Journal of Computational and Applied Mathematics 247 (2013) 162–171

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A comparison study of ADI and operator splitting methods on option pricing models

Darae Jeong, Junseok Kim *

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

ARTICLE INFO

Article history:
Received 31 August 2012
Received in revised form 29 December 2012

Keywords:
Black–Scholes equation
Finite difference method
ADI
OSM
Option pricing

ABSTRACT

In this paper we perform a comparison study of alternating direction implicit (ADI) and operator splitting (OS) methods on multi-dimensional Black–Scholes option pricing models. The ADI method is used extensively in mathematical finance for numerically solving multi-factor option pricing problems. However, numerical results from the ADI scheme show oscillatory solution behaviors with nonsmooth payoffs or discontinuous derivatives at the exercise price with large time steps. In the ADI scheme, there are source terms which include y-derivatives when we solve x-derivative involving equations. Then, due to the nonsmooth payoffs, source terms contain abrupt changes which are not in the range of implicit discrete operators and this leads to difficulty in solving the problem. On the other hand, the OS method does not contain the other variable's derivatives in the source terms. We provide computational results showing the performance of the methods for two-asset option pricing problems. The results show that the OS method is very efficient and gives better accuracy and robustness than the ADI method with large time steps.

1. Introduction

In today's financial markets, options are the most common securities that are frequently bought and sold. Under the Black–Scholes partial differential equation (BS PDE) framework, various numerical methods (see e.g., [1–5]) have been presented by using the finite difference method (FDM) to solve the option pricing problems (see e.g., [6–12]). However most option pricing problems have nonsmooth payoffs or discontinuous derivatives at the exercise price. Standard finite difference schemes used to solve the problems with nonsmooth payoffs and large time steps do not work well because of discontinuities introduced in the source terms. Moreover, these unwanted oscillations become problematic when we estimate the Greeks, the hedging parameters such as Delta, Gamma, Rho, Theta, and Vega.

Let $s_i(t),\ i = 1, 2, \ldots, d$ denote the value of the underlying i-th asset at time t and $u(\mathbf{s}, t)$ denote the price of an option. Here, $\mathbf{s} = (s_1, s_2, \ldots, s_d)$. In the Black–Scholes model [13], each underlying asset $s_i(t)$ satisfies the following stochastic differential equation:

$$ds_i(t) = \mu_i s_i(t)dt + \sigma_i s_i(t)dW_i(t), \quad i = 1, 2, \ldots, d,$$

where μ_i, σ_i, and $W_i(t)$ are the expected instantaneous rate of return, constant volatility, and standard Brownian motion on the underlying asset s_i, respectively. And the term dW contains the randomness which is certainly a feature of asset prices and is assumed to be a Wiener process. The Wiener processes are correlated by $\langle dW_i dW_j \rangle = \rho_{ij} dt$. Then the generalized BS

* Corresponding author. Tel.: +82 2 3290 3077; fax: +82 2 929 8562.
E-mail address: cfdkim@korea.ac.kr (J. Kim).
URL: http://math.korea.ac.kr/~cfdkim (J. Kim).

0377-0427/$ – see front matter
doi:10.1016/j.cam.2013.01.008

PDE can be derived by using Ito's lemma and the no-arbitrage principle:

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

where $r > 0$ is a constant riskless interest rate and $\Lambda(\mathbf{s})$ is the payoff function.

This paper is organized as follows. In Section 2, we introduce the Black–Scholes model in two-dimensional space and describe the ADI and OS numerical methods for the BS PDE. In Section 3, we present several numerical results showing the performance of the standard ADI and OS methods. Then we summarize our results in Section 4.

2. ADI and OS methods for the BS equation

In this paper, we focus on the two-dimensional Black–Scholes equation. Let $\mathcal{L}_{\mathrm{BS}}$ be the operator

$$\mathcal{L}_{\mathrm{BS}} = \frac{1}{2}\sigma_1^2 x^2 \frac{\partial^2 u}{\partial x^2} + \frac{1}{2}\sigma_2^2 y^2 \frac{\partial^2 u}{\partial y^2} + \rho\sigma_1\sigma_2 xy \frac{\partial^2 u}{\partial x \partial y} + rx\frac{\partial u}{\partial x} + ry\frac{\partial u}{\partial y} - ru.$$

Then the Black–Scholes equation can be written as

$$\frac{\partial u}{\partial \tau} = \mathcal{L}_{\mathrm{BS}} \quad \text{for } (x,y,\tau) \in \Omega \times (0,T], \tag{1}$$

where $\tau = T - t$. Originally, the option pricing problems are defined in the unbounded domain $\Omega \times (0,T] = \{(x,y,t) \mid x > 0,\ y > 0,\ \tau \in (0,T]\}$. However, we need to truncate this unbounded domain into a finite computational domain in order to solve Eq. (1) numerically by a finite difference method. Therefore, we consider Eq. (1) on a finite domain: $(0,L) \times (0,M) \times (0,T]$, where L and M are large enough so that the error in the price u is negligible. Let us first discretize the given 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. Furthermore, let us denote the numerical approximation of the solution by $u_{ij}^n \equiv u(x_i, y_j, \tau^n) = u(ih, jh, n\Delta\tau)$, where $i = 0, \ldots, N_x, j = 0, \ldots, N_y$, and $n = 0, \ldots, N_\tau$. We use the vertex-centered discretization since we will use a linear boundary condition [7,14–16]: $\frac{\partial^2 u}{\partial x^2}(0,y,\tau) = \frac{\partial^2 u}{\partial x^2}(L,y,\tau) = \frac{\partial^2 u}{\partial y^2}(x,0,\tau) = \frac{\partial^2 u}{\partial y^2}(x,M,\tau) = 0$, for $0 \le x \le L,\ 0 \le y \le M,\ 0 \le \tau \le T$.

2.1. Alternating directions implicit method

The main idea of the ADI method (see e.g., [17,18]) is to proceed in two stages, treating only one operator implicitly at each stage. First, a half-step is taken implicitly in x and explicitly in y. Then, the other half-step is taken implicitly in y and explicitly in x. The full scheme is

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

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

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

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

Note that the addition of Eqs. (2) and (3) results in Eq. (4):

$$\frac{u_{ij}^{n+1}-u_{ij}^{n}}{\Delta\tau}=\mathcal{L}_{\mathrm{ADI}}^{x}u_{ij}^{n+\frac{1}{2}}+\mathcal{L}_{\mathrm{ADI}}^{y}u_{ij}^{n+1}. \tag{4}$$

Algorithm ADI.

- *Step* 1: The first stage of the ADI method, Eq. (2), can be rewritten as

$$\alpha_i u_{i-1,j}^{n+\frac{1}{2}}+\beta_i u_{ij}^{n+\frac{1}{2}}+\gamma_i u_{i+1,j}^{n+\frac{1}{2}}=f_{ij}, \tag{5}$$

where

$$\alpha_i=-\frac{(\sigma_1 x_i)^2}{4h^2},\qquad \beta_i=\frac{1}{\Delta\tau}+\frac{(\sigma_1 x_i)^2}{2h^2}+\frac{rx_i}{2h}+\frac{r}{2},\qquad \gamma_i=-\frac{(\sigma_1 x_i)^2}{4h^2}-\frac{rx_i}{2h}, \tag{6}$$

$$\begin{aligned} f_{ij} &= \frac{u_{ij}^{n}}{\Delta\tau}+\frac{1}{4}(\sigma_2 y_j)^2\frac{u_{i,j+1}^{n}-2u_{ij}^{n}+u_{i,j-1}^{n}}{h^2}+\frac{1}{2}ry_j\frac{u_{i,j+1}^{n}-u_{i,j}^{n}}{h} \\ &\quad+\frac{1}{2}\rho\sigma_1\sigma_2 x_i y_j\frac{u_{i+1,j+1}^{n}+u_{i-1,j-1}^{n}-u_{i-1,j+1}^{n}-u_{i+1,j-1}^{n}}{4h^2}. \end{aligned} \tag{7}$$

For a fixed index j, the vector $u_{0:N_x,j}^{n+\frac{1}{2}}$ can be found by solving the tridiagonal system

$$A_x u_{0:N_x,j}^{n+\frac{1}{2}}=f_{0:N_x,j},$$

where A_x is a tridiagonal matrix constructed from Eq. (5) with a linear boundary condition, i.e.,

$$A_x=\begin{pmatrix} 2\alpha_0+\beta_0 & \gamma_0-\alpha_0 & 0 & \cdots & 0 & 0 \\ \alpha_1 & \beta_1 & \gamma_1 & \cdots & 0 & 0 \\ 0 & \alpha_2 & \beta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_{N_x-1} & \gamma_{N_x-1} \\ 0 & 0 & 0 & \cdots & \alpha_{N_x}-\gamma_{N_x} & \beta_{N_x}+2\gamma_{N_x} \end{pmatrix}.$$

Step 1 of the ADI method is then implemented in a loop over the y-direction:

for $j=0:N_y$
  for $i=0:N_x$
    Set α_i, β_i, γ_i, and f_{ij} by Eqs. (6) and (7)
  end
  Solve $A_x u_{0:N_x,j}^{n+\frac{1}{2}}=f_{0:N_x,j}$ by using the Thomas algorithm
end

- *Step* 2: The second stage of the ADI method, given by Eq. (3), is rewritten as

$$\alpha_j u_{i,j-1}^{n+1}+\beta_j u_{ij}^{n+1}+\gamma_j u_{i,j+1}^{n+1}=g_{ij}, \tag{8}$$

where

$$\alpha_j=-\frac{(\sigma_2 y_j)^2}{4h^2},\qquad \beta_j=\frac{1}{\Delta\tau}+\frac{(\sigma_2 y_j)^2}{2h^2}+\frac{ry_j}{2h}+\frac{r}{2},\qquad \gamma_j=-\frac{(\sigma_2 y_j)^2}{4h^2}-\frac{ry_j}{2h}, \tag{9}$$

$$\begin{aligned} g_{ij} &= \frac{u_{ij}^{n+\frac{1}{2}}}{\Delta\tau}+\frac{(\sigma_1 x_i)^2}{4}\frac{u_{i+1,j}^{n+\frac{1}{2}}-2u_{ij}^{n+\frac{1}{2}}+u_{i-1,j}^{n+\frac{1}{2}}}{h^2}+\frac{1}{2}rx_i\frac{u_{i+1,j}^{n+\frac{1}{2}}-u_{i,j}^{n+\frac{1}{2}}}{h} \\ &\quad+\frac{1}{2}\rho\sigma_1\sigma_2 x_i y_j\frac{u_{i+1,j+1}^{n+\frac{1}{2}}+u_{i-1,j-1}^{n+\frac{1}{2}}-u_{i-1,j+1}^{n+\frac{1}{2}}-u_{i+1,j-1}^{n+\frac{1}{2}}}{4h^2}. \end{aligned} \tag{10}$$

For a fixed index i, the vector $u_{i,0:N_y}^{n+1}$ can be found by solving the tridiagonal system

$$A_y u_{i,0:N_y}^{n+1}=g_{i,0:N_y},$$

where the matrix A_y is a tridiagonal matrix from Eq. (8):

$$A_y = \begin{pmatrix} 2\alpha_0+\beta_0 & -\alpha_0+\gamma_0 & 0 & \cdots & 0 & 0 \\ \alpha_1 & \beta_1 & \gamma_1 & \cdots & 0 & 0 \\ 0 & \alpha_2 & \beta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_{N_y-1} & \gamma_{N_y-1} \\ 0 & 0 & 0 & \cdots & \alpha_{N_y}-\gamma_{N_y} & \beta_{N_y}+2\gamma_{N_y} \end{pmatrix}.$$

Step 2 is then implemented in a loop over the x-direction:

for $i = 0 : N_x$
  for $j = 0 : N_y$
    Set α_j, β_j, γ_j, and g_{ij} by Eqs. (9) and (10)
  end
  Solve $A_y u^{n+1}_{i,0:N_y} = g_{i,0:N_y}$ by using the Thomas algorithm
end

Execution of *Step* 1 followed by *Step* 2 advances the solution with a $\Delta\tau$-step in time.

2.2. Operator splitting method

The idea of the OS method is to divide each time step into fractional time steps with simpler operators (see e.g., [11,19]). We shall introduce the basic OS scheme for the two-dimensional BS equation. The first leg is implicit in x while the second leg is implicit in y. The full scheme is

$$\frac{u_{ij}^{n+\frac{1}{2}} - u_{ij}^n}{\Delta\tau} = \mathcal{L}_{OS}^x u_{ij}^{n+\frac{1}{2}}, \tag{11}$$

$$\frac{u_{ij}^{n+1} - u_{ij}^{n+\frac{1}{2}}}{\Delta\tau} = \mathcal{L}_{OS}^y u_{ij}^{n+1}, \tag{12}$$

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

$$\begin{aligned} \mathcal{L}_{OS}^x u_{ij}^{n+\frac{1}{2}} &= \frac{(\sigma_1 x_i)^2}{2}\frac{u_{i-1,j}^{n+\frac{1}{2}} - 2u_{ij}^{n+\frac{1}{2}} + u_{i+1,j}^{n+\frac{1}{2}}}{h^2} + rx_i\frac{u_{i+1,j}^{n+\frac{1}{2}} - u_{ij}^{n+\frac{1}{2}}}{h} - \frac{r}{2}u_{ij}^{n+\frac{1}{2}} \\ &\quad + \frac{1}{2}\sigma_1\sigma_2\rho x_i y_j\frac{u_{i+1,j+1}^n + u_{i-1,j-1}^n - u_{i-1,j+1}^n - u_{i+1,j-1}^n}{4h^2}, \end{aligned} \tag{13}$$

$$\begin{aligned} \mathcal{L}_{OS}^y u_{ij}^{n+1} &= \frac{(\sigma_2 y_j)^2}{2}\frac{u_{i,j-1}^{n+1} - 2u_{ij}^{n+1} + u_{i,j+1}^{n+1}}{h^2} + ry_j\frac{u_{i,j+1}^{n+1} - u_{ij}^{n+1}}{h} - \frac{r}{2}u_{ij}^{n+1} \\ &\quad + \frac{1}{2}\sigma_1\sigma_2\rho x_i y_j\frac{u_{i+1,j+1}^{n+\frac{1}{2}} + u_{i-1,j-1}^{n+\frac{1}{2}} - u_{i-1,j+1}^{n+\frac{1}{2}} - u_{i+1,j-1}^{n+\frac{1}{2}}}{4h^2}. \end{aligned} \tag{14}$$

The OS scheme moves from the time level n to an intermediate time level $n+\frac{1}{2}$ and then to the time level $n+1$. The addition of Eqs. (11) and (12) results in Eq. (15):

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta\tau} = \mathcal{L}_{OS}^x u_{ij}^{n+\frac{1}{2}} + \mathcal{L}_{OS}^y u_{ij}^{n+1}. \tag{15}$$

Algorithm OS.

- *Step* 1: Eq. (11) is rewritten as follows:

$$\alpha_i u_{i-1j}^{n+\frac{1}{2}} + \beta_i u_{ij}^{n+\frac{1}{2}} + \gamma_i u_{i+1j}^{n+\frac{1}{2}} = f_{ij},$$

where

$$\alpha_i = -\frac{\sigma_1^2 x_i^2}{2h^2}, \qquad \beta_i = \frac{1}{\Delta\tau} + \frac{\sigma_1^2 x_i^2}{h^2} + \frac{rx_i}{h} + \frac{r}{2}, \qquad \gamma_i = -\frac{\sigma_1^2 x_i^2}{2h^2} - \frac{rx_i}{h},$$

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

We note that in the OS method we do not have $\partial^2 u/\partial y^2$ and $\partial u/\partial y$ terms in the source f_{ij}. Then the solution procedure is the same as the ADI method.

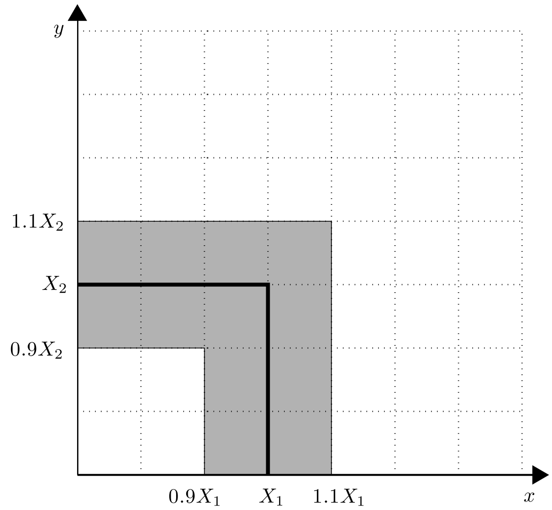

Fig. 1. RMSE is calculated on the gray region.

- *Step* 2: Eq. (12) is rewritten as follows:

$$\alpha_j u_{ij-1}^{n+1} + \beta_j u_{ij}^{n+1} + \gamma_j u_{ij+1}^{n+1} = g_{ij},$$

where

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

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

We also note that we do not have $\partial^2 u/\partial x^2$ and $\partial u/\partial x$ terms in the source g_{ij} and the solution procedure is the same as the ADI method.

3. Numerical experiments

In this section, various numerical examples are presented to compare the performance of the two different numerical schemes, the ADI and OS methods, for the BS equation. All computations were performed using MATLAB version 7.6 [20]. The error of the numerical solution was defined as $e_{ij} = u_{ij}^e - u_{ij}$, where u_{ij}^e is the exact solution and u_{ij} is the numerical solution. To compare the errors of the ADI and OS methods, we computed discrete l^2 norm $\|\mathbf{e}\|_2$ and maximum norm $\|\mathbf{e}\|_\infty$ of the error. We also used the root mean square error (RMSE) on a specific region. The RMSE is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i,j}^{N}\left(u_{ij}^e - u_{ij}\right)^2},$$

where N is the number of points on the gray region as shown in Fig. 1 and the region indicates a neighborhood of the exercise prices X_1 and X_2.

3.1. Two-asset cash or nothing option

First, let us consider the two-asset cash or nothing call option [21]. Given two assets x and y, the payoff of the call option is given as

$$\Lambda(x,y) = \begin{cases} \text{Cash} & \text{if } x \ge X_1 \text{ and } y \ge X_2, \\ 0 & \text{otherwise,} \end{cases}$$

where X_1 and X_2 are the strike prices of x and y, respectively (see Fig. 2).

The exact solution is obtained from a closed-form solution, which is provided in Appendix A.1.

The parameters used are $\sigma_1 = \sigma_2 = 0.3$, $r = 0.03$, $\rho = 0.5$, $T = 0.5$, Cash $= 1$, and $X_1 = X_2 = 100$. The computational domain is $\Omega = [0, 300] \times [0, 300]$.

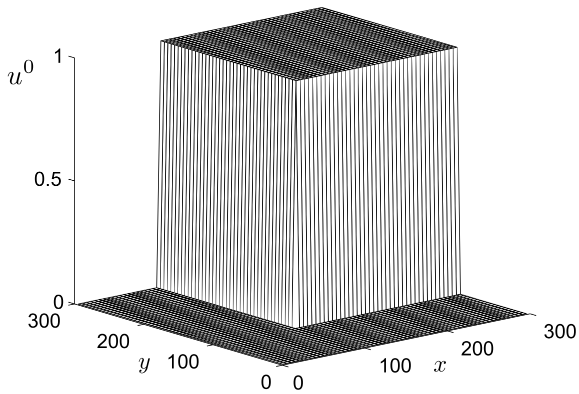

Fig. 2. Payoff of a two-asset cash or nothing call option.

Table 1
Numerical results of a two-asset cash or nothing call option with different time step $\Delta\tau$ and space step h. Here, $\|\mathbf{e}\|_2$ and $\|\mathbf{e}\|_\infty$ are measured in a quarter of the domain, $[0, 150] \times [0, 150]$.

Time $\Delta\tau$	Space h	ADI			OSM		
		$\|\mathbf{e}\|_2$	$\|\mathbf{e}\|_\infty$	RMSE	$\|\mathbf{e}\|_2$	$\|\mathbf{e}\|_\infty$	RMSE
0.05	5.0	0.000506	0.001952	0.000023	0.002411	0.010449	0.000147
0.025	2.5	0.000346	0.001266	0.000006	0.001043	0.004569	0.000028
0.0125	1.25	0.000207	0.000944	0.000003	0.000483	0.002136	0.000006
0.00625	0.625	N/A	N/A	N/A	0.000232	0.001030	0.000001

Table 2
Numerical results in the case of European option on the maximum of two-asset with respect to the time step $\Delta\tau$ and space step h. Here, $\|\mathbf{e}\|_2$ and $\|\mathbf{e}\|_\infty$ are measured in a quarter of the domain, $[0, 150] \times [0, 150]$ and the RMSEs are evaluated in the gray region which is represented in Fig. 1.

Time $\Delta\tau$	Space h	ADI			OSM		
		$\|\mathbf{e}\|_2$	$\|\mathbf{e}\|_\infty$	RMSE	$\|\mathbf{e}\|_2$	$\|\mathbf{e}\|_\infty$	RMSE
0.05	5.0	0.057677	0.120848	0.006319	0.059967	0.175874	0.002286
0.025	2.5	0.027863	0.059866	0.001581	0.029001	0.085344	0.000610
0.0125	1.25	0.013713	0.029731	0.000395	0.014248	0.041703	0.000158
0.00625	0.625	N/A	N/A	N/A	0.007060	0.020569	0.000040

As shown in Table 1, the ADI shows better convergence results than the OS method with relatively large space step sizes. However, with smaller space step size (equivalently with large temporal step size), the ADI shows blowup solutions while the OS method produces convergent results.

To investigate what made blowup solutions for the ADI scheme, we compare solutions, $u^{\frac{1}{2}}$ and u^1, and source terms, f and g, generated from the ADI and OS methods. We used time step size, $\Delta\tau = 0.5$, and space step size, $h = 5$.

In Fig. 3, the first and the second columns show the numerical results at each step of the ADI and OSM for a two-asset cash or nothing option, respectively. As we can see from the figure, the numerical result of the ADI with a relatively large time step shows oscillatory solution along the lines $x = X_1$ and $y = X_2$. In Fig. 3(a), source terms in the first steps are shown. The source term in the ADI method exhibits oscillation around $y = X_2$ which is from the y-derivatives in the source term. On the other hand, for the OS method, we do not have the y-derivatives in the source term and solution $u^{\frac{1}{2}}$ is monotone around $y = X_2$. Therefore, for the ADI we have an oscillatory solution at the first step. After one complete time step, the result with the ADI shows a nonsmooth numerical solution. However, the OS method results in a smooth numerical solution.

3.2. Call option on the maximum of two assets

Next, we consider a vanilla call option whose payoff is given as

$$\Lambda(x, y) = \max\{x - X, y - X, 0\}. \tag{16}$$

Fig. 4 shows the payoff function (16).

In this case, we use the Dirichlet boundary condition at $x = L$ and $y = M$ and the linear boundary condition at $x = 0$ and $y = 0$. The parameters used are $\sigma_1 = \sigma_2 = 0.3$, $r = 0.03$, $\rho = 0.5$, $T = 0.5$, and $X = 100$. The computational domain is $\Omega = [0, 300] \times [0, 300]$.

Table 2 shows the comparison of errors for the ADI and OS methods at time T. The exact solutions are obtained from a closed-form solution, which is provided in Appendix A.2. The errors are similar in magnitude for the two methods until

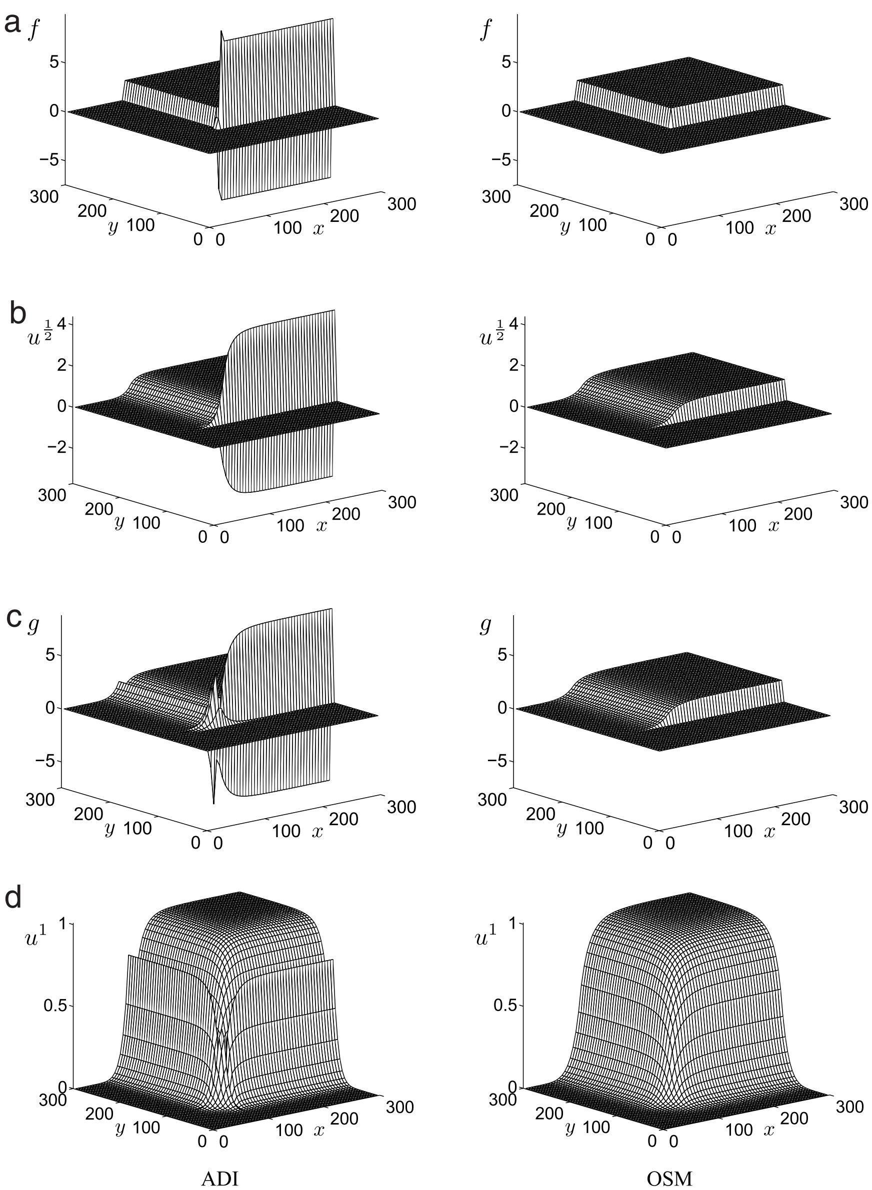

Fig. 3. Numerical results of cash or nothing option using the ADI and OSM. (a) Source term f at *Step* 1, (b) solution $u^{\frac{1}{2}}$ at *Step* 1, (c) source term g at *Step* 2, and (d) solution u^1 at *Step* 2.

space step $h = 1.25$. However, after that, the results from the ADI with smaller space steps show the blowup phenomenon of solutions. On the other hand, the errors with the OS method do decrease with respect to time and space step refinements.

Fig. 5 shows numerical results using the ADI and OS methods with $\Delta\tau = 0.5$ and $h = 3$. The first and second columns are results with the ADI and OS methods, respectively. In Fig. 5(a), source terms in the first steps are shown. The source term in the ADI method exhibits oscillation around $y = X$ which is from the y-derivatives in the source term. On the other hand, for the OS method, we do not have the y-derivatives in the source term and solution $u^{\frac{1}{2}}$ is smooth around $y = X$. After one complete time step, the result from the ADI shows a nonsmooth numerical solution. However, the OS method results in a smooth numerical solution.

4. Conclusion

In this paper we performed a comparison study of alternating direction implicit and operator splitting methods on two-dimensional Black–Scholes option pricing models. In the ADI scheme, there are source terms which include y-derivatives when we solve x-derivative involving equations. The source terms contain abrupt changes due to the nonsmooth payoffs

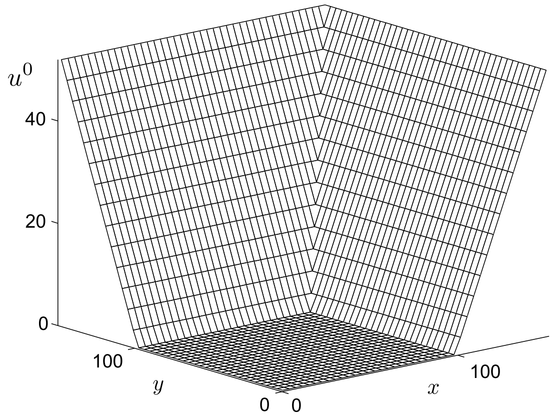

Fig. 4. European call option payoff on the maximum of two assets.

and are not in the range of implicit discrete operators, which leads to difficulty in solving the problem. On the other hand, the OS method does not contain the other variable's derivatives in the source terms. We provided computational results showing the performance of the methods for two-asset option pricing problems. The results showed that the OS method is very efficient and robust than the ADI method with large time steps.

Acknowledgments

This work was supported by Seed Program of the Korean Research Council of Fundamental Science and Technology (KRCF). The authors thank the reviewers for the constructive and useful comments on this paper.

Appendix. MATLAB code for closed-form solutions

A.1. Cash or nothing option

```
sigma1=0.3; sigma2=0.3; r=0.03; rho=0.5; T=0.5; Cash=1;
X1=100; X2=100; L=300; M=300; Nx=60; Ny=60;
x=linspace(0,L,Nx+1); y=linspace(0,M,Ny+1);
for i=1:Nx+1
 for j=1:Ny+1
 y1=(log(x(i)/X1)+(r-sigma{1}^{2}/2)*T)/(sigma1*sqrt(T));
 y2=(log(y(j)/X2)+(r-sigma{2}^{2}/2)*T)/(sigma2*sqrt(T));
 V(i,j)=Cash*exp(-r*T)*mvncdf([y1 y2],[0 0],[1 rho; rho 1]);
end
end; surf(x,y, V)
```

A.2. Max max option

```
sigma1=0.3; sigma2=0.3; r=0.03; rho=0.5; T=0.5; X=100; L=300;
M=300; Nx=60; Ny=60;x=linspace(0,L,Nx+1); y=linspace(0,M,Ny+1);
sig=sqrt(sigma{1}^{2}+sigma{2}^{2}-2*rho*sigma1*sigma2);
rho1=(sigma1-rho*sigma2)/sig; rho2=(sigma2-rho*sigma1)/sig;
for i=1:Nx+1
 for j=1:Ny+1
 d=(log(x(i)/y(j))+0.5*sig^2*T)/(sig*sqrt(T));
 y1=(log(x(i)/X)+0.5*sigma{1}^{2}*T)/(sig*sqrt(T));
 y2=(log(y(j)/X)+0.5*sigma{2}^{2}*T)/(sig*sqrt(T));
 V(i,j)=x(i)*mvncdf([y1 d],[0 0],[1 rho1; rho1 1]) ...
 +y(j)*mvncdf([y2 -d+sig*sqrt(T)],[0 0],[1 rho2; rho2 1]) ...
  -X*exp(-r*T)*(1-mvncdf([-y1+sigma1*sqrt(T) ...
  -y2+sigma2*sqrt(T)],[0 0],[1 rho; rho 1]));
end
end; surf(x,y, V)
```

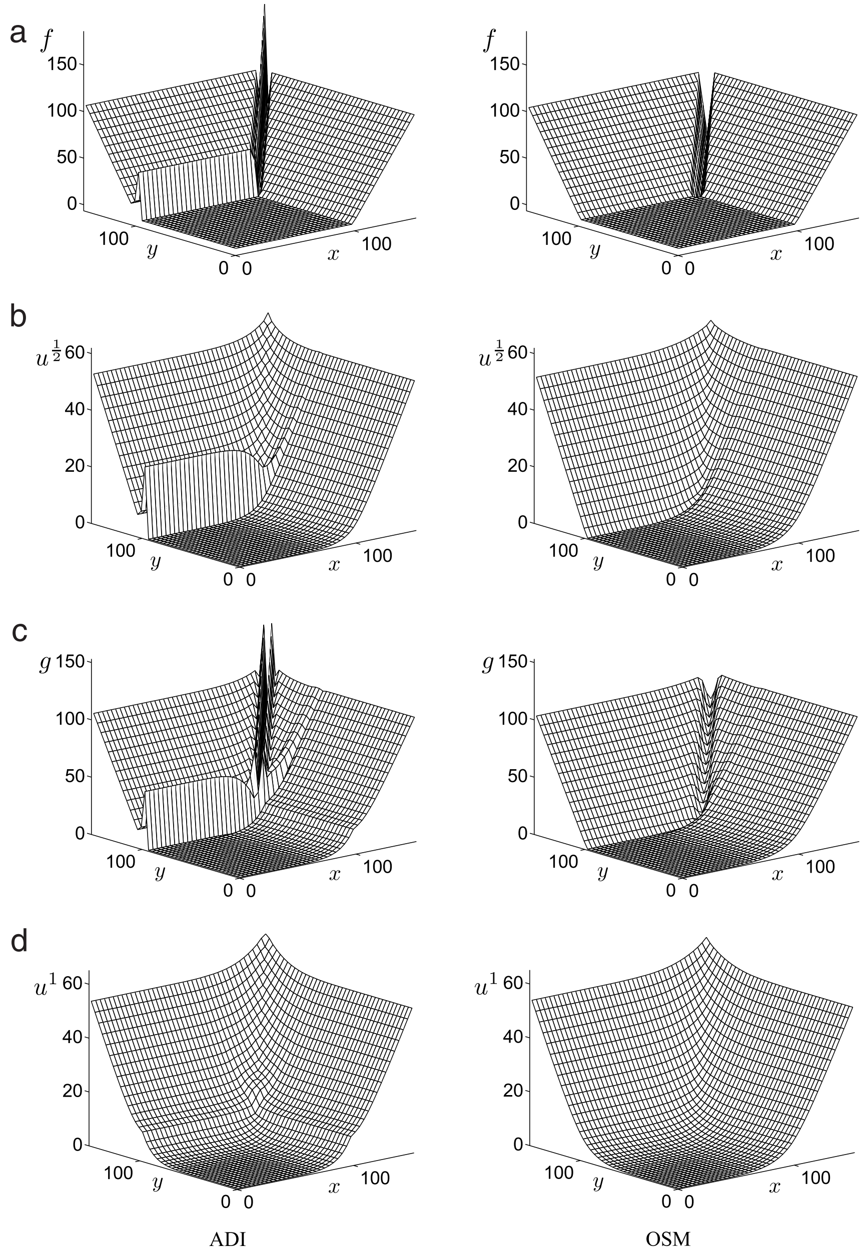

Fig. 5. Numerical results using the ADI and OS methods with European call option on the maximum of two assets. (a) Source term f at *Step* 1, (b) solution $u^{\frac{1}{2}}$ at *Step* 1, (c) source term g at *Step* 2, and (d) solution u^1 at *Step* 2.

References

[1] G.N. Milstein, M.V. Tretyakov, Numerical analysis of Monte Carlo evaluation of Greeks by finite differences, J. Comput. Finance 8 (3) (2005) 1–34.
[2] A.Q.M. Khaliq, D.A. Voss, K. Kazmi, Adaptive θ-methods for pricing American options, J. Comput. Appl. Math. 222 (2008) 210–227.
[3] A. Eckner, Computational techniques for basic affine models of portfolio credit risk, J. Comput. Finance 13 (1) (2009) 63–97.
[4] J. Toivanen, A high-order front-tracking finite diffrence method for pricing American options under jump–diffusion models, J. Comput. Finance 13 (3) (2010) 61–79.
[5] Z. Cen, A. Le, A robust and accurate finite difference method for a generalized Black–Scholes equation, J. Comput. Appl. Math. 235 (2011) 3728–3733.
[6] P. Wilmott, J. Dewynne, S. Howisson, Option Pricing: Mathematical Models and Computation, Oxford Financial Press, Oxford, 1993.
[7] D. Tavella, C. Randall, Pricing Financial Instruments: The Finite Difference Method, John Wiley and Sons, Chichester, 2000.
[8] R. Seydel, Tools for Computational Finance, Springer Verlag, Berlin, 2003.
[9] Y. Achdou, O. Pironneau, Computational Methods for Option Pricing, SIAM, Philadelphia, 2005.
[10] J. Topper, Financial Engineering with Finite Elements, John Wiley and Sons, Hoboken, 2005.
[11] D.J. Duffy, Finite Difference Methods in Financial Engineering: a Partial Differential Equation Approach, John Wiley and Sons, Sussex, 2006.

[12] H.-J. Bungartz, A. Heinecke, D. Pflüger, S. Schraufstetter, Option pricing with a direct adaptive sparse grid approach, J. Comput. Appl. Math. 236 (2012) 3741–3750.
[13] F. Black, M. Scholes, The pricing of options and corporate liabilities, J. Polit. Econ. 81 (1973) 637–659.
[14] R. Zvan, K.R. Zvan, P.A. Forsyth, PDE methods for pricing barrier options, J. Econom. Dynam. Control 24 (2000) 1563–1590.
[15] C.W. Oosterlee, On multigrid for linear complementarity problems with application to American-style options, Electron. Trans. Numer. Anal. 15 (2003) 165–185.
[16] J. Persson, L. von Sydow, Pricing European multi-asset options using a space–time adaptive FD-method, Comput. Vis. Sci. 10 (2007) 173–183.
[17] R.C.Y. Chin, T.A. Manteuffel, J. de Pillis, ADI as a preconditioning for solving the convection–diffusion equation, SIAM J. Sci. Comput. 5 (1984) 281–299.
[18] K.J. Hout, S. Foulon, ADI finite difference schemes for option pricing in the Heston model with correlation, Int. J. Numer. Anal. Model. 7 (2) (2010) 303–320.
[19] S. Ikonen, J. Toivanen, Operator splitting methods for American option pricing, Appl. Math. Lett. 17 (2004) 809–814.
[20] Using MATLAB, The MathWorks Inc., Natick, MA, 1998, http://www.mathworks.com/.
[21] E.G. Haug, The Complete Guide to Option Pricing Formulas, McGraw-Hill, New York, 1997.