

AN ADAPTIVE FINITE DIFFERENCE METHOD USING FAR-FIELD BOUNDARY CONDITIONS FOR THE BLACK-SCHOLES EQUATION

DARAE JEONG, TAEYOUNG HA, MYOUNGNYOUN KIM, JAEMIN SHIN,
IN-HAN YOON, AND JUNSEOK KIM

ABSTRACT. We present an accurate and efficient numerical method for solving the Black–Scholes equation. The method uses an adaptive grid technique which is based on a far-field boundary position and the Peclet condition. We present the algorithm for the automatic adaptive grid generation: First, we determine a priori suitable far-field boundary location using the mathematical model parameters. Second, generate the uniform fine grid around the non-smooth point of the payoff and a non-uniform grid in the remaining regions. Numerical tests are presented to demonstrate the accuracy and efficiency of the proposed method. The results show that the computational time is reduced substantially with the accuracy being maintained.

1. Introduction

In this paper, we develop an accurate and efficient numerical method to solve the Black–Scholes (BS) equation:

$$(1) \quad \frac{\partial u}{\partial t} = -\frac{1}{2}(\sigma x)^2 \frac{\partial^2 u}{\partial x^2} - rx \frac{\partial u}{\partial x} + ru \quad \text{for } (x, t) \in (0, \infty) \times [0, T),$$

where $u(x, t)$ is the value of the derivative security, x is the value of the underlying security, t is the time, T is the expiry date, r is the risk-free interest rate and σ is the volatility of the underlying asset [2, 16]. By changing the variable

Received August 26, 2013.

2010 *Mathematics Subject Classification.* Primary 65M06, 65M50.

Key words and phrases. Black–Scholes equation, finite difference method, far-field boundary conditions, adaptive grid, Peclet condition.

The first author (D. Jeong) was supported by a Korea University Grant. The last author (J.S. Kim) acknowledges the support of the National Institute for Mathematical Sciences (NIMS) grant funded by the Korea government (No. A21301). This work is based on the first authors Ph. D. thesis [10] and an earlier version of this paper was presented at the KSIAM 2011 Spring Conference.

with $\tau = T - t$, we can rewrite Eq.(1) as follows:

$$(2) \quad \frac{\partial u}{\partial \tau} = \frac{1}{2}(\sigma x)^2 \frac{\partial^2 u}{\partial x^2} + rx \frac{\partial u}{\partial x} - ru \quad \text{for } (x, \tau) \in (0, \infty) \times (0, T].$$

The initial condition is given by

$$(3) \quad u(x, 0) = p(x).$$

The original continuous BS problem is posed on an infinite spatial domain $(0, \infty)$. However, we need a finite domain and boundary condition for solving the problem numerically.

An adaptive grid generation technique has been used to reduce the number of grid-points maintaining the discretization error at a prescribed level. Many people have been studied the adaptive algorithm in space and time. In [17, 18], the authors used an adaptive technique proposed in [13] based on local discretization error. Moreover, the computational grid is refined in blocks and the grid and time step change at every discrete time point in [14]. The authors in [6] developed space-time adaptive and high-order methods for valuing American options using PDE approach. In [15], the grid and time step sizes were chosen dynamically to satisfy a bound on the global error at the expiry date. In addition, an adaptive finite element discretization was developed in [1, 19] for American options. Also, the authors in [25] applied a discrete singular convolution algorithm with an adaptive mesh.

The purpose of this work is to present an adaptive grid distribution depending on a far-field boundary position and the Peclet condition to solve the Black–Scholes equation accurately and efficiently. The proposed algorithm is as follows: First, we determine a priori suitable far-field location using estimates with a given error tolerance. Next, we distribute a uniform fine grid around the strike point and a non-uniform grid in the remaining regions, using the Peclet condition.

This paper is organized as follows. In Section 2, we discretize the BS equation on a non-uniformly spaced grid. In addition, we present a detailed discussion of our proposed adaptive grid generating method, which depends on a far-field boundary condition and the Peclet condition of the BS equation. Section 3 provides two numerical results for uniform and adaptive cases. Section 4 concludes with a short summary.

2. Numerical solution

2.1. Discretization with finite differences

The finite difference method has been applied to pricing financial contracts for many years [21]. For more details about finite difference methods in computational finance, we refer the reader to the books [7, 22, 23] and papers [3, 4, 12, 20].

The BS equation is discretized on a grid defined by $x_0 = 0$ and $x_{i+1} = x_i + h_i$ for $i = 0, \dots, N_x - 1$, where N_x is the number of grid intervals and h_i is the grid spacing, see Fig. 1. And we assume that $x_{N_x} = S_{\max}$ and $h_{N_x} = h_{N_x-1}$.

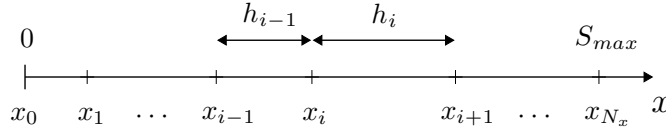


FIGURE 1. A non-uniform grid with space step sizes h_i .

Let us denote the numerical approximation of the solution by

$$u_i^n \approx u(x_i, n\Delta\tau),$$

where $\Delta\tau = T/N_\tau$ is the time step size and N_τ is the total number of time steps. By applying an implicit scheme to Eq.(2), we have

$$(4) \quad \frac{u_i^{n+1} - u_i^n}{\Delta\tau} = \frac{\sigma^2 x_i^2}{2} \left(\frac{\partial^2 u}{\partial x^2} \right)_i^{n+1} + r x_i \left(\frac{\partial u}{\partial x} \right)_i^{n+1} - r u_i^{n+1},$$

where the first and second derivatives are defined as

$$\begin{aligned} \left(\frac{\partial u}{\partial x} \right)_i^{n+1} &= -\frac{h_i}{h_{i-1}(h_{i-1} + h_i)} u_{i-1}^{n+1} + \frac{h_i - h_{i-1}}{h_{i-1} h_i} u_i^{n+1} + \frac{h_{i-1}}{h_i(h_{i-1} + h_i)} u_{i+1}^{n+1}, \\ \left(\frac{\partial^2 u}{\partial x^2} \right)_i^{n+1} &= \frac{2}{h_{i-1}(h_{i-1} + h_i)} u_{i-1}^{n+1} - \frac{2}{h_{i-1} h_i} u_i^{n+1} + \frac{2}{h_i(h_{i-1} + h_i)} u_{i+1}^{n+1}. \end{aligned}$$

Then we can rewrite Eq.(4) as

$$(5) \quad \alpha_i u_{i-1}^{n+1} + \beta_i u_i^{n+1} + \gamma_i u_{i+1}^{n+1} = f_i^n,$$

where

$$\begin{aligned} \alpha_i &= \frac{-\sigma^2 x_i^2 + r x_i h_i}{h_{i-1}(h_{i-1} + h_i)}, & \beta_i &= \frac{\sigma^2 x_i^2 - r x_i (h_i - h_{i-1})}{h_{i-1} h_i} + r + \frac{1}{\Delta\tau}, \\ \gamma_i &= \frac{-\sigma^2 x_i^2 - r x_i h_{i-1}}{h_i(h_{i-1} + h_i)}, & f_i^n &= \frac{u_i^n}{\Delta\tau}. \end{aligned}$$

We impose the zero Dirichlet boundary condition at $x = 0$ and the linear boundary condition [17, 21, 24] at $x = S_{\max}$, which is defined by

$$(6) \quad \frac{\partial^2 u}{\partial x^2}(S_{\max}, \tau) = 0 \quad \text{for } \tau \in [0, T].$$

Then Eq.(6) is discretized by $(u_{N_x-1}^{n+1} - 2u_{N_x}^{n+1} + u_{N_x+1}^{n+1})/h_{N_x-1}^2 = 0$. By substituting the relation $u_{N_x+1}^{n+1} = 2u_{N_x}^{n+1} - u_{N_x-1}^{n+1}$ into Eq.(5), we get

$$(7) \quad (\alpha_{N_x} - \gamma_{N_x}) u_{N_x-1}^{n+1} + (\beta_{N_x} + 2\gamma_{N_x}) u_{N_x}^{n+1} = f_{N_x}^n.$$

The matrix form of the linear system (5) and (7) can be rewritten as

$$\begin{pmatrix} \beta_1 & \gamma_1 & 0 & \dots & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \alpha_{N_x-1} & \beta_{N_x-1} & \gamma_{N_x-1} \\ 0 & \dots & 0 & \alpha_{N_x} - \gamma_{N_x} & \beta_{N_x} + 2\gamma_{N_x} \end{pmatrix} \begin{pmatrix} u_1^{n+1} \\ u_2^{n+1} \\ \vdots \\ u_{N_x-1}^{n+1} \\ u_{N_x}^{n+1} \end{pmatrix} = \begin{pmatrix} f_1^n \\ f_2^n \\ \vdots \\ f_{N_x-1}^n \\ f_{N_x}^n \end{pmatrix}.$$

We solve the linear system by using the Thomas algorithm, which inverts the tri-diagonal matrix directly.

2.2. Adaptive grid generation technique

In this section, we propose an improved adaptive grid technique which is based on a far-field boundary position of the BS equation and the Peclet condition for the stability of solutions. If the computational domain size and grid points are large enough, then we can get an accurate result, but we need more computational resources. For a given number of grid points, we want to improve the computational accuracy and efficiency by using adaptive grid techniques. First, we decide the computational domain size, S_{\max} , to be as small as possible while keeping the resulting solution within an error tolerance. Second, we adaptively distribute grid points so that we get an accurate numerical solution.

2.2.1. Choice of far-field boundary position. Let $u(x, \tau)$ and $w(x, \tau)$ be solutions of Eq.(2) on an infinite domain $(0, \infty) \times (0, T]$ and a finite domain $(0, S_{\max}) \times (0, T]$, respectively. For the boundary condition at $x = S_{\max}$, we set $w(S_{\max}, \tau) = p(S_{\max})$ for all $\tau \in (0, T)$.

First, we consider a European call option, whose payoff function is given as $p(x) = \max(x - K, 0)$. Since $-K + x \leq p(x) \leq x$ and Eqs.(11) and (12) in Appendix A, we can say that $-Ke^{-r\tau} + x \leq u(x, \tau) \leq x$. Therefore, we have $\sup_{\tau \in (0, T)} |u(S_{\max}, \tau) - w(S_{\max}, \tau)| \leq K$. Then, by Eq.(14) in Appendix A,

$$|u(x, \tau) - w(x, \tau)| \leq Ke^{-\frac{\ln \frac{S_{\max}}{K} (\ln \frac{S_{\max}}{K} + \min\{0, \sigma^2 - 2r\} \tau)}{2\sigma^2 \tau}}$$

for $x \in [0, K]$. Therefore, if we want the error on the finite domain to be less than K/A , then S_{\max} should satisfy the following inequality:

$$Ke^{-\frac{\ln \frac{S_{\max}}{K} (\ln \frac{S_{\max}}{K} + \min\{0, \sigma^2 - 2r\} \tau)}{2\sigma^2 \tau}} \leq \frac{K}{A}.$$

This estimation tells us that if

$$(8) \quad S_{\max} \geq Ke^{-\frac{1}{2} \min\{0, (\sigma^2 - 2r)\tau\} + \frac{1}{2} \sqrt{(\min\{0, (\sigma^2 - 2r)\tau\})^2 + 8\sigma^2 \tau \ln A}},$$

then we can be sure that $w(x, \tau)$, the solution of the truncated domain problem, gives us a call option value that is within K/A from the correct value [11].

Next, let us consider the other type of European call options, a cash-or-nothing option, whose payoff function is $p(x) = \text{Cash}$ if $x > K$ and $p(x) = 0$

otherwise. Similar to the first case, since $\sup_{\tau \in (0, T)} |u(S_{\max}, \tau) - w(S_{\max}, \tau)| \leq \text{Cash}$, for all $x \in (0, S_{\max})$, $u(x, \tau)$ and $w(x, \tau)$ satisfy the following inequality:

$$|u(x, \tau) - w(x, \tau)| \leq \text{Cash} e^{-\frac{\ln \frac{S_{\max}}{K} (\ln \frac{S_{\max}}{K} + \min\{0, \sigma^2 - 2r\} \tau)}{2\sigma^2 \tau}}.$$

Therefore, if we want the error on the finite domain to be less than K/A , then we need

$$(9) \quad S_{\max} \geq K e^{-\frac{1}{2} \min\{0, (\sigma^2 - 2r)\tau\} + \frac{1}{2} \sqrt{\min\{0, (\sigma^2 - 2r)\tau\}^2 + 8\sigma^2 \tau \ln(A \text{ Cash}/K)}}.$$

These estimations are essential when performing numerical approximations of infinite domain problems since we must use a finite domain in finite difference schemes. Therefore, for the accuracy and efficiency of our proposed method, we utilize this error estimation to choose the far-field boundary position.

2.2.2. Non-uniform grid generation process with the Peclet condition. The adaptive grid generation process aims to create a grid with a uniform fine grid around the strike price K and an increasingly large grid size as we move toward the far-field boundary. To do this, we propose a grid generating function $h(x)$ based on the Peclet condition

$$h(x) = \begin{cases} p(x - K - (m - 0.5)\bar{h})^d + \bar{h} & \text{if } x \geq K + (m - 0.5)\bar{h}, \\ p(x - K + (m - 0.5)\bar{h})^d + \bar{h} & \text{if } x \leq K - (m - 0.5)\bar{h}, \end{cases}$$

where p , d and \bar{h} are real positive numbers and m is a natural number. First, we allocate $2m$ grid points around the strike price K with a grid size of \bar{h} , see Fig. 2. Then, we start at $x_i = K + (m - 0.5)\bar{h}$ and define $x_{i+1} = x_i + h(x_i)$. We continue this procedure until we reach the point where $x_{N_x-1} \leq S_{\max} < x_{N_x}$, at this stage we reset $h_{N_x-1} = h_{N_x-2}$. Similarly, for the grid generation of left side, we start at $x_i = K - (m - 0.5)\bar{h}$ and define $x_{i-1} = x_i - h(x_i)$. We continue this process until we reach the point where $x_0 \leq 0 < x_1$. If $x_0 < 0$, then we redefine $x_0 = 0$. This procedure is described schematically in Fig. 2. Note that we also show an illustration of initial and later solutions on the adaptive grid.

Now, for numerical solutions which are free of spurious oscillations, the space step size must satisfy the following Peclet condition [26]:

$$(10) \quad h_i < \frac{\sigma^2}{r} x_i.$$

The derivation of this condition (10) is given in Appendix B. In this paper, we will choose a piecewise linear grid function $h(x)$ whose slope is less than σ^2/r to obtain non-oscillatory solutions. We will use the parameter $p = 0.05\sigma^2/r$ and $d = 1$ for numerical examples.

3. Computational results

In this section, we perform numerical experiments to test the accuracy and efficiency of the proposed method. The main focus of these tests is on the performance of the proposed adaptive grid technique compared to the standard uniform grid method. As the benchmark problems, we consider the European

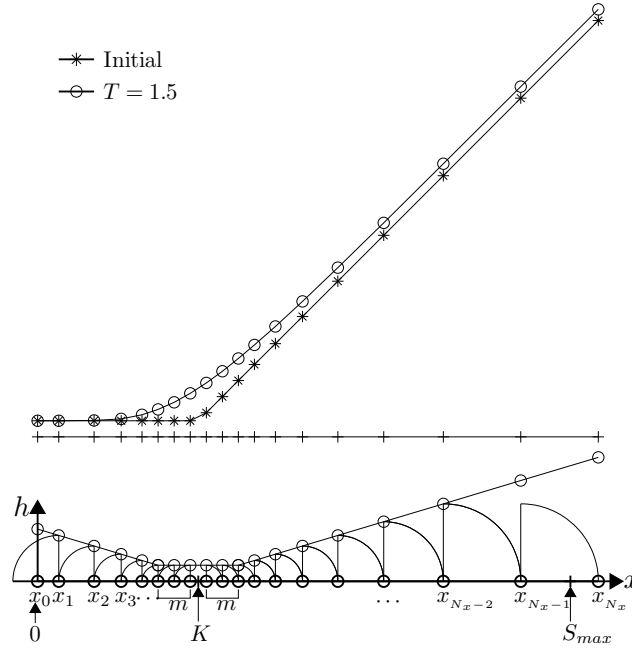


FIGURE 2. Construction of the adaptive grid using the function $h(x)$.

option problems for numerical examples. These problems are of great interest to academicians in the finance literature and often used to show the accuracy of a given numerical scheme [5, 8, 9].

3.1. Far-field position

We study the effect of the relation between the domain size L and expiry date T using the European vanilla call option. The computational domain is $\Omega = (0, L)$ with a uniform grid. The parameters $\sigma = 0.35$, $r = 0.05$ and $h = 1$ are used. For each case, we ran the calculation up to time T with a time step of $\Delta\tau = 0.01$. The initial condition is $u(x, 0) = \max(x - K, 0)$ with the strike price $K = 100$.

For this European call option, the closed form solution of the Black–Scholes equation is

$$u(x, \tau) = xN(d_1) - Ke^{-r\tau}N(d_2), \quad \forall x \in [0, L], \quad \forall \tau \in [0, T]$$

$$d_1 = \frac{\ln(x/K) + (r + \frac{1}{2}\sigma^2)\tau}{\sigma\sqrt{\tau}}, \quad d_2 = d_1 - \sigma\sqrt{\tau},$$

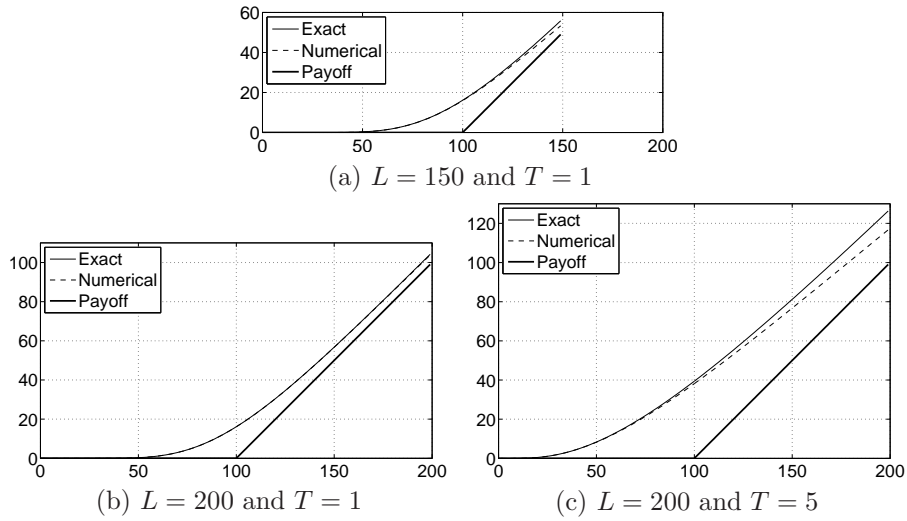


FIGURE 3. Initial profiles, numerical results and exact solutions with different domain sizes and times.

where

$$N(d) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^d \exp\left(-\frac{x^2}{2}\right) dx$$

is the cumulative distribution function for the standard normal distribution [2].

Figures 3(a), (b) and (c) show the initial profile, numerical and exact solutions. When $L = 150$ and $T = 1$, we can observe that there is a large difference between the numerical and exact solutions (see Fig. 3(a)). With increased domain size $L = 200$, we get a good result in Fig. 3(b). However, when we increase the time to $T = 5$, Fig. 3(c) shows a large difference between the numerical and exact solutions. This result implies that we need a large enough domain size in relation to the size of T .

3.2. Comparison of uniform and adaptive grid methods

To demonstrate the performance of the proposed adaptive grid technique with a far-field boundary and Peclet conditions, we compare the numerical results of both the uniform and adaptive grid methods. For the comparison study, we set $\sigma = 0.35$, $r = 0.05$, $\Delta\tau = 0.001$, $p = 0.05\sigma^2/r$ and $d = 1$. Also, the far-field boundary position S_{\max} is set to achieve an accuracy of $K/A = 0.1$ according to the conditions (8) and (9) for each case. To compare numerical results on uniform and adaptive grids, we use the relative root mean square

error (RMSE) on $[0.95K, 1.05K]$. Here, the relative RMSE is defined as

$$\text{Relative RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{u_i - u(x_i)}{u(x_i)} \right)^2},$$

where N is the total number of points on $[0.95K, 1.05K]$, u_i and $u(x_i)$ are numerical and exact solutions, respectively.

3.2.1. Call option on a maximum of one asset. The initial state is given by $u(x, 0) = \max(x - K, 0)$ where $K = 100$. Table 1 shows computational results such as RMSE in $[0.95K, 1.05K]$, relative CPU time and grid points N_x at time $T = 1$ with adaptive and uniform grids. In this numerical test, we use different space steps: $\bar{h} = 2/2^m$, where $m = 0, 1, 2, \dots, 9$. As shown in Table 1, the CPU times taken from the uniform grids are larger than those of the adaptive grid method. Also, the total number of grid points N_x on the adaptive grid is much smaller than on the uniform grid. Overall, the adaptive grid outperforms the uniform grid.

TABLE 1. Comparison of relative CPU time and grid points N_x on adaptive and uniform grids at time $T = 1$ with call option on the maximum of one asset.

Case	RMSE	CPU time		N_x	
		Adaptive	Uniform	Adaptive	Uniform
$h = 2$	1.179E-4	1	1	88	185
$h = 1$	1.179E-4	1	1.5	149	368
$h = 1/2$	1.179E-4	1	4.5	259	736
$h = 1/2^2$	1.179E-4	1	5.75	461	1470
$h = 1/2^3$	1.179E-4	1	14.50	851	2939
$h = 1/2^4$	1.179E-4	1	24.73	1619	5877
$h = 1/2^5$	1.179E-4	1	34.61	3137	11754
$h = 1/2^6$	1.179E-4	1	44.45	6165	23506
$h = 1/2^7$	1.179E-4	1	58.99	12207	47011
$h = 1/2^8$	1.179E-4	1	78.99	24301	94022

Figure 4 shows the result of the RMSE on $[0.95K, 1.05K]$ with different N_x at time $T = 1$. In Fig. 4, ‘ \circ ’ and ‘ \diamond ’ show the results using $\bar{h} = 1$ and 0.25, respectively and ‘ \bullet ’ and ‘ \blacklozenge ’ represent $\bar{h} = 1$ and 0.25 on the uniform mesh, respectively. From these results, we see the convergence of the relative RMSE of the adaptive grid as the number of grid points around the strike price K increases.

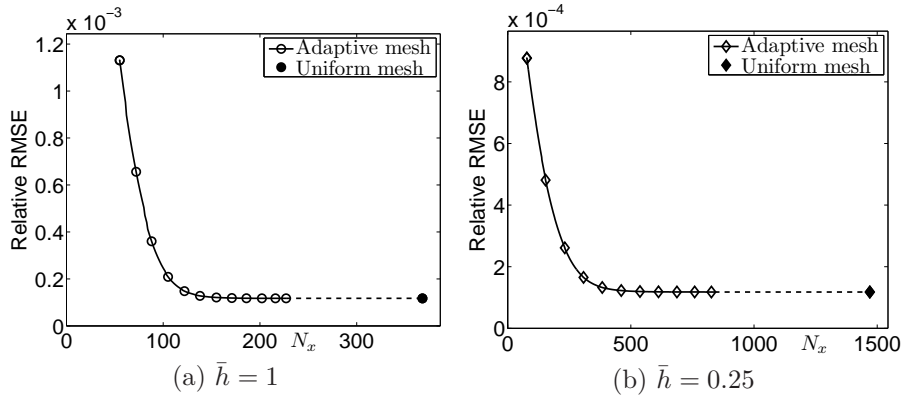


FIGURE 4. Relative RMSE on $[0.95K, 1.05K]$ with different N_x at $T = 1$. Lines with symbols, ‘ \circ ’ and ‘ \diamond ’ represent $\bar{h} = 1$ and 0.25 on the adaptive mesh, respectively. Also, symbols, ‘ \bullet ’ and ‘ \blacklozenge ’ represent $\bar{h} = 1$ and 0.25 on the uniform mesh, respectively.

3.2.2. Cash-or-nothing option. Next, we perform the same comparison study using a cash-or-nothing option. The initial state is given by

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

For this test, we use $Cash = 100$. Table 2 shows computational results such as relative CPU time, RMSE on $[0.95K, 1.05K]$ and grid points N_x at time $T = 1$ on adaptive and uniform grids with different space step size $h = 2/2^m$, where $m = 0, 1, 2, \dots, 9$. We can see that the adaptive grid technique is more efficient than uniform grid.

Figure 5 shows the result of the RMSE on $[0.95K, 1.05K]$ with different N_x at time $T = 1$. In Fig. 5, ‘ \circ ’ and ‘ \diamond ’ show the results of $\bar{h} = 1$ and 0.25 on the adaptive mesh, respectively. And symbols, ‘ \bullet ’ and ‘ \blacklozenge ’ represent $\bar{h} = 1$ and 0.25 on the uniform mesh, respectively. From these results, we see the convergence of the relative RMSE of the adaptive grid as the number of grid points around the strike price K increases.

4. Conclusions

An accurate and efficient numerical method for solving the Black–Scholes equation was derived in this article. The method uses an adaptive technique which is based on a far-field boundary position of the BS equation and the Peclet condition for non-oscillatory solutions. Furthermore, since the proposed adaptive grid method is based on a far-field boundary position and the Peclet

TABLE 2. Comparison of relative CPU time and grid points, N_x on adaptive and uniform grids at time $T = 1$ for a given relative RMSE tolerance with a cash-or-nothing option payoff.

Case	RMSE	CPU time		N_x	
		Adaptive	Uniform	Adaptive	Uniform
$h = 2$	$3.548E-5$	1	1.36	88	185
$h = 1$	$3.548E-5$	1	1.5	155	368
$h = 1/2$	$2.760E-5$	1	4.0	232	736
$h = 1/2^2$	$2.760E-5$	1	6.0	394	1470
$h = 1/2^3$	$2.760E-5$	1	18.88	710	2939
$h = 1/2^4$	$2.760E-5$	1	45.15	1332	5877
$h = 1/2^5$	$2.760E-5$	1	47.59	2564	11754
$h = 1/2^6$	$2.760E-5$	1	54.10	5014	23506
$h = 1/2^7$	$2.760E-5$	1	71.31	9904	47011
$h = 1/2^8$	$2.760E-5$	1	108.02	19672	94022

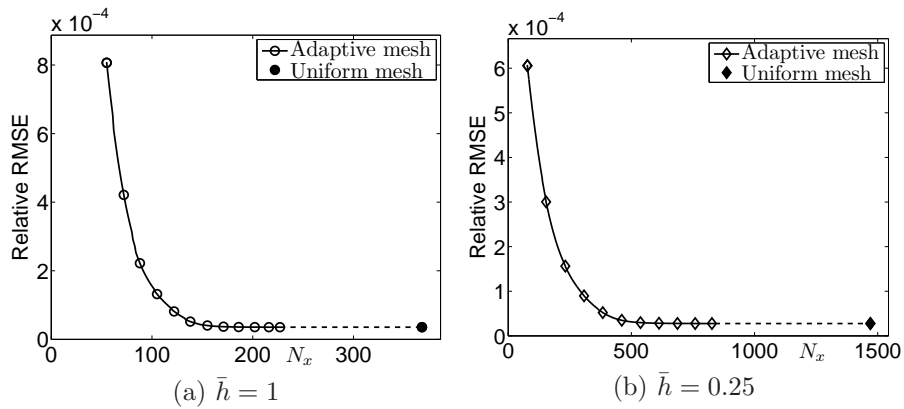


FIGURE 5. Relative RMSE on $[0.95K, 1.05K]$ with different N_x at $T = 1$. Lines with symbols, 'o' and '◇' represent $\bar{h} = 1$ and 0.25 on the adaptive mesh, respectively. Also, symbols, '•' and '◆' represent $\bar{h} = 1$ and 0.25 on the uniform mesh, respectively.

condition for non-oscillatory solutions, we get efficient and accurate numerical solutions.

In this paper, to demonstrate the accuracy and efficiency of our proposed method, numerical tests were performed. Test results show that the computational time on the adaptive grid was reduced substantially when compared to

the uniform grid. From these numerical results, we confirmed the effectiveness of the proposed adaptive grid method.

Appendix A. Far field boundary condition

Let u and w be classical solutions of Eqs.(2) and (3) on an infinite domain $(0, \infty) \times (0, T]$ and a finite domain $(0, S_{\max}) \times (0, T]$, respectively. Assume that, for some positive constant ν_0, ν_1, κ_0 and κ_1 , the payoff function $p(x)$ is satisfied as

$$(11) \quad -\nu_0 + \nu_1 x \leq p(x) \leq \kappa_0 + \kappa_1 x$$

for $x \in (0, \infty)$. Then

$$(12) \quad -\nu_0 e^{-r\tau} + \nu_1 x \leq u(x, \tau) \leq \kappa_0 e^{-r\tau} + \kappa_1 x$$

for $(x, \tau) \in (0, \infty) \times (0, T)$. Then, at every point $(x, \tau) \in (0, S_{\max}) \times (0, T]$ satisfying

$$(13) \quad \ln \frac{S_{\max}}{x} \geq -(\sigma^2 - 2r)\tau,$$

we have

$$(14) \quad |u(x, \tau) - w(x, \tau)| \leq \|u - w\|_{L_\infty(S_{\max} \times (0, \tau))} e^{\frac{-\ln \frac{S_{\max}}{K} (\ln \frac{S_{\max}}{K} + \min\{0, \sigma^2 - 2r\})\tau}{2\sigma^2\tau}}.$$

Please refer to [11] for more details about the far-field boundary conditions for the Black-Scholes equations.

Appendix B. Stability condition

In this Appendix, we will derive the conditions under which the implicit scheme for Eq.(2) will not make spurious oscillations by using the idea in reference [26]. For simplicity, we define as $k_i = \frac{1}{2}\sigma^2 x_i^2$ and $a_i = r x_i$, then we rewrite Eq.(5) as

$$(15) \quad \frac{(a_i h_i - k_i)\Delta\tau}{h_{i-1}(h_{i-1} + h_i)} u_i^{n+1} + \frac{(k_i - a_i(h_i - h_{i-1}))\Delta\tau + (1 + r\Delta\tau)h_{i-1}h_i}{h_{i-1}h_i} u_i^{n+1} - \frac{(k_i + a_i h_{i-1})\Delta\tau}{h_i(h_{i-1} + h_i)} u_{i+1}^{n+1} = u_i^n.$$

Next, we substitute $u_i^{n+1} = \beta_i^{n+1}/(1 + r\Delta\tau)^n$ into Eq.(15), where the superscript n for $(1 + r\Delta\tau)$ represents an exponent. Then, we obtain

$$(16) \quad \frac{(k_i - a_i(h_i - h_{i-1}))\Delta\tau + (1 + r\Delta\tau)h_{i-1}h_i}{h_{i-1}h_i} \beta_i^{n+1} = (1 + r\Delta\tau) \beta_i^n + \frac{(k_i - a_i h_i)\Delta\tau}{h_{i-1}(h_{i-1} + h_i)} \beta_{i-1}^{n+1} + \frac{(k_i + a_i h_{i-1})\Delta\tau}{h_i(h_{i-1} + h_i)} \beta_{i+1}^{n+1}.$$

In order for all coefficients of β_i^n in Eq.(16) to be positive, $k_i - h_i a_i > 0$ should be satisfied. That is, we have the Peclet condition [26]:

$$\frac{1}{h_i} > \frac{r}{\sigma^2 x_i}.$$

Now, if the Peclet condition is satisfied, then all the coefficients of β in Eq.(16) are positive. Let $\beta_i^{\max} = \max(\beta_i^n, \beta_{i-1}^{n+1}, \beta_{i+1}^{n+1})$, then Eq.(16) can be written as

$$\begin{aligned} & \frac{(k_i - a_i(h_i - h_{i-1}))\Delta\tau + (1 + r\Delta\tau)h_{i-1}h_i}{h_{i-1}h_i} \beta_i^{n+1} \\ & \leq (1 + r\Delta\tau) \beta_i^{\max} + \frac{(k_i - a_i h_i)\Delta\tau}{h_{i-1}(h_{i-1} + h_i)} \beta_i^{\max} + \frac{(k_i + a_i h_{i-1})\Delta\tau}{h_i(h_{i-1} + h_i)} \beta_i^{\max}. \end{aligned}$$

Therefore,

$$(17) \quad \beta_i^{n+1} \leq \beta_i^{\max}.$$

And by a similar argument we obtain

$$(18) \quad \beta_i^{n+1} \geq \beta_i^{\min},$$

where $\beta_i^{\min} = \min(\beta_i^n, \beta_{i-1}^{n+1}, \beta_{i+1}^{n+1})$. By Eqs. (17) and (18), new local maxima or minima of the numerical solution for β_i^{n+1} can not occur. Since $u_i^{n+1} = \beta_i^{n+1}/(1 + r\Delta\tau)^n$, the numerical solution u_i^{n+1} does not contain oscillations if conditions (17) and (18) are satisfied.

References

- [1] Y. Achdou and O. Pironneau, *Computational Methods for Option Pricing*, SIAM, Philadelphia, 2005.
- [2] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, J. Political Econ. **81** (1973), no. 3, 637–659.
- [3] M. Brennan and E. Schwartz, *The valuation of American put options*, J. Financ. **32** (1977), no. 2, 449–462.
- [4] ———, *Finite difference methods and jump processes arising in the pricing of contingent claims: a synthesis*, J. Financ. Quant. Anal. **13** (1978), no. 3, 461–474.
- [5] G. W. Buetow and J. S. Sochacki, *The trade-off between alternative finite difference techniques used to price derivative securities*, Appl. Math. Comput. **115** (2000), no. 2-3, 177–190.
- [6] C. Christara and D. M. Dang, *Adaptive and high-order methods for valuing American options*, J. Comput. Financ. **14** (2011), no. 4, 74–113.
- [7] D. J. Duffy, *Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach*, John Wiley & Sons, New York, 2006.
- [8] S. Figlewski and B. Gao, *The adaptive mesh model: a new approach to efficient option pricing*, J. Financ. Econ. **53** (1999), no. 3, 313–351.
- [9] R. Geske and K. Shastri, *Valuation by approximation: a comparison of alternative option valuation techniques*, J. Financ. Quant. Anal. **20** (1985), no. 1, 45–71.
- [10] D. Jeong, *Mathematical model and numerical simulation in computational finance*, Ph.D. Thesis, Dep. Mathematics, Korea Univ., Korea, 2012.
- [11] R. Kangro and R. Nicolaides, *Far field boundary conditions for Black–Scholes equations*, SIAM J. Numer. Anal. **38** (2000), no. 4, 1357–1368.

- [12] B. J. Kim, C. Ahn, and H. J. Choe, *Direct computation for American put option and free boundary using finite difference method*, Jpn. J. Ind. Appl. Math. **30** (2013), no. 1, 21–37.
- [13] G. Linde, J. Persson, and L. von Sydow, *A highly accurate adaptive finite difference solver for the Black–Scholes equation*, Int. J. Comput. Math. **86** (2009), no. 12, 2104–2121.
- [14] P. Lötstedt, S. Söderberg, A. Ramage, and L. Hemmingsson-Frändén, *Implicit solution of hyperbolic equations with space-time adaptivity*, BIT Numer. Math. **42** (2002), no. 1, 134–158.
- [15] P. Lötstedt, J. Persson, L. von Sydow, and J. Tysk, *Space-time adaptive finite difference method for European multi-asset options*, Comput. Math. Appl. **53** (2007), no. 8, 1159–1180.
- [16] R. C. Merton, *Theory of rational option pricing*, Bell J. Econ. Manag. Sci. **4** (1973), no. 1, 141–183.
- [17] J. Persson and L. von Sydow, *Pricing European multi-asset options using a space-time adaptive FD-method*, Comput. Vis. Sci. **10** (2007), no. 4, 173–183.
- [18] ———, *Pricing American options using a space-time adaptive finite difference method*, Math. Comput. Simulation **80** (2010), no. 9, 1922–1935.
- [19] O. Pironneau and F. Hecht, *Mesh adaption for the Black and Scholes equations*, East-West J. Numer. Math. **8** (2000), no. 1, 25–35.
- [20] E. Schwartz, *The valuation of warrants: Implementing a new approach*, J. Financ. Econ. **4** (1977), no. 1, 79–93.
- [21] D. Tavella and C. Randall, *Pricing Financial Instruments: The Finite Difference Method*, John Wiley & Sons, New York, 2000.
- [22] J. Topper, *Financial Engineering with Finite Elements*, John Wiley & Sons, New York, 2005.
- [23] P. Wilmott, J. Dewynne, and S. Howison, *Option Pricing: Mathematical Models and Computation*, Oxford Financial Press, Oxford, 1993.
- [24] R. Windcliff, P. A. Forsyth, and R. A. Vetzal, *Analysis of the stability of the linear boundary condition for the Black–Scholes equation*, J. Comput. Financ. **8** (2004), 65–92.
- [25] S. Zhao and G. W. Wei, *Option valuation by using discrete singular convolution*, Appl. Math. Comput. **167** (2005), no. 1, 383–418.
- [26] R. Zvan, P. A. Forsyth, and K. R. Vetzal, *Robust numerical methods for PDE models of Asian options*, J. Comput. Financ. **1** (1998), 39–78.

DARAE JEONG
DEPARTMENT OF MATHEMATICS
KOREA UNIVERSITY
SEOUL 136-713, KOREA
E-mail address: tinayoyo@korea.ac.kr

TAEYOUNG HA
DIVISION OF COMPUTATIONAL SCIENCES IN MATHEMATICS
NATIONAL INSTITUTE FOR MATHEMATICAL SCIENCES
DAEJEON 305-811, KOREA
E-mail address: tha@nims.re.kr

MYOUNGNYOUN KIM
DIVISION OF COMPUTATIONAL SCIENCES IN MATHEMATICS
NATIONAL INSTITUTE FOR MATHEMATICAL SCIENCES
DAEJEON 305-811, KOREA
E-mail address: gmnkim@nims.re.kr

JAEMIN SHIN
INSTITUTE OF MATHEMATICAL SCIENCES
EWHA WOMANS UNIVERSITY
SEOUL 120-750, KOREA
E-mail address: `zmshin@korea.ac.kr`

IN-HAN YOON
DEPARTMENT OF MATHEMATICS
KOREA UNIVERSITY
SEOUL 136-713, KOREA
E-mail address: `ketaro0910@korea.ac.kr`

JUNSEOK KIM
DEPARTMENT OF MATHEMATICS
KOREA UNIVERSITY
SEOUL 136-713, KOREA
E-mail address: `cfdkim@korea.ac.kr`