Commun. Korean Math. Soc. **24** (2009), No. 4, pp. 617–628
DOI 10.4134/CKMS.2009.24.4.617

AN ACCURATE AND EFFICIENT NUMERICAL METHOD FOR BLACK-SCHOLES EQUATIONS

Darae Jeong, Junseok Kim, and In-Suk Wee

Abstract. We present an efficient and accurate finite-difference method for computing Black-Scholes partial differential equations with multi-underlying assets. We directly solve Black-Scholes equations without transformations of variables. We provide computational results showing the performance of the method for two underlying asset option pricing problems.

1. Introduction

Black and Scholes [1], and Merton [10] derived a parabolic second order partial differential equation (PDE) for the value $u(\mathbf{s}, t)$ of an option on stocks. We propose a finite difference method to solve the generalized multi-asset Black-Scholes PDE. Let $s_i, i = 1, 2, \ldots, n$ denote the price of the underlying i-th asset and $u(s_1, s_2, \ldots, s_n, t)$ denote the value of the option. The prices s_i of the underlying assets are described by geometric Brownian motions

$$ds_i = \mu_i s_i dt + \sigma_i s_i dW_i, \quad i = 1, 2, \ldots, n,$$

where μ_i and σ_i denote a constant expected rate of return and a constant volatility of the i-th asset, respectively. Here, W_i is the standard Brownian motion. Let ρ_{ij} denote the correlation coefficient between two Brownian motions W_i and W_j where

$$dW_i dW_j = \rho_{ij} dt, \quad i, j = 1, 2, \ldots, n, i \neq j.$$

Received December 14, 2008.

2000 *Mathematics Subject Classification.* Primary 65N06, 65M55.

Key words and phrases. Black-Scholes equations, finite difference method, multigrid method.

This work was supported by the research fund (R0600842) of Seoul Research and Business Development Program.

Then, the no arbitrage principle leads to the following generalized n-asset Black-Scholes equation [7, 9, 20]:

$$\begin{aligned}(1)\qquad & \frac{\partial u(\mathbf{s},t)}{\partial t}+\frac{1}{2}\sum_{i,j=1}^{n}\sigma_i\sigma_j\rho_{ij}s_is_j\frac{\partial^2 u(\mathbf{s},t)}{\partial s_i\partial s_j}+r\sum_{i=1}^{n}s_i\frac{\partial u(\mathbf{s},t)}{\partial s_i}=ru(\mathbf{s},t)\\ & \text{for } (\mathbf{s},t)=(s_1,s_2,\ldots,s_n,t)\in\mathbf{R}_+^n\times[0,T),\end{aligned}$$

where $r > 0$ is a constant riskless interest rate. The final condition is the payoff function $u_T(\mathbf{s})$ at expiry T

$$(2)\qquad u(\mathbf{s},T)=u_T(\mathbf{s}).$$

The analytic solutions of Eqs. (1) and (2) for exotic options are very limited. Therefore, we need to rely on a numerical approximation. To obtain an approximation of the option value, we can compute a solution of Black-Scholes PDEs (1) and (2) using a finite difference method (FDM) [3, 15, 16, 17, 20].

We apply the FDM to the equation over a truncated finite domain. The original asymptotic infinite boundary conditions are shifted to the ends of the truncated finite domain. To avoid generating large errors in the solution due to this approximation of the boundary conditions, the truncated domain must be large enough resulting in large computational costs. The purpose of our work is to propose an efficient and accurate FDM to directly solve the Black-Scholes PDEs (1) and (2) without transformations of variables.

The outline of this paper is the following. In Section 2 we formulate the Black-Scholes (BS) partial differential equation with two underlying assets. In Section 3, we focus on the details of a multigrid solver for the BS equation. In Section 4, we present the results of numerical experiments. We draw conclusions in Section 5.

2. The Black-Scholes model

We use a Black-Scholes model with two underlying assets to keep this presentation simple. However, we can easily extend the current method for more than two underlying assets. Let us consider the computational domain $\Omega = (0, L) \times (0, M)$ for the two assets case. Let $x = s_1$ and $y = s_2$. Then from the change of variable $\tau = T - t$, we obtain an initial value problem:

$$\begin{aligned}(3)\qquad \frac{\partial u}{\partial \tau} &= \frac{1}{2}(\sigma_1 x)^2\frac{\partial^2 u}{\partial x^2}+\frac{1}{2}(\sigma_2 y)^2\frac{\partial^2 u}{\partial y^2}+\sigma_1\sigma_2\rho xy\frac{\partial^2 u}{\partial x\partial y}\\ &\quad +rx\frac{\partial u}{\partial x}+ry\frac{\partial u}{\partial y}-ru \quad \text{for } (x,y,\tau)\in\Omega\times(0,T],\end{aligned}$$

with an initial condition $u(x, y, 0) = u_T(x, y)$ for $(x, y) \in \Omega$. There are several possible boundary conditions such as Neumann [3, 5], Dirichlet, linear, and PDE [3, 16] that can be used for these kinds of problems. In this work, we use a linear boundary condition on all boundaries, i.e.,

$$\frac{\partial^2 u}{\partial x^2}(0,y,\tau)=\frac{\partial^2 u}{\partial x^2}(L,y,\tau)=\frac{\partial^2 u}{\partial y^2}(x,0,\tau)=\frac{\partial^2 u}{\partial y^2}(x,M,\tau)=0,$$
$$\forall\tau\in[0,T] \text{ for } 0\le x\le L,\ 0\le y\le M.$$

3. A numerical solution

3.1. Discretization with finite differences

A finite difference method is a common numerical method that has been used by many researchers in computational finance. For an introduction to these methods we recommend the books [3, 15, 16, 17, 20]. They all introduce the concept of finite differences for option pricing and provide basic knowledge needed for a simple implementation of the method. An approach for the Black-Scholes option problem is to use an efficient solver such as the Bi-CGSTAB (Biconjugate gradient stabilized) method [12, 14, 19], GMRES (Generalized minimal residual algorithm) method [11, 13], ADI (Alternating direction implicit) method [2, 3], and the OS (Operator splitting) method [3, 8].

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 $\Delta t=T/N_t$. Let us denote the numerical approximation of the solution by

$$u_{ij}^n\equiv u(x_i,y_j,t^n)=u\left((i-0.5)h,(j-0.5)h,n\Delta t\right),$$

where $i=1,\dots,N_x$ and $j=1,\dots,N_y$. We use a cell centered discretization since we use a linear boundary condition. By applying the implicit time scheme and centered difference for space derivatives to Eq. (3), we have

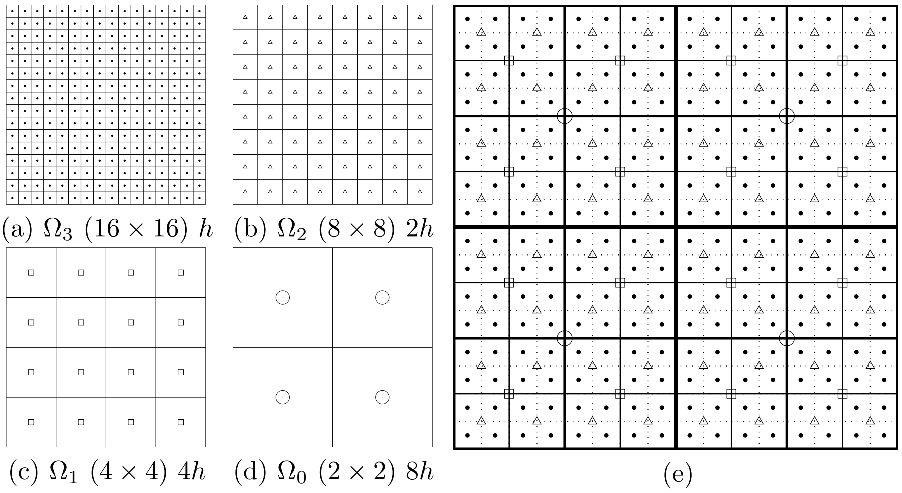
$$\frac{u_{ij}^{n+1}-u_{ij}^n}{\Delta t} = L_{BS}u_{ij}^{n+1}, \tag{4}$$

where the discrete difference operator L_{BS} is defined by

$$\begin{aligned} L_{BS}u_{ij}^{n+1} = & \frac{(\sigma_1 x_i)^2}{2}\frac{u_{i-1,j}^{n+1}-2u_{ij}^{n+1}+u_{i+1,j}^{n+1}}{h^2}\\ &+\frac{(\sigma_2 y_j)^2}{2}\frac{u_{i,j-1}^{n+1}-2u_{ij}^{n+1}+u_{i,j+1}^{n+1}}{h^2}\\ &+\sigma_1\sigma_2\rho x_i y_j\frac{u_{i+1,j+1}^{n+1}+u_{i-1,j-1}^{n+1}-u_{i-1,j+1}^{n+1}-u_{i+1,j-1}^{n+1}}{4h^2}\\ &+rx_i\frac{u_{i+1,j}^{n+1}-u_{i-1,j}^{n+1}}{2h}+ry_j\frac{u_{i,j+1}^{n+1}-u_{i,j-1}^{n+1}}{2h}-ru_{ij}^{n+1}. \end{aligned}$$

3.2. A multigrid method

Multigrid methods belong to the class of fastest iterations, because their convergence rate is independent of the space step size [4]. In order to explain



(a) Ω_3 (16×16) h (b) Ω_2 (8×8) $2h$

(c) Ω_1 (4×4) $4h$ (d) Ω_0 (2×2) $8h$

(e)

FIGURE 1. (a), (b), (c), and (d) are a sequence of coarse grids starting with $h = L/N_x$. (e) is a composition of grids, Ω_3, Ω_2, Ω_1, and Ω_0.

clearly the steps taken during a single V-cycle, we focus on a numerical solution on a 16×16 mesh. We define discrete domains, Ω_3, Ω_2, Ω_1, and Ω_0, where

$$\Omega_k = \{(x_{k,i} = (i - 0.5)h_k, y_{k,j} = (j - 0.5)h_k) \mid 1 \leq i, j \leq 2^{k+1} \text{ and } h_k = 2^{3-k}h\}.$$

Ω_{k-1} is coarser than Ω_k by a factor of 2. The multigrid solution of the discrete BS Eq. (4) makes use of a hierarchy of meshes (Ω_3, Ω_2, Ω_1, and Ω_0) created by successively coarsening the original mesh, Ω_3 as shown in Fig. 1. A pointwise Gauss-Seidel relaxation scheme is used as the smoother in the multigrid method. We use a notation u_k^n as a numerical solution on the discrete domain Ω_k at time $t = n\Delta t$. The algorithm of the multigrid method for solving the discrete BS Eq. (4) is as follows. We rewrite the above Eq. (4) by

$$L_3(u_{3,ij}^{n+1}) = \phi_{3,ij}^n \text{ on } \Omega_3, \tag{5}$$

where

$$L_3(u_{3,ij}^{n+1}) = u_{3,ij}^{n+1} - \Delta t L_{BS3} u_{3,ij}^{n+1} \text{ and } \phi_{3,ij}^n = u_{3,ij}^n.$$

Given the numbers, ν_1 and ν_2, of pre- and post- smoothing relaxation sweeps, an iteration step for the multigrid method using the V-cycle is formally written as follows [18]. That is, starting an initial condition u_3^0, we want to find u_3^n for $n = 1, 2, \ldots$. Given u_3^n, we want to find the u_3^{n+1} solution that satisfies Eq. (4). 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 $u_3^{n+1,0} = u_3^n$.

Multigrid cycle

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

That is, $u_k^{n+1,m}$ and $u_k^{n+1,m+1}$ are the approximations of u_k^{n+1} before and after an MGcycle. Now, define the MGcycle.

Step 1) Presmoothing

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

means performing ν_1 smoothing steps with the initial approximation $u_k^{n+1,m}$, source terms ϕ_k^n, and a $SMOOTH$ relaxation operator to get the approximation $\bar{u}_k^{n+1,m}$. Here, we derive the smoothing operator in two dimensions.

Now we derive a Gauss-Seidel relaxation operator. First, we rewrite Eq. (5) as

$$\begin{aligned} u_{k,ij}^{n+1} \quad = \quad & \Bigg[\phi_{k,ij}^n + \Delta t \Bigg(\frac{(\sigma_1 x_{k,i})^2}{2} \frac{u_{k,i-1,j}^{n+1} + u_{k,i+1,j}^{n+1}}{h_k^2} \\ & + \frac{(\sigma_2 y_{k,j})^2}{2} \frac{u_{k,i,j-1}^{n+1} + u_{k,i,j+1}^{n+1}}{h_k^2} \\ & + \sigma_1 \sigma_2 \rho x_{k,i} y_{k,j} \frac{u_{k,i+1,j+1}^{n+1} + u_{k,i-1,j-1}^{n+1} - u_{k,i-1,j+1}^{n+1} - u_{k,i+1,j-1}^{n+1}}{4h_k^2} \\ & + \ r x_{k,i} \frac{u_{k,i+1,j}^{n+1} - u_{k,i-1,j}^{n+1}}{2h_k} + r y_{k,j} \frac{u_{k,i,j+1}^{n+1} - u_{k,i,j-1}^{n+1}}{2h_k} \Bigg) \Bigg] \Bigg/ \\ & \left[1 + \Delta t \left(\frac{(\sigma_1 x_{k,i})^2 + (\sigma_2 y_{k,j})^2}{h_k^2} + r \right) \right]. \end{aligned} \tag{6}$$

Next, we replace $u_{k,\alpha\beta}^{n+1}$ in Eq. (6) with $\bar{u}_{k,\alpha\beta}^{n+1,m}$ if $(\alpha < i)$ or $(\alpha = i$ and $\beta \leq j)$, otherwise with $u_{k,\alpha\beta}^{n+1,m}$, i.e.,

$$\begin{aligned} \bar{u}_{k,ij}^{n+1,m} \quad = \quad & \Bigg[\phi_{k,ij}^n + \Delta t \Bigg(\frac{(\sigma_1 x_{k,i})^2}{2} \frac{\bar{u}_{k,i-1,j}^{n+1,m} + u_{k,i+1,j}^{n+1,m}}{h_k^2} \\ & + \frac{(\sigma_2 y_{k,j})^2}{2} \frac{\bar{u}_{k,i,j-1}^{n+1,m} + u_{k,i,j+1}^{n+1,m}}{h_k^2} \\ & + \sigma_1 \sigma_2 \rho x_{k,i} y_{k,j} \frac{u_{k,i+1,j+1}^{n+1,m} + \bar{u}_{k,i-1,j-1}^{n+1,m} - \bar{u}_{k,i-1,j+1}^{n+1,m} - u_{k,i+1,j-1}^{n+1,m}}{4h_k^2} \\ & + \ r x_{k,i} \frac{u_{k,i+1,j}^{n+1,m} - \bar{u}_{k,i-1,j}^{n+1,m}}{2h_k} + r y_{k,j} \frac{u_{i,j+1}^{n+1,m} - \bar{u}_{k,i,j-1}^{n+1,m}}{2h_k} \Bigg) \Bigg] \Bigg/ \\ & \left[1 + \Delta t \left(\frac{(\sigma_1 x_{k,i})^2 + (\sigma_2 y_{k,j})^2}{h_k^2} + r \right) \right]. \end{aligned} \tag{7}$$

Therefore, in a multigrid cycle, one smooth relaxation operator step consists of solving Eq. (7) given above for $1 \le i \le 2^{k-3}N_x$ and $1 \le j \le 2^{k-3}N_y$.

Step 2) Coarse grid correction

- Compute the defect: $\bar{d}_k^m = \phi_k^n - L_k(\bar{u}_k^{n+1,m})$.
- Restrict the defect and $\bar{u}_k^m$: $\bar{d}_{k-1}^m = I_k^{k-1}\bar{d}_k^m$

The restriction operator I_k^{k-1} maps k-level functions to $(k-1)$-level functions as shown in Fig. 2(a).

$$\begin{aligned} d_{k-1}(x_i, y_j) = I_k^{k-1} d_k(x_i, y_j) \quad &= \quad \frac{1}{4}[d_k(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}) + d_k(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}) \\ &\quad + d_k(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}) + d_k(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})]. \end{aligned}$$

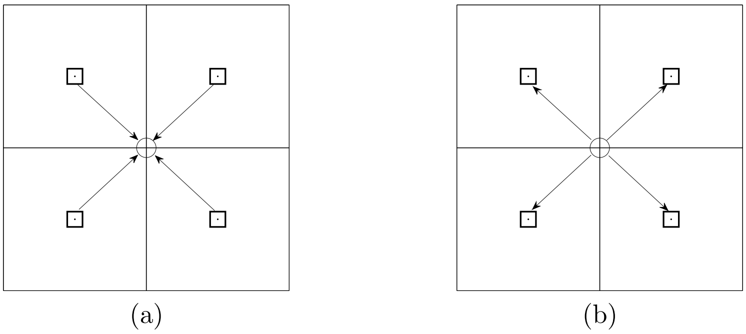


FIGURE 2. Transfer operators : (a) restriction and (b) interpolation.

- Compute an approximate solution $\hat{u}_{k-1}^{n+1,m}$ of the coarse grid equation on Ω_{k-1}, i.e.,

$$L_{k-1}(u_{k-1}^{n+1,m}) = \bar{d}_{k-1}^m. \tag{8}$$

If $k = 1$, we use a direct or fast iteration solver for (8). If $k > 1$, we solve (8) approximately by performing k-grid cycles using the zero grid function as an initial approximation:

$$\hat{v}_{k-1}^{n+1,m} = MGcycle(k-1, 0, L_{k-1}, \bar{d}_{k-1}^m, \nu_1, \nu_2).$$

- Interpolate the correction: $\hat{v}_k^{n+1,m} = I_{k-1}^k \hat{v}_{k-1}^{n+1,m}$. Here, the coarse values are simply transferred to the four nearby fine grid points as shown in Fig. 2(b), i.e., $v_k(x_i, y_j) = I_{k-1}^k v_{k-1}(x_i, y_j) = v_{k-1}(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$ for the i and j odd-numbered integers.
- Compute the corrected approximation on Ω_k

$$u_k^{m,\text{ after } CGC} = \bar{u}_k^{n+1,m} + \hat{v}_k^{n+1,m}.$$

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

This completes the description of a MGcycle. An illustration of the corresponding two-grid cycle is given in Fig. 3. For the multi-grid V-cycle, it is given in Fig. 4.

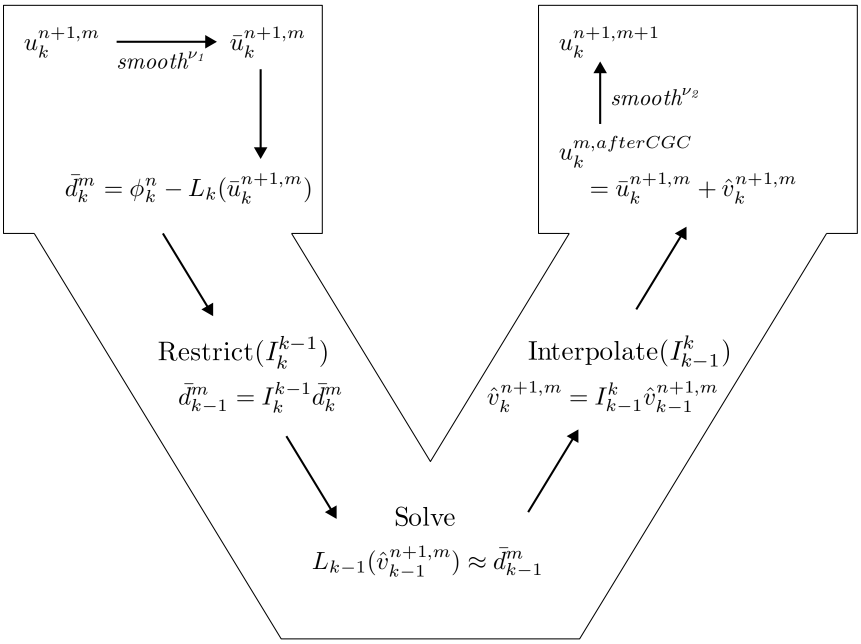

$u_k^{n+1,m}$
$smooth^{\nu_1}$
$\bar{u}_k^{n+1,m}$
$\bar{d}_k^m = \phi_k^n - L_k(\bar{u}_k^{n+1,m})$
Restrict(I_k^{k-1})
$\bar{d}_{k-1}^m = I_k^{k-1}\bar{d}_k^m$
Solve
$L_{k-1}(\hat{v}_{k-1}^{n+1,m}) \approx \bar{d}_{k-1}^m$
Interpolate(I_{k-1}^k)
$\hat{v}_k^{n+1,m} = I_{k-1}^k \hat{v}_{k-1}^{n+1,m}$
$u_k^{m,afterCGC}$
$= \bar{u}_k^{n+1,m} + \hat{v}_k^{n+1,m}$
$smooth^{\nu_2}$
$u_k^{n+1,m+1}$

FIGURE 3. The $MG\ (k, k-1)$ two-grid method.

4. Computational results

In this section, we perform a convergence test of the scheme and present several numerical experiments. Two-asset cash or nothing options can be useful building blocks for constructing more complex exotic option products. Let us consider a two-asset cash or nothing call option. This option 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

$$u(x, y, 0) = \begin{cases} K & \text{if } x \geq X_1 \text{ and } y \geq X_2, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

The formula for the exact value is known in [6] by

$$u(x, y, T) = Ke^{-rT}M(\alpha, \beta; \rho), \tag{10}$$

where

$$\alpha = \frac{\ln(x/K_1)+(r-\sigma_1^2/2)T}{\sigma_1\sqrt{T}}, \ \beta = \frac{\ln(y/K_2)+(r-\sigma_2^2/2)T}{\sigma_2\sqrt{T}}.$$

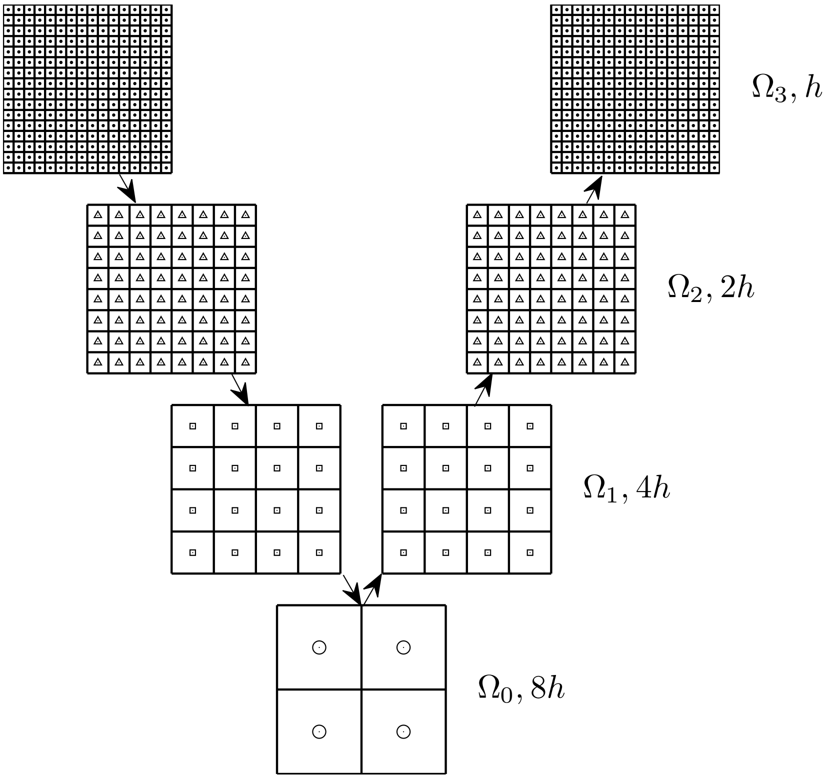


Figure 4. Schedule of grids for V-cycle.

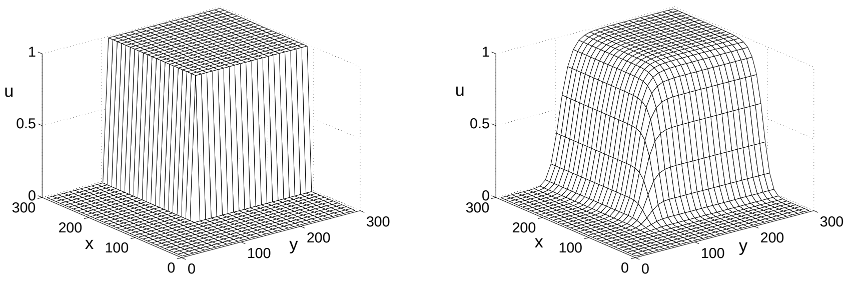
Here $M(\alpha, \beta; \rho)$ denotes a standardized cumulative normal function where one random variable is less than α and a second random variable is less than β. The correlation between the two variables is ρ:

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

The MATLAB code for the closed form solution of a two-asset cash or nothing call option is given in Appendix A.

4.1. Convergence test

To obtain an estimate of the rate of convergence, we performed a number of simulations for a sample initial problem on a set of increasingly finer grids. We considered a domain, $\Omega = [0, 300] \times [0, 300]$. We computed the numerical solutions on uniform grids, $h = 300/2^n$ for $n = 5, 6, 7,$ and 8. For each case, we ran the calculation to time $T = 0.1$ with a uniform time step depending on a mesh size, $\Delta t = 0.032/2^n$. The initial condition is Eq. (9) with $K = 1$ and $X_1 = X_2 = 100$. The volatilities are $\sigma_1 = 0.5$ and $\sigma_2 = 0.5$. The correlation is $\rho = 0.5$, and the riskless interest rate is $r = 0.03$. Figs. 5 (a) and (b) show the initial configuration and final profile at T, respectively.



(a) (b)

FIGURE 5. (a) The initial condition and (b) numerical result at $T = 0.1$.

We let $\mathbf{e}$ be the error matrix with components $e_{ij} = u(x_i, y_j) - u_{ij}$. $u(x_i, y_j)$ is the analytic solution of Eq. (10) and u_{ij} is the numerical solution. We compute its discrete L^2 norm $\|\mathbf{e}\|_2$ is defined

$$\|\mathbf{e}\|_2 = \sqrt{\frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} e_{ij}^2}.$$

The errors and rates of convergence are given in Table 1. The results show that the scheme is first-order accurate.

TABLE 1. The L^2 norms of errors and convergence rates for u at time $T = 0.1$.

Case	32×32	rate	64×64	rate	128×128	rate	256×256
$\|\mathbf{e}\|_2$	0.028161	0.95	0.014562	1.07	0.006928	0.96	0.003572

4.2. Multigrid performance

We investigated the convergence behavior of our MG method, especially mesh independence. The test problem was that of a two-asset cash or nothing call option with the convergence test parameter set. The average number of iterations per time step (see Fig. 6) and the CPU-time in seconds required for a solution to an identical convergence tolerance are displayed in Table 2.

Although the number of multigrid iterations for convergence at each time step slowly increased as the mesh was refined, from a practical viewpoint, it was essentially grid independent.

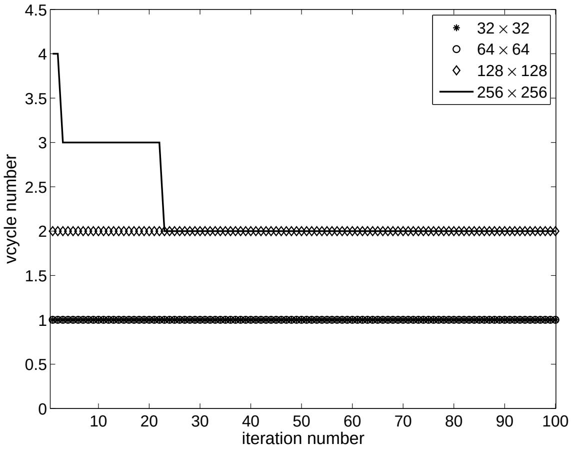


Figure 6. Number of V-cycles.

5. Conclusions

In this paper, we focused on the performance of a multigrid method for option pricing problems. The numerical results showed that the total computational cost was proportional to the number of grid points. The convergence test showed that the scheme was first-order accurate since we used an implicit Euler method. In a forthcoming paper, we will investigate a switching grid method, which uses a fine mesh when the solution is not smooth and otherwise uses a coarse mesh.

Appendix A. MATLAB code for a closed form solution

```
L=300; K=1; T=0.1; r=0.03; sigma1=0.5; sigma2=0.5; rho=0.5;
```

Table 2. Grid independence with an iteration convergence tolerance of 10^{-5}, $T = 0.1$ and $\Delta t = 0.001$.

Mesh	Average iterations per time step	CPU(s)
32 × 32	1.00	0.141
64 × 64	1.00	0.579
128 × 128	2.00	2.594
256 × 256	2.24	13.093

```
N=64; h=L/N; S=linspace(h/2,L-h/2,N); VE=zeros(N,N); mu=[0 0];

for i=1:N
      for j=1:N
         y1 = (log(S(i)/K)+(b1-sigma1^2/2)*T)/(sigma1*sqrt(T));
         y2 = (log(S(j)/K)+(b2-sigma2^2/2)*T)/(sigma2*sqrt(T));
          X = [y1 y2];
        cov = [1 rho; rho 1];
          M = mvncdf(X,mu,cov);
     V(i,j) = K*exp(-r*T)*M;
    end
end

[X, Y] = meshgrid(S); surf(X, Y, V)
```

References

[1] F. Black and M. Sholes, *The pricing of options and corporate liabilities*, J. Political Economy **81** (1973), no. 3, 637–659.

[2] R. C. Y. Chin, T. A. Manteuffel, and J. de Pillis, *ADI as a preconditioning for solving the convection-diffusion equation*, SIAM J. Sci. Stat. Comput. **5** (1984), no. 2, 281–299.

[3] D. J. Duffy, *Finite Difference Methods in Financial Engineering : a partial differential equation approach*, John Wiley and Sons, New York, 2006.

[4] W. Hackbusch, *Iterative Solution of Large Linear Systems of Equations*, Springer, New York, 1994.

[5] H. Han and X. Wu, *A fast numerical method for the Black-Scholes equation of American options*, SIAM J. Numer. Anal. **41** (2003), 2081–2095.

[6] E. G. Haug, *The Complete Guide to Option Pricing Formulas*, MaGraw-Hill, 2007.

[7] J. C. Hull, *Options, Futures and Others*, Prentice Hall, 2003.

[8] S. Ikonen and J. Toivanen, *Operator splitting methods for American option pricing*, Applied Mathematics Letters **17** (2004), 809–814.

[9] Y. K. Kwok, *Mathematical Models of Financial Derivatives*, Springer, 1998.

[10] R. C. Merton, *Theory of rational option pricing*, Bell J. Econ. Manag. Sci. **4** (1973), no. 1, 141–183.

[11] J. Persson and L. von Sydow, *Pricing European multi-asset options using a space-time adaptive FD-method*, Comput. Visual. Sci. **10** (2007), 173–183.

[12] C. Reisinger and G. Wittum, *On multigrid for anisotropic equations and variational inequalities*, Comput. Visual. Sci. **7** (2004), 189–197.

[13] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. **7** (1986), 856–869.

[14] Y. Saad and H. A. van der Vorst, *Iterative solution of linear systems in the 20th century*, J. Comput. Appl. Math. **123** (2000), 1–33.

[15] R. Seydel, *Tools for Computational Finance*, Springer Verlag, Berlin, 2003.

[16] D. Tavella and C. Randall, *Pricing Financial Instruments - The finite difference method*, John Wiley and Sons, Inc., 2000.

[17] J. Topper, *Financial Engineering with Finite Elements*, John Wiley and Sons, New York, 2005.

[18] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic press, 2001.

[19] H. A. 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.

[20] P. Wilmott, J. Dewynne, and S. Howison, *Option Pricing : mathematical models and computation*, Oxford Financia Press, Oxford, 1993.

Darae Jeong
Department of Mathematics
Korea University
Seoul 136-701, Korea
E-mail address: tinayoyo@korea.ac.kr

Junseok Kim
Department of Mathematics
Korea University
Seoul 136-701, Korea
E-mail address: cfdkim@korea.ac.kr

In-Suk Wee
Department of Mathematics
Korea University
Seoul 136-701, Korea
E-mail address: iswee@korea.ac.kr