

Article

Efficient Fourth-Order Weights in Kernel-Type Methods without Increasing the Stencil Size with an Application in a Time-Dependent Fractional PDE Problem

Tao Liu ¹  and Stanford Shateyi ^{2,*} 

¹ School of Mathematics and Statistics, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China; liutao@neuq.edu.cn or math.taoliu@gmail.com

² Department of Mathematics and Applied Mathematics, School of Mathematical and Natural Sciences, University of Venda, P. Bag X5050, Thohoyandou 0950, South Africa

* Correspondence: stanford.shateyi@univen.ac.za

Abstract: An effective strategy to enhance the convergence order of nodal approximations in interpolation or PDE problems is to increase the size of the stencil, albeit at the cost of increased computational burden. In this study, our goal is to improve the convergence orders for approximating the first and second derivatives of sufficiently differentiable functions using the radial basis function-generated Hermite finite-difference (RBF-HFD) scheme. By utilizing only three equally spaced points in 1D, we are able to boost the convergence rate to four. Extensive tests have been conducted to demonstrate the effectiveness of the proposed theoretical weighting coefficients in solving interpolation and PDE problems.

Keywords: radial basis function (RBF); convergence order; Hermite finite difference (HFD); analytical weights; fractional PDE

MSC: 41A25; 65M22; 35R11



Citation: Liu, T.; Shateyi, S. Efficient Fourth-Order Weights in Kernel-Type Methods without Increasing the Stencil Size with an Application in a Time-Dependent Fractional PDE Problem. *Mathematics* **2024**, *12*, 1121. <https://doi.org/10.3390/math12071121>

Academic Editors: M. Filomena Teodoro and Marina Alexandra Pedro Andrade

Received: 19 March 2024

Revised: 2 April 2024

Accepted: 6 April 2024

Published: 8 April 2024



Copyright: © 2024 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/>).

1. Introduction

1.1. Goals

The use of radial basis functions (RBFs) offers several advantages in various applications [1] (Chapters 1–2). RBFs are commonly used for interpolation problems, where they approximate a function based on a set of input–output data pairs. Compared to other interpolation methods, RBFs provide more accurate approximations in high-dimensional spaces [2]. RBFs [3] (Chapter 3) have received interest from researchers in both engineering [4,5] and scientific domains [6,7]. They can also be employed for function approximation problems, where the target is to find a function that models the input–output relationship of a given system. In fact, they can consider both global and local features of the function being approximated, allowing for more accurate predictions. Furthermore, RBFs can model non-linear relationships between input and output variables, making them appropriate for applications where linear models are not sufficient. This flexibility arises from the choice of non-linear basis functions used in RBFs [8,9].

This paper investigates how to improve the computational efficiency and, especially, the convergence order of approximations without increasing the stencil size. In fact, in most already published works in the literature [10,11], to increase the convergence order in 1D based on equally spaced nodes, the convergence order cannot be bigger than two when the stencil size is three, which means the only remaining way to boost the convergence order is to increase the stencil size and consider four or more discretization points in a stencil. Hence, the objective is to achieve improved convergence rates and, thus, to propose compact formulas under the Hermite finite-difference (HFD) approach. Thus, we pursue

obtaining the RBF-HFD weights. The objective is to compute analytical coefficients and local truncation errors (LTEs) for these compact formulas, focusing on the 1st derivative and the 2nd derivative.

1.2. Background and Challenges

In the dimension d , a kernel as a RBF can be expressed as $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$, where $r = \|\mathbf{z} - \mathbf{z}_k\|_2$ is the radius and $\|\cdot\|_2$ stands for the two-norm. The RBF-FD extends classical FD methods to accommodate scattered node configurations [12]. The method employs RBFs to perform a local approximation of a linear differential operator within a selected neighborhood. This neighborhood, known as the stencil for a specific point, is often determined by the $n - 1$ closest neighbors. The RBF-FD has undergone extensive investigation and demonstrated successful performance across various contexts; see [13] for more. Like the FD methodology, the RBF-FD methodology provides an approximation to a differential linear operator $\mathcal{L}v$ at a specific point $\mathbf{x}_c \in \mathbb{R}^d$ by forming a linear summation of v_i at the n nearest knots as follows [14] (Page 70):

$$\mathcal{L}v \big|_{\mathbf{x}_c} \simeq \sum_{i=1}^n a_i v_i. \quad (1)$$

In the RBF-FD procedure, the function v can be either a scalar or a vector-valued function, and the weights a_i associated with the RBF-FD need to be calculated. In the initial stages of RBF-FD research, smooth RBFs (infinitely) such as the multiquadric (MQ) as

$$\varphi(r) = \sqrt{c^2 + r^2},$$

or the Gaussian as

$$\varphi(r) = \exp(-(r/c)^2),$$

had been commonly employed (c represents the shape parameter) [15,16].

To enhance the applicability and usefulness of the RBF-FD methodology, one approach is to increase the number of stencil knots. Increasing the number of stencil knots in RBF-FD methods can improve the stability, the convergence, and the method's accuracy, but it can also increase the computational cost and sensitivity to noise.

1.3. Motivation

Recently, Fornberg proposed an efficient algorithm in [17] to compute the compact weights for different node layouts. The authors in the work [18] introduced a unified framework for the systematic derivation of optimized compact formulas tailored for uniform grids. These schemes are computed analytically through the solution of an optimization problem, with the objective of minimizing the error while satisfying prescribed accuracy constraints. Furthermore, it demonstrates the versatility of this framework by showing its ability to generate various types of formulas, including spatially explicit and FD formulas, as special cases.

In this work, we study how to improve the convergence order without increasing the stencil size. So, the motivation is to obtain a higher order of convergence using only three-node stencils. Thus, we propose a new set of weights under a variant of the MQ RBF.

1.4. Structure

We provide the rest of the manuscript as follows. In Section 2, the RBF-HFD scheme is furnished. Section 3 is devoted to approximate the 1st and 2nd function derivatives in 1D. The theoretical error equations corresponding to the RBF-HFD formulations under a variant of the MQ RBF are provided therein as well. In Section 4, we substantiate the validity, convergence, and precision of the constructed formulas via several experiments based on LTEs. Section 5 gives one application when the obtained set of weights can be used in solving a fractional time-dependent partial differential equation (PDE). Finally,

Section 6 is devoted to drawing a short conclusion of the paper, offering a summary of the obtained results.

2. RBF-HFD Formulations

To improve the convergence results while avoiding the increase of the stencil size, Wright and Fornberg [19] discussed a compact method via Hermite RBF interpolation called the RBF-HFD scheme. The RBF-HFD scheme includes not only the nodal functional values, but also their derivatives. This allows the compact FD to achieve higher accuracy in solving PDEs, particularly in problems with steep gradients or singularities. See [20,21] for more.

We now describe the RBF-HFD as follows. The computation of the operator $\mathcal{L}[\cdot]$ at $z = z_j$ involves examining a stencil comprising $m_1 < m$ nodes in dimension one, as expressed in the following relation [22]:

$$\mathcal{L}[v(z_j)] \simeq \sum_{i=1}^{m_1} a_i v(z_i) + \sum_{p=1}^{m_2} v_p \mathcal{L}[v(z_p)]. \quad (2)$$

In this work, $m_1 = 3$ and $m_2 = m_1 - 1 = 2$. The determination of the weights in this approach, represented by a_i and v_p , is accomplished through the use of Hermite RBF interpolants, as outlined below:

$$v(z) = \sum_{i=1}^{m_1} \lambda_i \varphi(\|z - z_i\|) + \sum_{p=1}^{m_2} \bar{\lambda}_p \mathcal{L}[\varphi(\|z - z_p\|)]. \quad (3)$$

Moreover, the weighting coefficients can be acquired by solving the ensuing linear system of equations, which emerges upon substituting (3) into (2) and expressing

$$\mathcal{L}[\varphi(\|z_k - z_j\|)] = \sum_{i=1}^{m_1} a_i \varphi(\|z_k - z_i\|) + \sum_{p=1}^{m_2} v_p \mathcal{L}[\varphi(\|z_k - z_p\|)], \quad (4)$$

$$\mathcal{L}\mathcal{L}[\varphi(\|z_s - z_j\|)] = \sum_{i=1}^{m_1} a_i \mathcal{L}(\varphi(\|z_s - z_i\|)) + \sum_{p=1}^{m_2} v_p \mathcal{L}\mathcal{L}[\varphi(\|z_s - z_p\|)]. \quad (5)$$

In (3), the coefficients are functions of c and h . Here, we explore the coefficients of the RBF-HFD scheme via a variant of the MQ RBF furnished as:

$$\varphi(\|\mathbf{z} - \mathbf{z}_i\|_2) = \left(c^2 + \|\mathbf{z} - \mathbf{z}_i\|_2^2\right)^{3/2}, \quad i = 1, 2, \dots, m. \quad (6)$$

3. Finding the Coefficients and the Solution Scheme

The importance of the weighting coefficients in RBF-HFD approximations resides in their capacity to enhance the speed when estimating the 1st and 2nd derivatives with just three neighboring equidistant nodes. By incorporating the Hermite concept, this approach considers not only the functional values at each adjacent point, but also considers the differentiations.

Initially, we examine a collection of evenly spaced points $\{z_1, z_2, \dots, z_m\}$, subsequently applying a three-point stencil organized as follows:

$$\{z_i - h, z_i, z_i + h\} = \{z_{i-1}, z_i, z_{i+1}\}, \quad h > 0. \quad (7)$$

Now, let us examine the RBF-HFD approximation (shown by $\hat{v}$) as follows:

$$\begin{aligned} v'(z_i) \simeq \hat{v}'(z_i) &= a_{i-1}v(z_{i-1}) + a_i v(z_i) + a_{i+1}v(z_{i+1}) \\ &\quad + b_{i-1}v'(z_{i-1}) + b_{i+1}v'(z_{i+1}), \quad 2 \leq i \leq m-1. \end{aligned} \quad (8)$$

To start, we substitute v using (6) at the stencil points, resulting in the subsequent system of equations:

$$c^3 a_{i+1} + a_i (c^2 + h^2)^{3/2} + a_{i-1} (c^2 + 4h^2)^{3/2} + 3h\sqrt{h^2 + c^2} = 6hb_{i-1}\sqrt{c^2 + 4h^2}, \quad (9)$$

$$\sqrt{h^2 + c^2} (a_{i+1} (h^2 + c^2) + 3h(b_{i+1} - b_{i-1})) + c^3 a_i + a_{i-1} (h^2 + c^2)^{3/2} = 0, \quad (10)$$

$$3h\sqrt{h^2 + c^2} = \sqrt{c^2 + 4h^2} (a_{i+1} (c^2 + 4h^2) + 6hb_{i+1}) + c^3 a_{i-1} + a_i (c^2 + h^2)^{3/2}, \quad (11)$$

$$6ha_{i-1}\sqrt{c^2 + 4h^2} + \frac{3(ha_i(c^2 + h^2) + c^2 + 2h^2)}{\sqrt{c^2 + h^2}} = \frac{3b_{i-1}(c^2 + 8h^2)}{\sqrt{c^2 + 4h^2}} + 3cb_{i+1}, \quad (12)$$

$$\frac{3(c^2 + 2h^2)}{\sqrt{c^2 + h^2}} = 3 \left(ha_i\sqrt{h^2 + c^2} + 2ha_{i+1}\sqrt{c^2 + 4h^2} + \frac{b_{i+1}(c^2 + 8h^2)}{\sqrt{c^2 + 4h^2}} + cb_{i-1} \right). \quad (13)$$

By resolving (9)–(13) through incorporating certain simplifications, we arrive at:

$$a_i = 0, \quad (14)$$

$$a_{i-1} = -a_{i+1} = -\frac{3h}{16c^2} - \frac{3}{4h}, \quad (15)$$

$$b_{i-1} = b_{i+1} = -\frac{3h^2}{16c^2} - \frac{1}{4}. \quad (16)$$

Here, we note that the weights (14)–(16) have been derived through simplifications and Taylor expansions up to the second order of the solution of the linear system (9)–(13). While it is conceivable to consider higher order terms, they result in a more intricate version of the final analytical weights without affecting the ultimate error equation of the approximations, as will be demonstrated in Theorem 1. To elaborate further, expanding the Taylor series up to third- and fourth-degree terms would yield the following equivalent weighting coefficients (respectively):

$$a_i = 0, \quad (17)$$

$$a_{i-1} = -a_{i+1} = \frac{3 \left(\frac{2h^4}{c^4} - \frac{h^2}{c^2} - 4 \right)}{16h}, \quad (18)$$

$$b_{i-1} = b_{i+1} = -\frac{3h^2}{16c^2} - \frac{1}{4}, \quad (19)$$

and

$$a_i = 0, \quad (20)$$

$$a_{i-1} = -a_{i+1} = \frac{3 \left(\frac{2h^4}{c^4} - \frac{h^2}{c^2} - 4 \right)}{16h}, \quad (21)$$

$$b_{i-1} = b_{i+1} = \frac{1}{16} \left(\frac{8h^4}{c^4} - \frac{3h^2}{c^2} - 4 \right). \quad (22)$$

Theorem 1. Let $\varphi(r)$ be given by (6). Then, the compact formula (8) for the first derivative of a function $v(z)$ (sufficiently smooth) on (7) converges quartically.

Proof. As explained in Section 2, we obtain the analytical weights for the first derivative using symbolic calculations. The set of weights (14)–(16) would yield in the following LTE:

$$e_1(z_i) = \left(-\frac{v^{(3)}(z_i)}{8c^2} - \frac{1}{120}v^{(5)}(z_i) \right) h^4 + \mathcal{O}(h^5), \quad (23)$$

where

$$e_1(z_i) = \hat{v}'(z_i) - v'(z_i). \quad (24)$$

The equation of error (23) shows a quartical convergence rate for the presented approximation on uniformly spaced grids. The details of the proof can be extracted from the *Mathematica* program given in Appendix A. This concludes the proof. $\square$

In tackling the situation related to the second derivative as outlined in Section 2 within the RBF-HFD approach, we utilize (7) and express:

$$\begin{aligned} v''(z_i) \simeq \hat{v}''(z_i) &= p_{i-1}v(z_{i-1}) + p_iv(z_i) + p_{i+1}v(z_{i+1}) \\ &+ q_{i-1}v''(z_{i-1}) + q_{i+1}v''(z_{i+1}), \quad 2 \leq i \leq m-1. \end{aligned} \quad (25)$$

We used different symbols p_i and q_i for the coefficients to differ in contrast to the first derivative approximation. Likewise, by formulating the subsequent set of five interrelated equations at the nodes of the stencil:

$$\frac{3(c^2 + 2h^2)}{\sqrt{h^2 + c^2}} = c^3 p_{i+1} + (h^2 + c^2)^{3/2} p_i + (c^2 + 4h^2)^{3/2} p_{i-1} + \frac{3(c^2 + 8h^2)q_{i-1}}{\sqrt{c^2 + 4h^2}} + 3cq_{i+1}, \quad (26)$$

$$3c = \frac{(h^2 + c^2)^2 p_{i-1} + (h^2 + c^2)^2 p_{i+1} + 3(c^2 + 2h^2)q_{i-1} + 3(c^2 + 2h^2)q_{i+1} + c^3 \sqrt{h^2 + c^2} p_i}{\sqrt{h^2 + c^2}}, \quad (27)$$

$$\frac{3(c^2 + 2h^2)}{\sqrt{h^2 + c^2}} = c^3 p_{i-1} + (h^2 + c^2)^{3/2} p_i + (c^2 + 4h^2)^{3/2} p_{i+1} + \frac{3(c^2 + 8h^2)q_{i+1}}{\sqrt{c^2 + 4h^2}} + 3cq_{i-1}, \quad (28)$$

$$6h\sqrt{c^2 + 4h^2}p_{i-1} + 3h\sqrt{h^2 + c^2}p_i + \frac{6(3c^2h + 8h^3)q_{i-1}}{(c^2 + 4h^2)^{3/2}} = \frac{9c^2h + 6h^3}{(h^2 + c^2)^{3/2}}, \quad (29)$$

$$\frac{9c^2h + 6h^3}{(h^2 + c^2)^{3/2}} = \frac{6h((c^2 + 4h^2)^2 p_{i+1} + (3c^2 + 8h^2)q_{i+1})}{(c^2 + 4h^2)^{3/2}} + 3h\sqrt{h^2 + c^2}p_i. \quad (30)$$

Solving the system (26)–(30) results in the following solution:

$$p_{i-1} = p_{i+1} = \frac{294h^2}{125c^4} + \frac{9}{50c^2} + \frac{6}{5h^2}, \quad (31)$$

$$p_i = \frac{3}{125} \left(-\frac{196h^2}{c^4} - \frac{15}{c^2} - \frac{100}{h^2} \right), \quad (32)$$

$$q_{i-1} = q_{i+1} = -\frac{9h^2}{100c^2} - \frac{1}{10}. \quad (33)$$

Here, it is mentioned that the weights (31)–(33) have been figured out by making the formulas simpler and using Taylor expansions up to the second order of the solution of the linear system (26)–(30). Although we could think about using higher order terms, doing so would make the final analytical weights more complex without making a difference for the error equation of the approximation, as we will show in Theorem 2. For example, if we consider the Taylor expansion up to fourth-order terms, we obtain:

$$p_{i-1} = p_{i+1} = \frac{3 \left(-\frac{64599h^4}{c^6} + \frac{7840h^2}{c^4} + \frac{600}{c^2} + \frac{4000}{h^2} \right)}{10,000}, \quad (34)$$

$$p_i = -\frac{3(500c^6 + 75c^4h^2 + 980c^2h^4 - 6903h^6)}{625c^6h^2}, \quad (35)$$

$$q_{i-1} = q_{i+1} = \frac{-\frac{327h^4}{c^4} - \frac{180h^2}{c^2} - 200}{2000}. \quad (36)$$

Theorem 2. Let $\varphi(r)$ is given by (6). Then, the compact formula (25) for approximating the second derivative of a function $v(z)$ (sufficiently smooth) on (7) converges quartically.

Proof. By incorporating the Taylor series having the second truncation order for the coefficients and putting them back in (25), it is finally obtained:

$$e_2(z_i) = \frac{1}{1000} \left(\frac{2352v''(z_i)}{c^4} - \frac{75v^{(4)}(z_i)}{c^2} - 5v^{(6)}(z_i) \right) h^4 + \mathcal{O}(h^5). \quad (37)$$

Here,

$$e_2(z_i) = \hat{v}''(z_i) - v''(z_i). \quad (38)$$

The relation (37) reveals a quartic convergence speed for the second derivative of an sufficiently differentiable function, using a three-node stencil. This concludes the proof. $\square$

4. The Advantage of the Analytical Coefficients

To affirm the precision of our formulas and examine how the LTE relies on h and c , we employed the subsequent test functions as illustrative instances:

$$v_1(z) = \cot(\sinh(z)), \quad \text{at } z = 1, \quad (39)$$

$$v_2(z) = \tan(z), \quad \text{at } z = 1, \quad (40)$$

$$v_3(z) = \exp(-z^3), \quad \text{at } z = 1, \quad (41)$$

$$v_4(z) = \exp(-z^4) + z^6, \quad \text{at } z = 1. \quad (42)$$

To attain this objective using $h = 1/100$, we employ (7) in conjunction with Formulas (14)–(16) to approximate the first derivatives. The subsequent findings, depicted in Figure 1, reveal the enhanced efficacy of theoretical coefficients when dealing with (9)–(13). This superiority becomes apparent in the absence of undesired oscillations and the considerable reduction of absolute errors (AEs). In contrast, computational weighting coefficients show divergence attributed to round-off errors, while theoretical weighting coefficients demonstrate stability.

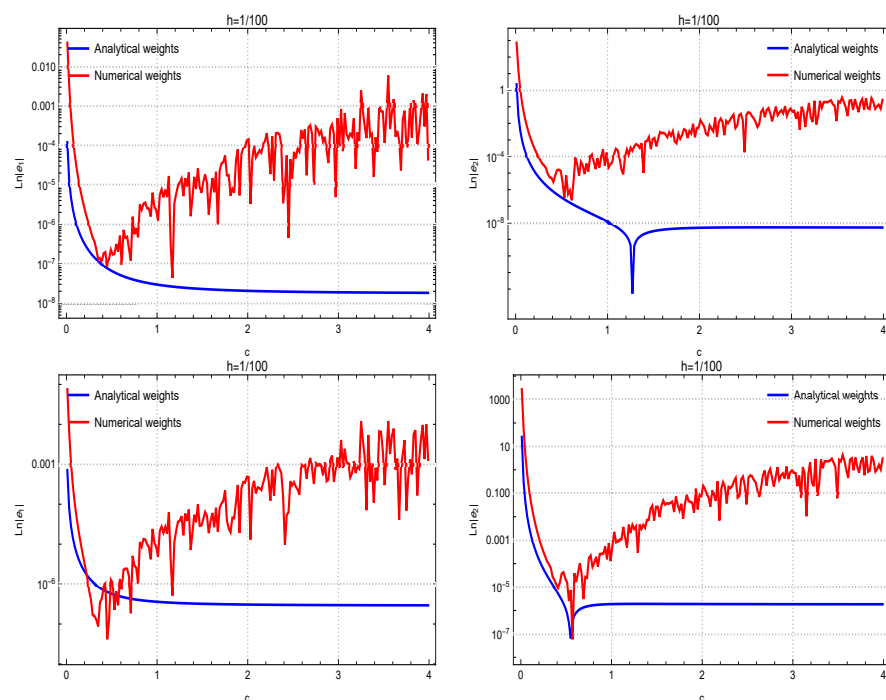


Figure 1. Cont.

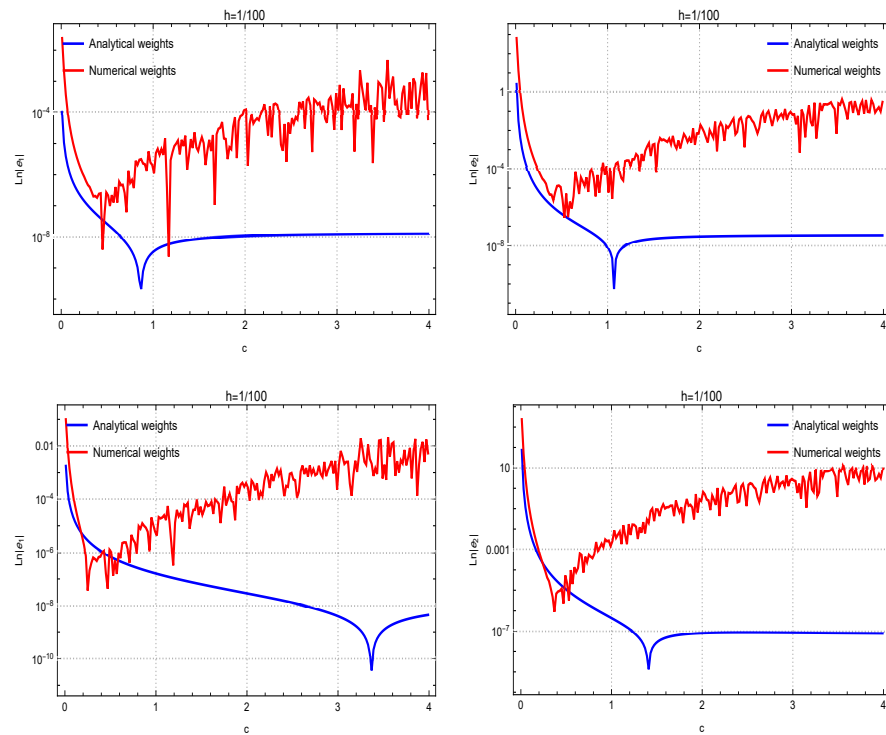


Figure 1. Evaluating the accuracy of theoretical coefficients against numerical weights is conducted across test functions $v_1, \dots, v_4$, organized in rows 1 through 4. The assessment is performed with the fixed $h = 1/100$. The outcomes for approximating the first and second derivatives are given in the left and right figures, respectively.

Another exploration was undertaken to show the capabilities of (31)–(33) for estimating the second derivatives, comparing their performance to their numerical counterparts. The results presented in Figure 2, when $h = 1/1000$, offer additional affirmation of the advantages of theoretical coefficients in this context. Discrepancies in the performance of the method for first and second derivatives at different grid spacings ($h = 1/100$ and $h = 1/1000$) can be observed in the results. In fact, the lower the grid spacing is, more accurate results in terms of absolute errors can be observed.

The gap observed between analytical and numerical weights in Figure 2 requires further explanation. The insights into the factors contributing to this difference is necessary to ensure a thorough understanding of the results. The main reason lies in solving the linear systems in machine precision, which introduces round-off and cancellation errors especially when h tends to zero or c tends to be very large. In such cases, the linear system is ill-conditioned. However, solving the system in exact arithmetic and using the analytical weights can diminish such errors in computing.

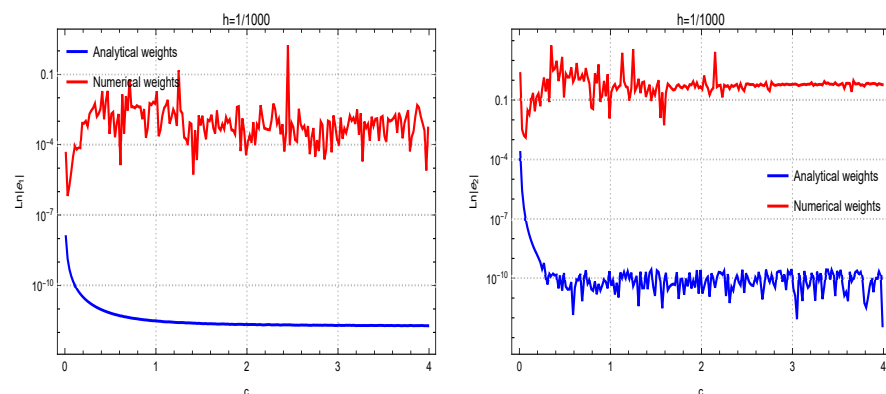


Figure 2. Cont.

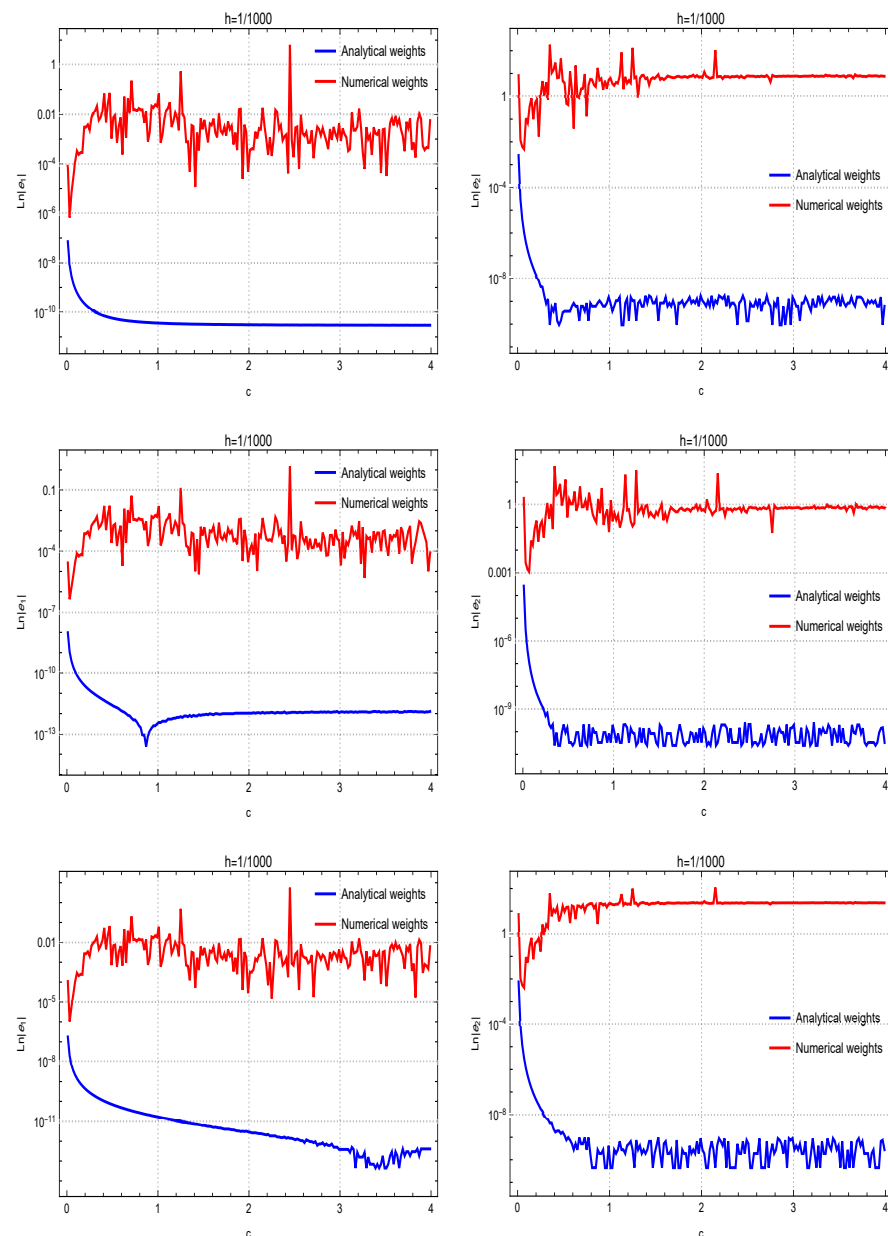


Figure 2. Evaluations of the accuracy of theoretical coefficients against numerical weights are conducted across test functions $v_1, \dots, v_4$ organized in rows 1 through 4. The assessment is performed with the fixed $h = 1/1000$. The results for approximating the first and second derivatives are given in the left and right figures, respectively.

Checking the Rate of Convergence

The numerical order of convergence (NOC) is investigated using the absolute error e_h (under the discretization step size h) and can be defined as follows:

$$\text{NOC}_h \approx \log_2 \left| \frac{e_{h/2}}{e_h} \right|. \quad (43)$$

The test for checking the NOC is as follows:

$$v_5(z) = \cos(100z) - \tan(2\pi z), \quad z \in [0, 1], \quad (44)$$

The convergence behavior and NOCs for four different approximations are presented in Figure 3. We approximate the first derivative of the function (44) at $z = 1/2$. The

compared methods on equi-spaced grids are the three-point FD, RBF-FD, and five-point RBF-FD weights given in [23]. The proposed weights (14)–(16) are shown by RBF-HFD. The slope corresponding to each set of weights obtained is reported next to the caption of each method in Figure 3. Noting that the comparison of the weights with a small shape parameter against those with a large shape parameter lacks justification, due to this, we have chosen different shape parameters, but the same for all methods when comparing. The results confirm the theoretical orders of Section 3.

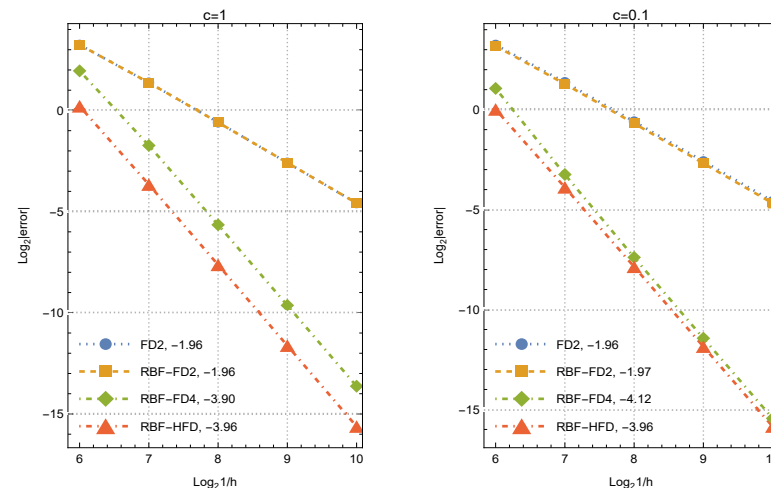


Figure 3. NOCs for various sets of weights in approximating the first derivative.

5. A PDE Problem

The time-dependent fractional Black–Scholes (FBS) PDE presents numerous advantages in options pricing within financial markets when compared to the conventional BS equation [24]. This formulation, applicable for $z \in [0, \infty)$, is expressed as follows [25]:

$$\frac{\partial^\alpha v(z, t)}{\partial t^\alpha} = \frac{1}{2} \sigma^2 z^2 \frac{\partial^2 v(z, t)}{\partial z^2} + (r - \omega) z \frac{\partial v(z, t)}{\partial z} - r v(z, t), \quad (45)$$

where $\alpha \in (0, 1)$, t represents the time to maturity, and r denotes the interest rate, while additional parameters include ω for dividend yield and σ for the volatility constant. Furthermore, expressing α as $2H$ indicates the actual Hurst exponent as $0 < H < 1/2$, as established in [26]. It is observed that Equation (45) simplifies to the classical Black–Scholes partial differential equation (BS PDE) when $H = 1/2$. If we designate E as the price of the strike, the initial condition for (45) can alternatively be characterized for the call and put variations [27,28]:

$$v(z, 0) = \max\{z - E, 0\}, \quad (46)$$

and

$$v(z, 0) = \max\{E - z, 0\}. \quad (47)$$

The boundary conditions for call/put options are expressed, respectively, by

$$\lim_{z \rightarrow \infty} v(z, t) \simeq z_{\max} \exp(-\omega t) - E \exp(-rt), \quad v(0, t) = 0, \quad (48)$$

$$\lim_{z \rightarrow \infty} v(z, t) = 0, \quad v(0, t) \simeq E \exp\{-rt\}. \quad (49)$$

Here, the errors are documented and subjected to comparison. The methods we used for the comparisons are as follows:

- The method based on the introduced RBF-HFD weights in Section 3 and shown by PM.
- The 2nd-order FD solver on equally spaced meshes and 1st-order Euler's method along time [29] shown by FD2.

- The RBF-FD approach on uniformly spaced grids is employed to tackle (45), as discussed in [30]. In this section, we denote it as SSM.

Mathematica 13.3 [31,32] is used to perform the computations in standard floating point arithmetic. Here, $z_{\max} = 3E$ is considered. Note that the proposed sets of weights are employed on uniform Cartesian meshes. This is done intentionally to gain as much as possible of the convergence order in practical problems. So, we do not employ non-Cartesian nodes for solving (45), which are unnecessary for this problem.

The shape parameter for any of the kernels is chosen adaptively as follows:

$$c = 4h. \quad (50)$$

The error is given by $|v_{\text{approx}} - v_{\text{ref}}|$, while v_{ref} and v_{approx} denote the reference and computational solutions, respectively. The computations were conducted on a system with Windows 11 and an Intel Core i7-9750H processor. The time required for these computations, measured in seconds, is presented in the tables under the label *Time*.

We analyzed a call option possessing the spot price of USD 100, by considering that today is 13 February 2024 and the option expires on 13 February 2025. The annualized rate of interest is 5%, with an annualized volatility of 40%, and ϖ equals 0. Additionally, $\alpha = 0.8$, and $v(E, T) \simeq 18.1883$ is established, as mentioned in [30]. This value serves as a benchmark for comparisons. The following section provides a detailed set of computational comparisons involving various solvers. Table 1 illustrates a comparison of various solvers for solving (45).

Table 1. Results among various solvers in Section 5.

m, n	v_{FD2}	Error FD2	Time FD2	v_{SSM}	Error SSM	Time SSM	v_{PM}	Error PM	Time PM
10	16.070	2.1×10^0	0.01	17.378	8.1×10^{-1}	0.01	17.612	5.7×10^{-1}	0.01
20	17.889	2.9×10^{-1}	0.02	17.694	4.9×10^{-1}	0.02	17.991	1.9×10^{-1}	0.03
40	17.913	2.7×10^{-1}	0.16	18.003	1.8×10^{-1}	0.18	18.152	3.6×10^{-2}	0.30
80	18.039	1.4×10^{-1}	3.96	18.213	2.4×10^{-2}	3.84	18.181	7.3×10^{-3}	4.94
120	18.055	1.3×10^{-1}	20.02	18.190	1.6×10^{-3}	20.96	18.188	3.5×10^{-4}	23.41

Three solvers were evaluated: the FD2, SSM, and PM schemes. The proposed scheme performed much better than the other methods in terms of both accuracy and efficiency. To evaluate its stability and positivity, we investigated the numerical solution with varying numbers of discretized points across the interval, as illustrated in Figure 4. The figure presents computational solutions using three sets of uniform points. The left subplots show-case the discrete computational resolutions, while the right ones display the continuous versions extracted through interpolation, produced by the programming package.

However, here, we note that the proposed methodology has a drawback of being employed in arbitrary domains for scattered node layouts. In fact, since it requires the Hermite condition, i.e., the derivative information for each point of the stencil except the center, then such information might be unavailable to time-consuming in arbitrary domain problems, which might restrict its application. So, it mainly is suitable for interpolations, as well as for PDE problems on regular domains.

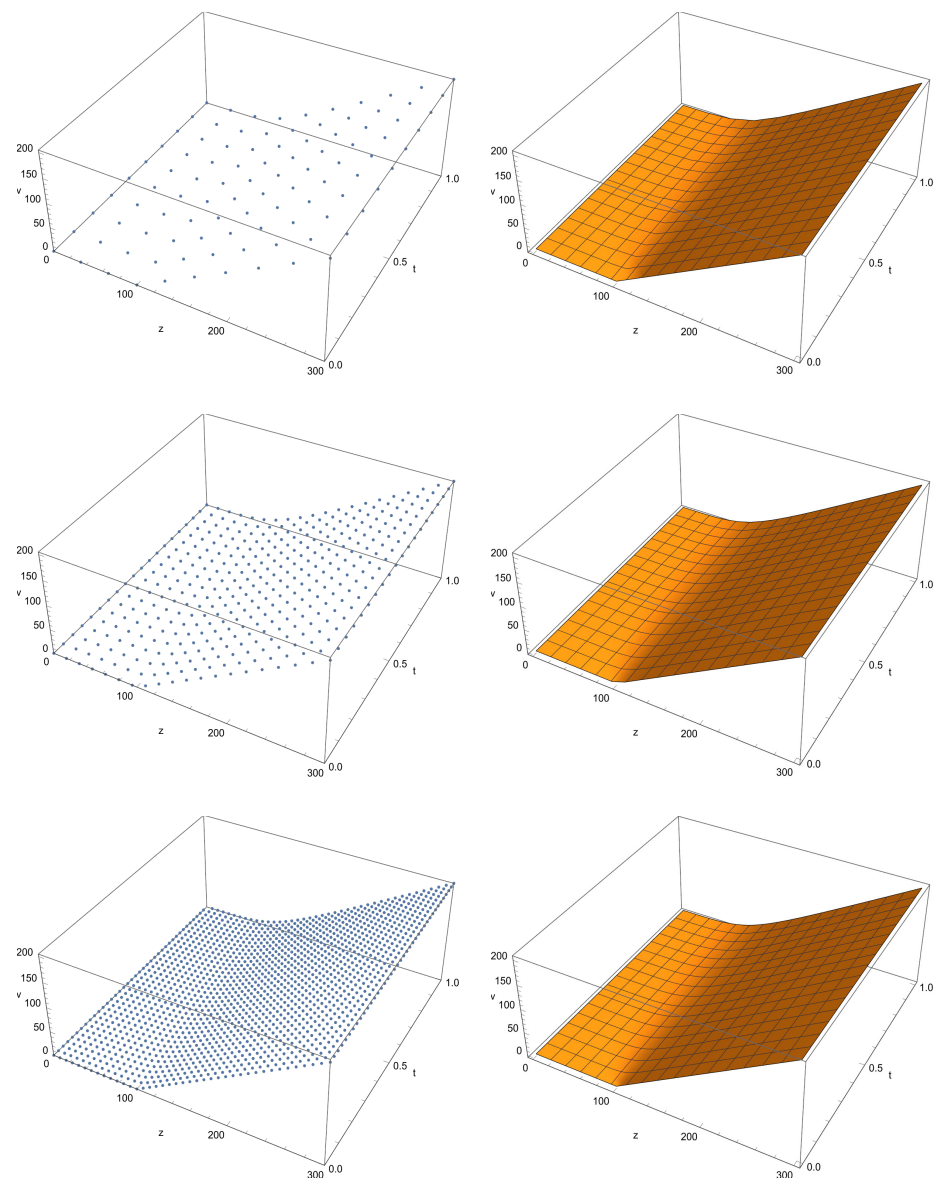


Figure 4. Numerical findings upon solving FBS PDE by PM. Upper left: A plot illustrating the situation having $n = m = 10$. Upper right: A plot of the solution having $m = n = 10$. Middle left: A point plot via $m = n = 20$. Middle right: A continuous plot showing the setup having $n = m = 20$. Lower left: A point plot showing the scenario having $n = m = 40$. Lower right: A continuous plot showing the configuration having $n = m = 40$.

6. Conclusions

This study proposes coefficients for the RBF-HFD methodology using a variant of the multiquadric (MQ) RBF to approximate the 1st and 2nd derivatives of a function. The analytical computation of weights and LTEs highlights the efficiency of this methodology. Computational results confirm the effectiveness of this approach, indicating its potential for various applications in numerical analysis and solving PDE problems. Future research will focus on enhancing the convergence order on a stencil with five nodes using the RBF-HFD approach, comparing it to the convergence rates of FD or RBF-FD approximations on such a stencil. Additionally, exploration will be conducted on applying this strategy to the state-of-the-art polyharmonic spline (PHS) RBF, which is parameter-free.

Author Contributions: Supervision, T.L. and S.S.; methodology, T.L. and S.S.; investigation, T.L. and S.S.; funding acquisition, T.L. and S.S.; formal analysis, T.L. and S.S.; conceptualization, T.L. and S.S.; validation, T.L. and S.S.; writing—original draft, T.L. and S.S.; writing—review and editing, T.L. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: The research of the first author was funded by the Research Project on Graduate Education and Teaching Reform of Hebei Province of China (YJG2024133).

Data Availability Statement: No new data were created or analyzed in this study.

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

Appendix A

The *Mathematica* 13.0 program used in the process of the proof in Theorem 1.

```
ClearAll["Global`*"];
ph[r1_] := (c^2 + r1^2)^(3/2)
F1 = D[ph[r1], r1] // FullSimplify;
F2 = D[ph[r1], {r1, 2}] // FullSimplify;
v1[z_] := a1*v[z - h] + a2*v[z] + a3*v[z + h] + b1*v'[z - h] + b2*v'[z + h]

(*First derivative*)
eq1 = ((F1 /. {r1 -> -h}) == {(ph[r1] /. {r1 -> -2 h}), (ph[
    r1] /. {r1 -> -h}), (ph[r1] /. {r1 -> 0})
    , (F1 /. {r1 -> -2 h}), (F1 /. {r1 -> 0})} . {a1, a2, a3, b1,
    b2}) // Simplify;
eq2 = ((F1 /. {r1 ->
    0}) == {(ph[r1] /. {r1 -> -h}), (ph[r1] /. {r1 -> 0}), (ph[
    r1] /. {r1 -> h})
    , (F1 /. {r1 -> -h}), (F1 /. {r1 -> h})} . {a1, a2, a3, b1,
    b2}) // Simplify;
eq3 = ((F1 /. {r1 ->
    h}) == {(ph[r1] /. {r1 -> 0}), (ph[r1] /. {r1 -> h}), (ph[
    r1] /. {r1 -> 2 h})
    , (F1 /. {r1 -> 0}), (F1 /. {r1 -> 2 h})} . {a1, a2, a3, b1,
    b2}) // Simplify;
eq4 = ((F2 /. {r1 -> -h}) == {(F1 /. {r1 -> -2 h}), (F1 /. {r1 -> \
-h}), (F1 /. {r1 -> 0})
    , (F2 /. {r1 -> -2 h}), (F2 /. {r1 -> 0})} . {a1, a2, a3, b1,
    b2}) // Simplify;
eq5 = ((F2 /. {r1 ->
    h}) == {(F1 /. {r1 -> 0}), (F1 /. {r1 -> h}), (F1 /. {r1 ->
    2 h})
    , (F2 /. {r1 -> 0}), (F2 /. {r1 -> 2 h})} . {a1, a2, a3, b1,
    b2}) // Simplify;
{b, A} = Simplify@
    CoefficientArrays[{eq1, eq2, eq3, eq4, eq5}, {a1, a2, a3, b1, b2}];
sol = Simplify@Inverse[A] . (-b));

taylororder = 2;
sol2 = (Normal@Series[Simplify@sol /. Thread[h -> ep*c]), {ep, 0,
    taylororder}] /. ep -> h/c // FullSimplify
(Series[v1[z] /. {a1 -> sol2[[1]], a2 -> sol2[[2]], a3 -> sol2[[3]],
    b1 -> sol2[[4]], b2 -> sol2[[5]]}, {h, 0, 4}] // FullSimplify)
```

References

1. Fasshauer, G.E. *Meshfree Approximation Methods with Matlab*; World Scientific, 5 Toh Tuck Link: Singapore, 2007.
2. Fornberg, B.; Flyer, N. Solving PDEs with radial basis functions. *Acta Numer.* **2015**, *24*, 215–258. [\[CrossRef\]](#)
3. Iske, A. *Multiresolution Methods in Scattered Data Modelling, Lecture Notes in Computational Science and Engineering*; Springer: Heidelberg, Germany, 2004; Volume 37.
4. Babayar-Razlighi, B. Numerical solution of an influenza model with vaccination and antiviral treatment by the Newton-Chebyshev polynomial method. *J. Math. Model.* **2023**, *11*, 103–116.
5. Ebrahimijahan, A.; Dehghan, M.; Abbaszadeh, M. Simulation of the incompressible Navier-Stokes via integrated radial basis function based on finite difference scheme. *Eng. Comput.* **2022**, *38*, 5069–5090. [\[CrossRef\]](#)
6. Itkin, A.; Soleymani, F. Four-factor model of quanto CDS with jumps-at-default and stochastic recovery. *J. Comput. Sci.* **2021**, *54*, 101434. [\[CrossRef\]](#)
7. Keshavarzi, C.G.; Ghoreishi, F. Numerical solution of the Allen-Cahn equation by using shifted surface spline radial basis functions, Iran. *Numer. Anal. Optim.* **2020**, *10*, 177–196.
8. Cavoretto, R.; Rossi, A.D. An adaptive residual sub-sampling algorithm for kernel interpolation based on maximum likelihood estimations. *J. Comput. Appl. Math.* **2023**, *418*, 114658. [\[CrossRef\]](#)
9. Cavoretto, R.; Rossi, A.D.; Dell’Accio, F.; Tommaso, F.D.; Siar, N.; Sommariva, A.; Vianello, M. Numerical cubature on scattered data by adaptive interpolation. *J. Comput. Appl. Math.* **2024**, 115793. [\[CrossRef\]](#)
10. Soleymani, F.; Zhu, S. On a high-order Gaussian radial basis function generated Hermite finite difference method and its application. *Calcolo* **2021**, *58*, 50. [\[CrossRef\]](#)
11. Yang, Y.; Soleymani, F.; Barfeie, M.; Tohidi, E. A radial basis function-Hermite finite difference approach to tackle cash-or-nothing and asset-or-nothing options. *J. Comput. Appl. Math.* **2020**, *368*, 112523. [\[CrossRef\]](#)
12. Bayona, V. An insight into RBF-FD approximations augmented with polynomials. *Comput. Math. Appl.* **2019**, *77*, 2337–2353. [\[CrossRef\]](#)
13. Sanyasiraju, Y.V.S.S.; Chandhini, G. Local radial basis function based gridfree scheme for unsteady incompressible viscous flows. *J. Comput. Phys.* **2008**, *227*, 8922–8948. [\[CrossRef\]](#)
14. Tolstykh, A.I. On using radial basis functions in a “finite difference mode” with applications to elasticity problems. *Comput. Mech.* **2003**, *33*, 68–79. [\[CrossRef\]](#)
15. Cheng, A.H.-D. Multiquadric and its shape parameter—A numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation. *Eng. Anal. Bound. Elem.* **2012**, *36*, 220–239. [\[CrossRef\]](#)
16. Company, R.; Egorova, V.N.; Jódar, L.; Soleymani, F. A stable local radial basis function method for option pricing problem under the Bates model. *Numer. Methods Partial. Differ. Equ.* **2019**, *35*, 1035–1055. [\[CrossRef\]](#)
17. Fornberg, B. An algorithm for calculating Hermite-based finite difference weights. *IMA J. Numer. Anal.* **2021**, *41*, 801–813. [\[CrossRef\]](#)
18. Deshpande, V.M.; Bhattacharya, R.; Donzis, D.A. A unified framework to generate optimized compact finite difference schemes. *J. Comput. Phys.* **2021**, *432*, 110157. [\[CrossRef\]](#)
19. Wright, G.B.; Fornberg, B. Scattered node compact finite difference-type formulas generated from radial basis functions. *J. Comput. Phys.* **2006**, *212*, 99–123. [\[CrossRef\]](#)
20. Fornberg, B.; Flyer, N. *A Primer on Radial Basis Functions with Applications to the Geosciences*; SIAM: Philadelphia, PA, USA, 2015.
21. Lele, S.K. Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.* **1992**, *103*, 16–42. [\[CrossRef\]](#)
22. Collatz, L. *The Numerical Treatment of Differential Equations*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 1966.
23. Soleymani, F.; Barfeie, M.; Khaksar Haghani, F. Inverse multi-quadric RBF for computing the weights of FD method: Application to American options. *Commun. Nonlinear Sci. Numer. Simulat.* **2018**, *64*, 74–88. [\[CrossRef\]](#)
24. Jumarie, G. Derivation and solutions of some fractional Black-Scholes equations in coarse-grained space and time: Application to Merton’s optimal portfolio. *Comput. Math. Appl.* **2010**, *59*, 1142–1164. [\[CrossRef\]](#)
25. Wyss, W. The fractional Black-Scholes equation. *Fract. Calc. Appl. Anal.* **2000**, *3*, 51–62.
26. Hurst, H.E. Long-term storage capacity of reservoirs. *Trans. Amer. Soc. Civil Eng.* **1951**, *116*, 770–799. [\[CrossRef\]](#)
27. Seydel, R.U. *Tools for Computational Finance*, 6th ed.; Springer: London, UK, 2017.
28. Dai, X.; Ullah, Z.M. An efficient higher-order numerical scheme for solving fractional Black-Scholes PDE using analytical weights. *Iran. J. Sci.* **2024**, *48*, 423–435. [\[CrossRef\]](#)
29. Zhang, H.; Liu, F.; Turner, I.; Yang, Q. Numerical solution of the time fractional Black-Scholes model governing European options. *Comput. Math. Appl.* **2016**, *71*, 1772–1783. [\[CrossRef\]](#)
30. Song, Y.; Shateyi, S. Inverse multiquadric function to price financial options under the fractional Black-Scholes model. *Fractal Fract.* **2022**, *6*, 599. [\[CrossRef\]](#)
31. Ruskeepää, H. *Mathematica Navigator*, 3rd ed.; Academic Press: Burlington, VT, USA, 2009.
32. Georgakopoulos, N.L. *Illustrating Finance Policy with Mathematica*; Springer International Publishing: Cham, Switzerland, 2018.

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.