J. KSIAM Vol.17, No.4, 295–306, 2013 http://dx.doi.org/10.12941/jksiam.2013.17.295

# AN ADAPTIVE MULTIGRID TECHNIQUE FOR OPTION PRICING UNDER THE BLACK–SCHOLES MODEL

DARAE JEONG[1], YIBAO LI[1], YONGHO CHOI[1], KYOUNG-SOOK MOON[2], AND JUNSEOK KIM[1]†

[1]DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SOUTH KOREA

[2]DEPARTMENT OF MATHEMATICS AND INFORMATION, GACHON UNIVERSITY, SOUTH KOREA
*E-mail address*: cfdkim@korea.ac.kr

ABSTRACT. In this paper, we consider the adaptive multigrid method for solving the Black–Scholes equation to improve the efficiency of the option pricing. Adaptive meshing is generally regarded as an indispensable tool because of reduction of the computational costs. The Black–Scholes equation is discretized using a Crank–Nicolson scheme on block-structured adaptively refined rectangular meshes. And the resulting discrete equations are solved by a fast solver such as a multigrid method. Numerical simulations are performed to confirm the efficiency of the adaptive multigrid technique. In particular, through the comparison of computational results on adaptively refined mesh and uniform mesh, we show that adaptively refined mesh solver is superior to a standard method.

## 1. INTRODUCTION

Financial option pricing model developed by Black and Scholes [1] in 1973 and extended by Merton [2]. Since then, methods for option pricing have been discovered and improved by many scholars. Details can be found in reference [3] which is a good review of valuation models and applications to the option pricing. However, because a closed-form solution cannot be obtained or the formulas for the exact solutions are too difficult to be practically usable, numerical solution has been a natural way to solve the problem in financial engineering [4].

To obtain an approximation of the option value, option pricing problems have been solved by the lattice method [5, 6, 7, 8], the finite difference method [7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18], the finite elements method [19, 20, 21, 22, 23, 24], and the finite volume method [25, 26]. In this paper, we propose an efficient and accurate method based on multigrid method and adaptive grid refinement method. Among the popular methods in recent years, multigrid methods [27, 28, 29, 30] are widely used for the numerical solution of partial differential equations (PDE). In reference [31], authors evaluated the option price by using multigrid method under Black–Scholes. Also, adaptive time-stepping has been proposed by some researchers

Received by the editors October 4 2013; Accepted November 5 2013.

2000 *Mathematics Subject Classification.* 65M06,35K20.

*Key words and phrases.* adaptive mesh refinement, multigrid method, Black–Scholes equations, Crank–Nicolson scheme, option pricing.

† Corresponding author.

[32], but few researchers use space-adaptive methods. Some examples, though, can be found in Achdou and Pironneau [33] and Pironneau and Hecht [34] who use a space-adaptive finite element method for discretization of the Black–Scholes PDE. In [35], an adaptive finite difference method is developed with full control of the local discretization error which is shown to be very efficient. In this paper, for the option pricing, we consider an adaptive mesh refinement (AMR) method. The AMR method [36] is very useful to combine the two goals of good accuracy and efficiency. In many science and engineering areas, such as fluid mechanics [37], electromagnetics [38], and materials science [39], an adaptive finite difference method has been very successful.

The purpose of our work is to propose an efficient adaptive FDM to solve the Black–Scholes PDEs. We computationally show that applying an adaptive method to this problem is very efficient compared to a standard FDM. We use the Crank–Nicolson method for the discretization. Other key components of the algorithm are the use of dynamic, block-structured Cartesian mesh refinement (see e.g., [40, 41]) and the use of an adaptive multigrid method [30] to solve the equations at an implicit time level. Locally refined block-structured Cartesian meshes are very natural to use together with multilevel multigrid methods. We note that other multilevel multigrid algorithms have been developed as part of the CHOMBO [42] software packages. Here, we follow the framework of a block-structured multilevel adaptive technique (MLAT) developed by Brandt [43].

The outline of this paper is the following. In Section 2 we describe the two-dimensional Black–Scholes PDE. In Section 3 we discretize the BS equation with finite difference method and we explain the adaptive mesh refinement method in Section 4. In Section 5, we show the numerical results by the proposed method. We draw conclusions in Section 6.

## 2. Black–Scholes model

In their award-winning work [1, 2], Black, Scholes, and Merton derived a parabolic second order PDE for the value $u(x, y, t)$ of an option on a stock. We use the original Black–Scholes model with two underlying assets to keep this presentation simple. Let $u(x, y, t)$ denote the value of the option at the underlying two assets $x$, $y$, and time $t$. The option value $u(x, y, t)$ is governed by the following two-asset Black–Scholes equation:

$$\begin{aligned}\frac{\partial u(x,y,t)}{\partial t} &+ \frac{1}{2}\sigma_1^2 x^2 \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{1}{2}\sigma_2^2 y^2 \frac{\partial^2 u(x,y,t)}{\partial y^2} + \rho\sigma_1\sigma_2 xy \frac{\partial^2 u(x,y,t)}{\partial x \partial y} \\ &+ rx\frac{\partial u(x,y,t)}{\partial x} + ry\frac{\partial u(x,y,t)}{\partial y} = ru(x,y,t), \qquad (2.1)\end{aligned}$$

where $\sigma_1$, $\sigma_2$ are the constant volatilities, $\rho$ is the correlation, and $r > 0$ is a constant riskless interest rate. The final condition is the payoff function $\Lambda(x, y)$ at expiry date $T$

$$u(x, y, T) = \Lambda(x, y). \qquad (2.2)$$

For instance, for European vanilla call option, the payoff at expiry is $\Lambda(x, y) = \max(x - K, y - K, 0)$ with a given strike price $K$.

## 3. Discretization with finite differences

A finite difference method approximates derivatives by difference operators and is a common numerical method. For an introduction to these methods we can recommend the books [10, 12, 13, 14, 15]. The original option pricing problems are defined in the unbounded domain $\{(x, y, t)|x \geq 0, y \geq 0, t \in [0, T]\}$. We need to truncate this domain into a finite computational domain $\Omega = \{(x, y, t)|0 \leq x \leq L, 0 \leq y \leq M, t \in [0, T]\}$, where $L$ and $M$ are large enough so that the error of the price $u$ arisen by the truncation is negligible [44]. Using $\tau = T - t$, we have

$$\frac{\partial u}{\partial \tau} = \frac{1}{2}(\sigma_1 x)^2 \frac{\partial^2 u}{\partial x^2} + \frac{1}{2}(\sigma_2 y)^2 \frac{\partial^2 u}{\partial y^2} + \sigma_1 \sigma_2 \rho x y \frac{\partial^2 u}{\partial x \partial y} + r x \frac{\partial u}{\partial x} + r y \frac{\partial u}{\partial y} - r u, \tag{3.1}$$

for $(x, y, \tau) \in \Omega \times (0, T]$ with an initial condition $u(x, y, 0) = \Lambda(x, y)$. Now, let us first discretize the given computational domain $\Omega$ 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, $N_x$, $N_y$, and $N_\tau$ are the number of space and time steps, respectively. Let us denote the numerical approximation of the solution by

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

where $i = 1, \ldots, N_x$, $j = 1, \ldots, N_y$, and $n = 1, \ldots, N_\tau$. By applying the Crank–Nicolson scheme to Eq. (3.1), which has an accuracy $\mathcal{O}(\Delta\tau^2 + h^2)$, we have

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta\tau} = \frac{1}{2}\left(\mathcal{L}u_{ij}^{n+1} + \mathcal{L}u_{ij}^n\right),$$

where the discrete difference operator $\mathcal{L}$ is defined by

$$\begin{aligned}\mathcal{L}u_{ij}^n = \; & \frac{(\sigma_1 x_i)^2}{2} \frac{u_{i-1,j}^n - 2u_{ij}^n + u_{i+1,j}^n}{h^2} + \frac{(\sigma_2 y_j)^2}{2} \frac{u_{i,j-1}^n - 2u_{ij}^n + u_{i,j+1}^n}{h^2} \\ & + \sigma_1 \sigma_2 \rho x_i y_j \frac{u_{i+1,j+1}^n + u_{i-1,j-1}^n - u_{i-1,j+1}^n - u_{i+1,j-1}^n}{4h^2} \\ & + r x_i \frac{u_{i+1,j}^n - u_{i-1,j}^n}{2h} + r y_j \frac{u_{i,j+1}^n - u_{i,j-1}^n}{2h} - r u_{ij}^n.\end{aligned} \tag{3.2}$$

We rewrite Eq. (3.2) by

$$N(u_{ij}^{n+1}) = \phi_{ij}^n, \tag{3.3}$$

where $N(u_{ij}^{n+1}) = u_{ij}^{n+1} - \frac{\Delta\tau}{2}\mathcal{L}u_{ij}^{n+1}$ and $\phi_{ij}^n = u_{ij}^n + \frac{\Delta\tau}{2}\mathcal{L}u_{ij}^n$.

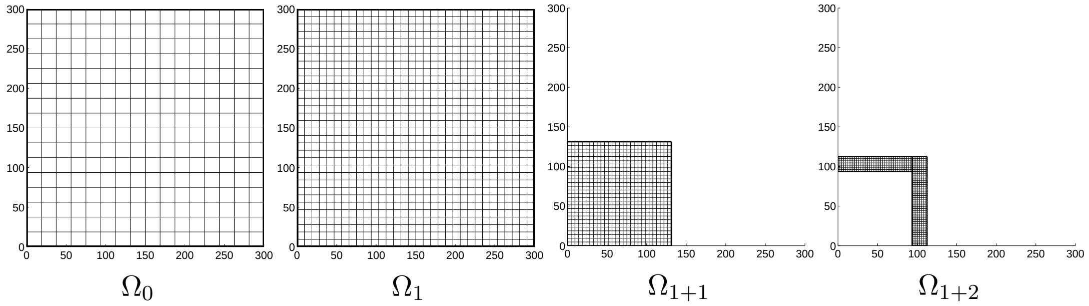
## 4. Numerical method

4.1. **Dynamic adaptive mesh refinement method.** In this section, we present an adaptive hierarchy of nested rectangular grids [37]. Both the initial creation of the grid hierarchy and the subsequent regriding operations in which the grids are dynamically changed to reflect changing solution conditions use the same procedure to create new grids [45]. Cells requiring additional refinement are identified and tagged using user-supplied criteria. The tagged cells are grouped into rectangular patches using the clustering algorithm given in Berger and Rigoutsos [46].

These rectangular patches are refined to form the grids at the next level. The process is repeated until a specified maximum level is reached. We consider a hierarchy of grids

$$\Omega_0, \ldots, \Omega_l, \Omega_{l+1}, \ldots, \Omega_{l+l^*},$$

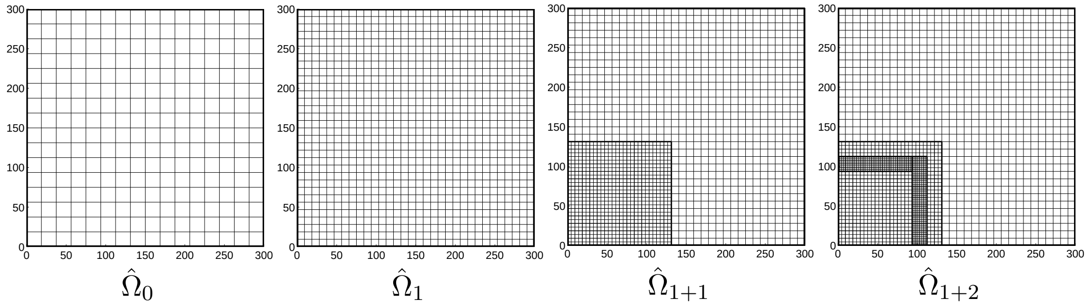
where $\Omega_0, \ldots, \Omega_l$ are global and $\Omega_{l+1}, \ldots, \Omega_{l+l^*}$ are local grids. A typical hierarchy of grids for the solution of the problem is shown in Fig. 1.



$\Omega_0$ $\Omega_1$ $\Omega_{1+1}$ $\Omega_{1+2}$

FIGURE 1. Hierarchy of grids when $l = 1$ and $l^* = 2$.

In this case, $\Omega_0$ and $\Omega_1$ are global grids $(l = 1)$ and the refined grids $\Omega_{l+1}$, $\Omega_{l+2}$ $(l^* = 2)$ cover increasingly smaller subdomains as indicated in Fig. 2. For instance, we can apply the refined local grids near the strike price since the values of options are not smooth near the strike price. We note that the grid refinement is automatically done by user-specified criteria. In addition to the global and the local grids, we consider their "composition". The corresponding sequence of *composite grids* (see Fig. 2) is defined by



$\hat{\Omega}_0$ $\hat{\Omega}_1$ $\hat{\Omega}_{1+1}$ $\hat{\Omega}_{1+2}$

FIGURE 2. Composite grids corresponding to the hierarchy of grids in Fig. 1 when $l = 1$ and $l^* = 2$.

$$\hat{\Omega}_k := \Omega_k \ (k = 0, \ldots, l) \text{ and } \hat{\Omega}_{l+k} := \Omega_l \cup \bigcup_{j=1}^{k} \Omega_{l+j} \ (k = 1, \ldots, l^*).$$

We use the original multi-level adaptive technique (MLAT) proposed by Brandt [47]. We now describe an adaptive multigrid cycle. Let us use the operator in Eq. (3.3) $N_k$ $(k = 0, 1, \dots, l, l+1, \dots, l+l^*)$ and the restriction and interpolation operators between $\Omega_k$ and $\Omega_{k-1}$, $I_k^{k-1}$, $I_{k-1}^k$ $(k = 1, 2, \dots, l, l+1, \dots, l+l^*)$ respectively. Let us assume the parameter $\gamma$ (the number of smoothing iterations), and starting on the finest grid $k = l + l^*$, the calculation of a new iterate $u_k^{m+1}$ from a given approximation $u_k^m$ proceeds: The details of overall steps are given in **Algorithm 1**.

**Algorithm 1** Adaptive cycle

$$u_k^{m+1} = adapcyc(k, u_k^m, u_{k-1}^m, N_k, \phi_k, \gamma)$$

1. **Presmoothing**
   - Compute $\bar{u}_k^m$ by applying $\gamma$ smoothing steps, Eq. (4.3), to $u_k^m$ on $\Omega_k$.
2. **Coarse-grid correction**
   - Compute

   $$\bar{u}_{k-1}^m = \begin{cases} I_k^{k-1}\bar{u}_k^m & \text{on } \Omega_{k-1} \cap \Omega_k \\ \bar{u}_{k-1}^m & \text{on } \Omega_{k-1} - \Omega_k \end{cases}$$

   - Compute the right-hand side

   $$\phi_{k-1}^n = \begin{cases} I_k^{k-1}(\phi_k^n - N_k(\bar{u}_k^m)) + N_{k-1}I_k^{k-1}\bar{u}_k^m & \text{on } \Omega_{k-1} \cap \Omega_k \\ \phi_{k-1}^n & \text{on } \Omega_{k-1} - \Omega_k \end{cases}$$

   - Compute an approximate solution $\hat{w}_{k-1}^m$ of the coarse grid equation on $\Omega_{k-1}$

   $$N_{k-1}(w_{k-1}^m) = \phi_{k-1}^n. \tag{4.1}$$

   If $k = 1$, employ smoothing steps.
   If $k > 1$, solve Eq. (4.1) using $\bar{u}_{k-1}^m$ as an initial approximation.

   $$\hat{w}_{k-1}^m = adapcyc(k-1, \bar{u}_{k-1}^m, u_{k-2}^m, N_{k-1}, \phi_{k-1}, \gamma).$$

   - Compute the correction $\hat{v}_{k-1}^m = \hat{w}_{k-1}^m - \bar{u}_{k-1}^m$, on $\Omega_{k-1} \cap \Omega_k$.
   - Set the solution $u_{k-1}^{m+1} = \hat{w}_{k-1}^m$, on $\Omega_{k-1} - \Omega_k$.
   - Interpolate the correction $\hat{v}_k^m = I_{k-1}^k \hat{v}_{k-1}^m$, on $\Omega_k$.
   - Compute the corrected approximation $u_k^{m,\ after\ CGC}=\bar{u}_k^m + \hat{v}_k^m$, on $\Omega_k$.
   - Carry out a quadratic interpolation at the ghost points.
3. **Postsmoothing**
   - Compute $u_k^{m+1}$ by applying $\gamma$ smoothing steps to $u_k^{m,\ after\ CGC}$ on $\Omega_k$.

Our implementation of this algorithm is constructed using the CHOMBO infrastructure [42], which has simplified the implementation of the locally adaptive algorithm. To perform the nonlinear multilevel AMR solver, we use and modify the CHOMBO AMR elliptic solver. This solver is based on a linear multigrid algorithm.

4.2. **Relaxation method in a multigrid cycle.** Now we derive a Gauss–Seidel relaxation operator. First, we rewrite Eq. (3.3) as

$$\begin{aligned} u_{ij}^{n+1} = & \left[\phi_{ij}^{n} + \frac{\Delta\tau}{2}\left(\frac{(\sigma_1 x_i)^2}{2}\frac{u_{i-1,j}^{n+1}+u_{i+1,j}^{n+1}}{h^2} + \frac{(\sigma_2 y_j)^2}{2}\frac{u_{i,j-1}^{n+1}+u_{i,j+1}^{n+1}}{h^2}\right.\right. \\ & +\sigma_1\sigma_2\rho x_i y_j \frac{u_{i+1,j+1}^{n+1}+u_{i-1,j-1}^{n+1}-u_{i-1,j+1}^{n+1}-u_{i+1,j-1}^{n+1}}{4h^2} \\ & \left.\left. + r x_i \frac{u_{i+1,j}^{n+1}-u_{i-1,j}^{n+1}}{2h} + r y_j \frac{u_{i,j+1}^{n+1}-u_{i,j-1}^{n+1}}{2h}\right)\right] / \\ & \left[1+\frac{\Delta\tau}{2}\left(\frac{(\sigma_1 x_i)^2+(\sigma_2 y_j)^2}{h^2}+r\right)\right]. \end{aligned} \tag{4.2}$$

Next, we replace $u_{kl}^{n+1}$ in Eq. (4.2) with $\bar{u}_{kl}^{m}$ if ($k < i$) or ($k = i$ and $l \le j$), otherwise with $u_{kl}^{m}$, i.e.,

$$\begin{aligned} \bar{u}_{ij}^{m} = & \left[\phi_{ij}^{n} + \frac{\Delta\tau}{2}\left(\frac{(\sigma_1 x_i)^2}{2}\frac{\bar{u}_{i-1,j}^{m}+u_{i+1,j}^{m}}{h^2} + \frac{(\sigma_2 y_j)^2}{2}\frac{\bar{u}_{i,j-1}^{m}+u_{i,j+1}^{m}}{h^2}\right.\right. \\ & +\sigma_1\sigma_2\rho x_i y_j \frac{u_{i+1,j+1}^{m}+\bar{u}_{i-1,j-1}^{m}-\bar{u}_{i-1,j+1}^{m}-u_{i+1,j-1}^{m}}{4h^2} \\ & \left.\left. + r x_i \frac{u_{i+1,j}^{m}-\bar{u}_{i-1,j}^{m}}{2h} + r y_j \frac{u_{i,j+1}^{m}-\bar{u}_{i,j-1}^{m}}{2h}\right)\right] / \\ & \left[1+\frac{\Delta\tau}{2}\left(\frac{(\sigma_1 x_i)^2+(\sigma_2 y_j)^2}{h^2}+r\right)\right]. \end{aligned} \tag{4.3}$$

Therefore, in a multigrid cycle, one smooth relaxation operator step consists of solving Eq. (4.3) given above.

## 5. Computational results

In this section, several numerical experiments are performed. To demonstrate its effectiveness, we compare the total computational cost, i.e., the CPU times with uniform mesh results on a test problem on the computational domain $\Omega = (0, 1200) \times (0, 1200)$ with $\Delta\tau = 1/1024$. The calculations have been performed on an IBM personal computer with 3.0GHz speed of 3.48GB RAM.

5.1. **European vanilla call option.** As the benchmark problem, we consider the European vanilla option problem. This problem is of great interest to academicians in the finance literature and often used to show the accuracy of a given numerical scheme [7, 48, 49]. The initial state is $u(x, y, 0) = \max[\max(x, y) - 100, 0]$. For the parameters, we take $\sigma_1 = \sigma_2 = 0.5$, $\rho = 0.5$, and $r = 0.03$. We perform an adaptive mesh refinement every 5 time steps. The refinement is based on the range of values of $u$, i.e., we refine the grids if $0.3 < u < 10$. We compute this with a base $64^2$ mesh with $3,\ 4$, and $5$ levels of refinements. To estimate the cost of the equivalent uniform-grid solution, we compute $1024$ time steps on the equivalent $512^2$, $1024^2$, and $2048^2$ meshes. In Fig. 3, (a) and (b) show the initial profile and the final configuration at time $\tau = 1$ on the adaptive mesh, respectively. We can observe fine meshes around the region of our interests which are neighborhood of strike prices. And Fig. 3(c) and (d) show magnified representations of (a) and (b), respectively.

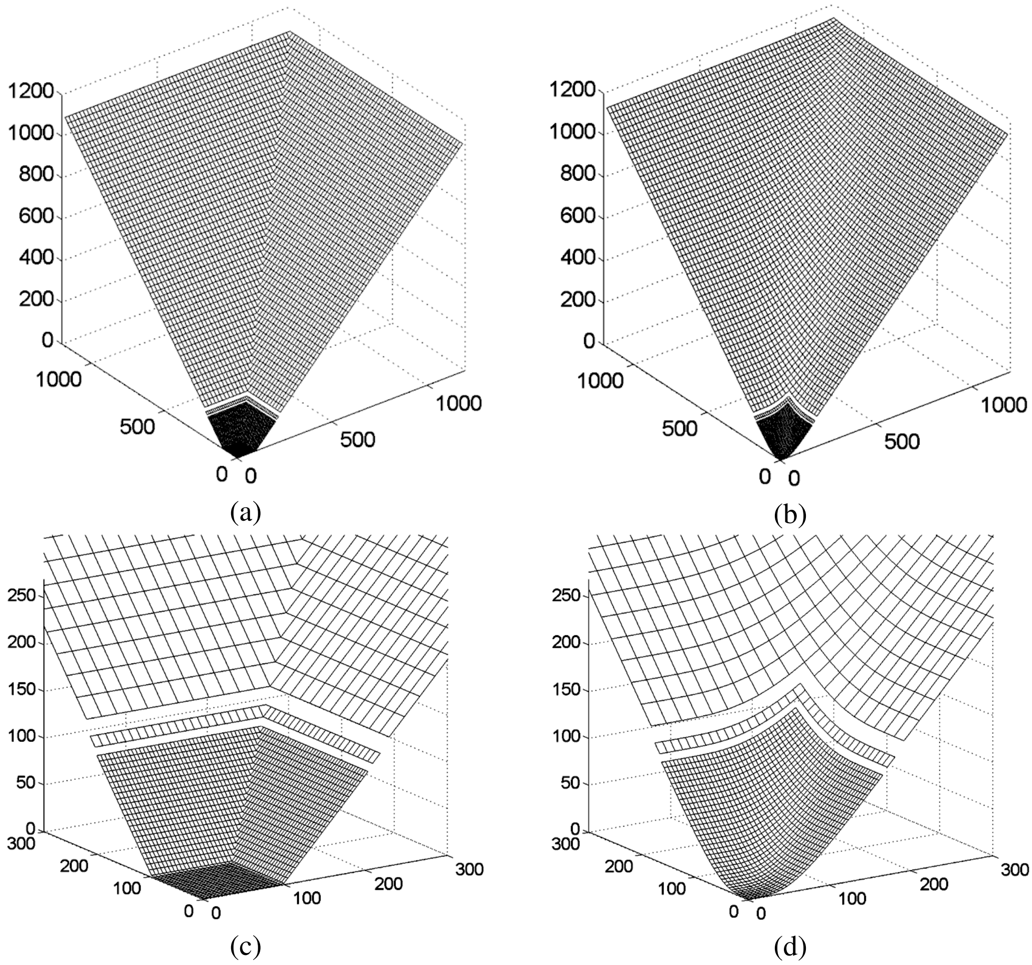


FIGURE 3. European vanilla call option: (a) The initial configuration at time $\tau = 0$. (b) The final configuration at time $\tau = 1$. (c) and (d): Magnified representations of (a) and (b), respectively.

Next, we compare the CPU times with AMR and uniform mesh results. The computational results are shown in Table 1 and it is clear that AMR is more efficient than the uniform mesh method. We scale CPU time with the AMR method. Here, 1 in CPU time of AMR stands for the calculation time for AMR method.

TABLE 1. CPU time comparison between uniform mesh and AMR of European vanilla call option.

| Case | Uniform mesh $512^2$ | AMR with base mesh size, $64^2$<br>3 levels, effective mesh size $512^2$ |
|---|---|---|
| CPU time | 68.3 | 1 |
| Case | Uniform mesh $1024^2$ | AMR with base mesh size, $64^2$<br>4 levels, effective mesh size $1024^2$ |
| CPU time | 169.7 | 1 |
| Case | Uniform mesh $2048^2$ | AMR with base mesh size, $64^2$<br>5 levels, effective mesh size $2048^2$ |
| CPU time | 285.3 | 1 |

5.2. **Cash-or-Nothing option.** Next, we perform the comparison with a cash-or-nothing option. The initial state is

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

Here, we simply set $Cash = 1$ and $K = 100$. And the other parameters and computational conditions are chosen as the same in the numerical experiment of European call option. Figure 4(a) and (b) show the initial profile and the final configuration at time $\tau = 1$ on the adaptive mesh, respectively. And Fig. 4(c) and (d) show magnified representations of (a) and (b), respectively. Next, the CPU times with AMR and uniform mesh results are presented in Table 2. As shown in Table 2, it is clear that AMR is more efficient than the uniform mesh method.

TABLE 2. CPU time comparison between uniform mesh and AMR of Cash-or-Nothing option.

| Case | Uniform mesh $512^2$ | AMR with base mesh size, $64^2$<br>3 levels, effective mesh size $512^2$ |
|---|---|---|
| CPU time | 80.2 | 1 |
| Case | Uniform mesh $1024^2$ | AMR with base mesh size, $64^2$<br>4 levels, effective mesh size $1024^2$ |
| CPU time | 167.2 | 1 |
| Case | Uniform mesh $2048^2$ | AMR with base mesh size, $64^2$<br>5 levels, effective mesh size $2048^2$ |
| CPU time | 180.5 | 1 |

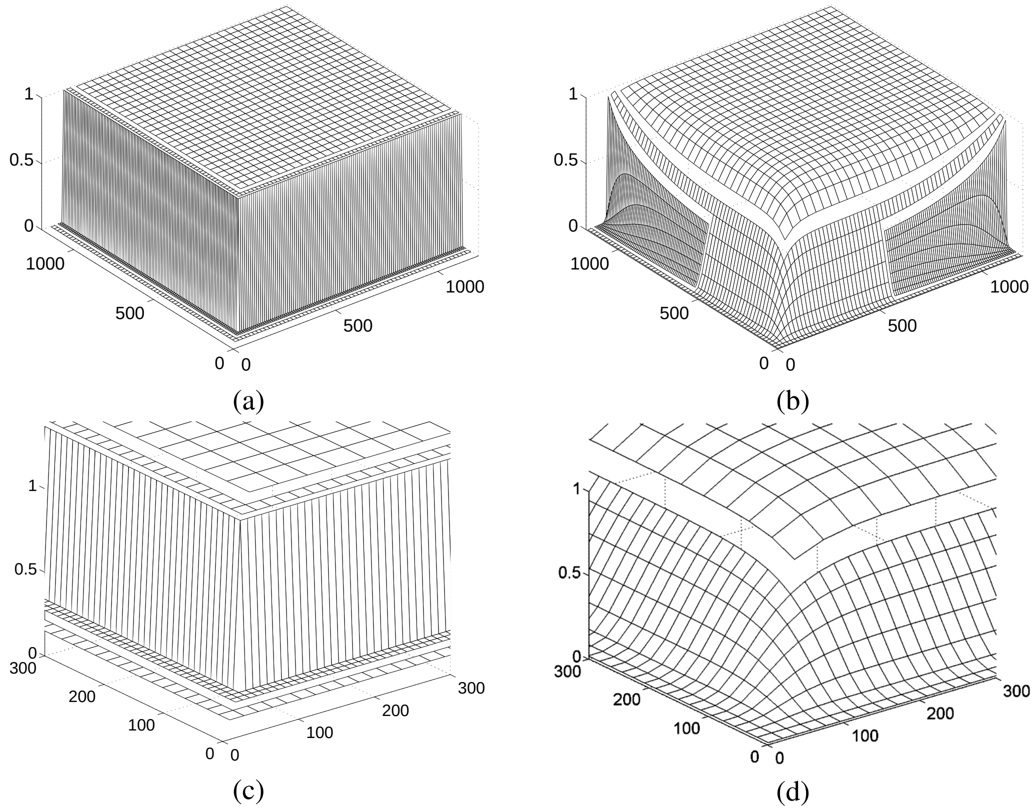


FIGURE 4. Cash-or-Nothing option: (a) The initial configuration at time $\tau = 0$. (b) The final configuration at time $\tau = 1$. (c) and (d): Magnified representations of (a) and (b), respectively.

## 6. CONCLUSIONS

In this paper, we focused on two major aspects that we encounter when applying numerical methods to option pricing problems such as grid resolutions and domain sizes. We proposed an adaptive mesh refinement method to solve the Black–Scholes equation. We computationally showed that the proposed adaptive scheme gave much better efficiency than the standard FDM. In particular, we showed that the use of local refinement resulted in significant savings in computational time and memory when compared to the equivalent uniform-mesh solution. Studies of these methods in higher dimensions will be the subject of future research.

## 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 [50].

## REFERENCES

[1] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, Journal of Political Economy, **81**, (1973), 637–654.

[2] R. C. Merton, *Theory of rational option pricing*, The Bell Journal of Economics and Management Science, **4**, (1973), 141–183.

[3] M. Broadie and J. B. Detemple, *Option pricing: valuation models and applications*, Management Sciences, **50**, (2004), 1145–1177.

[4] H. Han and X. Wu, *A Fast numerical method for the Black–Scholes equation of American options*, SIAM Journal on Numerical Analysis, **41**, (2003), 2081–2095.

[5] G. Cortazar, *Simulation and Numerical Methods in Real Options Valuation: Real Options and Investment Under Uncertainty*, The MIT Press, Cambridge, 601–620, 2004.

[6] J. C. Cox, S. A Ross, and M. Rubinstein, *Option pricing: a simplified approach*, Journal of Financial and Economics, **7**, (1979), 229–263.

[7] R. Geske and K. Shastri, *Valuation by approximation: a comparison of alternative option valuation techniques*, Journal of Financial and Quantitative Analysis, **20**, (1985), 45–71.

[8] K. Moon and H. Kim, *A cost-effective modification of the trinomial method for option pricing*, Journal of Korean Society for Industrial and Applied Mathematics, **15**, (2011), 1–17.

[9] M. J. Brennan and E. S. Schwartz, *Finite difference methods and jump processes arising in the pricing of contingent claims: asynthesis*, Journal of Financial and Quantitative Analysis, **13**, (1978), 461–474.

[10] D. Tavella and C. Randall, *Pricing Financial Instruments: The Finite Difference Method*, John Willey & Sons, New York, 2000.

[11] M. M. Chawla and D. J. Evans, *Numerical volatility in option valuation from Black–Scholes equation by finite differences*, International Journal of Computer Mathematics, **81**, (2004), 1039–1041.

[12] P. Wilmott, J. Dewynne, and S. Howison, *Option Pricing : Mathematical Models and Computation*, Oxford Financia Press, Oxford, 1993.

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

[14] R. Seydel, *Tools for Computational Finance*, Springer, Berlin, 2012.

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

[16] B. A. Wade, A. Q. M. Khaliq, M. Yousuf, J. Vigo-Aguiar, and R. Deininger, *On smoothing of the Crank-Nicolson scheme and higher order schemes for pricing barrier options*, Journal of Computational and Applied Mathematics, **204**, (2007), 144–158.

[17] A. Q. M. Khaliq, D. A. Voss, and K. Kazmi, *Adaptive $\theta$-methods for pricing American options*, Journal of Computational and Applied Mathematics, **222**, (2008), 210–227.

[18] D. Jeong, I. S. Wee, and J. Kim, *An operator splitting method for pricing the ELS option*, Journal of Korean Society for Industrial and Applied Mathematics, **14**, (2010), 175–187.

[19] Y. Achdou and N. Tchou, *Variational analysis for the Black and Scholes equation with stochastic volatility*, Mathematical Models and Numerical Analysis, **36**, (2002), 373–395.

[20] A. Ern, S. Villeneuve, and A. Zanette, *Adaptive finite element methods for local volatility European option pricing*, International Journal of Theoretical and Applied Finance, **7**, (2004), 659–684.

[21] C. Zhang, *Pricing American Options by Adaptive Finite Element Method*, Mathematics Department University of Maryland, 2005.

[22] Z. Zhu and N. Stokes, *A Finite Element Platform for Pricing Path-dependent Exotic Options*, CSIRO Mathematical & Information Sciences, Australia, 1998.

[23] J. Topper, *Option Pricing with Finite Elements*, Wilmott Magazine, 84–90, 2005.

[24] R. Zvan, P. A. Forsyth, and K. R. Vetzal, *A General Finite Element Approach for PDE Option Pricing Models*, University of Waterloo, Canada, 1998.

[25] P. A. Forsyth and K. R. Vetzal, *Quadratic convergence for valuing American options using a penalty method*, SIAM Journal on Scientific Computing, **23**, (2002), 2095–2122.

[26] R. Zvan, P. A. Forsyth, and K. R. Vetzal, *Penalty methods for American options with stochastic volatility*, Journal of Computational and Applied Mathematics, **91**, (1998), 199–218.

[27] Z. Zhou and H. Wu, *Finite element multigrid method for the boundary value problem of fractional advection dispersion equation*, Journal of Applied Mathematics, **2013**, (2013), 1–8.

[28] W. Hackbusch, *Multi-grid Methods and Applications*, Springer-Verlag, New York, 1985.

[29] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley and Sons, Chichester, 1995.

[30] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic press, London, 2000.

[31] D. Jeong, J. Kim, and I. S. Wee, *An accurate and efficient numerical method for Black–Scholes equations*, Communications of the Korean Mathematical Society, **24**, (2009), 617–628.

[32] R. Zvan, K. R. Vetzal, and P. A. Forsyth, *PDE methods for pricing barrier options*, Journal of Economic Dynamics & Control, **24**, (2000), 1563–1590.

[33] Y. Achdou and O. Pironneau, *Computational Methods for Option Pricing*, SIAM, Philadelphia, 2005.

[34] O. Pironneau and F. Hecht, *Mesh adaption for the Black and Scholes equations*, East-West Journal of Numerical Mathematics, bf 8, (2000), 25–36.

[35] J. Persson and L. von Persson, *Pricing European multi-asset options using a space-time adaptive FD-method*, Computing and Visualization in Science, **10**, (2007), 173–183.

[36] H. Ji, F. S. Lien, and E. Yee, *Parallel adaptive mesh refinement combined with additive multigrid for the efficient solution of the poisson equation*, ISRN Applied Mathematics, **2012**, (2012), 1–24.

[37] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, *A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations*, Journal of Computational Physics, **142**, (1998), 1–46.

[38] C. J. Garcia-Cervera and A. M. Roma, *Adaptive mesh refinement for micromagnetics simulations*, IEEE Transactions on Magnetics, **42**, (2006), 1648–1654.

[39] S. Wise, J. Kim, and J. Lowengrub, *Solving the regularized, strongly anisotropic Cahn–Hilliard equation by an adaptive nonlinear multigrid method*, Journal of Computational Physics, **226**, (2007), 414–446.

[40] M. J. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, Journal of Computational Physics, **82**, (1989), 64–84.

[41] M. J. Berger and J. Oliger, *Adaptive mesh refinement for hyperbolic partial differential equations*, Journal of Computational Physics, **53**, (1984), 484–512.

[42] Applied numerical algorithms group. *The chombo framework for block-structured adaptive mesh refinement*, Technical report, Lawrence Berkeley National Laboratory, 2005 (Available online at http://seesar.lbl.gov/ANAG/chombo).

[43] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of Computation, **31**, (1977), 333–390.

[44] R. Kangro and R. Nicolaides, *Far field boundary conditions for Black–Scholes equations*, SIAM Journal on Numerical Analysis, **38** (2000), 1357–1368.

[45] C. A. Rendleman, V. E. Beckner, M. Lijewski, W. Crutchfield, and J. B. Bell, *Parallelization of structured, hierarchical adaptive mesh refinement algorithms*, Computing and Visualization in Science, **3**, (2000), 147–157.

[46] M. Berger and I. Rigoutsos, *An algorithm for point clustering and grid generation*, Systems, Man and Cybernetics, IEEE Transactions on, **21**, (1991), 1278–1286.

[47] D. Bai and A. Brandt, *Local mesh refinement multilevel techniques*, SIAM Journal on Scientific and Statistical Computing, **8**, (1987), 109–134.

[48] G. W. Buetow and J. S. Sochacki, *The trade-offs between alternative finite difference techniques used to price derivative securities*, Applied Mathematics and Computation, **115**, (2000), 177–190.

[49] S. Figlewski and B. Gao, *The adaptive mesh model: a new approach to efficient option pricing*, Journal of Financial Economics, **53**, (1999), 313–351.

[50] D. Jeong, *Mathematical model and numerical simulation in computational finance*, Ph.D. Thesis, Department of Mathematics, Korea University, Korea, December, 2012.