

Article

An Efficient Numerical Method for Solving a Class of Nonlinear Fractional Differential Equations and Error Estimates

Xin Song ^{1,*} and Rui Wu ²¹ School of Data Science and Engineering, South China Normal University, Shanwei 516600, China² School of Mathematics and Statistics, Changchun University of Technology, Changchun 130012, China; wurui@ccut.edu.cn

* Correspondence: songxin_math@m.scnu.edu.cn

Abstract: In this paper, we present an efficient method for solving a class of higher order fractional differential equations with general boundary conditions. The convergence of the numerical method is proved and an error estimate is given. Finally, eight numerical examples, both linear and nonlinear, are presented to demonstrate the accuracy of our method. The proposed method introduces suitable base functions to calculate the approximate solutions and only requires us to deal with the linear or nonlinear systems. Thus, our method is convenient to implement. Furthermore, the numerical results show that the proposed method performs better compared to the existing ones.

Keywords: fractional differential equations; approximate solutions; base functions

MSC: 65L10



Citation: Song, X.; Wu, R. An Efficient Numerical Method for Solving a Class of Nonlinear Fractional Differential Equations and Error Estimate.

Mathematics **2024**, *12*, 1824.

<https://doi.org/10.3390/math12121824>

Received: 27 May 2024

Revised: 9 June 2024

Accepted: 10 June 2024

Published: 12 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The study of nonlinear fractional differential equations has attracted a lot of attention in recent years due to their widespread applications in the natural sciences. The history of fractional differential equations can be traced back to three hundred years ago. Leibniz and Euler first studied such equations. Then L'Hospital, Laplace, and Caputo made great contributions to fractional differential equation theory. In recent years, more and more researchers have devoted themselves to working on such equations. The application and theory of fractional equations are developing rapidly. The reader may refer to [1–5] for recent developments in fractional differential equations.

In this paper, we study the numerical method for solving nonlinear fractional BVPs:

$$p_0(x)u^{(\alpha_0)}(x) + p_1(x)u^{(\alpha_1)}(x) + \cdots + p_{m-1}(x)u^{(\alpha_{m-1})}(x) + p_m(x)u^{(\alpha_m)}(x) = f(x, u), \quad (1)$$
$$\lambda_i u = r_i, \quad (i = 1, 2, \dots, N),$$

where $0 \leq x \leq 1$, $m_j - 1 < \alpha_j < m_j$, m_j is a positive integer, $m = \max\{m_j\}$. $p_j(x) \in L^2[0, 1]$, ($j = 0, 1, 2, \dots, m$). $\lambda_i u$ ($i = 1, 2, \dots, N$) are boundary conditions. If $\lambda_i u = \int_0^1 k_i(t)u(t)dt$ ($i = 1, 2, \dots, N$), problem (1) is a nonlocal problem. If $\lambda_i u = u(x_i)$ ($i = 1, 2, \dots, N$), problem (1) is a multi-point problem, and so on. The α -order Caputo derivative of $u(x)$ is defined in Definition 1.

Definition 1. The Caputo fractional derivative of order $\alpha > 0$ is defined as

$$D^\alpha u(x) = \frac{1}{\Gamma(N - \alpha)} \int_0^x (x - t)^{N - \alpha - 1} u^{(N)}(t) dt \quad (2)$$

where $0 < x < 1$, $N - 1 < \alpha < N$, and $N \in \mathbb{N}$.

Based on Definition 1, the problem is equivalent to the following problem:

$$Lu \triangleq \sum_{j=0}^m \frac{p_j(x)}{\Gamma(m_j - \alpha_j)} \int_0^x (x-t)^{m_j - \alpha_j - 1} u^{(m_j)}(t) dt = f(x, u), \quad x \in [0, 1], \quad (3)$$

$$\lambda_i u = r_i, \quad (i = 1, 2, \dots, N),$$

where $m_j - 1 < \alpha_j < m_j$, $m = \max\{m_j\}$. Over the last couple of years, many numerical methods [6–18] have been developed for solving problem (3). However, the accuracy of the above method cannot meet our requirements. In this paper, we introduce polynomial base functions to solve (3). The proposed method only requires dealing with a nonlinear system; thus, it is easy to implement. Furthermore, the proposed method is more accurate than the other numerical methods.

This paper is organized as follows. In Section 2, we propose the method for solving problem (3). Then, we analyze the convergence and give an error estimate in Section 3. In Section 4, we present eight numerical examples to show the accuracy of the method. Finally, the conclusions are drawn in Section 6.

2. Numerical Method

Now, we will examine the method for solving problem (3). Considering the function space whose base functions are

$$\mathbf{X} = [\varphi_0(x), \varphi_1(x), \varphi_2(x), \varphi_3(x), \dots],$$

then, the true solution of (3) can be given by

$$u(x) = \sum_{k=0}^{\infty} \beta_k \varphi_k(x), \quad (4)$$

where β_k is to be determined.

Let

$$\mathbf{X}_n = [\varphi_0(x), \varphi_1(x), \varphi_2(x), \dots, \varphi_n(x)],$$

and

$$\mathbf{Y}_n = [\beta_0, \beta_1, \beta_2, \dots, \beta_n]^T,$$

then, the n -th order approximate solution of (3) can be expressed by

$$u_n(x) = \sum_{k=0}^n \beta_k \varphi_k(x) = \mathbf{X}_n \mathbf{Y}_n. \quad (5)$$

Furthermore, the higher derivatives of $u_n(x)$ can also be calculated by

$$u_n^{(i)} = \mathbf{X}_n \mathbf{B}^i \mathbf{Y}_n, \quad i \geq 1, \quad (6)$$

where matrix $\mathbf{B}$ is dependent on the choice of $\mathbf{X}_n$.

Assuming $H = L^2[a, b]$ is the working space with inner product,

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx.$$

Substituting Equation (6) into (3), we have

$$Lu_n = f(x, u_n). \quad (7)$$

Assume that $\mathbf{Q} = [\psi_0(x), \psi_1(x), \psi_2(x), \dots, \psi_n(x)]$ is a linear independent set in H . Taking the inner product of Equation (7) with the element in $\mathbf{Q}$, two new matrices $\mathbf{S}_{(n+1) \times 1}$ and $\mathbf{T}_{(n+1) \times 1}$ are defined as

$$\mathbf{S}_{(n+1) \times 1} = \mathbf{T}_{(n+1) \times 1},$$

and

$$\mathbf{S}_{q,1} = \langle Lu_n, \psi_{q-1} \rangle,$$

$$\mathbf{T}_{q,1} = \langle f(x, u_n), \psi_{q-1}(x) \rangle,$$

where $1 \leq q \leq n+1$.

Considering the boundary conditions in problem (3),

$$\lambda_i u_n = r_i, \quad (i = 1, 2, \dots, N), \quad (8)$$

these N equations just modify N rows of $\mathbf{S}_{(n+1) \times 1}$ and $\mathbf{T}_{(n+1) \times 1}$, respectively. If the linear or nonlinear system $\mathbf{S} = \mathbf{T}$ is solved, the coefficients $\mathbf{Y}_n = [\beta_0, \beta_1, \beta_2, \dots, \beta_n]^T$ are determined uniquely. Then, the numerical solution of (3) can be expressed by

$$u_n(x) = \sum_{k=0}^n \beta_k \varphi_k(x). \quad (9)$$

Notice that continuous polynomials $\{x^k : k \in \mathbb{Z}\}$ are recommended when $\mathbf{X}_n$ is chosen since they are calculated rapidly, precisely defined, and can represent a variety of functions [19].

3. The Convergence of the Method and Error Estimate

Suppose that $H = L^2[a, b]$ is the working space, $P_n = \{\phi_0, \phi_1, \dots, \phi_n\}$ is the set of polynomials of n -th degree and $Y = \text{span}(P_n)$. Then, the L_2 -error can be achieved [19,20]. For $\forall u \in H$, u has the unique best approximation in Y . That is to say that there exists $\bar{u} \in Y$, such that

$$\|u - \bar{u}\|_2 \leq \|u - h\|_2, \quad \text{for } \forall h \in Y.$$

Let $\Phi = [\phi_0, \phi_1, \dots, \phi_n]^T$; there exists unique coefficients $\mathbf{A} = [a_0, a_1, \dots, a_n]$ such that

$$u \simeq \bar{u} = \sum_{i=0}^n a_i \phi_i = \mathbf{A} \Phi,$$

$\mathbf{A}$ can be obtained by

$$\mathbf{A} \langle \Phi, \Phi \rangle = \langle u, \Phi \rangle,$$

where

$$\langle u, \Phi \rangle = \int_a^b y(x) \Phi(x)^T dx = [\langle u, \phi_0 \rangle, \langle u, \phi_1 \rangle, \dots, \langle u, \phi_n \rangle],$$

and $\langle \Phi, \Phi \rangle$ is a $(n+1) \times (n+1)$ matrix. Let

$$\mathbf{C} = \langle \Phi, \Phi \rangle = \int_a^b \Phi(x) \Phi(x)^T dx,$$

then,

$$\mathbf{A} = \langle u, \Phi \rangle \mathbf{C}^{-1}.$$

Since the coefficients $\mathbf{A}$ are determined, we obtain the approximate solution $\bar{u}$. We have the following conclusion. Since we have defined the inner product in H by $\langle f, g \rangle = \int_a^b f(x)g(x)dx$ and $Y = \text{span}(P_n)$, then L_2 -error can be obtained by

$$\|u - \bar{u}\|_2^2 = \frac{\det[\int_a^b \Psi(x)\Psi(x)^T dx]}{\det[\int_a^b \Phi(x)\Phi(x)^T dx]},$$

where $\Phi = [\phi_0, \phi_1, \dots, \phi_n]^T$ and $\Psi = [y, \phi_0, \phi_1, \dots, \phi_n]^T$.

Next, we will prove that the approximation $\bar{u}$ converges to u .

Lemma 1. A function $u(x)$ is continuous on $[a, b]$ if and only if

$$\lim_{\delta \rightarrow 0} \omega(u, \delta) = 0,$$

$\omega(u, \delta) = \sup |u(x_1) - u(x_2)|$ where $x_1, x_2 \in [a, b]$ and $|x_1 - x_2| \leq \delta$.

Proof. Refer to [21]. $\square$

If $u(x)$ is bounded on $[a, b]$, we come to the following conclusion.

Lemma 2. If $u(x)$ is bounded on $[a, b]$, then

$$\left\| u - \sum_{k=0}^n u\left(\frac{k}{n}\right) \phi_k \right\|_{\infty} \leq \frac{3}{2} \omega\left(u, \sqrt{\frac{1}{n}}\right),$$

where $\|f(x)\|_{\infty} = \sup\{|f(x)| : x \in [a, b]\}$.

Proof. Refer to [21]. $\square$

Theorem 1. If $u(x)$ is bounded on interval $[a, b]$ and $Y = \text{span}\{P_n\}$, then

$$\|u - \mathbf{A}\Phi\|_2 \leq \frac{3}{2} \omega\left(u, \sqrt{\frac{1}{n}}\right), \quad (10)$$

where $\mathbf{A}\Phi$ is the best approximation to u in Y .

Proof. Since $\mathbf{A}\Phi$ is the best approximation to u in Y , then

$$\|u - \mathbf{A}\Phi\|_2 \leq \left\| u - \sum_{k=0}^n u\left(\frac{k}{n}\right) \phi_k \right\|_2 \leq \left\| u - \sum_{k=0}^n u\left(\frac{k}{n}\right) \phi_k \right\|_{\infty} \leq \frac{3}{2} \omega\left(u, \sqrt{\frac{1}{n}}\right).$$

$\square$

If $u(x)$ is continuous on $[a, b]$, then we have the following conclusion by Lemma 1

$$\lim_{n \rightarrow \infty} \omega\left(u, \sqrt{\frac{1}{n}}\right) = 0,$$

which means that $\mathbf{A}\Phi$ converges to u when $n \rightarrow \infty$.

If u is $n + 1$ times continuously differentiable, the following conclusion can be obtained.

Theorem 2. Suppose $u(x)$ is the true solution and $u_n(x)$ is the approximate solution to problem (3), $Y = \text{span}(P_n)$ and u_n is the best approximation to u in Y , $u \in C^{n+1}[a, b]$, an L_2 -error estimate is given by

$$\|u - u_n\|_2 \leq \frac{M(b-a)^{\frac{2n+3}{2}}}{\sqrt{2n+3} \cdot (n+1)!}, \quad (11)$$

where $M = \max_{x \in [a, b]} |u^{(n+1)}(x)|$.

Proof. Using Taylor's formula,

$$u_n(x) = u(a) + u'(a)(x-a) + \frac{u''(a)}{2!}(x-a)^2 + \cdots + \frac{u^{(n)}(a)}{n!}(x-a)^n + R_n(x),$$

where $R_n(x) = \frac{u^{(n+1)}(\eta)}{(n+1)!}(x-a)^{n+1}$, $\eta \in (a, b)$. And let

$$p(x) = u(a) + u'(a)(x-a) + \frac{u''(a)}{2!}(x-a)^2 + \cdots + \frac{u^{(n)}(a)}{n!}(x-a)^n.$$

Then, we have

$$\begin{aligned} \|u - u_n\|_2 &\leq \|u - p\|_2 \leq \sqrt{\int_a^b |y(x) - p(x)|^2 dx} \leq \sqrt{\int_a^b R_n^2(x) dx} \\ &\leq \frac{M(b-a)^{\frac{2n+3}{2}}}{\sqrt{2n+3} \cdot (n+1)!}, \end{aligned}$$

where $M = \max_{x \in [a, b]} |u^{(n+1)}(x)|$. $\square$

4. Applications of the Method

In this section, eight numerical examples are presented to demonstrate the accuracy of the proposed method.

Example 1. Consider the following Bagley–Torvik equation [22]:

$$\begin{aligned} [D^2 + D^{3/2} + D^0]u(x) &= 2 + x^2 + \frac{4\sqrt{x}}{\sqrt{\pi}}, \quad x \in [0, 1], \\ u(0) = 0, \quad u'(0) &= 0. \end{aligned} \quad (12)$$

The exact solution is $u(x) = x^2$. Our results are shown in Equation (13). The exact solution can be obtained by our method. The numerical result is shown in Figure 1.

$$\begin{aligned} u_0(x) &= 0, \\ u_1(x) &= 0, \\ u_n(x) &= x^2, \quad n \geq 2. \end{aligned} \quad (13)$$

Example 2. Consider the following fractional differential equation [23]:

$$\begin{aligned} D^{1/2}u(x) + \sqrt{\pi}u(x) &= \frac{8}{3\sqrt{\pi}} + \sqrt{\pi}x^2, \quad x \in [0, 1], \\ u(0) &= 0. \end{aligned} \quad (14)$$

The exact solution is $u(x) = x^2$. The results are shown in Equation (15). The true solution is achieved by the proposed method. The numerical result is presented in Figure 2.

$$\begin{aligned} u_0(x) &= 0, \\ u_1(x) &= \frac{32 + 10\pi}{40 + 15\pi}x, \\ u_n(x) &= x^2, \quad n \geq 2. \end{aligned} \quad (15)$$

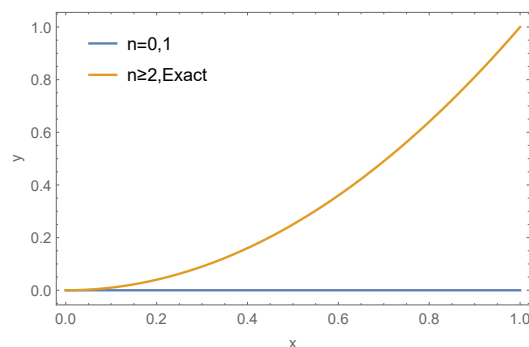


Figure 1. Numerical result for Example 1.

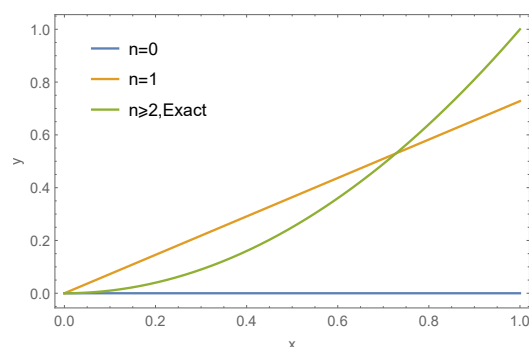


Figure 2. Numerical result for Example 2.

Example 3. Let us consider the following fractional differential equation [23]:

$$\begin{aligned} [D^{1/4} + D^0]u(x) &= -\frac{x^3}{2} + x^4 + \frac{64x^{11/4}(32x - 15)}{1155\Gamma(\frac{3}{4})}, \\ u(0) &= 0. \end{aligned} \quad (16)$$

The exact solution is $u(x) = x^4 - \frac{x^3}{2}$. Our solutions for various n are shown in Equation (17). The plots of the numerical solutions and the exact solution are shown in Figure 3. It can be found that the numerical solutions converge to the exact solution quickly.

$$\begin{aligned} u_0(x) &= 0, \\ u_1(x) &= \frac{-26624 - 13167c}{4180(32 + 21c)}x, \\ u_2(x) &= \frac{-687865856 - 728437248c - 194322975c^2}{8740(262144 + 303072c + 88935c^2)}x + \frac{3(26214400 + 28519568c + 7772919c^2)}{437(262144 + 303072c + 88935c^2)}x^2, \\ u_3(x) &= \frac{4(140737488355328 + 215214653964288c + 110716498835712c^2 + 19167948297729c^3)}{3933(1099511627776 + 1753074892800c + 941834456640c^2 + 170576707455c^3)}x \\ &\quad - \frac{2(35184372088832 + 54747587411968c + 28650322067136c^2 + 5044196920455c^3)}{69(1099511627776 + 1753074892800c + 941834456640c^2 + 170576707455c^3)}x^2 \\ &\quad + \frac{-25288767438848 - 39712678477824c - 20978221978560c^2 - 3728319462945c^3}{18(1099511627776 + 1753074892800c + 941834456640c^2 + 170576707455c^3)}x^3, \\ u_n(x) &= -\frac{1}{2}x^3 + x^4, \quad n \geq 4, \end{aligned} \quad (17)$$

where $c = \Gamma(\frac{3}{4})$. Again, the true solution is obtained by the proposed method.

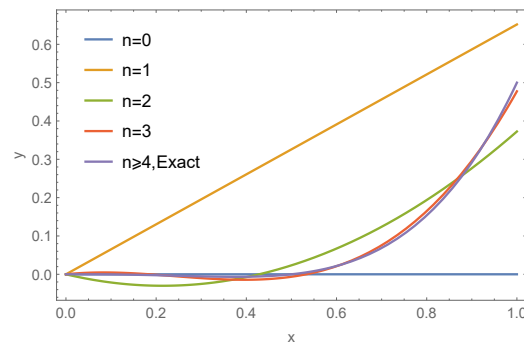


Figure 3. Numerical result for Example 3.

Example 4. Now consider the following Liouville–Bratu–Gelfand equation [6,14,24,25]:

$$\begin{aligned} u''(x) + \lambda e^{u(x)} &= 0, \\ u(0) &= 0, \quad u(1) = 0. \end{aligned} \quad (18)$$

The exact solution is $u(x) = -2 \ln \left[\frac{\cosh((x-\frac{1}{2})\frac{\theta}{2})}{\cosh(\frac{\theta}{4})} \right]$, where θ is the solution of $\theta = \sqrt{2\lambda} \cosh(\frac{\theta}{4})$. The absolute errors when $n = 10$ are shown in Table 1. It can be found that our method performs better than the other methods. The curves of the absolute errors are plotted in Figures 4 and 5 for $\lambda = 1$ and $\lambda = 2$ respectively.

Table 1. Comparison of absolute errors for Example 4.

λ	Node	LGSM [6]	B-spline [7]	Laplace [8]	ADM [9]	RKSM [23]	Present Method
$\lambda = 1$	0.1	7.5×10^{-7}	3.0×10^{-6}	2.0×10^{-6}	2.7×10^{-3}	6.2×10^{-5}	2.8×10^{-10}
	0.2	1.0×10^{-6}	5.5×10^{-6}	3.9×10^{-6}	2.0×10^{-3}	1.1×10^{-4}	3.6×10^{-10}
	0.4	5.2×10^{-7}	8.5×10^{-6}	7.7×10^{-6}	2.2×10^{-3}	1.7×10^{-4}	2.5×10^{-10}
	0.6	5.1×10^{-7}	8.5×10^{-6}	1.1×10^{-5}	2.2×10^{-3}	1.6×10^{-4}	2.5×10^{-10}
	0.9	7.4×10^{-7}	3.0×10^{-6}	1.2×10^{-5}	2.7×10^{-3}	5.6×10^{-5}	2.8×10^{-10}
$\lambda = 2$	0.2	5.7×10^{-6}	1.9×10^{-5}	4.2×10^{-3}	1.5×10^{-2}	5.8×10^{-4}	5.0×10^{-8}
	0.3	5.2×10^{-6}	2.8×10^{-5}	6.2×10^{-3}	5.9×10^{-3}	7.9×10^{-4}	2.0×10^{-8}
	0.5	1.5×10^{-6}	1.1×10^{-5}	9.6×10^{-3}	7.0×10^{-3}	9.7×10^{-4}	6.6×10^{-8}
	0.7	5.2×10^{-6}	6.9×10^{-5}	1.2×10^{-2}	5.9×10^{-3}	7.8×10^{-4}	2.0×10^{-8}
	0.9	4.0×10^{-6}	2.6×10^{-5}	1.1×10^{-2}	1.5×10^{-2}	3.0×10^{-4}	4.4×10^{-8}

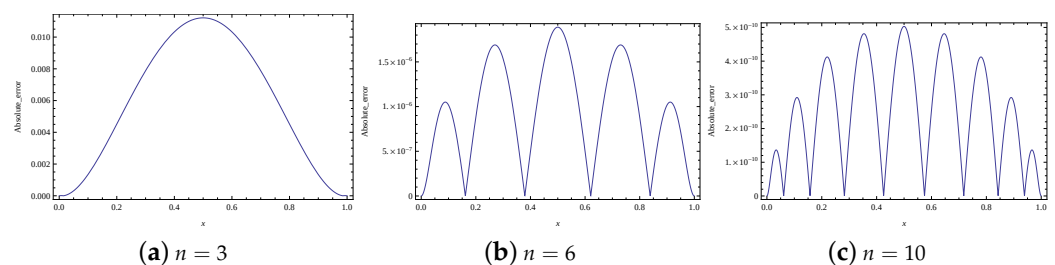


Figure 4. Absolute error for various n in Example 4 when $\lambda = 1$.

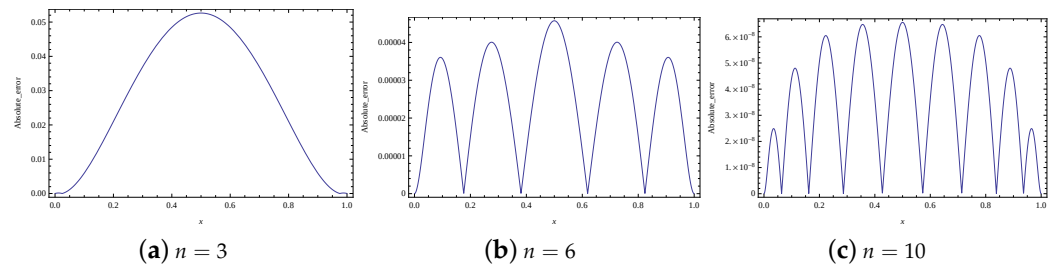


Figure 5. Absolute error for various n in Example 4 when $\lambda = 2$.

Example 5. Consider the following nonlinear boundary value problem [13,26–29]:

$$\begin{aligned} u^{(5)}(x) &= e^{-x}u^2(x), \\ u(0) &= u'(0) = u''(0) = 1, \quad u(1) = u'(1) = e. \end{aligned} \quad (19)$$

The exact solution is $u(x) = e^x$. Our result when $n = 10$ is shown in Equation (20). The comparison with the other methods is shown in Table 2. The proposed method is more accurate. The absolute errors are shown in Figure 6.

$$\begin{aligned} u_{10}(x) &= 1 + x + 0.5x^2 + 0.166667x^3 + 0.0416667x^4 + 0.00833333x^5 + 0.00138905x^6 \\ &\quad + 0.000197971x^7 + 0.0000254219x^8 + 2.27637 \times 10^{-6}x^9 + 4.63724 \times 10^{-7}x^{10}. \end{aligned} \quad (20)$$

Table 2. Comparison of absolute errors for Example 5.

Node	ADM [10]	HPM [11]	VIMHP [11]	VIM [11]	B-Spline [12]	Method in [13]	Method in [14]	RKSM [23]	Present Method
0.1	1.0×10^{-9}	1.0×10^{-9}	1.0×10^{-9}	1.0×10^{-9}	7.0×10^{-4}	2.3×10^{-7}	0	7.1×10^{-10}	1.1×10^{-13}
0.2	2.0×10^{-9}	2.0×10^{-9}	2.0×10^{-9}	2.0×10^{-9}	7.2×10^{-4}	1.6×10^{-6}	1.0×10^{-5}	4.6×10^{-9}	1.7×10^{-12}
0.3	1.0×10^{-8}	1.0×10^{-8}	1.0×10^{-8}	1.0×10^{-8}	4.1×10^{-4}	4.6×10^{-6}	1.0×10^{-5}	1.2×10^{-8}	5.4×10^{-12}
0.4	2.0×10^{-8}	2.0×10^{-8}	2.0×10^{-8}	2.0×10^{-8}	4.6×10^{-4}	8.9×10^{-6}	1.0×10^{-4}	2.2×10^{-8}	7.6×10^{-12}
0.5	3.1×10^{-8}	3.1×10^{-8}	3.1×10^{-8}	3.1×10^{-8}	4.7×10^{-4}	1.3×10^{-5}	3.2×10^{-4}	3.1×10^{-8}	5.6×10^{-12}
0.6	3.7×10^{-8}	3.7×10^{-8}	3.7×10^{-8}	3.7×10^{-8}	4.8×10^{-4}	1.6×10^{-5}	3.6×10^{-4}	3.5×10^{-8}	1.7×10^{-12}
0.7	4.1×10^{-8}	4.1×10^{-8}	4.1×10^{-8}	4.1×10^{-8}	3.9×10^{-4}	1.6×10^{-5}	1.4×10^{-4}	3.3×10^{-8}	3.4×10^{-13}
0.8	3.1×10^{-8}	3.1×10^{-8}	3.1×10^{-8}	3.1×10^{-8}	3.1×10^{-4}	1.2×10^{-5}	3.1×10^{-4}	2.2×10^{-8}	1.1×10^{-13}
0.9	1.4×10^{-8}	1.4×10^{-8}	1.4×10^{-8}	1.4×10^{-8}	1.6×10^{-4}	5.1×10^{-6}	5.8×10^{-4}	8.1×10^{-9}	1.6×10^{-13}

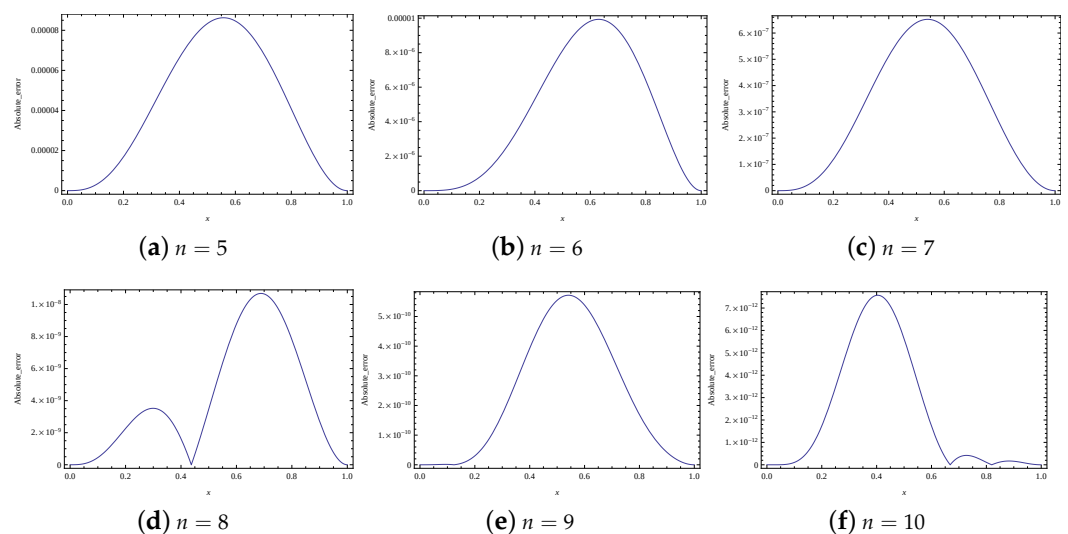


Figure 6. Absolute errors for various n in Example 5.

Example 6. Consider the nonlinear four-point BVP, which has widely been given as an example in [30–33].

$$\begin{aligned} u^{(4)}(x) + u(x)u'(x) - 4x^7 - 24 &= 0, \quad x \in [0, 1], \\ u(0) &= 0, \quad u^{(3)}\left(\frac{1}{4}\right) = 6, \quad u''\left(\frac{1}{2}\right) = 3, \quad u(1) = 1. \end{aligned} \quad (21)$$

Using the present method, we can obtain the approximate solutions which are shown in Equation (22). The exact solution is obtained when $n \geq 4$. The result is presented in Figure 7.

$$\begin{aligned} y_3(x) &= x^3, \\ y_n(x) &= x^4, \quad n \geq 4. \end{aligned} \quad (22)$$

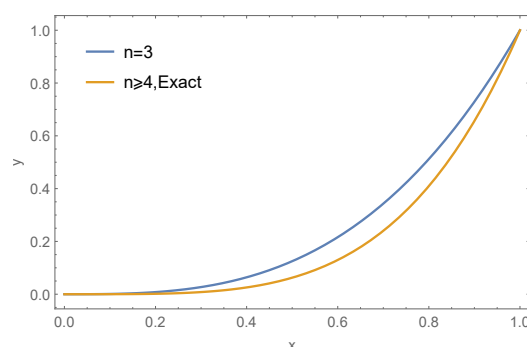


Figure 7. Numerical result for Example 6.

Example 7. Consider the following linear fourth order problem [15].

$$\begin{aligned} u^{(4)}(x) - (1+c)u''(x) + cu(x) &= \frac{1}{2}cx^2 - 1, \quad x \in [0, 1], \\ u(0) &= 1, \quad u'(0) = 1, \quad u(1) = 1 + \sinh(1), \quad u'(1) = 1 + \cosh(1). \end{aligned} \quad (23)$$

The exact solution is $u(x) = 1 + \frac{x^2}{2} + \sinh(x)$. The numerical results are shown in Table 3 when $c = 10^6$ and $n = 10$. Compared to the other existing method, the present method performs better. Proceeding as before, we describe the absolute errors of the numerical results in Figure 8.

Table 3. Comparison of absolute errors for Example 7.

Node	ADM [15]	HPM [15]	DTM [15]	RKSM [23]	Present Method
0.1	6.5×10^3	6.5×10^3	1.5×10^{-10}	1.8×10^{-8}	1.2×10^{-14}
0.2	8.8×10^4	8.8×10^4	3.7×10^{-8}	2.8×10^{-8}	4.6×10^{-14}
0.3	3.6×10^5	3.6×10^5	9.0×10^{-7}	4.6×10^{-8}	6.5×10^{-14}
0.4	9.1×10^5	9.1×10^5	8.5×10^{-6}	5.5×10^{-8}	1.8×10^{-14}
0.5	1.6×10^6	1.6×10^6	4.8×10^{-5}	6.3×10^{-8}	3.6×10^{-14}
0.6	2.3×10^6	2.3×10^6	1.9×10^{-4}	6.4×10^{-8}	1.2×10^{-14}
0.7	2.6×10^6	2.6×10^6	6.4×10^{-4}	5.8×10^{-8}	4.9×10^{-15}
0.8	2.1×10^6	2.1×10^6	1.7×10^{-3}	4.5×10^{-8}	1.9×10^{-14}
0.9	9.1×10^5	9.1×10^5	4.2×10^{-3}	2.1×10^{-8}	3.3×10^{-15}

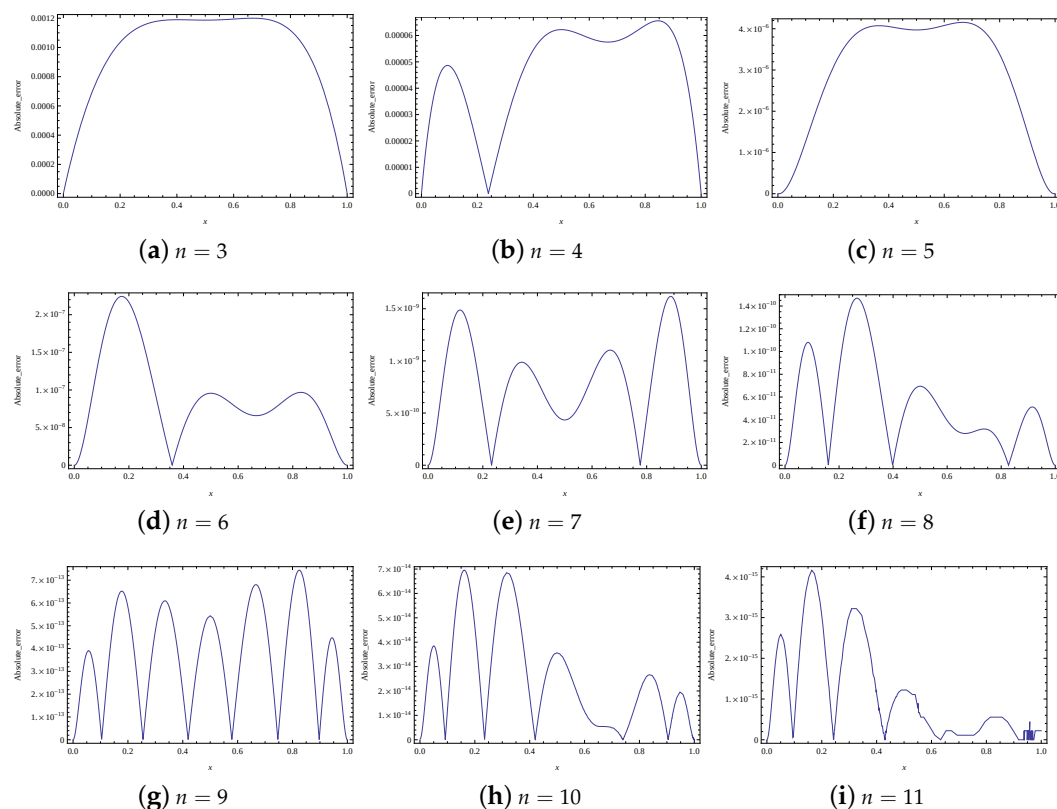


Figure 8. Absolute error for various n in Example 7.

Example 8. Finally, we consider the BVP governed by integro-differential equation involving nonlocal integral boundary conditions [34]:

$$\begin{aligned}
 u^{(4)}(x) + e^x u(x) + \int_0^x u(s) ds &= (1 + e^x) \sinh(x) + \cosh(x) - 1, \quad x \in [0, 1], \\
 u(0) = 0, \quad u''\left(\frac{1}{4}\right) + 3u^{(3)}\left(\frac{1}{4}\right) + \int_0^{\frac{1}{4}} u(t) dt &= \sinh\left(\frac{1}{4}\right) + 4 \cosh\left(\frac{1}{4}\right) - 1, \\
 2u''\left(\frac{1}{2}\right) - u^{(3)}\left(\frac{1}{2}\right) + \int_{\frac{1}{4}}^{\frac{1}{2}} u''(t) dt &= 2 \sinh\left(\frac{1}{2}\right) - \cosh\left(\frac{1}{4}\right), \quad u(1) = \sinh(1).
 \end{aligned} \tag{24}$$

The exact solution of this problem is given by $u(x) = \sinh(x)$. The numerical results are shown in Table 4. It can be observed that our method works well for this nonlocal boundary value problem.

Table 4. Comparison of absolute errors for Example 8.

Node	$ u - u_8 $ Method in [34]	$ u - u_{10} $ Method in [34]	$ u - u_{12} $ Method in [34]	$ u - u_8 $ Present Method	$ u - u_{10} $ Present Method	$ u - u_{12} $ Present Method
0.1	6.2×10^{-7}	4.2×10^{-9}	1.1×10^{-10}	3.7×10^{-8}	2.2×10^{-10}	2.6×10^{-11}
0.2	1.2×10^{-6}	8.1×10^{-9}	2.3×10^{-10}	6.7×10^{-8}	4.5×10^{-10}	5.1×10^{-11}
0.3	1.6×10^{-6}	1.2×10^{-8}	4.2×10^{-10}	9.1×10^{-8}	7.4×10^{-10}	7.6×10^{-11}
0.4	2.0×10^{-6}	1.4×10^{-8}	4.9×10^{-10}	1.1×10^{-7}	1.1×10^{-9}	1.0×10^{-10}
0.5	2.3×10^{-6}	1.7×10^{-8}	5.2×10^{-10}	1.2×10^{-7}	1.3×10^{-9}	1.2×10^{-10}
0.6	2.5×10^{-6}	2.0×10^{-8}	7.6×10^{-10}	1.2×10^{-7}	1.4×10^{-9}	1.2×10^{-10}
0.7	2.6×10^{-6}	2.1×10^{-8}	7.3×10^{-10}	1.1×10^{-7}	1.3×10^{-9}	1.1×10^{-10}
0.8	2.4×10^{-6}	2.1×10^{-8}	1.2×10^{-9}	8.2×10^{-8}	1.0×10^{-9}	9.0×10^{-11}
0.9	1.8×10^{-6}	1.7×10^{-8}	1.1×10^{-9}	4.7×10^{-8}	6.6×10^{-10}	5.5×10^{-11}

5. Results and Discussion

In this part, we give a brief discussion on the numerical results. For the numerical example whose exact solution is polynomial, the proposed method performs well and can give the true solution. And the numerical solution converges to the true solution quickly, as shown in Examples 1, 2, 3, and 6. The present method performs better than the other numerical methods for the remaining numerical examples. That is to say the current method is more accurate. Although the presented problems are subject to different types of boundary conditions, the numerical results show that the current method can handle these boundary conditions conveniently. The main difficulty of the proposed method is that we need to deal with the nonlinear system $\mathbf{S} = \mathbf{T}$. Since all the computations are performed by Mathematica, we can use ‘FindRoot’ or ‘NSolve’ command to solve such nonlinear equations.

6. Conclusions

In this paper, we give an efficient and reliable method for solving a class of nonlinear fractional differential equations. We have proved the convergence of the current method and give an error estimate. The proposed method is convenient to implement and only requires us to solve linear or nonlinear systems. Also, we present several numerical examples to show the accuracy of the method. Compared to the other existing methods, our method performs better. We can even achieve the exact solution if the true solution of the problem is in the polynomial form. In our future work, we may extend the proposed method to solve problems with boundary layers (singularity), such as Troesch’s problem [35].

Author Contributions: Methodology, X.S.; software, X.S.; writing—original draft, X.S.; writing—review and editing, R.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Sandev, T.; Tomovski, Z. *Fractional Equations and Models, Theory and Applications*; Springer Nature: Cham, Switzerland, 2019.
2. Sandev, T.; Tomovski, Z.; Dubbeldam, J.L.; Chechkin, A. Generalized diffusion-wave equation with memory kernel. *J. Phys. Math. Theor.* **2018**, *52*, 015201. [\[CrossRef\]](#)
3. Tomovski, Ž.; Dubbeldam, J.L.; Korbel, J. Applications of hilfer-prabhakar operator to option pricing financial model. *Fract. Calc. Appl. Anal.* **2020**, *23*, 996–1012. [\[CrossRef\]](#)
4. Kilbas, A. *Theory and Applications of Fractional Differential Equations*; Elsevier: Amsterdam, The Netherlands, 2006.
5. Marynets, K.; Pantova, D. Successive approximations and interval halving for fractional BVPs with integral boundary conditions. *J. Comput. Appl. Math.* **2024**, *436*, 115361. [\[CrossRef\]](#)
6. Abbasbandy, S.; Hashemi, M.; Liu, C.-S. The Lie-group shooting method for solving the bratu equation. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 4238–4249. [\[CrossRef\]](#)
7. Abu Arqub, O. Adaptation of reproducing kernel algorithm for solving fuzzy Fredholm–Volterra integrodifferential equations. *Neural Comput. Appl.* **2017**, *28*, 1591–1610. [\[CrossRef\]](#)
8. Arqub, O.A.; Al-Smadi, M.; Momani, S.; Hayat, T. Numerical solutions of fuzzy differential equations using reproducing kernel hilbert space method. *Soft Comput.* **2016**, *20*, 3283–3302. [\[CrossRef\]](#)
9. Arqub, O.A.; Maayah, B. Solutions of Bagley–Torvik and Painlevé equations of fractional order using iterative reproducing kernel algorithm with error estimates. *Neural Comput. Appl.* **2018**, *29*, 1465–1479. [\[CrossRef\]](#)
10. Akram, G.; Rehman, H.U. Solution of fifth order boundary value problems in reproducing kernel space. *Middle-East J. Sci. Res.* **2011**, *10*, 191–195.
11. Al-Said, E.A.; Noor, M.A. Cubic splines method for a system of third-order boundary value problems. *Appl. Math. Comput.* **2003**, *142*, 195–204. [\[CrossRef\]](#)
12. Bai, Z. On positive solutions of a nonlocal fractional boundary value problem. *Nonlinear Anal. Theory Methods Appl.* **2010**, *72*, 916–924. [\[CrossRef\]](#)
13. Caglar, H.; Caglar, S.; Twizell, E. The numerical solution of fifth-order boundary value problems with sixth-degree B-spline functions. *Appl. Math. Lett.* **1999**, *12*, 25–30. [\[CrossRef\]](#)

14. Caglar, H.; Caglar, N.; Özer, M.; Valaristos, A.; Anagnostopoulos, A.N. B-spline method for solving Bratu's problem. *Int. J. Comput. Math.* **2010**, *87*, 1885–1891. [[CrossRef](#)]
15. Momani, S.; Noor, M.A. Numerical comparison of methods for solving a special fourth-order boundary value problem. *Appl. Math. Comput.* **2007**, *191*, 218–224. [[CrossRef](#)]
16. Yousif, M.A.; Hamasalh, F.K. The fractional non-polynomial spline method: Precision and modeling improvements. *Math. Comput. Simul.* **2024**, *218*, 512–525. [[CrossRef](#)]
17. Jassim, H.K.; Hussein, M.A. A new approach for solving nonlinear fractional ordinary differential equations. *Mathematics* **2023**, *11*, 1565. [[CrossRef](#)]
18. Temimi, H.; Ben-Romdhane, M. A highly accurate discontinuous galerkin method for solving nonlinear Bratu's problem. *Alex. Eng. J.* **2024**, *95*, 50–58. [[CrossRef](#)]
19. Turkyilmazoglu, M. An effective approach for numerical solutions of high-order Fredholm integro-differential equations. *Appl. Math. Comput.* **2014**, *227*, 384–398. [[CrossRef](#)]
20. Kreyszig, E. *Introductory Functional Analysis with Applications*; John Wiley & Sons: Hoboken, NJ, USA, 1991; Volume 17.
21. Rivlin, T.J. *An Introduction to the Approximation of Functions*; Courier Corporation: North Chelmsford, MA, USA, 1981.
22. Torvik, P.J.; Bagley, R.L. On the appearance of the fractional derivative in the behavior of real materials. *J. Appl. Mech.* **1984**, *51*, 294–298. [[CrossRef](#)]
23. Wang, Y.-L.; Tian, D.; Bao, S.-H.; Li, Z.-Y. Using the iterative reproducing kernel method for solving a class of nonlinear fractional differential equations. *Int. J. Comput. Math.* **2017**, *94*, 2558–2572. [[CrossRef](#)]
24. Liao, S.; Tan, Y. A general approach to obtain series solutions of nonlinear differential equations. *Stud. Appl. Math.* **2007**, *119*, 297–354. [[CrossRef](#)]
25. McGough, J.S. Numerical continuation and the Gelfand problem. *Appl. Math. Comput.* **1998**, *89*, 225–239. [[CrossRef](#)]
26. Noor, M.A.; Mohyud-Din, S.T. Variational iteration method for fifth-order boundary value problems using he's polynomials. *Math. Probl. Eng.* **2008**, *2008*, 954794. [[CrossRef](#)]
27. Noor, M.A.; Mohyud-Din, S.T. A new approach to fifth-order boundary value problems. *Int. J. Nonlinear Sci.* **2009**, *7*, 143–148.
28. Wazwaz, A.-M. The numerical solution of fifth-order boundary value problems by the decomposition method. *J. Comput. Appl. Math.* **2001**, *136*, 259–270. [[CrossRef](#)]
29. Zhang, J. The numerical solution of fifth-order boundary value problems by the variational iteration method. *Comput. Math. Appl.* **2009**, *58*, 2347–2350. [[CrossRef](#)]
30. Ali, J.; Islam, S.; Islam, S.; Zaman, G. The solution of multipoint boundary value problems by the optimal homotopy asymptotic method. *Comput. Math. Appl.* **2010**, *59*, 2000–2006. [[CrossRef](#)]
31. Saadatmandi, A.; Dehghan, M. The use of Sinc-collocation method for solving multi-point boundary value problems. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 593–601. [[CrossRef](#)]
32. Tatari, M.; Dehghan, M. The use of the Adomian decomposition method for solving multipoint boundary value problems. *Phys. Scr.* **2006**, *73*, 672. [[CrossRef](#)]
33. Tatari, M.; Dehghan, M. An efficient method for solving multi-point boundary value problems and applications in physics. *J. Vib. Control.* **2012**, *18*, 1116–1124. [[CrossRef](#)]
34. Xie, L.-j.; Zhou, C.-l.; Xu, S. A new algorithm based on differential transform method for solving multi-point boundary value problems. *Int. J. Comput. Math.* **2016**, *93*, 981–994. [[CrossRef](#)]
35. Temimi, H.; Kurkcu, H. An accurate asymptotic approximation and precise numerical solution of highly sensitive Troesch's problem. *Appl. Math. Comput.* **2014**, *235*, 253–260. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.