Research Article

Nonpolynomial Spline Interpolation for Solving Fractional Subdiffusion Equations

Homan Emadifar \(^1, 2\) and Faraidun K. Hamasalh \(^2\)

\(^1\)Department of Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran
\(^2\)Department of Mathematics, College of Education, University of Sulaimani, Sulaimani, Kurdistan Region, Iraq

Correspondence should be addressed to Homan Emadifar; homan_emadi@yahoo.com

Received 6 June 2022; Accepted 3 August 2022; Published 5 September 2022

Abstract

The nonpolynomial spline interpolation is proposed to distinguish numerical analysis from the series boundary conditions, accuracy error estimations. The idea used in this article is readily applicable to obtain numerical solution of nonpolynomial spline interpolation. These analyze the methods that are suitable for the numerical solution of subdiffusion equation. The method has been shown to be stable by using von Neumann technique. The accuracy and efficiency of the scheme are checked by several examples to obtain numerical tests.

1. Introduction

In recent years, the derivation of solutions to fractional differential equations has become a hot topic in many areas of applied science and engineering. There are still not enough extremely precise numerical methods, although there have been many works on numerical methods for fractional differential equations. It is also very important to conduct research on fractional-order differential equations in unconstrained fields. A popular approach to solving these problems is to use artificial boundary conditions (ABCs) [1]. There are several applications of spline methods to the numerical solution of differential equations and, in particular, fractional differential equations [2–6]. The derivation of solutions to fractional differential equations has become a first-rate topic in many areas of applied science and engineering. Since fractional derivatives provide more accurate models than integer derivatives for most real-world problems, fractional differential equations are used to formulate a considerable number of applied problems. Applications are diverse and include viscoelastic and viscoelastic flows, control theory, transport problems, tumor evolution, random walks, continuum mechanics, and turbulence [7–13]. Finite differences, Fourier method, orthogonal spline collocation, implicit schemes with alternating direction, and compact finite difference methods have been successfully used in [14–22] to solve fractional subdiffusion equations. The author of [23] has presented an implicit numerical method for the fractional diffusion equation, where the fractional derivative is discretized by spline and the Crank–Nicolson discretization is used for the time variable. The authors of [24] have developed a novel nonpolynomial spline method for solving second-order hyperbolic equations, and their results are numerically more accurate than some finite difference methods, see [25–28]. The authors of [29] presented a quadratic nonpolynomial spline approach to approximate the solution of a system of second-order boundary value problems associated with a one-sided obstacle, and compared to collocation, finite difference, and certain common polynomial spline approaches, we used contact problems and produced approximations that were more accurate.

We consider the fractional differential equation on a bounded domain expressed by [1]

\[
D^\alpha_t u(x,t) - u_{xx}(x,t) = f(x,t), \quad (x,t) \in \Omega,
\]

\[
u(x,0) = \psi(x), \quad x \in [0,1],
\]
\[ u(0, t) = \phi(t), t > 0, \]  
\[ u(1, t) = \varphi(t), t > 0, \] where \( \Omega = \{(x, t)|0 \leq x \leq 1, 0 \leq t < +\infty\} \) and \( 0 < \alpha < 1. \)

2. Basic Definition

**Definition 1.** Let \( u(x) \) be a function defined on \((a, b)\), then the Riemann–Liouville fractional derivative is of the following form [12]:

\[
D^n u(x) = \frac{1}{\Gamma(m - \alpha)} \frac{d^m}{dx^m} \int_{0}^{x} (x - t)^{m-\alpha-1} u(t) dt, \alpha > 0, \\
m - 1 < \alpha < m.
\]

**Definition 2.** Let \( u(x) \) be a function defined on \((a, b)\), then the Caputo fractional derivative is of the following form [12]:

\[
D^n u(x) = \frac{1}{\Gamma(m - \alpha)} \int_{0}^{x} (x - t)^{m-\alpha-1} u^m(t) dt, \alpha > 0, \\
m - 1 < \alpha < m.
\]

3. Nonpolynomial Spline Method

In this section, we present nonpolynomial spline method for solving the problem equations (1)–(4) in the finite interval \([0, T]\). For the positive integers \( n \) and \( k \), we take \( h = 1/n \), \( \tau = T/k \) as the spatial stepsize and temporal stepsize, respectively. We use the notations \( x_i = ih, 0 \leq i \leq n \), \( t_j = j\tau, 0 \leq j \leq k \), and \( u_i^j = u(x_i, t_j) \). Let us consider a mesh with nodal points \( x_i = ih \) on \([0, 1]\) such that

\[ \Delta: 0 = x_0 < x_1 < x_2 < \ldots < x_{n-1} < x_n = 1. \]

For each segment \([x_i, x_{i+1}], i = 0, 1, 2, \ldots, n-1\), the nonpolynomial spline \( Q_i(x, t_j) \) has the following form:

\[
Q_i(x, t_j) = a_i(t_j) x_k(x - x_i) + b_j(t_j) \sin k(x - x_i) \\
+ c_j(t_j)(x - x_i)^2 + d_j(t_j)(x - x_i)^3 \\
+ e_j(t_j)(x - x_i)^4 + f_j(t_j)(x - x_i)^5, \]

where \( k \) is the frequency of the trigonometric part of the spline functions which can be real or pure imaginary and which will be used to raise the accuracy of the method [30]. Let \( u(x, t) \) be the exact solution, and let \( S_i^j \) be an approximation to \( u_i^j \) obtained by the segment \( Q_i(x, t_j) \) passing through the points \((x_i, S_i^j)\) and \((x_{i+1}, S_{i+1}^j)\).

To derive the coefficients of the equation (8), we first define

\[
Q_i(x, t_j) = S_i^j, Q_i(0, t_j) = M_i^j, Q_i^2(x, t_j) = M_i^{j+1}, Q_i^3(x, t_j) = M_i^{j+2}, Q_i^4(x, t_j) = M_i^{j+3}, Q_i^5(x, t_j) = M_i^{j+4}.
\]

By algebraic manipulation, we get

\[ \alpha_i(M_{i-2}^j + M_{i+2}^j) + \alpha_j(M_{i-1}^j + M_{i+1}^j) + \alpha_k M_i^j \]

\[ = \frac{1}{h^2} \left[ (u_{i-2}^j + u_{i+2}^j) + \beta_1(u_{i-1}^j + u_{i+1}^j) + \beta_2(u_{i-1}^j + u_{i+1}^j) + \beta_3 u_{i+1}^j \right], i = 3, \ldots, n-3.
\]
\[ t_i' = -2(2 + 2\beta_1 + 2\beta_2 + \beta_3)u_i' + h^2 \left( \frac{d^2u}{dx^2} \right)_i \left( (-9 + 2\alpha_1 + 2\alpha_2 + 2\alpha_3 + 4\alpha_4 - 4\beta_1 - \beta_2) + 1/12 \right) \]
\[ h^4 \left( \frac{d^4u}{dx^4} \right)_i \left( -81 + 108\alpha_1 + 48\alpha_2 + 12\alpha_3 - 16\beta_1 - \beta_2 \right) + \frac{1}{360} \]
\[ h^4 \left( \frac{d^4u}{dx^4} \right)_i \left( -729 + 2430\alpha_1 + 480\alpha_2 + 30\alpha_3 - 64\beta_1 - \beta_2 \right) + \frac{1}{20160} \]
\[ h^4 \left( \frac{d^4u}{dx^4} \right)_i \left( -6561 + 40824\alpha_1 + 3584\alpha_2 + 56\alpha_3 - 256\beta_1 - \beta_2 \right) \frac{1}{1814400} \]
\[ h^{10} \left( \frac{d^{10}u}{dx^{10}} \right)_i \left( -59049 + 590490\alpha_1 + 23040\alpha_2 + 90\alpha_3 - 1024\beta_1 - \beta_2 \right) + \frac{1}{239500800} \]
\[ h^{12} \left( \frac{d^{12}u}{dx^{12}} \right)_i \left( -531441 + 7794468\alpha_1 + 135168\alpha_2 - 132\alpha_3 - 4096\beta_1 \right) + \frac{1}{43589145600} \]
\[ h^{14} \left( \frac{d^{14}u}{dx^{14}} \right)_i \left( -4782969 + 96722262\alpha_1 + 7457472\alpha_2 - 182\alpha_3 - 16384\beta_1 - \beta_2 \right) + \ldots, \]

where \( \alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta_1, \beta_2, \) and \( \beta_3 \) are given in [1]. In this paper, we use fourth- and sixth-order method. From [30], \( \alpha_1 = 50, \alpha_2 = 10, \alpha_3 = -450, \alpha_4 = 900, \beta_1 = 24, \beta_2 = 15, \) and \( \beta_3 = -80 \) for fourth-order method and \( \alpha_1 = 1/42, \alpha_2 = 20/7, \alpha_3 = 397/14, \alpha_4 = 1208/21, \beta_1 = 24, \beta_2 = 15, \) and \( \beta_3 = -80 \) for sixth-order method.

**Lemma 1.** Let \( Q \) is an estimation of the smoothness function at class \( C_2 \), then the error estimation obtained as

\[ |u_i'' - Q_i''| \leq \frac{h^2}{12} Q_i^{(5)} + O(h^4), \]
\[ |Q_i'' - u_i''| \leq \frac{h^2}{12} u_i^{(4)} + O(h^4), \]
\[ |u_i'' - Q_i''| \leq \frac{h^4}{180} Q_i^{(5)} + O(h^5), \]
\[ |Q_i'' - u_i''| \leq \frac{h^4}{180} u_i^{(5)} + O(h^5). \]

**Proof.** The above relations are due to the expansion of Taylor sentences \( M_i, M_{i-1}, m_i, m_{i-1}, u_i, u_{i-1} \), at point \( x_i \) can be obtained as follows:

\[
\begin{align*}
\frac{d^i u}{dx^i} & = Q_i + \frac{h^2}{12} Q_i^{(4)} + \frac{h^4}{240} Q_i^{(6)} - \frac{h^6}{6048} Q_i^{(8)} + O(h^7), \quad i = 1, \ldots, n - 1, (I), \\
\frac{d^i u}{dx^i} & = u_i' - \frac{h^2}{12} u_i^{(4)} + \frac{h^4}{360} u_i^{(6)} + \frac{17h^6}{60480} u_i^{(8)} + O(h^7), \quad i = 1, \ldots, n - 1. (II).
\end{align*}
\]

Also, we have the same way.
\begin{align*}
\begin{cases}
    u'_i = Q'_i + \frac{h^4}{180}Q^{(5)}_i + \frac{h^6}{1512}Q^{(7)}_i - \frac{h^8}{14400}Q^{(9)}_i + O(h^9), \\
    Q'_i = u'_i - \frac{h^4}{180}d^{(5)}_i + \frac{h^6}{1512}d^{(7)}_i + \frac{h^8}{25920}d^{(8)}_i + O(h^9),
\end{cases} \\
\text{for } i = 1, \ldots, n - 1, \text{ (III)},
\end{align*}

(14)

And this is the end of the proof. \hfill \square

The local truncation error for the first- and second-order spline method is obtained as follows:

\begin{itemize}
    \item[(II)] Second-order method: \( t_i = 307h^6 u_i^{(6)} + O(h^7) \),
    \item[(III)] Sixth-order method: \( t_i = \frac{h^8 u_i^{(8)}}{252} + O(h^9) \),
    \item[(IV)] Eighth-order method: \( t_i = \frac{65629h^{10} u_i^{(10)}}{27936090} + O(h^{11}) \).
\end{itemize}

(15)

To obtain unique solution for the nonlinear system equation (10), we need four more equations. We define the following identities:

\begin{align*}
(12) \quad (\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4) &= (65, -104, 14, 24, 1), \\
(13) \quad (\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6, \eta_7, \eta_8, \eta_9, \eta_{10}, \eta_{11}, \eta_{12}) &= (-2248215317/19958400, 4539175/17600, -6055918291/6652800, \\
&\quad 9918918899/4989600, 892256279/285120, \\
&\quad 18019157507/4989600, -30650022317/9979200, 2380569353/1247400, \\
&\quad -16851712321/19958400, 503717713/1995840, -103447723/2851200, \\
&\quad 18976637/4989600), \\
(14) \quad (\mu_0, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5) &= (26, 14, -80, 15, 24, 1),
\end{align*}

(17)

\begin{equation}
\left(\begin{array}{c}
\gamma_2 = \gamma_{n-2} = \frac{521085679991}{373621248000}, \\
(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7, \sigma_8, \sigma_9, \sigma_{10}, \sigma_{11}, \sigma_{12})
\end{array}\right)
\end{equation}

\begin{equation}
\begin{pmatrix}
-7712745923 & 259885933 & -19222747601 & 8351791919 \\
159667200 & 5702400 & 53222400 & 11404800 \\
-3712133107 & 15883047 & -2614370381 & 28466192407 \\
3193344 & 71280 & 2280960 & 39916800 \\
-7204033313 & 16761737 & -43435489 & 113795873 \\
2280960 & 177408 & 2534400 & 7933600
\end{pmatrix}
\end{equation}

where from equation (1), we can write \( M^j_i \) in the form

\begin{equation}
M^j_i = \frac{\partial^j u_i}{\partial x^j} \approx \sum_k D^j_k u(x_i, t_j) - f(x_i, t_j).
\end{equation}

(18)

Now, we need to use the definition of discrete approximation of fractional derivative in time of any appropriate function \( z_2, 10 \) as
\[ D_i^a z(x_i, t_j) = a \sum_{q=0}^{i-1} R_{j,q}(z_{i-1} - z_i^q) + O(r^{2-a}), \quad (19) \]

\[ M_i^j = a \sum_{q=0}^{i-1} R_{j,q}(u_i^{q+1} - u_i^q) - f_i^j, \quad (20) \]

where \( a = 1/(2 - \alpha) \) and \( R_{j,q} = (j - q)^{1-\alpha} (j - q - 1)^{1-\alpha} \).

Then, from equations (18) and (19), we can express \( M_i^j \) in the following form:

\[
\begin{align*}
\left( a_1 a - \frac{1}{h_i^2} \right) (u_{i-3}^1 + u_{i+3}^1) + 
\left( a_2 a - \frac{\beta_1}{h_i^2} \right) (u_{i-2}^1 + u_{i+2}^1) + 
\left( a_3 a - \frac{\beta_2}{h_i^2} \right) (u_{i-1}^1 + u_{i+1}^1) + 
\left( a_4 a - \frac{\beta_3}{h_i^2} \right) u_i^1 \\
= a \left[ a_1 (u_{i-3}^0 + u_{i+3}^0) + a_2 (u_{i-2}^0 + u_{i+2}^0) + a_3 (u_{i-1}^0 + u_{i+1}^0) + a_4 u_i^0 \right] \\
\quad + a_1 (f_{i-3}^1 + f_{i+3}^1) + a_2 (f_{i-2}^1 + f_{i+2}^1) + a_3 (f_{i-1}^1 + f_{i+1}^1) + a_4 f_i^1,
\end{align*}
\]

for \( i = 3, 4, \ldots, n - 3 \), if \( j = 1 \), and

\[
\begin{align*}
\left( a_1 a - \frac{1}{h_i^2} \right) (u_{i-3}^1 + u_{i+3}^1) + 
\left( a_2 a - \frac{\beta_1}{h_i^2} \right) (u_{i-2}^1 + u_{i+2}^1) + 
\left( a_3 a - \frac{\beta_2}{h_i^2} \right) (u_{i-1}^1 + u_{i+1}^1) + 
\left( a_4 a - \frac{\beta_3}{h_i^2} \right) u_i^1 \\
= a \left[ a_1 (u_{i-3}^j + u_{i+3}^j) + a_2 (u_{i-2}^j + u_{i+2}^j) + a_3 (u_{i-1}^j + u_{i+1}^j) + a_4 u_i^j \right] \\
\quad + a_1 (f_{i-3}^j + f_{i+3}^j) + a_2 (f_{i-2}^j + f_{i+2}^j) + a_3 (f_{i-1}^j + f_{i+1}^j) + a_4 f_i^j,
\end{align*}
\]

for \( i = 3, 4, \ldots, n - 3 \), if \( j = 2, 3, \ldots, k \), with

\[
\mu_i^j = a \sum_{q=0}^{i-2} R_{j,q} \left( a_1 (u_{i-3}^q + u_{i+3}^q) + a_2 (u_{i-2}^q + u_{i+2}^q) + a_3 (u_{i-1}^q + u_{i+1}^q) + a_4 u_i^q \right)
\]

\[ - a \sum_{q=0}^{i-2} R_{j,q} \left( a_1 (u_{i-3}^{q+1} + u_{i+3}^{q+1}) + a_2 (u_{i-2}^{q+1} + u_{i+2}^{q+1}) + a_3 (u_{i-1}^{q+1} + u_{i+1}^{q+1}) + a_4 u_i^{q+1} \right). \]

Substituting equations (20) in (16), we get

\[
\begin{align*}
\sum_{k=0}^{4} \eta_k u_k^j + h^2 \sum_{k=1}^{12} \eta_k \left( a \sum_{q=0}^{j-1} R_{j,q} (u_k^{q+1} - u_k^q) - f_k^j \right) + \eta_k h^{14} (u^{(14)})^j_0 = 0, & \quad \text{at } i = 1, \\
\sum_{k=0}^{j-2} \mu_k u_k^j + h^2 \sum_{k=1}^{12} \sigma_k \left( a \sum_{q=0}^{j-1} R_{j,q} (u_k^{q+1} - u_k^q) - f_k^j \right) + \eta_k h^{14} (u^{(14)})^j_0 = 0, & \quad \text{at } i = 2, \\
\sum_{k=0}^{n-2} \mu_k u_k^j + h^2 \sum_{k=1}^{12} \sigma_k \left( a \sum_{q=0}^{j-1} R_{j,q} (u_{k-1}^{q+1} - u_{k-1}^q) - f_{k-1}^j \right) + \eta_k h^{14} (u^{(14)})^j_n = 0, & \quad \text{at } i = n - 2, \\
\sum_{k=0}^{4} \eta_k u_k^j + h^2 \sum_{k=1}^{12} \eta_k \left( a \sum_{q=0}^{j-1} R_{j,q} (u_{k-n}^{q+1} - u_{k-n}^q) - f_{k-n}^j \right) + \eta_k h^{14} (u^{(14)})^j_n = 0, & \quad \text{at } i = n - 1.
\end{align*}
\]
In the above systems, for each $j = 1, 2, \ldots, k$, clearly, we have $n$ linear equations in terms of $n$ unknowns $u_j^1, u_j^2, \ldots, u_j^n$.

4. Stability

In this section, we analyze the stability of equation (21), by means of the von Neumann method. We consider

\[
\xi_1 = \alpha_1 (\exp(I(i-3)\theta) + \exp(I(i+3)\theta)) + \left(\alpha_2 a - \frac{\beta_2}{h^2}\right) (\exp(I(i-2)\theta) + \exp(I(i+2)\theta)) + \left(\alpha_3 a - \frac{\beta_3}{h^2}\right) (\exp(Ii\theta))
\]

\[
\xi_2 = \alpha_2 (\exp(I(i-3)\theta) + \exp(I(i+3)\theta)) + \left(\alpha_2 a - \frac{\beta_2}{h^2}\right) (\exp(I(i-2)\theta) + \exp(I(i+2)\theta)) + \left(\alpha_3 a - \frac{\beta_3}{h^2}\right) (\exp(Ii\theta))
\]

where $\theta = \Phi h$, and this equation reduces to $\xi_1 = \rho \xi_0$ in which $\rho = x/y$ with

\[
x = a (2\alpha_1 \cos 3\theta + 2\alpha_2 \cos 2\theta + 2\alpha_3 \cos \theta + \alpha_4) + \frac{1}{h^2} (2 \cos 3\theta + 2\beta_1 \cos 2\theta + 2\beta_2 \cos \theta + \beta_3).
\]

Then, we obtain

\[
\xi_j = a \xi_{j-1} + \left(\alpha_3 a - \frac{\beta_3}{h^2}\right) (\exp(Ii\theta)) + \frac{1}{h^2} (2 \cos 3\theta + 2\beta_1 \cos 2\theta + 2\beta_2 \cos \theta + \beta_3) + \mu_j^i,
\]

in which

\[
\mu_j^i = a (2\alpha_1 \cos 3\theta + 2\alpha_2 \cos 2\theta + 2\alpha_3 \cos \theta + \alpha_4) \exp(Ii\theta) \sum_{q=0}^{i-2} R_{j,q} (\xi_q - \xi_{q+1}).
\]
From the denoted values of $x$ and $y$, the equation (30) can be simplified as

$$y\xi_j = x\xi_{j-1} + x \sum_{q=0}^{j-2} R_{j,q}(\xi_q - \xi_{q+1})$$

(32)

Then, we obtain

$$\xi_j = \rho \xi_{j-1} + \rho \sum_{q=0}^{j-2} R_{j,q}(\xi_q - \xi_{q+1}), \quad j = 2, 3, \ldots, k.$$  

(33)

Now, from this equation, it is obtained that $|\xi_j| \geq |\xi_0|$ for all $j = 2, 3, \ldots, k$ (see [3]). Hence, the nonpolynomial spline method is unconditionally stable.

5. Numerical Results

In this section, the wealth of the presented method non-polynomial spline method will be tested on three different examples, which are widely used in the literature. All problems were solved with best error estimations, and their results were compared with analytical or spline solutions.

<table>
<thead>
<tr>
<th>$T$</th>
<th>Fourth-order method</th>
<th>Sixth-order method</th>
<th>The method in [10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{4}$</td>
<td>$8.3e^{-3}$</td>
<td>$9.8444e^{-4}$</td>
<td>$1.0666e^{-6}$</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>$3.39e^{-2}$</td>
<td>$2.5638e^{-3}$</td>
<td>$3.8858e^{-6}$</td>
</tr>
<tr>
<td>1</td>
<td>$1.3e^{-1}$</td>
<td>$3.0239e^{-2}$</td>
<td>$1.4463e^{-5}$</td>
</tr>
</tbody>
</table>

Table 3: 2-norm error for $h = 1/16, \tau = 1/4096$.
given in the literature. In the following section, detailed information about this comparison will be presented.

Example 1. Consider problem equations (1)–(4) with
\[
f (x,t) = (x - 1)^2 \exp(-x) t^3
- 4x^2 (x - 1)^2 (14x^2 - 14x + 3),
\]
\((x,t) \in \Omega\)
\[
\psi(x) = x^4 (x - 1)^4, x \in [0, 1],
\]
\[
\phi(t) = t^{(3+\alpha)}, \phi(t) = 0, t \in [0, T].
\]
The exact solution of the problem is as follows [7]:
\[
u(x,t) = (x - 1)^4 \left[ \exp(-x) t^{3+\alpha} + x^4 \right], (x,t) \in \Omega.
\]

This problem is solved using the methods described in Section 2 for \(T = 1, 2, 3\). The results are reported in Tables 1 and 2.

Example 2. Consider problem equations (1)–(4) with
\[
f (x,t) = \frac{\Gamma(3)}{\Gamma(3-\alpha)} t^{-\alpha} \sin(2\pi x)
+ (2\pi)^2 t^2 \sin(2\pi x), (x,t) \in \Omega
\]
\[
\psi(x) = 0, x \in [0, 1],
\]
\[
\phi(t) = 0, \phi(t) = 0, t \in [0, T].
\]

It can be checked that the exact solution is \(\nu(x,t) = t^2 \sin(2\pi x) [10]\).

This problem is solved using the methods described in Section 2 for \(T = 1/4, 1/2, 1\). The results are reported in Table 3.

6. Conclusion

In this paper, we constructed the application of the non-polynomial spline method to obtain the numerical point solution of fractional partial differential equations. Numerical results showed that these techniques achieve accuracy, in numerical results illustrative graphs, the Figure 1 the comparison of the present results with the result of [10] shows the efficiency of the suggested technique. Also, the stability and error estimates of the methods are investigated and shown some figures of the results. Moreover, illustrations for the stability region of the schemes are derived.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References


