

Article

Not peer-reviewed version

An Analytically Modified Finite Difference Scheme for Pricing Discretely Monitored Options

[Guo Luo](#)^{*} and [Min Huang](#)

Posted Date: 30 November 2024

doi: 10.20944/preprints202411.2426.v1

Keywords: Discrete option pricing; finite difference method; analytical modification; autocallable structured product; barrier option; snowball option



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

An Analytically Modified Finite Difference Scheme for Pricing Discretely Monitored Options

Guo Luo ^{1,*} and Min Huang ²

¹ Department of Mathematics, Statistics and Insurance, The Hang Seng University of Hong Kong, Hang Shin Link, Siu Lek Yuen, Shatin, N.T., Hong Kong

² China Merchants Bank, 7088 Shennan Boulevard, Shenzhen, Guangdong, China

* Correspondence: gluo@hsu.edu.hk

Abstract: Finite difference methods are commonly used in the pricing of discretely monitored exotic options in the Black-Scholes framework, but they tend to converge slowly due to discontinuities contained in terminal conditions. We present an effective analytical modification to existing finite difference methods which greatly enhances their performance on discretely monitored options with non-smooth terminal conditions. We apply this modification to the popular Crank-Nicolson method and obtain highly accurate option pricing results with significantly reduced CPU cost. We also introduce an adaptive mesh refinement technique which further improves the computational speed of the modified finite difference method. The proposed method is especially useful for options with high monitoring frequencies, which are difficult to price using other existing methods.

Keywords: discrete option pricing; finite difference method; analytical modification; autocallable structured product; barrier option; snowball option

1. Introduction

Discretely monitored exotic options, including barrier options and autocallable structured products, are commonly traded throughout the world. Unlike continuously monitored barrier options, discrete barrier options cannot be priced using analytical formulas. Instead, these options are priced using numerical techniques such as Monte-Carlo simulations, finite difference methods, quadrature methods [1,2], and several other sophisticated mathematical techniques (e.g. [3–7]). Discrete barrier options can also be priced approximately using continuity correction techniques [8,9], but the accuracy of these methods is often quite limited.

To price options with low to moderate monitoring frequencies (e.g. monthly), quadrature methods can be quite effective [2]. However, these methods are not suitable for pricing options with high monitoring frequencies (e.g. daily), since they involve computations of convolution integrals at all monitoring points (or barriers), which can be very expensive when monitoring frequencies become large. Monte-Carlo simulations suffer from the same limitation, where significantly more random numbers need to be generated along a rapidly increasing number of paths as monitoring frequencies increase. Finite difference methods, in contrast, are not subject to such limitations, as long as monitoring points are placed on the computational grid, but they tend to converge slowly due to discontinuities contained in terminal conditions. Performance of finite difference methods can be improved using advanced numerical techniques such as adaptive mesh models [10], Rannacher time-stepping [11,12], and the TR-BDF2 method [13], but these methods either are formulated in a different setting (e.g. for tree-based pricing models [10]), or suffer from relatively low accuracies due to the incorporation of low-order approximations near terminal/boundary points (e.g. [11,12]).

We present an analytical modification to finite difference methods in the Black-Scholes framework to remove any discontinuities contained in the terminal conditions, which greatly improves the accuracy and efficiency of these methods. We will apply the modification to the popular Crank-Nicolson method, and introduce an effective adaptive mesh refinement technique to further improve the computational speed of the modified method. The modified finite difference method is valuable in practice thanks to its high accuracy and ease of implementation. It also complements our previous quadrature method [2] since its performance is unaffected by higher monitoring frequencies.

1.1. A Brief Review of Products Considered

The modified finite difference method is applicable to various types of discretely monitored options, including discrete barrier options, autocallable structured products, and snowball options.

Barrier options are among the most popular types of exotic options. A barrier option may be activated (knock-in option) or deactivated (knock-out option) when the underlying asset price crosses certain barrier levels. A single barrier option has one barrier at each observation date, while a double barrier option has two barriers at each observation date. The final payoff of a barrier option (if it is active at maturity) may be of the same type as that of a vanilla option or that of a digital option.

For an autocallable product, there is a pre-specified barrier level at each observation date. If the underlying asset price is greater (less) than or equal to the barrier level, the option is exercised and a pre-specified fixed-rate return is paid. If the asset price is below (above) the barriers at all observation dates, the option is never exercised and the investor receives a negative return at maturity.

A snowball option has an up-and-out barrier and a down-and-in barrier, and the down-and-in barrier is usually monitored daily. If the asset price reaches the up-and-out barrier, the option is exercised and a pre-specified fixed-rate return is paid. If the option expires without any barrier being reached, the investor receives a fixed coupon payment. If the up-and-out barrier is never reached but the down-and-in barrier is reached sometime before maturity, the investor automatically writes a vanilla put and may receive a negative return.

1.2. Organization of the Paper

The rest of the paper is organized as follows. Section 2.1 defines the class of (discrete) option pricing problems that our method is intended to solve, and the rest of Section 2 details the various ingredients of our method including, specifically, the analytical modification (Section 2.3) we introduce for standard finite difference methods (Section 2.2) in the context of discrete option pricing, and an adaptive mesh refinement technique for the analytically modified finite difference methods (Section 2.4). The efficacy and efficiency of our method is then demonstrated through extensive numerical studies in Section 3, and the paper is concluded in Section 4.

2. Materials and Methods

2.1. Basic Assumptions

We consider a general class of discretely monitored options with barriers. Note that an option with knock-in barriers is equivalent to the difference of an option without those knock-in barriers and another option with knock-out barriers at the same dates and same levels. Therefore, it suffices to consider options with only knock-out barriers. To this end, assume that

- (A) The option to be priced has two strike prices $K_m^-, K_m^+ \in [0, \infty]$, with $K_m^- \leq K_m^+$, at each observation date T_m , $m = 1, 2, \dots, M$. The expiration date is T_M .
- (B) The option is exercised if $S \leq K_m^-$ or $S \geq K_m^+$ at some T_m , and the payoffs are given by $a_m^- S + b_m^-$ (if $S \leq K_m^-$) and $a_m^+ S + b_m^+$ (if $S \geq K_m^+$), respectively, for some $a_m^\pm, b_m^\pm \in \mathbb{R}$.
- (C) The final payoff at maturity is

$$a_M S + b_M, \quad \text{for} \quad K_M^- < S < K_M^+,$$

for some $a_M, b_M \in \mathbb{R}$.

These assumptions are general enough to cover a wide class of discretely monitored options, such as the ones mentioned in the introduction. For instance, down-and-out put barrier options would have

$$\begin{aligned} 1 \leq m \leq M-1: & \quad K_m^+ = \infty, 0 < K_m^- < \infty, a_m^- = 0, b_m^- = 0; \\ m = M: & \quad K_M^- = 0, 0 < K_M^+ < \infty, a_M = -1, b_M = K_M^+, \\ & \quad a_M^+ = b_M^+ = 0. \end{aligned}$$

Double barrier knock-out call options would have

$$\begin{aligned} 1 \leq m \leq M-1: & \quad 0 < K_m^\pm < \infty, a_m^\pm = 0, b_m^\pm = 0; \\ m = M: & \quad K_M^+ = \infty, 0 < K_M^- < \infty, a_M = 1, b_M = -K_M^-, \\ & \quad a_M^- = b_M^- = 0. \end{aligned}$$

Common up-and-out autocallable products would have

$$\begin{aligned} 1 \leq m \leq M: & \quad 0 < K_m^+ < \infty, K_m^- = 0, a_m^+ = 0, b_m^+ > 0; \\ m = M: & \quad a_M = 0, b_M < 0. \end{aligned}$$

We further assume that the price of the underlying asset $S(t)$ satisfies the following stochastic differential equation under the risk-neutral measure:

$$dS(t) = [r(t) - q(t)]S(t)dt + \sigma(t)S(t)dW(t), \quad (2.1)$$

where $r(t)$ is the risk-free forward interest rate, $q(t)$ the yield rate, $\sigma(t)$ the volatility, and $W(t)$ the Wiener process. Interest rates, yield rates, and volatilities are assumed to be deterministic functions of time.

To summarize, our basic assumptions are

1. There are finitely many observation points, and two exercise levels (possibly ∞) at each observation point. If S is above the upper exercise level or below the lower exercise level at any observation point, the option is exercised and the payoff is a linear function in S .
2. At maturity, if S is between the two exercise levels, a payoff is incurred which is also a linear function in S .
3. The underlying asset price S follows a geometric Brownian motion with possibly time-dependent interest rates, yield rates, and volatilities.

Given the above assumptions, we shall begin the description of our method with a brief review of the finite difference methods commonly used in the pricing of discretely monitored options.

2.2. Finite Difference Methods

Suppose we wish to find an option's value at some $T_0 < T_M$ and the spot price of the underlying asset is S_0 . Let $V(t, S)$ denote the value of the option (as a function of the asset price S) at any time $t \geq T_0$. Since S follows a geometric Brownian motion, the option's value $V(t, S)$ satisfies the Black-Scholes PDE

$$\frac{\partial V}{\partial t}(t, S) + [r(t) - q(t)]S \frac{\partial V}{\partial S}(t, S) + \frac{\sigma^2(t)}{2} S^2 \frac{\partial^2 V}{\partial S^2}(t, S) = r(t)V(t, S), \quad (2.2a)$$

for $T_0 \leq t \leq T_M$, with the terminal condition

$$V(T_M, S) = \begin{cases} a_M S + b_M, & K_M^- < S < K_M^+ \\ a_M^- S + b_M^-, & S \leq K_M^- \\ a_M^+ S + b_M^+, & S \geq K_M^+ \end{cases}, \quad (2.2b)$$

and boundary conditions

$$V(T_m, S) = \begin{cases} a_m^- S + b_m^-, & S \leq K_m^- \\ a_m^+ S + b_m^+, & S \geq K_m^+ \end{cases}, \quad 1 \leq m \leq M-1, \quad (2.2c)$$

determined by Assumptions (B)–(C) from Section 2.1.

For convenience, we introduce the change of variable $Z = \ln S$ to transform equation (2.2a) into a simpler form:

$$\frac{\partial V}{\partial t}(t, e^Z) = \mathcal{L}[V(t, e^Z)], \quad (2.3a)$$

where the differential operator \mathcal{L} is defined as

$$\mathcal{L} := -\frac{\sigma^2(t)}{2} \frac{\partial^2}{\partial Z^2} - \rho(t) \frac{\partial}{\partial Z} + r(t), \quad \text{with} \quad \rho(t) := r(t) - q(t) - \frac{\sigma^2(t)}{2}. \quad (2.3b)$$

It is not convenient to solve the PDE (2.3a) directly due to the discrete boundary conditions (2.2c). Instead, for each $1 \leq m \leq M$, we consider the PDE

$$\frac{\partial Y_m}{\partial t}(t, Z) = \mathcal{L}Y_m(t, Z), \quad T_{m-1} \leq t \leq T_m, \quad (2.4a)$$

supplemented with the terminal condition

$$Y_m(T_m, Z) = V(T_m, e^Z). \quad (2.4b)$$

Since the terminal value problem (2.4) has a unique solution for each m , we have

$$V(t, e^Z) = Y_m(t, Z), \quad T_{m-1} < t \leq T_m,$$

and by (2.2c),

$$V(T_{m-1}, e^Z) = \begin{cases} Y_m(T_{m-1}, Z), & \ln(K_{m-1}^-) < Z < \ln(K_{m-1}^+) \\ a_{m-1}^- e^Z + b_{m-1}^-, & Z \leq \ln(K_{m-1}^-) \\ a_{m-1}^+ e^Z + b_{m-1}^+, & Z \geq \ln(K_{m-1}^+) \end{cases}, \quad 1 \leq m \leq M. \quad (2.5)$$

The terminal value problems (2.4) are solved successively for $m = M, M-1, \dots, 1$. Note that for $m = M$, the terminal condition $Y_M(T_M, Z) = V(T_M, e^Z)$ is known by (2.2b). Now for each $1 \leq m \leq M$, assume the terminal condition $Y_m(T_m, Z)$ is known. The PDE (2.4a) is solved backward in t to yield $Y_m(T_{m-1}, Z)$. Then we obtain $V(T_{m-1}, e^Z)$ by (2.5), which by (2.4b) is equal to $Y_{m-1}(T_{m-1}, Z)$, the terminal condition of the PDE (2.4a) for $m-1$. This process can then be repeated until we finally obtain $V(T_0, e^Z)$, from which the option's value $V(T_0, S_0)$ can be easily deduced.

Now we give a brief review of the standard procedure of solving (2.4) numerically on a uniform grid using finite difference methods. Let t_0, t_1, \dots, t_N denote the grid points in time, and Z_0, Z_1, \dots, Z_{2P} the grid points in space where $Z_P = \ln(S_0)$. Denote

$$Y_{l,j}^m = Y_m(t_l, Z_j), \quad r_l = r(t_l), \quad \rho_l = \rho(t_l), \quad \sigma_l = \sigma(t_l), \\ k = \frac{t_N - t_0}{N}, \quad h = \frac{Z_{2P} - Z_0}{2P}, \quad \lambda = \frac{k}{h}, \quad \mu = \frac{k}{h^2}.$$

In practice, barriers are observed at the end of a business day, so it is convenient to find a uniform t -grid such that every monitoring point lies on the grid, i.e. for every $1 \leq m \leq M$, there exists an l_m such that $t_{l_m} = T_m$. The PDE (2.4a) can be discretized using the backward-time centered-space (BTCS), forward-time centered-space (FTCS), or Crank-Nicolson (C-N) finite difference method.

The BTCS (or explicit Euler) method consists of the recurrence equations

$$Y_{l,j}^m - Y_{l-1,j}^m + \frac{1}{2} \lambda \rho_l (Y_{l,j+1}^m - Y_{l,j-1}^m) \\ + \frac{1}{2} \mu \sigma_l^2 (Y_{l,j+1}^m - 2Y_{l,j}^m + Y_{l,j-1}^m) = k r_l Y_{l,j}^m, \quad (2.6)$$

and is explicit (recall that we are solving (2.4a) backward in time). The FTCS (or implicit Euler) method, on the other hand, consists of the recurrence equations

$$\begin{aligned} Y_{l,j}^m - Y_{l-1,j}^m + \frac{1}{2}\lambda\rho_{l-1}(Y_{l-1,j+1}^m - Y_{l-1,j-1}^m) \\ + \frac{1}{2}\mu\sigma_{l-1}^2(Y_{l-1,j+1}^m - 2Y_{l-1,j}^m + Y_{l-1,j-1}^m) = kr_{l-1}Y_{l-1,j}^m, \end{aligned} \quad (2.7)$$

and is implicit. The C-N method is a combination of the BTCS and FTCS methods and can be written as

$$\begin{aligned} (\frac{1}{4}\mu\sigma_{l-1}^2 - \frac{1}{4}\lambda\rho_{l-1})Y_{l-1,j-1}^m + (-1 - \frac{1}{2}\mu\sigma_{l-1}^2 - \frac{1}{2}kr_{l-1})Y_{l-1,j}^m \\ + (\frac{1}{4}\mu\sigma_{l-1}^2 + \frac{1}{4}\lambda\rho_{l-1})Y_{l-1,j+1}^m = -(\frac{1}{4}\mu\sigma_l^2 - \frac{1}{4}\lambda\rho_l)Y_{l,j-1}^m \\ + (-1 + \frac{1}{2}\mu\sigma_l^2 + \frac{1}{2}kr_l)Y_{l,j}^m - (\frac{1}{4}\mu\sigma_l^2 + \frac{1}{4}\lambda\rho_l)Y_{l,j+1}^m. \end{aligned} \quad (2.8)$$

For $j \in \{0, 2P\}$, we impose the extrapolation boundary condition¹

$$\frac{\partial^2 V}{\partial S^2}(t, S) = 0, \quad (2.9a)$$

which is equivalent to

$$\frac{\partial^2 Y_m}{\partial Z^2}(t, Z) - \frac{\partial Y_m}{\partial Z}(t, Z) = 0, \quad (2.9b)$$

and can be discretized as

$$(1 - \frac{1}{2}h)Y_{i,j+1}^m - 2Y_{i,j}^m + (1 + \frac{1}{2}h)Y_{i,j-1}^m = 0, \quad i = l-1, l, \quad j = 0, 2P. \quad (2.9c)$$

The C-N method is generally more accurate than the BTCS and FTCS methods [14], although the FTCS method has better stability properties [13].

For convenience, we denote

$$\begin{aligned} \alpha_l &= \frac{1}{4}\mu\sigma_l^2 - \frac{1}{4}\lambda\rho_l, & \beta_l &= -1 - \frac{1}{2}\mu\sigma_l^2 - \frac{1}{2}kr_l, \\ \gamma_l &= \frac{1}{4}\mu\sigma_l^2 + \frac{1}{4}\lambda\rho_l, & \delta_l &= -1 + \frac{1}{2}\mu\sigma_l^2 + \frac{1}{2}kr_l. \end{aligned}$$

Equations (2.8) and (2.9c) can be written together as a matrix equation

$$\mathcal{A}_{l-1}\mathbf{Y}_{l-1}^m = \mathcal{B}_l\mathbf{Y}_l^m, \quad (2.10a)$$

where

$$\mathbf{Y}_l^m = \langle Y_{l,0}^m, Y_{l,1}^m, \dots, Y_{l,2P}^m \rangle^T,$$

¹ Note, in view of the linear boundary condition (2.2c), that the extrapolation boundary condition (2.9a) is a natural choice of the numerical boundary condition at $j = 0, 2P$.

is a $(2P + 1)$ -column vector and $\mathcal{A}_l, \mathcal{B}_l$ are $(2P + 1) \times (2P + 1)$ tridiagonal matrices:

$$\mathcal{A}_l = \begin{pmatrix} \beta_l + c_{1,+}\alpha_l & \gamma_l - c_{2,+}\alpha_l & & & \\ \alpha_l & \beta_l & \gamma_l & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha_l & \beta_l & \gamma_l \\ & & & \alpha_l - c_{2,-}\gamma_l & \beta_l + c_{1,-}\gamma_l \end{pmatrix}, \quad (2.10b)$$

$$\mathcal{B}_l = \begin{pmatrix} \delta_l - c_{1,+}\alpha_l & -\gamma_l + c_{2,+}\alpha_l & & & \\ -\alpha_l & \delta_l & -\gamma_l & & \\ & \ddots & \ddots & \ddots & \\ & & -\alpha_l & \delta_l & -\gamma_l \\ & & & -\alpha_l + c_{2,-}\gamma_l & \delta_l - c_{1,-}\gamma_l \end{pmatrix}, \quad (2.10c)$$

where

$$c_{1,\pm} = \frac{2}{1 \pm \frac{1}{2}h}, \quad c_{2,\pm} = \frac{1 \mp \frac{1}{2}h}{1 \pm \frac{1}{2}h}.$$

Note that the unknowns Y_{i-1}^m in the $j = 0$ equation and $Y_{i,2P+1}^m$ in the $j = 2P$ equation for $i = l - 1, l$ can be eliminated using the boundary condition (2.9c). Equation (2.10a) can be solved for \mathbf{Y}_{l-1}^m using the Thomas Algorithm, which has an $O(P)$ complexity.

Note that the terminal condition (2.4b) is typically discontinuous as can be seen from (2.5). This means that the C-N method, when applied as it is, cannot achieve a second-order convergence as for smooth terminal conditions [11].

2.3. Analytical Modification

We will remove the aforementioned discontinuities from terminal conditions by introducing an analytical modification to existing finite difference methods. First, we recall some classical results from the theory of binary options.

Lemma 2.1. Let $K > 0$, and let χ_A denote the characteristic function of a set A . Consider a binary option with expiration time T_m priced at T_{m-1} .

1. If the option has payoff $\hat{V}_m(y) = \chi_{[K,\infty)}y$, then $\hat{V}_{m-1}(S) = \phi_m^a(S, K, 1)$.
2. If $\hat{V}_m(y) = \chi_{(0,K]}y$, then $\hat{V}_{m-1}(S) = \phi_m^a(S, K, -1)$.
3. If $\hat{V}_m(y) = \chi_{[K,\infty)}$, then $\hat{V}_{m-1}(S) = \phi_m^b(S, K, 1)$.
4. If $\hat{V}_m(y) = \chi_{(0,K]}$, then $\hat{V}_{m-1}(S) = \phi_m^b(S, K, -1)$.

The functions ϕ_m^a and ϕ_m^b are defined as

$$\phi_m^a(S, K, \epsilon) = e^{-q_m \tau_m} \mathcal{N}(\epsilon d_1) S, \quad \phi_m^b(S, K, \epsilon) = e^{-r_m \tau_m} \mathcal{N}(\epsilon d_2),$$

where $\tau_m = T_m - T_{m-1}$,

$$r_m = \int_{T_{m-1}}^{T_m} \frac{r(t)}{\tau_m} dt, \quad q_m = \int_{T_{m-1}}^{T_m} \frac{q(t)}{\tau_m} dt, \quad \sigma_m^2 = \int_{T_{m-1}}^{T_m} \frac{\sigma^2(t)}{\tau_m} dt,$$

\mathcal{N} is the cumulative normal distribution function, and

$$d_1 = \frac{1}{\sigma_m \sqrt{\tau_m}} \left\{ \log \frac{S}{K} + \left(r_m - q_m + \frac{\sigma_m^2}{2} \right) \tau_m \right\}, \quad d_2 = d_1 - \sigma_m \sqrt{\tau_m}.$$

Proof. By definition, ϕ_m^a is the value of an asset-or-nothing option, and ϕ_m^b is the value of a cash-or-nothing option. The valuation formulas are just standard results for binary options [14]. \square

Since $Y_m(T_m, Z)$ in the terminal condition (2.4b) generally has discontinuities at $\ln(K_m^\pm)$, we do not solve the terminal value problem (2.4) directly, but instead seek to remove these discontinuities first. We describe our basic idea as follows.

2.3.1. Basic idea of the analytical modification

For each $1 \leq m \leq M$, note that $V(T_m, S)$ is continuous for $K_m^- < S < K_m^+$. We define the one-sided limits

$$\tilde{V}_m^+ = \lim_{\epsilon \rightarrow 0^+} V(T_m, K_m^+ - \epsilon), \quad \tilde{V}_m^- = \lim_{\epsilon \rightarrow 0^+} V(T_m, K_m^- + \epsilon), \quad (2.11a)$$

$$\tilde{V}_{S,m}^+ = \lim_{\epsilon \rightarrow 0^+} \frac{\partial V}{\partial S}(T_m, K_m^+ - \epsilon), \quad \tilde{V}_{S,m}^- = \lim_{\epsilon \rightarrow 0^+} \frac{\partial V}{\partial S}(T_m, K_m^- + \epsilon), \quad (2.11b)$$

and consider $U_m(t, Z)$ defined as the solution to the PDE

$$\frac{\partial U_m}{\partial t}(t, Z) = \mathcal{L}U_m(t, Z), \quad T_{m-1} \leq t \leq T_m, \quad (2.12a)$$

supplemented with the (continuously differentiable) terminal condition

$$U_m(T_m, Z) = \begin{cases} V(T_m, e^Z), & \ln(K_m^-) < Z < \ln(K_m^+) \\ \tilde{V}_{S,m}^-(e^Z - K_m^-) + \tilde{V}_m^-, & Z \leq \ln(K_m^-) \\ \tilde{V}_{S,m}^+(e^Z - K_m^+) + \tilde{V}_m^+, & Z \geq \ln(K_m^+) \end{cases}. \quad (2.12b)$$

Note that by (2.2b)-(2.2c), we have

$$V(T_m, e^Z) - U_m(T_m, Z) = \begin{cases} 0, & \ln(K_m^-) < Z < \ln(K_m^+) \\ \tilde{a}_m^- e^Z + \tilde{b}_m^-, & Z \leq \ln(K_m^-) \\ \tilde{a}_m^+ e^Z + \tilde{b}_m^+, & Z \geq \ln(K_m^+) \end{cases}, \quad (2.13)$$

where

$$\tilde{a}_m^\pm = a_m^\pm - \tilde{V}_{S,m}^\pm, \quad \tilde{b}_m^\pm = b_m^\pm + \tilde{V}_{S,m}^\pm K_m^\pm - \tilde{V}_m^\pm.$$

This shows that $V(t, e^Z) - U_m(t, Z)$ is the sum of values of four binary options for $T_{m-1} \leq t \leq T_m$, and hence by Lemma 2.1,

$$\begin{aligned} V(T_{m-1}, e^Z) - U_m(T_{m-1}, Z) &= \tilde{a}_m^- \phi_m^a(e^Z, K_m^-, -1) + \tilde{b}_m^- \phi_m^b(e^Z, K_m^-, -1) \\ &+ \tilde{a}_m^+ \phi_m^a(e^Z, K_m^+, 1) + \tilde{b}_m^+ \phi_m^b(e^Z, K_m^+, 1), \quad \ln(K_{m-1}^-) < Z < \ln(K_{m-1}^+), \end{aligned} \quad (2.14a)$$

or

$$\begin{aligned} V(T_{m-1}, e^Z) &= U_m(T_{m-1}, Z) + \tilde{a}_m^- \phi_m^a(e^Z, K_m^-, -1) + \tilde{b}_m^- \phi_m^b(e^Z, K_m^-, -1) \\ &+ \tilde{a}_m^+ \phi_m^a(e^Z, K_m^+, 1) + \tilde{b}_m^+ \phi_m^b(e^Z, K_m^+, 1), \quad \ln(K_{m-1}^-) < Z < \ln(K_{m-1}^+). \end{aligned} \quad (2.14b)$$

For each $1 \leq m \leq M$, we solve (2.12) numerically for $U_m(T_{m-1}, Z)$ and obtain $V(T_{m-1}, e^Z)$ using (2.14b). Repeating this process for $m = M, M-1, \dots, 1$, we finally obtain $V(T_0, e^Z)$, from which the option's value $V(T_0, S_0)$ can be easily deduced.

Remark 2.2. For the given terminal condition (2.2b), a closed-form expression can in fact be obtained for $U_M(t, Z)$; see, for example, [2, Proposition 2.4]. In practice, however, M is often “reasonably large”, e.g., in the order of tens or hundreds, so that the incorporation of such exact formulas of U_M makes little difference in the numerical approximations obtained for $V(T_0, S)$. Thus we do not specifically differentiate U_M from other U_m 's ($m \leq M-1$) in our study.

2.3.2. Implementation of the Analytical Modification

In practice, it may not be possible to place all barriers on the computational grid, which means we may not be able to compute the values of $\tilde{V}_m^\pm, \tilde{V}_{S,m}^\pm$ for all $1 \leq m \leq M-1$ as defined in (2.11). Note that the values of $\tilde{V}_M^\pm, \tilde{V}_{S,M}^\pm$ for $m = M$ follow from the closed-form expression of $V(T_M, e^Z)$ as given in (2.2b). For $1 \leq m \leq M-1$, however, we will have to find an approximation to these values. To this end, denote

$$\tilde{U}_M(t, Z) = U_M(t, Z), \quad (2.15)$$

which is the smooth solution defined in (2.12) for $m = M$. For each $1 \leq m \leq M-1$, assume we have already computed a three times continuously differentiable (C^3) function $\tilde{U}_{m+1}(T_m, Z)$ for $Z \in \{Z_0, Z_1, \dots, Z_{2P}\}$ and have obtained the smooth part of $V(T_m, e^Z)$, namely $Y_{m+1}(T_m, Z)$ (see (2.5)), at the grid points. Note that this is true for $m = M-1$, in which case we have already computed a C^3 function $\tilde{U}_M(T_{M-1}, Z)$ by (2.15) and have obtained $Y_M(T_{M-1}, Z)$ by (2.14b).

Denote

$$Y_{l,j}^{m+1} = Y_{m+1}(t_l, Z_j), \quad Y_{S,l,j}^{m+1} = \frac{Y_{l,j+1}^{m+1} - Y_{l,j-1}^{m+1}}{2he^{Z_j}} \approx \frac{\partial Y_{m+1}}{\partial S}(t_l, Z_j),$$

and let

$$p_m^- = \min\{j > 0: Z_j > \ln(K_m^-)\}, \quad p_m^+ = \max\{j < 2P: Z_j < \ln(K_m^+)\}.$$

Note that $\ln(K_m^\pm)$ are located between $Z_{p_m^\pm}$ and $Z_{p_m^\pm \pm 1}$. We use linear interpolations of $Y_{m+1}(T_m, Z)$ on the grid to approximate $\tilde{V}_m^\pm, \tilde{V}_{S,m}^\pm$ as follows:

$$c_m^- = Y_{S,l_m,p_m^-}^{m+1} - \frac{1}{h}(Y_{S,l_m,p_m^-}^{m+1} - Y_{S,l_m,p_m^-}^{m+1})[\ln(K_m^-) - Z_{p_m^-}], \quad (2.16a)$$

$$c_m^+ = Y_{S,l_m,p_m^+}^{m+1} + \frac{1}{h}(Y_{S,l_m,p_m^+}^{m+1} - Y_{S,l_m,p_m^+}^{m+1})[\ln(K_m^+) - Z_{p_m^+}], \quad (2.16b)$$

$$d_m^- = Y_{l_m,p_m^-}^{m+1} - \frac{1}{h}(Y_{l_m,p_m^-}^{m+1} - Y_{l_m,p_m^-}^{m+1})[\ln(K_m^-) - Z_{p_m^-}], \quad (2.16c)$$

$$d_m^+ = Y_{l_m,p_m^+}^{m+1} + \frac{1}{h}(Y_{l_m,p_m^+}^{m+1} - Y_{l_m,p_m^+}^{m+1})[\ln(K_m^+) - Z_{p_m^+}]. \quad (2.16d)$$

Since $\tilde{U}_{m+1}(T_m, Z)$ and hence $Y_{m+1}(T_m, Z)$ is C^3 in Z , we have

$$c_m^\pm - \tilde{V}_{S,m}^\pm = O(h^2), \quad d_m^\pm - \tilde{V}_m^\pm = O(h^2). \quad (2.17)$$

This means that the order of the interpolation error does not exceed that of the C-N method.

Consider $\tilde{U}_m(t, Z)$ defined as the solution to the PDE

$$\frac{\partial \tilde{U}_m}{\partial t}(t, Z) = \mathcal{L}\tilde{U}_m(t, Z), \quad T_{m-1} \leq t \leq T_m, \quad (2.18a)$$

supplemented with the terminal condition

$$\tilde{U}_m(T_m, Z) = \begin{cases} V(T_m, e^Z), & \ln(K_m^-) < Z < \ln(K_m^+) \\ c_m^-(e^Z - K_m^-) + d_m^-, & Z \leq \ln(K_m^-) \\ c_m^+(e^Z - K_m^+) + d_m^+, & Z \geq \ln(K_m^+) \end{cases}. \quad (2.18b)$$

Note that by (2.2c), we have

$$V(T_m, e^Z) - \tilde{U}_m(T_m, Z) = \begin{cases} 0, & \ln(K_m^-) < Z < \ln(K_m^+) \\ \tilde{c}_m^- e^Z + \tilde{d}_m^-, & Z \leq \ln(K_m^-) \\ \tilde{c}_m^+ e^Z + \tilde{d}_m^+, & Z \geq \ln(K_m^+) \end{cases}, \quad (2.19)$$

where

$$\tilde{c}_m^\pm = a_m^\pm - c_m^\pm, \quad \tilde{d}_m^\pm = b_m^\pm + c_m^\pm K_m^\pm - d_m^\pm.$$

This shows that $V(t, e^Z) - \tilde{U}_m(t, Z)$ is the sum of values of four binary options for $T_{m-1} \leq t \leq T_m$, and hence by Lemma 2.1,

$$\begin{aligned} V(T_{m-1}, e^Z) - \tilde{U}_m(T_{m-1}, Z) &= \tilde{c}_m^- \phi_m^a(e^Z, K_m^-, -1) + \tilde{d}_m^- \phi_m^b(e^Z, K_m^-, -1) \\ &\quad + \tilde{c}_m^+ \phi_m^a(e^Z, K_m^+, 1) + \tilde{d}_m^+ \phi_m^b(e^Z, K_m^+, 1), \quad \ln(K_{m-1}^-) < Z < \ln(K_{m-1}^+), \end{aligned} \quad (2.20a)$$

or

$$\begin{aligned} V(T_{m-1}, e^Z) &= \tilde{U}_m(T_{m-1}, Z) + \tilde{c}_m^- \phi_m^a(e^Z, K_m^-, -1) + \tilde{d}_m^- \phi_m^b(e^Z, K_m^-, -1) \\ &\quad + \tilde{c}_m^+ \phi_m^a(e^Z, K_m^+, 1) + \tilde{d}_m^+ \phi_m^b(e^Z, K_m^+, 1), \quad \ln(K_{m-1}^-) < Z < \ln(K_{m-1}^+). \end{aligned} \quad (2.20b)$$

Now that we have computed $\tilde{U}_m(T_{m-1}, Z)$ and the smooth part of $V(T_{m-1}, e^Z)$, namely $Y_m(T_{m-1}, Z)$, we continue to compute c_{m-1}^\pm, d_{m-1}^\pm as in (2.16), which means we have the terminal condition $\tilde{U}_{m-1}(T_{m-1}, Z)$ as in (2.18b). Then we repeat this process for $m - 1$.

We summarize the modified finite difference method as follows:

- 1: **for** $m = M - 1$ **downto** 1 **do**
- 2: Compute c_m^\pm, d_m^\pm using (2.16)
- 3: Solve (2.18) numerically for $\tilde{U}_m(T_{m-1}, Z)$
- 4: Compute $Y_m(T_{m-1}, Z)$, the smooth part of $V(T_{m-1}, e^Z)$, using (2.20b)
- 5: **end for**
- 6: Obtain option value $V(T_0, S_0) = V(T_0, e^{Z_P})$

Note that the terminal value problem (2.18) is the same as (2.4) except that the terminal condition (2.18b) now has a higher degree of smoothness, and the problem (2.18) can be solved numerically using the same finite difference methods as the ones described in Section 2.2, in particular the C-N method. Since the terminal condition $\tilde{U}_m(T_m, Z)$ in (2.18b) is almost continuously differentiable (the jumps are of order h^2 by (2.17)), the C-N method converges quite well in practice (Section 3.1). In principle, Rannacher time-stepping can be used to improve the performance of the C-N method near monitoring points, but this is important only when accurate discrete approximations to option Greeks are needed (Section 3.4).

2.4. Adaptive Mesh Refinement

Note that the function $\tilde{U}_m(t, Z)$ in (2.18) tends to have large derivatives near the barriers and behaves relatively smoothly away from the barriers. This means that it makes sense to use finer grids near the barriers and coarser ones elsewhere.

Adaptive mesh refinement techniques are well-known and are widely used in numerical solutions of PDEs. In the field of quantitative finance, adaptive mesh models have been developed for the BTCS method [10]. Alternatively, nonuniform grids based on changes of variables have also been introduced in the literature [13]. Since here we are using an analytically modified C-N method, we introduce an effective adaptive mesh refinement technique which further improves the computational efficiency of our method.

We begin by assuming that the upper barriers K_m^+ (and lower barriers K_m^- , respectively) are equal or close to each other for all $1 \leq m \leq M$, which is almost always satisfied in practice. Our basic idea is to construct fine grids near the barriers, and solve the Black-Scholes PDEs on both the original and the fine grids at each time step.

First, we find integers $0 < q_1^- < q_2^- < q_1^+ < q_2^+ < 2P$ such that

$$\exp(Z_{q_1^-}) < K_m^- < \exp(Z_{q_2^-}), \quad \text{and} \quad \exp(Z_{q_1^+}) < K_m^+ < \exp(Z_{q_2^+}),$$

for all $1 \leq m \leq M$. Then we construct a fine grid for $[q_1^-, q_2^-]$ and a similar one for $[q_1^+, q_2^+]$ by dividing each interval of size h into n equal parts. We illustrate the fine grid near K_m^+ for $n = 4$ in Figure 1.

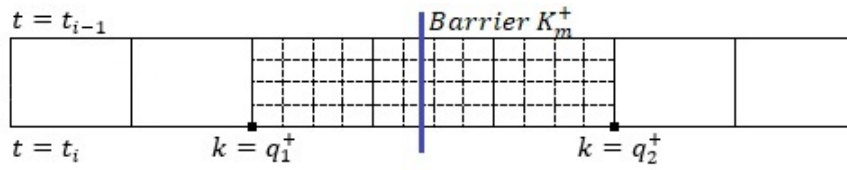


Figure 1. Illustration of the original grid (solid lines) and the fine grid (dashed lines) near the barrier K_m^+ (thick line).

Let

$$n^\pm = n(q_2^\pm - q_1^\pm), \quad x_j^\pm = Z_{q_1^\pm} + jh/n, \quad 0 \leq j \leq n^\pm,$$

and define

$$\hat{U}_M(t, Z) = U_M(t, Z), \quad (2.21)$$

which is the smooth solution defined in (2.12) for $m = M$. For each $1 \leq m \leq M - 1$, assume we have already computed a C^3 function $\hat{U}_{m+1}(T_m, Z)$ for

$$Z \in \mathcal{G}_h := \{Z_0, Z_1, \dots, Z_{2P}\} \cup \{x_0^\pm, x_1^\pm, \dots, x_{n^\pm}^\pm\},$$

and have obtained the smooth part of $V(T_m, e^Z)$, namely $Y_{m+1}(T_m, Z)$, at the grid points. Note that this is true for $m = M - 1$ by (2.21) and (2.14b).

Recall from (2.16) that c_m^\pm, d_m^\pm are approximations to $\tilde{V}_{S,m}^\pm, \tilde{V}_m^\pm$ computed using the values of $Y_{m+1}(T_m, Z)$ on the original (coarse) grid. We shall now find approximations to $\tilde{V}_{S,m}^\pm, \tilde{V}_m^\pm$ using the values of $Y_{m+1}(T_m, Z)$ on the fine grids. To this end, denote

$$\begin{aligned} Y_{l,j}^{\pm(m+1)} &= Y_{m+1}(t_l, x_j^\pm), \\ Y_{S,l,j}^{\pm(m+1)} &= \frac{Y_{l,j+1}^{\pm(m+1)} - Y_{l,j-1}^{\pm(m+1)}}{2he^{x_j^\pm}/n} \approx \frac{\partial Y_{m+1}}{\partial S}(t_l, x_j^\pm), \end{aligned}$$

and let

$$u_m^- = \min\{j > 0: x_j^- > \ln(K_m^-)\}, \quad u_m^+ = \max\{j < n^+: x_j^+ < \ln(K_m^+)\}.$$

Define

$$f_m^- = Y_{S,l_m,u_m^-}^{-(m+1)} - \frac{n}{h} [Y_{S,l_m,u_m^- - 1}^{-(m+1)} - Y_{S,l_m,u_m^-}^{-(m+1)}] [\ln(K_m^-) - x_{u_m^-}^-], \quad (2.22a)$$

$$f_m^+ = Y_{S,l_m,u_m^+}^{+(m+1)} + \frac{n}{h} [Y_{S,l_m,u_m^+ + 1}^{+(m+1)} - Y_{S,l_m,u_m^+}^{+(m+1)}] [\ln(K_m^+) - x_{u_m^+}^+], \quad (2.22b)$$

$$g_m^- = Y_{l_m,u_m^-}^{-(m+1)} - \frac{n}{h} [Y_{l_m,u_m^- - 1}^{-(m+1)} - Y_{l_m,u_m^-}^{-(m+1)}] [\ln(K_m^-) - x_{u_m^-}^-], \quad (2.22c)$$

$$g_m^+ = Y_{l_m,u_m^+}^{+(m+1)} + \frac{n}{h} [Y_{l_m,u_m^+ + 1}^{+(m+1)} - Y_{l_m,u_m^+}^{+(m+1)}] [\ln(K_m^+) - x_{u_m^+}^+]. \quad (2.22d)$$

Note that, in general, f_m^\pm, g_m^\pm provide a better approximation to $\tilde{V}_{S,m}^\pm, \tilde{V}_m^\pm$ than c_m^\pm, d_m^\pm .

Consider $\hat{U}_m(t, Z)$ defined as the solution to the PDE

$$\frac{\partial \hat{U}_m}{\partial t}(t, Z) = \mathcal{L}\hat{U}_m(t, Z), \quad T_{m-1} \leq t \leq T_m, \quad (2.23a)$$

supplemented with the terminal condition

$$\hat{U}_m(T_m, Z) = \begin{cases} V(T_m, e^Z), & \ln(K_m^-) < Z < \ln(K_m^+) \\ f_m^-(e^Z - K_m^-) + g_m^-, & Z \leq \ln(K_m^-) \\ f_m^+(e^Z - K_m^+) + g_m^+, & Z \geq \ln(K_m^+) \end{cases}. \quad (2.23b)$$

The PDE (2.23a) is the same as (2.18a), but we do not solve it all the way back to T_{m-1} . Instead, for each $l_{m-1} < l \leq l_m$, assume we have computed $\hat{U}_m(t_l, Z)$ for $Z \in \mathcal{G}_l$, which is true for $l = l_m$ by (2.23b). We first solve the PDE (2.23a) numerically using the terminal condition $\hat{U}_m(t_l, Z_k)$ on the coarse grid for just one time step from t_l to t_{l-1} . More specifically, denote by $\hat{U}_{l,k}^m = \hat{U}_m(t_l, Z_k)$ the solution computed on the coarse grid; we solve the matrix equation (2.10a) which in our new notation becomes

$$\mathcal{A}_{l-1} \hat{\mathbf{U}}_{l-1}^m = \mathcal{B}_l \hat{\mathbf{U}}_l^m, \quad (2.24)$$

where

$$\hat{\mathbf{U}}_l^m = \langle \hat{U}_{l,0}^m, \hat{U}_{l,1}^m, \dots, \hat{U}_{l,2P}^m \rangle^T,$$

is a $(2P+1)$ -column vector and $\mathcal{A}_l, \mathcal{B}_l$ are $(2P+1) \times (2P+1)$ tridiagonal matrices as defined in (2.10). After solving for $\hat{\mathbf{U}}_{l-1}^m$ from (2.24), we solve the PDE (2.23a) again on the fine grids for n time steps with reduced step size k/n , using the boundary conditions linearly interpolated (in time) from $\hat{\mathbf{U}}_{l-1}^m, \hat{\mathbf{U}}_l^m$ at q_1^\pm and q_2^\pm . For convenience, we denote, for $0 \leq i \leq n$,

$$\begin{aligned} r_{l,i} &= r(t_l + ik/n), \quad \rho_{l,i} = \rho(t_l + ik/n), \quad \sigma_{l,i} = \sigma(t_l + ik/n), \\ \alpha_{n,l,i} &= \frac{1}{4}n\mu\sigma_{l,i}^2 - \frac{1}{4}\lambda\rho_{l,i}, \quad \beta_{n,l,i} = -1 - \frac{1}{2}n\mu\sigma_{l,i}^2 - \frac{1}{2n}kr_{l,i}, \\ \gamma_{n,l,i} &= \frac{1}{4}n\mu\sigma_{l,i}^2 + \frac{1}{4}\lambda\rho_{l,i}, \quad \delta_{n,l,i} = -1 + \frac{1}{2}n\mu\sigma_{l,i}^2 + \frac{1}{2n}kr_{l,i}, \end{aligned}$$

and $W_{l,i,j}^{\pm m} = \hat{U}_m(t_{l,i}, x_j^\pm)$. The matrix equation on the fine grids then becomes

$$\mathcal{A}_{n,l-1,i-1} \mathbf{W}_{l-1,i-1}^{\pm m} = \mathcal{B}_{n,l-1,i} \mathbf{W}_{l-1,i}^{\pm m} + \mathbf{w}_{l-1,i}^\pm, \quad n \geq i \geq 1, \quad (2.25a)$$

where

$$\begin{aligned} \mathbf{W}_{l-1,i}^{\pm m} &= \langle W_{l-1,i,1}^{\pm m}, W_{l-1,i,2}^{\pm m}, \dots, W_{l-1,i,n^\pm-1}^{\pm m} \rangle^T, \\ \mathbf{w}_{l-1,i}^\pm &= \begin{pmatrix} -\alpha_{n,l-1,i} W_{l-1,i,0}^{\pm m} - \alpha_{n,l-1,i-1} W_{l-1,i-1,0}^{\pm m} \\ 0 \\ \vdots \\ 0 \\ -\gamma_{n,l-1,i} W_{l-1,i,n^\pm}^{\pm m} - \gamma_{n,l-1,i-1} W_{l-1,i-1,n^\pm}^{\pm m} \end{pmatrix}, \end{aligned} \quad (2.25b)$$

are $(n^\pm - 1)$ -column vectors and $\mathcal{A}_{n,l-1,i}$, $\mathcal{B}_{n,l-1,i}$ are $(n^\pm - 1) \times (n^\pm - 1)$ tridiagonal Toeplitz matrices:

$$\mathcal{A}_{n,l-1,i} = \begin{pmatrix} \beta_{n,l-1,i} & \gamma_{n,l-1,i} & & & \\ \alpha_{n,l-1,i} & \beta_{n,l-1,i} & \gamma_{n,l-1,i} & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha_{n,l-1,i} & \beta_{n,l-1,i} & \gamma_{n,l-1,i} \\ & & & \alpha_{n,l-1,i} & \beta_{n,l-1,i} \end{pmatrix}, \quad (2.25c)$$

$$\mathcal{B}_{n,l-1,i} = \begin{pmatrix} \delta_{n,l-1,i} & -\gamma_{n,l-1,i} & & & \\ -\alpha_{n,l-1,i} & \delta_{n,l-1,i} & -\gamma_{n,l-1,i} & & \\ & \ddots & \ddots & \ddots & \\ & & -\alpha_{n,l-1,i} & \delta_{n,l-1,i} & -\gamma_{n,l-1,i} \\ & & & -\alpha_{n,l-1,i} & \delta_{n,l-1,i} \end{pmatrix}. \quad (2.25d)$$

Note that, in (2.25b),

$$W_{l-1,i,0}^{\pm m} = \hat{U}_{l-1,q_1^\pm}^m + \frac{i}{n}(\hat{U}_{l,q_1^\pm}^m - \hat{U}_{l-1,q_1^\pm}^m), \quad (2.26a)$$

$$W_{l-1,i,n^\pm}^{\pm m} = \hat{U}_{l-1,q_2^\pm}^m + \frac{i}{n}(\hat{U}_{l,q_2^\pm}^m - \hat{U}_{l-1,q_2^\pm}^m), \quad (2.26b)$$

are boundary conditions linearly interpolated (in time) from coarse grid solutions \hat{U}_{l-1}^m , \hat{U}_l^m at q_1^\pm and q_2^\pm . After solving for $\mathbf{W}_{l-1,0}^{\pm m}$ from (2.25a), we set

$$\hat{U}_m(t_{l-1}, Z_j) = \hat{U}_{l-1,j}^m, \quad 0 \leq j \leq 2P, \quad j \notin \{q_1^\pm + 1, q_1^\pm + 2, \dots, q_2^\pm - 1\}, \quad (2.27a)$$

and

$$\hat{U}_m(t_{l-1}, x_j^\pm) = W_{l-1,0,j}^{\pm m}, \quad 1 \leq j \leq n^\pm - 1. \quad (2.27b)$$

Since $Z_{q_1^\pm+j} = x_{nj}^\pm$ for $0 \leq j \leq q_2^\pm - q_1^\pm$, equation (2.27) completes the definition of $\hat{U}_m(t_{l-1}, Z)$ for $Z \in \mathcal{G}_h$. We then repeat this process for $l = l_m, l_m - 1, \dots, l_{m-1} + 1$ to obtain $\hat{U}_m(T_{m-1}, Z)$ for $Z \in \mathcal{G}_h$.

By (2.23b), $V(T_m, e^Z) - \hat{U}_m(T_m, Z)$ is piecewise linear in e^Z , meaning $V(t, e^Z) - \hat{U}_m(t, Z)$ is the sum of values of four binary options. It follows from Lemma 2.1 that

$$\begin{aligned} V(T_{m-1}, e^Z) &= \hat{U}_m(T_{m-1}, Z) + \tilde{f}_m^- \phi_m^a(e^Z, K_m^-, -1) + \tilde{g}_m^- \phi_m^b(e^Z, K_m^-, -1) \\ &\quad + \tilde{f}_m^+ \phi_m^a(e^Z, K_m^+, 1) + \tilde{g}_m^+ \phi_m^b(e^Z, K_m^+, 1), \quad \ln(K_{m-1}^-) < Z < \ln(K_{m-1}^+), \end{aligned} \quad (2.28)$$

where

$$\tilde{f}_m^\pm = a_m^\pm - f_m^\pm, \quad \tilde{g}_m^\pm = b_m^\pm + f_m^\pm K_m^\pm - g_m^\pm.$$

Then we repeat this process for $m - 1$.

We summarize the adaptive mesh refinement procedure as follows:

- 1: Construct a fine grid for $[q_1^-, q_2^-]$ and a similar one for $[q_1^+, q_2^+]$ by dividing each interval of size h into n equal parts
- 2: **for** $m = M - 1$ **downto** 1 **do**
- 3: Compute f_m^\pm , g_m^\pm using (2.22)
- 4: Solve (2.23) numerically for one time step (with step size k) on coarse grid
- 5: Solve (2.23) numerically for n time steps (with step size k/n) on fine grids, using (2.26) as boundary conditions
- 6: Combine numerical solutions on coarse and fine grids to obtain $\hat{U}_m(T_{m-1}, Z)$
- 7: Compute $Y_m(T_{m-1}, Z)$, the smooth part of $V(T_{m-1}, e^Z)$, using (2.28)

8: **end for**

9: Obtain option value $V(T_0, S_0) = V(T_0, e^{Z_P})$

3. Results

3.1. Convergence Study

To study the convergence properties of the proposed method, either with or without mesh adaptation, we first apply the original and the analytically modified C-N method to the following five options:

- Option I: A cash-or-nothing call option with final payoff:

$$V(T_M, S) = \begin{cases} 0, & S < K_M^+ \\ b_M^+, & S \geq K_M^+ \end{cases}.$$

- Option II: An asset-or-nothing call option with final payoff:

$$V(T_M, S) = \begin{cases} 0, & S < K_M^+ \\ a_M^+ S, & S \geq K_M^+ \end{cases}.$$

- Option III: A vanilla call option with final payoff:

$$V(T_M, S) = \begin{cases} 0, & S < K_M^+ \\ a_M^+(S - K_M^+), & S \geq K_M^+ \end{cases}.$$

- Option IV: An exotic call option with final payoff:

$$V(T_M, S) = \begin{cases} 0, & S < K_M^+ \\ a_M^+(S - K_M^+)^2, & S \geq K_M^+ \end{cases}.$$

We remark that this option does not have a counterpart in real-life financial applications. It is designed here solely to test the convergence properties of the proposed method.

These four options are standard options without discretely monitored structures, and for all cases we set

$$r = 0.02, \quad q = 0, \quad \sigma = 0.2, \\ T_0 = 0, \quad T_M = 1, \quad S_0 = 1.1, \quad K_M^+ = 1.2, \quad a_M^+ = b_M^+ = 100.$$

Note also that the final payoffs of the above four options have increasingly higher degree of smoothness (discontinuous, discontinuous, C^0 , and C^1 , respectively). The fifth option, on the other hand, is a discretely monitored option:

- Option V: A double-barrier knock-out option with different monitoring frequencies for the two barriers. Assume there are 250 business days in a year, and the option will expire one year from the valuation date which we denote as day 0. The spot price of the underlying asset at day 0 is 1.1. The upper barrier level is 1.2, and its observation dates are {15, 36, 57, 78, 99, 120, 141, 162, 183, 204, 225, 246}, resembling a monthly monitoring structure. The lower barrier is monitored daily, and the barrier level is 0.9. We further assume that the notional value of the option is 100, that the payoff at maturity is 10% of the notional value, namely 10, and that

$$r = 0.02, \quad q = 0, \quad \sigma = 0.2.$$

In practice, this type of barrier structure is common for snowball options and their variants. Note that the lower barrier of a snowball option is a knock-in barrier, but the option can be priced as the difference between an up-and-out barrier option and a double-barrier knock-out option using PDE methods. Tables 1–8 present results from convergence studies of the four standard options (I–IV) priced using the original C–N method. Since our purpose here is to examine numerically the relationship between convergence rates and smoothness of terminal conditions, the analytically modified C–N method is not used which otherwise would produce (almost) exact pricing results for the first three options (I–III). All four options are priced on the interval²

$$I_0 := [S_0e^{-6\sigma}, S_0e^{6\sigma}] \approx [0.33131363, 3.65212862],$$

with a time step $k = (T_M - T_0)/N = 1/N$ and a mesh size $h = k$, where N ranges from 10 to 5,120. For the first three options, exact option values are available from analytical formulas, while for the fourth option, the option value is estimated on a finer grid with $N = 10,240$. For each numerical approximation $v_{0,j}$ to $V_{0,j} := V(T_0, S_0e^{(j-P)h})$ calculated at T_0 on the computational grid, two types of errors are defined:

$$L^2\text{-error} = \left(\sum_{j=0}^{2P} |v_{0,j} - V_{0,j}|^2 h \right)^{1/2},$$
$$L^\infty\text{-error} = \max_{j=0,1,\dots,2P} |v_{0,j} - V_{0,j}|.$$

The numerical order of an error (L^2 or L^∞) ϵ_N calculated on a grid of size N is estimated using the formula:

$$p_N = \frac{\ln(\epsilon_{N/2}/\epsilon_N)}{\ln 2}.$$

Table 1. Unmodified C–N on a cash-or-nothing call option.

Time steps N	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
10	5.8171×10^0	–	9.8610×10^0	–
20	2.9073×10^0	1.0006	4.8658×10^0	1.0191
40	1.4550×10^0	0.9986	2.4428×10^0	0.9941
80	7.2763×10^{-1}	0.9997	1.2224×10^0	0.9988
160	3.6379×10^{-1}	1.0001	6.1104×10^{-1}	1.0004
320	1.8189×10^{-1}	1.0000	3.0550×10^{-1}	1.0001
640	9.0947×10^{-2}	1.0000	1.5275×10^{-1}	1.0000
1280	4.5473×10^{-2}	1.0000	7.6376×10^{-2}	1.0000
2560	2.2737×10^{-2}	1.0000	3.8188×10^{-2}	1.0000
5120	1.1368×10^{-2}	1.0000	1.9094×10^{-2}	1.0000

Exact option value at S_0 : 32.5191.

² Note that the actual computational domain is $[\ln(S_0) - 6\sigma, \ln(S_0) + 6\sigma]$.

Table 2. Unmodified C-N with mesh adaptation (refinement region: grid points surrounding strike price; refinement ratio: 10% of grid points; refinement factor: 2) on a cash-or-nothing call option.

Time steps N	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
10	3.1317×10^0	—	5.2137×10^0	—
20	1.4875×10^0	1.0741	2.4860×10^0	1.0685
40	7.2892×10^{-1}	1.0290	1.2223×10^0	1.0242
80	3.6387×10^{-1}	1.0023	6.1124×10^{-1}	0.9998
160	1.8189×10^{-1}	1.0003	3.0551×10^{-1}	1.0005
320	9.0947×10^{-2}	1.0000	1.5275×10^{-1}	1.0000
640	4.5473×10^{-2}	1.0000	7.6376×10^{-2}	1.0000
1280	2.2737×10^{-2}	1.0000	3.8188×10^{-2}	1.0000
2560	1.1368×10^{-2}	1.0000	1.9094×10^{-2}	1.0000
5120	5.6842×10^{-3}	1.0000	9.5470×10^{-3}	1.0000

Exact option value at S_0 : 32.5191.

Table 3. Unmodified C-N on an asset-or-nothing call option.

Time steps N	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
10	6.9281×10^0	—	1.1685×10^1	—
20	3.4987×10^0	0.9856	5.8427×10^0	0.9999
40	1.7599×10^0	0.9913	2.9519×10^0	0.9850
80	8.7137×10^{-1}	1.0142	1.4632×10^0	1.0125
160	4.3621×10^{-1}	0.9983	7.3249×10^{-1}	0.9982
320	2.1824×10^{-1}	0.9991	3.6650×10^{-1}	0.9990
640	1.0915×10^{-1}	0.9995	1.8332×10^{-1}	0.9995
1280	5.4586×10^{-2}	0.9998	9.1678×10^{-2}	0.9997
2560	2.7285×10^{-2}	1.0004	4.5825×10^{-2}	1.0004
5120	1.3643×10^{-2}	0.9999	2.2914×10^{-2}	0.9999

Exact option value at S_0 : 44.7791.

Table 4. Unmodified C-N with mesh adaptation (refinement region: grid points surrounding strike price; refinement ratio: 10% of grid points; refinement factor: 2) on an asset-or-nothing call option.

Time steps N	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
10	3.7822×10^0	—	6.2695×10^0	—
20	1.8067×10^0	1.0659	3.0115×10^0	1.0579
40	8.7547×10^{-1}	1.0452	1.4660×10^0	1.0386
80	4.3694×10^{-1}	1.0026	7.3348×10^{-1}	0.9991
160	2.1841×10^{-1}	1.0004	3.6672×10^{-1}	1.0001
320	1.0920×10^{-1}	1.0000	1.8338×10^{-1}	0.9998
640	5.4600×10^{-2}	1.0000	9.1696×10^{-2}	0.9999
1280	2.7289×10^{-2}	1.0006	4.5831×10^{-2}	1.0005
2560	1.3644×10^{-2}	1.0000	2.2916×10^{-2}	1.0000
5120	6.8213×10^{-3}	1.0002	1.1457×10^{-2}	1.0001

Exact option value at S_0 : 44.7791.

Table 5. Unmodified C-N on a vanilla call option.

Time steps N	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
10	7.9800×10^{-2}	—	1.6636×10^{-1}	—
20	1.0897×10^{-2}	2.8725	1.5701×10^{-2}	3.4054
40	3.3141×10^{-3}	1.7172	5.0951×10^{-3}	1.6237
80	1.9199×10^{-3}	0.7876	3.7769×10^{-3}	0.4319
160	3.9259×10^{-4}	2.2900	7.8586×10^{-4}	2.2649
320	6.3321×10^{-5}	2.6323	1.2798×10^{-4}	2.6184
640	1.0403×10^{-5}	2.6057	1.6422×10^{-5}	2.9622
1280	2.8782×10^{-6}	1.8537	4.5740×10^{-6}	1.8441
2560	6.5587×10^{-7}	2.1337	9.9775×10^{-7}	2.1967
5120	2.0410×10^{-7}	1.6841	3.1296×10^{-7}	1.6727

Exact option value at S_0 : 5.75609968.

Table 6. Unmodified C-N with mesh adaptation (refinement region: grid points surrounding strike price; refinement ratio: 10% of grid points; refinement factor: 2) on a vanilla call option.

Time steps N	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
10	3.2530×10^{-2}	—	4.6209×10^{-2}	—
20	1.2858×10^{-2}	1.3391	1.6971×10^{-2}	1.4451
40	2.2103×10^{-3}	2.5404	3.4654×10^{-3}	2.2920
80	5.9012×10^{-4}	1.9052	9.1791×10^{-4}	1.9166
160	1.7804×10^{-4}	1.7288	2.6539×10^{-4}	1.7902
320	5.9818×10^{-5}	1.5736	8.2063×10^{-5}	1.6933
640	1.7355×10^{-5}	1.7852	2.3272×10^{-5}	1.8181
1280	4.5091×10^{-6}	1.9444	6.1307×10^{-6}	1.9245
2560	1.4927×10^{-6}	1.5949	1.9607×10^{-6}	1.6447
5120	3.0864×10^{-7}	2.2739	4.3884×10^{-7}	2.1596

Exact option value at S_0 : 5.75609968.

Table 7. Unmodified C-N on an exotic call option with quadratic payoff.

Time steps N	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
10	2.8546×10^{-1}	—	5.3502×10^{-1}	—
20	7.1561×10^{-2}	1.9961	1.3917×10^{-1}	1.9428
40	1.7810×10^{-2}	2.0065	3.4844×10^{-2}	1.9978
80	4.4342×10^{-3}	2.0059	8.7139×10^{-3}	1.9995
160	1.1058×10^{-3}	2.0035	2.1799×10^{-3}	1.9990
320	2.7600×10^{-4}	2.0024	5.4472×10^{-4}	2.0007
640	6.8835×10^{-5}	2.0034	1.3591×10^{-4}	2.0028
1280	1.7084×10^{-5}	2.0105	3.3713×10^{-5}	2.0113
2560	4.1531×10^{-6}	2.0404	8.1676×10^{-6}	2.0453
5120	9.0598×10^{-7}	2.1967	1.7631×10^{-6}	2.2118

Estimated option value at S_0 : 1.80238666.

Table 8. Unmodified C-N with mesh adaptation (refinement region: grid points surrounding strike price; refinement ratio: 10% of grid points; refinement factor: 2) on an exotic call option with quadratic payoff.

Time steps N	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
10	2.8384×10^{-1}	—	5.3502×10^{-1}	—
20	7.1193×10^{-2}	1.9953	1.3917×10^{-1}	1.9428
40	1.7723×10^{-2}	2.0061	3.4844×10^{-2}	1.9978
80	4.4128×10^{-3}	2.0059	8.7139×10^{-3}	1.9995
160	1.1009×10^{-3}	2.0030	2.1799×10^{-3}	1.9990
320	2.7493×10^{-4}	2.0016	5.4472×10^{-4}	2.0007
640	6.8627×10^{-5}	2.0022	1.3591×10^{-4}	2.0028
1280	1.7053×10^{-5}	2.0087	3.3713×10^{-5}	2.0113
2560	4.1528×10^{-6}	2.0379	8.1676×10^{-6}	2.0453
5120	9.0774×10^{-7}	2.1937	1.7631×10^{-6}	2.2118

Estimated option value at S_0 : 1.80238670.

It can be observed from these tables that:

- For the first two options (I-II), the C-N method converges at only first order due to the discontinuity at $S = K_M^+$ contained in the terminal conditions. On the other hand, mesh adaptation around the discontinuity $S = K_M^+$, where option prices are expected to undergo the most rapid variation, helps reduce the error by a factor of n where $n = 2$ here is the refinement factor.
- For the next two options (III-IV), the C-N method converges at a higher order due to the improved smoothness of the terminal condition at $S = K_M^+$. The precise relationship between the convergence rate and the smoothness of the terminal condition is the subject of a future research, but these examples seem to suggest that a C^1 terminal condition is already sufficient to restore the second-order accuracy of the C-N method (see Remark 3.2).
- When mesh adaptation is applied to the third option, the error first improves by roughly a factor of $n = 2$ up until $N = 160$, after which no significant improvement in error is observed (Table 6). This is due to the fact that the terminal condition in this case has a higher degree of smoothness at $S = K_M^+$ (i.e., C^0), as a result of which the largest error on a sufficiently fine grid does not concentrate around $S = K_M^+$, and hence is not captured by the adaptive mesh which surrounds the strike price $S = K_M^+$. Similar observations apply to the fourth option, whose terminal condition has an even higher degree of smoothness at $S = K_M^+$ (i.e., C^1) and for which no improvement in error is observed for the adaptive mesh calculation at all.
- Despite these observations, the mesh adaptation technique still turns out to be useful when applied to discretely monitored options with high monitoring frequencies, where errors near the barriers caused by the non-smoothness of the terminal conditions dominate the calculations (Section 3.1, 3.3).

Remark 3.1. Since the terminal conditions $V(T_M, S)$ for the first two options contain a discontinuity at $S = K_M^+$ and since the C-N method, with or without analytical modifications, is non-dissipative, the following modification is needed in discrete approximations $v_{N,j}$ to $V(T_M, S)$ so that the convergence of $v_{N,j}$ to $V(T_M, S)$ in a suitable weak sense (e.g., in the sense of H^s for some $s > 0$), and hence the convergence of $v_{0,j}$ to $V(T_0, S)$, can be ensured: let j_0 be chosen such that

$$\ln(S_0) + (j_0 - P)h \leq \ln(K_M^+) < \ln(S_0) + (j_0 + 1 - P)h,$$

or equivalently, such that

$$j_0 \leq P + \frac{1}{h} \ln\left(\frac{K_M^+}{S_0}\right) =: \delta_0 < j_0 + 1.$$

Then for the first option, we set

$$v_{N,j} = \begin{cases} 0, & j < j_0 \\ b_M^+, & j > j_0, \\ b_M^+(j_0 + 1 - \delta_0), & j = j_0 \end{cases}$$

and for the second option, we set

$$v_{N,j} = \begin{cases} 0, & j < j_0 \\ a_M^+ S_0 e^{(j-P)h}, & j > j_0, \\ \frac{1}{2} a_M^+ [S_0 e^{(j_0+1-P)h} + K_M^+] (j_0 + 1 - \delta_0), & j = j_0 \end{cases}$$

For the next two options, the terminal conditions $V(T_M, S)$ are continuous throughout I_0 and hence such modification is not needed. One may simply take

$$v_{N,j} = V(T_M, S_0 e^{(j-P)h}).$$

For dissipative schemes such as FTCS and TR-BDF2, the above modification is not needed either.

Remark 3.2. We have carried out a preliminary error analysis for the proposed method and found that the convergence rate corresponding to a discontinuous, C^0 , and C^1 terminal condition is $1/3$, 1 , and $5/3$, respectively. However, these results are likely suboptimal because they didn't make full use of the dissipation properties of the Black-Scholes PDE.

Tables 9–14 present results from convergence studies of the discretely monitored option (V) priced using both the original and the analytically modified C-N methods. To examine more closely the relationship between the convergence rate and the smoothness of the terminal conditions, we further distinguish between the following two cases for the analytically modified C-N method:

- A C^0 -modification at the barriers:

$$U_m(T_m, Z) = \begin{cases} V(T_m, e^Z), & \ln(K_m^-) < Z < \ln(K_m^+) \\ \tilde{V}_m^-, & Z \leq \ln(K_m^-) \\ \tilde{V}_m^+, & Z \geq \ln(K_m^+) \end{cases},$$

which is (2.12b) with $\tilde{V}_{S,m}^+ = \tilde{V}_{S,m}^- = 0$ and which is only continuous at the barriers.

- A C^1 -modification at the barriers:

$$U_m(T_m, Z) = \begin{cases} V(T_m, e^Z), & \ln(K_m^-) < Z < \ln(K_m^+) \\ \tilde{V}_{S,m}^-(e^Z - K_m^-) + \tilde{V}_m^-, & Z \leq \ln(K_m^-) \\ \tilde{V}_{S,m}^+(e^Z - K_m^+) + \tilde{V}_m^+, & Z \geq \ln(K_m^+) \end{cases},$$

which is just (2.12b) and is continuously differentiable at the barriers.

For all three methods, namely, original C-N and analytically modified C-N with C^0 - and C^1 -modifications, the option is priced on the same interval as in the previous cases:

$$I_0 := [S_0 e^{-6\sigma}, S_0 e^{6\sigma}] \approx [0.33131363, 3.65212862].$$

The calculations are advanced with a time step $k = 1/N_0$, i.e., with N_0 time steps per business day, and with a mesh size $h = k$ where N_0 ranges from 1 to 64. Since the exact option values $V(T_0, S)$ are not available, reference values $V_{0,j}$ against which all numerical approximations $v_{0,j}$ are compared are calculated on a uniform fine grid using the analytically modified C-N method with a C^1 -modification at the barriers and with $N_0 = 128$ time steps per business day.

Table 9. Unmodified C-N on a discretely monitored option.

Time steps per day N_0	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
1	3.6069×10^{-2}	—	7.7073×10^{-2}	—
2	2.9071×10^{-2}	0.3112	1.0085×10^{-1}	−0.3879
4	1.3482×10^{-2}	1.1085	5.5888×10^{-2}	0.8515
8	7.4259×10^{-3}	0.8604	7.8729×10^{-2}	−0.4944
16	3.4649×10^{-3}	1.0997	4.5890×10^{-2}	0.7787
32	2.0030×10^{-3}	0.7907	7.1400×10^{-2}	−0.6377
64	9.7519×10^{-4}	1.0384	4.3468×10^{-2}	0.7160

Reference option value at S_0 : 1.83652751.

Table 10. Unmodified C-N with mesh adaptation (refinement region: grid points surrounding barriers; refinement ratio: 10% of grid points; refinement factor: 2) on a discretely monitored option.

Time steps per day N_0	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
1	2.9450×10^{-2}	—	1.0086×10^{-1}	—
2	1.3587×10^{-2}	1.1161	5.5891×10^{-2}	0.8516
4	7.2129×10^{-3}	0.9135	1.5366×10^{-2}	1.8628
8	3.5757×10^{-3}	1.0123	4.5890×10^{-2}	−1.5784
16	1.8192×10^{-3}	0.9749	1.4147×10^{-2}	1.6977
32	1.0506×10^{-3}	0.7922	4.3468×10^{-2}	−1.6195
64	4.5819×10^{-4}	1.1972	1.5059×10^{-2}	1.5293

Reference option value at S_0 : 1.83652751.

Table 11. Modified C-N (order of modification: C^0) on a discretely monitored option.

Time steps per day N_0	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
1	9.1679×10^{-4}	—	1.0413×10^{-2}	—
2	5.1266×10^{-4}	0.8386	8.4570×10^{-3}	0.3002
4	1.4637×10^{-4}	1.8084	2.5410×10^{-3}	1.7347
8	6.3903×10^{-5}	1.1957	2.2136×10^{-3}	0.1990
16	1.7888×10^{-5}	1.8369	7.1738×10^{-4}	1.6256
32	7.8390×10^{-6}	1.1902	5.4218×10^{-4}	0.4040
64	2.2096×10^{-6}	1.8269	2.0286×10^{-4}	1.4183

Reference option value at S_0 : 1.83652751.

Table 12. Modified C-N (order of modification: C^0) with mesh adaptation (refinement region: grid points surrounding barriers; refinement ratio: 10% of grid points; refinement factor: 2) on a discretely monitored option.

Time steps per day N_0	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
1	6.0032×10^{-4}	—	8.4504×10^{-3}	—
2	1.2291×10^{-4}	2.2881	2.0014×10^{-3}	2.0780
4	5.0400×10^{-5}	1.2862	1.3139×10^{-3}	0.6072
8	1.3808×10^{-5}	1.8679	4.9008×10^{-4}	1.4227
16	6.3195×10^{-6}	1.1276	3.2296×10^{-4}	0.6017
32	1.3910×10^{-6}	2.1837	9.1382×10^{-5}	1.8214
64	8.1619×10^{-7}	0.7691	7.7478×10^{-5}	0.2381

Reference option value at S_0 : 1.83652751.

Table 13. Modified C-N (order of modification: C^1) on a discretely monitored option.

Time steps per day N_0	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
1	3.4536×10^{-4}	—	1.6624×10^{-3}	—
2	4.8800×10^{-5}	2.8232	3.9060×10^{-4}	2.0895
4	1.0948×10^{-5}	2.1561	1.4502×10^{-4}	1.4294
8	2.8768×10^{-6}	1.9282	2.8341×10^{-5}	2.3553
16	8.0608×10^{-7}	1.8355	8.4779×10^{-6}	1.7411
32	2.0010×10^{-7}	2.0102	1.8274×10^{-6}	2.2140
64	4.2000×10^{-8}	2.2523	4.8192×10^{-7}	1.9229

Reference option value at S_0 : 1.83652751.

Table 14. Modified C-N (order of modification: C^1) with mesh adaptation (refinement region: grid points surrounding barriers; refinement ratio: 15% of grid points; refinement factor: 2) on a discretely monitored option.

Time steps per day N_0	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
1	4.8584×10^{-5}	—	3.9005×10^{-4}	—
2	1.2210×10^{-5}	1.9925	1.4489×10^{-4}	1.4288
4	3.0740×10^{-6}	1.9899	2.8372×10^{-5}	2.3524
8	8.6209×10^{-7}	1.8342	8.4697×10^{-6}	1.7441
16	2.1269×10^{-7}	2.0191	1.8294×10^{-6}	2.2110
32	4.5269×10^{-8}	2.2322	4.8141×10^{-7}	1.9260
64	9.7158×10^{-10}	5.5421	3.4793×10^{-9}	7.1123

Reference option value at S_0 : 1.83652751.

It can be observed from these tables that:

- For the original C-N method, the numerical approximations $v_{0,j}$ converge to the reference values $V_{0,j}$ in L^2 at an average rate of roughly 1 but do not converge in L^∞ at all (Table 9). This reduced convergence rate or even lack of convergence is clearly a consequence of the discontinuities at the barriers contained in the terminal conditions. On the other hand, mesh adaptation around the barriers, where option prices are expected to undergo the most rapid variation, helps reduce the error by a factor of n where $n = 2$ here is the refinement factor (Table 10).
- For the analytically modified C-N method, with either C^0 - or C^1 -modifications, the numerical approximations $v_{0,j}$ converge to the reference values $V_{0,j}$ in both L^2 and L^∞ and do so at a higher rate due to the improved smoothness of the terminal conditions at the barriers. More specifically, with C^0 -modifications, $v_{0,j}$ converge to $V_{0,j}$ in L^2 at an average rate of roughly $3/2$ and in L^∞ at an average rate of roughly 1 (Table 11), while with C^1 -modifications, the convergence in both L^2 and L^∞ has an average rate of roughly 2 (Table 13). In other words, these examples seem to suggest that a C^1 -modification to the terminal conditions is already sufficient to restore the second-order accuracy of the C-N method.
- When mesh adaptation is applied with C^0 -modifications, both the L^2 - and L^∞ -errors improve by roughly a factor of $n^{3/2} = 2^{3/2} \approx 2.8284$ for all N_0 , demonstrating the effectiveness of the adaptive mesh (Table 12).
- When mesh adaptation is applied with C^1 -modifications, the numerical approximations $v_{0,j}$ calculated on the finest grid with $N_0 = 64$ are almost identical with the reference values $V_{0,j}$ calculated on the uniform fine grid with $N_0 = 128$, as suggested by the unusually small errors (last row, Table 14). On the other hand, for numerical approximations $v_{0,j}$ calculated on coarser grids with $N_0 \leq 32$, both the L^2 - and L^∞ -errors improve by roughly a factor of $n^2 = 2^2 = 4$ for all N_0 when mesh adaptation is enabled. This provides another strong evidence for the effectiveness of the adaptive mesh, especially on discretely monitored options with high monitoring frequencies.

- We remark that for mesh adaptation applied with C^1 -modifications, a sufficiently large area around the barriers needs to be refined in order for the adaptive mesh to be effective, in view of the improved smoothness of the terminal conditions near the barriers. Indeed, when the refinement ratio in Table 14 was reduced from 15% to 10%, meaning only the $(2P + 1) \times 10\%$ grid points around each barrier are refined, the resulting adaptive mesh was only able to produce marginal improvement in error for $N_0 \leq 4$ and was not able to produce any meaningful improvement in error at all for all larger N_0 . This is not an issue when mesh adaptation is applied with C^0 -modifications or with no analytical modifications, where the errors near the barriers caused by the non-smoothness of the terminal conditions dominate the calculations.

3.2. Comparison with other Numerical Methods

To assess the efficacy and efficiency of the proposed method on options with discretely monitored structures, we next apply other popular, (formally) second-order numerical methods such as C-N with Rannacher time-stepping (CN-RAN) and trapezoidal rule with second-order backward differentiation formula (TR-BDF2) to the discretely monitored option introduced in the previous section. Both methods (CN-RAN and TR-BDF2) have been applied to the four standard options introduced above, and their convergence properties have been found to be very similar to those of the original C-N method (Tables 1, 3, 5 and 7).

Tables 15 and 16 present results from convergence studies of the discretely monitored option priced using the CN-RAN and the TR-BDF2 methods. It can be observed from these tables that:

- Both methods exhibit very similar convergence behaviors and converge in both L^2 and L^∞ at an average rate of roughly 1. This shows, in particular, that both methods lose their second-order accuracy when applied to problems containing discontinuities.
- For a given time step k and mesh size h , both methods produce results that are more accurate than the original C-N method (Table 9) but less accurate than the analytically modified C-N method (Tables 11 and 13).

Table 15. CN-RAN (with the first two C-N steps after each monitoring point replaced by four half-step FTCS steps) on a discretely monitored option.

Time steps per day N_0	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
1	5.7654×10^{-2}	—	1.2330×10^{-1}	—
2	2.7923×10^{-2}	1.0460	5.9944×10^{-2}	1.0405
4	1.2227×10^{-2}	1.1914	2.6136×10^{-2}	1.1976
8	5.6326×10^{-3}	1.1181	1.2007×10^{-2}	1.1222
16	2.7276×10^{-3}	1.0462	5.8052×10^{-3}	1.0485
32	1.3349×10^{-3}	1.0309	2.8390×10^{-3}	1.0319
64	6.6204×10^{-4}	1.0118	1.4074×10^{-3}	1.0124

Reference option value at S_0 : 1.83652751.

Table 16. TR-BDF2 (with $\alpha = 2 - \sqrt{2}$) on a discretely monitored option.

Time steps per day N_0	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
1	4.1522×10^{-2}	—	8.7929×10^{-2}	—
2	2.1859×10^{-2}	0.9256	4.6308×10^{-2}	0.9251
4	1.0773×10^{-2}	1.0208	2.2860×10^{-2}	1.0184
8	5.2753×10^{-3}	1.0301	1.1204×10^{-2}	1.0289
16	2.6388×10^{-3}	0.9994	5.6054×10^{-3}	0.9991
32	1.3128×10^{-3}	1.0072	2.7893×10^{-3}	1.0069
64	6.5650×10^{-4}	0.9997	1.3950×10^{-3}	0.9997

Reference option value at S_0 : 1.83652751.

As a more direct comparison of the different numerical methods studied in the literature (including in this work) and commonly used in practice, we list in Table 17 the pricing results of the following five methods applied to the discretely monitored option introduced in the previous section:

- Analytically modified C-N with a C^1 -modification (CN-C1).
- C-N with Rannacher time-stepping (CN-RAN).
- Trapezoidal rule with second-order backward differentiation formula (TR-BDF2).
- Forward-time centered-space (or implicit Euler) (FTCS).
- Monte-Carlo simulations (MC).

Table 17. Comparison of different methods on a discretely monitored option.

Time steps	CN-C1		CN-RAN	
per day N_0	$V(T_0, S_0)$ error	CPU time	$V(T_0, S_0)$ error	CPU time
1	6.7233×10^{-4}	0.5618	1.1520×10^{-1}	0.1195
2	3.1867×10^{-5}	0.6440	5.5935×10^{-2}	0.2298
4	1.9737×10^{-5}	0.8517	2.4723×10^{-2}	0.3972
8	6.3714×10^{-6}	1.4262	1.1473×10^{-2}	0.8562
16	1.8043×10^{-6}	2.8770	5.5642×10^{-3}	2.1088
32	4.5282×10^{-7}	7.3977	2.7281×10^{-3}	8.5549
64	9.4676×10^{-8}	23.4999	1.3534×10^{-3}	46.2784
Reference option value at S_0 : 1.83652751.				

Time steps	TR-BDF2		FTCS	
per day N_0	$V(T_0, S_0)$ error	CPU time	$V(T_0, S_0)$ error	CPU time
1	8.5771×10^{-2}	0.1145	1.3770×10^{-1}	0.0648
2	4.4429×10^{-2}	0.2203	6.7278×10^{-2}	0.1141
4	2.1929×10^{-2}	0.4887	3.3502×10^{-2}	0.2415
8	1.0782×10^{-2}	1.2014	1.6546×10^{-2}	0.5837
16	5.3920×10^{-3}	3.8852	8.2700×10^{-3}	1.6694
32	2.6851×10^{-3}	19.8665	4.1230×10^{-3}	5.1475
64	1.3427×10^{-3}	96.9560	2.0614×10^{-3}	21.6645
Reference option value at S_0 : 1.83652751.				

Number of paths	MC	
	$V(T_0, S_0)$ error	CPU time
40000	2.6953×10^{-2}	13.1037
60000	5.6742×10^{-3}	17.2020
80000	6.3685×10^{-3}	22.7554
100000	3.2074×10^{-3}	28.0916
120000	1.6717×10^{-3}	33.6178
140000	3.2980×10^{-4}	38.8197
160000	3.2803×10^{-3}	44.4004
Reference option value at S_0 : 1.83652751.		

For each method, the (pointwise) error of the option value at S_0 , together with the CPU time (in seconds)³ used to obtain the pricing result, are shown for different time steps N_0 . Each CPU time is measured by running the corresponding code 5 times and dividing the total CPU time by 5. It can be observed from the table that for a given level of accuracy, the CN-C1 method requires the least amount of CPU time while all other four methods typically require significantly more CPU time. As a specific example, consider the accuracy level of 2×10^{-3} . A more refined study shows that to achieve this level of accuracy, the CN-C1 method requires less than 0.5618 seconds while the other four methods require

- 16.7132 seconds (CN-RAN),

³ The code is developed in Python and is run on a personal computer.

- 36.1549 seconds (TR-BDF2),
- 22.8074 seconds (FTCS), and
- 26.4529 seconds (MC),

respectively (Table 18). This confirms the efficiency of the CN-C1 method.

Table 18. CPU time (in seconds) required to achieve an accuracy level of 2×10^{-3} by different methods on a discretely monitored option.

CN-C1			CN-RAN		
Time steps			Time steps		
per day N_0	$V(T_0, S_0)$ error	CPU time	per day N_0	$V(T_0, S_0)$ error	CPU time
1	6.7233×10^{-4}	0.5618	43	2.0210×10^{-3}	16.2035
—	—	—	44	1.9749×10^{-3}	17.3225
—	—	—	43.4555*	2.0000×10^{-3}	16.7132*
Reference option value at S_0 : 1.83652751.					
*: Estimates obtained from linear interpolation.					

TR-BDF2			FTCS		
Time steps			Time steps		
per day N_0	$V(T_0, S_0)$ error	CPU time	per day N_0	$V(T_0, S_0)$ error	CPU time
42	2.0449×10^{-3}	34.4220	65	2.0309×10^{-3}	21.8704
43	1.9971×10^{-3}	36.2668	66	1.9978×10^{-3}	22.8741
42.9393*	2.0000×10^{-3}	36.1549*	65.9335*	2.0000×10^{-3}	22.8074*
Reference option value at S_0 : 1.83652751.					
*: Estimates obtained from linear interpolation.					

MC		
Number		
of paths	$V(T_0, S_0)$ error	CPU time
108300	2.1369×10^{-3}	26.7155
108400	1.6615×10^{-3}	25.8035
108328.80*	2.0000×10^{-3}	26.4529*
Reference option value at S_0 : 1.83652751.		
*: Estimates obtained from linear interpolation.		

Remark 3.3. It may appear from these results that MC is even more efficient than the TR-BDF2 method. The reality is that, if higher accuracy is demanded (higher than 2×10^{-3}), then the TR-BDF2 method will definitely outperform MC, even though it is relatively slow compared with other finite difference methods considered here.

3.3. Effectiveness of Adaptive Mesh Refinement

We also demonstrate the effectiveness of the mesh adaptation technique by applying the CN-C1 method to the discretely monitored option introduced in the previous sections, both with and without mesh adaptation. The result shows that, with a mesh refinement factor of $n = 8$, an adaptive mesh calculation with N_0 time steps per business day is able to achieve an accuracy level comparable to that of a uniform grid calculation with $8N_0$ time steps per business day, with a more than 60% save in CPU time (Table 19). This confirms the efficiency of the mesh adaptation technique.

Table 19. Comparison of CN-C1 with and without mesh adaptation (refinement region: grid points surrounding barriers; refinement ratio: 15% of grid points; refinement factor: 8) on a discretely monitored option.

N_0	Uniform grid with $8N_0$ time steps per day		Adaptive mesh with N_0 time steps per day		Save in CPU time
	$V(T_0, S_0)$ error	CPU time τ_0	$V(T_0, S_0)$ error	CPU time τ_1	$(\tau_0 - \tau_1) / \tau_0$
1	6.3714×10^{-6}	1.4262	1.3181×10^{-5}	0.5203	63.52%
2	1.8043×10^{-6}	2.8770	3.6323×10^{-6}	1.0140	64.75%
4	4.5282×10^{-7}	7.3977	9.2715×10^{-7}	2.6514	64.16%
8	9.4676×10^{-8}	23.4999	2.1480×10^{-7}	7.7067	67.21%

Reference option value at S_0 : 1.83652751.

3.4. Approximations to Derivatives

It is well known that finite difference methods applied to options with non-smooth terminal conditions tend to produce poor approximations to option Greeks such as Delta Δ and Gamma Γ . In fact, as demonstrated in [11], when the original C-N method is applied to a European call option, the discrete approximations to Δ and Γ both fail to converge, and the discrete approximation to Γ even grows unboundedly when the computational grid is refined. To examine the quality of the discrete approximations to option Greeks produced by the proposed method, we calculate Δ and Γ for the discretely monitored option introduced in the previous section, using the formulas

$$\Delta_{0,j} \approx \frac{V_{0,j+1} - V_{0,j-1}}{2hS_j},$$

$$\Gamma_{0,j} \approx \frac{V_{0,j+1} - 2V_{0,j} + V_{0,j-1}}{h^2 S_j^2} - \frac{V_{0,j+1} - V_{0,j-1}}{2hS_j^2},$$

where $V_{0,j}$ are the same reference option values calculated on a uniform fine grid using the CN-C1 method with $N_0 = 128$ time steps per business day. The approximated Greeks are shown in Figures 2 and 3 together with the option values, where it can be seen that while the discrete approximation to Δ is free of oscillations, the discrete approximation to Γ contains a small amount of oscillations near the lower barrier 0.9 which is monitored daily. These oscillations are likely consequences of the $O(h^2)$ errors introduced by the linear interpolation approximations to V , $\partial V / \partial S$ at the barriers (see (2.17)) which have been significantly amplified by the high-frequency monitoring at the lower barrier. To remove these high-wavenumber oscillations, we replace the first C-N step after each monitoring point by two half-step dissipative steps such as FTCS (implicit Euler) steps and apply the resulting method, coded CN-C1-RAN1, again to the discretely monitored option introduced in the previous section. The results (Figure 4) demonstrate the effectiveness of the modified method in the elimination of high-wavenumber oscillations and also confirm the second-order convergence of both the option value V and option Greeks Δ , Γ in both L^2 and L^∞ (Table 20).

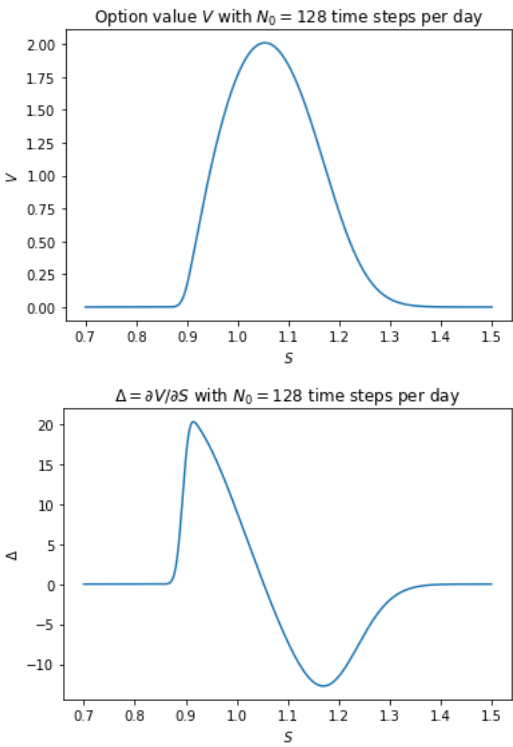


Figure 2. Option values V (top) and Delta Δ (bottom) estimated using the CN-C1 method on a uniform fine grid with $N_0 = 128$ time steps per day.

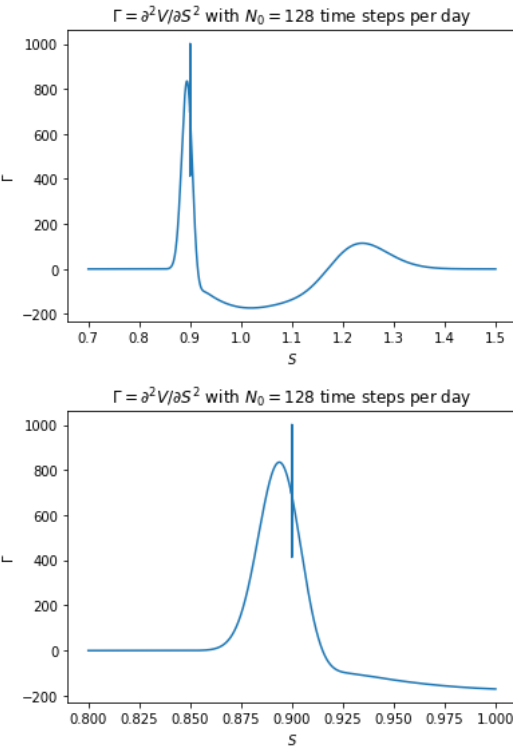


Figure 3. Option Gamma Γ (top) estimated using the CN-C1 method on a uniform fine grid with $N_0 = 128$ time steps per day and a closeup of the Γ plot (bottom).

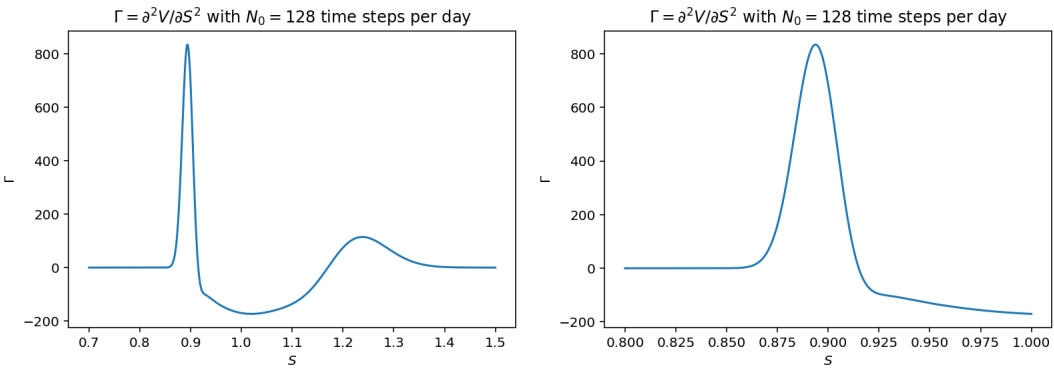


Figure 4. Option Gamma Γ (top) estimated using the CN-C1-RAN1 method on a uniform fine grid with $N_0 = 128$ time steps per day and a closeup of the Γ plot (bottom).

Table 20. CN-C1-RAN1 on a discretely monitored option.

Time steps per day N_0	V			
	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
1	4.9821×10^{-3}	—	1.1748×10^{-2}	—
2	1.2441×10^{-3}	2.0017	2.9359×10^{-3}	2.0006
4	3.1037×10^{-4}	2.0031	7.3259×10^{-4}	2.0027
8	7.6925×10^{-5}	2.0124	1.8167×10^{-4}	2.0117
16	1.9033×10^{-5}	2.0150	4.4944×10^{-5}	2.0151
32	4.5273×10^{-6}	2.0718	1.0692×10^{-5}	2.0716
64	9.0650×10^{-7}	2.3203	2.1407×10^{-6}	2.3203
Reference option value at S_0 : 1.83652817.				
Time steps per day N_0	Δ			
	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
1	5.7753×10^{-2}	—	2.9170×10^{-1}	—
2	1.3173×10^{-2}	2.1323	6.8630×10^{-2}	2.0876
4	3.2319×10^{-3}	2.0272	1.5790×10^{-2}	2.1198
8	8.0164×10^{-4}	2.0113	3.8938×10^{-3}	2.0198
16	1.9799×10^{-4}	2.0175	9.6006×10^{-4}	2.0200
32	4.7122×10^{-5}	2.0710	2.2842×10^{-4}	2.0714
64	9.4293×10^{-6}	2.3212	4.5697×10^{-5}	2.3215
Reference option Delta at S_0 : -7.15943969.				
Time steps per day N_0	Γ			
	L^2 -error	L^2 -order	L^∞ -error	L^∞ -order
1	5.2142×10^0	—	4.5269×10^1	—
2	1.0219×10^0	2.3512	7.7774×10^0	2.5411
4	2.1477×10^{-1}	2.2504	1.7798×10^0	2.1275
8	4.9782×10^{-2}	2.1091	4.9207×10^{-1}	1.8548
16	1.1808×10^{-2}	2.0758	1.2036×10^{-1}	2.0316
32	2.7632×10^{-3}	2.0954	3.1979×10^{-2}	1.9121
64	5.5058×10^{-4}	2.3273	6.8740×10^{-3}	2.2179
Reference option Gamma at S_0 : -135.82019205.				

Remark 3.4. When applied with mesh adaptation, the CN-C1-RAN1 method produces option Greeks Δ , Γ converging at reduced convergence rates, most likely due to the small mismatches among different parts of $\hat{U}_m(t_{l-1}, Z_j)$ introduced by the direct transfer of the fine grid solutions $\mathbf{W}_{l-1,0}^{\pm m}$ to coarse grid (see (2.27)). While this issue may be addressed using, say, a high-order smoothing operator, we decide not to pursue this direction further but instead recommend the readers to consider CN-C1-RAN1 (without mesh adaptation) if accurate approximation to option Greeks is important and to consider CN-C1 (with or without mesh adaptation) if option Greeks are not needed or not important.

4. Discussion

We have introduced an analytical modification for finite difference methods used in the pricing of discretely monitored options in the Black-Scholes framework. The modified finite difference methods can be used to price single and double barrier options, autocallable structured products, and snowball options. We have demonstrated the efficacy and efficiency of the proposed method on the traditional Crank-Nicolson scheme which, in its analytically modified form CN-C1 or CN-C1-RAN1, is capable of producing much more accurate pricing results with much lower CPU cost than other popular option pricing methods such as CN-RAN, TR-BDF2, and Monte-Carlo simulations. We have also introduced a mesh adaptation technique for the analytically modified Crank-Nicolson method, which is capable of achieving an accuracy level comparable to that of a uniform fine grid calculation with a more than 60% improvement in computational speed. We strongly believe that the proposed method and the related ideas will prove a valuable tool for practitioners working with the aforementioned types of options.

This work can be extended in several directions. First, it is desirable to carry out a rigorous error analysis and obtain a precise quantification of the relationship between convergence rates and smoothness of terminal conditions for the proposed method. We have done some preliminary analysis along this direction (Remark 3.2) and the full analysis is currently a work in progress. Second, the proposed method has assumed a volatility that is independent of the asset price S , which may be too restrictive in some practical situations. It is desirable to relax this assumption and apply similar analytical modifications to option pricing problems with variable (in S) volatility. This is the subject of a current research and the results will be reported in a forthcoming paper.

Author Contributions: Conceptualization, M.H.; methodology, G.L. and M.H.; software, G.L. and M.H.; validation, G.L.; formal analysis, G.L.; investigation, G.L. and M.H.; writing—original draft preparation, G.L. and M.H.; writing—review and editing, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: M.H. participated in this research as an independent individual. This work is not related in any way to his current employer, China Merchants Bank. The statements made and opinions expressed here in this paper are solely of the authors' own. They do not reflect the views of China Merchants Bank or its employees and affiliates whatsoever. The authors declare no conflict of interest.

The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. A. Andricopoulos, M. Widdicks, P. Duck, D. Newton, Universal option valuation using quadrature methods, *J. Financ. Econ.* 67 (3) (2003) 447–471.
2. M. Huang, G. Luo, A simple and efficient numerical method for pricing discretely monitored early-exercise options, *Appl. Math. Comput.* 422 (2022) 126985.
3. G. Fusai, I. Abrahams, C. Sgarra, An exact analytical solution for discrete barrier options, *Finance Stochast.* 10 (1) (2006) 1–26.
4. L. Feng, V. Linetsky, Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: a fast Hilbert transform approach, *Math. Financ.* 18 (3) (2008) 337–384.
5. F. Fang, C. Oosterlee, Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions, *Numer. Math.* 114 (2009) 27–62.
6. L. Feng, X. Lin, Pricing Bermudan options in Lévy process models, *SIAM J. Financ. Math.* 4 (1) (2013) 474–493.
7. A. Golbabai, L. Ballestra, D. Ahmadian, A highly accurate finite element method to price discrete double barrier options, *Comput. Econ.* 44 (2) (2014) 153–173.
8. M. Broadie, P. Glasserman, S. Kou, A continuity correction for discrete barrier options, *Math. Financ.* 7 (4) (1997) 325–348.
9. J. Wei, Valuation of discrete barrier options by interpolations, *J. Deriv.* 6 (1) (1998) 51–73.

10. D.-H. Ahn, S. Figlewski, B. Gao, Pricing discrete barrier options with an adaptive mesh model, *J. Deriv.* 6 (4) (1999) 33–43.
11. M. Giles, R. Carter, Convergence analysis of Crank-Nicolson and Rannacher time-marching, *J. Comput. Financ.* 9 (4) (2006) 89–112.
12. R. Rannacher, Finite element solution of diffusion problems with irregular data, *Numer. Math.* 43 (1984) 309–327.
13. F. Le Floc'h, TR-BDF2 for fast stable American option pricing, *J. Comput. Financ.* 17 (3) (2014) 31–56.
14. J. Hull, *Options, Futures, and Other Derivatives*, 9th Edition, Pearson, 2014.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.