

## COMPARISON OF NUMERICAL METHODS (BI-CGSTAB, OS, MG) FOR THE 2D BLACK–SCHOLES EQUATION

DARAE JEONG<sup>a</sup>, SUNGKI KIM<sup>b</sup>, YONGHO CHOI<sup>c</sup>, HYEONGSEOK HWANG<sup>d</sup> AND JUNSEOK KIM<sup>e,\*</sup>

**ABSTRACT.** In this paper, we present a detailed comparison of the performance of the numerical solvers such as the biconjugate gradient stabilized, operator splitting, and multigrid methods for solving the two-dimensional Black–Scholes equation. The equation is discretized by the finite difference method. The computational results demonstrate that the operator splitting method is fastest among these solvers with the same level of accuracy.

### 1. INTRODUCTION

Black and Scholes [2] derived the Black–Scholes (BS) partial differential equation for the valuation of a European option under the no-arbitrage assumption. Various types of exotic options are popular in the market. Finding the analytic closed-form solution of the BS equation is not easy. Therefore, it is necessary to apply numerical methods to obtain the values of exotic options. The finite difference methods (FDM), which converts the differential equations into a system of difference equations, are very popular to approximate the solution of the BS equations [5]. There have been many numerical methods and among them, we focus on biconjugate gradient stabilized (Bi-CGSTAB) [21], operator splitting (OS) [11] and multigrid (MG) [16] methods in this paper.

Bi-CGSTAB method was introduced by H.A. van der Vorst [21], which is similar to conjugate gradient stabilized (CGS) method with favorable stability properties. As a iterative type method, Bi-CGSTAB method is appropriate to solve the problem when the coefficient matrix of problem is large and sparse. MG method introduced

---

Received by the editors March 1, 2014. Revised April 16, 2014. Accepted April 25, 2014.

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

*Key words and phrases.* Black-Scholes equation, finite difference method, bi-CGSTAB, operator splitting method, multigrid.

\*Corresponding author.

by R.P. Fedorenko [6, 7] is numerical algorithm using a hierarchy of discretizations. By employing different mesh size, a multigrid algorithms are combined by smoothers and coarse-grid correction procedures. For this reason, this method provides rapid convergence rates than the standard iterative techniques such as the Jacobi and Gauss–Seidel schemes. There have been applications in option pricing by many researchers [12, 15, 16, 17]. On option pricing, OS method was proposed by S. Ikonen and J. Toivanen [11]. This method is by decoupling a complex equation in various simpler equations and solving the simpler equation with discretization. Since then, many researchers [4, 5, 12] have applied OS method to the BS equation.

For different types of problems, different system solvers gain advantages over the other methods, see [19]. To show the performance of the finite difference schemes for the two-dimensional problems, we compare the well-known solvers, Bi-CGSTAB, OSM, and MG methods, for the two-dimensional BS equations. There also have been other system solvers, such as alternating direction method (ADI) [3] and generalized minimal residual algorithm (GMRES) [13, 18], however we omit the comparison in this work since GMRES and ADI methods are similar to Bi-CGSTAB and OS methods, respectively. The outline of the paper is as follows. In Section 2, we first set up the problem to price stock options. In Section 3, we describe the general setting of numerical strategies and explain different solvers of linear system. In Section 4, we show the comparison of the numerical experiments between the solvers. The conclusions are drawn in Section 5.

## 2. BLACK-SCHOLES EQUATIONS

Let  $s_i(t)$ ,  $i = 1, 2, \dots, n$ , be the price of  $i$ -th asset at time  $t$  and be the unique solution to a geometric Brownian motion with a constant volatility  $\sigma_i > 0$ ,  $i = 1, 2, \dots, n$ . Let  $S = (s_1, s_2, \dots, s_n)$  be the vector of asset prices and  $\rho_{ij}$ ,  $i, j = 1, \dots, n$ , be the correlation coefficients between Brownian motions. We assume that the interest rate is constant,  $V(S, t)$  is the value of a European option that underlies assets  $1, \dots, n$ ,  $T$  is the expiration date, and  $\Lambda(S)$  is the payoff function. By following the ‘no-arbitrage’ argument for the BS equation, a partial differential equation for  $V$  is derived to be

$$(2.1) \quad \frac{\partial V}{\partial t} + \sum_{i=1}^n r s_i \frac{\partial V}{\partial s_i} + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \sigma_i \sigma_j \rho_{ij} s_i s_j \frac{\partial^2 V}{\partial s_i \partial s_j} - rV = 0.$$

In this paper, we use the original BS model with two underlying assets to keep

this presentation simple. However, we can easily extend the current method for more than two underlying assets [1]. Let us consider the computational domain  $\Omega = (0, L) \times (0, M)$  and  $x = s_1$  and  $y = s_2$ . Let us first convert the given backward equation (2.1) to the following forward equation by a change of variable  $\tau = T - t$ ,  $u(x, y, \tau) = V(s_1, s_2, T - \tau)$ :

$$(2.2) \quad \frac{\partial u}{\partial \tau} = \frac{1}{2}\sigma_1^2 x^2 \frac{\partial^2 u}{\partial x^2} + \sigma_1 \sigma_2 \rho x y \frac{\partial^2 u}{\partial x \partial y} + \frac{1}{2}\sigma_2^2 y^2 \frac{\partial^2 u}{\partial y^2} + r x \frac{\partial u}{\partial x} + r y \frac{\partial u}{\partial y} - r u,$$

$$u(x, y, 0) = \Lambda(x, y) \quad \text{for } (x, y, \tau) \in \Omega \times (0, T].$$

We use the following linear boundary conditions on all boundaries,

$$\frac{\partial^2 u}{\partial x^2}(0, y, \tau) = \frac{\partial^2 u}{\partial x^2}(L, y, \tau) = \frac{\partial^2 u}{\partial y^2}(x, 0, \tau) = \frac{\partial^2 u}{\partial y^2}(x, M, \tau) = 0.$$

### 3. NUMERICAL METHOD

In this paper, we discretize the partial derivatives in Eq. (2.2) using finite difference methods that have been used in option pricing.

**3.1. Discretization** Let us first discretize the given computational domain  $\Omega = (0, L) \times (0, M)$  as a uniform grid with a space step  $h = L/N_x = M/N_y$  and a time step  $\Delta\tau = T/N_\tau$ . Here,  $N_x$  and  $N_y$  are the number of grid points, and  $N_\tau$  is the total number of time steps. Let the numerical approximations of the solution be  $u_{ij}^n \approx u((i - 0.5)h, (j - 0.5)h, n\Delta\tau)$ , where  $i = 1, \dots, N_x$ ,  $j = 1, \dots, N_y$ , and  $n = 0, 1, \dots, N_\tau$ . We use  $\partial u / \partial x \approx (u_{i+1,j} - u_{ij}) / h$ ,  $\partial^2 u / \partial x^2 \approx (u_{i-1,j} - 2u_{ij} + u_{i+1,j}) / h^2$ ,  $\partial^2 u / \partial x \partial y \approx (u_{i+1,j+1} + u_{ij} - u_{i,j+1} - u_{i+1,j}) / h^2$ , and  $\partial u / \partial \tau \approx (u^{n+1} - u^n) / \Delta\tau$ .

**3.2. Bi-CGSTAB** The bi-conjugate gradient stabilized method (Bi-CGSTAB) was developed to solve nonsymmetric linear systems [21]. We solve Eq. (2.2) by Bi-CGSTAB method. We write Eq. (2.2) in a discretized form:

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta\tau} = \mathcal{L}_{BS} u_{ij}^{n+1},$$

where

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

Next, to renumber the multi-indexed data  $u_{ij}$  as the single-indexed data  $U_l$ , we denote by  $U_l = U_{N_x(j-1)+i} = u_{ij}$ , where  $l = 1, \dots, N_x \times N_y$ ,  $i = 1, \dots, N_x$ , and  $j = 1, \dots, N_y$ . Consequently, we get the following system

$$(3.1) \quad A\mathbf{U}^{n+1} = \mathbf{b}^n,$$

where  $\mathbf{U}^{n+1} = (U_1^{n+1}, \dots, U_{N_x \times N_y}^{n+1})$ ,  $\mathbf{b}^n = (U_1^n/\Delta\tau, \dots, U_{N_x \times N_y}^n/\Delta\tau)$ , and matrix  $A$  is composed of coefficients of  $\mathbf{U}$ . To solve the linear system (3.1), Bi-CGSTAB starts with an initial guess  $\mathbf{U}^0$  and proceeds as follows:

### Bi-CGSTAB cycle

Define the maximum number of iteration  $ITER$  and the error tolerance  $TOL$

Set  $\mathbf{r}^0 = \mathbf{b} - A\mathbf{U}^0$ ,  $\hat{\mathbf{r}}^0 = \mathbf{r}^0$ ,  $\rho^0 = \alpha = \omega^0 = 1$ ,  $\mathbf{v}^0 = \mathbf{p}^0 = \mathbf{0}$ ,  $k = 1$

While ( $k \leq ITER$  &  $\|\mathbf{r}^k\|_2 > TOL$ )

$$\rho^k = \sum_{i=1}^N \hat{r}_i^0 r_i^{k-1}, \quad \beta = \alpha \rho^k / (\rho^{k-1} \omega^{k-1})$$

$$\mathbf{p}^k = \mathbf{r}^{k-1} + \beta(\mathbf{p}^{k-1} - \omega^{k-1} \mathbf{v}^{k-1}), \quad \mathbf{v}^k = A\mathbf{p}^k, \quad \alpha = \rho^k / \sum_{i=1}^N \hat{r}_i^0 v_i^k$$

$$\mathbf{s} = \mathbf{r}^{k-1} - \alpha \mathbf{v}^k, \quad \mathbf{t} = A\mathbf{s}, \quad \omega^k = \sum_{i=1}^N t_i s_i / \sum_{i=1}^N t_i^2$$

$$\mathbf{U}^k = \mathbf{U}^{k-1} + \alpha \mathbf{p}^k + \omega^k \mathbf{s}, \quad \mathbf{r}^k = \mathbf{s} - \omega^k \mathbf{t}, \quad k = k + 1$$

End While

**3.3. Operator splitting method** The basic idea of operator splitting method is to split the spatial operator into one-dimensional operators and then fractional time steps are performed with these simpler operators. The operator splitting method computes the solutions in two time steps:

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta\tau} = \mathcal{L}_{BS}^x u_{ij}^* + \mathcal{L}_{BS}^y u_{ij}^{n+1},$$

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

$$\begin{aligned} \mathcal{L}_{BS}^x u_{ij}^* &= \sigma_1^2 x_i^2 \frac{u_{i-1,j}^* - 2u_{ij}^* + u_{i+1,j}^*}{2h^2} + r x_i \frac{u_{i+1,j}^* - u_{ij}^*}{h} - \frac{1}{2} r u_{ij}^* \\ &\quad + \frac{1}{2} \sigma_1 \sigma_2 \rho x_i y_j \frac{u_{i+1,j+1}^n + u_{ij}^n - u_{i,j+1}^n - u_{i+1,j}^n}{h^2}, \\ \mathcal{L}_{BS}^y u_{ij}^{n+1} &= \sigma_2^2 y_j^2 \frac{u_{i,j-1}^{n+1} - 2u_{ij}^{n+1} + u_{i,j+1}^{n+1}}{2h^2} + r y_j \frac{u_{i,j+1}^{n+1} - u_{ij}^{n+1}}{h} - \frac{1}{2} r u_{ij}^{n+1} \end{aligned}$$

$$+\frac{1}{2}\sigma_1\sigma_2\rho x_i y_j \frac{u_{i+1,j+1}^* + u_{ij}^* - u_{i,j+1}^* - u_{i+1,j}^*}{h^2}.$$

In OS method, we first solve  $(u_{ij}^* - u_{ij}^n)/\Delta\tau = \mathcal{L}_{BS}^x u_{ij}^*$ , and then we solve  $(u_{ij}^{n+1} - u_{ij}^*)/\Delta\tau = \mathcal{L}_{BS}^y u_{ij}^{n+1}$ .

**3.4. Multigrid method** Multigrid methods belong to the class of fastest iterations, because their convergence rate is independent of the step size  $h$ , see [8]. We define a discrete domain by  $\Omega_k = \{(h(i-0.5), h(j-0.5)) | 1 \leq i, j \leq 2^{k+1}\}$ .  $\Omega_{k-1}$  is coarser than  $\Omega_k$  by factor 2. The multigrid solution of the discrete BS equation

$$(3.2) \quad \frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta\tau} = \mathcal{L}_{BS} u_{ij}^{n+1}$$

makes use of a hierarchy of meshes created by successively coarsening the original mesh, see Fig. 1.

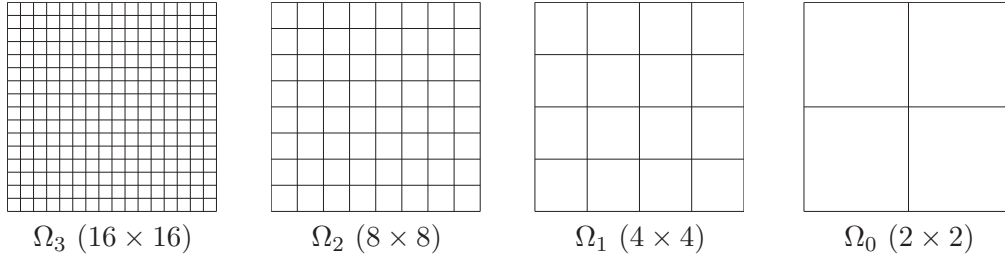


Figure 1. A sequence of coarse grids starting with  $h$ .

We use a multigrid cycle to solve the discrete system at the implicit time level. A pointwise Gauss–Seidel relaxation scheme is used as the smoother in the multigrid method. We first rewrite the above equation (3.2) by  $L(u_{ij}^{n+1}) = u_{ij}^n$  for each  $(i, j) \in \Omega_k$ , where  $L(u_{ij}^{n+1}) = u_{ij}^{n+1} - \Delta\tau \mathcal{L}_{BS} u_{ij}^{n+1}$ . Given the number  $\nu_1$  and  $\nu_2$  of pre- and post-smoothing relaxation sweeps, an iteration step for the multigrid method using the V-cycle is formally written as follows [20]. We use a notation  $u_k^n$  as a numerical solution on the discrete domain  $\Omega_k$  at time  $t = n\Delta\tau$ . Given  $u_k^n$ , we want to find  $u_k^{n+1}$  solution which satisfies equation (3.2). At the very beginning of the multigrid cycle the solution from the previous time step is used to provide an initial guess for the multigrid procedure. First, let  $u_k^{n+1,0} = u_k^n$ . The algorithm of the multigrid method for solving the discrete BS equation (3.2) is following:

#### Multigrid cycle

$$u_k^{n+1,m+1} = \text{MGcycle}(k, u_k^{n+1,m}, L_k, u_k^n, \nu_1, \nu_2).$$

*Step 1) Presmoothing:* perform  $\nu_1$  Gauss-Seidel relaxation steps.

$$(3.3) \quad \bar{u}_k^{n+1,m} = \text{SMOOTH}^{\nu_1}(u_k^{n+1,m}, L_k, u_k^n),$$

*Step 2) Coarse grid correction*

- Compute the residual on  $\Omega_k$ :  $\bar{d}_k^m = u_k^n - L_k(\bar{u}_k^{n+1,m})$ .
- Restriction to  $\Omega_{k-1}$ :  $\bar{d}_{k-1}^m = I_k^{k-1} \bar{d}_k^m$ ,  $\bar{u}_{k-1}^{n+1,m} = I_k^{k-1} \bar{u}_k^{n+1,m}$ .
- Compute an approximation solution on  $\Omega_{k-1}$ :

$$(3.4) \quad L_{k-1}(u_{k-1}^{n+1,m}) = \bar{d}_{k-1}^m.$$

- Solve the equation (3.4):

$$\hat{u}_{k-1}^{n+1,m} = \begin{cases} \text{MGcycle}(k-1, \bar{u}_{k-1}^{n+1,m}, L_{k-1}, \bar{d}_{k-1}^m, \nu_1, \nu_2) & \text{for } k > 1 \\ \text{apply the smoothing procedure in (3.3)} & \text{for } k = 1. \end{cases}$$

- Interpolate the correction:  $\hat{u}_k^m = I_{k-1}^k \hat{u}_{k-1}^m$ .
- Compute the corrected approximation on  $\Omega_k$ :  $u_k^m$ , after *CGC* =  $\bar{u}_k^{n+1,m} + \hat{u}_k^m$ .

*Step 3) Postsmoothing:*  $u_k^{n+1,m+1} = \text{SMOOTH}^{\nu_2}(u_k^m, \text{after } CGC, L_k, u_k^n)$ .

#### 4. COMPUTATIONAL RESULTS

In this section, we compare the performance of the numerical methods (Bi-CGSTAB, OS, and MG) using CPU times. Each method is implemented using MATLAB [14]. We consider three types of two-asset cash-or-nothing options. The cash-or-nothing options are useful building blocks for constructing more complex exotic option products and they are widely traded in the real world financial market.

*Case 1:* A two asset cash-or-nothing call pays out a fixed cash amount  $K$  if asset one,  $x$ , is above the strike  $X_1$  and asset two,  $y$ , is above strike  $X_2$  at expiration. The payoff is given by

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

*Case 2. and Case 3.:*

$$\Lambda(x, y) = \begin{cases} K & \text{if } x \leq X_1 \text{ and } y \leq X_2, \\ 0 & \text{otherwise.} \end{cases}, \quad \Lambda(x, y) = \begin{cases} K & \text{if } x \geq X_1 \text{ and } y \leq X_2, \\ 0 & \text{otherwise.} \end{cases}$$

Figures 2(a), (b), and (c) show the payoff function  $\Lambda(x, y)$  for *Case 1*, *Case 2*, and *Case 3*, respectively. The closed-form solutions [9] are *Case 1*:  $u(x, y, T) = Ke^{-rT} M(\alpha, \beta; \rho)$ , *Case 2*:  $u(x, y, T) = Ke^{-rT} M(-\alpha, -\beta; \rho)$ , *Case 3*:  $u(x, y, T) =$

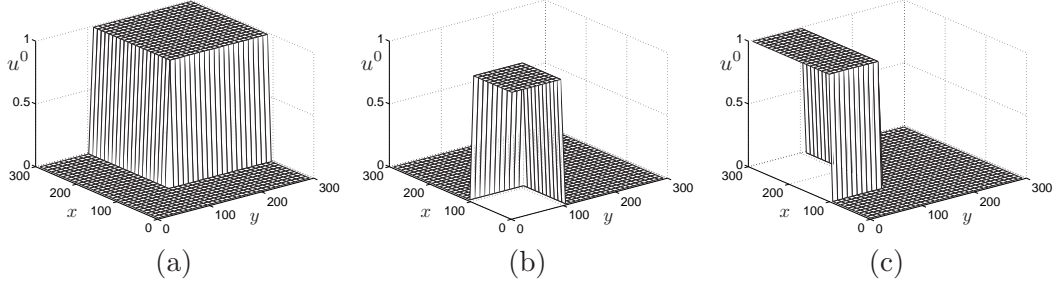


Figure 2. Payoff functions of (a) Case 1, (b) Case 2, and (c) Case 3, respectively.

$Ke^{-rT}M(-\alpha, \beta; -\rho)$ , where  $\alpha = [\ln(x/X_1) + (r - \sigma_1^2/2)T]/(\sigma_1\sqrt{T})$ ,  $\beta = [\ln(y/X_2) + (r - \sigma_2^2/2)T]/(\sigma_2\sqrt{T})$  [10]. Let  $\rho$  be the correlation between the two variables, then

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

We computed the numerical solution on uniform grids,  $h = 300/2^n$  for  $n = 5, 6, 7$ , and 8 on the computational domain  $\Omega = [0, 300] \times [0, 300]$ . For each case, we ran the calculation to time  $T = 1$  with a uniform time step  $\Delta\tau = 0.01$  with a given strike price of  $X_1 = 100$ ,  $X_2 = 100$  and cash amount  $K = 1$ . The volatilities are  $\sigma_1 = 0.3$ ,  $\sigma_2 = 0.3$  with a correlation  $\rho = 0.5$ , and the riskless interest rate  $r = 0.03$ . Figure 3 shows the numerical solution at  $T = 1$  case by case. We let  $\mathbf{e}$  be the matrix with components  $e_{ij} = u(x_i, y_j) - U_{ij}$  and compute its discrete  $l_2$ -norm of the error,  $\|\mathbf{e}\|_2$ .

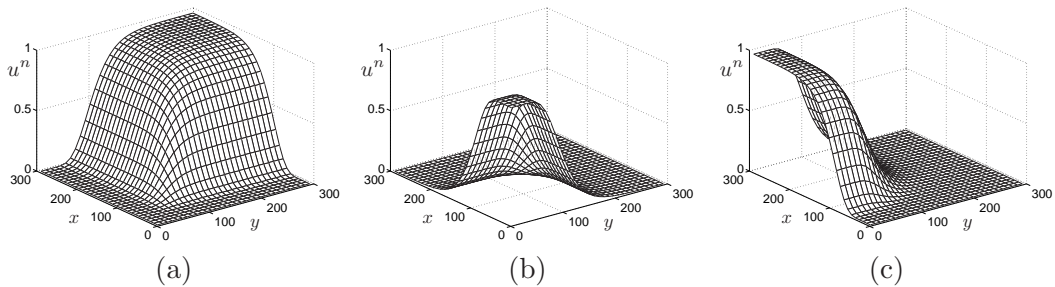


Figure 3. Numerical solutions at time  $T = 1$  of (a) Case 1, (b) Case 2, and (c) Case 3, respectively.

We test the numerical experiments of different case with three solvers, Bi-CGSTAB, OSM and MG. To make a fair comparison of these solvers, we match the accuracy of these solvers by changing iteration parameters.

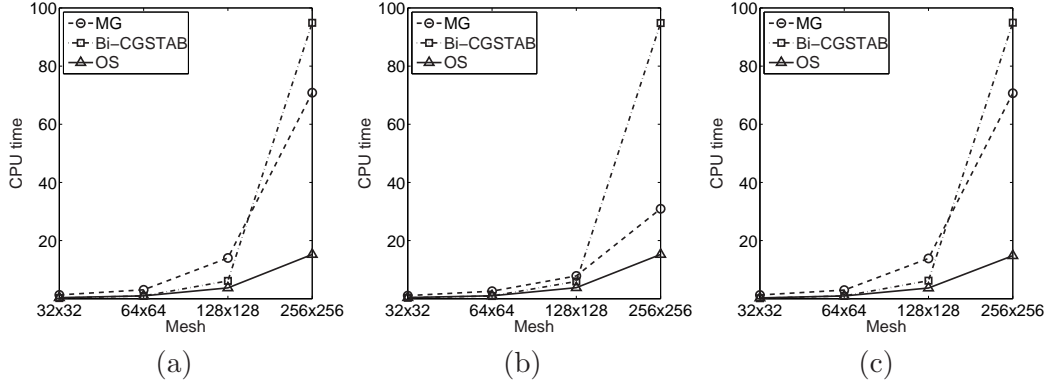


Figure 4. CPU times of (a) *Case 1*, (b) *Case 2*, and (c) *Case 3*, respectively.

In this figures, the solid line with triangles, the dash-dot line with squares, and the dashed line with stars express OSM, BI-CGSTAB, and MG, respectively. Next, let us check the CPU times to compare efficiency of these solvers. Table 1 also shows the CPU times and  $l_2$  error with each method. We can confirm that OS method has a linear CPU time cost as the spatial domain is doubled in each direction. Table 2 and Table 3 also show the CPU times and  $l_2$  error with *Case 2* and *Case 3*. And the corresponding results are plotted in Figs. 4(b) and (c), respectively. From all these results, we can confirm that OS method is faster than other methods under the same accuracy.

**Table 1.** (*Case 1*) Comparison of  $l_2$  error and CPU time.

Mesh	Bi-CGStab		OSM		Multigrid	
	$\ e\ _2$	CPU time	$\ e\ _2$	CPU time	$\ e\ _2$	CPU time
$32 \times 32$	0.02181	0.2340	0.02157	0.3744	0.02238	1.3104
$64 \times 64$	0.00926	0.9516	0.00930	0.9828	0.01012	3.0576
$128 \times 128$	0.00546	6.1308	0.00542	3.7284	0.00499	14.0245
$256 \times 256$	0.00216	94.8954	0.00237	15.2257	0.00258	70.8713

## 5. CONCLUSION

The main purpose of this paper is to present the performance comparison of finite difference schemes of the BS equation for stock option pricing. The large linear system, derived from the discrete BS equation, was solved by biconjugate gradient stabilized, operator splitting, and multigrid methods. The performance of these methods was compared for two asset option problems based on two-dimensional BS



**Table 2.** (Case 2) Comparison of  $l_2$  error and CPU time.

Mesh	Bi-CGStab		OSM		Multigrid	
	$\ e\ _2$	CPU time	$\ e\ _2$	CPU time	$\ e\ _2$	CPU time
$32 \times 32$	0.01458	0.3276	0.01484	0.3900	0.01667	1.0764
$64 \times 64$	0.00746	0.9360	0.00738	0.9984	0.00732	2.5896
$128 \times 128$	0.00361	5.9436	0.00368	3.8064	0.00362	7.8313
$256 \times 256$	0.00204	94.8330	0.00188	15.2257	0.00192	30.9350

**Table 3.** (Case 3) Comparison of  $l_2$  error and CPU time.

Mesh	Bi-CGStab		OSM		Multigrid	
	$\ e\ _2$	CPU time	$\ e\ _2$	CPU time	$\ e\ _2$	CPU time
$32 \times 32$	0.01465	0.1872	0.01478	0.2808	0.01616	1.2948
$64 \times 64$	0.00666	0.9516	0.00663	0.9672	0.00705	2.9796
$128 \times 128$	0.00367	6.2088	0.00370	3.6816	0.00345	13.8529
$256 \times 256$	0.00179	94.9578	0.00170	14.7889	0.00181	70.6841

equations. The numerical results indicated that although Bi-CGSTAB and multigrid solvers are accurate, they need a lot of computational times. On the other hand, operator splitting is faster than the other two methods under the same accuracy.

#### ACKNOWLEDGMENT

The corresponding author (J.S. Kim) was supported by the National Institute for Mathematical Sciences(NIMS) grant funded by the Korea government(No. A21301). And this work is based on the first author's Ph. D. thesis [12].

#### REFERENCES

1. P. Amstera, C. Averbuj, P. de Napoli & M. Mariani: A parabolic problem arising in financial mathematics. *Nonlinear Anal. Real World Appl.* **11** (2010), 759–763.
2. F. Black & M. Scholes: The pricing of options and corporate liabilities. *J. Polit. Econ.* **81** (1973), 637–659.
3. R. Chin, T. Manteuffel & J. de Pillis: ADI as a preconditioning for solving the convection-diffusion equation. *SIAM J. Sci. Comput.* **5** (1984), no. 2, 281–299.
4. Y. Daoud & T. Öziş: The operator splitting method for Black–Scholes equation. *Appl. Math.* **2** (2011), no. 6, 771–778.
5. D. Duffy: *Finite Difference Methods in Financial Engineering, A Partial Differential Equation Approach*. John Wiley and Sons, New York, USA, 2006.

6. R.P. Fedorenko: A relaxation method for solving elliptic difference equations. *USSR Computational Math. and Math. Phys.* **1** (1962), no. 4, 1092–1096.
7. R.P. Fedorenko: The speed of convergence of one iterative process. *USSR Computational Math. and Math. Phys.* **4** (1964), no. 3, 227–235.
8. W. Hackbusch: *Iterative Solution of Large Linear Systems of Equations*. Springer, New York, USA, 1994.
9. R. Heynen & H. Kat: Brick by Brick. *Risk Magazine* **9** (1996), no. 6, 28–31.
10. E. Haug: *The Complete Guide to Option Pricing Formulas*. McGraw-Hill, New York, USA, 2007.
11. S. Ikonen & J. Toivanen: Operator splitting methods for American option pricing. *Appl. Math. Lett.* **17** (2004), no. 7, 809–814.
12. D. Jeong: Mathematical model and numerical simulation in computational finance. Ph.D. Thesis, Department of Mathematics, Korea University, Korea, December, 2012.
13. P. Lotstedt, J. Persson, L. von Sydow & J. Tysk: Space-time adaptive finite difference method for European multi-asset options. *Comput. Math. Appl.* **53** (2007), no. 8, 1159–1180.
14. MATLAB, Users Guide: *Natick*. Mathworks Inc. MA, USA, 1990.
15. C.W. Oosterlee: On multigrid for linear complementarity problems with application to American-style options. *Electron. Trans. Numer. Anal.* **15** (2003), no. 2-7, 165–185.
16. A. Ramage & L. von Sydow: A multigrid preconditioner for an adaptive Black–Scholes solver. *BIT Numer. Math.* **51** (2011), no. 1, 217–233.
17. C. Reisinger & 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.
18. Y. Saad & M. Schultz: GMRES, a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. & Stat. Comput.* **7** (1986), no. 3, 856–869.
19. Y. Saad & H. van der Vorst: Iterative solution of linear systems in the 20th century. *J. Comput. Appl. Math.* **123** (2000), no. 1, 1–33.
20. U. Trottenberg, C. Oosterlee & A. Schüller: *Multigrid*. Academic press, London, England, 2001.
21. H. van der Vorst: Bi-CGSTAB, A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. & Stat. Comput.* **13** (1992), no. 2, 631–644.

<sup>a</sup>DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 136-713, REPUBLIC OF KOREA  
Email address: tinayoyo@korea.ac.kr

<sup>b</sup>DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 136-713, REPUBLIC OF KOREA  
Email address: sgkim82@korea.ac.kr

<sup>c</sup>DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 136-713, REPUBLIC OF KOREA  
Email address: poohyongho@korea.ac.kr

<sup>d</sup>DEPARTMENT OF FINANCIAL ENGINEERING, KOREA UNIVERSITY, SEOUL 136-071, REPUBLIC OF KOREA

*Email address:* `hhs288@naver.com`

<sup>e</sup>DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 136-713, REPUBLIC OF KOREA

*Email address:* `cfdkim@korea.ac.kr`