fractal and fractional

MDPI

Article

New Generalized Jacobi Galerkin Operational Matrices of Derivatives: An Algorithm for Solving Multi-Term Variable-Order Time-Fractional Diffusion-Wave Equations

Hany Mostafa Ahmed

Department of Mathematics, Faculty of Technology and Education, Helwan University, Cairo 11281, Egypt; hanyahmed@techedu.helwan.edu.eg

Abstract: The current study discusses a novel approach for numerically solving MTVO-TFDWEs under various conditions, such as IBCs and DBCs. It uses a class of GSJPs that satisfy the given conditions (IBCs or DBCs). One of the important parts of our method is establishing OMs for Ods and VOFDs of GSJPs. The second part is using the SCM by utilizing these OMs. This algorithm enables the extraction of precision and efficacy in numerical solutions. We provide theoretical assurances of the treatment's efficacy by validating its convergent and error investigations. Four examples are offered to clarify the approach's practicability and precision; in each one, the IBCs and DBCs are considered. The findings are compared to those of preceding studies, verifying that our treatment is more effective and precise than that of its competitors.

Keywords: Jacobi polynomials; fractional differential equations with variable order; collocation method; initial boundary conditions; Dirichlet boundary conditions

MSC: 42C05; 65L60; 34B05

Citation: Ahmed, H.M. New Generalized Jacobi Galerkin Operational Matrices of Derivatives: An Algorithm for Solving Multi-Term Variable-Order Time-Fractional Diffusion-Wave Equations. *Fractal Fract.* **2024**, *8*, 68. https://doi.org/10.3390/fractalfract8010068

Academic Editor: Riccardo Caponetto

Received: 14 December 2023
Revised: 11 January 2024
Accepted: 15 January 2024
Published: 18 January 2024

1. Introduction

Fractional calculus has emerged as a subject of great interest among researchers from diverse fields in recent decades. The universal perspective that fractional operators offer for comprehending system dynamics is what primarily motivates this interest. Fractional derivatives offer a more precise and comprehensive description of various physical phenomena compared to traditional integer-order derivatives [1–4]. Consequently, there exists a wide range of definitions for fractional differentiation in the literature (for further details, refer to [5–7]). A list of abbreviations used in the paper is given in Abbreviations Section.

Researchers have made worthy progress in the field of fractional calculus, emphasizing the extension of the framework to encompass VOFDs. This extension facilitates an understanding of diverse dynamic systems. The authors in [8] conducted a specific study in which they investigated the characteristics of VOFD operators [9,10].

VOFC is a strong framework that obtains the nonlocal properties of different systems very well [11–18]. For instance, in study [19], the utilization of VOFD operators enabled the modeling of the microscopic structure of materials. As shown in an additional publication [20], continuum elasticity also made use of the Riesz–Caputo fractional derivative of space-dependent order. In [21], the authors recommend using collocation and tau spectral techniques to solve the time-fractional heat equation numerically. Studies [22,23] specifically focused on understanding and analyzing how the fractional order, which characterizes the system's dynamics, evolves over time and influences its viscoelastic properties. By examining the time-dependent changes in the fractional order, valuable insights were gained into the complex behavior of such systems. Overall, VOFC has proven to be a valuable tool in engineering mechanics, providing a versatile approach to accurately describe and analyze a wide range of dynamic systems.

The pursuit of analytical solutions for FDEs poses a significant challenge, often necessitating reliance on numerical approximations. As a result, various numerical techniques have addressed the complexities associated with FDEs, enabling researchers to effectively tackle problems that would otherwise be challenging to solve analytically. Previous research has witnessed the utilization of a variety of approaches to create numerical solutions for FDEs through the use of both orthogonal and non-orthogonal polynomials. There are a number of papers [24–30] that talk about different OMs of JPs and some of their special cases. These papers use different spectral methods by utilizing these OMs to solve different kinds of DEs and FDEs numerically, subject to different kinds of IBCs. Additionally, researchers in [31–36] follow the same methodology to solve VOFDEs numerically. Additionally, in [37,38], Bernstein polynomials were employed to approximate solutions for FDEs. Another notable contribution in [39] proposed a numerical scheme based on Fourier analysis for solving FDEs. Furthermore, finite difference approximations were discussed in [40] as a means to construct numerical schemes for FDEs. In [41], the author uses a class of modified JPs to introduce a novel method for numerically solving MTVO-FDEs with initial conditions.

Many researchers [42–44] have shown that JPs have properties that make them very useful for solving different kinds of DEs, especially when spectral methods are used. Some of these properties are that they are orthogonal, have exponential accuracy, and have two parameters that allow us to shape approximate solutions in different ways. These inherent properties of JPs make them highly suitable for effectively solving a wide range of diverse problems encountered in various fields of study.

It is worth noting that a significant class of fractional partial differential equations that has received considerable attention in recent years is the TFDWE. This equation arises from the classical diffusion-wave equation by replacing the second-order time derivative term with a fractional derivative of order $1 < \alpha < 2$ [45]. The TFDWE describes important physical effects seen in many different types of systems, such as colloidal, amorphous, glassy, and porous materials, as well as dielectrics and semiconductors, comb structures, polymers, random and disordered media, biological systems, and geophysical and geological processes (see [46] and references therein).

Furthermore, it is worth mentioning that the TFDWE serves as an accurate model for many universal electromagnetic, acoustic, and mechanical responses [47,48]. A single TFDWE may not be able to fully describe the underlying processes in some real-life situations. This is why an MTFDWE was created, as shown in [46,49]. This MTFDWE formulation offers a more comprehensive representation of complex systems and their dynamic behavior, allowing for more accurate modeling of the underlying processes.

We considered the general form of MTVO-TFDE in our investigation as follows:

$$\left({}_0^c\mathcal{D}_t^{\nu(x,t)} + \sum_{j=1}^{m} \varrho_j \, {}_0^c\mathcal{D}_t^{\nu_j(x,t)}\right) y(x,t) + \varrho \, y_t(x,t) = \kappa \, y_{xx}(x,t) + g(x,t), (x,t) \in [0,\ell] \times [0,\mathcal{T}], \tag{1}$$

subject to the IBCs:

$$y(x,0) = f_1(x),\ y_t(x,0) = f_2(x),\ y(0,t) = f_3(t),\ y(\ell,t) = f_4(t), \tag{2}$$

or the DBCs:

$$y(x,0) = f_1(x),\ y(x,\mathcal{T}) = f_2(x),\ y(0,t) = f_3(t),\ y(\ell,t) = f_4(t), \tag{3}$$

where ϱ_j $(j = 1,2,\ldots,m,\ m \in \mathbb{N})$, $\varrho \geq 0$ and $\kappa > 0$ are constants; $1 < \nu_1(x,t) < \nu_2(x,t) < \ldots < \nu_m(x,t) < \nu(x,t) \leq 2$ holds; $f_0(x)$, $f_1(x)$, $f_2(x)$, $f_3(t)$, $f_4(t)$ and $g(x,t)$ are given functions; and ${}_0^c\mathcal{D}_t^{\nu(x,t)}$ *and* ${}_0^c\mathcal{D}_t^{\nu_j(x,t)}$ $(i = 1,2,\ldots,m)$ are the VOFDs defined in the Caputo sense, as given in Section 2. The significance and difficulty of proving the existence, unicity, and dependency of parameters should be emphasized. The significance of determination

of ν and ν_j in (1) is evident in theory and practice due to the strong relationship between them and the heterogeneity and associated physical features of media [50]. For instance, in the situation of a single term case, Cheng et al. [51] first demonstrated the uniqueness of ν using the boundary condition data that were provided.

We present a new Galerkin OM for Ods and new OMs for VOFDs of GSJPs in the sense of Caputo. This was made in order to find a new way to solve the problem shown by (1) and the conditions (2) or (3). These operational matrices are specifically tailored for the basis vectors of GSJPs. Leveraging these OMs, we have established a powerful tool that enables the accurate computation of numerical solutions using the SCM to solve a wide range of MTVO-TFDEs. This novel method opens up new ways to solve this type of FDE numerically and more effectively.

To summarize, the main article's contributions are as follows:

(i) We introduce two classes of GSJPs to satisfy the given IBCs and DBCs (see Section 3.2).
(ii) We establish Galerkin OMs for the Ods and for VOFDs of the introduced GSJPs in the sense of Caputo (see Sections 4 and 5).
(iii) We address the presented MTVO-TFDE using the proposed GSJPs and their constructed OMs in conjunction with the SCM (see Section 6).
(iv) We present a study of convergence and error analysis for the numerical solution obtained through the proposed scheme (see Section 7).

The paper has the following outline: Section 2 provides comprehensive coverage of the needed concepts of VOFC. Section 3 highlights the necessary attributes of JPs and GSJPs. Sections 4 and 5 emphasize the development of new Galerkin OMs for Ods and VOFDs of GSJPs. These OMs are intended to address Equation (1) when IBCs (2) or DBCs (3) are considered. Section 6 explores the selection of newly generated OMs in the SCM to address the aforementioned issue. Section 7 analyzes convergence and error estimation. Section 8 includes four examples to clarify the approach's practicability and precision, and the findings are compared to those of preceding studies, verifying that our treatment is more effective and precise than its competitors. Finally, Section 9 summarizes the key outcomes, implications derived from our investigation, and the scope of future work.

2. Basic Definition of Caputo VOFDs

This section serves to introduce and discuss important definitions and necessary assets that lay the groundwork for the development of our proposed technique. These tools are crucial building blocks that provide the necessary foundation for effectively handling the MTVO-TFDE under consideration.

Definition 1 ([39,45]). *Suppose $h(x,t)$ is a twice differentiable function. The Caputo derivative operator with the variable-order $\nu(x,t)$ is defined as:*

$$ {}_0^C\mathcal{D}_t^{\nu(x,t)} h(x,t) = \begin{cases} \frac{1}{\Gamma(2-\nu(x,t))} \int_0^t (t-\tau)^{1-\nu(x,t)} \frac{\partial^2 h(x,\tau)}{\partial \tau^2} \, d\tau, & 1 < \nu(x,t) < 2, \\ h_{tt}(x,t), & \nu(x,t) = 2. \end{cases} \tag{4} $$

The Caputo VOFD exhibits the characteristics:

$$ {}_0^C\mathcal{D}_t^{\nu(x,t)} (\lambda_1 h_1(t) + \lambda_2 h_2(t)) = \lambda_1 \, {}_0^C\mathcal{D}_t^{\nu(x,t)} h_1(t) + \lambda_2 \, {}_0^C\mathcal{D}_t^{\nu(x,t)} h_2(t), \tag{5} $$

$$ {}_0^C\mathcal{D}_t^{\nu(x,t)} (C) = 0, \qquad \text{(C is a constant)}, \tag{6a} $$

$$ {}_0^C\mathcal{D}_t^{\nu(x,t)} t^k = \begin{cases} 0, & k = 0, 1, \\ \frac{\Gamma(k+1)}{\Gamma(k+1-\nu(x,t))} t^{k-\nu(x,t)}, & k = 2, 3, \ldots. \end{cases} \tag{6b} $$

Remark 1. *The reader who is interested may find several definitions and additional attributes of VOFDs in [4] (pp. 35–42).*

3. An Overview of the Shifted JPs and Their Generalized Ones

Introducing the fundamental features of the JPs and their shifting form is the main objective of this section. Additionally, a group of GSJPs is introduced.

3.1. An Overview of the Shifted JPs

The orthogonal JPs, $\mathcal{J}_n^{(\hat{\alpha},\hat{\beta})}(x)$, $\hat{\alpha}$, $\hat{\beta} > -1$, satisfy the following relationship [52]:

$$\int_{-1}^{1} w^{\hat{\alpha},\hat{\beta}}(x)\mathcal{J}_n^{(\hat{\alpha},\hat{\beta})}(x)\,\mathcal{J}_m^{(\hat{\alpha},\hat{\beta})}(x)\,dx = \begin{cases} 0, & m \neq n, \\ h_n^{(\hat{\alpha},\hat{\beta})}, & m = n, \end{cases}$$

where $w^{\hat{\alpha},\hat{\beta}}(x) = (1-x)^{\hat{\alpha}}(1+x)^{\hat{\beta}}$ and $h_n^{(\hat{\alpha},\hat{\beta})} = \frac{2^{\lambda}\Gamma(n+\hat{\alpha}+1)\Gamma(n+\hat{\beta}+1)}{n!(2n+\lambda)\Gamma(n+\lambda)}$, $\lambda = \hat{\alpha}+\hat{\beta}+1$.

The shifted JPs, denoted as $\mathcal{J}_{\ell,n}^{(\hat{\alpha},\hat{\beta})}(t) = \mathcal{J}_n^{(\hat{\alpha},\hat{\beta})}(2t/\ell - 1)$, are in accordance with:

$$\int_{0}^{\ell} w_{\ell}^{\hat{\alpha},\hat{\beta}}(t)\mathcal{J}_{\ell,n}^{(\hat{\alpha},\hat{\beta})}(t)\,\mathcal{J}_{\ell,m}^{(\hat{\alpha},\hat{\beta})}(t)\,dt = \begin{cases} 0, & m \neq n, \\ \left(\frac{\ell}{2}\right)^{\lambda} h_n^{(\hat{\alpha},\hat{\beta})}, & m = n, \end{cases}$$

where $w_{\ell}^{\hat{\alpha},\hat{\beta}}(t) = (\ell - t)^{\hat{\alpha}}\, t^{\hat{\beta}}$.

The fundamental expansions that will be used in this paper are [53] (Section 11.3.4):

1. The power form representations of $\mathcal{J}_{\ell,n}^{(\hat{\alpha},\hat{\beta})}(t)$ are as follows:

$$\mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) = \sum_{k=0}^{i} c_k^{(i)}\, t^k = \sum_{k=0}^{i} \bar{c}_k^{(i)}\, (\ell - t)^k, \tag{7}$$

where

$$c_k^{(i)} = \frac{(-1)^{i-k}\Gamma(i+\hat{\beta}+1)\Gamma(i+k+\lambda)}{\ell^k\, k!(i-k)!\Gamma(k+\hat{\beta}+1)\Gamma(i+\lambda)} \text{ and } \bar{c}_k^{(i)} = \frac{(-1)^k(\hat{\alpha}+1)_i(\lambda+i)_k}{\ell^k\, k!(i-k)!(\hat{\alpha}+1)_k}. \tag{8}$$

2. Alternatively, the expressions for t^k and $(\ell - t)^k$ in relation to $\mathcal{J}_{\ell,r}^{(\hat{\alpha},\hat{\beta})}(t)$ have the forms:

$$t^k = \sum_{r=0}^{k} b_r^{(k)}\, \mathcal{J}_{\ell,r}^{(\hat{\alpha},\hat{\beta})}(t), \text{ and } (\ell - t)^k = \sum_{r=0}^{k} \bar{b}_r^{(k)}\, \mathcal{J}_{\ell,r}^{(\hat{\alpha},\hat{\beta})}(t), \tag{9}$$

where

$$b_r^{(k)} = \frac{\ell^k\, k!\,(\lambda+2r)\Gamma(k+\hat{\beta}+1)\Gamma(r+\lambda)}{(k-r)!\,\Gamma(r+\hat{\beta}+1)\Gamma(k+r+\lambda+1)} \text{ and } \bar{b}_r^{(k)} = \frac{(2j+\lambda)\ell^k(-k)_j(\lambda+1)_{j-1}(\hat{\alpha}+1)_k}{(\hat{\alpha}+1)_j(\lambda+1)_k(k+\lambda+1)_j}. \tag{10}$$

3.2. Introducing GSJPs

In this section, it is helpful to talk about two polynomials, $\left\{\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t)\right\}_{j\geq 0}$ and $\left\{\psi_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t)\right\}_{j\geq 0}$. These are needed to satisfy the homogeneous form of the given IBCs (2) and DBCs (3):

$$\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t) = t^2\mathcal{J}_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t), \tag{11}$$

$$\psi_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t) = t(\ell - t)\mathcal{J}_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t). \tag{12}$$

Subsequently, these polynomials satisfy the orthogonality relations as follows:

$$\int_0^{\mathcal{T}} \frac{w_{\mathcal{T}}^{\hat{\alpha},\hat{\beta}}(t)}{t^4} \phi_{\mathcal{T},i}^{(\hat{\alpha},\hat{\beta})}(t)\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t)dt = \begin{cases} 0, & i \neq j, \\ \left(\frac{\mathcal{T}}{2}\right)^{\lambda} h_i^{(\hat{\alpha},\hat{\beta})}, & i = j, \end{cases} \tag{13}$$

$$\int_0^{\ell} \frac{w_{\ell}^{\hat{\alpha},\hat{\beta}}(t)}{t^2(\ell-t)^2} \psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t)\psi_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t)dt = \begin{cases} 0, & i \neq j, \\ \left(\frac{\ell}{2}\right)^{\lambda} h_i^{(\hat{\alpha},\hat{\beta})}, & i = j, \end{cases} \tag{14}$$

where $w_{\rho}^{\hat{\alpha},\hat{\beta}}(t) = (\rho - t)^{\hat{\alpha}}\, t^{\hat{\beta}}$.

4. Two OMs for Ods and VOFDs of $\boldsymbol{\phi}_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t)$

In this part, we introduce two OMs related to ODEs and VOFDs of $\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t)$. To facilitate this, we commence with Theorem 1:

Theorem 1 ([41]). *$\mathcal{D}(t^n\, \mathcal{J}_{\mathcal{T},i}^{(\hat{\alpha},\hat{\beta})}(t))$ for all $i \geq 0$ can be computed as follows:*

$$\mathcal{D}(t^n\, \mathcal{J}_{\mathcal{T},i}^{(\hat{\alpha},\hat{\beta})}(t)) = \sum_{j=0}^{i-1} \theta_{i,j}^{\hat{\alpha},\hat{\beta}}(n,\mathcal{T})(t^n\, \mathcal{J}_{\mathcal{T},i}^{(\hat{\alpha},\hat{\beta})}(t)) + \epsilon_{n,i}(t), \tag{15}$$

where $\epsilon_{n,i}(t) = \frac{n}{i!}(-1)^i\, (\hat{\beta}+1)_i\, t^{n-1}$, and

$$\theta_{i,j}^{\hat{\alpha},\hat{\beta}}(n,\mathcal{T}) = C_{i,j}^{\hat{\alpha},\hat{\beta}} \sum_{r=0}^{i-j-1} \frac{(-1)^r (j+n+r+1)(i+j+\lambda+1)_r}{r!(j+r+1)(j+r+\hat{\beta}+1)\Gamma(i-j-r)\Gamma(2j+r+\lambda+1)}, \tag{16}$$

where

$$C_{i,j}^{\hat{\alpha},\hat{\beta}} = \frac{(-1)^{i+j-1}(\lambda+i)(\hat{\beta}+1)_i(\lambda+2j)\Gamma(j+\lambda)(i+\lambda+1)_j}{\mathcal{T}(\hat{\beta}+1)_j}.$$

Then, the two desired OMs of

$$\boldsymbol{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t) = \left[\phi_{\mathcal{T},0}^{(\hat{\alpha},\hat{\beta})}(t), \phi_{\mathcal{T},1}^{(\hat{\alpha},\hat{\beta})}(t), \ldots, \phi_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t)\right]^T \tag{17}$$

can be computed as follows:

Corollary 1 ([41]). *The general derivative of $\boldsymbol{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t)$ has the form:*

$$\frac{d^m \boldsymbol{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t)}{dt^m} = G^m \boldsymbol{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t) + \boldsymbol{\eta}_N^{(m)}(t), \tag{18}$$

with $\boldsymbol{\eta}_N^{(m)}(t) = \sum_{k=0}^{m-1} G^k\, \boldsymbol{\epsilon}_N^{(m-k-1)}(t)$, where $\boldsymbol{\epsilon}_N(t) = [\epsilon_{2,0}(t), \epsilon_{2,1}(t), \ldots, \epsilon_{2,N}(t)]^T$ and $G = \left(g_{i,j}\right)_{0 \leq i,j \leq N}$,

$$g_{i,j} = \begin{cases} \theta_{i,j}^{\hat{\alpha},\hat{\beta}}(2,\mathcal{T}), & i > j, \\ 0, & \textit{otherwise}. \end{cases}$$

Theorem 2. *$D^{\nu(x,t)}\phi_{\mathcal{T},i}^{(\hat{\alpha},\hat{\beta})}(t)$ for all $i \geq 0$, has the form:*

$$ {}_0^c\mathcal{D}_t^{\nu(x,t)}\phi_{\mathcal{T},i}^{(\hat{\alpha},\hat{\beta})}(t) = t^{-\nu(x,t)}\sum_{j=0}^{i}\Theta_{i,j}(\nu(x,t))\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t), \tag{19}$$

which leads to:

$$ {}_0^c\mathcal{D}_t^{\nu(x,t)}\mathbf{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t) = t^{-\nu(x,t)}\mathbf{D}^{(\nu(x,t))}\mathbf{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t), \tag{20}$$

where $\boldsymbol{D}^{(\nu(x,t))} = (d_{i,j}(\nu(x,t)))$ is a matrix of order $(N+1)\times(N+1)$, which can be expressed explicitly as:

$$\begin{pmatrix} \Theta_{0,0}(\nu(x,t)) & 0 & \cdots & \cdots & \cdots & 0 \\ \Theta_{1,0}(\nu(x,t)) & \Theta_{1,1}(\nu(x,t)) & 0 & \cdots & \cdots & 0 \\ \vdots & & & & & \vdots \\ \Theta_{i,0}(\nu(x,t)) & \cdots & \Theta_{i,i}(\nu(x,t)) & 0 & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & \ddots & 0 \\ \Theta_{N,0}(\nu(x,t)) & \cdots & \cdots & \cdots & \cdots & \Theta_{N,N}(\nu(x,t)) \end{pmatrix}, \tag{21}$$

where

$$ d_{i,j}(\nu(x,t)) = \begin{cases} \Theta_{i,j}(\nu(x,t)), & i \geq j, \\ 0, & \text{otherwise}, \end{cases} \tag{22}$$

and

$$ \Theta_{i,j}(\nu(x,t)) = \frac{(-1)^{i-j}(j+2)!\Gamma(i+\hat{\beta}+1)\Gamma(j+\lambda)\Gamma(i+j+\lambda)}{(i-j)!\Gamma(j+\hat{\beta}+1)\Gamma(2j+\lambda)\Gamma(i+\lambda)\Gamma(j-\nu(x,t)+3)} \times {}_3F_2\left(\begin{matrix} j-i, j+3, i+j+\lambda \\ 2j+\lambda+1, j-\nu(x,t)+3 \end{matrix} ; 1 \right). \tag{23}$$

Proof. By considering (7) and applying (6b), we obtain:

$$ {}_0^c\mathcal{D}_t^{\nu(x,t)}\phi_{\mathcal{T},i}^{(\hat{\alpha},\hat{\beta})}(t) = t^{2-\nu(x,t)}\sum_{k=0}^{i} c_k^{(i)}\frac{\Gamma(k+3)}{\Gamma(k-\nu(x,t)+3)}t^k. \tag{24}$$

By utilizing (9), expanding and collecting similar terms, and using useful computational formulae of the Pochhammer symbol and the Gamma function (see [54] (p. 758)), it is possible to convert (24) to (19), which is represented as follows:

$$ {}_0^c\mathcal{D}_t^{\nu(x,t)}\phi_{\mathcal{T},i}^{(\hat{\alpha},\hat{\beta})}(t) = t^{-\nu(x,t)}[\Theta_{i,0}(\nu(x,t)),\Theta_{i,1}(\nu(x,t)),\cdots,\Theta_{i,i}(\nu(x,t)),0,\cdots,0]\mathbf{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t), \tag{25}$$

and this representation enables us to declare that (20) is proved and that the theorem is completely proved. □

As an application of Theorem 2, for $N = 4$, $\hat{\alpha} = -\hat{\beta} = 1/2$, and $\nu(x,t) = x\,t$, the OM $\mathbf{D}^{(\nu(x,t))}$ has the form:

$$\mathbf{D}^{(\nu(x,t))} = \begin{pmatrix} \frac{2}{\Gamma(3-tx)} & 0 & 0 & 0 & 0 \\ \frac{tx}{\Gamma(4-tx)} & \frac{6}{\Gamma(4-tx)} & 0 & 0 & 0 \\ \frac{3tx(tx+2)}{4\Gamma(5-tx)} & \frac{27tx}{2\Gamma(5-tx)} & \frac{24}{\Gamma(5-tx)} & 0 & 0 \\ \frac{5tx(tx+1)(tx+5)}{8\Gamma(6-tx)} & \frac{45tx(tx+1)}{2\Gamma(6-tx)} & \frac{100tx}{\Gamma(6-tx)} & \frac{120}{\Gamma(6-tx)} & 0 \\ \frac{35tx(tx+1)(tx+2)(tx+9)}{64\Gamma(7-tx)} & \frac{525tx(tx+1)(tx+2)}{16\Gamma(7-tx)} & \frac{175tx(3tx+2)}{2\Gamma(7-tx)} & \frac{735tx}{\Gamma(7-tx)} & \frac{720}{\Gamma(7-tx)} \end{pmatrix}_{5\times 5}. \tag{26}$$

5. Two OMs for Ods and VOFDs of $\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t)$

In this part, we establish two OMs for Ods and VOFDs of $\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t)$. In order to achieve this, we commence by proving the subsequent lemma:

Lemma 1. *The polynomials* $t\,\mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t)$, $i \geq 0$ *have the following representation:*

$$t\,\mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) = -\sum_{j=0}^{i-1} \Upsilon_{i,j}^{\hat{\alpha},\hat{\beta}}(\ell)\psi_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t) + \frac{(\hat{\alpha}+1)_i}{i!}\,t, \tag{27}$$

where

$$\Upsilon_{i,j}^{\hat{\alpha},\hat{\beta}} = \frac{\Gamma(i+\hat{\alpha}+1)\Gamma(j+\lambda)\Gamma(i+j+\lambda+1)}{\ell\,(j+1)\Gamma(i-j)\Gamma(j+\hat{\alpha}+2)\Gamma(i+\lambda)\Gamma(2j+\lambda)}\;{}_4F_3\left(\begin{matrix}-i+j+1, j+1, \hat{\alpha}+j+1, \lambda+i+j+1\\ j+2, \hat{\alpha}+j+2, \lambda+2j+1\end{matrix};1\right). \tag{28}$$

Proof. We have

$$\mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) = (\ell-t)\left(\frac{\mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) - \mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(\ell)}{\ell-t}\right) + \mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(\ell). \tag{29}$$

In view of (7) and (9), we obtain:

$$\frac{\mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) - \mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(\ell)}{\ell-t} = \sum_{k=0}^{i-1} \bar{c}_{k+1}^{(i)}\,(\ell-t)^k = \sum_{k=0}^{i-1} \bar{c}_{k+1}^{(i)}\left(\sum_{r=0}^{k} \bar{b}_r^{(k)}\,\mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t)\right), \tag{30}$$

consequently, through expansion, the accumulation of similar terms, and using useful computational formulae of the Pochhammer symbol and the Gamma function (see [54] (p. 758)), it becomes evident that:

$$\frac{\mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) - \mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(\ell)}{\ell-t} = -\sum_{j=0}^{i-1} \Upsilon_{i,j}^{\hat{\alpha},\hat{\beta}}(\ell)\,\mathcal{J}_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t). \tag{31}$$

Substitution of (31) and $\mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(\ell) = \frac{(\hat{\alpha}+1)_i}{i!}$ into (29) leads to (27), and then the lemma is proved completely. □

Theorem 3. $\mathcal{D}\,\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t)$, *for all* $i \geq 0$, *has the form:*

$$\mathcal{D}\,\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) = \sum_{j=0}^{i-1} \Omega_{i,j}^{\hat{\alpha},\hat{\beta}}(\ell)\,\psi_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t) + \vartheta_{\ell,i}(t), \tag{32}$$

where $\vartheta_{\ell,i}(t) = \frac{1}{i!}((-1)^i(\ell-t)(\hat{\beta}+1)_i - t(\hat{\alpha}+1)_i)$, *and*

$$\Omega_{i,j}^{\hat{\alpha},\hat{\beta}}(\ell) = \theta_{i,j}^{\hat{\alpha},\hat{\beta}}(1,\ell) + \Upsilon_{i,j}^{\hat{\alpha},\hat{\beta}}(\ell), \tag{33}$$

Proof. We have

$$\mathcal{D}\,\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) = (\ell-t)\,\mathcal{D}\,(t\,\mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t)) - t\,\mathcal{J}_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t). \tag{34}$$

Using Theorem 1 and Lemma 1 leads to (32), and the proof of the theorem is complete. □

Then, the desired two OMs of

$$\mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(t) = \left[\psi_{\ell,0}^{(\hat{\alpha},\hat{\beta})}(t), \psi_{\ell,1}^{(\hat{\alpha},\hat{\beta})}(t), \cdots, \psi_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(t)\right]^T \tag{35}$$

can be computed as follows:

Corollary 2. *The general derivative of* $\mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(t)$ *is expressed as:*

$$\frac{d^m \mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(t)}{dt^m} = H^m\, \mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(t) + \boldsymbol{\xi}_{\ell,N}^{(m)}(t), \tag{36}$$

with $\boldsymbol{\xi}_{\ell,N}^{(m)}(t) = \sum_{k=0}^{m-1} H^k\, \boldsymbol{\vartheta}_{\ell,N}^{(m-k-1)}(t)$, *where* $\boldsymbol{\vartheta}_{\ell,N}(t) = [\vartheta_{\ell,0}(t), \vartheta_{\ell,1}(t), \ldots, \vartheta_{\ell,N}(t)]^T$ *and* $H = \left(h_{i,j}\right)_{0\le i,j\le N'}$

$$h_{i,j} = \begin{cases} \Omega_{i,j}^{\hat{\alpha},\hat{\beta}}(\ell), & i > j, \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 4. ${}_0^c\mathcal{D}_t^{\nu(x,t)} \psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t)$ *for all* $i \ge 0$, *has the expression*

$${}_0^c\mathcal{D}_t^{\nu(x,t)} \psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) = t^{-\nu(x,t)} \left(\sum_{j=0}^{i} \Lambda_{i,j}(\nu(x,t)) \psi_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t) + \varepsilon_{\ell,i}(x,t) \right), \tag{37}$$

which leads to

$${}_0^c\mathcal{D}_t^{\nu(x,t)} \mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(t) = t^{-\nu(x,t)} \left(\hat{\mathbf{D}}^{(\nu(x,t))} \mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(t) + \boldsymbol{\varepsilon}_{\ell,N}(x,t) \right), \tag{38}$$

where $\boldsymbol{\varepsilon}_{\ell,N}(x,t) = [\varepsilon_{\ell,0}(x,t), \varepsilon_{\ell,1}(x,t), \cdots, \varepsilon_{\ell,N}(x,t)]^T$ *and* $\hat{\boldsymbol{D}}^{(\nu(x,t))} = (\hat{d}_{i,j}(\nu(x,t)))$ *is a matrix of order* $(N+1)\times(N+1)$, *which can be expressed explicitly as:*

$$\begin{pmatrix} \Lambda_{0,0}(\nu(x,t)) & 0 & \cdots & \cdots & \cdots & 0 \\ \Lambda_{1,0}(\nu(x,t)) & \Lambda_{1,1}(\nu(x,t)) & 0 & \cdots & \cdots & 0 \\ \vdots & & & & & \vdots \\ \Lambda_{i,0}(\nu(x,t)) & \cdots & \Lambda_{i,i}(\nu(x,t)) & 0 & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & \ddots & 0 \\ \Lambda_{N,0}(\nu(x,t)) & \cdots & \cdots & \cdots & \cdots & \Lambda_{N,N}(\nu(x,t)) \end{pmatrix}, \tag{39}$$

where the matrix elements are defined as follows:

$$\hat{d}_{i,j}(\nu(x,t)) = \begin{cases} \Lambda_{i,j}(\nu(x,t)), & i \ge j, \\ 0, & \text{otherwise,} \end{cases} \tag{40}$$

and the coefficients $\Lambda_{i,j}(\nu(x,t))$ *have the form*

$$\Lambda_{i,j}(\nu(x,t)) = \Xi_{i,j}^{(1)}(\nu(x,t)) + \Xi_{i,j}^{(2)}(\nu(x,t)), \tag{41}$$

where

$$\Xi_{i,j}^{(1)}(\nu(x,t)) = \frac{(j+2)!(-1)^{i-j}\,(\lambda)_j(\lambda)_{i+j}\,(\hat{\beta}+1)_i}{(i-j)!(\lambda)_i(\lambda)_{2j}(\hat{\beta}+1)_j\,\Gamma(j-\nu(x,t)+3)} \times {}_3F_2\left(\begin{matrix} -i+j, j+3, i+j+\lambda \\ 2j+\lambda+1, 3+j-\nu(x,t) \end{matrix};1\right), \tag{42}$$

$$\begin{aligned}\Xi_{i,j}^{(2)}(\nu(x,t)) = \; & \frac{(-1)^{i+1}(2j+\lambda)(\hat{\beta}+1)_i(\lambda)_j\,\nu(x,t)}{(\hat{\beta}+1)_j\,(\lambda)_i} \sum_{r=j}^{i} \frac{(-1)^r(r+2)\,r!\,(\lambda+1)_{i+r}}{(\hat{\beta}+r+1)(i-r-1)!\,(r-j)!\,(\lambda+1)_{j+r}\,\Gamma(r-\nu(x,t)+4)} \times \\ & {}_4F_3\left(\begin{matrix} 1, r+3, -i+r+1, i+\lambda+r+1 \\ r+2, \hat{\beta}+r+2, 4+r-\nu(x,t) \end{matrix};1\right),\end{aligned} \tag{43}$$

and the functions $\varepsilon_{\ell,i}(x,t)$ *have the form*

$$\varepsilon_{\ell,i}(x,t) = \frac{(-1)^{i+1}\,\ell(\hat{\beta}+1)_i\,\nu(x,t)\,t}{i!\Gamma(3-\nu)}\,{}_3F_2\left(\begin{matrix} -i, i+\lambda, 2 \\ \hat{\beta}+1, 3-\nu(x,t) \end{matrix};1\right). \tag{44}$$

Proof. By considering (7) and applying (6b), it is easy to see that

$${}_0^c\mathcal{D}_t^{\nu(x,t)}\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) = t^{-\nu(x,t)}\sum_{s=0}^{i} c_k^{(i)}\,\frac{\Gamma(s+2)\,t^{s+1}}{\Gamma(s+3-\nu(x,t))}((s+2)(\ell-t)-\ell\,\nu(x,t)), \tag{45}$$

which may be expressed as:

$${}_0^c\mathcal{D}_t^{\nu(x,t)}\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) = t^{-\nu(x,t)}\left(\sum_1 + \sum_2 + \sum_3\right), \tag{46}$$

where

$$\sum_1 = \sum_{s=0}^{i} c_s^{(i)}\,\frac{\Gamma(s+3)}{\Gamma(s+3-\nu(x,t))}(t(\ell-t))t^s, \tag{47}$$

$$\sum_2 = \sum_{s=0}^{i} c_s^{(i)}\,\frac{\ell\,\Gamma(s+2)\,\nu(x,t)}{\Gamma(s+3-\nu(x,t))}t(\ell^s-t^s), \tag{48}$$

and

$$\sum_3 = -\left(\sum_{s=0}^{i} c_s^{(i)}\,\frac{\Gamma(s+2)\,\nu(x,t)}{\Gamma(s+3-\nu(x,t))}\ell^{s+1}\right)t. \tag{49}$$

Using (9), through expansion, the accumulation of similar terms, and some algebraic manipulation (see [54] (p. 758)), it becomes evident that

$$\sum_1 = \sum_{j=0}^{i}\Xi_{i,j}^{(1)}(\nu(x,t))\psi_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t). \tag{50}$$

Also, substitution of the form $c_s^{(i)}$ in (49)—after some simple algebraic manipulation—gives

$$\sum_3 = \varepsilon_{\ell,i}(x,t). \tag{51}$$

Now, it remains that we show that

$$\sum_2 = \sum_{j=0}^{i}\Xi_{i,j}^{(2)}(\nu(x,t))\psi_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t). \tag{52}$$

Using $\ell^s - t^s = \ell^{s-1}(\ell-t)\sum_{k=0}^{s-1}\ell^{-k}\,t^k$ and $c_{i+1}^{(i)} = 0$—after some algebraic manipulation—leads to:

$$\sum_{2} = \sum_{s=0}^{i} c_{s+1}^{(i)} \frac{\ell^{s+1}\,\Gamma(s+3)\,\nu(x,t)}{\Gamma(s+4-\nu(x,t))} t(\ell-t)\left(\sum_{k=0}^{s} \ell^{-k}\,t^{k}\right). \tag{53}$$

By expanding and collecting similar terms, we obtain:

$$\sum_{2} = \sum_{j=0}^{i} B_{i,j}\,t(\ell-t)t^{j}, \tag{54}$$

where

$$B_{i,j} = \sum_{s=j}^{i} c_{s+1}^{(i)} \frac{\ell^{s-j+1}\,\Gamma(s+3)\,\nu(x,t)}{\Gamma(s+4-\nu(x,t))}. \tag{55}$$

Again, using (9), expanding and collecting similar terms leads to:

$$\sum_{2} = \sum_{j=0}^{i}\left(\sum_{r=j}^{i} B_{i,r}\,b_{j}^{(r)}\right)\psi_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t). \tag{56}$$

Using the expressions of $c_{s+1}^{(i)}$ and $b_{j}^{(r)}$ after some algebraic manipulation leads to:

$$\sum_{r=j}^{i} B_{i,r}\,b_{j}^{(r)} = \Xi_{i,j}^{(2)}(\nu(x,t)), \tag{57}$$

then

$$\sum_{2} = \sum_{j=0}^{i} \Xi_{i,j}^{(2)}(\nu(x,t))\psi_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t). \tag{58}$$

In view of Equations (46), (50), (51), and (58), the proof of (37) is complete, which can be expressed as follows:

$${}_{0}^{c}\mathcal{D}_{t}^{\nu(x,t)}\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(t) = t^{-\nu(x,t)}\left([\Lambda_{i,0}(\nu(x,t)),\Lambda_{i,1}(\nu(x,t)),\ldots,\Lambda_{i,i}(\nu(x,t)),0,\ldots,0]\mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(t)+\varepsilon_{\ell,i}(x,t)\right); \tag{59}$$

this gives (38), and the theorem is completely proved. □

As an application of Corollary 2 and Theorem 4, for $N = 4$, $\hat{\alpha} = -\hat{\beta} = 1/2$, and $\nu(x,t) = x\,t$, the OMs H and $\hat{\mathbf{D}}^{(\nu(x,t))}$ have the forms:

$$H = \frac{1}{\ell}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 \\ -\frac{3}{2} & 12 & 0 & 0 & 0 \\ 10 & -\frac{5}{2} & \frac{50}{3} & 0 & 0 \\ -\frac{35}{16} & \frac{315}{16} & -\frac{35}{12} & 21 & 0 \end{pmatrix}_{5\times 5}, \tag{60}$$

and

$$\hat{\mathbf{D}}^{(\nu(x,t))} = \begin{pmatrix} \frac{2}{\Gamma(3-tx)} & 0 & 0 & 0 & 0 \\ \frac{5tx}{\Gamma(4-tx)} & \frac{6}{\Gamma(4-tx)} & 0 & 0 & 0 \\ \frac{3tx(13tx+14)}{4\Gamma(5-tx)} & \frac{63tx}{2\Gamma(5-tx)} & \frac{24}{\Gamma(5-tx)} & 0 & 0 \\ \frac{5tx(5tx(5tx+18)+41)}{8\Gamma(6-tx)} & \frac{15tx(13tx+9)}{2\Gamma(6-tx)} & \frac{180tx}{\Gamma(6-tx)} & \frac{120}{\Gamma(6-tx)} & 0 \\ \frac{35tx(tx(tx(41tx+312)+481)+186)}{64\Gamma(7-tx)} & \frac{105tx(7tx(5tx+11)+38)}{16\Gamma(7-tx)} & \frac{35tx(43tx+22)}{2\Gamma(7-tx)} & \frac{1155tx}{\Gamma(7-tx)} & \frac{720}{\Gamma(7-tx)} \end{pmatrix}_{5\times 5}. \tag{61}$$

6. Numerical Handling for MTVO-TFDWEs Subject to IBCs (2) or DBCs (3)

OMs from Sections 4 and 5 are used in this section to find numerical solutions for MTVO-TFDWEs (1) when IBCs (2) or DBCs (3) are present.

6.1. Homogeneous IBCs and DBCs

Suppose that the IBCs (2) or the DBCs (3) are homogeneous; that is, $f_1(x) = f_2(x) = f_3(t) = f_4(t) = 0$. We can consider

$$y(x,t) \simeq y_N(x,t) = \sum_{i=0}^{N}\sum_{j=0}^{N} a_{i,j}\,\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)\,\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t) = \mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,\mathbf{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t) \tag{62}$$

in the case of IBCs, while in the case of DBCs, we have

$$y(x,t) \simeq y_N(x,t) = \sum_{i=0}^{N}\sum_{j=0}^{N} a_{i,j}\,\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)\,\psi_{\ell,j}^{(\hat{\alpha},\hat{\beta})}(t) = \mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,\mathbf{\Psi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t), \tag{63}$$

where $\boldsymbol{A} = (a_{i,j})_{(N+1)\times(N+1)}$ is the unknown matrix.

Corollaries 1 and 2 and Theorems 2 and 4 allow us to approximate the derivatives of $y(x,t)$ that appear in Equation (1) according to the conditions considered, IBCs/DBCs, as follows:

$$\begin{aligned}
&{}_0^c\mathcal{D}_t^{\nu(x,t)} y_N(x,t) = \\
&\begin{cases}
\sum_{i=0}^{N}\sum_{j=0}^{N} a_{i,j}\,\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)\,{}_0^c\mathcal{D}_t^{\nu(x,t)}\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t) = t^{-\nu(x,t)}\mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,\boldsymbol{D}^{(\nu(x,t))}\mathbf{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t), & \text{forIBCs},\\
\sum_{i=0}^{N}\sum_{j=0}^{N} a_{i,j}\,\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)\,{}_0^c\mathcal{D}_t^{\nu(x,t)}\psi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t) = t^{-\nu(x,t)}\mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,(\hat{\boldsymbol{D}}^{(\nu(x,t))}\mathbf{\Psi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t) + \boldsymbol{\varepsilon}_{\mathcal{T},N}(x,t)), & \text{forDBCs},
\end{cases}
\end{aligned} \tag{64}$$

$$\mathcal{D}_t y_N(x,t) = \begin{cases}
\sum_{i=0}^{N}\sum_{j=0}^{N} a_{i,j}\,\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)\,\mathcal{D}_t\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t) = \mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,(G\mathbf{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t) + \boldsymbol{\eta}_{\mathcal{T},N}^{(1)}(t)), & \text{forIBCs},\\
\sum_{i=0}^{N}\sum_{j=0}^{N} a_{i,j}\,\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)\,\mathcal{D}_t\psi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t) = \mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,(H\,\mathbf{\Psi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t) + \boldsymbol{\xi}_{\mathcal{T},N}^{(1)}(t)), & \text{forDBCs},
\end{cases} \tag{65}$$

and

$$\mathcal{D}_{xx} y_N(x,t) = \begin{cases}
\sum_{i=0}^{N}\sum_{j=0}^{N} a_{i,j}\,\mathcal{D}_{xx}\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)\,\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t) = (H^2\,\mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x) + \boldsymbol{\xi}_{\ell,N}^{(2)}(x))\,\boldsymbol{A}\,\mathbf{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t), & \text{forIBCs},\\
\sum_{i=0}^{N}\sum_{j=0}^{N} a_{i,j}\,\mathcal{D}_{xx}\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)\,\psi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t) = (H^2\,\mathbf{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x) + \boldsymbol{\xi}_{\ell,N}^{(2)}(x))\,\boldsymbol{A}\,\mathbf{\Psi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t), & \text{forDBCs}.
\end{cases} \tag{66}$$

In this method, using the approximations (62)–(66) allows one to write the residual of Equation (1) as:

$$R_N(x,t) = \left({}_0^c\mathcal{D}_t^{\nu(x,t)} + \sum_{j=1}^{m}\varrho_j\,{}_0^c\mathcal{D}_t^{\nu_j(x,t)}\right) y_N(x,t) + \varrho\,\mathcal{D}_t\,y_N(x,t) - \kappa\,\mathcal{D}_{xx}\,y_N(x,t) - g(x,t). \tag{67}$$

Now, we suggest a spectral approach called GSJCOPMM to obtain an approximate solution for (1) under the IBCs (2) or DBCs (3) (with $f_1(x) = f_2(x) = f_3(t) = f_4(t) = 0$). The collocation points x_i, t_i $(0 \le i \le N)$ are chosen to be either the zeros of $\mathcal{J}_{\ell,N+1}^{(\hat{\alpha},\hat{\beta})}(x)$ and $\mathcal{J}_{\mathcal{T},N+1}^{(\hat{\alpha},\hat{\beta})}(t)$ or the points $x_i = \frac{\ell(i+1)}{N+2}$ and $t_i = \frac{\mathcal{T}(j+1)}{N+2}, i,j = 0,1,\ldots,N$, respectively. Therefore, we have

$$R_N(x_i,t_j) = 0, i,j = 0,1,\ldots,N. \tag{68}$$

We can compute $a_{i,j}$ (where $i, j = 0, 1, \ldots, N$) by using the right solver to work through a set of $(N+1)^2$ Equation (68). Achieving the target numerical solution relies heavily on these coefficients.

6.2. Non-Homogeneous IBCs and DBCs

Changing Equation (1) along with the non-homogeneous conditions (2) or (3) into a similar form with homogeneous conditions is an important part of creating the proposed algorithm. The following transformation accomplishes this change:

$$u(x,t) = y(x,t) - q(x,t), q(x,t) = q_1(x,t) + q_2(x,t), \tag{69}$$

where the two functions $q_1(x,t)$ and $q_2(x,t)$ are defined as follows:

In the Case of IBCs:

$$\left.\begin{array}{ll} q_1(x,t) = & \frac{1}{\ell}[(\ell - x)(f_3(t) - f_2(0)\, t - f_3(0)) + x(f_4(t) - f_2(\ell)\, t - f_4(0))], \\ q_2(x,t) = & f_1(x) + f_2(x)\, t. \end{array}\right\} \tag{70}$$

In the Case of DBCs:

$$\left.\begin{array}{ll} q_1(x,t) = & \frac{1}{\ell\mathcal{T}}[(\ell - x)(\mathcal{T} f_3(t) - f_2(0)\, t - f_1(0)\,(\mathcal{T} - t)) - f_1(\ell)\, x\,(\mathcal{T} - t) - f_2(\ell)\, x\, t + \mathcal{T}\, x\, f_4(t)], \\ q_2(x,t) = & \frac{1}{\mathcal{T}}[(\mathcal{T} - t)\, f_1(x) + t\, f_2(x)]. \end{array}\right\} \tag{71}$$

As a result, the current issue may be simplified by solving the following updated equation:

$$\left({}_0^c\mathcal{D}_t^{\nu(x,t)} + \sum_{j=1}^{m} \varrho_j\, {}_0^c\mathcal{D}_t^{\nu_j(x,t)}\right) u_N(x,t) + \varrho\, \mathcal{D}_t u_N(x,t) = \kappa\, \mathcal{D}_{xx} u_N(x,t) + \widetilde{g}(x,t), \tag{72}$$

where

$$\widetilde{g}(x,t) = g(x,t) + \left(\kappa\, \mathcal{D}_{xx} - {}_0^c\mathcal{D}_t^{\nu(x,t)} - \sum_{j=1}^{m} \varrho_j\, {}_0^c\mathcal{D}_t^{\nu_j(x,t)} - \varrho\, \mathcal{D}_t\right) q(x,t),$$

subject to the homogeneous IBCs

$$u(x,0) = u_t(x,0) = u(0,t) = u(\ell,t) = 0, \tag{73}$$

or the homogeneous DBCs

$$u(x,0) = u(x,0) = u(0,t) = u(\ell,t) = 0. \tag{74}$$

Then,

$$y_N(x,t) = u_N(x,t) + q(x,t). \tag{75}$$

Remark 2. *Section 8 explains the algorithm that was used to solve various numerical examples. An Intel®Core™ i9-10850 CPU running at 3.60 GHz, with 10 cores and 20 logical processors and 64.0 GB RAM, was used to do the calculations on a computer system equipped with Mathematica 13.3.1.0 with GSJCOPMM, and the following algorithmic steps may be described to solve the MTVO-TFDWE (Algorithms 1 and 2):*

Algorithm 1 GSJCOPMM algorithm to solve (1) subject to IBCs.

Stage 1.	Given $\hat{\alpha}$, $\hat{\beta}$, ℓ, $\mathcal{T}$, N, $\nu(x,t)$, $\nu_i(x,t)$ and f_j, $i = 1,2,\ldots,m$, $j = 1,2,3,4$.
Stage 2.	Define the basis $\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)$ and $\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t)$, the matrices $\boldsymbol{A}$, $\boldsymbol{G}$, $\boldsymbol{H}$, $\boldsymbol{\eta}_N^{(m)}(t)$, $\boldsymbol{\xi}_{\ell,N}^{(m)}(x)$, $\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)$, $\boldsymbol{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t)$, and calculate the elements of matrices $\boldsymbol{G}$, $\boldsymbol{H}^2$, $\boldsymbol{\eta}_N^{(1)}(t)$, $\boldsymbol{\xi}_{\ell,N}^{(2)}(x)$, $\mathbf{D}^{(\nu(x,t))}$, and $\mathbf{D}^{(\nu_i(x,t))}$, $i = 1,2,\ldots,m$.
Stage 3.	Calculate the matrices: 1. $\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,\boldsymbol{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t)$, 2. $\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,(G\boldsymbol{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t) + \boldsymbol{\eta}_N^{(1)}(t))$, 3. $(H^2\,\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x) + \boldsymbol{\xi}_{\ell,N}^{(2)}(x))\,\boldsymbol{A}\,\boldsymbol{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t)$, 4. $\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,\boldsymbol{D}^{(\nu(x,t))}\boldsymbol{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t)$, 5. $\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,\boldsymbol{D}^{(\nu_i(x,t))}\boldsymbol{\Phi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t)$, $i = 0,1,\ldots,m$.
Stage 4.	Define $R_N(x,t)$ as in Equation (67).
Stage 5.	List $R_N(x_i,t_j) = 0$, $i,j = 0,1,\ldots,N$, defined in Equation (68).
Stage 6.	Use Mathematica's built-in numerical solver to solve the system obtained in [Output 5].
Stage 6.	Calculate $y_N(x,t)$ defined in Equation (62) (Homogeneous IBCs).
Stage 7.	Calculate $q(x,t)$ and $y_N(x,t)$ defined in Equation (75) (Non-homogeneous IBCs).

Algorithm 2 GSJCOPMM algorithm to solve (1) subject to DBCs.

Stage 1.	Given $\hat{\alpha}$, $\hat{\beta}$, ℓ, $\mathcal{T}$, N, $\nu(x,t)$, $\nu_i(x,t)$ and f_j, $i = 1,2,\ldots,m$, $j = 1,2,3,4$.
Stage 2.	Define the basis $\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)$, the matrices $\boldsymbol{A}$, $\boldsymbol{H}$, $\boldsymbol{\varepsilon}_{\mathcal{T},N}(x,t)$, $\boldsymbol{\xi}_{\ell,N}^{(m)}(t)$, $\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)$, and calculate the elements of matrices $\boldsymbol{H}^2$, $\boldsymbol{\xi}_{\mathcal{T},N}^{(1)}(t)$, $\boldsymbol{\xi}_{\ell,N}^{(2)}(x)$, $\hat{\boldsymbol{D}}^{(\nu(x,t))}$, and $\hat{\boldsymbol{D}}^{(\nu_i(x,t))}$, $i = 1,2,\ldots,m$.
Stage 3.	Calculate the matrices: 1. $\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,\boldsymbol{\Psi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t)$, 2. $\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,(H\boldsymbol{\Psi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t) + \boldsymbol{\xi}_{\mathcal{T},N}^{(1)}(t))$, 3. $(H^2\,\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x) + \boldsymbol{\xi}_{\ell,N}^{(2)}(x))\,\boldsymbol{A}\,\boldsymbol{\Psi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t)$, 4. $\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,(\hat{\boldsymbol{D}}^{(\nu(x,t))}\boldsymbol{\Psi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t) + \boldsymbol{\varepsilon}_{\mathcal{T},N}(x,t))$, 5. $\boldsymbol{\Psi}_{\ell,N}^{(\hat{\alpha},\hat{\beta})}(x)\,\boldsymbol{A}\,(\hat{\boldsymbol{D}}^{(\nu_i(x,t))}\boldsymbol{\Psi}_{\mathcal{T},N}^{(\hat{\alpha},\hat{\beta})}(t) + \boldsymbol{\varepsilon}_{\mathcal{T},N}(x,t))$, $i = 0,1,\ldots,m$.
Stage 4.	Define $R_N(x,t)$ as in Equation (67).
Stage 5.	List $R_N(x_i,t_j) = 0$, $i,j = 0,1,\ldots,N$, defined in Equation (68).
Stage 6.	Use Mathematica's built-in numerical solver to solve the system obtained in [Output 5].
Stage 6.	Calculate $y_N(x,t)$ defined in Equation (62) (Homogeneous DBCs).
Stage 7.	Calculate $q(x,t)$ and $y_N(x,t)$ defined in Equation (75) (Non-homogeneous DBCs).

7. Convergence and Error Analysis

Here, we look at the suggested method's convergence and error estimations. The space $\mathcal{S}_{\mathcal{N}}$, defined as follows, is our primary area of interest,

$$\mathcal{S}_{\mathcal{N}} = \begin{cases} Span\left\{\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t) : i,j = 0,1,\ldots,N\right\}, & \text{forIBCs,} \\ Span\left\{\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)\psi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t) : i,j = 0,1,\ldots,N\right\}, & \text{forDBCs.} \end{cases}$$

Additionally, the error between $y(x,t)$ and its approximation $y_N(x,t)$ can be defined by

$$E_N(x,t) = |y(x,t) - y_N(x,t)|. \tag{76}$$

The numerical scheme's error is examined in the paper via the use of the L_∞ norm error estimation

$$\| E_N \|_\infty = \| y - y_N \|_\infty = \max_{(x,t)\in I} |y(x,t) - y_N(x,t)|, I = [0,\ell] \times [0,\mathcal{T}]. \tag{77}$$

In the following, Theorem 5 discusses the error estimate when the solution satisfies IBCs, while Theorem 6 discusses it when DBCs are considered.

Theorem 5. *Assume that $y(x,t) = t^2\, x\,(\ell - x)\, v(x,t)$ and that $y_N(x,t)$ has the expression (62) and represents the best possible approximation (BPA) for $y(x,t)$ out of $\mathcal{S}_N$. Then,*

$$\| E_N \|_\infty \leq \frac{1}{4} (\ell\,\mathcal{T})^2\, W_{q,N}(\ell,\mathcal{T}), q = max\{\hat{\alpha},\hat{\beta},-1/2\} < N+1, \tag{78}$$

where

$$W_{q,N}(\ell,\mathcal{T}) = K_1 \frac{\ell}{2^\lambda}\left(\frac{e\,\ell}{4}\right)^N (N+1)^{q-N-1} + K_2 \frac{\mathcal{T}}{2^\lambda}\left(\frac{e\,\mathcal{T}}{4}\right)^N (N+1)^{q-N-1} + K_3\, \frac{\ell\,\mathcal{T}}{2^\lambda}\left(\frac{e^2\,\ell\,\mathcal{T}}{16}\right)^N (N+1)^{2(q-N-1)}, \tag{79}$$

where the constants K_i, $i = 1,2,3$, are defined as follows:

$$K_1 = \max_{(x,t)\in I} \left|\frac{\partial^{N+1} v(x,t)}{\partial\, x^{N+1}}\right|,\ K_2 = \max_{(x,t)\in I} \left|\frac{\partial^{N+1} v(x,t)}{\partial\, t^{N+1}}\right|,\ K_3 = \max_{(x,t)\in I} \left|\frac{\partial^{2N+2} v(x,t)}{\partial\, x^{N+1}\,\partial\, t^{N+1}}\right|. \tag{80}$$

Proof. Let $v_N(x,t)$ be the interpolating polynomial for $v(x,t)$ at the points (x_i,t_j), $i,j = 0,1,\ldots,N$, where $x_i, (0 \leq i \leq N)$ and $t_j, (0 \leq j \leq N)$ are the roots of $\mathcal{J}^{(\hat{\alpha},\hat{\beta})}_{\ell,N+1}(x)$ and $\mathcal{J}^{(\hat{\alpha},\hat{\beta})}_{\mathcal{T},N+1}(t)$, respectively, such that $N > q-1$. Then, the function $v(x,t)$ can be written as [55]:

$$v(x,t) = v_N(x,t) + \frac{\partial^{N+1} v(\eta_x,t)}{\partial\, x^{N+1}\,(N+1)!}\mathcal{Q}_1(x) + \frac{\partial^{N+1} v(x,\eta_t)}{\partial\, t^{N+1}\,(N+1)!}\mathcal{Q}_2(t) - \frac{\partial^{2N+2} v(\eta\prime_x,\eta\prime_t)}{\partial\, x^{N+1}\,\partial\, t^{N+1}\,((N+1)!)^2}\mathcal{Q}_1(x)\mathcal{Q}_2(t), \tag{81}$$

where $\mathcal{Q}_1(x) = \prod_{i=0}^{N}(x - x_i)$, $\mathcal{Q}_2(t) = \prod_{j=0}^{N}(t - t_j)$, η_x, $\eta\prime_x \in [0,\ell]$, and η_t, $\eta\prime_t \in [0,\mathcal{T}]$.

This is to acquire that:

$$\begin{aligned}\| v - v_N \|_\infty \leq\ & \tfrac{K_1\,\ell^{N+1}}{(N+1)!\widetilde{c}_N} \| \mathcal{J}^{(\hat{\alpha},\hat{\beta})}_{\ell,N+1}(x) \|_\infty + \tfrac{K_2\,\mathcal{T}^{N+1}}{(N+1)!\widetilde{c}_N} \| \mathcal{J}^{(\hat{\alpha},\hat{\beta})}_{\mathcal{T},N+1}(t) \|_\infty \\ & + \tfrac{K_3\,(\ell\,\mathcal{T})^{N+1}}{((N+1)!)^2(\widetilde{c}_N)^2} \| \mathcal{J}^{(\hat{\alpha},\hat{\beta})}_{\ell,N+1}(x) \|_\infty \| \mathcal{J}^{(\hat{\alpha},\hat{\beta})}_{\mathcal{T},N+1}(t) \|_\infty,\end{aligned} \tag{82}$$

where $\widetilde{c}_N = \ell^{N+1}\, c^{N+1}_{N+1} = \frac{\Gamma(2N+\lambda+2)}{(N+1)!\,\Gamma(N+\lambda+1)}$, and c^{N+1}_{N+1} is the leading coefficient of $\mathcal{J}^{(\hat{\alpha},\hat{\beta})}_{\ell,N+1}(x)$.

Using the expression [52] (formula (7.32.2)), we derive:

$$\| \mathcal{J}^{(\hat{\alpha},\hat{\beta})}_{\ell,N+1} \|_\infty \simeq (N+1)^q, \tag{83}$$

and thus the inequality (82) takes the form:

$$\begin{aligned}\| v - v_N \|_\infty \leq\ & K_1 \tfrac{\ell^{N+1}\Gamma(N+\lambda+1)(N+1)^q}{\Gamma(2N+\lambda+2)} + K_2 \tfrac{\mathcal{T}^{N+1}\Gamma(N+\lambda+1)(N+1)^q}{\Gamma(2N+\lambda+2)} \\ & + K_3\,(\ell\,\mathcal{T})^{N+1} \tfrac{\Gamma^2(N+\lambda+1)(N+1)^{2q}}{\Gamma^2(2N+\lambda+2)}.\end{aligned} \tag{84}$$

By utilizing results (see [56] (pp. 232–233)),

$$\Gamma(m+\lambda) = \mathcal{O}\left(m^{\lambda-1}\, m!\right), (2m)! = \frac{1}{\sqrt{\pi}}\, 4^m\, m!\, \Gamma(m+1/2), m! = \mathcal{O}\left(\sqrt{2\pi\, m}\left(\frac{m}{e}\right)^m\right), \quad (85)$$

the inequality (84) takes the form:

$$\| v - v_N \|_\infty \leq W_{q,N}(\ell, \mathcal{T}), \quad (86)$$

where

$$W_{q,N}(\ell, \mathcal{T}) = K_1 \frac{\ell}{2^\lambda}\left(\frac{e\,\ell}{4}\right)^N (N+1)^{q-N-1} + K_2 \frac{\mathcal{T}}{2^\lambda}\left(\frac{e\,\mathcal{T}}{4}\right)^N (N+1)^{q-N-1} + K_3 \frac{\ell\,\mathcal{T}}{2^{2\lambda}}\left(\frac{e^2\,\ell\,\mathcal{T}}{16}\right)^N (N+1)^{2(q-N-1)}. \quad (87)$$

Now, consider the approximation $y(x,t) \simeq Y_N(x,t) = t^2\, x\, (\ell - x) v_N(x,t)$; then,

$$\| y - Y_N \|_\infty \leq \frac{1}{4}(\ell\,\mathcal{T})^2 \| u - v_N \|_\infty \leq \frac{1}{4}(\ell\,\mathcal{T})^2\, W_{q,N}(\ell, \mathcal{T}). \quad (88)$$

Since $y_N(x,t) \in \mathcal{S}_N$ represents the BPA to $y(x,t)$, then

$$\| y - y_N \|_\infty \leq \| y - h \|_\infty, \forall h \in \mathcal{S}_N, \quad (89)$$

and therefore,

$$\| y - y_N \|_\infty \leq \| y - Y_N \|_\infty \leq \frac{1}{4}(\ell\,\mathcal{T})^2\, W_{q,N}(\ell, \mathcal{T}). \quad (90)$$

□

Theorem 6. *Assume that $y(x,t) = t\, x\, (\ell - x)(\mathcal{T} - t)\, v(x,t)$ and that $y_N(x,t)$ has the expression (63) and represents the BPA for $y(x,t)$ out of $\mathcal{S}_\mathcal{N}$. Then,*

$$\| E_N \|_\infty \leq \frac{1}{16}(\ell\,\mathcal{T})^2\, W_{q,N}(\ell, \mathcal{T}), q = max\{\hat{\alpha}, \hat{\beta}, -1/2\} < N+1, \quad (91)$$

where $W_{q,N}(\ell, \mathcal{T})$ is defined by (79).

Proof. Following the same procedures in the proof of Theorem 5, we obtain (91). □

The next corollary demonstrates how quickly the resulting errors are convergent.

Corollary 3. *For all $N > q - 1$, the following estimate holds:*

$$\| E_N \|_\infty = \mathcal{O}\left(\left(\frac{e^2\,\ell\,\mathcal{T}}{16}\right)^N N^{q-N-1}\right). \quad (92)$$

The stability of error, or the process of estimating the propagation of error, is the focus of the subsequent theorem.

Theorem 7. *Given any two iterative estimates of $y(x,t)$, the result is:*

$$|y_{N+1} - y_N| \lesssim \mathcal{O}\left(\left(\frac{e^2\,\ell\,\mathcal{T}}{16}\right)^N N^{q-N-1}\right), N > q - 1. \quad (93)$$

Proof. We have

$$|y_{N+1} - y_N| = |y_{N+1} - y + y - y_N| \leq |y - y_{N+1}| + |y - y_N| \leq \| E_{N+1} \|_\infty + \| E_N \|_\infty.$$

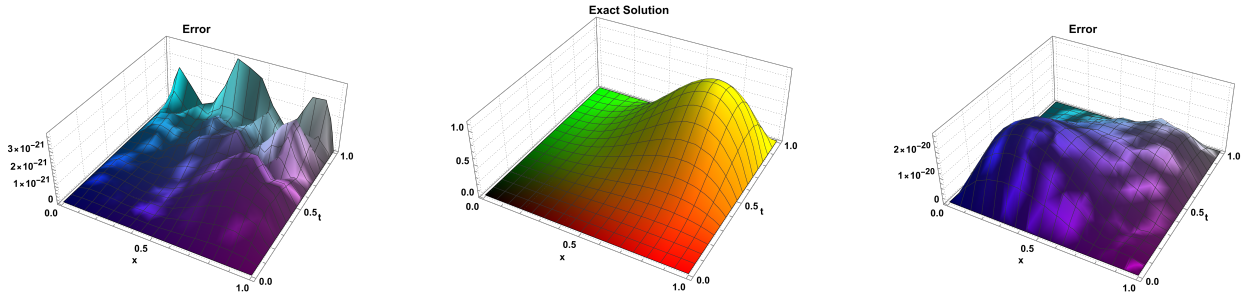
By considering (92), we can obtain (93). □

Remark 3. *While relative error can be a useful measure in certain situations, the choice to utilize absolute error in our study was driven by considerations of stability, consistency with established literature, practical interpretation, and the characteristics of the problem at hand. We believe that using absolute error allows for a reliable and meaningful assessment of the accuracy of our proposed method in solving MTVO-TFDWEs.*

8. Numerical Simulations

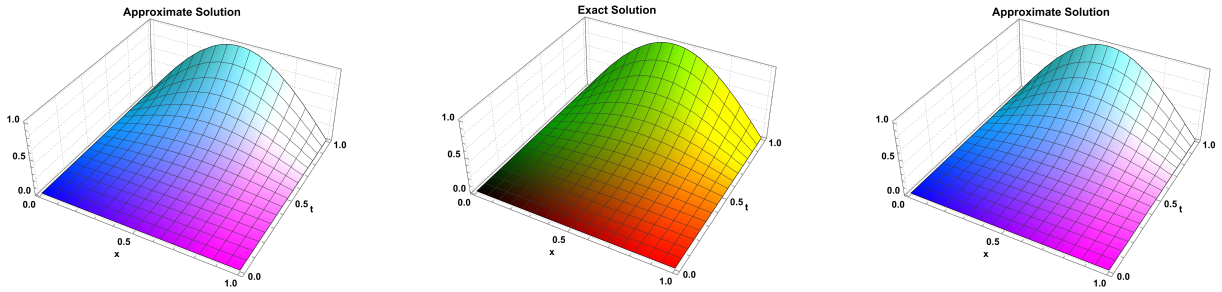
In order to show how the approach presented in Section 6 works and how efficient it is, four examples are provided in this section. The assessment is based on MAE between the precise and approximate solutions. We show in Example 1 that, for certain simple cases of the functions $\nu(x,t)$ and $\nu_j(x,t)$, $j = 1,2,\ldots,m$, we can use *GSJCOPMM* to find the precise solution to the given numerical problem, whether it involves IBCs or DBCs, and that it has a polynomial solution of degree N. This solution can be found by combining $\psi_{\ell,i}^{(\hat{\alpha},\hat{\beta})}(x)$, $\phi_{\mathcal{T},j}^{(\hat{\alpha},\hat{\beta})}(t)$, $i,j = 0,1,\ldots,N-2$. Otherwise, some numerical solutions are obtained with high accuracy. Additionally, the choice of examples in our study is motivated by the desire to cover different conditions, compare with existing results, showcase accuracy and efficiency, and demonstrate practical relevance. These examples collectively provide a comprehensive assessment of the performance and applicability of our proposed method for solving MTVO-TFDWEs.

In addition, Tables 1–4 display the calculated errors that were obtained to produce numerical solutions $y_N(x,t)$ using *GSJCOPMM* with $N = 0,1,\ldots,14$. In these tables, excellent computational results are obtained. The comparisons between *GSJCOPMM* and other techniques in [57,58] are presented in Tables 5 and 6. The tables show that, compared to the other methods, *GSJCOPMM* gives the most accurate results. Furthermore, as can be seen in Figures 1, 2, 3, 4a, 5a, 6, 7, 8a, 9a, and 10, the precise and approximate solutions in Examples 1–4 have a high level of agreement. The absolute and log errors in Figures 4b, 5b, and 9 serve to illustrate the convergence and stability of the solutions to the given Problems 2 and 3 when applying *GSJCOPMM* and using different N, $\hat{\alpha}$, and $\hat{\beta}$. The choice of parameter combinations enables us to fine-tune our method, which leads to the most accurate results.



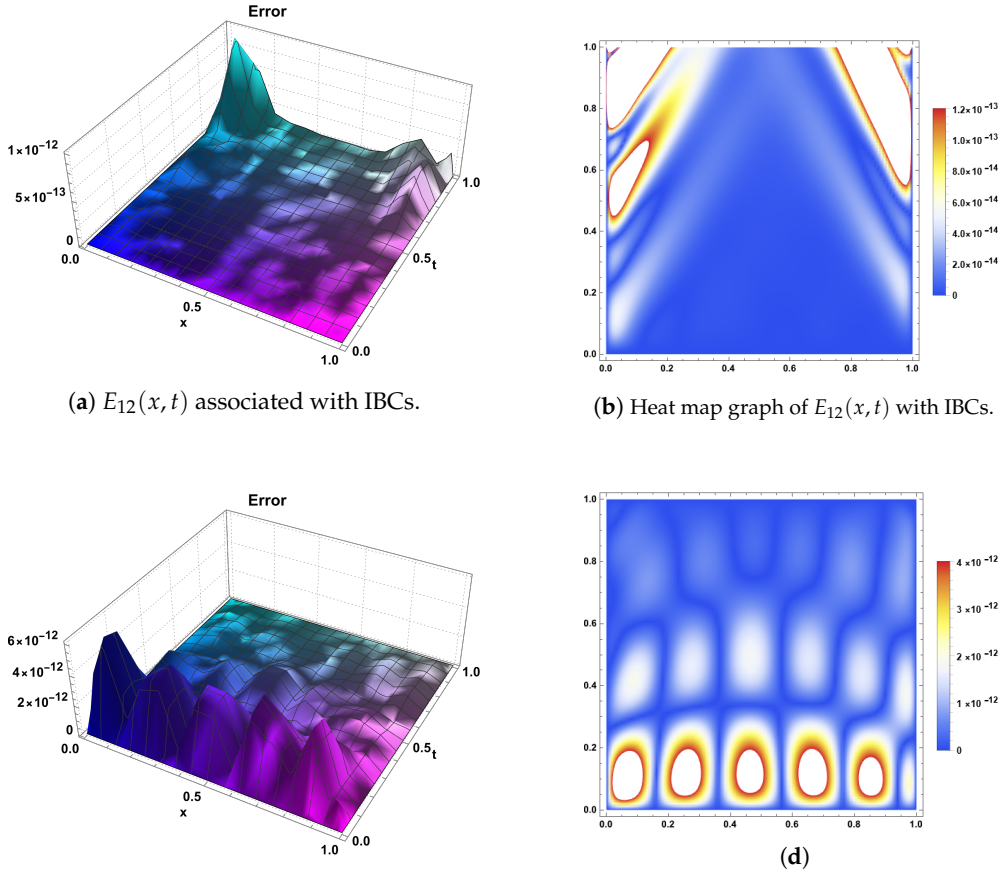
(**a**) Error associated with IBCs (**b**) Exact solution (**c**) Error associated with DBCs

Figure 1. Exact solution and error plots for Example 1 using $\hat{\alpha} = 0$, $\hat{\beta} = 1/2$ and $N = 7$.



(**a**) $y_{12}(x,t)$ associated with IBCs. (**b**) Exact solution plot for Example 2. (**c**) $y_{12}(x,t)$ associated with DBCs.

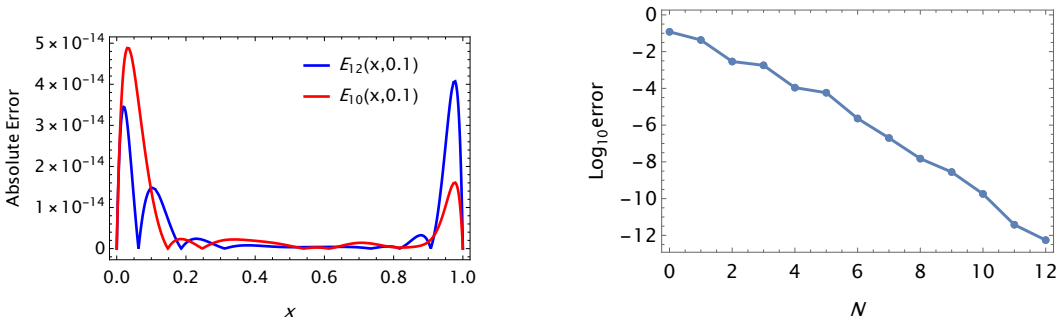
Figure 2. Exact and approximate solutions using $(N = 12,\ \hat{\alpha} = 0,\ \hat{\beta} = 1/2)$ for Example 2.



(**a**) $E_{12}(x,t)$ associated with IBCs. (**b**) Heat map graph of $E_{12}(x,t)$ with IBCs.

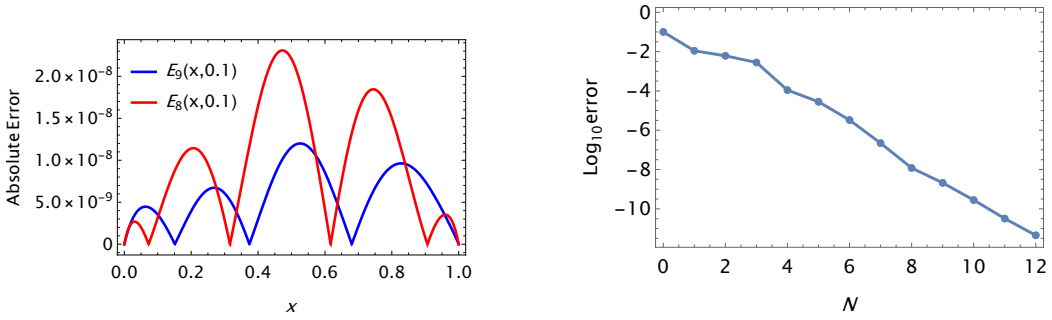
(**c**) $E_{12}(x,t)$ associated with DBCs. (**d**) Heat map graph of $E_{12}(x,t)$ with DBCs.

Figure 3. Error functions and their heat map graphs using $(N = 12,\ \hat{\alpha} = 0,\ \hat{\beta} = 1/2)$ for Example 2.



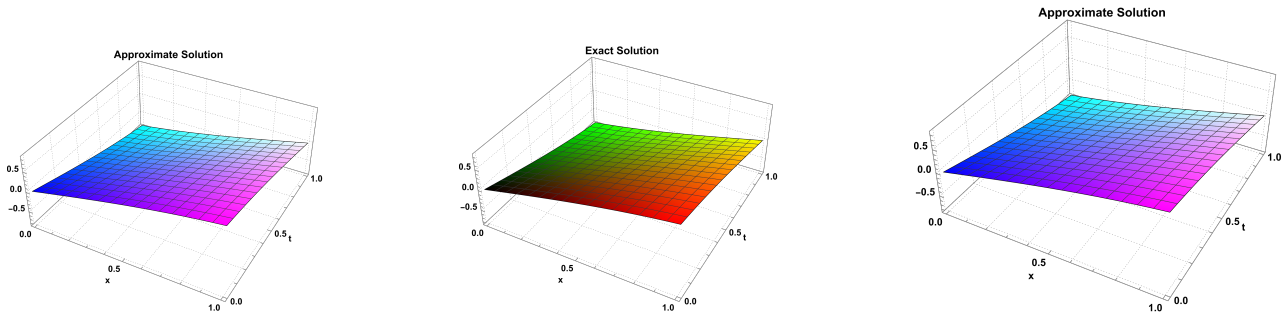
(**a**) Absolute errors at $t = 0.1$. (**b**) Graph of $Log_{10}E_N$ against N.

Figure 4. Error results using $(\hat{\alpha} = 0,\ \hat{\beta} = 1/2)$ for Example 2 associated with IBCs.



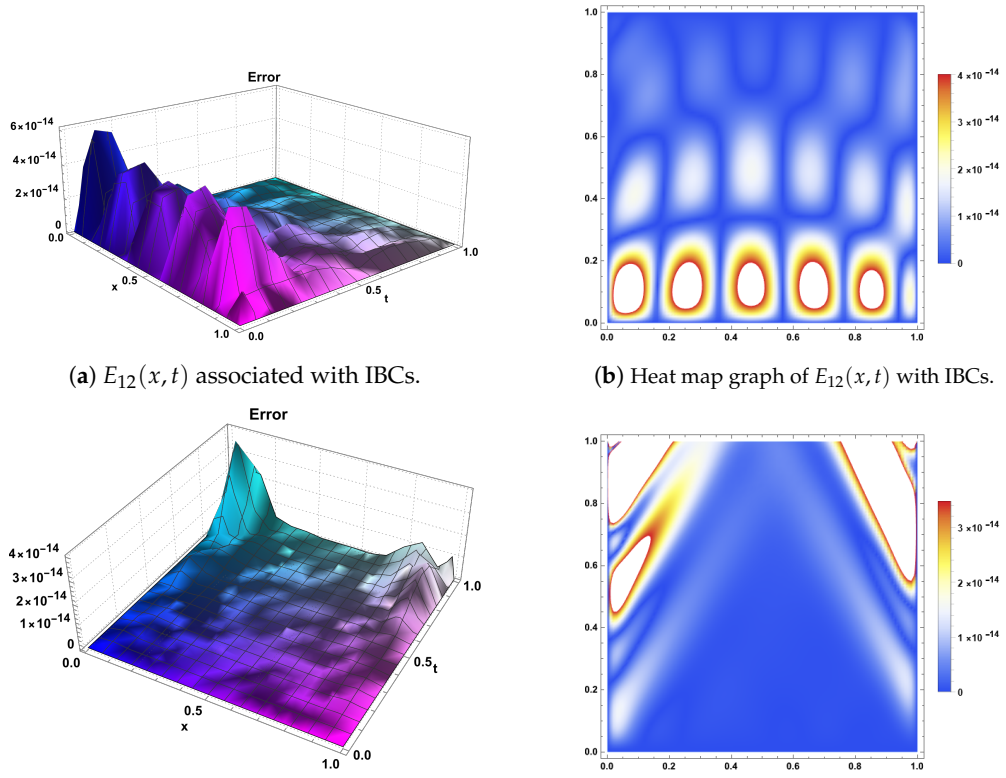
(**a**) Absolute errors at $t = 0.1$. (**b**) Graph of $Log_{10}E_N$ against N.

Figure 5. Error results using $(\hat{\alpha} = 0, \hat{\beta} = 1/2)$ for Example 2 associated with DBCs.



(**a**) $y_{12}(x,t)$ associated with IBCs. (**b**) Exact solution plot for Example 3. (**c**) $y_{12}(x,t)$ associated with DBCs.

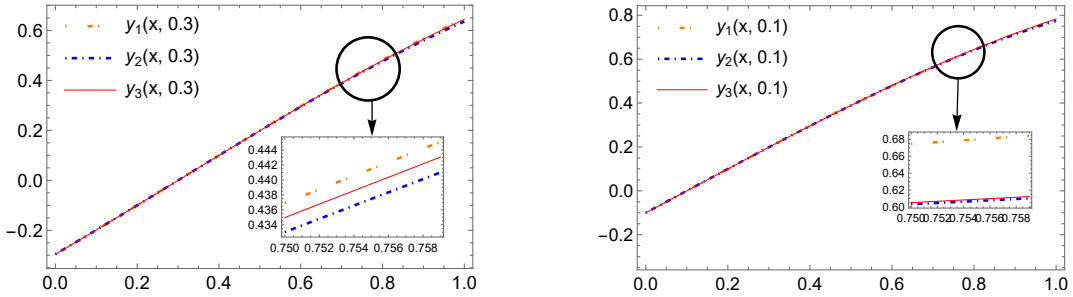
Figure 6. Exact and approximate solutions using $(N = 12, \hat{\alpha} = \hat{\beta} = 1/2)$ for Example 3.



(**a**) $E_{12}(x,t)$ associated with IBCs. (**b**) Heat map graph of $E_{12}(x,t)$ with IBCs.

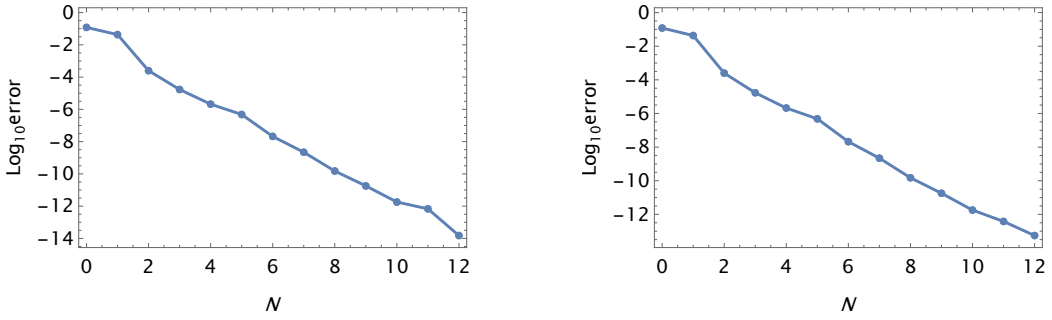
(**c**) $E_{12}(x,t)$ associated with DBCs. (**d**) Heat map graph of $E_{12}(x,t)$ with DBCs.

Figure 7. Error functions and their heat map graphs using $(N = 12, \hat{\alpha} = \hat{\beta} = 1/2)$ for Example 3.



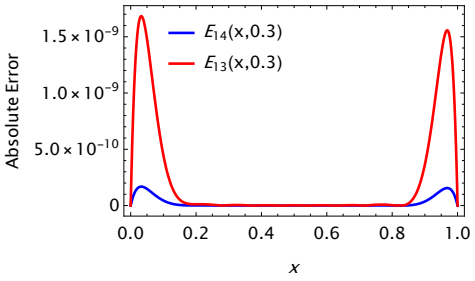
(**a**) (**b**)

Figure 8. Approximate solution plots for Example 3 associated with IBCs or DBCs using $(N = 1,2,3, \hat{\alpha} = \hat{\beta} = 1/2)$. (**a**) Approximate solution plots at $t = 0.3$ associated with IBCs; (**b**) approximate solution plots $t = 0.1$ associated with DBCs.

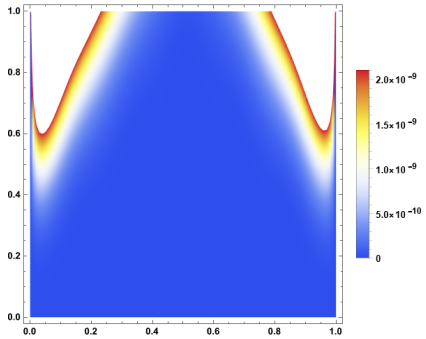


(**a**) (**b**)

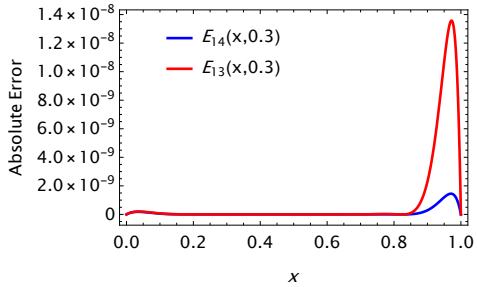
Figure 9. Graph of $Log_{10}E_N$ for Example 3 associated with IBCs (**a**) or DBCs (**b**) using $(N = 0,1,\ldots,12, \hat{\alpha} = \hat{\beta} = 1/2)$.



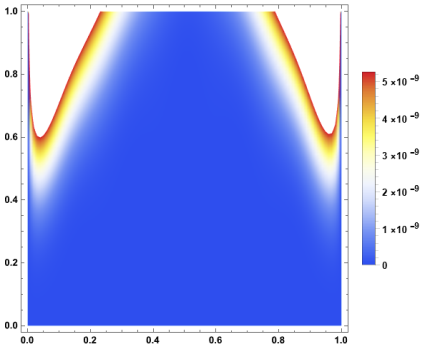
(**a**) Absolute errors at $t = 0.3$ with IBCs.



(**b**) Heat map graph of $E_{14}(x,t)$ with IBCs.



(**c**) Absolute errors at $t = 0.3$ with DBCs.



(**d**) Heat map graph of $E_{14}(x,t)$ with DBCs.

Figure 10. Error results and their heat map graphs using $(\hat{\alpha} = \hat{\beta} = 1/2)$ for Example 4.

Table 1. MAEs of Example 1 using various $\hat{\alpha}$, $\hat{\beta}$, and N.

$\hat{\alpha}$	$\hat{\beta}$	IBCs/DBCs	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$	$N = 7$
0	0	IBCs	1.3×10^{-16}	2.3×10^{-17}	1.9×10^{-18}	2.4×10^{-19}	2.7×10^{-20}	2.4×10^{-21}
		DBCs	2.6×10^{-16}	3.4×10^{-17}	3.5×10^{-18}	3.9×10^{-19}	3.6×10^{-20}	1.6×10^{-20}
		CPU time	0.231	0.312	0.401	0.421	0.451	0.515
1/2	1/2	IBCs	2.4×10^{-16}	1.3×10^{-17}	4.1×10^{-18}	7.2×10^{-19}	1.8×10^{-20}	2.5×10^{-21}
		DBCs	2.7×10^{-16}	3.6×10^{-17}	3.8×10^{-18}	3.9×10^{-19}	4.4×10^{-20}	2.7×10^{-20}
		CPU time	0.232	0.313	0.403	0.422	0.453	0.517
−1/2	−1/2	IBCs	1.5×10^{-16}	2.8×10^{-17}	4.8×10^{-18}	1.2×10^{-19}	1.8×10^{-20}	2.6×10^{-21}
		DBCs	1.6×10^{-16}	3.6×10^{-17}	1.9×10^{-18}	1.9×10^{-19}	5.4×10^{-20}	2.1×10^{-20}
		CPU time	0.221	0.310	0.402	0.422	0.450	0.517
1	0	IBCs	3.3×10^{-15}	4.3×10^{-16}	3.7×10^{-17}	2.2×10^{-18}	1.8×10^{-19}	3.1×10^{-21}
		DBCs	3.9×10^{-16}	3.6×10^{-17}	2.1×10^{-18}	1.9×10^{-19}	2.6×10^{-20}	2.2×10^{-20}
		CPU time	0.233	0.314	0.407	0.425	0.458	0.519
0	1	IBCs	4.3×10^{-15}	2.3×10^{-16}	4.5×10^{-17}	4.4×10^{-18}	1.7×10^{-19}	2.9×10^{-21}
		DBCs	2.9×10^{-16}	3.2×10^{-17}	2.5×10^{-18}	3.8×10^{-19}	8.6×10^{-20}	2.3×10^{-20}
		CPU time	0.234	0.315	0.407	0.426	0.451	0.519

Table 2. MAEs of Example 2 using various $\hat{\alpha}$, $\hat{\beta}$, and N.

$\hat{\alpha}$	$\hat{\beta}$	IBCs/DBCs	$N = 0$	$N = 1$	$N = 2$	$N = 5$	$N = 8$	$N = 12$
0	0	IBCs	1.2×10^{-1}	4.3×10^{-2}	2.9×10^{-3}	5.8×10^{-5}	1.5×10^{-8}	5.5×10^{-13}
		DBCs	1.0×10^{-1}	1.1×10^{-2}	6.1×10^{-3}	4.0×10^{-5}	1.2×10^{-8}	4.5×10^{-12}
		CPU time	0.101	0.122	0.232	0.432	0.521	0.735
1/2	1/2	IBCs	1.3×10^{-1}	4.4×10^{-2}	2.8×10^{-3}	5.4×10^{-5}	1.2×10^{-8}	5.4×10^{-13}
		DBCs	1.1×10^{-1}	1.1×10^{-2}	5.9×10^{-3}	4.2×10^{-5}	1.4×10^{-8}	4.0×10^{-12}
		CPU time	0.102	0.123	0.234	0.433	0.525	0.745
−1/2	−1/2	IBCs	1.4×10^{-1}	3.3×10^{-2}	2.1×10^{-3}	4.9×10^{-5}	1.3×10^{-8}	5.7×10^{-13}
		DBCs	1.2×10^{-1}	1.3×10^{-2}	5.2×10^{-3}	3.9×10^{-5}	1.5×10^{-8}	4.7×10^{-12}
		CPU time	0.101	0.121	0.231	0.433	0.520	0.732
1	0	IBCs	1.5×10^{-1}	3.9×10^{-2}	3.0×10^{-3}	6.0×10^{-5}	2.5×10^{-8}	4.5×10^{-13}
		DBCs	1.3×10^{-1}	1.4×10^{-2}	6.2×10^{-3}	3.3×10^{-5}	2.0×10^{-8}	3.5×10^{-12}
		CPU time	0.104	0.124	0.234	0.435	0.521	0.738
0	1	IBCs	1.7×10^{-1}	4.5×10^{-2}	2.7×10^{-3}	5.1×10^{-5}	1.3×10^{-8}	5.5×10^{-13}
		DBCs	1.7×10^{-1}	1.6×10^{-2}	6.5×10^{-3}	4.5×10^{-5}	1.8×10^{-8}	4.1×10^{-12}
		CPU time	0.102	0.125	0.237	0.438	0.529	0.741

Table 3. MAEs of Example 3 using various $\hat{\alpha}$, $\hat{\beta}$, and N.

$\hat{\alpha}$	$\hat{\beta}$	IBCs/DBCs	$N = 2$	$N = 4$	$N = 6$	$N = 8$	$N = 10$	$N = 12$
0	0	IBCs	1.4×10^{-4}	3.3×10^{-6}	2.2×10^{-8}	4.8×10^{-10}	2.5×10^{-12}	4.4×10^{-13}
		DBCs	1.2×10^{-4}	2.1×10^{-6}	1.1×10^{-8}	3.2×10^{-10}	2.3×10^{-12}	6.8×10^{-14}
		CPU time	0.231	0.401	0.451	0.521	0.601	0.735
1/2	1/2	IBCs	1.0×10^{-4}	2.1×10^{-6}	2.4×10^{-8}	4.4×10^{-10}	2.2×10^{-12}	5.4×10^{-14}
		DBCs	1.3×10^{-4}	2.1×10^{-6}	4.9×10^{-8}	4.3×10^{-10}	2.1×10^{-12}	4.2×10^{-14}
		CPU time	0.233	0.405	0.454	0.525	0.607	0.740
−1/2	−1/2	IBCs	1.5×10^{-4}	2.4×10^{-6}	2.3×10^{-8}	2.4×10^{-10}	1.4×10^{-12}	5.7×10^{-14}
		DBCs	1.1×10^{-4}	1.5×10^{-6}	4.2×10^{-8}	2.9×10^{-10}	3.5×10^{-12}	2.2×10^{-13}
		CPU time	0.240	0.410	0.459	0.528	0.610	0.740
1	0	IBCs	1.2×10^{-4}	2.3×10^{-6}	3.1×10^{-8}	5.0×10^{-10}	3.5×10^{-12}	8.2×10^{-14}
		DBCs	1.3×10^{-4}	2.5×10^{-6}	5.2×10^{-8}	4.3×10^{-10}	3.0×10^{-12}	6.5×10^{-14}
		CPU time	0.238	0.409	0.458	0.529	0.608	0.739
0	1	IBCs	1.8×10^{-4}	2.3×10^{-6}	2.4×10^{-8}	5.2×10^{-10}	2.3×10^{-12}	4.5×10^{-13}
		DBCs	1.7×10^{-4}	2.5×10^{-6}	5.5×10^{-8}	3.5×10^{-10}	2.8×10^{-12}	6.1×10^{-14}
		CPU time	0.237	0.407	0.456	0.527	0.608	0.738

Table 4. MAEs of Example 4 using various $\hat{\alpha}$, $\hat{\beta}$, and N.

$\hat{\alpha}$	$\hat{\beta}$	IBCs/DBCs	$N = 2$	$N = 4$	$N = 6$	$N = 8$	$N = 10$	$N = 14$
1	1	IBCs	1.2×10^{-2}	2.3×10^{-3}	2.0×10^{-4}	3.8×10^{-5}	2.1×10^{-6}	4.5×10^{-9}
		DBCs	1.1×10^{-2}	3.3×10^{-3}	2.2×10^{-4}	4.8×10^{-5}	3.1×10^{-6}	2.2×10^{-8}
		CPU time	0.235	0.402	0.454	0.522	0.604	0.988
3/2	1/2	IBCs	1.3×10^{-2}	4.3×10^{-3}	5.0×10^{-4}	3.7×10^{-5}	2.6×10^{-6}	4.4×10^{-9}
		DBCs	2.2×10^{-2}	3.1×10^{-3}	2.4×10^{-4}	3.5×10^{-5}	2.7×10^{-6}	4.4×10^{-8}
		CPU time	0.230	0.317	0.411	0.430	0.460	0.980
1/2	3/2	IBCs	2.2×10^{-2}	4.3×10^{-3}	3.0×10^{-4}	3.9×10^{-5}	2.3×10^{-6}	4.6×10^{-9}
		DBCs	4.2×10^{-2}	3.2×10^{-3}	1.5×10^{-4}	5.8×10^{-5}	2.9×10^{-6}	4.4×10^{-8}
		CPU time	0.234	0.319	0.412	0.429	0.459	0.991
2	3	IBCs	3.2×10^{-2}	4.3×10^{-3}	1.0×10^{-4}	1.8×10^{-5}	2.2×10^{-6}	4.6×10^{-9}
		DBCs	5.2×10^{-2}	3.4×10^{-3}	2.3×10^{-4}	3.6×10^{-5}	2.0×10^{-6}	6.1×10^{-9}
		CPU time	0.239	0.320	0.413	0.430	0.461	1.001
3	2	IBCs	1.7×10^{-2}	2.8×10^{-3}	2.5×10^{-4}	2.8×10^{-5}	2.4×10^{-6}	4.7×10^{-9}
		DBCs	2.3×10^{-2}	4.3×10^{-3}	2.5×10^{-4}	3.9×10^{-5}	2.8×10^{-6}	4.6×10^{-9}
		CPU time	0.238	0.319	0.412	0.428	0.458	0.998

Table 5. The method's comparisons [57,58] and *GSJCOPMM* in Example 2 with IBCs and DBCs.

(x, t)	***GSJCOPMM* (IBCs)** $(\hat{\alpha} = \hat{\beta} = 1, N = 10)$	***GSJCOPMM* (DBCs)** $(\hat{\alpha} = \hat{\beta} = 1, N = 8)$	**[57] (IBCs)** $(k = 1, M = 10)$	**[58] (DBCs)** $(N = M = 8)$
(0.1, 0.1)	1.47×10^{-14}	3.15×10^{-9}	3.05×10^{-4}	1.69×10^{-7}
(0.2, 0.2)	4.34×10^{-15}	1.09×10^{-8}	8.35×10^{-6}	1.12×10^{-7}
(0.3, 0.3)	2.55×10^{-15}	4.39×10^{-9}	2.35×10^{-7}	1.97×10^{-8}
(0.4, 0.4)	5.41×10^{-16}	1.48×10^{-8}	5.40×10^{-7}	1.33×10^{-7}
(0.5, 0.5)	2.08×10^{-16}	8.08×10^{-9}	6.60×10^{-8}	1.38×10^{-8}
(0.6, 0.6)	4.88×10^{-15}	3.43×10^{-9}	1.53×10^{-7}	1.36×10^{-7}
(0.7, 0.7)	4.78×10^{-13}	2.28×10^{-9}	3.73×10^{-8}	1.04×10^{-7}
(0.8, 0.8)	8.71×10^{-12}	1.70×10^{-9}	1.06×10^{-7}	1.90×10^{-6}
(0.9, 0.9)	5.22×10^{-11}	3.71×10^{-9}	6.65×10^{-8}	4.68×10^{-5}

Table 6. The method's comparisons [57,58] and *GSJCOPMM* in Example 3 with DBCs.

(x, t)	***GSJCOPMM*** $(\hat{\alpha} = \hat{\beta} = 1, \mathbf{N} = 10)$	***GSJCOPMM*** $(\hat{\alpha} = \hat{\beta} = 1, N = 7)$	**[58]** $(M = N = 7)$	**[57]** $(k = 1, M = 7)$	**[57]** $(k = 1, M = 10)$
(0.1, 0.1)	1.39×10^{-14}	1.1×10^{-13}	1.29×10^{-13}	5.62×10^{-8}	1.04×10^{-12}
(0.2, 0.2)	3.66×10^{-15}	6.2×10^{-13}	3.41×10^{-12}	1.87×10^{-8}	8.61×10^{-12}
(0.3, 0.3)	2.07×10^{-15}	7.5×10^{-13}	2.37×10^{-12}	1.34×10^{-7}	1.11×10^{-12}
(0.4, 0.4)	1.60×10^{-15}	1.6×10^{-12}	4.27×10^{-11}	3.25×10^{-7}	6.39×10^{-13}
(0.5, 0.5)	2.58×10^{-15}	2.6×10^{-12}	9.46×10^{-11}	2.57×10^{-7}	2.06×10^{-12}
(0.6, 0.6)	4.64×10^{-15}	3.3×10^{-12}	1.14×10^{-10}	1.66×10^{-7}	5.52×10^{-12}
(0.7, 0.7)	2.07×10^{-14}	4.6×10^{-12}	3.26×10^{-11}	8.95×10^{-8}	3.11×10^{-12}
(0.8, 0.8)	2.39×10^{-14}	1.7×10^{-10}	1.93×10^{-9}	3.78×10^{-8}	8.14×10^{-13}
(0.9, 0.9)	3.60×10^{-13}	1.2×10^{-9}	1.32×10^{-8}	3.34×10^{-8}	1.89×10^{-12}

Problem 1. *Consider the following MTVO-TFDE equation [57,58]:*

$$ {}_0^c\mathcal{D}_t^{\nu(x,t)} y(x,t) + {}_0^c\mathcal{D}_t^{\nu_1(x,t)} y(x,t) + y_t(x,t) = y_{xx}(x,t) + g(x,t), (x,t) \in [0,1] \times [0,1], \quad (94) $$

subject to IBCs:

$$ y(x,0) = 0,\ y_t(x,0) = 0,\ y(0,t) = 0,\ y(1,t) = 0, \quad (95) $$

or DBCs:

$$ y(x,0) = 0,\ y(x,1) = 0,\ y(0,t) = 0,\ y(1,t) = 0, \quad (96) $$

where $g(x,t)$ is selected such that the solution to (94) is $y(x,t) = 100\, x^3\, t^3\, (1-x)(1-t)$.

The application of GSJCOPMM to obtain approximated solutions with the two cases of IBCs (95) and DBCs (96) using $N = 2, \ldots, 7$, $\nu(x,t) = 2 - 0.3\,e^{-x\,t}$ and $\nu_1(x,t) = 2 - 0.6\,e^{-x\,t}$ gives acceptable accuracy, as shown in Table 1. Figure 1a,c demonstrates that the obtained solutions achieve an accuracy of 10^{-21} at $N = 7$, aligning with the exact solution depicted in Figure 1b.

Remark 4. *It is worth noting that, although the precise solution to (94) is a polynomial, the obtained approximate solutions did not give this precise solution; this is because of the complexity form of the two functions $\nu(x,t)$, $\nu_1(x,t)$. However, in other special cases, for instance, $\nu(x,t) = x\,t, \nu_1(x,t) = 2x\,t$, for all available values of $\hat{\alpha}$ and $\hat{\beta}$, the precise solution is obtained using $N = 2$, where the expansion coefficients $c_{i,j} = 0, i, j = 0, 1, 2$, have the form:*

$$c_{0,0} = \frac{100\left(\hat{\alpha}\hat{\beta}^3+4\hat{\alpha}\hat{\beta}^2+5\hat{\alpha}\hat{\beta}+2\hat{\alpha}+\hat{\beta}^3+4\hat{\beta}^2+5\hat{\beta}+2\right)}{(\hat{\alpha}+\hat{\beta}+2)^2(\hat{\alpha}+\hat{\beta}+3)^2}, \quad c_{0,1} = \frac{200\left(\hat{\alpha}\hat{\beta}^2+3\hat{\alpha}\hat{\beta}+2\hat{\alpha}+\hat{\beta}^2+3\hat{\beta}+2\right)}{(\hat{\alpha}+\hat{\beta}+2)^2(\hat{\alpha}+\hat{\beta}+3)(\hat{\alpha}+\hat{\beta}+4)}, \quad c_{0,2} = \frac{200(\hat{\alpha}\hat{\beta}+\hat{\alpha}+\hat{\beta}+1)}{(\hat{\alpha}+\hat{\beta}+2)(\hat{\alpha}+\hat{\beta}+3)^2(\hat{\alpha}+\hat{\beta}+4)},$$
$$c_{1,0} = -\frac{100\left(-\hat{\alpha}\hat{\beta}^2-3\hat{\alpha}\hat{\beta}-2\hat{\alpha}+\hat{\beta}^3+3\hat{\beta}^2+2\hat{\beta}\right)}{(\hat{\alpha}+\hat{\beta}+2)^2(\hat{\alpha}+\hat{\beta}+3)(\hat{\alpha}+\hat{\beta}+4)}, \quad c_{1,1} = -\frac{200(\hat{\beta}+2)(\hat{\beta}-\hat{\alpha})}{(\hat{\alpha}+\hat{\beta}+2)^2(\hat{\alpha}+\hat{\beta}+4)^2}, \quad c_{1,2} = -\frac{200(\hat{\beta}-\hat{\alpha})}{(\hat{\alpha}+\hat{\beta}+2)(\hat{\alpha}+\hat{\beta}+3)(\hat{\alpha}+\hat{\beta}+4)^2},$$
$$c_{2,0} = -\frac{200\left(\hat{\beta}^2+3\hat{\beta}+2\right)}{(\hat{\alpha}+\hat{\beta}+2)(\hat{\alpha}+\hat{\beta}+3)^2(\hat{\alpha}+\hat{\beta}+4)}, \quad c_{2,1} = -\frac{400(\hat{\beta}+2)}{(\hat{\alpha}+\hat{\beta}+2)(\hat{\alpha}+\hat{\beta}+3)(\hat{\alpha}+\hat{\beta}+4)^2}, \quad c_{2,2} = -\frac{400}{(\hat{\alpha}+\hat{\beta}+3)^2(\hat{\alpha}+\hat{\beta}+4)^2}.$$

Remark 5. *Although errors smaller than 10^{-16} may not have direct practical significance, increasing the value of N can still meaningfully improve accuracy in numerical computations. The choice of N depends on the specific requirements and desired level of accuracy for the problem under consideration. We emphasize that the relative improvement in accuracy should be considered when evaluating the meaningfulness of different values of N.*

Problem 2. *Consider the following MTVO-TFDE equation [57,58]:*

$$\left({}_0^c\mathcal{D}_t^{\nu(x,t)} + \sum_{j=1}^{4} \frac{1}{j+1}\,{}_0^c\mathcal{D}_t^{\nu_j(x,t)}\right) y(x,t) + y_t(x,t) = \frac{1}{3}\,y_{xx}(x,t) + g(x,t), (x,t) \in [0,1] \times [0,1], \tag{97}$$

subject to IBCs:

$$y(x,0) = 0,\ y_t(x,0) = 0,\ y(0,t) = 0,\ y(1,t) = 0, \tag{98}$$

or DBCs:

$$y(x,0) = 0,\ y(x,1) = \sin(\pi\,x),\ y(0,t) = 0,\ y(1,t) = 0, \tag{99}$$

where $g(x,t)$ is selected such that the solution to (97) is $y(x,t) = t^3\,\sin(\pi\,x)$.

The application of GSJCOPMM to obtain approximated solutions to (97), subject to IBCs (98) or DBCs (99), respectively, using $N = 0, 1, 2, 5, 8, 12$, $\nu(x,t) = 2 - 0.3\,e^{-x\,t}$ and $\nu_j(x,t) = 2 - (0.1)(j+3)\,e^{-x\,t}$, $j = 1, 2, 3, 4$, gives acceptable accuracy, as shown in Table 2. Figure 3a,c demonstrates that the obtained solutions achieve an accuracy of 10^{-13} at $N = 12$, aligning with the approximate and exact solutions depicted in Figure 2. Additionally, we may obtain valuable insights into the developments gained by applying GSJCOPMM from the heat map graphs displayed in Figure 3b,d and supported by the error graphs in Figures 4a and 5a.

Problem 3. *Consider the following MTVO-TFDE equation [57,58]:*

$$\left({}_0^c\mathcal{D}_t^{\nu(x,t)} + \sum_{j=1}^{5} \frac{1}{j+1}\,{}_0^c\mathcal{D}_t^{\nu_j(x,t)}\right) y(x,t) + \frac{1}{2}\,y_t(x,t) = \frac{1}{3}\,y_{xx}(x,t) + g(x,t), (x,t) \in [0,1] \times [0,1], \tag{100}$$

subject to IBCs:

$$y(x,0) = \sin(x),\ y_t(x,0) = -\cos(x),\ y(0,t) = -\sin(t),\ y(1,t) = \sin(1-t), \tag{101}$$

or DBCs:

$$y(x,0) = \sin(x),\ y(x,1) = \sin(x-1),\ y(0,t) = -\sin(t),\ y(1,t) = \sin(1-t), \tag{102}$$

where $g(x,t)$ is selected such that the solution to (100) is $y(x,t) = \sin(x-t)$.

The application of GSJCOPMM to obtain approximated solutions to (100) subject to IBCs (101) or DBCs (102), respectively, using $N = 2,4,6,8,10,12$, $\nu(x,t) = 1.8 + 0.2\ \sin(x\,t)$ and $\nu_j(x,t) = 1.8 - (0.1)j + 0.2\ \sin(x\,t)$, $j = 1,2,3,4,5$, gives acceptable accuracy for the solutions obtained, as shown in Table 3 and Figure 7a,c. These solutions reach an agreement of an accuracy of 10^{-14} at $N = 12$, aligning with the approximate and exact solutions depicted in Figure 6. Again, the heat map graphs displayed in Figure 7b,d and the error graphs in Figures 8a and 9a confirm the effectiveness of GSJCOPMM.

Remark 6. *In view of the presented CPU time (in seconds), our approach has efficient performance. The calculations show that the memory consumption was excellent. For example, the calculated CPU time using $N = 10$ is 6% slower than $N = 7$ and, moreover, requires increasing by 20% the memory consumption of RAM compared to the $N = 7$ calculation. The numerical examples and comparisons provided in our paper highlight the superior accuracy and efficiency of our algorithm, solidifying its potential for solving MTVO-TFDEs effectively. When we compared the resource use of our method to that described in [57,58], we saw that those papers did not provide CPU time and memory usage. However, based on our analysis, our approach demonstrates better performance compared to the referenced methods.*

Problem 4. *Consider the following MTVO-TFDE equation:*

$$\left({}_0^c\mathcal{D}_t^{\nu(x,t)} + \sum_{j=1}^{2} {}_0^c\mathcal{D}_t^{\nu_j(x,t)}\right) y(x,t) + y_t(x,t) = y_{xx}(x,t) + g(x,t), (x,t) \in [0,1] \times [0,1], \tag{103}$$

subject to IBCs:

$$y(x,0) = E_{\frac{1}{2},0}(x) + 1,\ y_t(x,0) = \frac{2}{\sqrt{\pi}},\ y(0,t) = e^{t^2}(1 + erf(t)),\ y(1,t) = e^{t^2}(1 + erf(t)) + E_{\frac{1}{2},0}(1), \tag{104}$$

or DBCs:

$$y(x,0) = E_{\frac{1}{2},0}(x) + 1,\ y(x,1) = (1 + erf(1))e + E_{\frac{1}{2},0}(x),\ y(0,t) = e^{t^2}(1 + erf(t)),\ y(1,t) = e^{t^2}(1 + erf(t)) + E_{\frac{1}{2},0}(1), \tag{105}$$

where $g(x,t)$ is selected such that the solution of (103) is

$$y(x,t) = E_{\frac{1}{2},1}(t) + E_{\frac{1}{2},0}(x), \tag{106}$$

and where the functions

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt and E_{\alpha,\beta}(x) = \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(k\alpha + \beta)} \tag{107}$$

are the Gaussian error function and the generalized Mitta–Leffler function, respectively. The application of GSJCOPMM to obtain approximated solutions to (103) subject to IBCs (104) or DBCs (105), respectively, using $N = 2,4,6,8,10,14$, $\nu(x,t) = 1 + x\,t$ and $\nu_1(x,t) = 2 - x\,t$, $\nu_2(x,t) = \frac{3}{2} - \frac{1}{4}x\,t$, gives acceptable accuracy, as shown in Table 4, and error graphs as shown in Figure 10. These solutions reach an agreement of an accuracy of 10^{-9} at $N = 14$.

9. Conclusions

This study presents a generalized form of shifted JPs that fulfill homogeneous IBCs or DBCs. Then, by making use of the OMs derived in Sections 4 and 5 with the SCM, an

approximation algorithm for the given MTVO-TFDWE is established. Three examples, including MTVO-TFDWE (1), were tested using the suggested technique, *GSJCOPMM*, to prove its high accuracy and efficiency. The examples were subjected to either the IBCs (2) or the DBCs (3). It would be great to see our results generalized to other kinds of initial and boundary conditions. Studying the system's behavior in different settings would provide interesting insights and make our conclusions even more applicable. Additionally, the theoretical results obtained in this study can be developed to deal with more complex versions of MTVO-TFDWEs, such as multi-dimensional multi-term Caputo's time-fractional mixed sub-diffusion and diffusion-wave equations. Additionally, incorporating adaptive strategies and parallel computing techniques could further enhance the efficiency and scalability of the algorithm. Overall, this research contributes to the advancement of numerical methods for MTVO-TFDWEs and opens up avenues for exploring their applications in various fields.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The author is grateful to the reviewers and editor for their insightful comments and recommendations, which substantially enhanced this research.

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

Abbreviations

The following abbreviations are used in this manuscript:

Abbreviation	Description
DEs	Differential equations
FDEs	Fractional differential equations
VOFDEs	Variable-order fractional differential equations
MTVO	Multi-term variable-order
TFDWEs	Time-fractional diffusion-wave equations
MTVO-TFDWEs	Multi-term variable-order time-fractional diffusion-wave equations
IBCs	Initial boundary conditions
DBCs	Dirichlet boundary conditions
OMs	Operational matrices
Ods	Ordinary derivatives
VOFDs	Variable-order fractional derivatives
SCM	Spectral collocation method
VOFC	Variable-order fractional calculus
JPs	Jacobi polynomials
GSJPs	Generalized shifted Jacobi polynomials
GSJCOPMM	Generalized shifted Jacobi collocation operational matrix method
BPA	Best possible approximation
MAE	Maximum absolute error

References

1. Maayah, B.; Arqub, O.A.; Alnabulsi, S.; Alsulami, H. Numerical solutions and geometric attractors of a fractional model of the cancer-immune based on the Atangana–Baleanu–Caputo derivative and the reproducing kernel scheme. *Chinese J. Phys.* **2022**, *80*, 463–483. [CrossRef]
2. Berredjem, N.; Maayah, B.; Arqub, O.A. A numerical method for solving conformable fractional integrodifferential systems of second-order, two-points periodic boundary conditions. *Alex. Eng. J.* **2022**, *61*, 5699–5711. [CrossRef]
3. Arqub, O.A.; Rabah, A.B.; Momani, S. A spline construction scheme for numerically solving fractional Bagley–Torvik and Painlevé models correlating initial value problems concerning the Caputo–Fabrizio derivative approach. *Int. J. Mod. Phys. C* **2023**, *34*, 2350115. [CrossRef]
4. Almeida, R.; Tavares, D.; Torres, D.F.M. *The Variable-Order Fractional Calculus of Variations*; Springer: Cham, Switzerland, 2019.
5. Podlubny, I. *Fractional Differential Equations*; Academic Press: San Diego, CA, USA, 1999.

6. Kilbas, A.A.; Srivastava, H.M.; Trujillo, J.J. *Theory and Applications of Fractional Differential Equations*; Elsevier: Amsterdam, The Netherlands, 2006; Volume 204.
7. Mainardi, F. *Fractional Calculus and Waves in Linear Viscoelasticity*; Imperial College Press: London, UK, 2010.
8. Samko, S.G.; Ross, B. Integration and differentiation to a variable fractional order. *Integr. Transf. Spec. Funct.* **1993**, *1*, 277–300. [CrossRef]
9. Odzijewicz, T.; Malinowska, A.B.; Torres, D.F.M. Noether's theorem for fractional variational problems of variable order. *Cent. Eur. J. Phys.* **2013**, *11*, 691–701. [CrossRef]
10. Chen, S.; Liu, F.; Burrage, K. Numerical simulation of a new two-dimensional variable-order fractional percolation equation in non-homogeneous porous media. *Comput. Math. Appl.* **2014**, *67*, 1673–1681. [CrossRef]
11. Coimbra, C.F.M.; Soon, C.M.; Kobayashi, M.H. The variable viscoelasticity operator. *Ann. Phys.* **2005**, *14*, 378–389.
12. Odzijewicz, T.; Malinowska, A.B.; Torres, D.F.M. Fractional variational calculus of variable order. In *Advances in Harmonic Analysis and Operator Theory: Advances and Applications*; Almeida, A., Castro, L., Speck, F.O., Eds.; Birkhäuser: Basel, Switzerland, 2013; Volume 229, pp. 291–301.
13. Ostalczyk, P.W.; Duch, P.; Brzeziński, D.W.; Sankowski, D. Order functions selection in the variable-fractional-order PID controller. Advances in Modelling and Control of Non-integer-Order Systems. *Lect. Notes Electr. Eng.* **2015**, *320*, 159–170.
14. Rapaić, M.R.; Pisano, A. Variable-order fractional operators for adaptive order and parameter estimation. *IEEE Trans. Autom. Contr.* **2013**, *59*, 798–803. [CrossRef]
15. Izadi, M.; Yüzbasi, S.; Adel, W. Two novel Bessel matrix techniques to solve the squeezing flow problem between infinite parallel plates. *Comput. Math. Math. Phy.* **2021**, *61*, 2034–2053. [CrossRef]
16. Coimbra, C.F.M. Mechanics with variable-order differential operators. *AdP* **2003**, *515*, 692–703. [CrossRef]
17. Lin, R.; Liu, F.; Anh, V.; Turner, I. Stability and convergence of a new explicit finite-difference approximation for the variable-order nonlinear fractional diffusion equation. *Appl. Math. Comput.* **2009**, *212*, 435–445. [CrossRef]
18. Birajdar, G.A.; Rashidi, M.M. Finite Difference Schemes for Variable Order Time-Fractional First Initial Boundary Value Problems. *Appl. Appl. Math.* **2017**, *12*, 112–135.
19. Patnaik, S.; Semperlotti, F. Variable-order particle dynamics: Formulation and application to the simulation of edge dislocations. *Philos. Trans. R. Soc. A* **2020**, *378*, 0190290. [CrossRef] [PubMed]
20. Blaszczyk, T.; Bekus, K.; Szajek, K.; Sumelka, W. Approximation and application of the Riesz-caputo fractional derivative of variable order with fixed memory. *Meccanica* **2022**, *57*, 861–870. [CrossRef]
21. Abd-Elhameed, W.M.; Ahmed, H.M. Spectral solutions for the time-fractional heat differential equation through a novel unified sequence of Chebyshev polynomials. *AIMS Math.* **2024**, *9*, 2137–2166. [CrossRef]
22. Paola, M.D.; Alotta, G.; Burlon, A.; Failla, G. A novel approach to nonlinear variable-order fractional viscoelasticity. *Philos. Trans. R. Soc. A* **2020**, *378*, 20190296. [CrossRef] [PubMed]
23. Burlon, A.; Alotta, G.; Paola, M.D.; Failla, G. An original perspective on variable-order fractional operators for viscoelastic materials. *Meccanica* **2021**, *56*, 769–784. [CrossRef]
24. Ahmed, H.M. A new first finite class of classical orthogonal polynomials operational matrices: An application for solving fractional differential equations. *Contemp. Math.* **2023**, *4*, 974–994. [CrossRef]
25. Napoli, A.; Abd-Elhameed, W.M. An innovative harmonic numbers operational matrix method for solving initial value problems. *Calcolo* **2017**, *54*, 57–76. [CrossRef]
26. Izadi, M.; Yüzbasi, S.; Adel, W. A new Chelyshkov matrix method to solve linear and nonlinear fractional delay differential equations with error analysis. *Math. Sci.* **2023**, *17*, 267–284. [CrossRef]
27. Izadi, M.; Sene, N.; Adel, W.; El-Mesady, A. The Layla and Majnun mathematical model of fractional order: Stability analysis and numerical study. *Results Phys.* **2023**, *51*, 106650. [CrossRef]
28. Liu, J.; Li, X.; Wu, L. An operational matrix of fractional differentiation of the second kind of Chebyshev polynomial for solving multiterm variable order fractional differential equation. *Math. Probl. Eng.* **2016**, *2016*, 7126080. [CrossRef]
29. Youssri, Y.H.; Abd-Elhameed, W.M.; Ahmed, H.M. New fractional derivative expression of the shifted third-kind Chebyshev polynomials: Application to a type of nonlinear fractional pantograph differential equations. *J. Funct. Spaces* **2022**, *2022*, 3966135. [CrossRef]
30. Abd-Elhameed, W.M.; Alkenedri, A.M. New formulas for the repeated integrals of some Jacobi polynomials: Spectral solutions of even-order boundary value problems. *Int. J. Appl. Comput. Math.* **2021**, *7*, 166. [CrossRef]
31. Sheikhi, S.; Matinfar, M.; Firoozjaee, M.A. Numerical solution of variable-order differential equations via the Ritz-approximation method by shifted Legendre polynomials. *Int. J. Appl. Comput. Math.* **2021**, *7*, 22. [CrossRef]
32. El-Sayed, A.A.; Baleanu, D.; Agarwal, P. A novel Jacobi operational matrix for numerical solution of multi-term variable-order fractional differential equations. *J. Taibah Univ. Sci.* **2020**, *14*, 963–974. [CrossRef]
33. Nagy, A.M.; Sweilam, N.H.; El-Sayed, A.A. New operational matrix for solving multiterm variable order fractional differential equations. *J. Comp. Nonlinear Dyn.* **2018**, *13*, 011001–011007. [CrossRef]
34. El-Sayed, A.A.; Agarwal, P. Numerical solution of multiterm variable-order fractional differential equations via shifted Legendre polynomials. *Math. Meth. Appl. Sci.* **2019**, *42*, 3978–3991. [CrossRef]
35. Wang, L.F.; Ma, Y.P.; Yang, Y.Q. Legendre polynomials method for solving a class of variable order fractional differential equation. *CMES-Comp. Model. Eng.* **2014**, *101*, 97–111.

36. Chen, Y.M.; Wei, Y.Q.; Liu, D.Y.; Yu, H. Numerical solution for a class of nonlinear variable order fractional differential equations with Legendre wavelets. *Appl. Math. Lett.* **2015**, *46*, 83–88. [CrossRef]
37. Bushnaq, S.; Shah, K.; Tahir, S.; Ansari, K.J.; Sarwar, M.; Abdeljawad, T. Computation of numerical solutions to variable order fractional differential equations by using non-orthogonal basis. *AIMS Math.* **2022**, *7*, 10917–10938. [CrossRef]
38. Chen, Y.M.; Liu, L.Q.; Li, B.F.; Sun, Y. Numerical solution for the variable order linear cable equation with Bernstein polynomials. *Appl. Math. Comput.* **2014**, *238*, 329–341. [CrossRef]
39. Shen, S.; Liu, F.; Chen, J.; Turner, I.; Anh, V. Numerical techniques for the variable order time fractional diffusion equation. *Appl. Math. and Comput.* **2012**, *218*, 10861–10870. [CrossRef]
40. Moghaddam, B.P.; Machado, J.A.T. Extended algorithms for approximating variable order fractional derivatives with applications. *J. Sci. Comput.* **2017**, *71*, 1351–1374. [CrossRef]
41. Ahmed, H.M. Enhanced shifted Jacobi operational matrices of derivatives: Spectral algorithm for solving multiterm variable-order fractional differential equations. *Bound. Value Probl.* **2023**, *2023*, 108. [CrossRef]
42. Shen, J.; Tang, T.; Wang, L. *Spectral Methods: Algorithms, Analysis and Applications*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 41.
43. Abd-Elhameed, W.M.; Ahmed, H.M.; Youssri, Y.H. A new generalized Jacobi Galerkin operational matrix of derivatives: Two algorithms for solving fourth-order boundary value problems. *Adv. Differ. Equ.* **2016**, *2016*, 22. [CrossRef]
44. Abd-Elhameed, W.M.; Ahmed, H.M. Tau and Galerkin operational matrices of derivatives for treating singular and Emden-Fowler third-order-type equations. *Int. J. Mod. Phys. C* **2022**, *33*, 2250061. [CrossRef]
45. Heydari, M.H.; Hooshmandasl, M.R.; Ghaini, F.M.M.; Cattani, C. Wavelets method for the time fractional diffusion-wave equation. *Phys. Lett. A* **2015**, *379*, 71–76. [CrossRef]
46. Jiang, H.; Liu, F.; Turner, I.; Burrage, K. Analytical solutions for the multi-term time-fractional diffusion-wave/diffusion equations in a finite domain. *Comput. Math. Appl.* **2012**, *64*, 3377–3388. [CrossRef]
47. Nigmatullin, R.R. To the theoretical explanation of the universal response. *Phys. Status Solidi (B) Basic Res.* **1984**, *123*, 739–745. [CrossRef]
48. Nigmatullin, R.R. Realization of the generalized transfer equation in a medium with fractal geometry, phys. status solidi, b basic res. *Phys. Status Solidi (B) Basic Res.* **1986**, *133*, 425–430. [CrossRef]
49. Luchko, Y. Initial-boundary-value problems for the generalized multi-term time-fractional diffusion equation. *J. Math. Anal. Appl.* **2011**, *374*, 538–548. [CrossRef]
50. Liu, Y.; Yamamoto, M. Uniqueness of orders and parameters in multi-term time-fractional diffusion equations by short-time behavior. *Inverse Probl.* **2022**, *39*, 024003. [CrossRef]
51. Cheng, J.; Nakagawa, J.; Yamamoto, M.; Yamazaki, T. Uniqueness in an inverse problem for a one dimensional fractional diffusion equation. *Inverse Probl.* **2009**, *25*, 115002. [CrossRef]
52. Szegö, G. *Orthogonal Polynomials*, 4th ed.; American Mathematical Soc.: Providence, RI, USA, 1975; Volume XXIII.
53. Luke, Y.L. *Mathematical Functions and Their Approximations*; Academic Press: London, UK, 1975.
54. Prudnikov, A.P.; Brychkov, Y.A.; Marichev, O.I. *More Special Functions*; Integrals and Series; Gordon and Breach: New York, NY, USA, 1990; Volume 3.
55. Narumi, S. Some formulas in the theory of interpolation of many independent variables. *Tohoku Math. J.* **1920**, *18*, 309–321.
56. Jeffrey, A.; Dai, H.H. *Handbook of Mathematical Formulas and Integrals*, 4th ed.; Elsevier: Amsterdam, The Netherlands, 2008.
57. Heydari, M.H.; Avazzadeh, Z.; Haromi, M.F. A wavelet approach for solving multi-term variable-order time fractional diffusion-wave equation. *Appl. Math. Comput.* **2019**, *341*, 215–228. [CrossRef]
58. Sadri, K.; Aminikhah, H. A new efficient algorithm based on fifth-kind Chebyshev polynomials for solving multi-term variable-order time-fractional diffusion-wave equation. *Int. J. Comput. Math.* **2022**, *99*, 966–992. [CrossRef]