[9] A. Q. M. Khaliq, D. A. Voss, and K. Kazmi, *Adaptive θ-methods for pricing American options*, J. Comput. Appl. Math. **222** (2008), no. 1, 210–227.

[10] G. Linde, J. Persson, and L. Von Sydow, *A highly accurate adaptive finite difference solver for the Black–Scholes equation*, Int. J. Comput. Math. **86** (2009), no. 12, 2104–2121.

[11] MathWorks, Inc.,*MATLAB*: *the language of technical computing*, http://www.mathworks.com/, The MathWorks, Natick, MA., 1998.

[12] C. Reisinger and G. Wittum, *On multigrid for anisotropic equations and variational inequalities Pricing multi-dimensional European and American options*, Comput. Vis. Sci. **7** (2004), no. 3-4, 189–197.

[13] A. Tagliani and M. Milev, *Laplace Transform and finite difference methods for the Black–Scholes equation*, Appl. Math. Comput. **220** (2013), 649–658.

[14] P. G. Zhang, *Exotic Options*: *a Guide to Second Generation Options*, World Scientific, Singapore, 1998.

[15] N. Zheng and J. F. Yin, *On the convergence of projected triangular decomposition methods for pricing American options with stochastic volatility*, Appl. Math. Comput. **223** (2013), 411–422.

[16] R. Zvan, K. R. Vetzal, and P. A. Forsyth, *PDE methods for pricing barrier options*, J. Econom. Dynam. Control **24** (2000), no. 11, 1563–1590.

Yongho Choi
Department of Mathematics
Korea University
Seoul 136-713, Korea
E-mail address: poohyongho@korea.ac.kr

Darae Jeong
Department of Mathematics
Korea University
Seoul 136-713, Korea
E-mail address: tinayoyo@korea.ac.kr

Junseok Kim
Department of Mathematics
Korea University
Seoul 136-713, Korea
E-mail address: cfdkim@korea.ac.kr

Young Rock Kim
Major in Mathematics Education
Hankuk University of Foreign Studies
Seoul 130-791, Korea
E-mail address: rocky777@hufs.ac.kr

Seunggyu Lee
Department of Mathematics
Korea University
Seoul 136-713, Korea
E-mail address: sky509@korea.ac.kr

Seungsuk Seo
Garam Analytics
Seoul 120-749, Korea
E-mail address: sseo@ganalytics.co.kr

Minhyun Yoo
Department of Financial Engineering
Korea University
Seoul 136-701, Korea
E-mail address: ymh1989@korea.ac.kr