mathematics

MDPI

Article

A Novel and Accurate Algorithm for Solving Fractional Diffusion-Wave Equations

Haifa Bin Jebreen [1,*] and Ioannis Dassios [2]

[1] Department of Mathematics, College of Science, King Saud University, P.O. Box 2455, Riyadh 11451, Saudi Arabia
[2] Faculty of Engineering, Aristotle University of Thessaloniki, 541 24 Thessaloniki, Greece; ioannisdassios@gmail.com
* Correspondence: hjebreen@ksu.edu.sa

Abstract: The main objective of this work is to apply a novel and accurate algorithm for solving the second-order and fourth-order fractional diffusion-wave equations (FDWEs). First, the desired equation is reduced to the corresponding Volterra integral equation (VIE). Then, the collocation method is applied, for which the Chebyshev cardinal functions (CCFs) have been considered as the bases. In this paper, the CCFs based on a Lobatto grid are introduced and used for the first time to solve these kinds of equations. To this end, the derivative and fractional integral operators are represented in CCFs. The main features of the method are simplicity, compliance with boundary conditions, and good accuracy. An exact analysis to show the convergence of the scheme is presented, and illustrative examples confirm our investigation.

Keywords: Chebyshev cardinal functions; lobatto grid; fractional wave equation; collocation method

MSC: 54A25; 65N35; 35L05; 35R11

check for updates

Citation: Bin Jebreen, H.; Dassios, I. A Novel and Accurate Algorithm for Solving Fractional Diffusion-Wave Equations. *Mathematics* **2024**, *12*, 3307. https://doi.org/10.3390/math12213307

Academic Editors: Dongfang Li, Hongyu Qin and Duarte Valério

Received: 22 September 2024
Revised: 11 October 2024
Accepted: 15 October 2024
Published: 22 October 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

Recently, the modeling of various physical phenomena greatly relies on the significance of fractional differential equations (FDEs), making this subject a focal point for many researchers [1–4]. Numerous mathematical schemes have been introduced and applied to study and solve FDEs, including the adaptive grid technique [5], the B-spline collocation method [6], the wavelet method [7,8], multi-step methods [9], fast second-order accurate difference schemes [10], the Petrov–Galerkin finite element-meshfree formulation [11], the finite element method [12], an implicit integration factor method [13], Adomian decomposition [14], the Kuratowski measure of the noncompactness technique [15], etc.

The FDWE serves as a valuable mathematical model for representing various significant physical phenomena. The foundation of this model lies in a linear integro-partial fractional differential equation derived from the classical diffusion-wave equation. Notably, in its formulation, the conventional second-order time derivative is substituted with a fractional derivative of order κ ($\kappa \in (1,2]$). The FWDE accurately describes numerous universal mechanical, electromagnetic, and acoustic responses; see, e.g., [16]. The fourth-order spatial derivative is accrued in the context of wave propagation in beams and the modeling of grooves on a flat surface. As a result, substantial attention has been focused on the fourth-order FDWE and its practical applications [17]. The second-order and fourth-order FDWEs have been considered in this paper as follows.

Our objective in this study was to solve the second-order FDWE,

$$ {}_0^C\mathcal{D}_t^{\kappa}(y)(x,t) = \mathcal{D}_x^2(y)(x,t) + g(x,t), \quad t \in [0,T],\ x \in [0,b], \tag{1} $$

with

$$\begin{cases} y(x,0) = \phi_0(x), & \mathcal{D}_t(y)(x,0) = \phi_1(x), & x \in [0,b], \text{ (initial conditions)}, \\ y(0,t) = \psi_0(t), & y(1,t) = \psi_1(t), & t \in [0,T], \text{ (boundary conditions)}, \end{cases}$$

and the fourth-order FDWE,

$$ {}_0^C\mathcal{D}_t^{\kappa}(y)(x,t) + \mathcal{D}_x^4(y)(x,t) = g(x,t), \quad t \in [0,T],\ x \in [0,b], \tag{2}$$

with

$$\begin{cases} y(x,0) = \phi_0(x), & \mathcal{D}_t(y)(x,0) = \phi_1(x), & x \in [0,b], \text{ (initial conditions)}, \\ y(0,t) = \psi_0(t), & y(1,t) = \psi_1(t), & \\ \mathcal{D}_x^2(y)(0,t) = \psi_2(t), & \mathcal{D}_x^2(y)(1,t) = \psi_3(t), & \text{(boundary conditions)}, \end{cases}$$

where $\kappa \in (1,2]$, and $\mathcal{D}$ indicates the derivative operator.

Several schemes have been introduced and implemented to solve these kinds of equations, including a compact difference scheme [18,19], the finite difference scheme [20], the Sumudu transform method [21], high-order compact finite difference methods [22], the implicit difference scheme [23], and the spectral tau method [24].

We strive to present a novel and accurate algorithm for solving Equations (1) and (2). After converting the desired equation into a Volterra integral equation, the next step is to solve it using the collocation method that relies on the CCFs. The CCFs are the powerful bases for solving a variety of equations [6,25,26]. In this work, the CCFs obtained from a Lobatto grid are used for the first time. Notably, the Lobatto grid is popular for solving boundary value problems because the boundary conditions instantly furnish two grid point values for the unknown. On the other hand, when the differential equation is singular at the endpoints, then the Lobatto grid is preferable.

The upcoming portions of the document are arranged as follows: Section 2 contains a review and introduction of CCFs and their properties. The collocation method is applied to solve second-order and fourth-order FDWEs using CCFs in Section 3. This section includes an investigation of the convergence analysis as well. Section 4 is dedicated to illustrating the accuracy and applicability of the method. To sum up our work, we included a conclusion in Section 5.

2. Chebyshev Cardinal Polynomials Based on Lobatto Grid

Let us denote the Chebyshev polynomials of order N by T_N. As we know, T_N has N real, simple, and different roots, sometimes called Chebyshev nodes. The Chebyshev nodes can be set as $\{x_n\}_{n=1}^N$. Defining the Chebyshev polynomials onto the interval $[a,b]$, called shifted Chebyshev polynomials and characterized by T_N^*, is easy, and can be carried out through an affine transformation $x \mapsto \frac{2(x-a)}{b-a} - 1$ that maps $[-1,1]$ into $[a,b]$, viz.,

$$T_N^*(x) := T_N\left(\frac{2(x-a)}{b-a} - 1\right). \tag{3}$$

Consequently, the roots of T_N^* are obtained through

$$x_n^* = \frac{1}{2}(x_n + 1)(b-a) + a, \quad \forall 1 \le n \le N. \tag{4}$$

The CCFs defined rely on two types of grids of the Chebyshev nodes. The first definition is associated with the Gauss–Chebyshev nodes of T_{N+1}^*, and the second involves the Lobatto grid, which includes the extrema of Chebyshev polynomial T_N^* along with

the endpoints. Considering $\mathcal{D}_x$ as the derivative operator with respect to x, the CCFs corresponding to the Gauss–Chebyshev grid [25–27] are stated by the following:

$$C_n(x) = \frac{T^*_{N+1}(x)}{\mathcal{D}_x(T^*_{N+1})(x^*_n)(x - x^*_n)}, \quad x \in [a,b],\ 1 \le n \le N+1. \tag{5}$$

As mentioned above, there is another definition of CCFs with a substitute of nodes, namely the Lobatto grid, which are determined as follows:

$$C_n(x) = \frac{X(x)\mathcal{D}_x(T^*_N)(x)}{\mathcal{D}_x(XT^*_N)(x^*_n)(x - x^*_n)}, \quad x \in [a,b],\ 1 \le n \le N+1, \tag{6}$$

in which $X(x) = 1 - (\frac{2(x-a)}{b-a} - 1)^2$.

For $\nu \in \mathbb{N}$, the Sobolev space $\mathcal{H}^\nu([a,b])$ is denoted by

$$\mathcal{H}^\nu([a,b]) = \left\{ y \in C^\nu([a,b]) : \mathcal{D}^\sigma_x y \in L^2([a,b]), \sigma \le \nu, \sigma \in \mathbb{N}_0 \right\},$$

endowed with the inner product

$$(y_1, y_2)_{\mathcal{H}^\nu([a,b])} = \sum_{\sigma=0}^{\nu} (\mathcal{D}^\sigma_x(y_1), \mathcal{D}^\sigma_x(y_2))_{L^2([a,b])}, \tag{7}$$

which induces the norm

$$\|y\|^2_{\mathcal{H}^\nu([a,b])} = \sum_{\sigma=0}^{\nu} \|\mathcal{D}^\sigma_x(y)(x)\|^2_2, \tag{8}$$

and the semi-norm

$$|y|^2_{\mathcal{H}^{\nu,N}([a,b])} = \sum_{\sigma=\min\{\nu,N\}}^{N} \|\mathcal{D}^\sigma_x(y)(x)\|^2_2. \tag{9}$$

The remarkable characteristic of the CCFs is their cardinality, i.e.,

$$C_n(x_{n'}) = \delta_{nn'}, \qquad 1 \le n, n' \le N+1, \tag{10}$$

where $\delta_{nn'}$ denotes the Kronecker delta. The significance of this property lies in its ability to help determine coefficients without the need for calculating integrals when approximating any function $y \in \mathcal{H}^\nu([a,b])$ with them. Introducing a projection operator $\mathcal{Q}_N$ is essential to map $y \in C[a,b]$ into P_{N+1} (the space of all polynomials of degree less than $N+1$), i.e.,

$$y(x) \approx \mathcal{Q}_N(y)(x) = \sum_{n=1}^{N+1} y(x^*_n) C_n(x) \in P_{N+1}. \tag{11}$$

Lemma 1 (cf [28]). *Given $N \ge 0$, if $y \in \mathcal{H}^\nu([a,b])$, we have*

$$\|y - \mathcal{Q}_N(y)\|_2 \le C(b-a)^\nu N^{-\nu} |y|_{\mathcal{H}^{\nu,N}([a,b])}, \tag{12}$$

where C is a constant independent of N.

2.1. Matrix Representation of the Derivative Operator in CCFs Obtained by Lobatto Grids

Consider $\mathbf{C}(x)$ as a vector function whose components are $\{C_n\}_{1 \le n \le N+1}$. The focus of this part is to specify a square matrix D, denoted as

$$\mathcal{D}_x(\mathbf{C})(x) = D\mathbf{C}(x). \tag{13}$$

The process below outlines how to specify the entries for D. Thanks to Equation (11), it is easy to confirm that

$$[D]_{n,n'} = \mathcal{D}_x(C_n)(x^*_{n'}). \tag{14}$$

According to the definition of the CCFs using the Lobbato grid, an alternate formulation can be presented (see, e.g., [29]):

$$C_n(x) = \rho_n \prod_{\substack{n'=1 \\ n' \neq n}}^{N+1} (x - x^*_{n'}), \tag{15}$$

where

$$\rho_n = \frac{-4\Gamma(2N-2)}{(b-a)^{N+1}\Gamma^2(N-1)\big(\mathcal{D}_x(XT^*_N)(x^*_n)\big)}.$$

Motivated by (6), taking the derivative from both sides of (15) leads to the following:

$$\mathcal{D}_x(C_n)(x) = \rho_n \prod_{\substack{n'=1 \\ n' \neq n}}^{N+1} \mathcal{D}_x(x - x^*_{n'}). \tag{16}$$

- If $n = n'$, then we obtain

$$\begin{aligned} \mathcal{D}_x(C_n)(x) &= \rho_n \sum_{\substack{n''=1 \\ n'' \neq n}}^{N+1} \prod_{\substack{n'=1 \\ n' \neq n, n''}}^{N+1} (x - x^*_{n'}) \\ &= \sum_{\substack{n''=1 \\ n'' \neq n}}^{N+1} \frac{C_n(x)}{(x - x^*_{n''})}. \end{aligned} \tag{17}$$

From Equation (10), we conclude that

$$\mathcal{D}_x(C_n)(x^*_{n'}) = \sum_{\substack{n''=1 \\ n'' \neq n}}^{N+1} \frac{1}{(x_{n'} - x_{n''})}. \tag{18}$$

- If $n \neq n'$, then one can obtain

$$\mathcal{D}_x(C_n)(x^*_{n'}) = \rho_n \prod_{\substack{n''=1 \\ n'' \neq n, n'}}^{N+1} (x^*_{n'} - x^*_{n''}). \tag{19}$$

2.2. Matrix Representation of the Fractional Integral Operator in CCFs Obtained by Lobatto Grids

In recalling the fractional integral operator (FIO), see, e.g., [30,31],

$$\mathcal{I}^{\kappa}_0(y)(x) = \frac{1}{\Gamma(\kappa)} \int_0^x (x-s)^{\kappa-1} y(s) ds, \quad x \in [0, b], \quad \kappa \in \mathbb{R}^+, \tag{20}$$

the square matrix I_κ fulfills

$$\mathcal{I}^{\kappa}_0(\mathbf{C}(x)) \approx I_\kappa \mathbf{C}(x), \quad x \in (0, b). \tag{21}$$

Thanks to (11), one can calculate the elements of I_κ as

$$[I_\kappa]_{n,n'} = \mathcal{I}^{\kappa}_0(C_n(x^*_{n'})). \tag{22}$$

For simplicity, the alternate formula of (15) can be considered, introduced in [25], as follows:

$$C_n(x) = \rho_n \prod_{\substack{n'=1 \\ n' \neq n}}^{N+1} (x - x^*_{n'}) = \rho_n \sum_{n'=0}^{N} \varrho_{n,n'} x^{N-n'} \tag{23}$$

where

$$\varrho_{n,0} = 1, \varrho_{n,n'} = \frac{1}{n'} \sum_{n''=0}^{n'} \varsigma_{n,n''} \varrho_{n,n'-n''},\ n' = 1, \ldots, N, n = 1, \ldots, N+1,$$

with

$$\varsigma_{n,n''} = \sum_{\substack{i=1 \\ i \neq n}}^{N+1} (x_i^*)^{n''},\ n'' = 1, \ldots, N, n = 1, \ldots, N+1.$$

Taking into account the Property 2.1 [30] and using (23), one can write

$$\begin{aligned} \mathcal{I}_0^{\kappa}(C_n(x)) &= \rho_n \mathcal{I}_0^{\mu}\Big(\sum_{n'=0}^{N} \varrho_{n,n'} x^{N-n'}\Big) \\ &= \rho_n \sum_{n'=0}^{N} \varrho_{n,n'} \mathcal{I}_0^{\kappa}(x^{N-n'}) \\ &= \rho_n \sum_{n'=0}^{N} \varrho_{n,n'} \frac{(N-n')!}{\Gamma(N-n'+\kappa+1)} x^{N-n'+\kappa}. \end{aligned} \tag{24}$$

Consequently, (22) and (24) lead to determining

$$[I_{\kappa}]_{n,n'} = \rho_n \sum_{n'=0}^{N} \varrho_{n,n'} \frac{(N-n')!}{\Gamma(N-n'+\kappa+1)} (x_{n'}^*)^{N-n'+\kappa}. \tag{25}$$

3. Proposed Algorithm

Let us combine Equations (1) and (2) and refer to them as

$${}_0^C\mathcal{D}_t^{\kappa}(y)(x,t) + (-1)^{m/2}\mathcal{D}_x^m(y)(x,t) = g(x,t), \qquad \text{with} \quad m = 2, 4. \tag{26}$$

It is clear that Equation (26) corresponds to (1) when $m = 2$, and it corresponds to (2) when $m = 4$.

Taking into account the Lemma 2.22 [30] and taking the fractional integral from both sides of (26), one can obtain the corresponding integral equation, viz.,

$$y(x,t) - v(x,t) + \mathcal{I}_0^{\kappa}\Big((-1)^{m/2}\mathcal{D}_x^m(y) - g\Big)(x,t) = 0, \quad t \in [0,T],\ x \in [0,b], \tag{27}$$

where $1 < \kappa \leq 2$, and $v(x,t) = t\mathcal{D}_t(y)(x,0) + y(x,0)$. To give rise to the proposed algorithm, the solution y must be approximated by CCFs, i.e.,

$$y(x,t) \approx \mathcal{Q}_{N,N'}(y)(x,t) = \sum_{n=1}^{N+1} \sum_{n'=1}^{N'+1} y_{n,n'} C_n(x) C_{n'}(t) := \tilde{y}(x,t), \tag{28}$$

where N and N' can be different. Strictly speaking, they depend on our selection of the number of bases in each axis. Substituting $\tilde{y}$ in integral Equation (27) leads to

$$\tilde{y}(x,t) - \tilde{v}(x,t) + \mathcal{I}_0^{\kappa}\Big((-1)^{m/2}\mathcal{D}_x^m(\tilde{y}) - \tilde{g}\Big)(x,t) = 0, \tag{29}$$

where $\tilde{v}(x,t) = \mathcal{Q}_{N,N'}(v)(x,t)$, and $\tilde{g}(x,t) = \mathcal{Q}_{N,N'}(g)(x,t)$. In the matrix form, one can write

$$\mathbf{C}^T(x)\Big(Y - V + D^2 I_{\kappa} Y - G\Big)\mathbf{C}^T(t) = 0 \tag{30}$$

where $Y, V, G \in \mathbb{R}^{(N+1)\times(N'+1)}$, and

$$\begin{aligned}\tilde{y}(x,t) &= \mathbf{C}^T(x) Y \mathbf{C}^T(t),\\ \tilde{v}(x,t) &= \mathbf{C}^T(x) V \mathbf{C}^T(t),\\ \tilde{g}(x,t) &= \mathbf{C}^T(x) G \mathbf{C}^T(t).\end{aligned}$$

The suggested algorithm utilizes the collocation approach as its foundation. Note that the collocation method selects the solution that satisfies the equation at the collocation points. Equivalently, in this approach, the residual function $r(x,t)$ must approach zero at the collocation points. So, we obtain the residual function

$$r(x,t) = \tilde{y}(x,t) - \tilde{v}(x,t) + \mathcal{I}_0^{\kappa}\Big((-1)^{m/2}\mathcal{D}_x^m(\tilde{y}) - g\Big)(x,t), \tag{31}$$

and set the Lobatto nodes as collocation points.

Thanks to the matrix representation of FIO and derivative operator, and using the Lobatto grid, the following linear system can be obtained:

$$\tilde{y}(x_n^*, x_{n'}^*) - \tilde{v}(x_n^*,, x_{n'}^*) + \mathcal{I}_0^{\kappa}\Big((-1)^{m/2}\mathcal{D}_x^m(\tilde{y}) - g\Big)(x_n^*, x_{n'}^*) = 0, \quad 1 \le n' \le N', 1 \le n \le N. \tag{32}$$

Consequently, we have the linear system $AY = G$.

In considering the endpoints as collocation points in the Lobatto grid, only two conditions in the fourth-order FDWE remain unapplied. To use these two conditions, we make the following substitutions:

$$\begin{aligned}[AY]_{2,n'} &= [\mathbf{C}^T(0) D^{2^T} Y]_{n'}, [G]_{2,n'} = [\Psi_2]_{n'},\\ [AY]_{N'-1,n'} &= [\mathbf{C}^T(b) D^{2^T} Y]_{n'}, [G]_{N'-1,n'} = [\Psi_3]_{n'},\end{aligned}$$

where $\Psi_2, \Psi_3 \in \mathbb{R}^{N'+1}$ are obtained by

$$\psi_i \approx \mathcal{Q}_{N'}(\psi_i)(t) = \sum_{n'=1}^{N'+1} [\Psi]_{n'} C_{n'}(t) := \Psi_i^T \mathbf{C}(t), \quad \text{for } i = 2, 3.$$

By converting the matrices A, Y, and G into vectors $\bar{A}$, $\bar{Y}$, and $\bar{G}$, respectively, the following system can be obtained:

$$\bar{A}\bar{Y} = \bar{G}. \tag{33}$$

Solving this system using the Linsolve command of Matlab (version 2022) leads to finding the unknowns $\{y_{n,n'}\}$, for $1 \le n' \le N'+1, 1 \le n \le N+1$.

Convergence Analysis

The boundedness of the FIO in $L^p[0,b]$ was presented in [30] as follows:

$$\|\mathcal{I}_0^{\kappa}(y)\|_p \le K\|y\|_p, \qquad \text{with} \qquad K = \frac{b^{\kappa}}{\Gamma(\kappa+1)}. \tag{34}$$

Theorem 1. *Let $1 \le N \in \mathbb{N}$. The proposed method converges, and the error satisfies*

$$\|e_N\|_2 \le C\Lambda h^{\nu} N^{m-\nu} |y|_{\mathcal{H}^{\nu,N}}, \tag{35}$$

where C and Λ are constants, $\Lambda = \max\{|v|_{\mathcal{H}^{\nu,N}}, |g|_{\mathcal{H}^{\nu,N}}\}$, and $h = \max\{b, T\}$.

Proof. Subtracting Equation (29) from (27) leads to

$$e_N - (v - \tilde{v}) + \mathcal{I}_0^{\kappa}\Big((-1)^{m/2}\mathcal{D}_x^m(e_N) - (g - \tilde{g})\Big) = 0, \tag{36}$$

where $e_N = y - \tilde{y}$. Taking the norm of (36) and using the triangular inequality gives rise to

$$\|e_N\|_2 \leq \|v - \tilde{v}\|_2 + \|\mathcal{I}_0^{\kappa}(\mathcal{D}_x^m(e_N))\|_2 + \|\mathcal{I}_0^{\kappa}(g - \tilde{g})\|_2. \tag{37}$$

Motivated by Lemma 1 and in using $h = \max\{b, T\}$, the first norm on the left side can be bounded as follows:

$$\|v - \tilde{v}\|_2 \leq C_0 h^{\nu} N^{-\nu} |v|_{\mathcal{H}^{\nu,N}}, \tag{38}$$

where C_0 is a constant. Similarly, thanks to Lemma 1 and using Equation (34), we have

$$\|\mathcal{I}_0^{\kappa}(g - \tilde{g})\|_2 \leq K\|g - \tilde{g}\|_2 \leq C_1 h^{\nu} N^{-\nu} |g|_{\mathcal{H}^{\nu,N}}, \tag{39}$$

in which C_1 is a constant, and K is determined by (34). In addition to Lemma 1 and Equation (34), we need to use Theorem 8.6 [28] to find the bound of the second norm on the left side of Equation (37), viz.,

$$|\mathcal{I}_0^{\kappa}(\mathcal{D}_x^m(e_N))\|_2 \leq K\|\mathcal{D}_x^m(e_N)\|_2 \leq C_2 h^{\nu-m} N^{m-\nu} |y|_{\mathcal{H}^{\nu,N}}, \quad 0 \leq m \leq \nu \leq N+1. \tag{40}$$

Consequently, it is easy to confirm that

$$\|e_N\|_2 \leq C_0 h^{\nu} N^{-\nu} (\max\{|v|_{\mathcal{H}^{\nu,N}}, |g|_{\mathcal{H}^{\nu,N}}\}) + C_2 h^{\nu-m} N^{m-\nu} |y|_{\mathcal{H}^{\nu,N}}. \tag{41}$$

Considering $\Lambda = \max\{|v|_{\mathcal{H}^{\nu,N}}, |g|_{\mathcal{H}^{\nu,N}}\}$, one can obtain

$$\|e_N\|_2 \leq C\Lambda h^{\nu} N^{m-\nu} |y|_{\mathcal{H}^{\nu,N}}. \tag{42}$$

□

4. Numerical Experiments

To show the accuracy of the proposed scheme, we calculated the L^2-error and L^{∞}-error. These errors are given by the following equations:

$$L^2 - error = \left(\int_0^b \int_0^T |e_N|^2 dt dx \right)^{1/2},$$

and

$$L^{\infty} - error = \max_{x \in [0,b], t \in [0,T]} |e_N(x,t)|.$$

Example 1. *Consider the second-order FDWE as follows [32]:*

$$ {}_0^C\mathcal{D}_t^{\kappa}(y)(x,t) = \frac{x^2}{2} \mathcal{D}_x^2(y)(x,t), \qquad 0 \leq t \leq 1, 0 \leq x \leq 1, 1 < \kappa \leq 2,$$

with conditions

$$\begin{cases} y(0,t) = 0, & y(1,t) = \sinh(t), \\ y(x,0) = x, & \mathcal{D}_t(y)(x,0) = x^2. \end{cases}$$

The analytic solution for this equation is reported in [33], obtained by VIM, as

$$y(x,t) = x + x^2 t E_{\kappa,2}(t^{\kappa}),$$

in which $E_{\kappa,2}$ denotes the Mittag–Leffler function [30].

To demonstrate the presented method's ability to solve second-order FDWE, Table 1 is provided. As you can see, the method solves the problem with high accuracy. Table 2 tabulates the method's convergence. Increasing the value of N (number of bases) reduces error. Also, in this table, we report the CPU time. Figure 1 illustrates the approximate solution and corresponding absolute error, taking $\kappa = 2$ and $N = 10$. Figure 2 plots the

absolute errors obtained with different values of N at $t = 1$, where κ was selected as $15/8$. It is worth mentioning that when $\kappa \to 2$, we expected the solutions to tend toward the corresponding solution obtained with $\kappa = 2$. Figure 3 demonstrates this fact.

Table 1. The errors obtained using the presented method for Example 1.

κ	9/8	11/8	13/8	15/8	2
L^2-error	1.799×10^{-07}	8.693×10^{-08}	1.460×10^{-08}	1.828×10^{-09}	4.331×10^{-15}
L^∞-error	2.493×10^{-06}	2.438×10^{-06}	8.292×10^{-07}	1.730×10^{-07}	5.507×10^{-13}

Table 2. The L^2-error, taking $\kappa = 15/8$ for Example 1.

$t \setminus N$	6	8	10	12
0.1	7.189×10^{-07}	4.229×10^{-07}	4.617×10^{-08}	2.081×10^{-08}
0.2	3.153×10^{-06}	8.469×10^{-08}	7.727×10^{-08}	1.740×10^{-08}
0.3	1.440×10^{-06}	3.683×10^{-07}	6.500×10^{-08}	3.862×10^{-09}
0.4	1.455×10^{-06}	5.512×10^{-08}	1.610×10^{-08}	1.078×10^{-08}
0.5	2.465×10^{-06}	3.009×10^{-07}	5.719×10^{-08}	1.527×10^{-08}
0.6	1.150×10^{-06}	4.268×10^{-08}	1.175×10^{-08}	7.676×10^{-09}
0.7	8.585×10^{-07}	2.090×10^{-07}	3.415×10^{-08}	1.919×10^{-09}
0.8	1.443×10^{-06}	3.508×10^{-08}	2.764×10^{-08}	5.731×10^{-09}
0.9	8.747×10^{-08}	1.162×10^{-08}	1.016×10^{-08}	3.922×10^{-09}
1.0	2.740×10^{-08}	2.053×10^{-09}	2.269×10^{-10}	3.882×10^{-11}
CPU Time	0.281	1.422	7.266	19.484

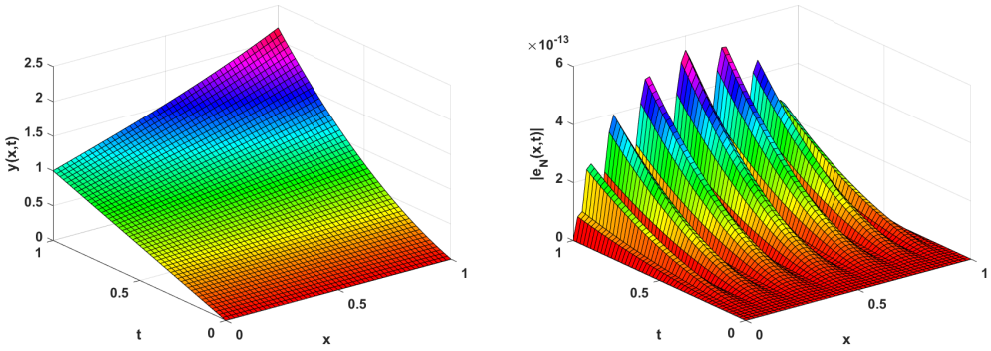


Figure 1. Plots of the approximate solution and corresponding error, where $\kappa = 2$ and $N = 10$ (Example 1).

Example 2. *The second example delves into the fourth-order FDWE [20]:*

$$ {}_0^C\mathcal{D}_t^\kappa(y)(x,t) + \mathcal{D}_x^4(y)(x,t) = e^x\left(\frac{\Gamma(\kappa+4)}{6}t^3 + t^{\kappa+3} + t\right), \qquad 0 < t \le 1, 0 < x \le 1, 1 < \kappa \le 2, $$

with

$$ y(x,0) = 0,\ \mathcal{D}_t(y)(x,0) = e^x, \quad \textit{(initial conditions)}, $$

and

$$ \begin{cases} y(0,t) = t^{\kappa+3} + t, & y(1,t) = e(t^{\kappa+3} + t), \\ \mathcal{D}_x^2(y)(0,t) = t^{\kappa+3} + t, & \mathcal{D}_x^2(y)(1,t) = e(t^{\kappa+3} + t), \quad \textit{(boundary conditions)}. \end{cases} $$

As reported in [20], the exact solution is denoted as $y(x,t) = e^x(t^{\kappa+3} + t)$.

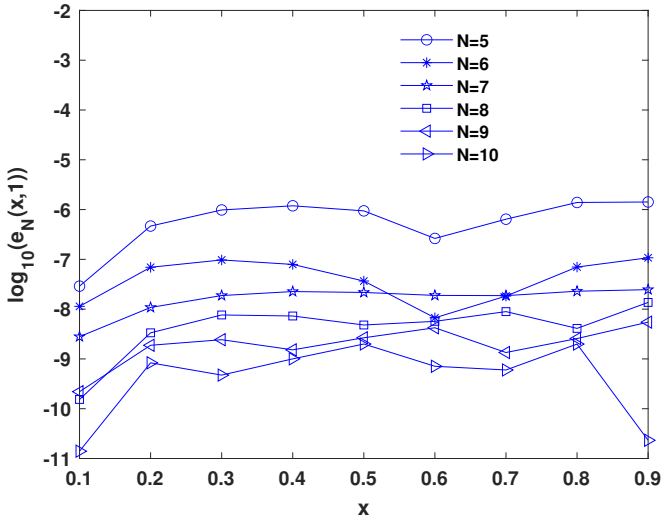


Figure 2. Errors reported at $t = 1$ for different values of N, taking $\kappa = 15/8$ for Example 1.

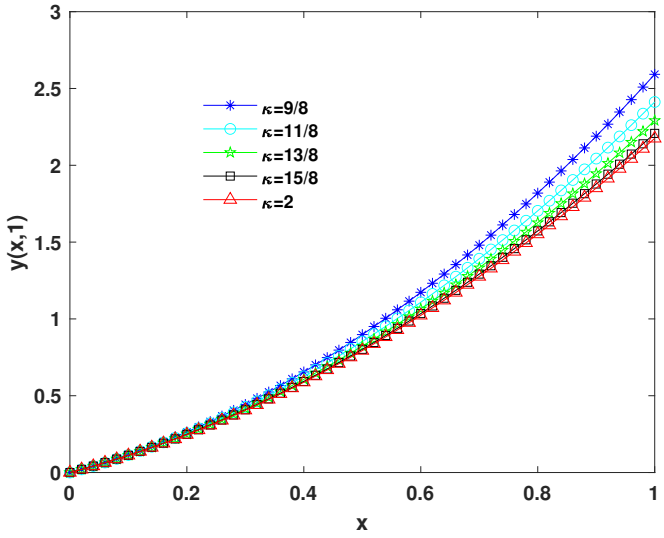


Figure 3. Approximate solution obtained for different values of κ at $t = 1$ for Example 1.

The capability of solving fourth-order FDWEs is showcased in Table 3 through the presented method. We tabulated Table 4 to confirm that the method is convergent. Increasing the number of bases (N) reduces error. Also, in this table, we report the CPU time. Table 5 is reported to compare the presented method with CCNFDM [20]. Our scheme gives better accuracy than CCNFDM, and it requires fewer calculations. Figure 4 illustrates the approximate solution and corresponding absolute error, taking $\kappa = 1.8$ and $N = 10$. We plot the absolute errors at $t = 1$, obtained for different values of N, in Figure 5.

Table 3. The errors obtained using the presented method for Example 2.

κ	1.2	1.3	1.5	1.7	1.8
L^2-error	1.563×10^{-08}	1.552×10^{-08}	1.528×10^{-08}	1.499×10^{-08}	1.485×10^{-08}
L^∞-error	2.532×10^{-07}	3.035×10^{-07}	2.889×10^{-07}	1.838×10^{-07}	1.193×10^{-07}

Table 4. The L^2-error, taking $\kappa = 1.5$ for Example 2.

$t \setminus N$	6	8	10	12
0.1	6.698×10^{-05}	8.771×10^{-07}	1.898×10^{-07}	3.667×10^{-09}
0.2	2.578×10^{-05}	6.383×10^{-07}	1.503×10^{-07}	2.720×10^{-09}
0.3	3.728×10^{-05}	1.890×10^{-06}	7.078×10^{-08}	3.717×10^{-09}
0.4	1.158×10^{-04}	4.502×10^{-07}	4.619×10^{-08}	1.043×10^{-08}
0.5	1.754×10^{-04}	9.287×10^{-07}	9.714×10^{-08}	1.201×10^{-08}
0.6	1.731×10^{-04}	7.166×10^{-07}	3.592×10^{-08}	7.929×10^{-09}
0.7	1.676×10^{-04}	1.923×10^{-06}	4.265×10^{-08}	2.120×10^{-09}
0.8	2.330×10^{-04}	7.509×10^{-07}	7.270×10^{-08}	1.126×10^{-09}
0.9	3.776×10^{-04}	1.104×10^{-06}	6.305×10^{-08}	1.028×10^{-09}
1.0	3.967×10^{-04}	1.173×10^{-06}	5.836×10^{-08}	7.207×10^{-09}
CPU Time	0.391	0.469	0.953	3.281

Table 5. A comparison between the presented scheme and CCNFDM [20] for Example 2.

	Our Scheme				**CCNFDM [20] with $h = 1/500$**		
	κ				κ		
$N = N'$	**1.3**	**1.5**	**1.8**	τ	**1.3**	**1.5**	**1.8**
6	5.859×10^{-04}	5.563×10^{-04}	5.604×10^{-04}	1/10	2.25×10^{-03}	6.21×10^{-03}	2.57×10^{-02}
8	2.254×10^{-06}	2.480×10^{-06}	2.060×10^{-06}	1/20	6.96×10^{-04}	2.21×10^{-03}	1.13×10^{-02}
10	3.035×10^{-07}	2.889×10^{-07}	1.193×10^{-07}	1/40	2.15×10^{-04}	7.88×10^{-04}	4.99×10^{-03}

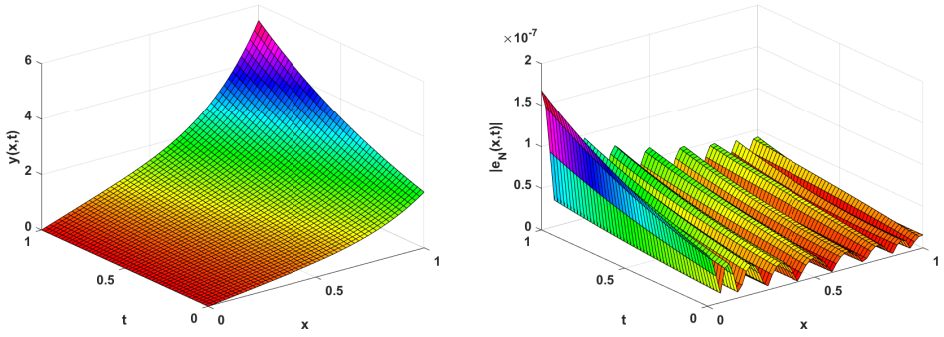


Figure 4. Plots of the approximate solution and corresponding error, where $\kappa = 1.8$ and $N = 10$ (Example 2).

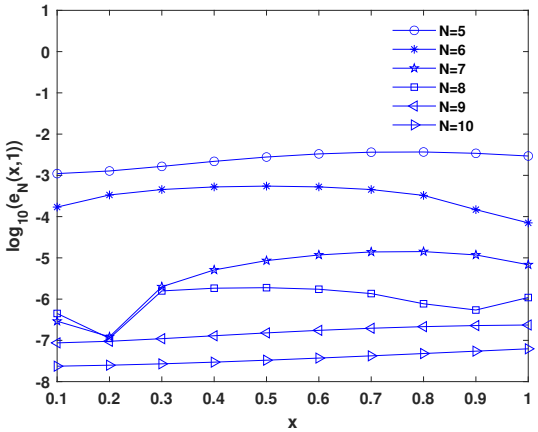


Figure 5. Errors reported at $t = 1$ for different values of N, taking $\kappa = 1.7$ for Example 2.

5. Conclusions

Considering the importance and application of the FDWE, mentioned in the introduction section, solving this type of equation can be extremely valuable. In this study, we introduce an innovative and precise algorithm for solving the given equation. The algorithm is derived by reducing the equation to the Volterra integral equation and solving this integral equation with the collocation method. To confirm the convergence of the method, we performed a rigorous analysis, and the numerical examples support our findings.

Based on the experimental observations, the following conclusions can be drawn:

- The presented algorithm demonstrates efficacy in solving the FDWE.
- Our algorithm demonstrates convergence when tackling these equations.
- The method presented offers reduced computational cost by avoiding integration to find coefficients.
- The proposed algorithm is simple to implement, and it also provides good accuracy.

Author Contributions: Conceptualization, H.B.J. and I.D.; methodology, H.B.J.; software, H.B.J. and I.D.; validation, H.B.J. and I.D.; formal analysis, H.B.J. and I.D.; investigation, H.B.J. and I.D.; writing—original draft preparation, H.B.J. and I.D.; writing—review and editing, H.B.J. and I.D.; visualization, H.B.J. and I.D.; supervision, H.B.J.; project administration, H.B.J. and I.D.; funding acquisition, H.B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This project was supported by Researchers Supporting Project number RSP2024R210, King Saud University, Riyadh, Saudi Arabia.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The writers state that they have no known personal relationships or competing financial interests that could have appeared to affect the work reported in this work.

References

1. Arif, M.; Ali, F.; Khan, I.; Nisar, K.S. A time fractional model with non-singular kernel the generalized couette flow of couple stress nanofluid. *IEEE Access* **2020**, *8*, 77378–77395. [CrossRef]
2. Chang, A.; Sun, H.; Zheng, C.; Bingqing, L.; Chengpeng, L.; Rui, M.; Yong, Z. A Time Fractional Convection-Diffusion Equation to Model Gas Transport through Heterogeneous Soil and Gas Reservoirs. *Physica A* **2018**, *502*, 356–369. [CrossRef]
3. Tenreiro, M.; Silva, M.F.; Barbosa, R.S.; Jesus, I.S.; Reis, C.M.; Marcos, M.G.; Galhano, A.F. Some Applications of Fractional Calculus in Engineering. *Math. Probl. Eng.* **2010**, *2010*, 639801. [CrossRef]
4. Mainardi, F. *Fractional Calculus and Waves in Linear Viscoelasticity*; Imperial College Press: London, UK, 2010.
5. Maji, S.; Natesan, S. Adaptive-grid technique for the numerical solution of a class of fractional boundary-value-problems. *Comput. Methods Differ. Equ.* **2010**, *12*, 338–349.
6. Lakestani, M.; Dehghan, M. The use of Chebyshev cardinal functions for the solution of a partial differential equation with an unknown time-dependent coefficient subject to an extra measurement. *J. Comput. Appl. Math.* **2010**, *235*, 669–678. [CrossRef]
7. Asadzadeh, M.; Saray, B.N. On a multiwavelet spectral element method for integral equation of a generalized Cauchy problem. *BIT Numer. Math.* **2022**, *62*, 383–1416. [CrossRef]
8. Shi, L.; Saray, B.N.; Soleymani, F. Sparse wavelet Galerkin method: Application for fractional Pantograph problem. *J. Comput. Appl. Math.* **2024**, *451*, 116081. [CrossRef]
9. Garrappa, R. On some explicit Adams multistep methods for fractional differential equations. *J. Comput. Appl. Math.* **2009**, *229*, 392–399. [CrossRef]
10. Jian, H.Y.; Huang, T.Z.; Zhao, X.; Zhao, Y.L. A fast second-order accurate difference schemes for time distributed-order and Riesz space fractional diffusion equations. *J. Appl. Anal. Comput.* **2019**, *9*, 1359–1392. [CrossRef]
11. Lin, Z.; Wang, D.; Qi, D.; Deng, L. A Petrov–Galerkin finite element-meshfree formulation for multi-dimensional fractional diffusion equations. *Comput. Mech.* **2020**, *66*, 323–350. [CrossRef]
12. Fix, G.J.; Roop, J.P. Least squares finite element solution of a fractional order two-point boundary value problem. *Comput. Math. Appl.* **2004**, *48*, 1017–1033. [CrossRef]
13. Zhao, Y.L.; Zhu, P.; Gu, X.M.; Zhao, X.; Jian, H.Y. An implicit integration factor method for a kind of spatial fractional diffusion equations. *J. Phys. Conf. Ser.* **2019**, *1324*, 012030. [CrossRef]
14. Daftardar-Gejji, V.; Jafari, A. Adomian decomposition: A tool for solving a system of fractional differential equations. *J. Math. Anal. Appl.* **2005**, *301*, 508–518. [CrossRef]

15. Benkerrouche, B.; Baleanu, D.; Souid, M.S.; Hakem, A. Boundary value problem for nonlinear fractional differential equations of variable order via Kuratowski MNC technique. *Adv. Differ. Equ.* **2021**, *2021*, 365. [CrossRef]
16. Nigmatullin, R.R. To the theoretical explanation of the universal response. *Phys. Status Solidi B Basic Res.* **1984**, *123*, 739–745. [CrossRef]
17. Oldham, K.B.; Spanier, J. *The Fractional Calculus*; Academic Press: New York, NY, USA, 1974.
18. Du, R.; Cao, W.R.; Sun, Z.Z. A compact difference scheme for the fractional diffusion-wave equation. *Appl. Math. Model.* **2010**, *34*, 2998–3007. [CrossRef]
19. Hu, X.; Zhang, L. A compact finite difference scheme for the fourth-order fractional diffusion-wave system. *Comput. Phys. Commun.* **2011**, *182*, 1645–1650. [CrossRef]
20. Hu, X.; Zhang, L. On finite difference methods for fourth-order fractional diffusion-wave and subdiffusion systems. *Appl. Math. Comput.* **2012**, *218*, 5019–5034. [CrossRef]
21. Darzi, R.; Mohammadzade, B.; Mousavi, S.; Beheshti, R. Sumudu transform method for solving fractional differential equations and fractional diffusionwave equation. *J. Math. Comput. Sci.* **2013**, *6*, 79–84. [CrossRef]
22. Cui, M.R. Convergence analysis of high-order compact alternating direction implicit schemes for the two-dimensional time fractional diffusion equation. *Numer. Algorithms* **2013**, *62*, 383–409. [CrossRef]
23. Chen, J.; Liu, F.; Anh, V.; Shen, S.; Liu, Q.; Liao, C. The analytical solution and numerical solution of the fractional diffusion-wave equation with damping. *Appl. Math. Comput.* **2012**, *219*, 1737–1748. [CrossRef]
24. Bhrawy, A.H.; Doha, E.H.; Baleanu, D.; Ezz-Eldien, S.S. A spectral tau algorithm based on Jacobi operational matrix for numerical solution of time fractional diffusion-wave equations. *J. Comput. Phys.* **2015**, *293*, 142–156. [CrossRef]
25. Afarideh, A.; Dastmalchi Saei, F.; Lakestani, M.; Saray, B.N. Pseudospectral method for solving fractional Sturm-Liouville problem using Chebyshev cardinal functions. *Phys. Scr.* **2021**, *96*, 125267. [CrossRef]
26. Shahriari, M.; Saray, B.N.; Mohammadalipour, B.; Saeidian, S. Pseudospectral method for solving the fractional one-dimensional Dirac operator using Chebyshev cardinal functions. *Phys. Scr.* **2023**, *98*, 055205. [CrossRef]
27. Boyd, J.P. *Chebyshev and Fourier Spectral Methods*, 2nd ed.; Dover Publications: Garden City, NY, USA, 2001.
28. Shen, J.; Tang, T.; Wang, L.L. *Spectral Methods: Algorithms. Analysis, Applications*; Springer: Berlin, Germany, 2011.
29. Afarideh, A.; Dastmalchi Saei, F.; Saray, B.N. Eigenvalue problem with fractional differential operator: Chebyshev cardinal spectral method. *J. Math. Model.* **2021**, *11*, 343–355.
30. Kilbas, A.; Srivastava, H.M.; Trujillo, J.J. *Theory and Applications of Fractional Differential Equations*; Elsevier B. V.: Amsterdam, The Netherlands, 2006.
31. Saray, B.N. Abel's integral operator: Sparse representation based on multiwavelets. *BIT Numer. Math.* **2021**, *61*, 587–606. [CrossRef]
32. Jafari, H.; Momani, S. Solving fractional diffusion and wave equations by modified homotopy perturbation method. *Phys. Lett. A* **2007**, *370*, 388–396. [CrossRef]
33. Momani, S.; Odibat, Z. Comparison between the homotopy perturbation method and the variational iteration method for linear fractional partial differential equations. *Comput. Math. Appl.* **2007**, *54*, 910–919. [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.