Advanced Methods for Conformable Time-Fractional Differential Equations: Logarithmic Non-Polynomial Splines

Majeed A. Yousif 1, Ravi P. Agarwal 2, Pshtiwan Othman Mohammed 3,4,*, Alina Alb Lupas 5,*, Rashid Jan 6,7 and Nejmeddine Chorfi 8

1 Department of Mathematics, College of Education, University of Zakho, Duhok 42001, Iraq; majeed.yousif@uoz.edu.krd
2 Department of Mathematics and Systems Engineering, Florida Institute of Technology, Melbourne, FL 32901, USA
3 Department of Mathematics, College of Education, University of Sulaimani, Sulaimani 46001, Iraq
4 Research and Development Center, University of Sulaimani, Sulaymaniyah 46001, Iraq
5 Department of Mathematics and Computer Science, University of Oradea, 410087 Oradea, Romania
6 Institute of Energy Infrastructure (IEI), Department of Civil Engineering, College of Engineering, Universiti Tenaga Nasional (UNITEN), Putrajaya Campus, Jalan IKRAM-UNITEN, Kajang 43000, Malaysia
7 Mathematics Research Center, King Saud University, P.O. Box 2455, Riyadh 11451, Saudi Arabia
8 Department of Mathematics, College of Science, King Saud University, P.O. Box 2455, Riyadh 11451, Saudi Arabia

* Correspondence: pshtiwan.muhammad@univsul.edu.iq (P.O.M.); dalb@uoradea.ro (A.A.L.)

Abstract: In this study, we present a numerical method named the logarithmic non-polynomial spline method. This method combines conformable derivative, finite difference, and non-polynomial spline techniques to solve the nonlinear inhomogeneous time-fractional Burgers–Huxley equation. The developed numerical scheme is characterized by a sixth-order convergence and conditional stability. The accuracy of the method is demonstrated with 3D mesh plots, while the effects of time and fractional order are shown in 2D plots. Comparative evaluations with the cubic B-spline collocation method are provided. To illustrate the suitability and effectiveness of the proposed method, two examples are tested, with the results are evaluated using $L_2$ and $L_{\infty}$ norms.

Keywords: conformable derivative; natural Logarithmic non-polynomial splines; time-fractional Burgers–Huxley equation; stability analysis

MSC: 26A33; 65D07; 35K55; 65M12

1. Introduction

The theory of fractional calculus, which deals with integrals and derivatives of arbitrary orders, intersects with the field of machine learning, providing a robust framework for addressing complex engineering challenges [1,2]. Fractional differential equations (FDEs) are pivotal across a spectrum of scientific and engineering disciplines, including dynamical systems, control engineering, signal processing, and more [3–7]. These equations play a crucial role in modeling evolutionary processes due to their ability to capture infinitesimal generative behaviors. Various formulations of fractional derivatives, such as Riemann–Liouville, Grunewald–Letnikov, Hadamard, Weyl, Caputo, $\psi$-Caputo and conformable derivatives, offer diverse tools for mathematical modeling and analysis [8–12].

Recently, there has been growing interest in the numerical solutions of FDEs, driven by their significance in modeling complex phenomena. This discussion explores pioneering works that have made significant contributions to this field. Examples include the use of B-spline functions to solve time-fractional gas dynamics equations [13], the application of the extended tanh-function approach to handle time-fractional Klein–Gordon and Sine-Gordon equations [14], and the utilization of non-polynomial spline methods for solving time-fractional nonlinear Schrödinger equations [15]. Other notable contributions involve
the resolution of the time-fractional KdV equation using fractional non-polynomial spline methods [16], spectral collocation methods with Legendre polynomials for multi-space fractional order equations like Kuramoto–Sivashinsky and KdV [17], and the finite element method for time-fractional stochastic partial differential equations [18]. Additionally, numerical methods have been applied in works such as [19–21] to solve FDEs, marking significant advancements in our ability to effectively tackle these equations numerically.

These studies collectively advance our understanding and capability to solve fractional differential equations (FDEs), paving the way for broader applications across various scientific and engineering domains. The introduction of the logarithmic non-polynomial spline method, with its high convergence rate and conditional stability, represents a significant step forward. By combining conformable derivatives with finite difference and non-polynomial spline techniques, this method offers a robust tool for tackling complex nonlinear time-fractional equations, enhancing both accuracy and computational efficiency.

This paper presents a generalized version of the time-fractional Burgers–Huxley (TFBH) equation, incorporating conformable fractional-order derivatives:

\[ T^α_{t} \Psi - c_1 \frac{\partial^2 \Psi}{\partial \zeta^2} + c_2 \Psi^p \frac{\partial \Psi}{\partial \zeta} - c_3 \Psi (1 - \Psi^p)(\Psi^p - k) = f(\zeta, t), \quad a \in (0, 1), \quad \zeta, t \geq 0. \quad (1) \]

Considering the following initial and boundary conditions,

\[ \Psi(\zeta, 0) = g(\zeta), \quad \Psi(a, t) = g_1(t), \quad \Psi(b, t) = g_2(t). \quad (2) \]

The conformable time-fractional derivative \( T^α_{t} \Psi(t) \) for \( \Psi : [0, \infty) \rightarrow \mathbb{R} \), as defined in reference [12], is given by:

\[ T^α_{t} \Psi(t) = \lim_{\rho \rightarrow \infty} \frac{\Psi(t + \rho t^{1-\alpha}) - \Psi(t)}{\rho}, \quad (3) \]

and \( c_1, c_2 \) and \( c_3 \) are diffusion, advection and reaction coefficients, respectively, and \( \rho \) is a positive integer number. When \( c_2 = 0 \) and \( c_1 = \rho = 1 \), Equation (1) transforms into the time-fractional Huxley equation, which is significant for describing phenomena such as wall motion in liquid crystals and the propagation of nerve pulses [22]. Similarly, with \( c_3 = 0 \) and \( c_1 = \rho = 1 \), it becomes the time-fractional Burger’s equation, which is important for explaining the far field of wave propagation in nonlinear systems [22]. When \( c_2 = 0 \) and \( c_1 = c_3 = \rho = 1 \), Equation (1) is known as the time-fractional FitzHugh–Nagumo equation [23,24]. Lastly, with \( \rho = 1 \) and \( c_1, c_2, c_3 \neq 0 \), Equation (1) is the time-fractional Burgers–Huxley equation [25].

Given the significance of the TFBH equation, numerous researchers have investigated it through various numerical and analytical approaches. For instance, the cubic B-spline collocation method was employed in [25], while the first integral method was utilized in [26]. Further contributions include the linearly semi-implicit compact scheme presented in [27], and the monotone finite difference method explored in [28]. Additionally, the finite difference collocation method was applied in [29], and compact operators coupled with Newton’s methods were investigated in [30]. Other notable techniques encompass the nonstandard finite difference method [31], the power series method [32], and the use of tension B-spline functions [33]. These diverse methodologies reflect the extensive research and development aimed at effectively solving the TFBH equation, highlighting its crucial role in the study of fractional differential equations.

The motivation for this work stems from the critical importance of accurately solving the time-fractional Burgers–Huxley equation, which appears in various engineering and scientific applications. Despite the extensive amounts of research using methods such as cubic B-spline collocation, the first integral method, and semi-implicit compact schemes, there remains a need for more efficient and higher-order accurate numerical methods. The primary aim of this research is to create a reliable and precise numerical technique, known as the logarithmic non-polynomial spline method (LNPSM), to solve the TFBH
Axioms 2024, 13, 551

The novelty of this work lies in the combination of conformable derivatives, finite difference techniques, and non-polynomial splines to achieve a sixth-order convergence and conditional stability. This approach not only enhances the accuracy of the solution but also provides a versatile framework that can be applied to other complex fractional differential equations. Additionally, the comparative analysis with existing methods, and the investigation of the effects of time and fractional order through detailed plots further underscore the effectiveness and applicability of the proposed method.

In the following sections, we explore the details of our proposed conformable LNPSM. Section 2 offers an in-depth explanation of the construction of logarithmic non-polynomial splines and their application in numerically solving the TFBH equation. Section 3 delves into the analysis of truncation errors for the conformable LNPSM, providing insights into the method’s accuracy and convergence characteristics. In Section 4, we demonstrate the practical application of the conformable LNPSM to the TFBH equation, highlighting its effectiveness through various examples. Section 5 is dedicated to a thorough stability analysis of the numerical scheme, evaluating its robustness under different conditions. In Section 6, we present numerical results and discuss their implications, offering a detailed assessment of the conformable LNPSM’s performance. Finally, Section 7 concludes our study with a summary of findings and reflections on the significance of our work. These sections collectively aim to showcase the innovation, efficacy, and potential impact of the conformable LNPSM in enhancing numerical solutions for the TFBH equation.

2. Logarithmic Non-Polynomial Spline (LNPS) Construct

This section outlines the construction of a non-polynomial spline enhanced with natural logarithmic functions. These splines are essential tools for numerical methods designed to solve equations with complex behaviors or intricate features. Here, we establish the framework for spline construction, setting the stage for subsequent detailed analysis and implementation.

Both spatial and temporal domains are discretized uniformly. Spatial points are indexed as $\zeta_j = jh$ for $j = 0, 1, \ldots, M$, while temporal points are indexed as $t_n = n\tau$ for $n = 0, 1, \ldots, N$. In this context, $h = \frac{b-a}{M}$ denotes the uniform spatial step size, and $\tau = \frac{T}{N}$ represents the uniform temporal step size. This uniform partitioning strategy simplifies the handling of evenly spaced data.

$$L_{j,n}(\zeta_j, t_n) = a^n_j \log(\tau (\zeta_j - \zeta_j) + 1) + b^n_j (\zeta_j - \zeta_j)^2 + c^n_j (\zeta_j - \zeta_j) + d^n_j.$$  \hspace{1cm} (4)

Consider the non-polynomial spline function $L_{j,n}(\zeta_j, t_n)$, which serves as an approximate solution with $\Psi^n_j = \Psi(\zeta_j, t_n)$. In this context, $\tau$ represents the frequency parameter for logarithmic functions. Initially, the coefficients $a^n_j$, $b^n_j$, $c^n_j$, and $d^n_j$ are not determined.

These coefficients are determined based on specific conditions imposed on the spline functions. These conditions typically involve ensuring continuity or the matching of function values and derivatives at certain points in both space and time.

$$L_{j,n}(\zeta_j, t_n) = \Psi^n_j,$$  \hspace{1cm} $L_{j,n}(\zeta_{j+1}, t_n) = \Psi^n_{j+1},$$

$$L_{j,n}^{(2)}(\zeta_j, t_n) = M^n_j,$$  \hspace{1cm} $L_{j,n}^{(2)}(\zeta_{j+1}, t_n) = M^n_{j+1}.$  \hspace{1cm} (5)

By applying the conditions outlined in (5) and utilizing the expression provided in Equation (4), we can derive the following:
\begin{align*}
  a^n_j &= -\frac{(\Delta + 1)^2 \left(M^n_j - M^n_{j+1}\right)}{\Delta(\Delta + 2)\tau^2}, \\
  b^n_j &= \frac{\Delta^2 M^n_{j+1} + 2\Delta M^n_j - M^n_j + M^n_{j+1}}{2\Delta(\Delta + 2)}, \\
  c^n_j &= \frac{M^n_j (h^2 \tau^2 + 2(\Delta + 1)^2 \log(\Delta + 1)) - (\Delta + 1)^2 M^n_{j+1} (h^2 \tau^2 + 2 \log(\Delta + 1))}{2h\Delta(\Delta + 2)\tau^2}, \\
  d^n_j &= \Psi^n_j.
\end{align*}

Expressing the relationship \( \Delta = h\tau \) through the continuity equation \( \frac{\partial j}{\partial t} = \frac{\partial}{\partial x} \left( \frac{\partial j}{\partial x} \right) \) entails a direct connection between the spatial and temporal variables is obtained as follows:

\begin{align}
  \tau a^n_j + c^n_j &= \frac{\tau a^n_{j-1}}{\Delta + 1} + 2hb^n_{j-1} + c^n_{j-1}.
\end{align}

Substituting the derived variables from (6) to (9) into Equation (10), and following simplification and grouping, the resulting expression is:

\begin{align}
  \Psi^n_{j-1} - 2\Psi^n_j + \Psi^n_{j+1} &= h \left( \beta_1 M^n_{j-1} + \beta_2 M^n_j + \beta_3 M^n_{j+1} \right),
\end{align}

where

\begin{align*}
  \beta_1 &= \frac{(\Delta(3\Delta + 2) - 2(\Delta + 1)^2 \log(\Delta + 1))}{2\Delta^2(\Delta + 2)\tau}, \\
  \beta_2 &= \frac{4(\Delta + 1)^2 \log(\Delta + 1) - \Delta(\Delta + 2)(\Delta(\Delta + 2) + 2)}{2\Delta^2(\Delta + 2)\tau}, \\
  \beta_3 &= \frac{(\Delta + 1)^2 ((\Delta - 2)\Delta + 2 \log(\Delta + 1))}{2\Delta^2(\Delta + 2)\tau}.
\end{align*}

### 3. Truncation Error

This section delves into the local truncation error associated with scheme (11) at the \( j \)-th step. To assess the accuracy and stability of the scheme, we utilize the Taylor expansion to determine the unknown values \( \beta_j \) for \( j = 1, 2, 3 \). This approach enables us to derive expressions for the local truncation error, which in turn helps in evaluating the performance of the numerical scheme. By analyzing these errors, we can gain insights into the error behavior and the effectiveness of the scheme in approximating the solution to the differential equation.

\begin{align*}
  T_j &= \Psi^n_{j-1} - 2\Psi^n_j + \Psi^n_{j+1} - h \left( \beta_1 M^n_{j-1} + \beta_2 M^n_j + \beta_3 M^n_{j+1} \right) \\
  &= \Psi^n_{j-1} - 2\Psi^n_j + \Psi^n_{j+1} - h\beta_1 M^n_{j-1} - h\beta_2 M^n_j - h\beta_3 M^n_{j+1}.
\end{align*}

Employing Taylor expansion and gathering the coefficients of derivatives, we have:

\begin{align}
  T_j &= \left( 2 - \frac{\beta_1}{h} - \frac{\beta_2}{h} - \frac{\beta_3}{h} \right) h^2 \Psi^{(2)}_j + \left( \frac{\beta_1}{h} - \frac{\beta_3}{h} \right) h^3 \Psi^{(3)}_j \\
  &\quad + \frac{1}{6} \left( -\frac{\beta_1}{2h} + \frac{\beta_2}{2h} \right) h^4 \Psi^{(4)}_j + \left( \frac{\beta_1}{6h} - \frac{\beta_3}{6h} \right) h^5 \Psi^{(5)}_j \\
  &\quad + \frac{1}{180} \left( -\frac{\beta_1}{24h} + \frac{\beta_3}{24h} \right) h^6 \Psi^{(6)}_j + \cdots
\end{align}
By considering Equation (13) and setting the coefficients of $\Psi_n^\rho$ equal to each other for $\rho = 2, 3, 4$, after equating the coefficients using Taylor series expansion, we obtained three linear equations, which allow us to solve for the unknowns systematically. These equations are crucial for determining the precise relationship between the variables in our model. We have the following matrices:

\[
\begin{bmatrix}
-\frac{1}{h} & -\frac{1}{h} & -\frac{1}{h} & 0 \\
\frac{1}{2} & 0 & -\frac{1}{h} & 0 \\
0 & -\frac{1}{2} & 0 & -\frac{1}{h}
\end{bmatrix}
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\beta_3
\end{bmatrix}
= 
\begin{bmatrix}
-2 \\
0 \\
-\frac{1}{6}
\end{bmatrix}.
\] (14)

Using the Gaussian elimination method for solving the system (14), we obtain

\[
\beta_1 = \frac{h}{6}, \quad \beta_2 = \frac{10h}{6}, \quad \beta_3 = \frac{h}{6}.
\]

After replacing the coefficients, the local truncation error can be described as:

\[
T_j = -\frac{1}{120} h^6 \Psi_n^{(6)}(t) + O(\tau^{2-\alpha} + h^6),
\]

and the equation represented by (11) can be formulated as follows:

\[
\Psi_n^j - 2\Psi_n^{j+1} + \Psi_n^{j+1} = h^2 \left( M_{j-1}^n + 10M_j^n + M_{j+1}^n \right).
\] (15)

4. Application of Conformable LNPSM to Time-Fractional Coupled TFBH Equation

This section presents a pioneering approach for solving the time-fractional Burgers–Huxley equation by integrating the logarithmic non-polynomial spline method with conformable derivatives. The method begins with a thorough examination of the properties of conformable derivatives, which are then adeptly combined with finite difference techniques and logarithmic non-polynomial splines. This fusion of methods creates a robust framework for addressing the TFBH equation, offering enhanced accuracy and computational efficiency. The novel approach not only improves the solution process but also opens new avenues for applying advanced numerical methods to complex differential equations.

Lemma 1 ([12]). Assume $\alpha \in (0, 1]$ and that both $\Psi$ and $\Phi$ are $\alpha$-derivative differentiable at point $t > 0$. In this case,

- $T_t^\alpha (A\Psi + B\Phi) = AT_t^\alpha \Psi + BT_t^\alpha \Phi$ for $A, B \in \mathbb{R}$.
- $T_t^\alpha (t^A) = At^{\alpha-\alpha}$ for all $A \in \mathbb{R}$.
- $T_t^\alpha \Psi(t) = 0$ if $\Psi(t)$ is a constant function.
- $T_t^\alpha (\Psi\Phi) = \Psi(T_t^\alpha \Phi) + \Phi(T_t^\alpha \Psi)$.
- $T_t^\alpha (\frac{\partial \Psi}{\partial \varphi}) = \frac{\Phi(T_t^\alpha \Psi) - \Psi(T_t^\alpha \Phi)}{\varphi}$.
- $T_t^\alpha \Psi(t) = t^{1-\alpha} \frac{\partial \Psi(t)}{\partial t}$, if $\Psi(t)$ is differentiable.

Corollary 1. Let $\tau$ denote the temporal step size within the context of the finite difference scheme. The terms $\frac{\partial \Psi}{\partial \varphi}$ are defined as:

\[
\frac{\partial \Psi}{\partial \varphi} \approx \frac{\Psi_{j+1}^n - \Psi_j^n}{\tau}, \text{ where } \Psi(\zeta, t) = \Psi_j^n,
\] (16)
Using the last property of Lemma 1 and Equation (16), then

\[ T^a_i \Psi(x, t) \approx t^{1-a} \frac{\partial \Psi}{\partial t} = \omega \frac{\Psi^{n+1} - \Psi^n}{\tau}, \quad \text{where} \quad \omega = t^{1-a}. \]  

(17)

By expressing \( M^n_j \) in the form derived from Equation (1), and using Equation (17), we have:

\[ M^n_j = \frac{1}{c_1} \omega \frac{\Psi^{n+1}_j - \Psi^n_j}{\tau} + \frac{c_2}{c_1} (\Psi^n_j)^{\rho} \frac{\Psi^n_{i+1} - \Psi^n_i}{h} \]

\[ - \frac{c_3}{c_1} \Psi^n_j (1 - (\Psi^n_j)^{\rho}) ((\Psi^n_j)^{\rho} - k) - \frac{1}{c_1} f^n_{j}, \]

(18)

Replacing \( j - 1 \) and \( j + 1 \) in Equation (18) yields:

\[ M^n_{j-1} = \frac{1}{c_1} \omega \frac{\Psi^{n+1}_{j-1} - \Psi^n_{j-1}}{\tau} + \frac{c_2}{c_1} (\Psi^n_{j-1})^{\rho} \frac{\Psi^n_{j+1} - \Psi^n_j}{h} \]

\[ - \frac{c_3}{c_1} \Psi^n_{j-1} (1 - (\Psi^n_{j-1})^{\rho}) ((\Psi^n_{j-1})^{\rho} - k) - \frac{1}{c_1} f^n_{j-1}, \]

(19)

\[ M^n_{j+1} = \frac{1}{c_1} \omega \frac{\Psi^{n+1}_{j+1} - \Psi^n_{j+1}}{\tau} + \frac{c_2}{c_1} (\Psi^n_{j+1})^{\rho} \frac{\Psi^n_{j+2} - \Psi^n_{j+1}}{h} \]

\[ - \frac{c_3}{c_1} \Psi^n_{j+1} (1 - (\Psi^n_{j+1})^{\rho}) ((\Psi^n_{j+1})^{\rho} - k) - \frac{1}{c_1} f^n_{j+1}, \]

(20)

Equation (15) can be represented through Equation (18), leading to

\[ \frac{1}{h^2} \left( \Psi^n_{j-1} - 2 \Psi^n_j + \Psi^n_{j+1} \right) \]

\[ = \left( \frac{1}{c_1} \omega \frac{\Psi^{n+1}_{j-1} - \Psi^n_{j-1}}{\tau} + \frac{c_2}{c_1} (\Psi^n_{j-1})^{\rho} \frac{\Psi^n_{j+1} - \Psi^n_j}{h} \right) \]

\[ - \frac{c_3}{c_1} \Psi^n_{j-1} (1 - (\Psi^n_{j-1})^{\rho}) ((\Psi^n_{j-1})^{\rho} - k) - \frac{1}{c_1} f^n_{j-1} \]

\[ + 10 \left( \frac{1}{c_1} \omega \frac{\Psi^{n+1}_j - \Psi^n_j}{\tau} + \frac{c_2}{c_1} (\Psi^n_j)^{\rho} \frac{\Psi^n_{i+1} - \Psi^n_i}{h} \right) \]

\[ - \frac{c_3}{c_1} \Psi^n_j (1 - (\Psi^n_j)^{\rho}) ((\Psi^n_j)^{\rho} - k) - \frac{1}{c_1} f^n_j \]

\[ + \left( \frac{1}{c_1} \omega \frac{\Psi^{n+1}_{j+1} - \Psi^n_{j+1}}{\tau} + \frac{c_2}{c_1} (\Psi^n_{j+1})^{\rho} \frac{\Psi^n_{j+2} - \Psi^n_{j+1}}{h} \right) \]

\[ - \frac{c_3}{c_1} \Psi^n_{j+1} (1 - (\Psi^n_{j+1})^{\rho}) ((\Psi^n_{j+1})^{\rho} - k) - \frac{1}{c_1} f^n_{j+1} \). \]

(21)

After some simplification and collection in Equation (21), we obtain

\[ \delta \Psi^{n+1}_{i-1} + 10 \delta \Psi^{n+1}_i + \delta \Psi^{n+1}_{i+1} = A_i \Psi^n_{i-1} + B_i \Psi^n_i + C_i \Psi^n_{i+1} + D_i \Psi^n_{i+2} \]

\[ - \frac{1}{c_1} f^n_{j-1} - \frac{10}{c_1} f^n_{j} - \frac{1}{c_1} f^n_{j+1}, \]

(22)

where
\[ \delta = -\frac{\omega^2}{\tau c_1}, \]
\[ A_j = -\frac{1}{h^2} - \frac{\omega}{\tau c_1} \left( \frac{p_{j-1}^n}{h^2} \right) - \frac{c_2}{h c_1} \left( \frac{p_{j-1}^{n+1}}{h^2} \right) + \frac{c_3}{c_1} \left( \frac{p_{j-1}^n}{h^2} \right) \]
\[ B_j = \frac{2}{h^2} + \frac{c_2}{h c_1} \left( \frac{p_{j-1}^n}{h^2} \right) - \frac{10 \omega}{\tau c_1} \left( \frac{p_{j}^{n+1}}{h^2} \right) - \frac{10 c_3}{h c_1} \left( \frac{p_{j}^n}{h^2} \right) + \frac{10 k c_3}{c_1} \left( \frac{n_{j+1}^n}{h^2} \right) - \frac{10 c_3}{c_1} \left( \frac{p_{j+1}^n}{h^2} \right)^2 \]
\[ C_j = -\frac{1}{h^2} + \frac{10 c_2}{h c_1} \left( \frac{p_{j-1}^n}{h^2} \right) - \frac{10 \omega}{\tau c_1} \left( \frac{p_{j}^{n+1}}{h^2} \right) + \frac{c_3}{c_1} \left( \frac{p_{j-1}^n}{h^2} \right) + \frac{k c_3}{c_1} \left( \frac{n_{j+1}^n}{h^2} \right) + \frac{c_3}{c_1} \left( \frac{p_{j+1}^n}{h^2} \right)^2 \]
\[ D_j = \frac{c_2}{h c_1} \left( \frac{p_{j+1}^n}{h^2} \right)^2. \]

Equation (22) comprises \( n - 1 \) equations and \( n + 1 \) unknowns. To resolve this imbalance, two additional equations are required, derived from the initial and boundary conditions.

5. Stability Analysis of Conformable LNPSM

In this section, the stability of the obtained numerical scheme, denoted as (22), aimed at solving the TFHB equation, is scrutinized. Utilizing the Fourier stability principle, an in-depth analysis is conducted to understand how this scheme behaves under diverse conditions, providing valuable insights into its effectiveness in accurately approximating the solution of the TFHB equation. According to Fourier stability analysis, leading to

\[ \Psi_j^n = \mu^n e^{i \Omega j}, \tag{23} \]

In this context, \( \Omega \) represents the actual spatial wave number, and \( i \) denotes the imaginary unit, defined as \( i^2 = -1 \). By linearizing the nonlinear term and substituting the expression from Equation (23) into Equation (22), we obtain:

\[ \delta \mu^{n+1} e^{i \Omega (j-1)} + 10 \delta \mu^{n+1} e^{i \Omega j} + \delta \mu^{n+1} e^{i \Omega (j+1)} = A_j \mu^n e^{i \Omega (j-1)} + B_j \mu^n e^{i \Omega j} + C_j \mu^n e^{i \Omega (j+1)} + D_j \mu^n e^{i \Omega (j+2)}. \tag{24} \]

Dividing both sides of Equation (24) by \( \mu^n e^{i \Omega j} \), and with some simplification, we obtain

\[ \mu = \frac{A_j e^{-i \Omega j} + B_j e^{i \Omega j} + C_j e^{2i \Omega j}}{\delta e^{-i \Omega j} + 10 \delta + \delta e^{i \Omega j}}. \tag{25} \]

After using Euler’s formula and some collections, we have

\[ \mu = \frac{S_1 + i S_2}{S_3}, \tag{26} \]

where

\[ S_1 = A_j \delta + B_j 10 \delta + D_j \delta + (B_j \delta + A_j 10 \delta + C_j 10 \delta - C_j \delta) \cos(\Omega h) \]
\[ - (B_j \delta - C_j \delta - D_j 10 \delta) \cos(2 \Omega h) + (A_j \delta + D_j \delta) \cos(3 \Omega h), \]
\[ S_2 = (-B_j \delta + A_j 10 \delta + C_j 10 \delta) \sin(\Omega h) - (B_j \delta + C_j \delta - D_j 10 \delta) \sin(2 \Omega h) \]
\[ + (C_j \delta - D_j \delta) \sin(3 \Omega h) + D_j \delta \sin(4 \Omega h), \]
\[ S_3 = 10 \delta^2 + (\delta^2 + 20 \delta^2) (2 \cos(\Omega h)^2 - 1) + (2 \delta^2 + 20 \delta^2) \cos(\Omega h) \]
\[ + \delta^2 \left( 2 (2 \cos(\Omega h)^2 - 1)^2 - 1 \right). \]
This implies that:

$$|\mu| = \sqrt{\left(\frac{S_1}{S_3}\right)^2 + \left(\frac{S_2}{S_3}\right)^2}.$$  (27)

Then, the numerical scheme (22) is stable if

$$\sqrt{\left(\frac{S_1}{S_3}\right)^2 + \left(\frac{S_2}{S_3}\right)^2} \leq 1.$$

According to the above stability condition, the obtained numerical scheme (22) is conditionally stable.

6. Numerical Results and Discussion

In this section, the efficiency and performance of the proposed conformable LNPSM are evaluated through two test problems. The approximate solutions generated by the LNPSM are compared with the exact solutions and results from previous studies using error norms as metrics. The figures and tables included in this study were meticulously generated using MATLAB software, which facilitated the comprehensive analysis and visualization of the results. This tool was instrumental in accurately depicting the data and supporting the findings presented in the paper. These error norms quantify the accuracy of the approximations. This detailed analysis highlights the strengths and limitations of conformable LNPSM, offering insights into its practical applicability and theoretical foundations.

$$L_\infty = \max_{1 \leq j \leq M} |u_{j,\text{exact}} - u_{j,\text{approximate}}|$$  (28)

and

$$L_2 = \sqrt{\sum_{i=1}^{M} (u_{i,\text{exact}} - u_{i,\text{approximate}})^2}.$$  (29)

Example 1. Considering the inhomogeneous TFBH equation,

$$T^\alpha \Psi - c_1 \frac{\partial^2 \Psi}{\partial \zeta^2} + c_2 \Psi \frac{\partial \Psi}{\partial \zeta} - c_3 \Psi (1 - \Psi^\rho) (\Psi^\rho - k) = f(\zeta, t), \quad \alpha \in (0, 1], \quad \zeta, t \geq 0.$$  (30)

Considering the following initial and boundary conditions,

$$\Psi(\zeta, 0) = t^\alpha, \quad \Psi(0, t) = 0, \quad \Psi(1, t) = 0,$$  (31)

with the source term

$$f(\zeta, t) = (1 - 2\zeta) \cos \frac{3\pi\zeta}{2} \Gamma(1 + \alpha) + t^\alpha \left(\frac{9\pi^2}{4} (1 - 2\zeta) \cos \frac{3\pi\zeta}{2} - 6\pi \sin \frac{3\pi\zeta}{2}\right)
- c_2 t^\alpha \left(\frac{3\pi\zeta}{2} \cos \frac{3\pi\zeta}{2} + 2(1 - 2\zeta) \sin \frac{3\pi\zeta}{2}\right)
- c_3 \left(1 - (1 - \zeta) t^\alpha \cos \frac{3\pi\zeta}{2} \right) \left(1 - (1 - \zeta) t^\alpha \cos \frac{3\pi\zeta}{2} - k\right).$$

The exact solution is

$$\Psi(\zeta, t) = \Phi(\zeta, t) = (1 - 2\zeta) t^\alpha \cos \frac{3\pi\zeta}{2}.$$

The 3D mesh plots shown in Figure 1a,b depict the comparison between the conformable LNPSM and the exact solution for Example 1, where $c_1 = c_3 = 1$, $c_2 = 0.01$, $k = 0.5$, and $\alpha = 0.5$. It is evident that the numerical scheme closely approximates the exact solution, demonstrating high accuracy across both spatial and temporal intervals. Figure 2a illustrates the influence of time $t$ on $\Psi(\zeta, t)$ for $t = 0.25:1$ with $\alpha = 0.5$. The plot shows that $\Psi(\zeta)$ increases within the range of $0 \leq x \leq 0.4$ as time progresses, while the behavior reverses for $0.4 < \zeta \leq 1.0$. 
Figure 1. The 3D Solution profiles for Example 1, where $c_1 = c_3 = 1$, $c_2 = 0.01$, $k = 0.75$ and $\alpha = 0.5$. (a) The conformable NPS solution. (b) The exact Solution.

Figure 2. Comparison between exact and numerical solutions for different values of time and fractional derivatives on $\Psi(\zeta)$ for Example 1, where $c_1 = c_3 = 1$, $c_2 = 0.01$, and $k = 0.5$. (a) Effect of time on $\Psi(\zeta)$, where $\alpha = 0.5$. (b) Effect of $\alpha$ on $\Psi(\zeta)$, where $t = 0.5$. (c) Absolute error between exact solution and LNPSM.

Figure 2b examines the impact of the fractional derivative $\alpha$ on $\Psi(\zeta)$ at $t = 0.5$. It demonstrates a decrease in $\Psi(\zeta)$ for $0 \leq \zeta \leq 0.4$ with increasing $\alpha$. Conversely, the trend reverses for $0.4 < \zeta \leq 1.0$. Figure 2c shows an absolute error comparison between the exact solution and the LNPSM. Table 1 presents a comparative analysis using error norms between our LNPSM and the cubic B-spline collocation method [25] for Example 1. Table 2 shows the experimental order convergent for Example 1 to present the order of the convergent effectively. The error norms provide quantitative insights into the method’s accuracy and effectiveness, showcasing its superiority in approximating the exact solution. These results collectively validate the robustness and applicability of the LNPSM in solving the nonlinear TFBH equation, offering significant advancements over existing numerical techniques.
The exact solution is $\alpha$ with an increasing

Considering the following initial and boundary conditions,

$$
T^{\alpha}\Psi - c_1\frac{\partial^2\Psi}{\partial\zeta^2} + c_2\Psi\frac{\partial\Psi}{\partial\zeta} - c_3\Psi(1 - \Psi^p)(\Psi^p - k) = f(\zeta, t), \quad \alpha \in (0, 1], \quad \zeta, t \geq 0. \tag{32}
$$

Considering the following initial and boundary conditions,

$$
\Psi(\zeta, 0) = 0, \quad \Psi(0, t) = 0, \quad \Psi(1, t) = 0, \tag{33}
$$

with the source term

$$
f(\zeta, t) = (\zeta - \zeta^2)^{\alpha+1}\frac{\Gamma(2\alpha + 1)}{\Gamma(1 + \alpha)} + c_2(\zeta - \zeta^2)(1 - 2\zeta)^{\alpha} + 2c_1t^{2\alpha} - c_3((\zeta - \zeta^2)^{\alpha+1})(1 - (\zeta - \zeta^2)^{2\alpha})(\zeta - \zeta^2)^{\alpha} - k).
$$

The exact solution is $\Psi(\zeta, t) = (\zeta - \zeta^2)^{\alpha}$.

Figure 3a,b display the conformable LNPSM and exact solution 3D mesh plots for Example 2, where $c_1 = c_3 = 1, c_2 = 0.01, k = 0.5$, and $\alpha = 0.5$. The figures clearly illustrate that the numerical scheme closely matches the exact solution, demonstrating high accuracy across both spatial and temporal intervals. Figure 4a illustrates the effect of time $t$ on $\Psi(\zeta, t)$ for $t = 0.25 : 1$ with $\alpha = 0.5$. It is evident that $\Psi(\zeta)$ increases uniformly as time progresses, reflecting the expected behavior over the specified time interval. Figure 4b examines the impact of the fractional derivative $\alpha$ on $\Psi(\zeta)$ at $t = 0.5$. The plot shows that $\Psi(\zeta)$ decreases with an increasing $\alpha$, consistent with the anticipated impact of fractional derivatives on the solution profile. Figure 4c shows an absolute error comparison between the exact solution and the LNPSM. Table 3 presents a comparative analysis using error norms between our LNPSM and the cubic B-spline collocation method [25] for Example 2. The error norms serve as a quantitative measure of the method’s accuracy and effectiveness, highlighting its superior performance in approximating the exact solution compared to existing numerical approaches. Table 4 shows the experimental order convergent for Example 2 to present the order of the convergent effectively. These findings underscore the robustness and applicability of the LNPSM in solving nonlinear TFH equations, demonstrating its potential to advance numerical solutions in fractional differential equations.
Figure 3. The 3D Solution profiles for Example 2, where $c_1 = c_3 = 1$, $c_2 = 0.01$, $k = 0.75$ and $\alpha = 0.5$. 
(a) The conformable NPS solution. (b) The exact Solution.

(a) 
(b)

Figure 4. Comparison between the exact and numerical solutions in different values of time and fractional derivatives on $\Psi(\zeta)$ for Example 2, where $c_1 = c_3 = 1$, $c_2 = 0.01$ and $k = 0.75$. (a) The effect of time on $\Psi(\zeta)$, where $\alpha = 0.5$. (b) The effect of $\alpha$ on $\Psi(\zeta)$, where $t = 0.5$. (c) The absolute error between the exact solution and the LNPSM.

Table 3. Comparison of error norms: LNPSM with cubic B-spline collocation method for Example 2, where $c_1 = c_3 = 1$, $c_2 = 0.01$, $k = 0.5$ and $\alpha = 0.5$.

<table>
<thead>
<tr>
<th>$\Psi(\zeta,t)$</th>
<th>LNPSM</th>
<th>[25]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>$L_2$</td>
<td>$L_\infty$</td>
</tr>
<tr>
<td>$0.5$</td>
<td>$2.6413 \times 10^{-8}$</td>
<td>$1.7632 \times 10^{-7}$</td>
</tr>
<tr>
<td>$0.75$</td>
<td>$4.7631 \times 10^{-9}$</td>
<td>$2.8721 \times 10^{-8}$</td>
</tr>
<tr>
<td>$1.0$</td>
<td>$1.4321 \times 10^{-7}$</td>
<td>$5.4412 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Table 4. Experimental order of convergence for Example 2.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$L_2$</th>
<th>EOC</th>
<th>$L_\infty$</th>
<th>EOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>$3.22 \times 10^{-7}$</td>
<td>$\ldots$</td>
<td>$4.53 \times 10^{-7}$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>100</td>
<td>$2.76 \times 10^{-8}$</td>
<td>5.9432</td>
<td>$3.61 \times 10^{-8}$</td>
<td>6.0011</td>
</tr>
<tr>
<td>200</td>
<td>$1.63 \times 10^{-9}$</td>
<td>5.7981</td>
<td>$2.49 \times 10^{-9}$</td>
<td>5.9012</td>
</tr>
</tbody>
</table>
7. Conclusions

In conclusion, the logarithmic non-polynomial spline method presented in this study effectively combines conformable derivative, finite difference, and non-polynomial spline techniques to solve the nonlinear inhomogeneous time-fractional Burgers–Huxley equation. The numerical scheme developed demonstrates impressive sixth-order convergence and conditional stability, which are essential for accurate and reliable computations. The method’s accuracy is validated through 3D mesh plots, highlighting its capability to handle complex dynamics influenced by both time and fractional-order variations, as depicted in 2D plots. Comparative analysis with the cubic B-spline collocation method underscores the advantages of our approach in terms of computational efficiency and solution accuracy. Two illustrative examples further confirm the method’s applicability, with $L_2$ and $L_\infty$ norms providing quantitative evidence of its suitability and effectiveness in practical scenarios of fractional differential equations.

The proposed numerical scheme based on the logarithmic non-polynomial spline method offers several key advantages:

- **High Accuracy**: The scheme achieves a truncation error with sixth-order convergence, ensuring high precision in numerical solutions.
- **Unconditional Stability**: The logarithmic non-polynomial spline method is proven to be unconditionally stable, making it reliable for a wide range of problems.
- **Superior Performance**: Comparative analysis using $L_2$ and $L_\infty$ norms demonstrates that the method outperforms both the cubic B-spline and Caputo non-polynomial methods.
- **Applicability**: The method is well-suited for solving conformable time-fractional equations, showing its versatility and effectiveness in complex scenarios.

**Author Contributions**: Conceptualization, M.A.Y.; data curation, A.A.L.; investigation, M.A.Y., P.O.M. and N.C.; methodology, A.A.L and R.J.; project administration, P.O.M.; resources, P.O.M.; software, R.P.A.; supervision, A.A.L.; validation, N.C.; visualization, R.P.A.; writing—original draft, M.A.Y. and R.P.A.; writing—review and editing, R.J. and N.C. All authors have read and agreed to the published version of the manuscript.

**Funding**: The publication of this research was supported by the University of Oradea, Romania.

**Institutional Review Board Statement**: Not applicable.

**Informed Consent Statement**: Not applicable.

**Data Availability Statement**: Data are contained within the article.

**Acknowledgments**: Researchers Supporting Project number (RSP2024R153), King Saud University, Riyadh, Saudi Arabia.

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

**Abbreviations**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDEs</td>
<td>Fractional Differential Equations</td>
</tr>
<tr>
<td>TFBH</td>
<td>Time-Fractional Burgers–Huxley</td>
</tr>
<tr>
<td>LNPSM</td>
<td>Logarithmic Non-Polynomial Spline Method</td>
</tr>
<tr>
<td>KdV</td>
<td>Korteweg–De Vries</td>
</tr>
</tbody>
</table>

**References**


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.