Bull. Korean Math. Soc. **50** (2013), No. 6, pp. 2035–2051
http://dx.doi.org/10.4134/BKMS.2013.50.6.2035

COMPARISON OF NUMERICAL SCHEMES ON MULTI-DIMENSIONAL BLACK-SCHOLES EQUATIONS

Joonglee Jo and Yongsik Kim

Abstract. In this paper, we study numerical schemes for solving multidimensional option pricing problem. We compare the direct solving method and the *Operator Splitting Method*(OSM) by using finite difference approximations. By varying parameters of the *Black-Scholes equations* for the maximum on the call option problem, we observed that there is no significant difference between the two methods on the convergence criterion except a huge difference in computation cost. Therefore, the two methods are compatible in practice and one can improve the time efficiency by combining the OSM with parallel computation technique. We show numerical examples including the *Equity-Linked Security*(ELS) pricing based on either two assets or three assets by using the OSM with the *Monte-Carlo Simulation* as the benchmark.

1. Introduction

Ever since *Fischer Black* and *Myron Scholes* proposed their renowned *partial differential equations* on option pricing problems in 1973, there have been numerous attempts to find the analytic form solutions of the Black-Scholes equation for various derivatives in the financial market. However, it is very hard to get an analytic form solution for most of financial derivatives due to the complexity of the financial product itself and the system of the financial market. Hence people in the financial industry have used numerical methods such as *Monte Carlo Simulation*(MC) or *Finite Difference Method*(FDM) for solving option pricing problems. The fast and accurate numerical computation for option pricing is a crucial matter in the financial industry. In view of this, it is of significance to study quantitative analysis such as parameter studies and

Received December 19, 2012; Revised March 28, 2013.

2010 *Mathematics Subject Classification.* Primary 65M06, 91G20, 91G60.

Key words and phrases. Black-Scholes equation, operator splitting method, equity-linked security, parallel computation, message passing interface.

The work of Kim was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0011855). The work of Jo was supported by WCU-project(MEST)(R31-20007).

performance comparison on numerical schemes for solving the Black-Scholes equations.

The objectives of our study in this paper are the following. First, we compare two different numerical schemes for the multi-dimensional Black-Scholes equations by performance comparison; the direct solving method and the *Operator Splitting Method.* The direct solving method in this paper means a general FDM scheme for multi-dimensional option pricing, which gives a large size of linear system depending on dimensionality. In the direct solving method, the computational cost is exponentially increasing as the problem's dimension is getting larger. An alternative to overcome this drawback is the *Operator Splitting Method*(OSM). The OSM is more efficient than the direct solving method in the sense of computational cost. The OSM transmutes the original problem which involves solving one large linear system into solving several small linear systems. Further, the small linear systems can be solved independently with each other. Hence it is suitable to combine the parallel computation technique with the OSM. For the heat equation, it is well known that the OSM and the direct solving method are equivalent on the accuracy in the sense that they have the same convergence order. However, for the Black-Scholes equations, there is no known mathematical error analysis as in the heat equation in which the OSM and the direct solving method are compatible. Thus we try numerical comparison for those schemes quantitatively. Second, we examine the efficiency of the OSM for the Black-Scholes equations combined with a parallel computation technique called *Message Passing Interface*(MPI) implementation. The MPI implementation makes it possible to allocate a serial computation work to several cores (processors) so that the computation efficiency gets improved.

Our main conclusion is that the OSM not only gives same accuracy but also provides huge efficiency for solving the Black-Scholes equations against the direct solving method. Theoretically, the direct solving method is better than the OSM in view of absolute error. However, our numerical experiments present that both schemes show same convergence order numerically. Therefore, we can consider that the two methods are virtually compatible on the option pricing problems. We checked the cost efficiency of the OSM through numerical comparison on serial processing. When we utilize parallel processing, the efficiency of the OSM is more outstanding from the other. For the direct solving method, there are parallel computation schemes such as the domain decomposition methods. The domain decomposition methods require information exchanges on interface regions of divided computational domain. Hence as the number of divided regions is increased, the speed-up of the parallel computing is significantly decreased. On the contrary, the OSM does not require information exchanges. Thus one can easily implement a parallel computing of the OSM without significant speed-up decreasing problem.

The rest of this paper is organized as follows. In Section 2, we introduce the multi-dimensional option pricing problem and a relevant governing equation. Next we summarize FDM schemes for both the direct solving method and the

OSM. In Section 3, we conduct a performance analysis with two dimensional maximum on the call option problem by comparing the two methods. We will also show several numerical examples based on two or three dimensional problems from real financial market. Then we show a technical improvement in efficiency by parallel computation for the OSM. The paper's final section is the conclusion.

2. Numerical schemes

In this section, we state the multi-dimensional option pricing problem with governing equation and introduce two numerical schemes to solve the problem.

2.1. Governing equation

Various numerical methods have been adopted to solve option pricing problems which are difficult to obtain analytic form solutions, including the American option and various exotic options. Since all the options share the same fundamental structure, such as the portfolio replication under the no-arbitrage condition, we consider the following Black-Scholes equations as the governing equation.

$$\frac{\partial u}{\partial t}+\frac{1}{2}\sigma_1^2x^2\frac{\partial^2 u}{\partial x^2}+\frac{1}{2}\sigma_2^2y^2\frac{\partial^2 u}{\partial y^2}+\rho\sigma_1\sigma_2xy\frac{\partial^2 u}{\partial x\partial y}+rx\frac{\partial u}{\partial x}+ry\frac{\partial u}{\partial y}-ru=0. \tag{1}$$

Here, we restrict only two dimensional case for simplicity. Meanings of variables and parameters are the following: $u(t,x(t),y(t))$ is the option value with t as time, x as the first asset value, and y as the second asset value; σ_1 and σ_2 are the volatilities of the two assets, respectively; ρ is the correlation of the two assets; and r is the risk-free interest rate. Since the PDE (1) is typically augmented with a final time condition, we change the variable $\tau=T-t$ so that we could solve from the initial condition for convenience. Under this change of variable, the equation (1) becomes

$$\frac{\partial u}{\partial \tau}-\frac{1}{2}\sigma_1^2x^2\frac{\partial^2 u}{\partial x^2}-\frac{1}{2}\sigma_2^2y^2\frac{\partial^2 u}{\partial y^2}-\rho\sigma_1\sigma_2xy\frac{\partial^2 u}{\partial x\partial y}-rx\frac{\partial u}{\partial x}-ry\frac{\partial u}{\partial y}+ru=0. \tag{2}$$

Note that (1) can be extended to more general (n-dimensional) PDE such as

$$\frac{\partial u}{\partial t}+\frac{1}{2}\sum_{i,j=1}^{n}\rho_{ij}\sigma_i\sigma_jx_ix_j\frac{\partial^2 u}{\partial x_i\partial x_j}+r\sum_{i=1}^{n}x_i\frac{\partial u}{\partial x_i}-ru=0. \tag{3}$$

We will consider a three dimensional case as a numerical example in section three.

2.2. Direct solving method

The direct solving method is the general FDM scheme to solve the equation (2) numerically considering the whole dimension at once. If we use the central difference for spatial discretization and the implicit scheme for time discretization, the governing equation (2) is discretized as the following:

$$
\begin{aligned}
&\frac{u_{ij}^{n+1}-u_{ij}^{n}}{\Delta\tau}-\frac{1}{2}\sigma_1^2x_i^2\frac{u_{i-1,j}^{n+1}-2u_{ij}^{n+1}+u_{i+1,j}^{n+1}}{\Delta x^2}-\frac{1}{2}\sigma_2^2y_j^2\frac{u_{i,j-1}^{n+1}-2u_{ij}^{n+1}+u_{i,j+1}^{n+1}}{\Delta y^2}\\
&\qquad-\rho\sigma_1\sigma_2x_iy_j\frac{u_{i+1,j+1}^{n+1}+u_{i-1,j-1}^{n+1}-u_{i+1,j-1}^{n+1}-u_{i-1,j+1}^{n+1}}{4\Delta x\Delta y}\\
&\qquad-rx_i\frac{u_{i+1,j}^{n+1}-u_{i-1,j}^{n+1}}{2\Delta x}-ry_j\frac{u_{i,j+1}^{n+1}-u_{i,j-1}^{n+1}}{2\Delta y}+ru_{ij}^{n+1}=0.
\end{aligned}\tag{4}
$$

Here the subscripts i and j stand for horizontal node index and vertical node index respectively, and the superscript n stands for time step index. For simplicity, we consider uniform grid where the nodal distances for each axis are same such that $\Delta x=\Delta y=h$. When we mention that we solve the above system (4) "directly" with the additional boundary conditions, it means we solve the whole system at once by linear system solver such as CGM(*Conjugate-Gradient Method*) or PARDISO [8].

2.3. Operator splitting method

The *Operator Splitting Method*(OSM) is another numerical scheme to solve the equation (2), but in different way with the direct solving method. The main idea of the OSM is to "divide and conquer" such that we literally divide the multi-dimensional problem into several subproblems and solve in fractional time step [1, 2, 7, 10]. In this two dimensional problem, we split the differential operator $\mathcal{L}$ into $\mathcal{L}_1$ and $\mathcal{L}_2$ such that

$$\frac{\partial u}{\partial\tau}+\mathcal{L}_1u+\mathcal{L}_2u=0,\quad\text{where}$$

$$
\begin{aligned}
\mathcal{L}u&:=-\frac{1}{2}\sigma_1^2x^2u_{xx}-rxu_x-\frac{1}{2}\sigma_2^2y^2u_{yy}-ryu_y+ru-\rho\sigma_1\sigma_2xyu_{xy},\\
\mathcal{L}_1u&:=-\frac{1}{2}\sigma_1^2x^2u_{xx}-rxu_x+\lambda_2ru-\lambda_1\rho\sigma_1\sigma_2xyu_{xy},\\
\mathcal{L}_2u&:=-\frac{1}{2}\sigma_2^2y^2u_{yy}-ryu_y+(1-\lambda_2)ru-(1-\lambda_1)\rho\sigma_1\sigma_2xyu_{xy}.
\end{aligned}
$$

Then the systems (4) are sequentially split as the following.

1) x-step:

$$
\begin{aligned}
&\frac{u_{ij}^{n+\frac{1}{2}}}{\Delta\tau}-\frac{1}{2}\sigma_1^2x_i^2\frac{u_{i-1,j}^{n+\frac{1}{2}}-2u_{ij}^{n+\frac{1}{2}}+u_{i+1,j}^{n+\frac{1}{2}}}{h^2}-rx_i\frac{u_{i+1,j}^{n+\frac{1}{2}}-u_{i-1,j}^{n+\frac{1}{2}}}{2h}+\lambda_2ru_{ij}^{n+\frac{1}{2}}\\
&=\lambda_1\rho\sigma_1\sigma_2x_iy_j\frac{u_{i+1,j+1}^{n}+u_{i-1,j-1}^{n}-u_{i+1,j-1}^{n}-u_{i-1,j+1}^{n}}{4h^2}+\frac{u_{ij}^{n}}{\Delta\tau},
\end{aligned}
$$

2) y-step:

$$\frac{u_{ij}^{n+1}}{\Delta\tau}-\frac{1}{2}\sigma_2^2y_j^2\frac{u_{i,j-1}^{n+1}-2u_{ij}^{n+1}+u_{i,j+1}^{n+1}}{h^2}-ry_j\frac{u_{i,j+1}^{n+1}-u_{i,j-1}^{n+1}}{2h}+(1-\lambda_2)ru_{ij}^{n+1}$$

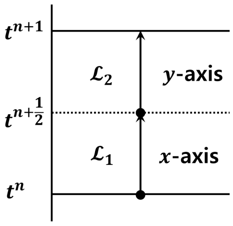

FIGURE 1. Operator splitting scheme for two dimension problem

$$= (1-\lambda_1)\rho\sigma_1\sigma_2 x_i y_j \frac{u^{n+\frac{1}{2}}_{i+1,j+1} + u^{n+\frac{1}{2}}_{i-1,j-1} - u^{n+\frac{1}{2}}_{i+1,j-1} - u^{n+\frac{1}{2}}_{i-1,j+1}}{4h^2} + \frac{u^{n+\frac{1}{2}}_{ij}}{\Delta\tau},$$

where $h = \Delta x = \Delta y$ and $\lambda_1 = \lambda_2 = 0.5$ which are the weights for each step. Here we weighted each step equally because we want to flatten the splitting effect. After solving each problem for x-axis and y-axis separately as in Figure 1, we get the solution of next time step which we expect to be the compatible numerical solution as the one from the direct solving method.

A classical error analysis indicates that the OSM for two dimensional heat equation shows same accuracy order as the direct solving method [9]. For the multi-dimensional Black-Scholes equations, unlike the heat equations, there are the cross-derivatives such as u_{xy} which would strengthen the splitting error of the OSM. Hence the convergence criterion for the Black-Scholes equations by two numerical schemes would be different than that of the heat equation. However, the mathematical error analysis has not been studied yet on the OSM for the Black-Scholes equations. We will examine this issue in the following section through numerical experiments.

3. Numerical simulation

In this section, four numerical examples are presented. First, we study numerical comparison of the accuracy between two numerical schemes, the direct solving method and the OSM, with two dimensional maximum on the call problem by varying parameters. Second, we adopt these methods to price two assets Equity Linked Security(ELS) as a real world problem. Since the analytic form solutions are not known for ELS, we compared numerical solutions with the *Monte Carlo simulation* as the benchmark. The third example consists of the parallel implementation of the OSM by MPI. We checked that the efficiency of the OSM could be enhanced even more with the parallel computing technique. The last is the three dimensional ELS problem as an example of flexibility of the scheme.

3.1. Numerical example: 2D max on call

The first example is the two dimensional maximum on the call problem. Since the analytic form solution of this problem is known, we will compare the numerical solutions with the analytic form solution. Note that the analytic form solution of call on the maximum of two risky assets is known [4] as

$$\begin{aligned} c_{\max}(S_1,S_2,K,T) &= S_1 e^{-rT} M(y_1,d;\rho_1) + S_2 e^{-rT} M(y_2,-d+\sigma\sqrt{T};\rho_2) \\ &\quad - Ke^{-rT} \times (1 - M(-y_1+\sigma_1\sqrt{T}, -y_2+\sigma_2\sqrt{T};\rho)), \end{aligned}$$

where

$$d = \frac{\ln(S_1/S_2) + (b_1 - b_2 + \sigma^2/2)T}{\sigma\sqrt{T}},$$

$$y_1 = \frac{\ln(S_1/K) + (b_1 + \sigma_1^2/2)T}{\sigma_1\sqrt{T}}, \quad y_2 = \frac{\ln(S_2/K) + (b_2 + \sigma_2^2/2)T}{\sigma_2\sqrt{T}},$$

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2}, \quad \rho_1 = \frac{\sigma_1 - \rho\sigma_2}{\sigma}, \quad \rho_2 = \frac{\sigma_2 - \rho\sigma_1}{\sigma},$$

and

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

The computational domain of the problem is $\Omega := [0,3] \times [0,3]$ for $t \in [0, 0.5]$ with known-analytic one dimensional Black-Scholes solution as the Dirichlet boundary condition. We compare the performance of the two numerical methods by varying nodal distance and parameters in the equation. Here the exercise price and the interest rate are fixed at 1.0 and 0.05, respectively. For this problem, we have the terminal payoff such that

$$\max\left[\max(S_1,S_2) - K, 0\right],$$

where S_1, and S_2 are the asset values and K is the exercise price.

The performance will be evaluated with relative l^2 errors and relative l^∞ errors which are defined as follows [6].

$$e_{l^\infty}^{(0,0)}(t) := \frac{\max_i |(u^{\text{true}}(\mathbf{x}_i,t) - u(\mathbf{x}_i,t)|}{\max_i |u^{\text{true}}(\mathbf{x}_i,t)|},$$

$$e_{l^\infty}^{(1,0)}(t) := \frac{\max_i |\frac{\partial u^{\text{true}}}{\partial x}(\mathbf{x}_i,t) - \frac{\partial u}{\partial x}(\mathbf{x}_i,t)|}{\max_i |\frac{\partial u}{\partial x}(\mathbf{x}_i,t)|},$$

$$e_{l^2}^{(0,0)}(t) := \sqrt{\frac{\sum_i \left(u(\mathbf{x}_i,t) - u^{\text{true}}(\mathbf{x}_i,t)\right)^2}{\sum_i u^{\text{true}}(\mathbf{x}_i,t)^2}},$$

$$e_{l^2}^{(1,0)}(t) = \sqrt{\frac{\sum_i (\frac{\partial u^{\text{true}}}{\partial x}(\mathbf{x}_i,t) - \frac{\partial u}{\partial x}(\mathbf{x}_i,t))^2}{\sum_i \frac{\partial u}{\partial x}(\mathbf{x}_i,t)^2}}.$$

We also define the parabolic norm of the errors as follows.

$$pe_{l^2}^{(0,0)} := \sqrt{\int_0^T \left(e_{l^2}^{(0,0)}(t)\right)^2 dt}, \quad pe_{l^2}^{(1,0)} := \sqrt{\int_0^T \left(e_{l^2}^{(1,0)}(t)\right)^2 dt},$$

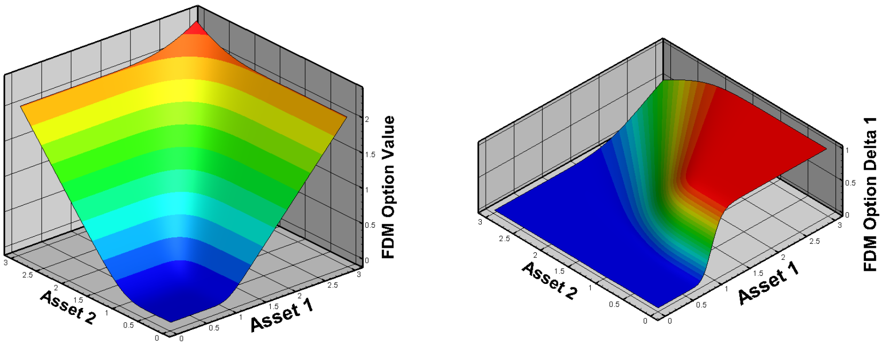

FIGURE 2. Numerical solution of 2D max on call (Option value and delta, FDM Grid: 201 × 201, $\rho = 0.4$, $\sigma_1 = 0.2$, $\sigma_2 = 0.4$)

TABLE 1. Speed and convergence for 2D max on call ($\rho = 0.4$, $\sigma_1 = 0.2$, $\sigma_2 = 0.4$). Time unit: Second

Grid size	Elapsed Time		$e^{(0,0)}_{l\infty}$ / $pe^{(0,0)}_{l\infty}$	
(Δx, Δt)	DSM	OSM	DSM	OSM
51 by 51 $(0.06, 0.015)$	0.578	0.047	1.3108e-003 2.6000e-003	1.5644e-003 2.7382e-003
101 by 101 $(0.03, 0.0075)$	4.545	0.472	5.3971e-004 1.0539e-003	6.6473e-004 1.2496e-003
201 by 201 $(0.015, 0.00375)$	41.231	2.464	2.4259e-004 4.9434e-004	3.0400e-004 6.2757e-004
401 by 401 $(0.0075, 0.001875)$	385.772	13.59	1.1504e-004 2.4664e-004	1.4552e-004 3.2669e-004
801 by 801 $(0.00375, 0.0009375)$	4159.66	83.539	5.5924e-005 1.2672e-004	7.1077e-005 1.7199e-004

$$pe^{(0,0)}_{l\infty} := \sqrt{\int_0^T \left(e^{(0,0)}_{l\infty}(t)\right)^2 dt}, \quad pe^{(1,0)}_{l\infty} := \sqrt{\int_0^T \left(e^{(1,0)}_{l\infty}(t)\right)^2 dt}.$$

In Figure 2, we depict the numerical value and *delta* of the option. The delta $\frac{\partial u}{\partial x}$ is one of the *Greeks*, which are also a main interest in the financial industry, especially for *hedging* of portfolio position. In Table 1 and Figure 3, the two numerical methods show same convergence order. Furthermore, we note that the OSM is highly efficient in computation time than the direct solving method. In Table 2, we tabulate relative errors by varying correlation parameter and

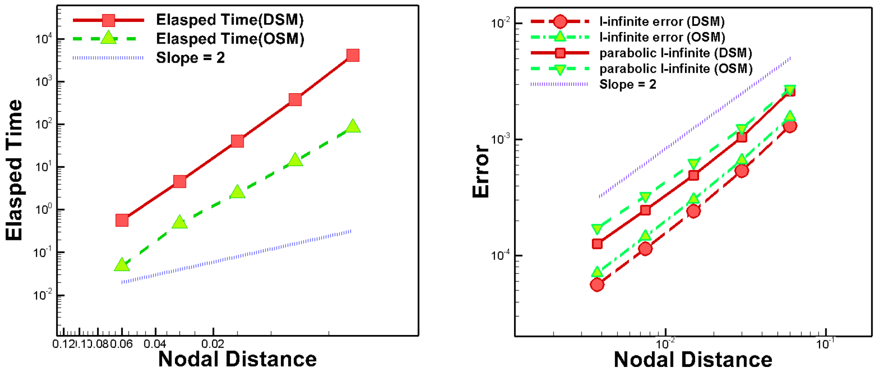

FIGURE 3. Elapsed time and convergence varying nodal distance for 2D max on call ($\rho = 0.4$, $\sigma_1 = 0.2$, $\sigma_2 = 0.4$, log scale for y-axis)

volatilities of each asset. We note that there are some differences about the errors when the parameters are relatively big such as $\rho = 0.6$, $\sigma_1 = 0.4$, and $\sigma_2 = 0.6$. This clearly shows that the splitting error stems from the cross-derivatives term. However, these bigger parameters would be considered as outliers. In the real financial market about real multi-asset derivatives such as ELS, it is very rare to consider bigger correlation than 0.5 between assets. Hence we exclude the cases of higher correlation further than 0.6.

As in the computation result, it is natural to think that the direct solving method should be more accurate than the OSM. However, the numerical result also shows that the accuracy of the OSM is not significantly different from that of the direct solving method. Even though the direct solving method shows smaller absolute errors, the overall convergence orders are same for both schemes. From these computation results, we conclude that the two methods show same order of accuracy in general on this problem.

3.2. Numerical example: 2D ELS

In this sub-section, we examine *Equity-Linked Security*(ELS) pricing problem with a real financial product.[1] The ELS is a debt security that its entity is similar to the barrier options. It has been widely traded in Korea financial market, with annual issuance volume exceeding half trillion U.S. dollars. A distinguishable feature of the ELS is the automatic early-redemption condition before its maturity, which gives the payoff structure such as Figure 4.[2]

[1] MYSTAR ELS, Product number 1741, Dongyang Securities.

[2] We note that there are more complicated ELS products in the market [5].

TABLE 2. Parameter analysis using relative errors and parabolic norms for 2D max on call (FDM Grid: 101 × 101)

		$e_{l\infty}^{(0,0)}$ / $pe_{l\infty}^{(0,0)}$		$e_{l2}^{(0,0)}$ / $pe_{l2}^{(0,0)}$	
Correlation	Volatility	DSM	OSM	DSM	OSM
$\rho = 0.2$	$\sigma_x = 0.1$	7.5245e-004	7.8955e-004	4.8041e-004	4.4764e-004
	$\sigma_y = 0.2$	1.1834e-003	1.1276e-003	4.6506e-004	4.4459e-004
	$\sigma_x = 0.2$	6.1618e-004	5.5483e-004	5.6369e-004	5.3747e-004
	$\sigma_y = 0.4$	1.2013e-003	1.1010e-003	6.4663e-004	6.2484e-004
	$\sigma_x = 0.4$	6.9570e-004	6.6153e-004	7.9280e-004	7.5155e-004
	$\sigma_y = 0.6$	1.5009e-003	1.3024e-003	9.8861e-004	9.3312e-004
$\rho = 0.4$	$\sigma_x = 0.1$	7.6286e-004	8.2201e-004	4.2123e-004	4.2407e-004
	$\sigma_y = 0.2$	1.0400e-003	1.0519e-003	4.0438e-004	4.1616e-004
	$\sigma_x = 0.2$	5.3971e-004	6.6473e-004	5.0024e-004	5.7574e-004
	$\sigma_y = 0.4$	1.0539e-003	1.2496e-003	5.7257e-004	6.5534e-004
	$\sigma_x = 0.4$	6.4607e-004	8.6250e-004	7.3343e-004	8.6064e-004
	$\sigma_y = 0.6$	1.3077e-003	1.6075e-003	8.8546e-004	1.0406e-003
$\rho = 0.6$	$\sigma_x = 0.1$	8.1264e-004	9.1223e-004	3.7085e-004	3.9679e-004
	$\sigma_y = 0.2$	9.0137e-004	9.7539e-004	3.4249e-004	3.7420e-004
	$\sigma_x = 0.2$	4.2385e-004	8.0610e-004	4.2660e-004	6.1444e-004
	$\sigma_y = 0.4$	8.5131e-004	1.3999e-003	4.8573e-004	6.7771e-004
	$\sigma_x = 0.4$	5.7522e-004	1.1788e-003	6.5533e-004	9.9792e-004
	$\sigma_y = 0.6$	1.0847e-003	1.9923e-003	7.6257e-004	1.1586e-003

Since there is no known analytic form solution for this option's value, numerical methods are used for valuation. Here we adopt the direct solving method and the OSM to compute the numerical solution. Note that the option value is not dollar-based value but the return rate. The computational domain is $\Omega = [0, 3] \times [0, 3]$ for $t \in [0, 3]$. We set the linear boundary condition which is common in financial industry [3] such that

$$\frac{\partial^2 u}{\partial x^2}(0, y, t) = \frac{\partial^2 u}{\partial x^2}(3, y, t) = 0, \qquad \frac{\partial^2 u}{\partial y^2}(x, 0, t) = \frac{\partial^2 u}{\partial y^2}(x, 3, t) = 0.$$

Since the analytic form solution is not available for the benchmark, we compare the numerical solution with the one from Monte Carlo simulation(MC) at several comparison points.[3] Figure 5 expresses the numerical option value and

[3] Two-dimensional Monte Carlo asset path is generated by the following scheme:

$$\begin{aligned} S_1(t_{n+1}) &= S_1(t_n)\exp\left\{\left(r - \frac{1}{2}\sigma_1^2\right)\Delta t + \sigma_1\sqrt{\Delta t}\Delta W_1\right\} \\ S_2(t_{n+1}) &= S_2(t_n)\exp\left\{\left(r - \frac{1}{2}\sigma_2^2\right)\Delta t + \sigma_2\sqrt{\Delta t}\left(\rho_{12}\Delta W_1 + \sqrt{1-\rho_{12}^2}\Delta W_2\right)\right\} \end{aligned}$$

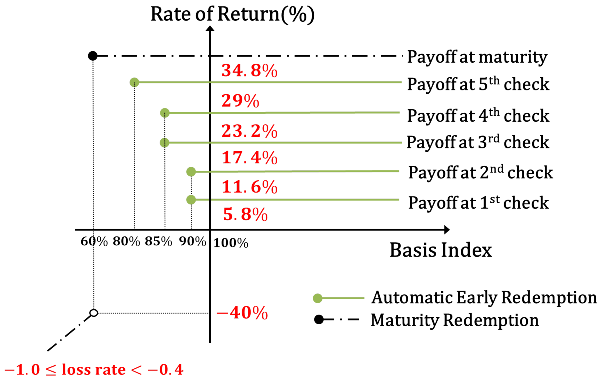

FIGURE 4. 2D ELS payoff structure

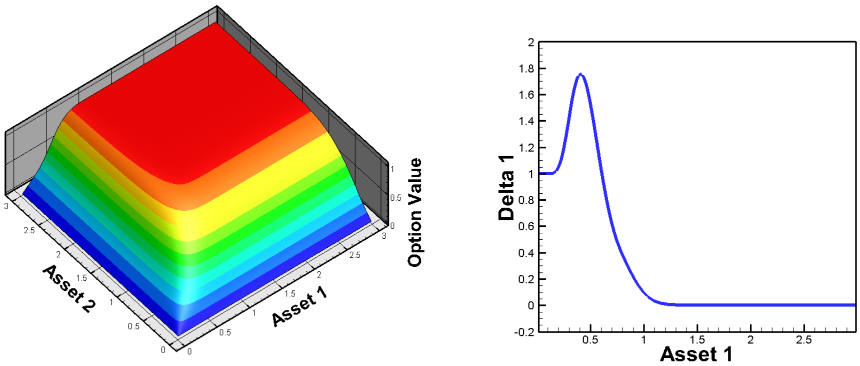

FIGURE 5. 2D ELS value(return rate) and delta 1 (Sectional view with asset 2 value = 1.0, FDM Grid: 301×301, $\sigma_1 = 0.2$, $\sigma_2 = 0.4$, $\rho = 0.5$)

its sectional view of delta for asset 1 when asset 2 is fixed at 1.0 through the direct solving method. For the direct solving method and the OSM, we used 301×301 node points. The number of Monte-Carlo paths is one million for each comparison spot. In Table 3 and Figure 6, we tabulate the difference between numerical solutions at several spots. They show that there are no significant differences between the three numerical methods for this problem in the sense of absolute difference with MC as the benchmark.

Based on the numerical results so far, we conclude that the operator splitting method is accurate as much as the direct solving method for the option pricing but it gives a huge time efficiency. This gives the reason to improve the time

Table 3. 2D ELS Value Comparison (FDM Grid: 301 × 301, MC paths:10^6, $\sigma_1 = 0.3$, $\sigma_2 = 0.3$, $\rho = 0.5$)

Comparison point	MC	DSM (Difference)	OSM (Difference)
$S_1 = S_2 = 0.81$	0.836264	0.841530 (0.005266)	0.841579 (0.005315)
$S_1 = S_2 = 0.86$	0.869240	0.875708 (0.006468)	0.875746 (0.006506)
$S_1 = S_2 = 0.90$	0.900167	0.905373 (0.005206)	0.905397 (0.005230)
$S_1 = S_2 = 0.96$	0.933335	0.937857 (0.004522)	0.937859 (0.004524)
$S_1 = S_2 = 1.00$	0.953581	0.956012 (0.004206)	0.957229 (0.005423)

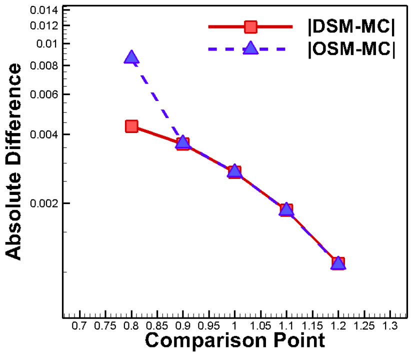

Figure 6. 2D ELS Value Comparison (FDM Grid: 301×300, MC paths: 10^6, $\sigma_x = 0.3$, $\sigma_y = 0.3$, $\rho = 0.5$)

efficiency of the OSM even more, which will be explained at the following section.

3.3. OSM by MPI implementation

Here, we examine the speed-up of the parallel implementation of the two dimensional maximum on the call option. These days, we can utilize rich resource of multi-core CPU. In a word, a serial processing can be divided into several cores and the computation time decreases as the number of cores(processors) increases. Thus, the technique known as *parallel computation* attracted the industry's attention due to its time efficiency. Among several parallel computing libraries, we choose MPICH2(in short MPI in this paper) for parallel implementation.[4] One possible application of MPI to numerical method is for the Monte-Carlo(MC) simulation. The serial asset path generation routine of the MC simulation can be divided into several computation cores without difficulty. On the other hand, direct numerical implementation for multi-dimensional PDE is

[4] We refer http://www.mpich.org/ for details.

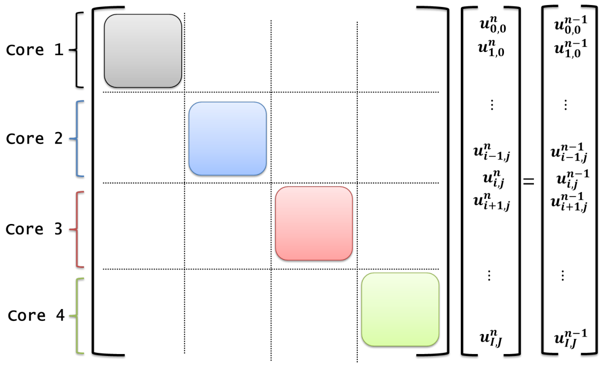

Figure 7. MPI implementation for OSM

Table 4. Elapsed time comparison for 2D max on call problem (FDM Grid: 101 × 101)

Number of Cores	1	2	4	8
Elapsed Time (sec)	0.0121768	0.0061159	0.0036957	0.0014801

known to be relatively difficult. However, we can see the OSM gives a small and tri-diagonal linear systems to solve, and this property provides appropriate environments for MPI implementation. That is, in the OSM, we solve serial block systems per node for each step as depicted in Figure 7. We tabulate the computing time of the maximum on the call pricing by the OSM with MPI in Table 4. Here we used two computers which have Linux operating system and four multi-cores respectively. We also depict the speed-up of parallelization in Figure 8. As we see from the results, the computation time almost linearly decreases as the number of the computation cores increases.

3.4. Numerical example: 3D ELS

As the last example, we choose a complicated three dimensional real financial product from the Korea financial market. We compared numerical solution by MPI implemented OSM with the MC results. There are many multi-dimensional options which are based on three or more assets in the financial market. The computation cost gets increasing exponentially as the dimensionality of underlying assets grows. This gives us the reason to utilize the OSM by the MPI implementation about three dimensional ELS pricing.[5]

[5] The referred ELS product is Product Number 2337 ELS of Dongyang Securities and its underlying assets are 1) KOSPI200, 2) HSCEI, and 3) S&P500.

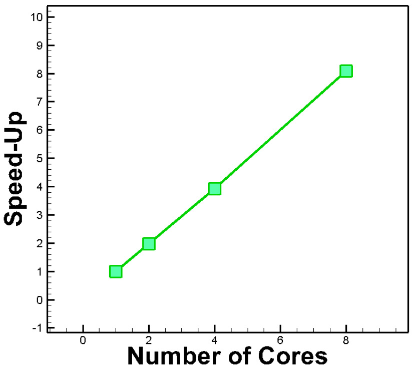

FIGURE 8. Speed-up by MPI implementation of the OSM for 2D max on call problem(Unit: Second)

Now the governing equation extends to three-dimensional equation such as

$$
\begin{aligned}
0 = {} & \frac{\partial u}{\partial t} - \frac{1}{2}\sigma_1^2 x^2 u_{xx} - \frac{1}{2}\sigma_2^2 y^2 u_{yy} - \frac{1}{2}\sigma_3^2 z^2 u_{zz} \\
& - \rho_{12}\sigma_1\sigma_2 xyu_{xy} - \rho_{23}\sigma_2\sigma_3 yzu_{yz} - \rho_{31}\sigma_3\sigma_1 zxu_{zx} \\
& - r(xu_x + yu_y + zu_z - u).
\end{aligned} \tag{5}
$$

If we apply the OSM similarly as before,

$$
\frac{\partial u}{\partial t} + \mathcal{L}_1 u + \mathcal{L}_2 u + \mathcal{L}_3 u = 0, \quad \text{where} \tag{6}
$$

$$
\begin{aligned}
\mathcal{L}_1 u &:= -\frac{1}{2}\sigma_1^2 x^2 u_{xx} - rxu_x + \frac{1}{3}ru - \frac{1}{2}\rho_{12}\sigma_1\sigma_2 xyu_{xy} - \frac{1}{2}\rho_{13}\sigma_1\sigma_3 xzu_{xz}, \\
\mathcal{L}_2 u &:= -\frac{1}{2}\sigma_2^2 y^2 u_{yy} - ryu_y + \frac{1}{3}ru - \frac{1}{2}\rho_{12}\sigma_1\sigma_2 xyu_{xy} - \frac{1}{2}\rho_{23}\sigma_2\sigma_3 yzu_{yz}, \\
\mathcal{L}_3 u &:= -\frac{1}{2}\sigma_3^2 z^2 u_{zz} - rzu_z + \frac{1}{3}ru - \frac{1}{2}\rho_{13}\sigma_1\sigma_3 xzu_{xz} - \frac{1}{2}\rho_{23}\sigma_2\sigma_3 yzu_{yz}.
\end{aligned}
$$

Then we have the discretized equations at each step such that

1) x-step

$$
\begin{aligned}
& \frac{u_{i,j,k}^{n+\frac{1}{3}}}{\Delta\tau} - \frac{1}{2}\sigma_1^2 x_i^2 \frac{u_{i-1,j,k}^{n+\frac{1}{3}} - 2u_{i,j,k}^{n+\frac{1}{3}} + u_{i+1,j,k}^{n+\frac{1}{3}}}{h^2} - rx_i \frac{u_{i+1,j,k}^{n+\frac{1}{3}} - u_{i-1,j,k}^{n+\frac{1}{3}}}{2h} + \frac{1}{3}ru_{i,j,k}^{n+\frac{1}{3}} \\
= {} & \frac{1}{2}\rho_{12}\sigma_1\sigma_2 x_i y_j \frac{u_{i+1,j+1,k}^n + u_{i-1,j-1,k}^n - u_{i+1,j-1,k}^n - u_{i-1,j+1,k}^n}{4h^2} \\
& + \frac{1}{2}\rho_{13}\sigma_1\sigma_3 x_i z_k \frac{u_{i+1,j,k+1}^n + u_{i-1,j,k-1}^n - u_{i+1,j,k-1}^n - u_{i-1,j,k+1}^n}{4h^2} + \frac{u_{i,j,k}^n}{\Delta\tau},
\end{aligned}
$$

2) y-step

$$
\frac{u_{i,j,k}^{n+\frac{2}{3}}}{\Delta\tau} - \frac{1}{2}\sigma_2^2 y_j^2 \frac{u_{i,j-1,k}^{n+\frac{2}{3}} - 2u_{i,j,k}^{n+\frac{2}{3}} + u_{i,j+1,k}^{n+\frac{2}{3}}}{h^2} - ry_j \frac{u_{i,j+1,k}^{n+\frac{2}{3}} - u_{i,j-1,k}^{n+\frac{2}{3}}}{2h} + \frac{1}{3}ru_{i,j,k}^{n+\frac{2}{3}}
$$

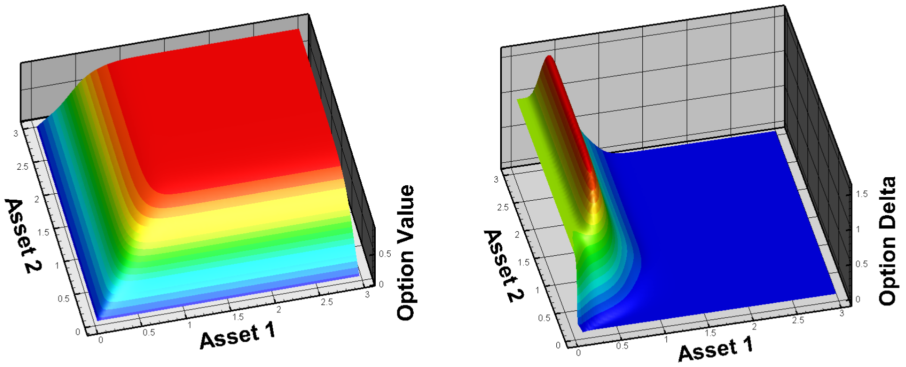

FIGURE 9. 3D ELS sectional view (Option value and delta 1, FDM Grid:101 × 101 × 101, $\sigma_1 = 0.2$, $\sigma_2 = 0.3$, $\sigma_3 = 0.4$, $\rho_{12} = \rho_{23} = \rho_{31} = 0.5$)

$$\begin{aligned}
&= \frac{1}{2}\rho_{12}\sigma_1\sigma_2 x_i y_j \frac{u^{n+\frac{1}{3}}_{i+1,j+1,k} + u^{n+\frac{1}{3}}_{i-1,j-1,k} - u^{n+\frac{1}{3}}_{i+1,j-1,k} - u^{n+\frac{1}{3}}_{i-1,j+1,k}}{4h^2} \\
&\quad + \frac{1}{2}\rho_{23}\sigma_2\sigma_3 y_j z_k \frac{u^{n+\frac{1}{3}}_{i,j+1,k+1} + u^{n+\frac{1}{3}}_{i,j-1,k-1} - u^{n+\frac{1}{3}}_{i,j+1,k-1} - u^{n+\frac{1}{3}}_{i,j-1,k+1}}{4h^2} + \frac{u^{n+\frac{1}{3}}_{i,j,k}}{\Delta\tau},
\end{aligned}$$

3) z-step

$$\begin{aligned}
&\frac{u^n_{i,j,k}}{\Delta\tau} - \frac{1}{2}\sigma_3^2 z_k^2 \frac{u^n_{i,j,k-1} - 2u^n_{i,j,k} + u^n_{i,j,k+1}}{h^2} - r z_k \frac{u^n_{i,j,k+1} - u^n_{i,j,k-1}}{2h} + \frac{1}{3} r u^n_{i,j,k} \\
&= \frac{1}{2}\rho_{13}\sigma_1\sigma_3 x_i z_k \frac{u^{n+\frac{2}{3}}_{i+1,j,k+1} + u^{n+\frac{2}{3}}_{i-1,j,k-1} - u^{n+\frac{2}{3}}_{i+1,j,k-1} - u^{n+\frac{2}{3}}_{i-1,j,k+1}}{4h^2} \\
&\quad + \frac{1}{2}\rho_{23}\sigma_2\sigma_3 y_j z_k \frac{u^{n+\frac{2}{3}}_{i,j+1,k+1} + u^{n+\frac{2}{3}}_{i,j-1,k-1} - u^{n+\frac{2}{3}}_{i,j+1,k-1} - u^{n+\frac{2}{3}}_{i,j-1,k+1}}{4h^2} + \frac{u^{n+\frac{2}{3}}_{i,j,k}}{\Delta\tau},
\end{aligned}$$

where $h = \Delta x = \Delta y = \Delta z$. The computational domain is now in three dimensional space such that $\Omega := [0,3] \times [0,3] \times [0,3]$ for time $t \in [0,3]$ with linear boundary condition such that

$$\begin{aligned}
\frac{\partial^2 u}{\partial x^2}(0,y,z,t) &= \frac{\partial^2 u}{\partial x^2}(3,y,z,t) = 0, \\
\frac{\partial^2 u}{\partial y^2}(x,0,z,t) &= \frac{\partial^2 u}{\partial y^2}(x,3,z,t) = 0, \\
\frac{\partial^2 u}{\partial z^2}(x,y,0,t) &= \frac{\partial^2 u}{\partial z^2}(x,y,3,t) = 0.
\end{aligned}$$

Note that the payoff function is essentially same as the case of 2D ELS, although this time three assets are checked for the early-redemption condition. Figure 9 represents a sectional numerical result with the third asset value is fixed. We could check that the option value looks similar with 2D ELS but the value gets

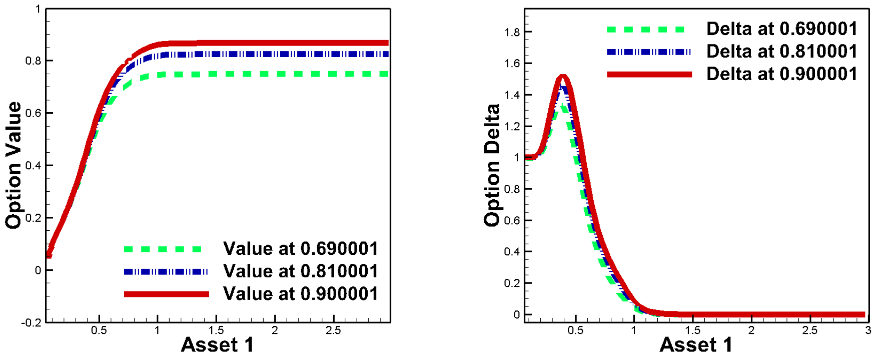

FIGURE 10. 3D ELS sectional view (Option value and delta 1, FDM Grid: 101 × 101 × 101, $\sigma_1 = 0.2$, $\sigma_2 = 0.3$, $\sigma_3 = 0.4$, $\rho_{12} = \rho_{23} = \rho_{31} = 0.5$)

increasing as an asset value does such as Figure 10. We also compared the result with MC as a benchmark.[6] Table 5 and Figure 11 show absolute differences between numerical solutions of OSM and MC at several comparison points. Again, the numerical result implies that there are no significant differences between two methods, OSM and MC. Therefore, we could successively utilize the OSM by MPI implementation to three dimensional ELS pricing.

4. Conclusion

In this paper, we examined two numerical schemes for multi-dimensional Black-Scholes equations. In the direct solving method, we solve the equation by whole dimension at once, which requires to solve the large size of linear system. Although it is practically tractable to implement, there is an issue of the computational cost for multi-dimensional problems. An alternative is

[6] Three-dimensional Monte Carlo asset path is generated by the following scheme:

$$\begin{aligned}
S_1(t_{n+1}) &= S_1(t_n)\exp\left\{\left(r-\frac{1}{2}\sigma_1^2\right)\Delta t+\sigma_1\sqrt{\Delta t}\Delta W_1\right\}\\
S_2(t_{n+1}) &= S_2(t_n)\exp\left\{\left(r-\frac{1}{2}\sigma_2^2\right)\Delta t+\sigma_2\sqrt{\Delta t}\left(\rho_{12}\Delta W_1+\sqrt{1-\rho_{12}^2}\Delta W_2\right)\right\}\\
S_3(t_{n+1}) &= S_3(t_n)\exp\left\{\left(r-\frac{1}{2}\sigma_3^2\right)\Delta t+\sigma_3\sqrt{\Delta t}\left(\rho_{13}\Delta W_1+\frac{\rho_{23}-\rho_{13}\rho_{12}}{\sqrt{1-\rho_{12}^2}}\Delta W_2\right.\right.\\
&\qquad\left.\left.+\sqrt{\frac{1+2\rho_{23}\rho_{12}\rho_{13}-\rho_{12}^2-\rho_{13}^2-\rho_{23}^2}{1-\rho_{12}^2}}\Delta W_3\right)\right\}
\end{aligned}$$

Table 5. 3D ELS value comparison (FDM Grid: 101 × 101 × 101, MC paths: 10^6, $\sigma_1 = 0.2$, $\sigma_2 = 0.3$, $\sigma_3 = 0.4$, $\rho_{12} = \rho_{23} = \rho_{31} = 0.5$)

Comparison point	MC	OSM	Difference
$S_1 = S_2 = S_3 = 0.81$	0.736690	0.748769	0.012079
$S_1 = S_2 = S_3 = 0.90$	0.813797	0.823149	0.009352
$S_1 = S_2 = S_3 = 1.02$	0.894725	0.900774	0.006049
$S_1 = S_2 = S_3 = 1.11$	0.936352	0.941243	0.004891
$S_1 = S_2 = S_3 = 1.20$	0.964224	0.968576	0.004352

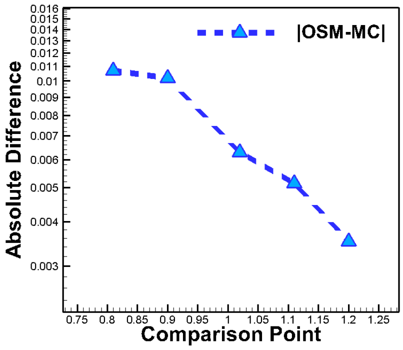

Figure 11. 3D ELS value comparison (FDM Grid: 101 × 101 × 101, MC paths: 10^6, $\sigma_1 = 0.2$, $\sigma_2 = 0.3$, $\sigma_3 = 0.4$, $\rho_{12} = \rho_{23} = \rho_{31} = 0.5$)

the Operator Splitting Method(OSM) which brings huge efficiency in computational cost compared to the direct solving method. However, one might wonder whether there are splitting errors which depend on the parameters of Black-Scholes equation. Through multi-dimensional examples, we could reasonably conclude that the two methods, the direct solving method and the OSM, are practically compatible when it comes to the option pricing problems.

The OSM not only gives the same accuracy in error as the direct solving method, but it also provides huge efficiency. Moreover, we could utilize even better efficiency by combining it with the parallel computation technique. We adopted the technique to either two or three dimensional option pricing problems. Hence, with the OSM, we could solve multi-dimensional Black-Scholes equations accurately as the direct solving method and more efficiently than that.

References

[1] Y. Daoud and T. Ozis, *The Operator Splitting Method for Black-Scholes Equation*, Appl. Math. (Irvine) **2** (2011), no. 6, 771–778.

[2] D. J. Duffy, *Finite Difference Methods in Financial Engineering*, Wiley, 2006.

[3] R. Kangro and R. Niclaides, *Far field boundary conditions for Black-Scholes equations*, SIAM Journal on Numerical Analysis **38** (2001), no. 4, 1357–1368.

[4] E. G. Haug, *The Complete Guide To Option Pricing Formulas*, McGraw-Hill, 2006.

[5] Y. Kim, H.-O. Bae, and H. Roh, *FDM Algorithm for Pricing of ELS with Exit-Probability*, Korea Derivative Association **19** (2011), no. 4, 428–446.

[6] Y. Kim, T.-C. Jo, and H.-O Bae, *A hybrid method for pricing options with multi-underlying assets*, Working paper, 2012 SIAM Annual Meeting, July 11, 2012, USA.

[7] J. B. Perot, *An analysis of the fractional step method*, J. Comput. Phys. **108** (1993), no. 1, 51–58.

[8] O. Schenk, M. Bollhoefer, and R. Roemer, *On large-scale diagonalization techniques for the Anderson model of localization*, SIAM Rev. **50** (2008), no. 1, 91–112.

[9] J. W. Thomas, *Numerical Partial Differential Equations*: *Finite Difference Methods*, Springer, 1995.

[10] N. N. Yanenko, *The Method of Fractional Steps*, Springer, Berlin, 1971.

Joonglee Jo
Department of Financial Engineering
Ajou University
Suwon 443-749, Korea
E-mail address: jogoon@ajou.ac.kr

Yongsik Kim
Department of Financial Engineering
Ajou University
Suwon 443-749, Korea
E-mail address: yskim67@ajou.ac.kr