

Article

Local Refinement and Adaptive Strategy for a System of Free Boundary Power Options with High Order Compact Differencing

Chinonso Nwankwo ^{1,*}  and Weizhong Dai ²

¹ Department of Mathematics and Statistics, University of Calgary, Calgary, AB T2N 1N4, Canada

² Mathematics and Statistics, Louisiana Tech University, Ruston, LA 71272, USA; dai@coes.latech.edu

* Correspondence: chinonso.nwankwo@ucalgary.ca

Abstract: In this research, we propose fourth-order non-uniform Hermitian differencing with a fifth-order adaptive time integration method for pricing system of free boundary exotic power put options consisting of the option value, delta sensitivity, and gamma. The main objective for implementing the above scheme is to carefully account for the irregularity in the locality of the left corner point after fixing the free boundary. Specifically and mainly, we stretch the performance of our proposed method threefold. First, we exploit the non-uniform fourth-order Hermitian scheme to locally concentrate space grid points arbitrarily close to the left boundary. Secondly, we further leverage the adaptive nature of the embedded time integration method, which allows optimal selection of a time step based on the space grid point distribution and regional variation. Thirdly, we introduce a fourth-order combined Hermitian scheme, which requires fewer grid points for computing the near boundary point of the delta sensitivity and gamma. Another novelty is how we approximate the optimal exercise boundary and its derivative using a fifth-order Robin boundary scheme and fourth-order combined Hermitian scheme. Our proposed method consistently achieves reasonable accuracy with very coarse grids and little runtime across the numerical experiments. We further compare the results with existing methods and the ones we obtained from the uniform space grid.



Citation: Nwankwo, C.; Dai, W. Local Refinement and Adaptive Strategy for a System of Free Boundary Power Options with High Order Compact Differencing. *Axioms* **2023**, *12*, 602. <https://doi.org/10.3390/axioms12060602>

Academic Editor: Kamsing Nonlaopon, Ali Shokri, Daniela Marian and Daniela Inoan

Received: 19 May 2023
Revised: 8 June 2023
Accepted: 15 June 2023
Published: 17 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: power options; optimal exercise boundary; compact finite difference scheme; 5(4) Dormand–Prince embedded pairs

MSC: 65N06; 65M50; 35R35

1. Mathematical Model

In this research, we propose a high-order non-uniform space and adaptive time stepping scheme for pricing a system of free fixed-boundary exotic power put options pricing models. Specifically, the high-order schemes in space and time are obtained from the non-uniform fourth-order Hermitian differencing and Runge–Kutta adaptive time integration based on 5(4) Dormand–Prince embedded pairs [1]. We formulate our solution framework as a free boundary problem which approximates, simultaneously, the optimal exercise boundary, option value, and Greeks. To this end, we first revisit the original Black–Scholes equation. Let us consider non-dividend-paying put options $V(t, S)$ written on an underlying asset with price $S(t)$, strike price K , and time-to-maturity T ; the free boundary partial differential equation is then given as

$$\frac{\partial V(t, S)}{\partial t} + \frac{\sigma^2 S^2}{2} \frac{\partial^2 V(t, S)}{\partial S^2} + rS \frac{\partial V(t, S)}{\partial S} - rV(t, S) = 0. \quad (1)$$

The asset price $S(t)$ is driven by geometric Brownian motion $W(t)$, given as

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t). \quad (2)$$

Here, $\{W(t), t \geq 0\}$ is a standard Brownian motion. σ represents the volatility, and r is the interest rate. Let us consider $F(S_t) = S^m(t)$ as a new underlying asset, with m representing the power of the option value. Applying Itô's formula, and under risk-neutral probability \mathbb{Q} , we then obtain

$$dF(S_t) = dS^m(t) = m \left(\frac{\sigma^2(m-1)}{2} + r \right) S^m(t) dt + \sigma m S^{m-1}(t) dW(t), \tag{3}$$

with the governing differential equation given as below as

$$\frac{\partial V(t, S^m)}{\partial t} + \frac{\sigma^2 m^2 (S^m)^2}{2} \frac{\partial^2 V(t, S^m)}{\partial (S^m)^2} + m S^m \left(\frac{\sigma^2(m-1)}{2} + r \right) \frac{\partial V(t, S^m)}{\partial S^m} - rV(t, S^m) = 0. \tag{4}$$

In this context, $m \in \mathbb{N}_1$. Kim [2] classified power options as standard power options, capped power options, and powered options. For standard power options, the payoff is given as

$$V(T, S^m) = (K^m - S_T^m)^+, \quad m > 0. \tag{5}$$

Here, $(x)^+ = \max\{x, 0\}$. For the capped power options, we also have the following

$$V(T, S^m) = ((K^m - S_T^m)^+, L)^- \quad m > 0, \quad L > 0. \tag{6}$$

Here, $(x)^- = \min\{x, 0\}$, and L represents a pre-defined, capped level. For the powered option, we also provide the following:

$$V(T, S^m) = [(K - S_T)^+]^m \quad m > 0. \tag{7}$$

Lee et al. [3] and Topper [4] described an option representing the payoff in (7) as symmetric power options. Heynen and Kat [5] further presented power options with a more generalized payoff given as

$$V(T, S_T^m) = \left(\sum_{i=1}^m K^i - \sum_{i=1}^m S_T^i \right)^+, \quad m > 0. \tag{8}$$

Here, we focus on the power payoff function presented in the work of Nwozo and Fadugba [6] and Lee [7], which is as follows

$$V(T, S^m) = (K - S_T^m)^+. \tag{9}$$

This is similar to the standard power options in (5), except that K is fixed with $m = 1$. It is also known as an asymmetric power option [4]. Lee et al. [3] and Wang [8] described a generalization of such options in (9) as polynomial options, given as

$$V(T, S^m) = (K - A(S_T))^+, \quad A(S_T) = \sum_{i=1}^m S_T^i. \tag{10}$$

Power options are high-risk, high-reward options. They leverage a magnified position compared to vanilla options due to the non-linear nature of the payoff. Small changes in the asset price can lead to a substantial change in the price of the option based on the power term m . Therefore, for a security that it is expected to rise or fall, the call option holder is bound to incur substantial gains or losses [9,10]. As described in the work of Kim [2] and Huang et al. [9], power options are widely traded in the financial market and have been issued in some countries in the form of polynomial power options [5] and capped FX power options [4].

In the American style framework with $\tau = T - t$, the governing differential equation in (4) is then reduced to a free boundary partial differential equation as shown below

$$\frac{\partial V(\tau, S^m)}{\partial \tau} - \frac{\sigma^2 m^2 (S^m)^2}{2} \frac{\partial^2 V(\tau, S^m)}{\partial (S^m)^2} - m \left(\frac{\sigma^2 (m-1)}{2} + r \right) S^m \frac{\partial V(\tau, S^m)}{\partial S^m} + rV(\tau, S^m) = 0, \quad S^m > s_f(\tau); \tag{11}$$

$$V(\tau, S^m) = K - S^m, \quad \frac{\partial V(\tau, S^m)}{\partial S^m} = -1, \quad S^m < s_f(\tau). \tag{12}$$

The boundary and the initial condition are further given as shown below

$$V(\tau, s_f(\tau)) = K - s_f(\tau), \quad \frac{\partial V(\tau, s_f(\tau))}{\partial S^m} = -1; \tag{13}$$

$$V(\tau, \infty) = 0, \quad \frac{\partial V(\tau, \infty)}{\partial S^m} = 0; \tag{14}$$

$$V(0, S^m) = (K - S^m)^+. \tag{15}$$

To fix the free boundary and remove the convective term that could further introduce error, we then implement a Landau transformation and further take the derivative as given below

$$x = \ln \frac{S^m}{s_f(\tau)}, \quad V(\tau, S^m) = U(\tau, x), \quad W(\tau, x) = \frac{\partial U(\tau, x)}{\partial x}, \quad Y(\tau, x) = \frac{\partial^2 U(\tau, x)}{\partial x^2}. \tag{16}$$

Hence, the transformed system of nonlinear American options pricing model consisting of option value, delta sensitivity, and gamma is given below

$$\frac{\partial U(\tau, x)}{\partial \tau} - \frac{\sigma^2 m^2}{2} \frac{\partial^2 U(\tau, x)}{\partial x^2} - \xi_m(\tau)W(\tau, x) + rU(\tau, x) = 0, \quad x > 0; \tag{17}$$

$$\frac{\partial W(\tau, x)}{\partial \tau} - \frac{\sigma^2 m^2}{2} \frac{\partial^2 W(\tau, x)}{\partial x^2} - \xi_m(\tau) \frac{\partial^2 U(\tau, x)}{\partial x^2} + rW(\tau, x) = 0, \quad x > 0; \tag{18}$$

$$\frac{\partial Y(\tau, x)}{\partial \tau} - \frac{\sigma^2 m^2}{2} \frac{\partial^2 Y(\tau, x)}{\partial x^2} - \xi_m(\tau) \frac{\partial^2 W(\tau, x)}{\partial x^2} + rY(\tau, x) = 0, \quad x > 0; \tag{19}$$

$$\xi_m(\tau) = m \left(\frac{\sigma^2 (m-1)}{2} + r \right) + \frac{s'_f(\tau)}{s_f(\tau)} - \frac{\sigma^2 m^2}{2} = mr + \frac{s'_f(\tau)}{s_f(\tau)} - \frac{m\sigma^2}{2}, \tag{20}$$

$$U(\tau, x) = K - e^x s_f(\tau), \quad W(\tau, x) = -e^x s_f(\tau), \quad x < 0; \tag{21}$$

$$U(0, x) = 0, \quad U(\tau, 0) = K - s_f(\tau), \quad U(\tau, \infty) = 0, \quad \tau > 0; \tag{22}$$

$$W(0, x) = 0, \quad W(t, 0) = -s_f(\tau), \quad W(\tau, \infty) = 0, \quad \tau > 0; \tag{23}$$

$$Y(0, x) = 0, \quad Y(\tau, \infty) = 0, \quad x > 0. \tag{24}$$

When $x \rightarrow 0^+$ and $\tau > 0$, we need to substitute the left boundary values of the option value with the first derivative in time and space to (17), from which we obtain

$$Y(\tau, 0) = \frac{\partial^2 U(\tau, 0)}{\partial x^2} = \frac{2rK}{m^2\sigma^2} - \frac{2}{m^2\sigma^2} \left(\frac{m\sigma^2}{2} - mr + r \right) s_f(\tau). \tag{25}$$

Several authors have proposed methods for solving exotic power options in both European- and American-style frameworks using both integral and differential solution forms. Okelola et al. [10] solved the power options model in the European framework using the Lie group method. Heynen and Kat [5] presented power options in the European framework with a more generalized polynomial payoff. The authors then considered standard power options, generalized power options, and parabola options. Kim [2] solved the power options in the European framework based on the regime switching model. Rao [11] solved the Asian power options under fractional Brownian motion. Esser [12] derived a compact formulation for general payoff and applied it to standard power options, powered options, and capped power options under stochastic volatility. Kim et al. [13] solved the exotic power options under the Heston volatility model. Blenman et al. [14] solved a power exchange options written on a zero-coupon bond in a stochastic interest rate framework. Ha et al. [15] solved a time power option under stochastic volatility.

Lee [7] numerically solved a non-dividend-paying American-style power option PDE model as a free boundary problem with a second-order finite difference scheme. To the best of our knowledge, only this author has attempted to solve such an exotic option PDE model as a free boundary problem with a low-order scheme. Our main objective here is to propose a high-order, fast, stable, and accurate non-uniform and adaptable numerical scheme that solves a system of American-style exotic power option pricing models as a free boundary problem. These systems of free boundary PDEs will solve the option value, delta sensitivity, gamma, and optimal exercise boundary simultaneously. To the best of our knowledge, we are the first to propose such a high-order scheme, non-uniform in space and adaptive in time, for solving systems of free boundary exotic power option pricing PDEs that simultaneously approximate the option value, Greeks, and the free boundary.

The remaining sections of this work are organized as follows. In Section 2, we introduce a fourth-order non-uniform compact finite difference scheme and fifth-order adaptive time integration based on 5(4) Dormand–Prince, Runge–Kutta embedded pairs [1] for solving the free boundary exotic power options pricing problem on a locally refined space grid. We verify the performance of this non-uniform and adaptive scheme and present our numerical results in Section 3. We further compare with the existing method and conclude our study in Section 4.

2. Numerical Methods

In this section, we present the numerical methods for computing the value function, Greeks, and the optimal exercise feature of the American power put options. The numerical computation was carried out in the domain $(\tau, x) \in ([0, T] \times [0, x_M])$. Here, we replace the infinite space domain $[0, \infty)$ with the far-field boundary $[0, x_M]$. This is because, for the put options, the value function vanishes rapidly as we move further away from the right boundary (out of the money). The space–time grid and its node points are described below

$$h_i = x_{i+1} - x_i, \quad k_n = \tau_{n+1} - \tau_n, \quad |[0, x_M]| = \sum_{i=0}^{M-1} h_i, \quad |[0, T]| = \sum_{n=0}^{N-1} \tau_n. \quad (26)$$

We further label the numerical approximation of the option price, delta sensitivity, gamma, and the optimal exercise boundary as u^n, w^n, y^n , and s_f^n , with

$$u_{x_M}^n = w_{x_M}^n = y_{x_M}^n = 0, \quad u_{x_0}^n = K - s_f(\tau_n), \quad w_{x_0} = -s_f(\tau_n); \quad (27)$$

$$y_{x_0}^n = \frac{2rK}{m^2\sigma^2} - \frac{2}{m^2\sigma^2} \left(\frac{m\sigma^2}{2} + mr - r \right) s_f(\tau_n). \quad (28)$$

Stable treatment of the variation at the left corner point and computation of the optimal exercise boundary and its derivative with precision are at the heart of this research work, and both objectives are intertwined. Moreover, the precise computation of the optimal

exercise boundary and its first derivative will influence the accuracy of the option value and the hedge sensitivity. The secondary intention here is to understand how we can use a few grid points to achieve greater accuracy by feeding more of those grid points to the locality of the left corner point. To this end, we seek some leverage and control in the locality of the left corner point of the grid such that we can manipulate the space grid distribution in that neighborhood and further allow the fifth-order time integration method based on Dormand–Prince, Runge–Kutta embedded pairs [1] to select the optimal time step at each time level that adapts well to the regional variation and space grid distribution. The adaptive attribute of the embedded time stepping plays well in our favor here by allowing us to initialize our proposed method with arbitrary time steps k_0 and fixed step sizes h , thus circumventing the stability challenge associated with explicit time integration schemes. The strategic implementation is described fully in the subsections of this section.

2.1. Fourth-Order Non-Equidistant Hermitian Differencing on a Locally Refined Grid

In this subsection, we present a high-order numerical scheme that enables grid refinement about the locality of the left corner point by introducing local mesh refinement results in a non-uniform grid. Hence, the conventional fourth-order compact scheme is no longer applicable. To this end, we introduce the non-equidistant fourth-order Hermitian differencing presented in the work of Shukla and Zhong [16] and Shukla et al. [17] for discretizing interior nodes as given below

$$d_{i,1}f''(x_{i-1}) + f''(x_i) + d_{i,3}f''(x_{i+1}) \approx e_{i,1}f(x_{i-1}) + e_{i,2}f(x_i) + e_{i,3}f(x_{i+1}), \quad i = 2, 3, \dots, M - 1; \tag{29}$$

with

$$d_{i,1} = \left(\frac{h_{i+1}}{h_i + h_{i+1}} \right) \left(\frac{h_i^2 + h_i^2 h_{i+1}^2 - h_{i+1}^2}{h_i^2 + 3h_{i+1}h_i + h_{i+1}^2} \right), \quad d_{i,3} = \left(\frac{h_i}{h_i + h_{i+1}} \right) \left(\frac{h_{i+1}^2 + h_i^2 h_{i+1}^2 - h_i^2}{h_i^2 + 3h_{i+1}h_i + h_{i+1}^2} \right); \tag{30}$$

$$e_{i,1} = \left(\frac{h_{i+1}}{h_i + h_{i+1}} \right) \left(\frac{12}{h_i^2 + 3h_{i+1}h_i + h_{i+1}^2} \right), \quad e_{i,2} = \frac{-12}{h_i^2 + 3h_{i+1}h_i + h_{i+1}^2}; \tag{31}$$

$$e_{i,3} = \left(\frac{h_i}{h_i + h_{i+1}} \right) \left(\frac{12}{h_i^2 + 3h_{i+1}h_i + h_{i+1}^2} \right). \tag{32}$$

Refining grids in the locality of the optimal exercise boundary point gives us some advantages, which will be described below. It allows us to use more nodal points very close to x_0 when computing the optimal exercise boundary. We will exploit this feature in a more strategic way in this subsection.

Here, we will first establish a uniform grid and then manually add a few extra grid points very close to the left boundary point. We will use the near-left boundary grid points to initialize the first derivative of the optimal exercise boundary and further compute the optimal exercise boundary and its first derivative after each successive function evaluation stage with the fifth-order Runge–Kutta embedded time integration method based on Dormand–Prince pairs. This implementation will both allow us to use points very close to the left boundary to approximate the optimal exercise boundary and its derivative and enable a locally refined grid in the left corner, where inherent irregularities and variation reside. Here, we delicately add these few grid points in such a way that the points we use for computing the free boundary and the near boundary discrete values are both very refined and equidistant, even though the grid is globally non-uniform. For a description, please see Figure 1. For a similar implementation for solving the options pricing problem with a high-order compact scheme, please see the work of Chen et al. [18] and Lee and Sun [19].

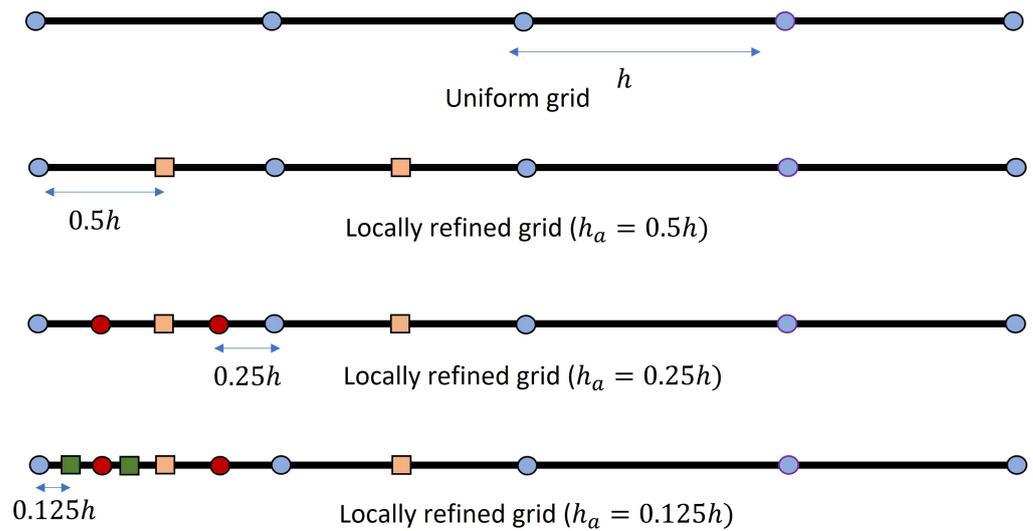


Figure 1. Uniform and locally refined space grids ($h_a \leq \min h_i$).

For the near boundary scheme, i.e, x_1 , we will consider fourth-order combined compact finite difference schemes for approximating the solution of the option value as given below

$$6u''(\tau_n, x_1) = \frac{12}{h_a^2}(u(\tau_n, x_0) - 2u(\tau_n, x_1) + u(\tau_n, x_2)) - \frac{3}{h_a}(w(\tau_n, x_2) - w(\tau_n, x_0)), \quad h_a \leq \min h_i. \quad (33)$$

We point out that the fourth-order, one-sided, non-combined compact scheme given below

$$14u''(\tau_n, x_1) - 5u''(\tau_n, x_2) + 4u''(\tau_n, x_3) - u''(\tau_n, x_4) = \frac{12}{h_a^2}(u(\tau_n, x_0) - 2u(\tau_n, x_1) + u(\tau_n, x_2)), \quad h_a \leq \min h_i \quad (34)$$

can also be used as the near boundary scheme for the option value as presented in the work of Zhao and Corless [20] and Nwankwo and Dai [21]. However, we observe that, for uniform implementation, the fourth-order combined compact scheme in (33) provides more reasonable accuracy, although the discrepancy when compared with (34) is not substantial. We imagine that the reason could be that we use fewer grid points for approximation. The availability of discrete solutions for hedge sensitivity also works in our favor here. Moreover, for the non-uniform grid, we are trying to avoid using more grid points to account for the near boundary scheme based on the slight uniformity we achieve near the left boundary. Any selected grid points for accounting for the near boundary value must fall within the local uniform domain, even though our space grid is globally non-uniform. Hence, in the Section 3, we consider only (33). Furthermore, for the delta sensitivity, we use a different fourth-order near boundary combined compact finite difference scheme by considering the following lemma

Lemma 1. Assume that $u(\tau, x) \in C^{1,3}((0, T], (0, x_{max}])$; we have

$$u'''(\tau, x_1) = \frac{15}{2h_a^3}[u(\tau, x_2) - u(\tau, x_0)] - \frac{3}{2h_a^2}[w(\tau, x_0) + 8w(\tau, x_1) + w(\tau, x_2)] + O(h^4). \quad (35)$$

Proof. Let $x_{i+j} - x_i = jh$. To prove the lemma in (35), we consider two well-known differences of two Taylor series expansions

$$u(\tau, x_{i-1}) - u(\tau, x_{i+1}) = 2hu_x(\tau, x_i) + \frac{h^3}{3}u_{x^3}(\tau, x_i) + \frac{h^5}{60}u_{x^5}(\tau, x_i) + O(h^7), \quad (36)$$

$$u(\tau, x_{i-1}) + u(\tau, x_{i+1}) = 2u(\tau, x_i) + h^2 u_{x^2}(\tau, x_i) + \frac{h^4}{12} u_{x^4}(\tau, x_i) + O(h^6). \tag{37}$$

Taking the derivative of (37) with respect to x and multiplying by $\frac{h}{5}$, we obtain the equation

$$\frac{h}{5} u_x(\tau, x_{i-1}) + \frac{h}{5} u_x(\tau, x_{i+1}) = \frac{2h}{5} u_x(\tau, x_i) + \frac{h^3}{5} u_{x^3}(\tau, x_i) + \frac{h}{60} u_{x^5}(\tau, x_i) + O(h^7). \tag{38}$$

Subtracting (36) from (38) and re-arranging, we then obtain (35), and the derivation is complete. \square

We observed after deriving this fourth-order combined compact scheme in (35) that the latter had already been presented and used in the work of Abrahamsen and Fornberg [22] and, as such, we recognized the need to acknowledge their work here. The high-order scheme in (35) allows us to use the discrete value of the option value to approximate the near boundary value of delta, thereby reducing sensitivity. Moreover, fewer nodal values of the higher derivatives are used when deriving the near boundary scheme for the delta sensitivity, as shown in (35). It is worth mentioning that, for the delta sensitivity, the novel near-boundary fourth-order combined compact scheme in (35) is much more suitable and provides a more accurate result when compared with (34). In the uniform case, the condition number of the discrete matrix system generated with (34) is larger when compared with (35). Similarly, the near boundary value for the gamma is approximated as follows

$$u''''(\tau, x_1) = \frac{15}{2h_a^3} [w(\tau, x_2) - w(\tau, x_0)] - \frac{3}{2h_a^2} [y(\tau, x_0) + 8y(\tau, x_1) + y(\tau, x_2)] + O(h^4). \tag{39}$$

More importantly, we present below a high-order Robin boundary scheme for approximating the optimal exercise boundary and its derivative after each function evaluation stage with the high-order embedded time integration method. To this end, we consider the following lemma

Lemma 2. Assume that $u(\tau, x) \in C^{1,2}((0, T], [0, x_{max}])$; we have

$$-\frac{85}{18} u(\tau, x_0) + 6u(\tau, x_1) - \frac{3}{2} u(\tau, x_2) + \frac{2}{9} u(\tau, x_3) = \frac{11}{3} h_a u'(\tau, x_0) + h_a^2 u''(\tau, x_0) + O(h^5). \tag{40}$$

Proof. Let $x_{i+1} - x_i = h_a, i = 1, 2, 3$. To prove the lemma in (40), we first present the Taylor series expansion around $x_0 = 0$ as follows

$$u(\tau, x_1) = u(\tau, x_0) + h_a u_x(\tau, x_0) + \frac{h_a^2}{2} u_{x^2}(\tau, x_0) + \frac{h_a^3}{6} u_{x^3}(\tau, x_0) + \frac{h_a^4}{24} u_{x^4}(\tau, x_0) + O(h^5), \tag{41}$$

$$u(\tau, x_2) = u(\tau, x_0) + 2h_a u_x(\tau, x_0) + \frac{4h_a^2}{2} u_{x^2}(\tau, x_0) + \frac{4h_a^3}{3} u_{x^3}(\tau, x_0) + \frac{2h_a^4}{3} u_{x^4}(\tau, x_0) + O(h^5), \tag{42}$$

$$u(\tau, x_3) = u(\tau, x_0) + 3h_a u_x(\tau, x_0) + \frac{9h_a^2}{2} u_{x^2}(\tau, x_0) + \frac{9h_a^3}{2} u_{x^3}(\tau, x_0) + \frac{27h_a^4}{8} u_{x^4}(\tau, x_0) + O(h^5). \tag{43}$$

Multiplying (41) by 8 and subtracting from (42), we obtain

$$8u(\tau, x_1) - u(\tau, x_2) = 7u(\tau, x_0) + 6h_a u_x(\tau, x_0) + 2u_{x^2}(\tau, x_0) + \frac{h_a^4}{3} u_{x^4}(\tau, x_0) + O(h^5). \tag{44}$$

Multiplying (42) by $\frac{27}{8}$ and subtracting from (43), we obtain

$$\frac{27}{8} u(\tau, x_2) - u(\tau, x_3) = \frac{19}{8} u(\tau, x_0) + \frac{15}{4} h_a u_x(\tau, x_0) + \frac{9h_a^2}{4} u_{x^2}(\tau, x_0) - \frac{9h_a^4}{3} u_{x^4}(\tau, x_0) + O(h^5). \tag{45}$$

Multiplying (44) by $\frac{27}{8}$ and subtracting from (42) with some re-arrangement, we obtain (40), and the proof is completed. \square

The high-order scheme above will enable us to approximate the optimal exercise boundary and its first derivative in time with better precision. If we consider value matching, smooth pasting, and the implied second derivative boundary conditions

$$u(\tau, x_0) = K - s_f(\tau), \quad u'(\tau, x_0) = -s_f'(\tau); \tag{46}$$

$$u''(\tau, x_0) = \frac{2rK}{m^2\sigma^2} - \frac{2}{m^2\sigma^2} \left(\frac{m\sigma^2}{2} + mr - r \right) s_f(\tau), \tag{47}$$

we obtain

$$6u(\tau, x_1) - \frac{3}{2}u(\tau, x_2) + \frac{2}{9}u(\tau, x_3) = \left(\frac{85}{18} + \frac{2rh_a^2}{m^2\sigma^2} \right) K - \left[\frac{85}{18} + \frac{11}{3}h_a + \frac{2}{m^2\sigma^2} \left(\frac{m\sigma^2}{2} + mr - r \right) h_a^2 \right] s_f(\tau) + O(h^5). \tag{48}$$

For simplicity, let

$$\bar{\sigma} = m^2\sigma^2, \quad a_1 = \left(\frac{85}{18} + \frac{2rh_a^2}{\bar{\sigma}^2} \right), \quad a_2 = \frac{85}{18} + \frac{11}{3}h_a + \frac{2}{\bar{\sigma}^2} \left(\frac{m\sigma^2}{2} + mr - r \right) h_a^2. \tag{49}$$

Hence,

$$6u(\tau, x_1) - \frac{3}{2}u(\tau, x_2) + \frac{2}{9}u(\tau, x_3) = a_1K - a_2s_f(\tau) + O(h^5). \tag{50}$$

Equation (50) approximates the optimal exercise boundary after each function evaluation based on Runge–Kutta embedded pairs that are described in the following subsection. Equation (50) is also used to formulate an analytical approximation for initializing and further correcting the first derivative of the optimal exercise boundary. To this end, we take the derivative of (48) with respect to time and obtain the following

$$6u_\tau(\tau, x_1) - \frac{3}{2}u_\tau(\tau, x_2) + \frac{2}{9}u_\tau(\tau, x_3) = -a_2s_f'(\tau) + O(h^5). \tag{51}$$

Due to the time-dependent coefficient present in our model, the first derivative of the optimal exercise boundary needs to be computed with precision, especially very close to the expiration. Here, we derive a novel high-order analytical approximation for the initialization (prediction) of the first derivative of the optimal exercise boundary at each time level, which will be subsequently corrected from the solutions of the Runge–Kutta pairs at each stage of function evaluation using (51). To this end, we recall the fourth-order combined compact finite difference presented as given below

$$6u''(\tau_n, x_i) \approx \frac{12}{h_a^2}(u(\tau_n, x_{i-1}) - 2u(\tau_n, x_i) + u(\tau_n, x_{i+1})) - \frac{3}{h_a}(w(\tau_n, x_{i+1}) - w(\tau_n, x_{i-1})). \tag{52}$$

For simplicity, let

$$\mathcal{U}(\tau, x_i) = \frac{12}{h_a^2}(u(\tau_n, x_{i-1}) - 2u(\tau_n, x_i) + u(\tau_n, x_{i+1})) - \frac{3}{h_a}(w(\tau_n, x_{i+1}) - w(\tau_n, x_{i-1})), \quad i = 1, 2, 3. \tag{53}$$

Furthermore, considering the PDE governing the fixed free boundary American options, we obtain the following

$$\begin{aligned}
 & \frac{\bar{\sigma}^2}{2} \left(6\mathcal{U}(\tau_n, x_1) - \frac{3}{2}\mathcal{U}(\tau_n, x_2) + \frac{2}{9}\mathcal{U}(\tau_n, x_3) \right) = \frac{\bar{\sigma}^2}{2} \left(6u_{xx}(\tau_n, x_1) - \frac{3}{2}u_{xx}(\tau_n, x_2) + \frac{2}{9}u_{xx}(\tau_n, x_3) \right) = \\
 & 6u_\tau(\tau_n, x_1) - \frac{3}{2}u_\tau(\tau_n, x_2) + \frac{2}{9}u_\tau(\tau_n, x_3) - \frac{1}{s_f(\tau_n)} s'_f(\tau_n) \left(6w(\tau_n, x_1) - \frac{3}{2}w(\tau_n, x_2) + \frac{2}{9}w(\tau_n, x_3) \right) \\
 & - \left(mr - \frac{m\sigma^2}{2} \right) \left(6w(\tau_n, x_1) - \frac{3}{2}w(\tau_n, x_2) + \frac{2}{9}w(\tau_n, x_3) \right) \\
 & + r \left(6u(\tau_n, x_1) - \frac{3}{2}u(\tau_n, x_2) + \frac{2}{9}u(\tau_n, x_3) \right) = \\
 & - \left[a_2 + \frac{1}{s_f(\tau)} \left(6w(\tau, x_1) - \frac{3}{2}w(\tau, x_2) + \frac{2}{9}w(\tau, x_3) \right) \right] s'_f(\tau_n) \\
 & - \left(mr - \frac{m\sigma^2}{2} \right) \left(6w(\tau, x_1) - \frac{3}{2}w(\tau, x_2) + \frac{2}{9}w(\tau, x_3) \right) \\
 & + r \left(6u(\tau, x_1) - \frac{3}{2}u(\tau, x_2) + \frac{2}{9}u(\tau, x_3) \right) + O(h^4). \tag{54}
 \end{aligned}$$

Further simplification reveals that

$$s'_f(\tau_n) = \frac{\mathcal{F}(\tau_n)}{\mathcal{G}(\tau_n)} + O(h^4), \tag{55}$$

with

$$\begin{aligned}
 \mathcal{F}(\tau_n) &= \frac{\bar{\sigma}^2}{2} \left(6\mathcal{U}(\tau_n, x_1) - \frac{3}{2}\mathcal{U}(\tau_n, x_2) + \frac{2}{9}\mathcal{U}(\tau_n, x_3) \right) - r \left[a_1 K - a_2 s_f(\tau_n) \right] \\
 &+ \left(mr - \frac{m\sigma^2}{2} \right) \left(6w(\tau_n, x_1) - \frac{3}{2}w(\tau_n, x_2) + \frac{2}{9}w(\tau_n, x_3) \right), \tag{56}
 \end{aligned}$$

$$\mathcal{G}(\tau) = - \left[a_2 + \frac{1}{s_f(\tau_n)} \left(6w(\tau_n, x_1) - \frac{3}{2}w(\tau_n, x_2) + \frac{2}{9}w(\tau_n, x_3) \right) \right]. \tag{57}$$

It is important to observe that establishing a system of free boundary PDEs consisting of the option value and hedge sensitivity is very useful here. Not only does it enable us to deal with the convective term that could further introduce error and allow simple implementation of the fourth-order compact scheme, but we also use the discrete solution of the delta sensitivity for predicting the initial value of the first derivative of the optimal exercise boundary for each time level. Furthermore, computing the Greeks simultaneously with the option value and optimal exercise boundary using a fourth-order compact operator presents some benefits that could be substantial if the computational cost is very low. We hope to demonstrate extensively in the numerical experiment section that our

implementation holds such an advantage. For the rest of the interior points for the option value and delta sensitivity, we use the same fourth-order compact scheme presented in (29). Hence,

$$A_u \mathbf{u}_{xx} = B_u \mathbf{u} + \mathbf{b}_u, \quad A_{w,y} \mathbf{w}_{xx} = B_{w,y} \mathbf{w} + \mathbf{b}_w, \quad A_{w,y} \mathbf{y}_{xx} = B_{w,y} \mathbf{y} + \mathbf{b}_y; \quad (58)$$

with $A_u, B_u, \mathbf{b}_u, A_{w,y}, B_{w,y}, \mathbf{b}_w$, and \mathbf{b}_y , given as:

$$\begin{aligned}
 A_u &= \begin{bmatrix} 6 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ d_{i,1} & 1 & d_{i,3} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & d_{3,2} & 1 & d_{3,4} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & d_{4,3} & 1 & d_{4,5} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & d_{M-5,M-4} & 1 & d_{M-3,M-4} & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & d_{M-4,M-3} & 1 & d_{M-2,M-3} & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & d_{M-3,M-2} & 1 & d_{M-1,M-2} \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & d_{M-2,M-1} & 1 \end{bmatrix}, \\
 A_{w,y} &= \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & d_{3,2} & 1 & d_{3,4} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & d_{4,3} & 1 & d_{4,5} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & d_{M-5,M-4} & 1 & d_{M-3,M-4} & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & d_{M-4,M-3} & 1 & d_{M-2,M-3} & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & d_{M-3,M-2} & 1 & d_{M-1,M-2} \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & d_{M-2,M-1} & 1 \end{bmatrix}, \\
 B_u &= \begin{bmatrix} -\frac{24}{h_a^2} & \frac{12}{h_a^2} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ e_{2,1} & e_{2,2} & e_{2,3} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & e_{3,2} & e_{3,3} & e_{3,4} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & e_{4,3} & e_{4,4} & e_{4,5} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & e_{M-5,M-4} & e_{M-4,M-4} & e_{M-3,M-4} & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & e_{M-4,M-3} & e_{M-3,M-3} & e_{M-2,M-3} & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & e_{M-3,M-2} & e_{M-2,M-2} & e_{M-1,M-2} \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & e_{M-2,M-1} & e_{M-1,M-1} \end{bmatrix}, \\
 B_{w,y} &= \begin{bmatrix} -\frac{120}{h_a^2} & -\frac{15}{h_a^2} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ e_{2,1} & e_{2,2} & e_{2,3} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & e_{3,2} & e_{3,3} & e_{3,4} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & e_{4,3} & e_{4,4} & e_{4,5} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & e_{M-5,M-4} & e_{M-4,M-4} & e_{M-3,M-4} & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & e_{M-4,M-3} & e_{M-3,M-3} & e_{M-2,M-3} & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & e_{M-3,M-2} & e_{M-2,M-2} & e_{M-1,M-2} \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & e_{M-2,M-1} & e_{M-1,M-1} \end{bmatrix},
 \end{aligned}$$

$$\mathbf{b}_u = \begin{bmatrix} \frac{12}{h_a^2} u_0^n - \frac{3}{h_a} (w_2^n - w_0^n) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_w = \begin{bmatrix} \frac{75}{h_a^3} (u_2^n - u_0^n) - \frac{15}{h_a^2} w_0^n \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_y = \begin{bmatrix} \frac{75}{h_a^3} (w_2^n - w_0^n) - \frac{15}{h_a^2} y_0^n \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

With the discrete matrix system above, we then obtain the semi-discrete system for the option value and hedge sensitivity as follows

$$\mathbf{u}_\tau^n = \frac{\bar{\sigma}^2}{2} A_u^{-1} (B_u \mathbf{u}^n + \mathbf{b}_u^n) + \zeta_{m(n)} \mathbf{w}^n - r \mathbf{u}^n + O(h^4), \tag{59}$$

$$\mathbf{w}_\tau^n = \frac{\bar{\sigma}^2}{2} A_{w,y}^{-1} (B_{w,y} \mathbf{w}^n + \mathbf{b}_w^n) + \zeta_{m(n)} A_u^{-1} (B_u \mathbf{u}^n + \mathbf{b}_u^n) - r \mathbf{w}^n + O(h^4), \tag{60}$$

$$\mathbf{y}_\tau^n = \frac{\bar{\sigma}^2}{2} A_{w,y}^{-1} (B_{w,y} \mathbf{y}^n + \mathbf{b}_y^n) + \zeta_{m(n)} A_{w,y}^{-1} (B_{w,y} \mathbf{w}^n + \mathbf{b}_w^n) - r \mathbf{y}^n + O(h^4). \tag{61}$$

Remark 1. It is worth acknowledging here the recent work of [23], where the authors implemented a high-order non-uniform Hermitian scheme for solving the fractional Black–Scholes model. The non-uniform high-order scheme proposed by the authors in their paper can also be used in this research work.

2.2. 5(4) Dormand–Prince, Runge–Kutta Embedded Time Integration Method

As mentioned in the previous subsection, at each time level (and more importantly, when $\tau = 0$), the first derivative of the optimal exercise boundary is initialized (predicted) as given below.

Preliminary stage:

$$s'_f(\tau_n) = \frac{\mathcal{F}^n}{\mathcal{G}^n} + O(h^4), \quad \zeta_{m(n)} = mr + \frac{s'_f(\tau_n)}{s_f(\tau_n)} - m \frac{\sigma^2}{2}. \tag{62}$$

The numerical procedure for the implementation of adaptive time stepping at each time level based on the 5(4) Dormand–Prince, Runge–Kutta embedded time integration method is described below.

First stage:

$$\mathbf{u}_\tau^n = \frac{\bar{\sigma}^2}{2} A_u^{-1} (B_u \mathbf{u}^n + \mathbf{b}_u^n) + \zeta_{m(n)} \mathbf{w}^n - r \mathbf{u}^n + O(h^4)$$

$$\mathbf{u}^{n+1/7} = \mathbf{u}^n + \frac{k}{5} \mathbf{u}_\tau^n$$

$$\mathbf{w}_\tau^n = \frac{\bar{\sigma}^2}{2} A_{w,y}^{-1} (B_{w,y} \mathbf{w}^n + \mathbf{b}_{w,y}^n) + \zeta_{m(n)} A_u^{-1} (B_u \mathbf{u}^n + \mathbf{b}_u^n) - r \mathbf{w}^n + O(h^4)$$

$$\mathbf{w}^{n+1/7} = \mathbf{w}^n + \frac{k}{5} \mathbf{w}_\tau^n$$

$$\mathbf{y}_\tau^n = \frac{\bar{\sigma}^2}{2} A_{w,y}^{-1} (B_{w,y} \mathbf{y}^n + \mathbf{b}_{w,y}^n) + \zeta_{m(n)} A_{w,y}^{-1} (B_{w,y} \mathbf{w}^n + \mathbf{b}_{w,y}^n) - r \mathbf{y}^n + O(h^4)$$

$$\mathbf{y}^{n+1/7} = \mathbf{y}^n + \frac{k}{5} \mathbf{y}_\tau^n$$

$$\begin{aligned}
 s_f(\tau_{n+1/7}) &= -\frac{\mathcal{M}(\tau_{n+1/7}) - a_1K}{a_2} + O(h^5), \quad s'_f(\tau_{n+1/7}) = -\frac{\mathcal{M}'(\tau_{n+1/7})}{a_2} + O(h^5) \\
 u(\tau_{n+1/7}, x_0) &= K - s_f(\tau_{n+1/7}), \quad w(\tau_{n+1/7}, x_0) = -s_f(\tau_{n+1/7}) \\
 \mathcal{M}(\tau_{n+1/7}) &= 6u(\tau_{n+1/7}, x_1) - \frac{3}{2}u(\tau_{n+1/7}, x_2) + \frac{2}{9}u(\tau_{n+1/7}, x_3) \\
 \xi_{m(n+1/7)} &= mr + \frac{s'_f(\tau_{n+1/7})}{s_f(\tau_{n+1/7})} - m\frac{\sigma^2}{2}
 \end{aligned}$$

Second stage:

$$\begin{aligned}
 \mathbf{u}_\tau^{n+1/7} &= \frac{\bar{\sigma}^2}{2}A_u^{-1}\left(B_u\mathbf{u}^{n+1/7} + \mathbf{b}_u^{n+1/7}\right) + \xi_{m(n+1/7)}\mathbf{w}^{n+1/7} - r\mathbf{u}^{n+1/7} + O(h^4) \\
 \mathbf{u}^{n+2/7} &= \mathbf{u}^n + \frac{3k}{40}\mathbf{u}_\tau^n + \frac{9k}{40}\mathbf{u}_\tau^{n+1/7} \\
 \mathbf{w}_\tau^{n+1/7} &= \frac{\bar{\sigma}^2}{2}A_{w,y}^{-1}\left(B_{w,y}\mathbf{w}^{n+1/7} + \mathbf{b}_w^{n+1/7}\right) + \xi_{n+1/7}A_u^{-1}\left(B_u\mathbf{u}^{n+1/7} + \mathbf{b}_u^{n+1/7}\right) \\
 &\quad - r\mathbf{w}^{n+1/7} + O(h^4) \\
 \mathbf{w}^{n+2/7} &= \mathbf{w}^n + \frac{3k}{40}\mathbf{w}_\tau^n + \frac{9k}{40}\mathbf{w}_\tau^{n+1/7} \\
 \mathbf{y}_\tau^{n+1/7} &= \frac{\bar{\sigma}^2}{2}A_{w,y}^{-1}\left(B_{w,y}\mathbf{y}^{n+1/7} + \mathbf{b}_y^{n+1/7}\right) + \xi_{m(n+1/7)}A_{w,y}^{-1}\left(B_{w,y}\mathbf{w}^{n+1/7} + \mathbf{b}_w^{n+1/7}\right) \\
 &\quad - r\mathbf{y}^{n+1/7} + O(h^4) \\
 \mathbf{y}^{n+2/7} &= \mathbf{y}^n + \frac{3k}{40}\mathbf{y}_\tau^n + \frac{9k}{40}\mathbf{y}_\tau^{n+1/7} \\
 s_f(\tau_{n+2/7}) &= -\frac{\mathcal{M}(\tau_{n+2/7}) - a_1K}{a_2} + O(h^5), \quad s'_f(\tau_{n+2/7}) = -\frac{\mathcal{M}'(\tau_{n+2/7})}{a_2} + O(h^5) \\
 u(\tau_{n+2/7}, x_0) &= K - s_f(\tau_{n+2/7}), \quad w(\tau_{n+2/7}, x_0) = -s_f(\tau_{n+2/7}) \\
 \mathcal{M}(\tau_{n+2/7}) &= 6u(\tau_{n+2/7}, x_1) - \frac{3}{2}u(\tau_{n+2/7}, x_2) + \frac{2}{9}u(\tau_{n+2/7}, x_3) \\
 \xi_{m(n+2/7)} &= mr + \frac{s'_f(\tau_{n+2/7})}{s_f(\tau_{n+2/7})} - m\frac{\sigma^2}{2}
 \end{aligned}$$

Third stage:

$$\begin{aligned}
 \mathbf{u}_\tau^{n+2/7} &= \frac{\bar{\sigma}^2}{2}A_u^{-1}\left(B_u\mathbf{u}^{n+2/7} + \mathbf{b}_u^{n+2/7}\right) + \xi_{m(n+2/7)}\mathbf{w}^{n+2/7} - r\mathbf{u}^{n+2/7} + O(h^4) \\
 \mathbf{u}^{n+3/7} &= \mathbf{u}^n + \frac{44k}{45}\mathbf{u}_\tau^n - \frac{55k}{16}\mathbf{u}_\tau^{n+1/7} + \frac{32k}{9}\mathbf{u}_\tau^{n+2/7} \\
 \mathbf{w}_\tau^{n+2/7} &= \frac{\bar{\sigma}^2}{2}A_{w,y}^{-1}\left(B_{w,y}\mathbf{w}^{n+2/7} + \mathbf{b}_w^{n+2/7}\right) + \xi_{m(n+2/7)}A_u^{-1}\left(B_u\mathbf{u}^{n+2/7} + \mathbf{b}_u^{n+2/7}\right) \\
 &\quad - r\mathbf{w}^{n+2/7} + O(h^4)
 \end{aligned}$$

$$\mathbf{w}^{n+3/7} = \mathbf{w}^n + \frac{44k}{45}\mathbf{w}_\tau^n - \frac{55k}{16}\mathbf{w}_\tau^{n+1/7} + \frac{32k}{9}\mathbf{w}_\tau^{n+2/7}$$

$$\mathbf{y}_\tau^{n+2/7} = \frac{\bar{\sigma}^2}{2}A_{w,y}^{-1}\left(B_{w,y}\mathbf{y}^{n+2/7} + \mathbf{b}_y^{n+2/7}\right) + \zeta_{m(n+2/7)}A_{w,y}^{-1}\left(B_{w,y}\mathbf{w}^{n+2/7} + \mathbf{b}_w^{n+2/7}\right) - r\mathbf{y}^{n+2/7} + O(h^4)$$

$$\mathbf{y}^{n+3/7} = \mathbf{y}^n + \frac{44k}{45}\mathbf{y}_\tau^n - \frac{55k}{16}\mathbf{y}_\tau^{n+1/7} + \frac{32k}{9}\mathbf{y}_\tau^{n+2/7}$$

$$s_f(\tau_{n+3/7}) = -\frac{\mathcal{M}(\tau_{n+3/7}) - a_1K}{a_2} + O(h^5), \quad s'_f(\tau_{n+2/7}) = -\frac{\mathcal{M}'(\tau_{n+3/7})}{a_2} + O(h^5);$$

$$u(\tau_{n+3/7}, x_0) = K - s_f(\tau_{n+3/7}), \quad w(\tau_{n+3/7}, x_0) = -s_f(\tau_{n+3/7})$$

$$\mathcal{M}(\tau_{n+3/7}) = 6u(\tau_{n+3/7}, x_1) - \frac{3}{2}u(\tau_{n+3/7}, x_2) + \frac{2}{9}u(\tau_{n+3/7}, x_3),$$

$$\zeta_{m(n+3/7)} = mr + \frac{s'_f(\tau_{n+3/7})}{s_f(\tau_{n+3/7})} - m\frac{\sigma^2}{2}$$

Fourth stage:

$$\mathbf{u}_\tau^{n+3/7} = \frac{\bar{\sigma}^2}{2}A_u^{-1}\left(B_u\mathbf{u}^{n+3/7} + \mathbf{b}_u^{n+3/7}\right) + \zeta_{n+3/7}\mathbf{w}^{n+3/7} - r\mathbf{u}^{n+3/7} + O(h^4)$$

$$\mathbf{u}^{n+4/7} = \mathbf{u}^n + \frac{19732k}{6561}\mathbf{u}_\tau^n - \frac{25360k}{2187}\mathbf{u}_\tau^{n+1/7} + \frac{64448k}{6561}\mathbf{u}_\tau^{n+2/7} - \frac{212k}{729}\mathbf{u}_\tau^{n+3/7}$$

$$\mathbf{w}_\tau^{n+3/7} = \frac{\sigma^2}{2}A_{w,y}^{-1}\left(B_{w,y}\mathbf{w}^{n+3/7} + \mathbf{b}_w^{n+3/7}\right) + \zeta_{m(n+3/7)}A_u^{-1}\left(B_u\mathbf{u}^{n+3/7} + \mathbf{b}_u^{n+3/7}\right) - r\mathbf{w}^{n+3/7} + O(h^4)$$

$$\mathbf{w}^{n+4/7} = \mathbf{w}^n + \frac{19732k}{6561}\mathbf{w}_\tau^n - \frac{25360k}{2187}\mathbf{w}_\tau^{n+1/7} + \frac{64448k}{6561}\mathbf{w}_\tau^{n+2/7} - \frac{212k}{729}\mathbf{w}_\tau^{n+3/7}$$

$$\mathbf{y}_\tau^{n+3/7} = \frac{\bar{\sigma}^2}{2}A_{w,y}^{-1}\left(B_{w,y}\mathbf{y}^{n+3/7} + \mathbf{b}_y^{n+3/7}\right) + \zeta_{m(n+3/7)}A_{w,y}^{-1}\left(B_{w,y}\mathbf{w}^{n+3/7} + \mathbf{b}_w^{n+3/7}\right) - r\mathbf{y}^{n+3/7} + O(h^4)$$

$$\mathbf{y}^{n+4/7} = \mathbf{y}^n + \frac{19732k}{6561}\mathbf{y}_\tau^n - \frac{25360k}{2187}\mathbf{y}_\tau^{n+1/7} + \frac{64448k}{6561}\mathbf{y}_\tau^{n+2/7} - \frac{212k}{729}\mathbf{y}_\tau^{n+3/7}$$

$$s_f(\tau_{n+4/7}) = -\frac{\mathcal{M}(\tau_{n+4/7}) - a_1K}{a_2} + O(h^5), \quad s'_f(\tau_{n+4/7}) = -\frac{\mathcal{M}'(\tau_{n+4/7})}{a_2} + O(h^5)$$

$$u(\tau_{n+4/7}, x_0) = K - s_f(\tau_{n+4/7}), \quad w(\tau_{n+4/7}, x_0) = -s_f(\tau_{n+4/7})$$

$$\mathcal{M}(\tau_{n+4/7}) = 6u(\tau_{n+4/7}, x_1) - \frac{3}{2}u(\tau_{n+4/7}, x_2) + \frac{2}{9}u(\tau_{n+4/7}, x_3),$$

$$\zeta_{m(n+4/7)} = mr + \frac{s'_f(\tau_{n+4/7})}{s_f(\tau_{n+4/7})} - m\frac{\sigma^2}{2}$$

Fifth stage:

$$\begin{aligned}
 \mathbf{u}_\tau^{n+4/7} &= \frac{\bar{\sigma}^2}{2} A_u^{-1} \left(B_u \mathbf{u}^{n+4/7} + \mathbf{b}_u^{n+4/7} \right) + \zeta_{m(n+4/7)} \mathbf{w}^{n+4/7} - r \mathbf{u}^{n+4/7} + O(h^4) \\
 \mathbf{u}^{n+5/7} &= \mathbf{u}^n + \frac{9017k}{3168} \mathbf{u}_\tau^n - \frac{355k}{33} \mathbf{u}_\tau^{n+1/7} + \frac{46732k}{5247} \mathbf{u}_\tau^{n+2/7} + \frac{49k}{176} \mathbf{u}_\tau^{n+3/7} - \frac{5103k}{18656} \mathbf{u}_\tau^{n+4/7} \\
 \mathbf{w}_\tau^{n+4/7} &= \frac{\bar{\sigma}^2}{2} A_{w,y}^{-1} \left(B_{w,y} \mathbf{w}^{n+4/7} + \mathbf{b}_w^{n+4/7} \right) + \zeta_{m(n+4/7)} A_u^{-1} \left(B_u \mathbf{u}^{n+4/7} + \mathbf{b}_u^{n+4/7} \right) \\
 &\quad - r \mathbf{w}^{n+4/7} + O(h^4) \\
 \mathbf{w}^{n+5/7} &= \mathbf{w}^n + \frac{9017k}{3168} \mathbf{w}_\tau^n - \frac{355k}{33} \mathbf{w}_\tau^{n+1/7} + \frac{46732k}{5247} \mathbf{w}_\tau^{n+2/7} + \frac{49k}{176} \mathbf{w}_\tau^{n+3/7} - \frac{5103k}{18656} \mathbf{w}_\tau^{n+4/7} \\
 \mathbf{y}_\tau^{n+4/7} &= \frac{\bar{\sigma}^2}{2} A_{w,y}^{-1} \left(B_{w,y} \mathbf{y}^{n+4/7} + \mathbf{b}_y^{n+4/7} \right) + \zeta_{m(n+4/7)} A_{w,y}^{-1} \left(B_{w,y} \mathbf{w}^{n+4/7} + \mathbf{b}_w^{n+4/7} \right) \\
 &\quad - r \mathbf{y}^{n+4/7} + O(h^4) \\
 \mathbf{y}^{n+5/7} &= \mathbf{y}^n + \frac{9017k}{3168} \mathbf{y}_\tau^n - \frac{355k}{33} \mathbf{y}_\tau^{n+1/7} + \frac{46732k}{5247} \mathbf{y}_\tau^{n+2/7} + \frac{49k}{176} \mathbf{y}_\tau^{n+3/7} - \frac{5103k}{18656} \mathbf{y}_\tau^{n+4/7} \\
 s_f(\tau_{n+5/7}) &= -\frac{\mathcal{M}(\tau_{n+5/7}) - a_1 K}{a_2} + O(h^5), \quad s'_f(\tau_{n+5/7}) = -\frac{\mathcal{M}'(\tau_{n+5/7})}{a_2} + O(h^5) \\
 u(\tau_{n+5/7}, x_0) &= K - s_f(\tau_{n+5/7}), \quad w(\tau_{n+5/7}, x_0) = -s_f(\tau_{n+5/7}) \\
 \mathcal{M}(\tau_{n+5/7}) &= 6u(\tau_{n+5/7}, x_1) - \frac{3}{2}u(\tau_{n+5/7}, x_2) + \frac{2}{9}u(\tau_{n+5/7}, x_3) \\
 \zeta_{m(n+5/7)} &= mr + \frac{s'_f(\tau_{n+5/7})}{s_f(\tau_{n+5/7})} - m \frac{\bar{\sigma}^2}{2}
 \end{aligned}$$

Sixth stage:

$$\begin{aligned}
 \mathbf{u}_\tau^{n+5/7} &= \frac{\bar{\sigma}^2}{2} A_u^{-1} \left(B_u \mathbf{u}^{n+5/7} + \mathbf{b}_u^{n+5/7} \right) + \zeta_{m(n+5/7)} \mathbf{w}^{n+5/7} - r \mathbf{u}^{n+5/7} + O(h^4) \\
 \mathbf{u}^{n+6/7} &= \mathbf{u}^n + \frac{35k}{384} \mathbf{u}_\tau^n + \frac{500k}{1113} \mathbf{u}_\tau^{n+1/7} + \frac{125k}{192} \mathbf{u}_\tau^{n+2/7} - \frac{2187k}{6784} \mathbf{u}_\tau^{n+3/7} + \frac{11k}{84} \mathbf{u}_\tau^{n+4/7} \\
 \mathbf{w}_\tau^{n+5/7} &= \frac{\bar{\sigma}^2}{2} A_{w,y}^{-1} \left(B_{w,y} \mathbf{w}^{n+5/7} + \mathbf{b}_w^{n+5/7} \right) + \zeta_{m(n+5/7)} A_u^{-1} \left(B_u \mathbf{u}^{n+5/7} + \mathbf{b}_u^{n+5/7} \right) \\
 &\quad - r \mathbf{w}^{n+5/7} + O(h^4) \\
 \mathbf{w}^{n+6/7} &= \mathbf{w}^n + \frac{35k}{384} \mathbf{w}_\tau^n + \frac{500k}{1113} \mathbf{w}_\tau^{n+1/7} + \frac{125k}{192} \mathbf{w}_\tau^{n+2/7} - \frac{2187k}{6784} \mathbf{w}_\tau^{n+3/7} + \frac{11k}{84} \mathbf{w}_\tau^{n+4/7} \\
 \mathbf{y}_\tau^{n+5/7} &= \frac{\bar{\sigma}^2}{2} A_{w,y}^{-1} \left(B_{w,y} \mathbf{y}^{n+5/7} + \mathbf{b}_y^{n+5/7} \right) + \zeta_{m(n+5/7)} A_{w,y}^{-1} \left(B_{w,y} \mathbf{w}^{n+5/7} + \mathbf{b}_w^{n+5/7} \right) \\
 &\quad - r \mathbf{y}^{n+5/7} + O(h^4)
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{y}^{n+6/7} &= \mathbf{y}^n + \frac{35k}{384}\mathbf{y}_\tau^n + \frac{500k}{1113}\mathbf{y}_\tau^{n+1/7} + \frac{125k}{192}\mathbf{y}_\tau^{n+2/7} - \frac{2187k}{6784}\mathbf{y}_\tau^{n+3/7} + \frac{11k}{84}\mathbf{y}_\tau^{n+4/7} \\
 s_f(\tau_{n+6/7}) &= -\frac{\mathcal{M}(\tau_{n+6/7}) - a_1K}{a_2} + O(h^5), \quad s'_f(\tau_{n+6/7}) = -\frac{\mathcal{M}'(\tau_{n+6/7})}{a_2} + O(h^5) \\
 u(\tau_{n+6/7}, x_0) &= K - s_f(\tau_{n+6/7}), \quad w(\tau_{n+6/7}, x_0) = -s_f(\tau_{n+6/7}) \\
 \mathcal{M}(\tau_{n+6/7}) &= 6u(\tau_{n+6/7}, x_1) - \frac{3}{2}u(\tau_{n+6/7}, x_2) + \frac{2}{9}u(\tau_{n+6/7}, x_3) \\
 \xi_{m(n+6/7)} &= mr + \frac{s'_f(\tau_{n+6/7})}{s_f(\tau_{n+6/7})} - m\frac{\sigma^2}{2}
 \end{aligned}$$

Seventh stage:

$$\begin{aligned}
 s_f(\tau_{n+1}) &= s_f(\tau_{n+6/7}), \quad \mathbf{u}^{n+1} = \mathbf{u}^{n+6/7}, \quad \mathbf{w}^{n+1} = \mathbf{w}^{n+6/7}, \quad \mathbf{y}^{n+1} = \mathbf{y}^{n+6/7} \\
 \mathbf{u}_\tau^{n+6/7} &= \frac{\bar{\sigma}^2}{2}A_u^{-1}\left(B_u\mathbf{u}^{n+6/7} + \mathbf{b}_u^{n+6/7}\right) + \xi_{m(n+6/7)}\mathbf{w}^{n+6/7} - r\mathbf{u}^{n+6/7} + O(h^4) \\
 \bar{\mathbf{u}}^{n+1} &= \mathbf{u}^n + \frac{5179k}{57600}\mathbf{u}_\tau^n + \frac{7571k}{16695}\mathbf{u}_\tau^{n+2/7} + \frac{393k}{640}\mathbf{u}_\tau^{n+3/7} - \frac{92097k}{339200}\mathbf{u}_\tau^{n+4/7} + \frac{187k}{2100}\mathbf{u}_\tau^{n+5/7} \\
 &\quad + \frac{k}{40}\mathbf{u}_\tau^{n+6/7} \\
 \mathbf{w}_\tau^{n+6/7} &= \frac{\bar{\sigma}^2}{2}A_{w,y}^{-1}\left(B_{w,y}\mathbf{w}^{n+6/7} + \mathbf{b}_w^{n+6/7}\right) + \xi_{m(n+6/7)}A_u^{-1}\left(B_u\mathbf{u}^{n+6/7} + \mathbf{b}_u^{n+6/7}\right) \\
 &\quad - r\mathbf{w}^{n+6/7} + O(h^4) \\
 \bar{\mathbf{w}}^{n+1} &= \mathbf{w}^n + \frac{5179k}{57600}\mathbf{w}_\tau^n + \frac{7571k}{16695}\mathbf{w}_\tau^{n+2/7} + \frac{393k}{640}\mathbf{w}_\tau^{n+3/7} - \frac{92097k}{339200}\mathbf{w}_\tau^{n+4/7} \\
 &\quad + \frac{187k}{2100}\mathbf{w}_\tau^{n+5/7} + \frac{k}{40}\mathbf{w}_\tau^{n+6/7} \\
 \mathbf{y}_\tau^{n+6/7} &= \frac{\bar{\sigma}^2}{2}A_{w,y}^{-1}\left(B_{w,y}\mathbf{y}^{n+6/7} + \mathbf{b}_y^{n+6/7}\right) + \xi_{m(n+6/7)}A_{w,y}^{-1}\left(B_{w,y}\mathbf{w}^{n+6/7} + \mathbf{b}_w^{n+6/7}\right) \\
 &\quad - r\mathbf{y}^{n+6/7} + O(h^4) \\
 \bar{\mathbf{y}}^{n+1} &= \mathbf{y}^n + \frac{5179k}{57600}\mathbf{y}_\tau^n + \frac{7571k}{16695}\mathbf{y}_\tau^{n+2/7} + \frac{393k}{640}\mathbf{y}_\tau^{n+3/7} - \frac{92097k}{339200}\mathbf{y}_\tau^{n+4/7} \\
 &\quad + \frac{187k}{2100}\mathbf{y}_\tau^{n+5/7} + \frac{k}{40}\mathbf{y}_\tau^{n+6/7} \\
 \bar{s}_f(\tau_{n+1}) &= -\frac{\bar{\mathcal{M}}(\tau_{n+1}) - a_1K}{a_2} + O(h^5) \\
 \bar{u}(\tau_{n+1}, x_0) &= K - \bar{s}_f(\tau_{n+1}), \quad \bar{w}(\tau_{n+1}, x_0) = -\bar{s}_f(\tau_{n+1})
 \end{aligned}$$

$$\bar{\mathcal{M}}(\tau_{n+1}) = 6\bar{u}(\tau_{n+1}, x_1) - \frac{3}{2}\bar{u}(\tau_{n+1}, x_2) + \frac{2}{9}\bar{u}(\tau_{n+1}, x_3)$$

Here, $s_f(\tau_{n+1})$, \mathbf{u}^{n+1} , and \mathbf{w}^{n+1} are obtained from the fifth-order Runge–Kutta integration scheme, which will represent the numerical approximation of the optimal exercise boundary, option value, and hedge sensitivity. Moreover, $\bar{s}_f(\tau_{n+1})$, $\bar{\mathbf{u}}^{n+1}$, and $\bar{\mathbf{w}}^{n+1}$ are fourth-order accurate in time and will be used in conjunction with the solution from the fifth-order scheme to establish an error threshold for the selection of the optimal time step based on the regional variation and grid point distribution. For brevity, we skip further explanation. Please see the work of Nwankwo and Dai [21,24,25], where the author describes the implementation of these embedded pair(s) for solving the American options pricing model that involves coupled systems of diffusive–convective–reaction PDEs on a uniform space grid.

3. Numerical Experiment

In this section, we investigate the efficiency of our implementation and compare its performance with some of the existing methods in the literature, as well as with the results obtained with the uniform fourth-order compact scheme and 5(4) Dormand–Prince embedded pairs. Numerical experiments were carried out on a workstation with a 12th Gen Intel(R) Core(TM) i7-12700H at 2.30 GHz on a 64-bit Windows 11 operating system. Furthermore, the MATLAB programming language was used for numerical experiments and visualization. For convenience, we label our methods as follows

- DPC-Uniform: Fourth-order compact scheme with 5(4) Dormand–Prince embedded pairs on an equidistant space grid;
- DPC-Loc1: Fourth-order compact scheme with 5(4) Dormand–Prince embedded pairs on a locally refined space grid ($h_a = 0.5$ h).
- DPC-Loc2: Fourth-order compact scheme with 5(4) Dormand–Prince embedded pairs on a locally refined space grid ($h_a = 0.25$ h).
- DPC-Loc3: Fourth-order compact scheme with 5(4) Dormand–Prince embedded pairs on a locally refined space grid ($h_a = 0.125$ h).

In all the examples in this section, we conduct our experiment with very coarse grids. $h = 0.025$ is the minimum step size we use for computing our numerical solutions. Furthermore, $h_a = 0.5$ h, $h_a = 0.25$ h, and $h_a = 0.125$ h represent the smallest step size in the space grid, which represents the step size of the local uniform domain close to the left corner point in the non-uniform grids for DPC-Loc1, DPC-Loc2, and DPC-Loc3, respectively. The reason for conducting our experiment with a very coarse grid is to better understand if there are significant impacts our mesh refinement strategies could have on the solution accuracy, optimal time selection, and computational speed.

3.1. Investigating Solution Accuracy on Both Uniform and Non-Equidistant Space Grids with Unit Power Terms

In this subsection, we first consider a case for the power put option where the power term $m = 1$. When $m = 1$, we are simply investigating the vanilla American put options pricing problem. Here, we want to verify the performance of the presented method in terms of solution accuracy and computational runtime on both uniform and non-equidistant space grids with adaptive time stepping techniques. With a non-equidistant grid, we investigate the impact of local refinement and coordinate transformation on the solution accuracy and runtime in seconds using very coarse grids. To this end, we first consider the example presented in the work of Nwankwo and Dai [21] with the parameter shown in Table 1.

Table 1. First numerical data.

K	T	r	σ	x_{max}
100	6/12	5%	20%	3.00

Here, we will use the numerical data above to investigate the solution accuracy of the optimal exercise boundary using both uniform and non-equidistant grids. Our experiment for this example is conducted on very coarse grids ($h = 0.05$ and $h = 0.025$). The plot profile of the optimal exercise display is shown in Figure 2. In Tables 2 and 3, we present the numerical approximation when $\tau = T$ with a runtime in seconds. Here, the runtime in seconds is defined as the time required to run the whole code from initialization to visualization. The benchmark solution is obtained from a sixth-order compact finite difference scheme with a third-order Runge–Kutta adaptive time integration method and free boundary improvement [25]. A step size of $h = 0.01$ is used to obtain the benchmark value.

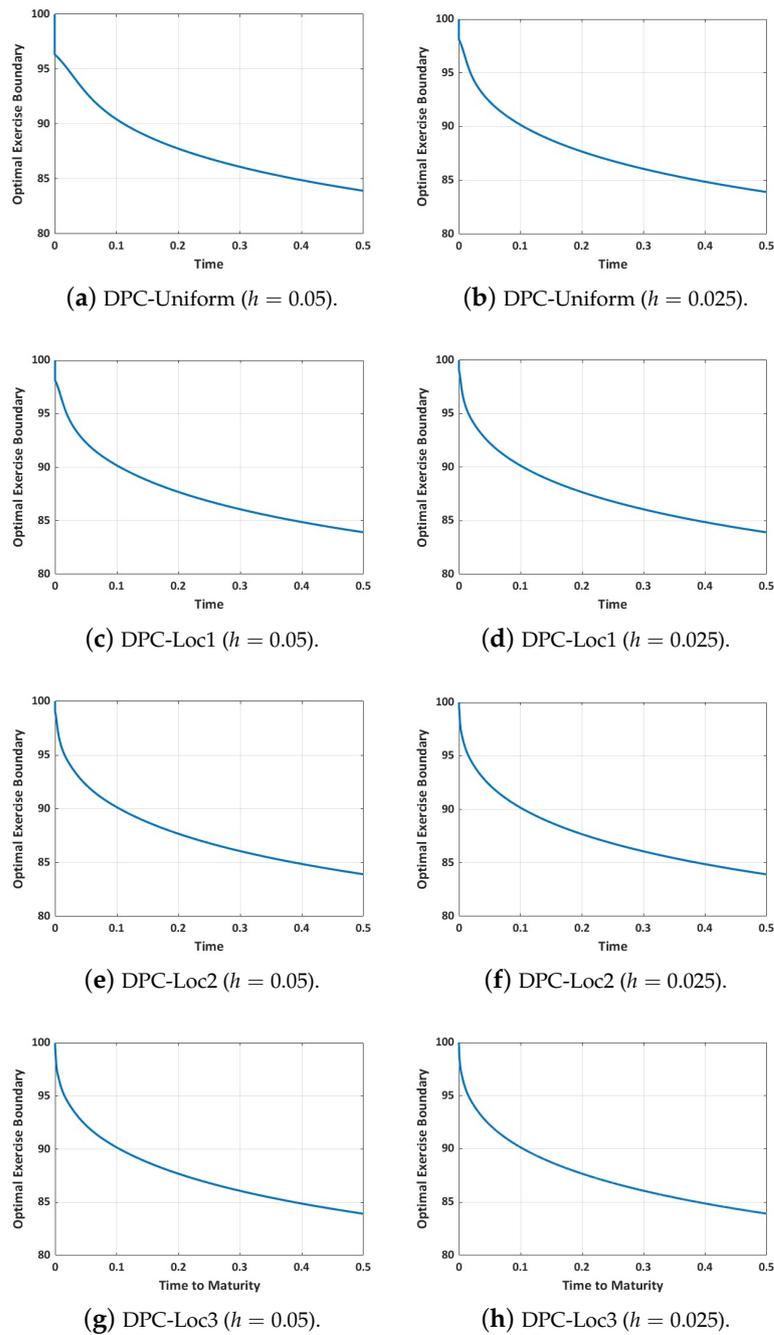


Figure 2. Plot of the optimal exercise boundary with uniform and non-uniform Hermitian scheme $\epsilon = 10^{-3}$.

Table 2. Comparing the optimal exercise boundary and runtime in seconds ($\epsilon = 10^{-3}$). $s_f(T) = 83.91996$ is the benchmark value.

h	DPC-Loc1	DPC-Loc2	DPC-Loc3	DPC-Uniform
0.05	83.92016	83.92055	83.92042	83.92703
0.025	83.92008	83.91988	83.91985	83.91961

Table 3. Comparing the runtime in seconds.

h	DPC-Loc1	DPC-Loc2	DPC-Loc3	DPC-Uniform
0.05	0.186678	0.355253	1.161236	0.279023
0.025	1.059206	2.429464	8.883537	0.628332

We observe from Figure 2 the importance of the local mesh refinement as it improves accuracy close to the left corner point where $x = 0$ and $\tau = 0$. In Figure 2, for instance, with a uniform grid, when $h = 0.05$, the optimal exercise boundary is not very smooth close to the payoff. However, with locally refined grids, especially with DPC-Loc2 and DP-Loc3, we observe that the optimal exercise boundary is very smooth with $h = 0.05$, thus establishing the importance of refining the grid at the left corner point. Moreover, the scheme with a locally refined grid, as shown in Table 4, presents the most accurate result when compared with the one obtained from the uniform grid. Another advantage of the local mesh refinement as presented here is that only a few grid points are introduced while retaining the uniformity of the grid in most regions.

Next, we establish that our proposed methods can be implemented with arbitrary step sizes and time steps, which is an important feature of the Runge–Kutta embedded time integration method. To this end, we fix $h = 0.025$ and initialize with varying initial time steps $k_0 = 6.25$ and 0.000625 . We display the result in Table 4.

Table 4. Initializing with varying initial time step k_0 and fixed step size ($h = 0.025$ and $\epsilon = 10^{-3}$).

k_0	DPC-Loc1	DPC-Loc2	DPC-Loc3	DPC-Uniform
6.25	83.92008	83.91988	83.91985	83.91961
0.000625	83.92008	83.91988	83.91985	83.91961

We observe from Table 4 that our proposed method is well-adapted to fixed step size h and to varying initial time step k_0 , and that this provides consistent and reasonable approximation for both large and small time steps. This is because the time integration scheme selects an optimal time step at each time level adaptively based on regional variation and grid point distribution. Hence, the choice of initialization for the time step does not matter and has no significant implication.

Finally, for this example, to better understand the impact of the grid distribution on the optimal time selection at each time level, we display the time stepping profile based on the uniform and non-equidistant grid distribution in Figure 3 with a step size of $h = 0.025$ and $\epsilon = 10^{-4}$. We observe that the time stepping is very small close to the payoff, which is expected based on the adaptability of our numerical scheme and the inherent irregularity that is more pronounced close to the payoff. Furthermore, we observe from Figure 3 that if we introduce more grid points in the locality of the left corner point, as can be observed from DPC-Loc1, DPC-Loc2, and DPC-Loc3, the time step for a fixed time level is smaller when compared with the uniform space grid distribution. However, from Table 3, we observe that the overall impact on the total runtime is insignificant when compared with the solution accuracy.

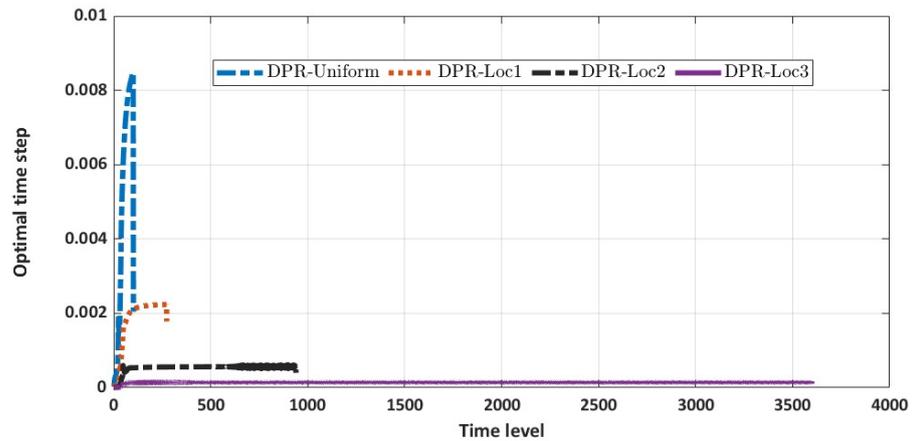


Figure 3. Plot of the optimal time stepping for each time level ($h = 0.025$ and $\epsilon = 10^{-3}$).

In the second example, we will focus on option value and delta sensitivity. Here, we consider the example in the work of Bunch and Johnson [26] and Gutiérrez [27] with the following parameters displayed in Table 5.

Table 5. Second numerical data.

K	T	r	σ	x_{max}	ϵ
35	4/12	4.88%	30%	3.00	10^{-4}

The benchmark value used in their work was obtained from the binomial tree method with 10,000 steps. Here, we consider the numerical solution of the option value and delta sensitivity with a very coarse grid ($h = 0.05$) and compare the performance of the uniform and non-equidistant space-grid implementation with the benchmark value. We display the results in Tables 6 and 7.

Table 6. Comparing the option value with varying strike price ($h = 0.05, S_0 = 40$).

K	DPC-Uniform	DPC Loc1	DPC Loc2	DPC Loc3	Binomial Method
35	0.6944	0.6967	0.6974	0.6976	0.6975
40	2.4782	2.4815	2.4823	2.4825	2.4825
45	5.7028	5.7050	5.7055	5.7057	5.7056

Table 7. Comparing the delta sensitivity with varying strike price ($h = 0.05, S_0 = 40$).

K	DPC-Uniform	DPC Loc1	DPC Loc2	DPC Loc3	Binomial Method
35	-0.1738	-0.1740	-0.1741	-0.1741	-0.1741
40	-0.4422	-0.4420	-0.4420	-0.4420	-0.4420
45	-0.7272	-0.7268	-0.7267	-0.7266	-0.7266

From Tables 6 and 7, we observe reasonable solution accuracy for both the option value and delta sensitivity when compared with the binomial method. The interesting part is that reasonable solution accuracy was obtained with the non-equidistant space grid, which is closer to the benchmark value when compared with the results from the uniform grid. Achieving such solution accuracy on very coarse grids as described in Tables 6 and 7 further validates the importance of our local mesh refinement strategy.

Remark 2. It is worth mentioning that the approximation in the first and second examples involves only two system of PDEs in (17) and (18) consisting of the option value and delta sensitivity. The error threshold for selecting the optimal time step is established with the difference in the solution of the optimal exercise boundary, option value, and delta sensitivity for these examples based on the Runge–Kutta pairs. Moreover, the reported runtime in seconds only accounts for both PDEs. In the last example below, we consider the three systems of PDEs in (17)–(19) when accounting for gamma.

In this last example for this subsection, we investigate the importance of our proposed method in approximating gamma. Consider the example in the work of Tangman et al. [28] given in Table 8.

Table 8. Third numerical data.

K	T	r	σ	x_{max}
100	3	5%	20%	3.00

The differential and semi-discrete equations in (17), (18), (59), and (60) governing the option value and delta sensitivity are not coupled with that of their gamma counterpart. Hence, the investigation for gamma is done separately in this example. Here, we include the three PDEs in (17)–(19) and the three semi-discrete PDEs in (59)–(61). For the error threshold, which we establish for the optimal time step at each time level, we use maximum error obtained from the difference of the solutions of the optimal exercise boundary, option value, delta sensitivity, and gamma from the Runge–Kutta pairs as follows:

$$e_u = \|\mathbf{u}^{n+1} - \bar{\mathbf{u}}^{n+1}\|_\infty, \quad e_{s_f} = |s_f^{n+1} - \bar{s}_f^{n+1}|; \tag{63}$$

$$e_w = \|\mathbf{w}^{n+1} - \bar{\mathbf{w}}^{n+1}\|_\infty, \quad e_y = \|\mathbf{y}^{n+1} - \bar{\mathbf{y}}^{n+1}\|_\infty, \quad e = \max\{e_{s_f}, e_u, e_w, e_y\}; \tag{64}$$

$$k_{new}^{n+1} = \begin{cases} \rho \left(\frac{\epsilon}{e}\right)^{\frac{1}{4}} k_{old}^{n+1}, & \text{if } e < \epsilon, \\ \rho \left(\frac{\epsilon}{e}\right)^{\frac{1}{5}} k_{old}^{n+1}, & \text{if } e \geq \epsilon. \end{cases} \tag{65}$$

Here, $0 < \rho < 1$. The optimal time step is selected if $e < \epsilon$ and if the true solutions are \mathbf{u}^{n+1} , \mathbf{w}^{n+1} , \mathbf{y}^{n+1} , and s_f^{n+1} are selected. We observe that including the error from gamma in establishing the optimal time step ensures a stable and non-oscillatory solution for gamma. The important part, as we see in the result below, is that we can achieve reasonable solution accuracy with a large tolerance and very coarse grids, provided we include the error term from gamma when accounting for the optimal time step. Our main interest for this last example is to understand if we can achieve reasonable accuracy and little computational runtime with our local grid refinement. Furthermore, we also want to investigate how the local grid refinement impacts the solution accuracy and runtime in seconds. Here, our runtime in seconds includes the time it takes to run the whole code (approximate the early exercise boundary, option value, delta sensitivity, and gamma), excluding initialization to visualization.

To this end, we present the solution accuracy of the gamma obtained with our proposed method and compare our result from the non-equidistant space grid with the binomial tree method [29], which serves as benchmark value. For the non-equidistant grid, we consider DPC-Loc2. The binomial tree method is obtained with 15,001 steps as reported in the work of Tangman et al. [28]. Because of the high accuracy we observed from the gamma solution, we only consider very coarse grids with $h = 0.2, 0.1, \text{ and } 0.05$. The results are displayed in Tables 9–11.

Table 9. Gamma solution with DPC-Loc1 ($\epsilon = 10^{-2}$).

S	$h = 0.2$	$h = 0.1$	$h = 0.05$	Binomial Method
80	0.0138	0.0340	0.0361	0.0361
90	0.0257	0.0236	0.0234	0.0234
100	0.0147	0.0153	0.0154	0.0155
110	0.0104	0.0103	0.0102	0.0103
120	0.0064	0.0067	0.0068	0.0068
Runtime (s)	0.2651	0.5294	1.2769	None

Table 10. Gamma solution with DPC-Loc2 ($\epsilon = 10^{-2}$).

S	$h = 0.2$	$h = 0.1$	$h = 0.05$	Binomial Method
80	0.0340	0.0361	0.0361	0.0361
90	0.0236	0.0234	0.0234	0.0234
100	0.0153	0.0154	0.0154	0.0155
110	0.0103	0.0103	0.0103	0.0103
120	0.0067	0.0068	0.0068	0.0068
Runtime (s)	0.4810	0.7951	3.4038	None

Table 11. Gamma solution with DPC-Loc3 ($\epsilon = 10^{-2}$).

S	$h = 0.2$	$h = 0.1$	$h = 0.05$	Binomial Method
80	0.0361	0.0361	0.0361	0.0361
90	0.0234	0.0234	0.0234	0.0234
100	0.0154	0.0154	0.0155	0.0155
110	0.0102	0.0103	0.0103	0.0103
120	0.0067	0.0068	0.0068	0.0068
Runtime (s)	0.5354	1.7565	9.5990	None

The importance of our local mesh refinement is very visible in Tables 9–11. We observe that our solution accuracy is very close to the benchmark with a very coarse grid. More importantly, however, as we introduce few more grid points to the grid from DPC-Loc1–DPC-Loc3, the solution accuracy improve substantially. It is easy to see that we have achieved a result that is the same in most cases as the benchmark value, even with a very coarse grid ($h = 0.2$) using DPC-Loc3. This important gain will be very useful in high-dimensional models due to space complexity. In general, very little computational time is required to approximate the early exercise boundary, option value, delta sensitivity, and gamma simultaneously with the locally refined grid.

Finally, to see the importance of including the error term e_y in the optimal time step equation presented in (63)–(65), we call the implementation in the latter Scenario 1. For Scenario 2, we remove the error term for gamma e_y from (63)–(65) and compute the gamma. To ensure an adequate result comparison for Scenario 2, we compute the result with very small and very large ϵ . The results are displayed in Tables 12 and 13.

Table 12. Gamma solution with DPC-Loc3 ($\epsilon = 10^{-2}, h = 0.1$).

S	Scenario 1 (with e_y)	Scenario 2 (without e_y)	Binomial Method
80	0.0361	-2.548×10^{40}	0.0361
90	0.0234	-9.047×10^{39}	0.0234
100	0.0154	-4.343×10^{39}	0.0155
110	0.0103	-1.555×10^{39}	0.0103
120	0.0068	-6.230×10^{38}	0.0068

Table 13. Gamma solution with DPC-Loc3 ($\epsilon = 10^{-6}, h = 0.1$).

S	Scenario 1 (with e_y)	Scenario 2 (without e_y)	Binomial Method
80	0.0361	0.0361	0.0361
90	0.0234	0.0234	0.0234
100	0.0154	0.0154	0.0155
110	0.0103	0.0103	0.0103
120	0.0068	0.0068	0.0068

From Tables 12 and 13, we observe that the solution accuracy for Scenario 2 is quite unreasonable with $\epsilon = 10^{-2}$ when compared with Scenario 1. However, the result of Scenario 2 starts stabilizing as we reduce the tolerance, which can be observed with $\epsilon = 10^{-6}$. Moreover, increasing the tolerance ϵ will impact the computational cost. Hence, we can conclude that, to enable better estimation of the optimal exercise boundary, option value, delta sensitivity, and gamma from the three systems of PDEs consisting of the option value, delta sensitivity, and gamma, it might be ideal to include the error term associated with the four solutions as described in (63)–(65).

3.2. Investigating Solution Accuracy on both Uniform and Non-Equidistant Space Grids with an Arbitrary Power Term

Here, we consider the power put options pricing problem with an arbitrary power term m and also consider the three systems of fixed free boundary PDEs in (17)–(19) and their discrete systems in (59)–(61). We verify the performance of our proposed method for solving such a model. By implementing an adaptive scheme, we have circumvented the stability challenges. We now focus more on understanding how varying m impacts the adaptive selection of time steps and the local mesh refinement strategy while presenting the numerical solutions. We carry out the whole experiment here with DPC-Loc2 because it presented more optimality in terms of computational time and solution accuracy in the previous subsection. Further, our experiment here is done with a very coarse grid $h = 0.025$ and $\epsilon = 10^{-3}$.

Few research studies exist that solve free boundary power options. Consider the example in the work of Lee [7] with the parameters as described in Table 14 below.

Table 14. Fourth numerical data.

K	T	r	σ	x_{max}
100	6/12	8%	10%	3.00

First, we compute the optimal exercise boundary for varying power term $m = 1, 2, 3, 4,$ and 5 , similar to the one presented in Lee’s [7] paper, and display the results in Figure 4. We observe that, for a fixed time, the optimal exercise boundary decreases as the power term m increases. This is further reflected in the value of the optimal exercise boundary with varying power term m as presented in Table 15. Our plot also closely resembles the one presented in the work of Lee [7].

Table 15. Optimal exercise boundary with varying power term m ($h = 0.025, h_a = 0.25$ h).

Parameter	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
$s_f(T)$	95.17035	90.80024	86.82542	83.19370	79.86161

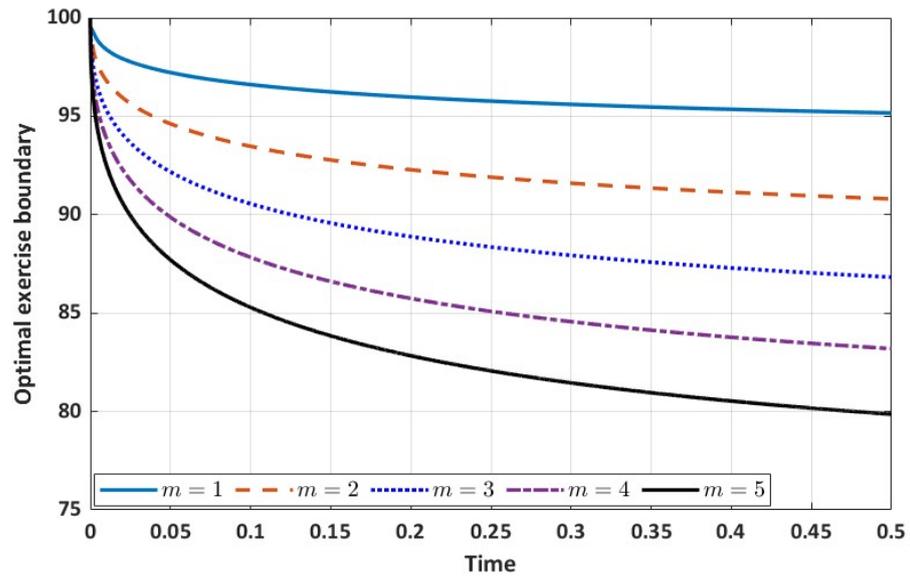


Figure 4. Optimal exercise boundary with varying power term m .

Next, we present the plot profile of the time step versus time level for each considered power term m . The plot is displayed in Figure 5. The optimal time step at each time level decreases substantially as the power term increases. This is expected because of the m^2 associated with the diffusive term. This power term m may impact the stability condition if a non-adaptive scheme is implemented.

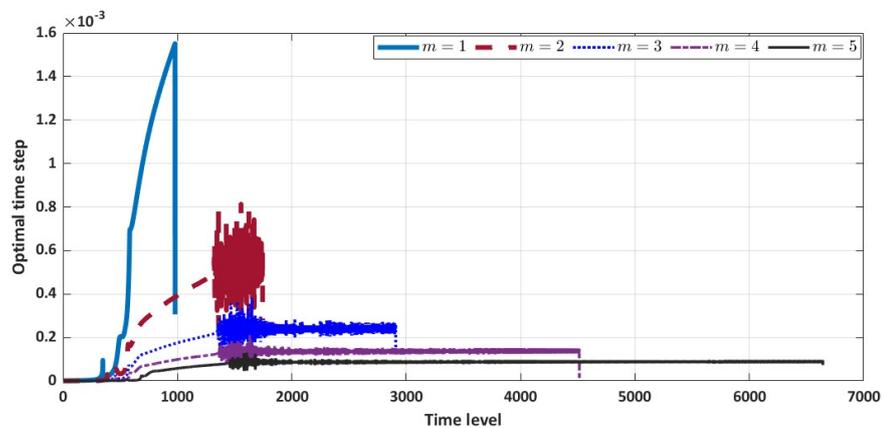


Figure 5. Optimal time step for each time level with varying power m ($h = 0.025, h_a = 0.25$ h).

Finally, we display the plot profile and the numerical solutions of the option value and delta sensitivity for each power term m , as presented in Figure 6. For a fixed asset price, Figure 6 shows that the option value increases as the power term m increases. This is also reflected in the numerical solution presented in Table 16. Lee [7] describes that this feature given by the exotic options provides the buyer with the possibility of receiving a reasonably higher payoff than its vanilla counterpart. Moreover, in the money, the delta sensitivity decreases as the power term increases. Furthermore, out of the money, delta sensitivity increases as the power term increases. This is fully described in Table 17. Additionally, the numerical solution for gamma at some interpolated grid points is also listed in Table 18.

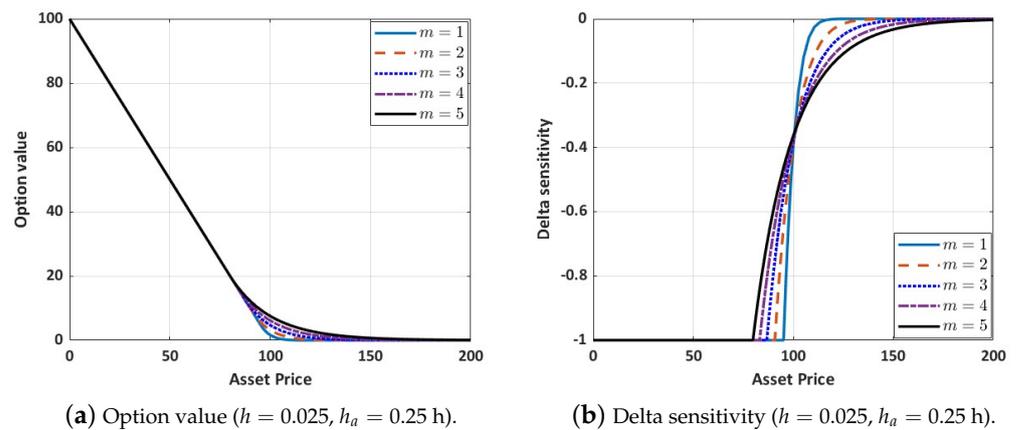


Figure 6. Option value and delta sensitivity with varying power term m .

Table 16. Option values with varying power term m ($h = 0.025, h_a = 0.25$ h).

S	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
80	20.00000	20.00000	20.00000	20.00000	20.00051
100	1.63347	3.19848	4.70042	6.14327	7.53094
120	0.00185	0.20491	0.85746	1.79875	2.88475

Table 17. Delta sensitivity with varying power term m ($h = 0.025, h_a = 0.25$ h).

S	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
80	-1.00000	-1.00000	-1.00000	-1.00000	-0.99282
100	-0.39823	-0.38896	-0.38013	-0.37174	-0.36375
120	-0.00080	-0.03126	-0.07634	-0.11197	-0.13754

Table 18. Gamma with varying power term m ($h = 0.025, h_a = 0.25$ h).

S	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
80	0.00000	0.00000	0.00000	0.00000	0.00947
100	0.08271	0.04223	0.02870	0.02192	0.01784
120	0.00033	0.00446	0.00647	0.00683	0.00664

4. Conclusions

We have proposed a high-order adaptive numerical scheme for approximating a free boundary exotic power put options pricing problem on a non-uniform space grid for which a locally adopted mesh refinement strategy is implemented. Our proposed implementation leverages some advantages. By selecting time steps adaptively at each time level, we circumvent the stability challenges in an explicit time integration scheme, which could be more pronounced here due to the coefficient of the diffusive term. This is because the coefficient of the diffusive term involves m^2 , which can substantially vary and impact any stability criterion. However, the adaptive nature of our proposed method worked in our favor here. More importantly, by refining our space grid locally and allowing time steps to be adaptively selected based on regional variation and space grid distribution, the solution accuracy in the neighborhood of the left corner point, where large variation, discontinuity, and even singularity exist in the pricing model, was substantially improved.

The importance of our implementation can further be observed in the manner in which we recover highly accurate results with very coarse grids using the locally refined grid

strategy at each time level. This gain will be very useful in a high-dimensional pricing model due to space complexity. We have already established in the numerical experiment that, even with a very coarse grid ($h = 0.2$), a reasonable solution accuracy was achieved with gamma when we concentrated more of the grid points in the locality of the left corner point.

In our future work, we hope to further extend our proposed method to include a suite of space grid stretching strategies and, possibly, a pure high-order adaptive space–time scheme for pricing free boundary Asian stochastic volatility models and other non-standard options. In the context of a pure high-order adaptive space–time scheme, the grid points in both the space and time grids will be dynamically selected and updated.

Author Contributions: Conceptualization, C.N. and W.D.; methodology, C.N.; software, C.N.; validation, C.N.; formal analysis, C.N.; investigation, C.N.; resources, C.N.; data curation, C.N.; writing—original draft preparation, C.N.; writing—review and editing, W.D.; visualization, C.N.; supervision, W.D.; project administration, W.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The first author is funded in part by an NSERC Discovery Grant.

Conflicts of Interest: The authors declare no conflict of interest.

Code Availability: The MATLAB code for this work is freely available in GitHub through this link: https://github.com/Ifenonso/exotic/blob/main/Dormand_Prince_Refined_space_Adaptive_time_Exotic_Power_Options.m, (accessed on 8 June 2023).

References

- Dorm, J.R.; Prince, P.J. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.* **1980**, *6*, 19–26.
- Kim, J. Pricing of power options under the regime-switching model. *J. Appl. Math. Inform.* **2014**, *32*, 665–673. [[CrossRef](#)]
- Lee, Y.; Kim, Y.; Lee, J. Pricing various types of power options under stochastic volatility. *Symmetry* **2020**, *12*, 1911. [[CrossRef](#)]
- Topper, J. Finite element modeling of exotic options. In *Operations Research Proceedings 1999*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 336–341.
- Heynen, R.C.; Kat, H.M.. Pricing and hedging power options. *Financ. Eng. Jpn. Mark.* **1996**, *3*, 253–261. [[CrossRef](#)]
- Nwozo, C.R.; Fadugba, S.E. Mellin transform method for the valuation of some vanilla power options with non-dividend yield. *Int. J. Pure Appl. Math.* **2014**, *96*, 79–104. [[CrossRef](#)]
- Lee, J.K. A simple numerical method for pricing American power put options. *Chaos Solitons Fractals* **2020**, *139*, 110254. [[CrossRef](#)]
- Wang, J.Y.; Wang, C.J.; Dai, T.S.; Chen, T.C.; Liu, L.C.; Zhou, L. Efficient and robust combinatorial option pricing algorithms on the trinomial lattice for polynomial and barrier options. *Math. Probl. Eng.* **2022**, *2022*, 5843491. [[CrossRef](#)]
- Huang, C.S.; O’Hara, J.G.; Mataramvura, S. Highly efficient Shannon wavelet-based pricing of power options under the double exponential jump framework with stochastic jump intensity and volatility. *Appl. Math. Comput.* **2022**, *414*, 126669. [[CrossRef](#)]
- Okelola, M.O.; Govinder, K.S.; O’Hara, J.G. Solving a partial differential equation associated with the pricing of power options with time-dependent parameters. *Math. Methods Appl. Sci.* **2015**, *38*, 2901–2910. [[CrossRef](#)]
- Rao, B.P. Pricing geometric Asian power options under mixed fractional Brownian motion environment. *Phys. A Stat. Mech. Its Appl.* **2016**, *446*, 92–99.
- Esser, A. General valuation principles for arbitrary payoffs and applications to power options under stochastic volatility. *Financ. Mark. Portf. Manag.* **2003**, *17*, 351. [[CrossRef](#)]
- Kim, J.; Kim, B.; Moon, K.S.; Wee, I.S. Valuation of power options under Heston’s stochastic volatility model. *J. Econ. Dyn. Control.* **2012**, *36*, 1796–1813. [[CrossRef](#)]
- Blenman, L.P.; Bueno-Guerrero, A.; Clark, S.P. Pricing and hedging bond power exchange options in a stochastic string term-structure model. *Risks* **2022**, *10*, 188. [[CrossRef](#)]
- Ha, M.; Kim, D.; Ahn, S.; Yoon, J.H. The valuation of timer power options with stochastic volatility. *J. Korean Soc. Ind. Appl. Math.* **2022**, *26*, 296–309.
- Shukla, R.K.; Zhong, X. Derivation of high-order compact finite difference schemes for non-uniform grid using polynomial interpolation. *J. Comput. Phys.* **2005**, *204*, 404–429. [[CrossRef](#)]
- Shukla, R.K.; Tatineni, M.; Zhong, X. Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier–Stokes equations. *J. Comput. Phys.* **2007**, *224*, 1064–1094. [[CrossRef](#)]

18. Chen, Y.; Xiao, A.; Wang, W. An IMEX-BDF2 compact scheme for pricing options under regime-switching jump-diffusion models. *Math. Methods Appl. Sci.* **2019**, *42*, 2646–2663. [[CrossRef](#)]
19. Lee, S.T.; Sun, H.W. Fourth-order compact scheme with local mesh refinement for option pricing in jump-diffusion model. *Numer. Methods Partial. Differ. Equations* **2012**, *28*, 1079–1098. [[CrossRef](#)]
20. Zhao, J.; Corless, R.M. Compact finite difference method for integro-differential equations. *Appl. Math. Comput.* **2006**, *177*, 271–288. [[CrossRef](#)]
21. Nwankwo, C.; Dai, W. An adaptive and explicit fourth order Runge–Kutta–Fehlberg method coupled with compact finite differencing for pricing American put options. *Jpn. J. Ind. Appl. Math.* **2021**, *38*, 921–946. [[CrossRef](#)]
22. Abrahamsen, D.; Fornberg, B. Solving the Korteweg-de Vries equation with Hermite-based finite differences. *Appl. Math. Comput.* **2021**, *401*, 126101. [[CrossRef](#)]
23. Dimitrov, Y.M.; Vulkov, L.G. High order finite difference schemes on non-uniform meshes for the time-fractional Black-Scholes equation. *arXiv* **2016**, arXiv:1604.05178.
24. Nwankwo, C.; Dai, W. On the efficiency of 5 (4) RK-embedded pairs with high order compact scheme and Robin boundary condition for options valuation. *Jpn. J. Ind. Appl. Math.* **2022**, *39*, 753–775. [[CrossRef](#)]
25. Nwankwo, C.; Dai, W. Sixth-order compact differencing with staggered boundary schemes and 3 (2) Bogacki-Shampine pairs for pricing free-boundary options. *arXiv* **2022**, arXiv:2207.14379.
26. Bunch, D.S.; Johnson, H. The American put option and its critical stock price. *J. Financ.* **2000**, *55*, 2333–2356. [[CrossRef](#)]
27. Gutiérrez, Ó. American option valuation using first-passage densities. *Quant. Financ.* **2013**, *13*, 1831–1843. [[CrossRef](#)]
28. Tangman, D.Y.; Gopaul, A.; Bhuruth, M. A fast high-order finite difference algorithm for pricing American options. *J. Comput. Appl. Math.* **2008**, *222*, 17–29. [[CrossRef](#)]
29. Leisen, D.P.; Reimer, M. Binomial models for option valuation-examining and improving convergence. *Appl. Math. Financ.* **1996**, *3*, 319–346. [[CrossRef](#)]

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.