mathematics

Article

# Investigation of Higher Order Localized Approximations for a Fractional Pricing Model in Finance 

Malik Zaka Ullah ${ }^{1}{ }^{(\mathbb{D}}$, Abdullah Khamis Alzahrani ${ }^{1}$, Hashim Mohammed Alshehri ${ }^{1}$ and Stanford Shateyi ${ }^{2, *}$ (©)<br>1 Mathematical Modelling and Applied Computation (MMAC) Research Group, Department of Mathematics, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia; zmalek@kau.edu.sa (M.Z.U.); akalzahrani@kau.edu.sa (A.K.A.); hmalshehri@kau.edu.sa (H.M.A.)<br>2 Department of Mathematics and Applied Mathematics, School of Mathematical and Natural Sciences, University of Venda, P. Bag X5050, Thohoyandou 0950, South Africa<br>* Correspondence: stanford.shateyi@univen.ac.za

Citation: Ullah, M.Z.; Alzahrani, A.K.; Alshehri, H.M.; Shateyi, S. Investigation of Higher Order Localized Approximations for a Fractional Pricing Model in Finance. Mathematics 2023,11, 2641. https:// doi.org/10.3390/math11122641

Academic Editors: Tatiana Filatova and Peter Brusov

Received: 26 April 2023
Revised: 29 May 2023
Accepted: 8 June 2023
Published: 9 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/).


#### Abstract

In this work, by considering spatial uniform meshes and stencils having five adjacent discretization nodes, we furnish a numerical scheme to solve the time-fractional Black-Scholes (partial differential equation) PDE to price financial options under the generalized multiquadric radial basis function (RBF). The time-fractional derivative is estimated by an L1-scheme but the spatial variable is discretized using fourth-order RBF-FD methodology. As a matter of fact, the PDE problem is transformed in the form of a linear set of algebraic equations. To support analytical discussions, numerical tests are furnished and reveal the efficacy of the presented solver.


Keywords: option pricing; fractional Black-Scholes; radial basis function (RBF); RBF-FD; numerical method

MSC: 65M22; 35R11

## 1. Preliminaries and Literature

The topic of computational methods in resolving partial differential equations (PDEs) of the fractional type is an active area of research with a wide range of applications in various fields. Let the real function $a$ be continuous and $v$ be a constant, then the modified Riemann-Liouville (RL) fractional differentiation has the following definition [1]:

$$
a^{(\alpha)}(t):=\frac{d^{\alpha} a(t)}{d t^{\alpha}}= \begin{cases}(\Gamma(1-\alpha))^{-1} \frac{d}{d t} \int_{0}^{t}(t-\omega)^{-\alpha}(-a(0)+a(\omega)) d \omega, & 0<\alpha<1  \tag{1}\\ (\Gamma(-\alpha))^{-1} \int_{0}^{t}(t-\omega)^{-\alpha-1}(-a(0)+a(\omega)) d \omega, & \alpha<0 \\ \frac{d^{v}\left(a^{(\alpha-v)}(t)\right)}{d t^{v}}, & v \leq \alpha<v+1\end{cases}
$$

where $\Gamma(\cdot)$ is the Gamma's function. In addition, the Caputo fractional derivative can be expressed as [2]:

$$
\begin{equation*}
{ }_{0}^{C} D_{t}^{\alpha} a(t)=(1 / \Gamma(-\alpha+1))\left(\int_{0}^{t}(t-\omega)^{-\alpha} a^{\prime}(\omega) d \omega\right), \quad 0<\alpha<1 . \tag{2}
\end{equation*}
$$

By considering the (right) modified Riemann-Liouville (RL) time derivative, the BlackScholes (BS) option price in the fractional sense could be given as comes next (forward in time) [3,4]:

$$
\begin{equation*}
\frac{\partial^{\alpha} b(y, t)}{\partial t^{\alpha}}=\frac{1}{2} \sigma^{2} y^{2} \frac{\partial^{2} b(y, t)}{\partial y^{2}}+(r-q) y \frac{\partial b(y, t)}{\partial y}-r b(y, t), \quad y \geq 0 \tag{3}
\end{equation*}
$$

wherein $T>0$ is the maturity time, the rate of interest is $r$ and $0<\alpha \leq 1$. Moreover, $q, \sigma$ are scalars. Here, $y$ is unbounded from one side. Additionally, $\alpha=2 H$ is the real Hurst exponent and $1 / 2>H>0$, [5]. It is clear that (3) would yield in the classical BS PDE by considering $H=0.5$.

The initial condition for (3) can be, respectively, given by the put and call vanilla options as comes next [6] (Chapter 1):

$$
\begin{equation*}
b(y, 0)=(K-y)^{+}, \tag{4}
\end{equation*}
$$

and

$$
\begin{equation*}
b(y, 0)=(y-K)^{+}, \tag{5}
\end{equation*}
$$

wherein $K$ is the price of strike. Here, the symbol of + stands for the function $(\cdot)^{+}=$ $\max \{0, \cdot\}$.

Note that the stochastic integrals under the Itô theory cannot be used since the fractional Brownian motion (FBM) is a semi-martingale. In fact, as the underlying market is driven by FBM, it must be clarified how to ensure the unique price of the market, as for this market, in the absence of transaction costs (taxes, commissions, etc.), there could still be an arbitrage strategy. In this paper, we follow the approach presented in [7-9], where the authors introduce the concept of a centered Gaussian martingale, referred to as the fundamental martingale. They demonstrate that this martingale produces the same filtration as the FBM. Given that the filtration captures the market information rather than the stochastic process alone, it is sensible to utilize this martingale for option pricing purposes.

On the other hand, one of the fundamental computational schemes for resolving the fractional BS PDE is the finite difference (FD) scheme [10] (Part IV) [11,12], which is mostly via equidistant discretization points and transforming the (3) into a system of discrete ones. To discuss this further, a spectral method for solving time-fractional PDEs in mathematical finance has been introduced in [13]. Recently, the FD method has been proposed in the compact form for solving the temporal-fractional BS PDE in [14].

Another type of numerical method that has attracted the attention of researchers is the meshfree method, see, for example, [15] and the references therein. Radial basis function (RBF) methods are a category of meshfree schemes that have been divided into two main categories of local and global ones. The local RBF meshfree method has the advantage of the sparse matrices when doing the discretizations. Another well-resulted approach that inherits both from the RBF and FD methods is the localized RBF-FD scheme [16,17]. The appearance and progressively comprehensive adoption of meshfree RBF-FD methods submit supplementary opportunities to reduce the computational cost when dealing with arbitrary domains in high dimensions.

In some works from the literature, such as [18], a logarithmic transformation is first performed on the fractional BS PDE to transform it into a PDE with constant coefficients. Although this may be useful at first sight, its problematic issue is that it makes the transformed PDE be defined on unbounded domains from both sides. This makes the application of numerical methods and the incorporation of boundary conditions more challenging. Thus, we do not perform such an act here in this work.

The major aim of this study is to pursue the recent literature [19,20] and present an effective numerical method for resolving (3) with the help of a well-known and general RBF, i.e., the generalized multiquadric (GMQ) RBF. We are here motivated to propose a new method-based RBF-FD methodology, which is based on a general family of RBF and can contain different types of RBFs as well as has fourth order of convergence on stencils having five uniform discretization points.

The remaining portions of this work unfold as follows:

- Section 2 is assigned to three parts, in which the first part discusses the fractional temporal discretization, while the second and the third parts are for the RBF-FD methodology and obtaining the weighting coefficients for a stencil having five nodes.
- Section 2 also furnishes a quartically convergent discretization for the space variable (i.e., the underlying asset) by using five adjacent uniform points.
- Section 3 studies how the presented method could be derived in matrix notations for the convenience of the users.
- Section 4 is furnished to put on show the comparisons of our solver in contrast to the state-of-the-art solvers of the same type.
- Section 5 gives brief conclusions of this manuscript with several outlines for forthcoming works.


## 2. Discretizations

### 2.1. Temporal Discretization

Assuming that the mesh of time discretization nodes $0=t_{0}<t_{1}<\cdots<t_{n}=T$, is given on $[0, T]$ with equidistant increments $\Delta t$. The work [21] gives the L1 scheme in the calculation of the differentiation in the fractional RL sense on $a$ as follows:

$$
\begin{equation*}
{ }_{0}^{R} D_{t}^{\alpha}\left(a\left(t_{n}\right)\right)=\Delta t^{-\alpha} \sum_{j=0}^{n} f_{j, n} a\left(t_{n-j}\right)+\mathcal{O}\left(\Delta t^{2-\alpha}\right) \tag{6}
\end{equation*}
$$

where the weights $f_{j, n}$ can be extracted from

$$
\Gamma(2-\alpha) f_{j, n}= \begin{cases}(-1+j)^{1-\alpha}+(j+1)^{-\alpha+1}-2 j^{1-\alpha}, & 1 \leq j \leq n-1  \tag{7}\\ -j^{1-\alpha}+(j-1)^{-\alpha+1}, & j=n \\ 1, & j=0\end{cases}
$$

Noting that for the uniform stencils, the L1 scheme might be furnished as follows:

$$
\begin{equation*}
{ }_{0}^{C} D_{t}^{\alpha}\left(b\left(y_{i}, t_{k+1}\right)\right)=\frac{\Delta t^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{k} \frac{b_{i, j+1}-b_{i, j}}{\Delta t}\left(-(j)^{-\alpha+1}+(1+j)^{-\alpha+1}\right) . \tag{8}
\end{equation*}
$$

### 2.2. RBF-FD Formulation

Let us assume $L$ as a linear operator. For calculating $\mu_{i}$ in the RBF-FD at $\underline{y}=\underline{y}_{p}$ for the point locations $\underline{y}_{i^{\prime}}$, we write down [22]:

$$
\left[\begin{array}{cccc}
\Lambda_{1}\left(\underline{y}_{1}\right) & \Lambda_{1}\left(\underline{y}_{2}\right) & \cdots & \Lambda_{1}\left(\underline{y}_{m}\right)  \tag{9}\\
\Lambda_{2}\left(\underline{y}_{1}\right) & \Lambda_{2}\left(\underline{y}_{2}\right) & \cdots & \Lambda_{2}\left(\underline{y}_{m}\right) \\
\vdots & \vdots & & \vdots \\
\Lambda_{m}\left(\underline{y}_{1}\right) & \Lambda_{m}\left(\underline{y}_{2}\right) & \cdots & \Lambda_{m}\left(\underline{y}_{m}\right)
\end{array}\right]\left[\begin{array}{c}
\mu_{1} \\
\mu_{2} \\
\vdots \\
\mu_{m}
\end{array}\right]=\left[\begin{array}{c}
\left.L \Lambda_{1}(\underline{y})\right|_{\underline{y}=\underline{y}_{p}} \\
\left.L \Lambda_{2}(\underline{y})\right|_{\underline{y}=\underline{y}_{p}} \\
\vdots \\
\left.L \Lambda_{m}(\underline{y})\right|_{\underline{y}_{1}=\underline{y}_{p}}
\end{array}\right]
$$

where $\underline{y}$ indicates a vector quantity in $d$ as its dimension for $\Lambda(\underline{y})$. Here, the weights $\mu_{i}$ are the RBF-FD weights obtained by considering a stencil and then imposing the RBF and its derivatives to find them. As discussed in [22], for the appropriate choice of the shape parameter and the RBF, the resulting linear system obtained through (9) is invertible and the weights can be obtained both theoretically and numerically. We here focus on finding and using the weights in their closed form since finding them numerically will cause both higher CPU time and instability. Here, although $L$ is a linear operator, it can be proved that to obtain the weights in the RBF-FD methodology, this operator will finally only act on the right-hand side vector, and the matrix in (9) is constructed based on the stance matrix.

### 2.3. Spatial Discretization Nodes

Now, we take into account the famous GMQ RBF as comes next [23,24]:

$$
\begin{equation*}
\Lambda\left(r_{i}\right)=\left(p^{2}+r_{i}^{2}\right)^{l}, \quad 1 \leq i \leq n \tag{10}
\end{equation*}
$$

for a suitable value, $l, r_{i}=\left\|y-y_{i}\right\|$ indicates the Euclidean distance and $p$ is the parameter of shape. In this work, we examine a stencil comprising $n$ scattered nodes. To determine the weights for the RBF-FD methodology, we formulate the following expression at a given point $y=y_{j}, 1 \leq j \leq n$ :

$$
\begin{equation*}
L\left[\Lambda\left(\left\|y_{k}-y_{j}\right\|\right] \simeq \sum_{i=1}^{\psi} \mu_{i} \Lambda\left(\left\|y_{k}-y_{i}\right\|\right), \quad 1 \leq k \leq \psi\right. \tag{11}
\end{equation*}
$$

where $\psi$ is the number of knots inside the stencil. The purpose here is to obtain the weights of the GMQ RBF-FD scheme for a stencil with five discretization points. Hence, it is taken into account that:

$$
\begin{equation*}
\left\{y_{i}-2 h, y_{i}-h, y_{i}, y_{i}+h, y_{i}+2 h\right\}, \quad h>0 \tag{12}
\end{equation*}
$$

is an equally-spaced stencil. To compute the weights associated with the estimation of the first derivative, one now considers

$$
\begin{equation*}
a^{\prime}\left(y_{i}\right) \simeq \sum_{j=i-2}^{i+2} \mu_{j} a\left(y_{j}\right)=\hat{a}^{\prime}\left(y_{i}\right) \tag{13}
\end{equation*}
$$

Here, $\mu_{j}$ are the weights in this case. Substituting the function $a$ by (10), centered at $y_{i-2}=y_{i}-2 h, y_{i-1}=y_{i}-h, y_{i}, y_{i+1}=y_{i}+h$, and $y_{i+2}=y_{i}+2 h$ gives a set of symbolic linear equations as follows:

$$
\begin{align*}
& 4 h l\left(p^{2}+4 h^{2}\right)^{l-1}=\mu_{i-2} p^{2 l}+\mu_{i-1}\left(p^{2}+h^{2}\right)^{l}+\mu_{i}\left(p^{2}+4 h^{2}\right)^{l} \\
& +\mu_{i+1}\left(p^{2}+9 h^{2}\right)^{l}+\mu_{i+2}\left(p^{2}+16 h^{2}\right)^{l}, \\
& 2 h l\left(p^{2}+h^{2}\right)^{l-1}=\mu_{i-1} p^{2 l}+\mu_{i-2}\left(p^{2}+h^{2}\right)^{l}+\mu_{i}\left(p^{2}+h^{2}\right)^{l} \\
& +\mu_{i+1}\left(p^{2}+4 h^{2}\right)^{l}+\mu_{i+2}\left(p^{2}+9 h^{2}\right)^{l}, \\
& \mu_{i} p^{2 l}+\mu_{i-1}\left(p^{2}+h^{2}\right)^{l}+\mu_{i+1}\left(p^{2}+h^{2}\right)^{l} \\
& +\mu_{i-2}\left(p^{2}+4 h^{2}\right)^{l}+\mu_{i+2}\left(p^{2}+4 h^{2}\right)^{l}=0,  \tag{14}\\
& -2 h l\left(p^{2}+h^{2}\right)^{l-1}=\mu_{i+1} p^{2 l}+\mu_{i}\left(p^{2}+h^{2}\right)^{l}+\mu_{i+2}\left(p^{2}+h^{2}\right)^{l} \\
& +\mu_{i-2}\left(p^{2}+9 h^{2}\right)^{l}+\mu_{i-1}\left(p^{2}+4 h^{2}\right)^{l}, \\
& -4 h l\left(p^{2}+4 h^{2}\right)^{l-1}=\mu_{i+2} p^{2 l}+\mu_{i+1}\left(p^{2}+h^{2}\right)^{l} \\
& +\mu_{i-2}\left(p^{2}+16 h^{2}\right)^{l}+\mu_{i-1}\left(p^{2}+9 h^{2}\right)^{l}+\mu_{i}\left(p^{2}+4 h^{2}\right)^{l} .
\end{align*}
$$

The system (14) can be written as: $\mathrm{H}\left(\mu_{i-2}, \mu_{i-1}, \mu_{i}, \mu_{i+1}, \mu_{i+2},\right)^{*}=-c$, where

$$
\mathrm{H}=\left(\begin{array}{ccccc}
-p^{2 l} & -\left(p^{2}+h^{2}\right)^{l} & -\left(p^{2}+4 h^{2}\right)^{l} & -\left(p^{2}+9 h^{2}\right)^{l} & -\left(p^{2}+16 h^{2}\right)^{l}  \tag{15}\\
-\left(p^{2}+h^{2}\right)^{l} & -p^{2 l} & -\left(p^{2}+h^{2}\right)^{l} & -\left(p^{2}+4 h^{2}\right)^{l} & -\left(p^{2}+9 h^{2}\right)^{l} \\
\left(p^{2}+4 h^{2}\right)^{l} & \left(p^{2}+h^{2}\right)^{l} & p^{2 l} & \left(p^{2}+h^{2}\right)^{l} & \left(p^{2}+4 h^{2}\right)^{l} \\
-\left(p^{2}+9 h^{2}\right)^{l} & -\left(p^{2}+4 h^{2}\right)^{l} & -\left(p^{2}+h^{2}\right)^{l} & -p^{2 l} & -\left(p^{2}+h^{2}\right)^{l} \\
-\left(p^{2}+16 h^{2}\right)^{l} & -\left(p^{2}+9 h^{2}\right)^{l} & -\left(p^{2}+4 h^{2}\right)^{l} & -\left(p^{2}+h^{2}\right)^{l} & -p^{2 l}
\end{array}\right),
$$

and

$$
c=\left(4 h\left(p^{2}+4 h^{2}\right)^{l-1} l, 2 h\left(p^{2}+h^{2}\right)^{l-1} l, 0,-2 h\left(p^{2}+h^{2}\right)^{l-1} l,-4 h\left(p^{2}+4 h^{2}\right)^{l-1} l\right)^{*} .
$$

Thus, the weighting coefficients of the GMQ RBF-FD technique to estimate $f^{\prime}$ when $p \gg h$ can be simplified and presented as follows:

$$
\begin{align*}
& \mu_{i-2}=\frac{p^{2}(2 l-7)-4 h^{2}(l-2)(2 l-9)}{12 p^{2} h(2 l-7)}  \tag{16}\\
& \mu_{i-1}=\frac{p^{2}(14-4 l)+2 h^{2}(l-2)(2 l-9)}{3 p^{2} h(2 l-7)}  \tag{17}\\
& \mu_{i}=0  \tag{18}\\
& \mu_{i-1}=-\mu_{i+1}  \tag{19}\\
& \mu_{i-2}=-\mu_{i+2} \tag{20}
\end{align*}
$$

Here, the term "simplified" means that the weights constructed through the RBF-FD methodology (11) are a bit bulky and contain many terms. Hence, to make the weights more revealing and useful in practice, we employ a Taylor expansion on them around zero by having the condition $p \gg h$. Now, the error equation can be obtained by putting back the weights into the approximation formula (13) and writing Taylor expansion up to the desired order:

$$
\begin{equation*}
\varepsilon\left(y_{i}\right)=\left(\frac{2(l-2)(2 l-9) a^{(3)}\left(y_{i}\right)}{3 p^{2}(2 l-7)}-\frac{1}{30} a^{(5)}\left(y_{i}\right)\right) h^{4}+\mathcal{O}\left(h^{5}\right) \tag{21}
\end{equation*}
$$

where $\varepsilon\left(y_{i}\right)=\hat{a}^{\prime}\left(y_{i}\right)-a^{\prime}\left(y_{i}\right)$.
For finding the approximation of the second derivative using RBF-FD methodology, we write

$$
\begin{equation*}
a^{\prime \prime}\left(y_{i}\right) \simeq \sum_{j=i-2}^{i+2} \beta_{j} a\left(y_{j}\right)=\hat{a}^{\prime \prime}\left(y_{i}\right) \tag{22}
\end{equation*}
$$

Similarly, as described above, we attain the weighting coefficients as follows:

$$
\begin{align*}
& \beta_{i-2}=\frac{1}{36}\left(-\frac{3}{h^{2}}+\frac{12(l-2)\left(8 l^{3}-92 l^{2}+294 l-273\right)}{p^{2}\left(2 l\left(4 l^{2}-38 l+99\right)-165\right)}-\eta_{1}\right),  \tag{23}\\
& \beta_{i-1}=\frac{2}{9}\left(\frac{6}{h^{2}}-\frac{6(l-2)\left(8 l^{3}-92 l^{2}+294 l-273\right)}{p^{2}\left(2 l\left(4 l^{2}-38 l+99\right)-165\right)}-\eta_{2}\right),  \tag{24}\\
& \beta_{i}=\frac{1}{6 p^{4} h^{2}\left(165-2 l\left(4 l^{2}-38 l+99\right)\right)^{2}}(\omega+\vartheta),  \tag{25}\\
& \beta_{i-1}=\beta_{i+1}  \tag{26}\\
& \beta_{i-2}=\beta_{i+2} . \tag{27}
\end{align*}
$$

with

$$
\begin{aligned}
& \eta_{1}=\frac{a_{1}}{p^{4}\left(165-2 l\left(4 l^{2}-38 l+99\right)\right)^{2}} \\
& \eta_{2}=\frac{a_{2}}{p^{4}\left(165-2 l\left(4 l^{2}-38 l+99\right)\right)^{2}}
\end{aligned}
$$

and

$$
\begin{aligned}
\omega= & 12 p^{2} h^{2}(l-2)\left(8 l^{3}-92 l^{2}+294 l-273\right)\left(2 l \left(4 l^{2}\right.\right. \\
& -38 l+99)-165)-15 p^{4}\left(165-2 l\left(4 l^{2}-38 l+99\right)\right)^{2}, \\
\vartheta= & 4 h^{4}(l-2)(2 l(2 l(2 l(2 l(2 l(2 l(2 l \\
& -61)+1427)-17237)+116451)-443571)+897345)-748035), \\
a_{1}= & 4 h^{2}(l-2)(4 l(l(4 l(l(4 l(l(4 l-83) \\
& +659)-9817)+16869)-28713)-58365)+228285), \\
a_{2}= & h^{2}(l-2)(l(4 l(l(4 l(l(4(l-50) l \\
& +2963)-20947)+315615)-651000)+2808765)-1236195) .
\end{aligned}
$$

The error equation, in this case, can be obtained as follows:

$$
\begin{align*}
\bar{\varepsilon}\left(y_{i}\right)= & \left(\left(30 l-2\left(-6(l-1)\left(2 l\left(4 l^{2}-54 l+211\right)-525\right) a^{\prime \prime}\left(y_{i}\right)\right.\right.\right. \\
& \left.+p^{2}\left(8 l^{3}-92 l^{2}+294 l-273\right) a^{(4)}\left(y_{i}\right)\right)  \tag{28}\\
& \left.\left.+p^{4}\left(165-2 l\left(4 l^{2}-38 l+99\right)\right) a^{(6)}\left(y_{i}\right)\right)\right) h^{4} / \mathrm{d}+\mathcal{O}\left(h^{5}\right),
\end{align*}
$$

where $\bar{\varepsilon}\left(y_{i}\right)=\hat{a}^{\prime \prime}\left(y_{i}\right)-a^{\prime \prime}\left(y_{i}\right)$ and

$$
\mathrm{d}=\left(90 p^{4}\left(2 l\left(4 l^{2}-38 l+99\right)-165\right)\right)
$$

The approximated formulas are used for the practical fractional PDE problem (3) and it is discussed how they can help our numerical method to possess faster convergence in Section 4. To judge the quality and effectiveness of the proposed method and the potential advantages with respect to the existing methods, it is also remarked that due to employing five adjacent nodes in (11), we are able to construct linear systems consisting of five equations with five unknowns theoretically and then obtain the weights by solving the system analytically. The weights can one time be programmed and then each time the five adjacent points change the weights will be changed and fill the appropriate row of a differentiation matrix. This procedure saves computational time as well as reduces the instability that occurs in finding the weights numerically for each set of five adjacent points. Although we focused on finding the weights for the first and second derivatives, the weights for higher-order derivatives can also be computed, which are not provided here since we focus on solving a second-order PDE. Furthermore, currently, we are unable to provide an analysis of the error bounds theoretically.

## 3. The Proposed Solver in Matrix Notation

By denoting $b_{i, j} \simeq b\left(y_{i}, t_{j}\right)$, we can write a full discretization scheme for (3) in what follows:

$$
\begin{align*}
D_{t}^{\alpha} b\left(y_{i}, t_{j+1}\right)+\mathcal{O}\left(\Delta t^{2-\alpha}\right)= & \frac{1}{2} \sigma^{2} y_{i}^{2}\left[\beta_{i-2, j} b_{i-2, j}+\beta_{i-1, j} b_{i-1, j}\right. \\
& \left.+\beta_{i, j} b_{i, j}+\beta_{i+1, j} b_{i+1, j}+\beta_{i+2, j} b_{i+2, j}\right]+\mathcal{O}\left(h^{4}\right) \\
& +(r-q) y_{i}\left[\mu_{i-2, j} b_{i-2, j}+\mu_{i-1, j} b_{i-1, j}+\mu_{i, j} b_{i, j}\right.  \tag{29}\\
& \left.+\mu_{i+1, j} b_{i+1, j}+\mu_{i+2, j} b_{i+2, j}\right]+\mathcal{O}\left(h^{4}\right) \\
& -r b_{i, j} .
\end{align*}
$$

It is now possible to write down

$$
\begin{equation*}
D_{t}^{\alpha} b\left(y_{i}, t_{j+1}\right)=(1 /(\Gamma(-\alpha+2))) \sum_{l=0}^{j} \frac{b_{i, l+1}-b_{i, l}}{\Delta t_{l+1}}\left(\left(-t_{l}+t_{j+1}\right)^{1-\alpha}-\left(t_{j+1}-t_{l+1}\right)^{1-\alpha}\right) . \tag{30}
\end{equation*}
$$

Noting that the application of the fourth-order RBF-FD formulations given in Section 2.3 will arise in sparse differentiation matrices having penta-diagonal structure.

To follow our procedure more carefully and make the contribution by furnishing an efficient method in matrix notations, let us put all the differentiation weights into appropriate matrices of proper dimensions. For the weights associated with the first and the second derivatives, we can write down

$$
M_{y}= \begin{cases}\mu_{i, j} \text { with (16) } & i-j=2  \tag{31}\\ \mu_{i, j} \text { with (17) } & i-j=1 \\ \mu_{i, j} \text { with (18) } & i-j=0, \\ \mu_{i, j} \text { with (19) } & j-i=1 \\ \mu_{i, j} \text { with (20) } & j-i=2 \\ 0 & \text { otherwise }\end{cases}
$$

and

$$
M_{y y}= \begin{cases}\beta_{i, j} \text { with (23) } & i-j=2,  \tag{32}\\ \beta_{i, j} \text { with (24) } & i-j=1, \\ \beta_{i, j} \text { with (25) } & i-j=0, \\ \beta_{i, j} \text { with (26) } & j-i=1, \\ \beta_{i, j} \text { with (27) } & j-i=2, \\ 0 & \text { otherwise. }\end{cases}
$$

It is remarked that for the nodes $\left\{y_{1}, y_{2}, y_{m-1}, y_{m}\right\}$ in which the central RBF-D formulas given in Section 2.3 are not valid, one may derive similar RBF-FD weights using similar methodology given therein. Now, we are in a position to write down and fill up the following matrix:

$$
\begin{equation*}
A=-r I_{N}+\frac{1}{2} \sigma^{2} Y^{2}\left(M_{y y} \otimes I_{t}\right)+(r-q) Y\left(M_{y} \otimes I_{t}\right) \tag{33}
\end{equation*}
$$

wherein $\otimes$ stands for the product of Kronecker [25] and

$$
\begin{equation*}
Y=\operatorname{diag}\left(y_{1}, y_{2}, \cdots, y_{m}\right) \otimes I_{t} . \tag{34}
\end{equation*}
$$

Noticing that

$$
I_{N}=I_{y} \otimes I_{t}
$$

is an $N \times N$ unit matrix, while $I_{t}$ and $I_{y}$ are $n \times n$ and $m \times m$ unit matrices, respectively. Therefore, the time-fractional PDE (29) can be discretized fully through:

$$
\begin{equation*}
\left(I_{y} \otimes M_{t}\right) B=A B, \tag{35}
\end{equation*}
$$

wherein

$$
B=\left(b_{1,1}, b_{1,2}, \ldots, b_{m, n}\right)^{*}
$$

Let us consider:

$$
\begin{equation*}
\mathcal{A}=\left(I_{y} \otimes M_{t}\right)-A \tag{36}
\end{equation*}
$$

Therefore, we obtain $\mathcal{A B}=0$. Noting that now for put and call options, the boundary conditions could be written, respectively, as [20]:

$$
\begin{gather*}
b(0, t)=K \exp \{-r t\}, \lim _{y \rightarrow \infty} b(y, t)=0,  \tag{37}\\
b(0, t)=0, \lim _{y \rightarrow \infty} b(y, t)=y_{\max } \exp (-q t)-K \exp (-r t), \tag{38}
\end{gather*}
$$

wherein $y_{\max }$ is a large positive scalar. Ultimately, the whole numerical discretization procedure for resolving the time-fractional BS PDE (29) could be simplified in the form of an algebraic linear system of equations as comes next:

$$
\begin{equation*}
\overline{\mathcal{A}} B=\bar{b}, \tag{39}
\end{equation*}
$$

where, respectively, $\overline{\mathcal{A}}$ and $\bar{b}$ are the system matrix and its associated vector after imposing the initial and boundary conditions. Moreover, $B$ contains the vector of unknowns after having the Kronecker product. After having a theoretical discussion in this section about how to construct the proposed method in matrix form as a linear system of equations, it is now time to move on and test it on different types of input values on (3). This is the concept of the next section.

The differentiation matrix $M_{t}$ corresponding to the L1 scheme is an $M$-matrix (i.e., it is a Z-matrix with eigenvalues whose real parts are non-negative), which makes $I_{y} \otimes M_{t}$ invertible. On the other hand, $A$ in (33) is non-singular since no zero rows two so much close knots are thereby in the spatial discretization (under appropriate choices of the financial parameters in (3)). Thus, the system matrix $\overline{\mathcal{A}}$ in (39) is invertible.

The whole computational high-order method can be summarized in Algorithm 1.

```
Algorithm 1 Solving the temporal-fractional BS PDE
Require: The boundary conditions as well as the initial data are given.
    Consider the 1D domain for the spatial variable as well as the shape parameter in (10).
    Distribute uniform/non-uniform points for discretization along time and discretize the
    fractional temporal variable as in Section 2.1.
    Distribute uniform points for discretization along space and then impose the weights
    given in Section 2.3.
    First, build up the matrix (33) and then (36).
    Impose the boundary conditions (37) or (38) to obtain the final linear algebraic system
    of equations (39).
    6: Solve the system using the command LineaSolve[] in Mathematica.
```


## 4. Computational Results

This section is devoted to re-check the discussions given in Sections 2 and 3 by pricing some financial options when $y_{\max }=3 K$. To this goal, several famous solvers are compared such as:

- The fundamental FD method on uniform meshes with quadratic convergence rate along space and the first order of discretization along time given in [18] and denoted by FDM2.
- The first published work based on the RBF-FD methodology on equally-spaced meshes for solving (3) in [20] and shown by RBF-FD2.
- Our proposed solver based on uniform meshes but with quartical convergence along space and the RBF-FD methodology and first-order discretization along the fractional temporal variable shown by RBF-FD4.
Our laptop specifications are Windows 11 with Intel Core i7-12700H RTX 3070. Mathematica 13.0 [26,27] is employed to do the implementations and solve the linear system of equations using automatic built-in commands. The elapsed CPU time is furnished in seconds. Additionally, the absolute error is given by:

$$
\begin{equation*}
\text { error }=\left|b_{\text {numerical }}-b_{\text {ref }}\right|, \tag{40}
\end{equation*}
$$

wherein $b_{\text {ref }}$ and $b_{\text {numerical }}$ are the referenced and computational solutions, respectively.
One of the effective ways for obtaining a proper shape parameter can be used as $c=4 h$.

Example 1. Using the following set of parameters, here we investigate the efficacy of different methods for call option pricing:

$$
\begin{equation*}
T=1 \text { year, } r=5 \% \sigma=40 \%, q=0, K=100 \$, \alpha=0.8 \tag{41}
\end{equation*}
$$

The price of reference is $b(K, T) \simeq 18.1883$, [20].
Table 1 is provided to compare the behavior of various solvers in resolving the fractional problem (3). Three types of solvers have been compared. The standard second-order FD method along with the second-order RBF-FD method over non-uniform stencils and the proposed high-order RBF-FD scheme with fourth order of convergence. The presented scheme beats the other solvers in terms of accuracy and CPU time. To also check the stability and positivity of the presented method, it is necessary to observe the numerical solutions for different numbers of discretization nodes in the computational domain. This is performed in Figure 1. In Figure 1, we give the numerical solution based on two sets of discretization points. The knots are uniform. The list point plots are given on the left, which is the numerical solution in the discrete format while the right figures have been drawn by passing an (automatically generated via the software Mathematica 12.0) interpolated solution in the continuous mode.


Figure 1. Numerical reports in Example 1 for RBF-FD4. (Top Left): The point plot when $n=m=10$. (Top Right): The continuous plot when $n=m=10$. (Bottom Left): The point plot when $n=m=40$. (Bottom Right): The continuous plot when $n=m=40$.

Table 1. Computational comparisons given for various solvers under Example 1.

| $\boldsymbol{m}, \boldsymbol{n}$ | $\boldsymbol{b}_{F D M 2}$ | error $_{F D M 2}$ | $\mathrm{~T}_{\mathrm{FDM} 2}$ | $\boldsymbol{b}_{\text {RBF-FD2}}$ | error $_{\text {RBF-FD2 }}$ | $\mathrm{T}_{\text {RBF-FD2 }}$ | $\boldsymbol{b}_{\text {RBF-FD4 }}$ | error $_{\text {RBF-FD4 }}$ | $\mathrm{T}_{\text {RBF-FD4 }}$ |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 10 | 16.070 | 2.11 E 0 | 0.01 | 17.378 | $8.10 \mathrm{E}-1$ | 0.01 | 17.219 | $9.69 \mathrm{E}-1$ | 0.01 |
| 20 | 17.889 | $2.98 \mathrm{E}-1$ | 0.02 | 17.694 | $4.94 \mathrm{E}-1$ | 0.02 | 17.810 | $3.78 \mathrm{E}-1$ | 0.03 |
| 40 | 17.913 | $2.75 \mathrm{E}-1$ | 0.16 | 18.003 | $1.85 \mathrm{E}-1$ | 0.18 | 18.102 | $8.63 \mathrm{E}-2$ | 0.20 |
| 80 | 18.039 | $1.48 \mathrm{E}-1$ | 3.96 | 18.213 | $2.46 \mathrm{E}-2$ | 3.84 | 18.169 | $1.93 \mathrm{E}-2$ | 4.18 |
| 120 | 18.055 | $1.33 \mathrm{E}-1$ | 20.02 | 18.190 | $1.64 \mathrm{E}-3$ | 20.96 | 18.186 | $2.35 \mathrm{E}-3$ | 21.67 |

Example 2. Using the following set of parameters, here we investigate the efficacy of the different for the put option pricing:

$$
\begin{equation*}
q=20 \%, \sigma=30 \%, T=2 \text { year, } r=5 \%, K=100 \$, \alpha=0.7 . \tag{42}
\end{equation*}
$$

The reference price is $b(K, T) \simeq 18.1883,[20]$.
Computational reports for this test are furnished in Table 2 reconfirming the efficacy of the proposed RBF-FD solver with fourth order of convergence along space. Furthermore, we showcase the numerical solution in Figure 2, demonstrating the robustness and stability of our proposed solver while also highlighting its inherent positivity. Again, in Figure 2, when $m=n=21$ uniform knots are considered, and Algorithm 1 is followed in order to obtain the numerical solution. The left figure shows the numerical solution in the discrete mode while the right one is the continuous interpolated numerical solution.

The rationale behind using $\alpha=0.7$ or $\alpha=0.8$ is to consider the fractional value of the derivative in (3) and test the efficiency of the proposed method for such values, which makes the Hurst index two different values on the fractional Brownian motion in modeling the data.

Table 2. Computational comparisons given for various solvers under Example 2.

| $\boldsymbol{m}, \boldsymbol{n}$ | $\boldsymbol{b}_{\boldsymbol{F D M 2}}$ | error $_{\boldsymbol{F D M} 2}$ | $\mathrm{~T}_{\text {FDM2 }}$ | $\boldsymbol{b}_{\boldsymbol{R B F - F D 2}}$ | error $_{\text {RBF-FD2 }}$ | $\mathrm{T}_{\text {RBF-FD2 }}$ | $\boldsymbol{b}_{\text {RBF-FD4 }}$ | error $_{\text {RBF-FD4 }}$ | $\mathrm{T}_{\text {RBF-FD4 }}$ |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 11 | 23.914 | $7.1 \mathrm{E}-1$ | 0.01 | 23.631 | $9.9 \mathrm{E}-1$ | 0.03 | 23.709 | $9.20 \mathrm{E}-1$ | 0.03 |
| 21 | 24.316 | $3.1 \mathrm{E}-1$ | 0.03 | 24.117 | $5.1 \mathrm{E}-1$ | 0.05 | 24.218 | $4.11 \mathrm{E}-1$ | 0.06 |
| 41 | 24.466 | $1.6 \mathrm{E}-1$ | 0.15 | 24.356 | $2.7 \mathrm{E}-1$ | 0.21 | 24.551 | $7.89 \mathrm{E}-2$ | 0.23 |
| 81 | 24.545 | $8.4 \mathrm{E}-2$ | 4.14 | 24.565 | $6.4 \mathrm{E}-2$ | 4.69 | 24.603 | $2.69 \mathrm{E}-2$ | 4.97 |
| 161 | 24.580 | $4.9 \mathrm{E}-2$ | 63.97 | 24.636 | $6.1 \mathrm{E}-3$ | 64.01 | 24.626 | $3.90 \mathrm{E}-3$ | 65.69 |



Figure 2. In Example 2, we present numerical results for RBF-FD4. On the (left), we provide a point plot for the case of $m=n=21$. On the (right), we display a continuous plot of the solution for the same set of $m=n=21$.

## 5. Concluding Remarks

In this work, to solve temporal-fractional option pricing PDE problems arising in finance, we introduced the weights for the GMQ RBF-FD methodology on stencils having five uniform discretization points. The temporal-fractional PDE problem was transformed by fourth-order discretizations coming from the RBF-FD methodology along the spatial variable and the L1 scheme for the discretization along the time variable. The solver was proposed in matrix notations to simplify the whole procedure. To the best of our knowledge, this is the first paper that employed fourth-order local RBF-FD approximations on solving the temporal fractional BS PDE under the GMQ RBF. Numerical tests were considered and solved in Section 4. Computational results clearly showed that the proposed solver increase the accuracy of the final approximations. Some further remarks are in order:

- Forthcoming works might concentrate on deriving super-efficient solvers based on the Hermite idea to increase the speed of convergence without adding further stencil nodes in solving time-dependent PDEs.
- We too investigate extending the applicability of the L2 scheme in estimating the temporal fractional derivative to obtain higher-order accurate approximations for the time-fractional PDE (29).

Author Contributions: Conceptualization, M.Z.U., A.K.A. and H.M.A.; formal analysis, A.K.A. and H.M.A.; funding acquisition, M.Z.U. and S.S.; investigation, A.K.A. and S.S.; methodology, A.K.A. and H.M.A.; supervision, M.Z.U. and S.S.; validation, M.Z.U., A.K.A. and S.S.; writing-original draft, S.S. and H.M.A.; writing-review \& editing, A.K.A. and H.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research recieved no external funding.
Institutional Review Board Statement: Not applicable.
Informed Consent Statement: Not applicable.
Data Availability Statement: Not applicable.
Acknowledgments: This research work was funded by Institutional Fund Projects under grant no. (IFPIP: 1316-130-1443). The authors gratefully acknowledge technical and financial support provided by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

Conflicts of Interest: The authors confirm that they have no personal relationships or competing financial interests that could have influenced the integrity and objectivity of the research presented in this manuscript.

## References

1. Jumarie, G. Modified Reimann-Liouville derivative and fractional Taylor series of non-differentiable functions further results. Comput. Math. Appl. 2006, 51, 1367-1376. [CrossRef]
2. Caputo, M. Linear model of dissipation whose $Q$ is almost frequency independent II. Geophys. J. Int. 1967, 13, 529-539. [CrossRef]
3. Wyss, W. The fractional Black-Scholes equation. Fract. Calc. Appl. Anal. 2000, 3, 51-62.
4. 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]
5. Hurst, H.E. Long-term storage capacity of reservoirs. Trans. Amer. Soc. Civil Eng. 1951, 116, 770-799. [CrossRef]
6. Seydel, R.U. Tools for Computational Finance, 6th ed.; Springer: London, UK, 2017.
7. Norros, I.; Valkeila, E.; Virtamo, J. A Girsanov-type formula for the fractional Brownian motion. In Proceedings of the First Nordic-Russian Symposium on Stochastics, Helsinki, Finland, 14-16 August 1996.
8. Norros, I. Four approaches to the fractional Brownian storage. In Fractals in Engineering; Lévy Véhel, J., Lutton, E., Tricot, C., Eds.; Springer: London, UK, 1997, pp. 154-169.
9. Feyel, D.; de la Pradelle, A. On fractional Brownian processes. Potential Anal. 1999, 10, 273-288. [CrossRef]
10. Duffy, D.J. Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach; Wiley: Hoboken, NJ, USA, 2006.
11. Heston, S.L. A closed-form solution for options with stochastic volatility with applications to bond and currency options. Rev. Finan. Stud. 1993, 6, 327-343. [CrossRef]
12. Itkin, A. Pricing Derivatives Under Lévy Models: Modern Finite-Difference and Pseudo-Differential Operators Approach; Birkhäuser: Basel, Switzerland, 2017.
13. An, X.; Liu, F.; Zheng, M.; Anh, V.V.; Turner, I.W. A space-time spectral method for time-fractional Black-Scholes equation. Appl. Numer. Math. 2021, 165, 152-166. [CrossRef]
14. Roul, P.; Prasad Goura, V.M.K. A compact finite difference scheme for fractional Black-Scholes option pricing model. Appl. Numer. Math. 2021, 166, 40-60. [CrossRef]
15. Hon, Y.C. A quasi-radial basis functions method for American options pricing. Comput. Math. Appl. 2002, 43, 513-524. [CrossRef]
16. Bollig, E.F.; Flyer, N.; Erlebacher, G. Solution to PDEs using radial basis function finite-differences (RBF-FD) on multiple GPUs. J. Comput. Phys. 2012, 231, 7133-7151. [CrossRef]
17. Liu, T.; Ullah, M.Z.; Shateyi, S.; Liu, C.; Yang, Y. An efficient localized RBF-FD method to simulate the Heston-Hull-White PDE in finance. Mathematics 2023, 11, 833. [CrossRef]
18. 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]
19. Yang, P.; Xu, Z. Numerical valuation of European and American options under fractional Black-Scholes model. Fractal Fract. 2022, 6, 143. [CrossRef]
20. Song, Y.; Shateyi, S. Inverse multiquadric function to price financial options under the fractional Black-Scholes model. Fractal Fract. 2022, 6, 599. [CrossRef]
21. Diethelm, K. An algorithm for the numerical solution of differential equations of fractional order. Electron. Trans. Numer. Anal. 1997, 5, 1-6.
22. Bayona, V.; Moscoso, M.; Carretero, M.; Kindelan, M. RBF-FD formulas and convergence properties, J. Comput. Phys. 2010, 229, 8281-8295. [CrossRef]
23. Sarra, S.A.; Chenoweth, M.E. A numerical study of generalized multiquadric radial basis function interpolation. SIAM Undergrad. Res. Online 2009, 22, 58-70.
24. Soleymani, F.; Ullah, M.Z. A multiquadric RBF-FD scheme for simulating the financial HHW equation utilizing exponential integrator. Calcolo 2018, 55, 51. [CrossRef]
25. Zhang, H.; Ding, F. On the Kronecker products and their applications. J. Appl. Math. 2013, 2013, 296185. [CrossRef]
26. Soleymani, F.; Akgül, A. European option valuation under the Bates PIDE in finance: A numerical implementation of the Gaussian scheme. Disc. Cont. Dyn. Sys. Ser. S 2020, 13, 889-909. [CrossRef]
27. Wellin, P.R.; Gaylord, R.J.; Kamin, S.N.- An Introduction to Programming with Mathematica; Cambridge University Press: Cambridge, UK, 2005.

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.

