

VARIABLE TIME-STEPPING HYBRID FINITE DIFFERENCE METHODS FOR PRICING BINARY OPTIONS

HONGJOONG KIM[†] AND KYOUNG-SOOK MOON[‡]

ABSTRACT. Two types of new methods with variable time steps are proposed in order to value binary options efficiently. Type I changes adaptively the size of the time step at each time based on the magnitude of the local error, while Type II combines two uniform meshes. The new methods are hybrid finite difference methods, namely starting the computation with a fully implicit finite difference method for a few time steps for accuracy then performing a θ -method during the rest of computation for efficiency. Numerical experiments for standard European vanilla, binary, and American options show that both Type I and II variable time step methods are much more efficient than the fully implicit method or hybrid methods with uniform time steps.

1. Introduction

Options are traded on all of the world's major exchanges. Among them, binary options or digital options are not only very popular in the over-the-counter (OTC) markets but also important tools for designing more complex financial derivatives, such as equity-linked-securities (ELS). For instance, holding the simplest cash-or-nothing call option pays a predefined cash amount at expiration date if the option is in-the-money, i.e., the prices of the underlying assets exceed a prescribed strike price, otherwise pays nothing. Therefore the payoff at expiration date has a discontinuity at the strike price. Several exotic options, such as asset-or-nothing options, supershare options, gap options etc, are also binary type options.

Received September 21, 2009.

2010 *Mathematics Subject Classification.* Primary 91B28, 65C20.

Key words and phrases. option pricing, variable time steps, hybrid finite difference method, binary options, American options.

[‡] This research was supported by National Research Foundation of Korea Grant funded by the Korean Government (KRF-2008-531-C00016) and the Kyungwon University Research Fund in 2011.

[†] This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0012584).

Let us assume that the underlying assets follow geometric Brownian motions. Based on no arbitrage arguments and no cost for trading, Black, Scholes and Merton derived the celebrated Black-Scholes *partial differential equation* (PDE) for the valuation of European stock options in [2], [12]. The prices of binary options also satisfy the classical Black-Scholes PDE with final conditions containing discontinuities and may have the closed forms, see [16]. However, in many cases, binary options are embedded in complex structured derivatives or American type, therefore one need to apply numerical approximations in order to value the derivatives.

Among several approaches of numerical methods, we consider the most popular *finite difference method* (FDM), since it is efficient and can handle American type easily, see [1, 3, 8, 10, 17, 18, 19] for a general framework of FDM for option pricing. Even though the solution to the Black-Scholes equation is smooth, the final condition has discontinuity and this produces oscillation in the numerical solutions using classical finite difference methods. In particular, the Greeks, such as Δ (*delta*, the partial derivative of option price with respect to asset price) and Γ (*gamma*, the second partial derivative of option price with respect to asset price), which are very important quantities to measure the sensitivity of the option prices for hedging, are even harder to compute accurately using classical finite difference methods.

In order to cure this oscillation from the initial discontinuities, there have been studied different numerical methods. One approach is to use cell-averaged payoff instead of grid sampled payoff function to reduce the error near discontinuity, see [18]. Heston and Zhou [7] proposed a smoothing approach, to smooth the payoff function at singular points, and an adjustment approach to replace the binomial prices prior to the end of the tree by the Black-Scholes values. Rannacher [15] proposed a simple damping procedure, starting the computation with the backward Euler steps and showed that the procedure yields the full second order accuracy. This Rannacher idea was applied to Black-Scholes equation in [5]. Pooley, Vetzal, and Forsyth [14] compared various smoothing techniques together with Rannacher time-stepping based on finite element analysis.

The purpose of this paper is to develop an efficient and accurate numerical method to price options with payoff containing discontinuities. We first introduce adaptive mesh refinement for time steps (denote Type I), in order to compute accurately the prices near discontinuity. In this work, we focus on adaptivity for time steps only in order to see the effects of variable time step, but one can simply extend the method to spacial adaptivity as well, see time and space adaptive algorithms of finite difference method [11] for pricing European vanilla options and of finite element method [13] for pricing American options. See also [9] and [4] for the penalty method with time adaptivity for American option problems. Our adaptive refinement strategy is performed by an a posteriori error approximation based on local errors.

For standard binary options, the discontinuity lies only in the initial condition, therefore we need to use small time steps initially then use bigger time steps to keep the efficiency. Therefore, specially for binary options, we suggest a method which combines two uniform grids in space and time (denote Type II). Type II method uses smaller grid for the first few time steps then uses bigger grid. However, if there exists several singularities on a given exotic derivative, then the Type I method may refine the time steps near each singular point. In this respect, Type I is more general than Type II method.

In particular, inspiring from Rannacher in [15], both Type I and II methods are hybrid finite difference methods, namely initially we apply a fully implicit finite difference method to cure the oscillations from the discontinuities, then later we use any finite difference method such as the Crank-Nicolson method during the rest of computation for an efficiency.

Several numerical tests in Section 4 show that both Type I and II methods approximate the solution more accurately and efficiently than uniform time stepping method. Type I adaptive method gets more efficient than Type II and uniform methods as the accuracy increases, see the figures in Section 4 .

The outline of the paper is as follows. In Section 2 we first set up the problem to price European and American binary options. In Section 3 we explain new numerical schemes with adaptive time stepping. In Section 4 we show the numerical experiments and the conclusions are drawn in Section 5.

2. Option pricing

Let $s(t)$ be the price of the underlying asset at time t ($0 \leq t \leq T$) with a given expiry date T and follow a geometric Brownian motion with a constant interest rate $r > 0$ and a constant volatility $\sigma > 0$. From no arbitrage arguments, the value, $V(s, t)$ of binary options under classical Black-Scholes model can be computed by solving the following partial differential equations

$$(1) \quad \frac{\partial V}{\partial t} + rs \frac{\partial V}{\partial s} + \frac{1}{2} \sigma^2 s^2 \frac{\partial^2 V}{\partial s^2} - rV = 0$$

with a final condition

$$(2) \quad V(s, T) = \Lambda(s),$$

where $\Lambda(s)$ is a payoff function at expiry T .

For instance, European cash-or-nothing call options pay a cash amount A at expiration T if the option is in-the-money, i.e.,

$$\Lambda(s(T)) = \begin{cases} A & \text{if } s(T) > K, \\ A/2 & \text{if } s(T) = K, \\ 0 & \text{otherwise,} \end{cases}$$

where $K > 0$ is a predefined strike price. Here, since the probability of the event $s(T) = K$ is zero, in order to be consistent with the formula, we assume that $A/2$ is paid off in this at-the-money case, see also [8]. From the change of

variables, one can convert the PDE (1) to a heat equation and find an analytic formula for the cash-or-nothing call option, $V(s, t) = C(s, t)$, described in [16]

$$(3) \quad C(s, t) = Ae^{-r(T-t)}N(d), \quad d = \frac{\ln(s/K) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}.$$

Here $N(x)$ is the cumulative normal distribution function. From this formula, we can readily compute the Greeks for call options by

$$(4) \quad \Delta_C := \frac{\partial C}{\partial s} = \frac{Ae^{-r(T-t)}N'(d)}{\sigma s\sqrt{T-t}}, \quad N'(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2},$$

and

$$(5) \quad \Gamma_C := \frac{\partial^2 C}{\partial s^2} = -\frac{Ae^{-r(T-t)}d_1N'(d)}{\sigma^2 s^2(T-t)}, \quad d_1 = d + \sigma\sqrt{T-t}.$$

The boundary conditions of European cash-or-nothing call options are the following: for $s = 0$, the asset remains at zero for all times, thus the payoff is also zero and for large $s \gg 0$, it is almost certain to get the cash rebate A , thus we may use the following as our boundary condition in a computational domain.

$$(6) \quad C(s, t) \approx Ae^{-r(T-t)} \text{ for large } s.$$

American options are similar to European options except that the holder may exercise the option at any time before the expiration. Therefore the exercise time is not known in priori, this is so called free boundary problem [1, 19]. From no arbitrage arguments, the prices of American options satisfy the following linear complementary problem, see [8, 19]

$$(7) \quad \frac{\partial V}{\partial t} + rs\frac{\partial V}{\partial s} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 V}{\partial s^2} - rV \leq 0,$$

$$(8) \quad V(s, t) \geq \Lambda(s(t)), \quad 0 \leq t \leq T, s \geq 0$$

for each s, t one of (7) and (8) is at equality

with the final condition $V(s, T) = \Lambda(s)$, where $\Lambda(s)$ is a payoff function at expiry T . The above linear complementary problem does not admit analytic solutions, therefore we should rely on numerical approximations.

For instance, for the American vanilla put case the payoff is $\Lambda(s) = \max(K - s, 0)$ with a strike price $K > 0$. The boundary conditions of American vanilla put options $V(s, t) = P(s, t)$ are the following: For $s = 0$, the asset always remains zero, therefore the payoff at expiry is the exercise price, i.e., $P(0, t) = K$ for all $0 \leq t \leq T$. Similarly if s is large, then the option is unlikely to produce a positive payoff, so we have $P(s, t) \rightarrow 0$ as $s \rightarrow \infty$ for all $0 \leq t \leq T$. In our numerical test, for sufficiently large s_{\max} , we use the Neumann boundary condition

$$\frac{\partial P}{\partial s}(s_{\max}, t) = 0 \quad \text{for all } 0 \leq t \leq T.$$

In the numerical computations, we first convert the original backward equation in (1) and (2) to the following forward equation using the change of variables $u(x, \tau) := V(s, T - t)$, $x = s$, $\tau = T - t$,

$$(9) \quad \begin{aligned} \frac{\partial u}{\partial \tau} &= rx \frac{\partial u}{\partial x} + \frac{1}{2} \sigma^2 x^2 \frac{\partial^2 u}{\partial x^2} - ru, \\ u(x, 0) &= \Lambda(x). \end{aligned}$$

Similarly, for American case, we have

$$(10) \quad \frac{\partial u}{\partial \tau} \geq rx \frac{\partial u}{\partial x} + \frac{1}{2} \sigma^2 x^2 \frac{\partial^2 u}{\partial x^2} - ru,$$

$$(11) \quad V(x, \tau) \geq \Lambda(x), \quad 0 \leq \tau \leq T, x \geq 0,$$

for each x, τ one of (10) and (11) is at equality,

$$u(x, 0) = \Lambda(x).$$

3. Numerical schemes

3.1. Finite difference methods

The domain $[x_{\min}, x_{\max}] \times [0, T]$ is discretized into M intervals in space and N intervals in time,

$$x_{\min} = x_0 < x_1 < \cdots < x_M = x_{\max}, \quad \text{and} \quad 0 = \tau_0 < \tau_1 < \cdots < \tau_N = T.$$

Given a function $u = u(x, \tau)$, let U_j^n denote the numerical approximation of u at $(x, \tau) = (x_j, \tau_n)$. The time derivative u_τ is approximated by

$$u_\tau(x_j, \tau_n) = \frac{U_j^{n+1} - U_j^n}{\Delta \tau_n} + O(\Delta \tau_{\max}),$$

where a time step size $\Delta \tau_n := \tau_{n+1} - \tau_n$ and $\Delta \tau_{\max}$ denotes the maximum of $\Delta \tau_n$'s, $\Delta \tau_{\max} := \max_n \Delta \tau_n$. The first spatial derivative u_x is approximated by the θ -method

$$u_x(x_j, \tau_n) = \theta \frac{U_{j+1}^{n+1} - U_{j-1}^{n+1}}{2\Delta x} + (1 - \theta) \frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x} + O(\Delta x^2)$$

with the spatial step size $\Delta x := (x_{\max} - x_{\min})/M$ and the second spatial derivative u_{xx} by

$$u_{xx}(x_j, \tau_n) = \theta \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{\Delta x^2} + (1 - \theta) \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2} + O(\Delta x^2).$$

We apply these approximations into the equation (9) to obtain the following system

$$(12) \quad \begin{aligned} U_j^{n+1} &= U_j^n + rU_j^n \frac{\Delta \tau_n}{2\Delta x} (\theta (U_{j+1}^{n+1} - U_{j-1}^{n+1}) + (1 - \theta) (U_{j+1}^n - U_{j-1}^n)) \\ &\quad - \Delta \tau_n r U_j^n + \frac{1}{2} r^2 (U_j^n)^2 \frac{\Delta \tau_n}{\Delta x^2} (\theta (U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}) \\ &\quad + (1 - \theta) (U_{j+1}^n - 2U_j^n + U_{j-1}^n)). \end{aligned}$$

The weight $\theta = 0$ gives the explicit scheme, $\theta = 1$ the fully implicit scheme, and $\theta = \frac{1}{2}$ the Crank-Nicolson method. Rannacher [15] introduced a combination of the fully implicit ($\theta = 1$) method and the Crank-Nicolson ($\theta = 1/2$) method. Rannacher method uses the fully implicit scheme for the first 4 time steps and then applies the Crank-Nicolson scheme for the remaining of time steps. In this study, we extend this idea and want to consider a combination of various θ -methods with various time stepping schemes. The fully implicit method is applied during the early stage to reduce effects from initial or boundary conditions or from variables having discontinuity or large gradient (if it exists). Later time dissipation dominates in the solution of Black Scholes equation regardless of initial abrupt changes, then both implicit and explicit schemes can be used. Note that the explicit scheme ($\theta = 0$) is easy to implement, and does not need to solve a system of equations. But it is limited by the stability condition, which limits the size of the time step. For example, Figure 1 represents the values of Γ (gamma) of the European cash-or-nothing call option in Section 4.2 at $\tau = 1/1200$ from the explicit finite difference schemes with uniform $\Delta x = 0.25$ and $\Delta\tau = 1/2400$. Even for this short time $\tau = 1/1200 = 2\Delta\tau$, the explicit scheme with uniform time step (the dotted line) triggers an oscillation near the discontinuity.

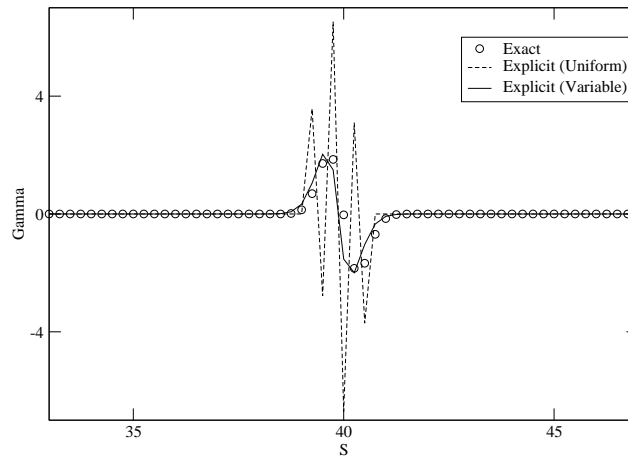


FIGURE 1. Gamma from the explicit methods with uniform and variable (Type I) time steps

In order to overcome this shortcoming, we introduce the combination of θ -methods with variable time step based on the adaptive mesh refinement, which may even reduce the computational cost.

3.2. Time-stepping algorithms

Giles [5] introduces methods with variable step sizes in time. But, they choose a very small $\Delta\tau$ initially, then increases the step size in time regardless of the magnitude of the error, which eventually degrades the accuracy of the computation. Forsyth [4], on the other hand, controls the step size based on the computation results by introducing a timestep selector. We introduce two types of time stepping methods, *Type I* and *Type II*. The *Type I variable time step method* uses non-uniform $\Delta\tau$ based on the following algorithm and fixed Δx . At each time $\tau = \tau_n$, $u(x_j, \tau_n + \Delta\tau_n)$ is approximated in two different ways, once using $\Delta\tau_n$ and once again using $\Delta\tau_n/2$ as time step size. Let U_j^{n+1} and \tilde{U}_j^{n+1} denote approximations of $u(x_j, \tau_n + \Delta\tau_n)$ with time step sizes $\Delta\tau_n$ and $\Delta\tau_n/2$, respectively. If the error defined by

$$(13) \quad E^n := \|U^{n+1} - \tilde{U}^{n+1}\|_\infty$$

is smaller than pre-defined tolerance, TOL, we use \tilde{U}_j^{n+1} as approximation of $u(x_j, \tau_n + \Delta\tau_n)$ and proceed in time. Here $\|U^{n+1}\|_\infty := \max_j |U_j^{n+1}|$ denotes L_∞ norm of U over x grids. If the error is greater than the TOL, then $\Delta\tau_n$ is halved, and then $u(x_j, \tau_n + \Delta\tau_n)$ is computed again in two different ways as above. This procedure is repeated until the error becomes smaller than the tolerance. If, on the other hand, the error defined above is within the tolerance, the next time step is changed by

$$(14) \quad \Delta\tau_{n+1} = \Delta\tau_n (1 + c \log_{10} 1/E^n),$$

where $c > 0$ is a problem independent constant. The log function prevents the step size from increasing too fast for a very small error. The algorithm is simple and controls the time step size adaptively. In the numerical examples below, the constant $c = 0.0025$ and the error tolerance $\text{TOL} = 10^{-5}$ are used. The algorithm can be summarized as follows:

[Type I Adaptive Algorithm]

- Step 0. Set up the parameters c and TOL. Set $n = 0$ and choose the initial time step size $\Delta\tau_0$.
- Step 1. Solve the system (12) twice with $\Delta\tau_n$ and $\Delta\tau_n/2$ to obtain U_j^{n+1} and \tilde{U}_j^{n+1} .
- Step 2. Compute the error E^n in (13).
- Step 3. If $E^n > \text{TOL}$, then $\Delta\tau_n$ is halved, and go to Step 1 to solve the system again, otherwise go to Step 4.
- Step 4. Save \tilde{U}_j^{n+1} , and set $\Delta\tau_{n+1}$ by (14) and go to Step 1 by increasing n by 1.

The *Type II variable time step method* consists of two pairs of uniform grids in space and time. It uses constant Δx_1 and $\Delta\tau_1$ as step sizes for space and time, respectively, for the first J steps in time. Then it uses another constant pair, Δx_2 and $\Delta\tau_2$, for the remaining computation, as in Figure 2. For accuracy,

we choose smaller grids for the initial J time steps, i.e., $\Delta x_1 < \Delta x_2$ and $\Delta \tau_1 < \Delta \tau_2$.

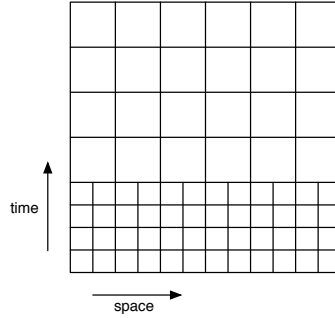


FIGURE 2. Grids for the Type II variable time step method

The fully implicit scheme is used for the first J steps in time for both Type I and II, and any θ -method is used for the remaining time steps. In the numerical tests, different numerical schemes have been applied to the numerical experiments. When the uniform time step is used, the fully implicit scheme (IM) and the combination of the fully implicit scheme and the Crank Nicolson method (IM-CN) have been used. The tridiagonal systems from IM or IM-CN schemes are solved by the Thomas algorithm [19]. When Type I or II variable time step is used, (IM), (IM-CN), and the combination of the fully implicit scheme and the explicit scheme (IM-EX) have been used. For the Type I variable time step methods, $\Delta \tau$ is initially chosen the same as those using uniform time step, then it is allowed to change adaptively following the algorithm above. For the Type II variable time step methods in the examples below, the spatial and time grids are related by

$$\Delta x_2 = 4\Delta x_1 \text{ and } \Delta \tau_2 = 4\Delta \tau_1,$$

and $\Delta \tau_2$ is the same as the step size of uniform method.

3.3. Free boundary problem

The prices of American options can be computed by solving the free boundary problem in (10, 11). There have been developed several numerical methods to solve the free boundary problems, see [3, 19]. In this work, since we focus more on the time stepping method, we simply apply projection method at each time step. For instance, we need to change the Step 1 of [Type I Adaptive Algorithm] in Section 3.2 as follows:

Step 1'. Solve the system (12) twice with $\Delta \tau_n$ and $\Delta \tau_n/2$ to obtain \hat{U}_j^{n+1} and $\hat{\tilde{U}}_j^{n+1}$. To satisfy the condition (11) for

American case, we project the approximation to the pay-off function, i.e.,

$$U_j^{n+1} = \max \left(\hat{U}_j^{n+1}, \max(K - x_j, 0) \right),$$

$$\tilde{U}_j^{n+1} = \max \left(\hat{\tilde{U}}_j^{n+1}, \max(K - x_j, 0) \right).$$

4. Numerical experiments

In this section, we compare Type I and II variable time stepping methods to uniform time stepping method. Different finite difference schemes explained above have been applied to the examples, European and American vanilla/binary options. At each case we compute the approximation errors between the exact values and numerically estimated values in L_∞ -norm,

$$(15) \quad \mathcal{E} := \|u(\cdot, T) - U^N\|_\infty.$$

Computations are performed on Apple Imac computer with 2.4 GHz Intel Core 2 Duo processor and 2 GB RAM memory.

4.1. European vanilla option

Let us first consider the European vanilla option. We solve the initial value problem (9) numerically and compare the results with the exact prices, for example, see [2, 12, 6, 8, 19]. Table 1 shows parameters used for the computation. The computation has been performed over the domain $[x_{\min}, x_{\max}] \times [0, T] = [0, 80] \times [0, 0.5]$.

TABLE 1. Parameters used for pricing the European vanilla option.

Type	Put
Time to expiry (T)	6 months
Interest rate (r)	5%
Exercise price (K)	40
Volatility (σ)	30%

Table 2 shows the numbers of time steps (N), the errors of option prices (\mathcal{E}_{option}) in (15), and the errors of delta (\mathcal{E}_Δ) and gamma (\mathcal{E}_Γ) values, when the combination of the fully implicit scheme and the Crank Nicolson method (IM-CN) with uniform time step or variable time steps (Type I, II) are used. From this table, Type I adaptive time step method results in as accurate prices and Greeks as the uniform and Type II methods at much reduced costs. As M increases, saving from Type I adaptive method increases. For instance, when $M = 640$ is used, Type I need around 5 times less number of time steps than the other methods.

TABLE 2. The numbers of time steps (N) and the errors of the option prices (\mathcal{E}_{option}), delta (\mathcal{E}_Δ) and gamma (\mathcal{E}_Γ) for the European vanilla put option based on the IM-CN method.

M	N	Uniform			Type I			Type II				
		\mathcal{E}_{option}	\mathcal{E}_Δ	\mathcal{E}_Γ	N	\mathcal{E}_{option}	\mathcal{E}_Δ	\mathcal{E}_Γ	N	\mathcal{E}_{option}	\mathcal{E}_Δ	\mathcal{E}_Γ
20	40	0.097425	0.007592	0.0018257	162	0.096339	0.0075372	0.0017939	52	0.03345	0.010706	0.0013013
40	80	0.02348	0.0022545	0.00051243	226	0.023224	0.0022404	0.00050117	92	0.008561	0.0028823	0.00040162
80	160	0.0058541	0.0005725	0.00013373	262	0.0057913	0.0005674	0.0001307	172	0.0021933	0.00073165	0.00010158
160	320	0.0014612	0.00014446	3.3357e-05	269	0.0014426	0.00014289	3.2435e-05	332	0.00055256	0.00018406	2.5796e-05
320	640	0.00036535	3.6142e-05	8.3345e-06	270	0.00035821	3.5387e-05	7.9612e-06	652	0.00013874	4.6136e-05	6.4804e-06
640	1280	9.1327e-05	9.0428e-06	2.0848e-06	272	8.7059e-05	8.5015e-06	1.8522e-06	1292	3.4774e-05	1.1541e-05	1.6248e-06

Figure 3 shows errors in option pricing with respect to the computational time. Type II (triangle) uses smaller grids for the first few steps in time compared to uniform time step (solid line), thus CPU time is not altered much. But, such a reduction of grids for a few initial steps improves the accuracy by several times as in Table 2 or Figure 3. Figure 3 shows that Type II improves the accuracy compared to uniform time step given the same computational time. But, Type II does not change the slope of the curve. In case of Type I (circle), the slope of the curve is steeper than those of the uniform or Type II time steps. Thus, when CPU time is small, Type II gives better accuracy. But, as CPU time increases, Type I becomes more efficient than the others.

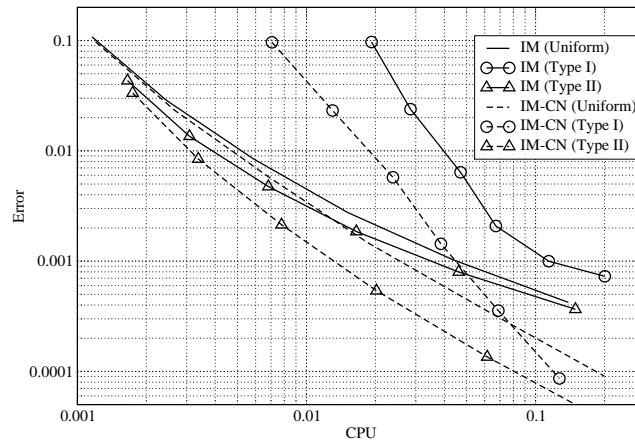


FIGURE 3. Errors in option pricing (Error) vs. the computational cost (CPU) for the European vanilla put option based on the IM-CN method.

4.2. European cash-or-nothing option

Type I and II algorithms are simple that they can be easily applicable to various exotic cases, such as binary options or American options. The European

TABLE 3. Parameters used for pricing the cash-or-nothing option.

Type	Call
Time to expiry (T)	6 months
Interest rate (r)	5%
Exercise price (K)	40
Volatility (σ)	30%
Cash rebate (A)	1.0

cash-or-nothing call option pays a predefined cash rebate $A > 0$ if the terminal price of the underlying asset exceeds the exercise price $K > 0$. Table 3 shows parameters used for the computation for the cash-or-nothing option. The computation has been performed over the domain $[x_{\min}, x_{\max}] \times [0, T] = [0, 80] \times [0, 0.5]$.

Figure 4 shows errors in option prices vs. computational costs (CPU times) based on the fully Implicit, or IM-CN method with uniform or variable time stepping methods. We see the same trend as in the European vanilla option. When Type II variable time step method is used, the accuracy is improved several times compared to the uniform time step, but the slope is still not changed. Though Type I variable time step method needs extra work to compute local errors, Type I method is the most efficient under the same level of accuracy, when the error tolerance is given sufficiently small.

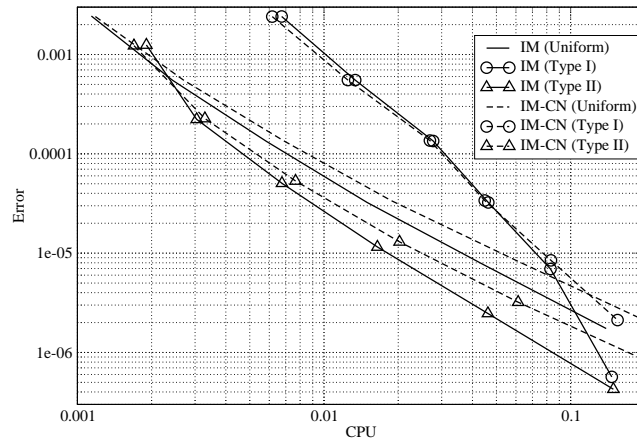


FIGURE 4. Errors in option pricing (Error) vs. computational cost (CPU) for the cash-or-nothing option from various schemes.

4.3. American cash-or-nothing option

Variable time step methods can be easily extended to American options based on the projection method in Section 3.3. Let us consider American cash-or-noting put option with uniform or variable time steps. Unlike the European options, the American options can be exercised at any time before the expiry. The same parameters as Table 3 are used except the type of the option. The computational domain is $[x_{\min}, x_{\max}] \times [0, T] = [0, 160] \times [0, 1]$. Since the exact option price is not available, the values from the fully Implicit method with $M = 10240$ uniform steps are used as the benchmark values. Hybrid finite difference schemes with variable time steps all work very well. Accuracy and computational cost are important factors in option pricing, especially in American options. The results show that variable time steps reduce the computational time over the uniform step. That is, variable time step methods use fewer time steps, and they give comparable accuracy. Figure 5 shows errors vs. computational costs for the American cash-or-noting put option. Similarly to results from Figure 3 and Figure 4, values from the variable time step methods converge faster than those using uniform time steps.

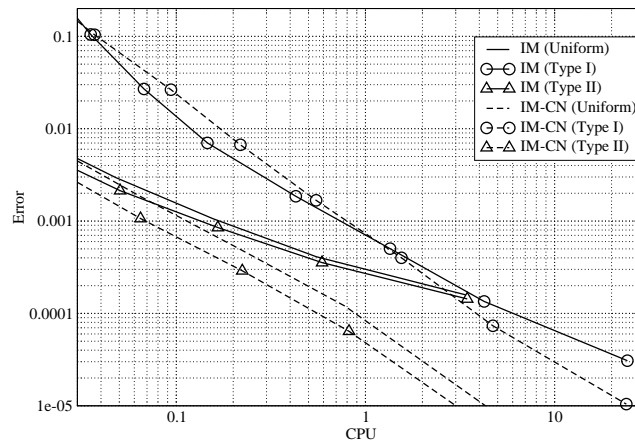


FIGURE 5. Errors in option pricing (Error) vs. computational cost (CPU) for the American cash-or-nothing put option based on IM and IM-CN methods.

5. Conclusion

A new adaptive time-stepping hybrid finite difference method (Type I) and a method with the combination of two uniform meshes (Type II) are proposed to solve the Black-Scholes partial differential equation with a payoff containing discontinuities. The time step sizes of Type I are adaptively chosen by the control of local errors. Numerical experiments show that the new adaptive

Type I method is much more efficient than Type II and the method with uniform time steps. Type II improves the accuracy by several times over the uniform time step methods. Studies of these methods together with space adaptivity in higher dimensions with more complicated payoff will be subject of future research.

References

- [1] Y. Achdou and O. Pironneau, *Computational Methods for Option Pricing*, Frontiers in Applied Mathematics, 30. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.
- [2] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, Journal of Political Economy **81** (1973), 637–659.
- [3] D. J. Duffy, *Finite Difference Methods in Financial Engineering*, Wiley Finance Series. John Wiley & Sons, Ltd., Chichester, 2006.
- [4] P. A. Forsyth and K. R. Vetzal, *Quadratic convergence for valuing American options using a penalty method*, SIAM J. Sci. Comput. **23** (2002), no. 6, 2095–2122.
- [5] M. B. Giles and B. Carter, *Convergence analysis of Crank-Nicolson and Rannacher time-marching*, J. Computat. Finance **9** (2006), 89–112.
- [6] E. G. Haug, *The Complete Guide to Option Pricing Formulas (2nd ed.)*, McGraw-Hill, 2007.
- [7] S. Heston and G. Zhou, *On the rate of convergence of discrete-time contingent claims*, Math. Finance **10** (2000), no. 1, 53–75.
- [8] D. J. Higham, *An Introduction to Financial Option Valuation*, Mathematics, stochastic and computation. Cambridge University Press, Cambridge, 2004.
- [9] A. Q. M. Khaliq, D. A. Voss, and K. Kazmi, *Adaptive θ -methods for pricing American options*, J. Comput. Appl. Math. **222** (2008), no. 1, 210–227.
- [10] Y. K. Kwok, *Mathematical Models of Financial Derivatives*, Springer-Verlag Singapore, 1998.
- [11] 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.
- [12] R. C. Merton, *Theory of rational option pricing*, Bell J. Econom. and Management Sci. **4** (1973), 141–183.
- [13] K.-S. Moon, R. Nochetto, T. von Petersdorff, and C. S. Zhang, *A posteriori error analysis for parabolic variational inequalities*, M2AN Math. Model. Numer. Anal. **41** (2007), no. 3, 485–511.
- [14] D. M. Pooley, K. R. Vetzal, and P. A. Forsyth, *Convergence remedies for non-smooth payoffs in option pricing*, The Journal of Computational Finance **6** (2003), 25–40.
- [15] R. Rannacher, *Finite element solution of diffusion problems with irregular data*, Numer. Math. **43** (1984), no. 2, 309–327.
- [16] E. Reimer and M. Rubinstein, *Unscrambling the binary code*, Risk **4** (1991), 37–41.
- [17] R. Seydel, *Tools for Computational Finance*, Springer Verlag:Berlin, 2003.
- [18] D. Tavella and C. Randall, *Pricing Financial Instruments: The finite difference method*, John Wiley & Sons, 2000.
- [19] P. Wilmott, J. Dewynne, and S. Howison, *Option Pricing: Mathematical Models and Computation*, Oxford Financial Press, 1993.

HONGJOONG KIM
DEPARTMENT OF MATHEMATICS
KOREA UNIVERSITY
SEOUL 136-701, KOREA
E-mail address: `hongjoong@korea.ac.kr`

KYOUNG-SOOK MOON
DEPARTMENT OF MATHEMATICS & INFORMATION
KYUNGWON UNIVERSITY
GYEONGGI-DO 461-701, KOREA
E-mail address: `ksmoon@kyungwon.ac.kr`