

## A Computational Method Based on Non-Polynomial Tension Splines for the Fourth-Order ODEs

Mohanad N. Abdul Sayed<sup>1</sup>, Ahmed R. Khlefha<sup>2</sup>, Mohammed H. Lafta<sup>3</sup>

<sup>1</sup>Department of Computer engineering and Artificial Intelligence Techniques, Qurna Polytechnic College, Southern Technical University, Basrah, Iraq

<sup>2,3</sup>Department of Mathematics, College of Education, University of Sumer, Thi-Qar, Iraq.

[arkdsh85@gmail.com](mailto:arkdsh85@gmail.com)

### Abstract

In this study, a non-polynomial tension spline method is proposed for approximating the solution of fourth-order boundary value problems. The original fourth-order problem is reduced to an equivalent system of second-order problems, which is then solved using a mixed non-polynomial tension spline formulation. Numerical examples compared with exact solutions demonstrate the accuracy and efficiency of the proposed method.

**Keywords:** Non-polynomial tension Spline method, boundary-value problems, exact solution.

### طريقة التلمه غير متعددة الحدود لحل مسائل القيم الحدية من الرتبة الرابعة

مهند نعمة عبد السيد<sup>1</sup>، احمد رشيد خليفة<sup>2</sup>، محمد هادي لفتة\*

<sup>1</sup> تقنيات هندسة الحاسوب والذكاء الاصطناعي، كلية البوليتكنك القرنة، الجامعة التقنية الجنوبية، البصرة، العراق

<sup>2,3</sup> الرياضيات، كلية التربية، جامعة سومر، ذي قار، العراق

### الخلاصة

في هذه الدراسة، نقترح طريقة تعتمد على دوال التلمه غير متعددة الحدود لتقريب حلول مسائل القيم الحدية من الرتبة الرابعة. تُختزل المسألة الأصلية من الرتبة الرابعة إلى نظام مكافئ من مسائل من الرتبة الثانية، ثم تُحل باستخدام صيغة دوال التلمه غير متعددة الحدود المختلطة. تُظهر الأمثلة العددية، مقارنةً بالحلول الدقيقة، دقة وكفاءة الطريقة المقترحة.

## 1. Introduction

A fourth-order boundary value problem is considered:

$$y^{(4)}(x) + Q_1(x)y^{(3)}(x) + Q_2(x)y^{(2)}(x) + Q_3(x)y^{(1)}(x) + Q_4(x)y(x) = r(x) \quad (1)$$

With the boundary conditions

$$\begin{cases} y(a) = \mu_0, y(b) = \mu_1 \\ y^{(2)}(a) = \Phi_0, y^{(2)}(b) = \Phi_1 \end{cases} \quad (2)$$

Where  $\mu_0, \mu_1, \Phi_0$  and  $\Phi_1$  are finite real constants and  $Q_1(x), Q_2(x), Q_3(x), Q_4(x)$  and  $r(x)$  are continuous on  $[a, b]$ . The problem's analytical solution (1), boundary conditions (2) Cannot get any options of  $Q_1(x), Q_2(x), Q_3(x), Q_4(x)$  and  $r(x)$ .

To get an approximate solution to the problem (1), we must use a numerical method.

Considered and built a cubic-spline technique to solve the following fourth-order boundary-value problem [7]:

$$w^{(4)}(x) + r(x)y(x) = p(x) \quad (3)$$

Where

$$\begin{cases} w(a) = \mu_0, w(b) = \mu_1 \\ w^{(2)}(a) = \Phi_0, w^{(2)}(b) = \Phi_1 \end{cases} \quad (4)$$

Where  $\mu_0, \mu_1, \Phi_0$  and  $\Phi_1$  are actual finite constants and equations  $r(x)$  and  $p(x)$  are continuous in  $[a, b]$ . The same boundary-value problems were solved using quantic splines [4]. An B-Spline approach is presented for a second-order linear system of boundary-value issues [2]. To handle fourth-order obstacle problems, we developed quadratic and cubic spline algorithms [1] and [3], provided a quadratic non-polynomial spline approach for solving second order obstacle issues, respectively [6]. Fifth-order boundary-value problems with derived Quartic spline solutions [5]. In order to resolve the issue (1) We decreased (1) by adding the boundary conditions to a system of second-order differential equations (2):

$$\left. \begin{aligned} S^{(2)}(x) + Q_1(x)S^{(1)}(x) + Q_2(x)S(x) + Q_3(x)y^{(1)}(x) + Q_4(x)y(x) &= r(x) \\ y^{(2)}(x) - S(x) &= 0 \end{aligned} \right\} \quad (5)$$

with the boundary conditions

$$\begin{cases} y(a) = \mu_0, y(b) = \mu_1 \\ S(a) = \Phi_0, S(b) = \Phi_1 \end{cases} \quad (6)$$

## 2. Description of the method

We start with a limited number of grid points  $x_i$ , the interval is divided  $[a, b]$  into  $n$  equal parts.  $x_i = a + ih, i = 0, 1, 2, \dots, n, x_0 = a, x_n = b$ . Let  $y(x)$  be the system's exact solution (1-2) and  $G_i$  be an approximation to  $y_i = y(x)$  obtained by the spline function  $G_i(x)$  passing through the points  $(x_i, y_i)$  and  $(x_{i+1}, y_{i+1})$ . a function  $G$  of class  $C^2[a, b]$ , which interpolates  $y(x)$  at the mesh points  $x_i, i = 0, 1, 2, \dots, n$ . the parameter  $\tau$ , and reduces to the normal cubic spline  $G(x)$  in  $[a, b]$  as  $\tau \rightarrow 0$ , is termed as Non-polynomial tension Spline function. For each segment  $[x_i, x_{i+1}], i = 0, 1, \dots, n - 1$  the Non-polynomial tension Spline  $E_i(x)$  has the form:

$$G_i(x) = a_i(e^{\tau(x-x_i)} - e^{-\tau(x-x_i)}) + b_i(e^{\tau(x-x_i)} + e^{-\tau(x-x_i)}) + c_i(x - x_i) + d_i \quad (7)$$

Where  $a_i, b_i, c_i$  and  $d_i$  are constants and  $\tau$  is free parameter to be determined. Let  $y_i$  be an approximation to  $y(x)$ , obtained by the segment  $G_i(x)$  the points are intersected by a mixed spline function  $[x_i, y_i]$  and  $[x_{i+1}, y_{i+1}]$ .  $G_i(x)$  is required to meet the interpolator requirements at  $x_i$  and  $x_{i+1}$ , the boundary conditions (6) and the first derivative continuity condition at the shared nodes  $[x_i, y_i]$ . To calculate the coefficients in equation (7) in terms of  $y_i, y_{i+1}, \Psi_i$  and  $\Psi_{i+1}$ , we define:

$$G_i(x_i) = y_i, G_i(x_{i+1}) = y_{i+1}$$

$$G_i^{(2)}(x_i) = \Psi_i, G_i^{(2)}(x_{i+1}) = \Psi_{i+1}. \quad (8)$$

We get the following expression using algebraic manipulation:

$$\begin{aligned} a_i &= \frac{-h^2\theta^2 \left( (e^\theta + e^{-\theta})\Psi_i - \Psi_{i+1} \right)}{2\theta^4(e^\theta - e^{-\theta})}, & b_i &= \frac{h^2 \Psi_{i+1}}{2\theta^2} \\ c_i &= \frac{-(\theta^2(y_i + y_{i+1}) + h^2(\Psi_{i+1} - \Psi_i))}{\theta^2}, & d_i &= y_i - \frac{h^2\Psi_i}{\theta^2} \end{aligned}$$

We will use the first derivative at  $(x_i, y_i)$ , that is  $G'_{i-1}(x_i) = G'_i(x_i)$  to obtain a regular relationship as follow:

$$\begin{aligned} y_{i-1} - 2y_i + y_{i+1} &= h^2 \left( \frac{e^{-\theta} - e^\theta + 2\theta}{\theta^2(e^\theta - e^{-\theta})} \right) \Psi_{i-1} + 2h^2 \left( \frac{\theta(e^\theta - e^{-\theta}) - (e^\theta - e^{-\theta})}{\theta^2(e^\theta - e^{-\theta})} \right) \Psi_i \\ &+ \left( \frac{h^2(e^{-\theta} - e^\theta + 2\theta)}{\theta^2(e^\theta - e^{-\theta})} \right) \Psi_{i+1}, \quad i = 1, \dots, n \end{aligned} \quad (9)$$

to simplicity the equation (9), we get

$$y_{i-1} - 2y_i + y_{i+1} = h^2[\xi(\Psi_{i-1} + \Psi_{i+1}) + 2\delta\Psi_i] \tag{10}$$

$$\xi = \frac{(e^{-\theta} - e^\theta + 2\theta)}{\theta^2(e^\theta - e^{-\theta})}$$

$$\delta = \frac{\theta(e^\theta + e^{-\theta}) - (e^\theta - e^{-\theta})}{\theta^2(e^\theta - e^{-\theta})}$$

In a similar manner, we get for  $S(x)$

$$S_{i-1} - 2S_i + S_{i+1} = h^2[\xi(\Sigma_{i-1} + \Sigma_{i+1}) + 2\delta\Sigma_i] \tag{11}$$

### 3. Analysis of the method

Explain how to apply the preceding section's numerical method. At mesh locations  $(x_i, y_i)$   $(x_i, y_i)$  and  $(x_i, w_i)$ , we discretize (5). As a result, we have

$$s_i'' + Q_1(x)S'(x) + Q_2(x)S(x) + Q_3(x)y'(x) + Q_4(x)y(x) = r(x) \tag{12}$$

For changing  $\Psi_i = y_i''$  and  $\Sigma_i = S''_i$  in (10) we find that.

$$\left. \begin{aligned} \Sigma_i &= r(x_i) - Q_1(x_i)S'(x_i) - Q_2(x_i)S(x_i) - Q_3(x_i)y'(x_i) - Q_4(x_i)y(x_i) \\ \Psi_i &= S_i \end{aligned} \right\} \tag{13}$$

We can use finite differences to approximate first derivative of  $y$  and  $S$  in Equation (12).

$$\left\{ \begin{aligned} y'_{i-1} &= \frac{-y_{i+1} + 4y_i - c}{2h} \\ y'_i &= \frac{y_{i+1} - y_{i-1}}{2h} \\ y'_{i+1} &= \frac{3y_{i+1} - 4y_i + y_{i-1}}{2h} \\ S'_{i-1} &= \frac{-S_{i+1} + 4S_i - 3S_{i-1}}{2h} \\ S'_i &= \frac{S_{i+1} - S_{i-1}}{2h} \\ S'_{i+1} &= \frac{3S_{i+1} - 4S_i + S_{i-1}}{2h} \end{aligned} \right. \tag{14}$$

To offset (14) in (13), we find as follow:

$$\left\{ \begin{aligned} \Sigma_i &= r(x_i) - Q_1(x_i)\frac{S_{i+1} - S_{i-1}}{2h} - Q_2(x_i)S(x_i) - Q_3(x_i)\frac{y_{i+1} - y_{i-1}}{2h} - Q_4(x_i)y(x_i) \\ \Sigma_{i+1} &= r(x_{i+1}) - Q_1(x_{i+1})\frac{3S_{i+1} - 4S_i + S_{i-1}}{2h} - Q_2(x_{i+1})S(x_{i+1}) - Q_3(x_{i+1})\frac{3y_{i+1} - 4y_i + y_{i-1}}{2h} - Q_4(x_{i+1})y(x_{i+1}) \\ \Sigma_{i-1} &= r(x_{i-1}) - Q_1(x_{i-1})\frac{-S_{i+1} + 4S_i - 3S_{i-1}}{2h} - Q_2(x_{i-1})S(x_{i-1}) - Q_3(x_{i-1})\frac{-y_{i+1} + 4y_i - 3y_{i-1}}{2h} - Q_4(x_{i-1})y(x_{i-1}) \\ \Sigma_{i-1} &= r(x_{i-1}) - Q_1(x_{i-1})\frac{-S_{i+1} + 4S_i - 3S_{i-1}}{2h} - Q_2(x_{i-1})S(x_{i-1}) - Q_3(x_{i-1})\frac{-y_{i+1} + 4y_i - 3y_{i-1}}{2h} - Q_4(x_{i-1})y(x_{i-1}) \end{aligned} \right. \tag{15}$$

$$\begin{aligned} \Psi_i &= S_i \\ \Psi_{i+1} &= S_{i+1} \\ \Psi_{i-1} &= S_{i-1} \end{aligned}$$

Now, to offset equations (15) into (11) and simplifying, we get

$$\left[ \frac{3\xi Q_1(x_{i-1})}{2h} - \xi Q_2(x_{i-1}) + \frac{2\delta Q_1(x_i)}{2h} - \frac{\xi Q_1(x_{i+1})}{2h} - \frac{1}{h^2} \right] S(x_{i-1}) + \left[ \frac{-4\xi Q_1(x_{i-1})}{2h} - 2\delta Q_2(x_i) + \frac{4\xi Q_1(x_{i+1})}{2h} + \frac{2}{h^2} \right] S(x_i) + \left[ \frac{\xi Q_1(x_{i-1})}{2h} - \frac{2\delta Q_1(x_i)}{2h} - \frac{3\xi Q_1(x_{i+1})}{2h} - \xi Q_2(x_{i+1}) - \frac{1}{h^2} \right] S(x_{i+1}) + \left[ \frac{3\xi Q_3(x_{i-1})}{2h} - \xi Q_4(x_{i-1}) + \frac{2\delta Q_3(x_i)}{2h} - \frac{\xi Q_3(x_{i+1})}{2h} \right] y(x_{i-1}) + \left[ \frac{-4\xi Q_3(x_{i-1})}{2h} - 2\delta Q_4(x_i) + \frac{4\xi Q_3(x_{i+1})}{2h} \right] y(x_i) + \left[ \frac{\xi Q_3(x_{i-1})}{2h} - \frac{2\delta Q_3(x_i)}{2h} - \frac{3\xi Q_3(x_{i+1})}{2h} - \xi Q_4(x_{i+1}) \right] y(x_{i+1}) = -\xi r(x_{i-1}) + 2\delta r(x_i) - \xi r(x_{i+1}) \tag{16}$$

### 4. Numerical examples

We'll now look at three numerical examples to show how our technique 16 compares to others. Maple 22 implements all calculations using step lengths of  $h=0.2$  and  $h=0.1$  to test the method's performance.

**Example 1:** Consider the boundary value problem.

$$y^{(4)}(x) - y(x) = 0$$

With the boundary conditions:

$$y(0) = y^{(2)}(0) = 1$$

$$y(1) = y^{(2)}(1) = 0$$

The exact solution is  $y(x) = \frac{1}{\sinh(1)} (e^{1-x} - e^{x-1})$ .

Table 1 and Table 2 describe the numerical results for absolute errors.

**Example 2:** Consider the boundary value problem

$$y^{(4)}(x) - y'''(x) + y''(x) = -8e^x$$

With the boundary conditions:

$$\begin{cases} y(0) = y(1) = 0 \end{cases}$$

$$\begin{cases} y^{(2)}(0) = 0, y^{(2)}(1) = -4e \end{cases}$$

The exact solution is  $y(x) = x(1-x)e^x$

Table 3 and Table 4 summarize the numerical data for absolute errors.

**Example 3:** Consider the boundary value problem

$$y^{(4)}(x) - 2y^{(2)}(x) + y(x) = 0$$

With the boundary conditions:

$$\begin{cases} y(0) = y^{(2)}(0) = 0 \end{cases}$$

$$\begin{cases} y(1) = y^{(2)}(1) = e \end{cases}$$

The exact solution is  $y(x) = e^x$

The numerical results for absolute errors are summarized in Table 5 and Table 6.

**Table 1-** Numerical solution for example 1 for subintervals.

x	Numerical solution				Exact. solution.
	h=0.2	h=0.1	h=0.01	h=0.001	
1	0.7557054801	0.7557054798	0.7557054797	0.7557054796	0.7557054795
2	0.5417400832	0.5417400743	0.5417400742	0.5417400741	0.5417400740
3	0.3495166101	0.3495166006	0.3495166005	0.3495166004	0.3495166003
4	0.1713204611	0.1713204546	0.1713204545	0.1713204544	0.1713204543
5	0	0	0	0	0

**Table 2-** Maximum absolute errors, Example 1

x	y(x)		y <sup>(1)</sup> (x)		y <sup>(2)</sup> (x)	
	Our method	Omotayo Taiwo[7]	Our method	Omotayo Taiwo[7]	Our method	Omotayo . Taiwo[7]
	h = 0.2	h = 0.2	h = 0.2	h = 0.2	h = 0.2	h = 0.2
0.2	2.614 × 10 <sup>-8</sup>	2.723 × 10 <sup>-7</sup>	4.361 × 10 <sup>-5</sup>	4.372 × 10 <sup>-4</sup>	2.881 × 10 <sup>-8</sup>	2.723 × 10 <sup>-7</sup>
0.4	3.423 × 10 <sup>-8</sup>	3.531 × 10 <sup>-7</sup>	3.137 × 10 <sup>-5</sup>	3.028 × 10 <sup>-4</sup>	3.640 × 10 <sup>-8</sup>	3.531 × 10 <sup>-7</sup>
0.6	3.136 × 10 <sup>-8</sup>	3.027 × 10 <sup>-7</sup>	1.878 × 10 <sup>-5</sup>	1.769 × 10 <sup>-4</sup>	3.218 × 10 <sup>-8</sup>	3.027 × 10 <sup>-7</sup>
0.8	1.815 × 10 <sup>-8</sup>	1.709 × 10 <sup>-7</sup>	6.105 × 10 <sup>-6</sup>	6.005 × 10 <sup>-5</sup>	1.618 × 10 <sup>-8</sup>	1.709 × 10 <sup>-7</sup>
1	0	0	7.314 × 10 <sup>-6</sup>	5.415 × 10 <sup>-5</sup>	0	0

**Table 3-** Numerical solution for example 2 for subintervals.

<i>x</i>	<i>Numerical solution</i>				<b>Exact. solution.</b>
	<b>h=0.2</b>	<b>h=0.1</b>	<b>h=0.01</b>	<b>h=0.001</b>	
1	0.1954244501	0.1954244421	0.1954244415	0.1954244414	0.1954244413
2	0.3580379331	0.3580379281	0.3580379277	0.3580379276	0.3580379275
3	0.4373085211	0.4373085131	0.4373085122	0.4373085121	0.4373085120
4	0.3560865513	0.3560865491	0.3560865487	0.3560865486	0.3560865485
5	0	0	0	0	0

**Table 4-** Maximum absolute errors, Example 2

<i>x</i>	<i>y(x)</i>		<i>y<sup>(1)</sup>(x)</i>		<i>y<sup>(2)</sup>(x)</i>	
	<b>Our method</b>	<b>Omotayo Taiwo[7]</b>	<b>Our method</b>	<b>Omotayo Taiwo[7]</b>	<b>Our method</b>	<b>Omotayo . Taiwo[7]</b>
	<b>h = 0.2</b>	<b>h = 0.2</b>	<b>h = 0.2</b>	<b>h = 0.2</b>	<b>h = 0.2</b>	<b>h = 0.2</b>
0.2	$1.960 \times 10^{-7}$	$1.960 \times 10^{-5}$	$5.154 \times 10^{-5}$	$5.058 \times 10^{-3}$	$3.858 \times 10^{-7}$	$3.858 \times 10^{-5}$
0.4	$3.211 \times 10^{-7}$	$3.211 \times 10^{-5}$	$7.183 \times 10^{-5}$	$7.390 \times 10^{-3}$	$6.271 \times 10^{-7}$	$6.271 \times 10^{-5}$
0.6	$3.588 \times 10^{-7}$	$3.568 \times 10^{-5}$	$1.742 \times 10^{-5}$	$1.062 \times 10^{-2}$	$6.887 \times 10^{-7}$	$6.887 \times 10^{-5}$
0.8	$2.683 \times 10^{-6}$	$2.683 \times 10^{-5}$	$1.505 \times 10^{-4}$	$1.505 \times 10^{-2}$	$5.106 \times 10^{-5}$	$5.106 \times 10^{-5}$
1	0	0	$1.568 \times 10^{-4}$	$1.568 \times 10^{-2}$	0	0

**Table 5-** Numerical solution for example 3 for subintervals.

<i>x</i>	<i>Numerical solution</i>				<b>Exact. solution.</b>
	<b>h=0.2</b>	<b>h=0.1</b>	<b>h=0.01</b>	<b>h=0.001</b>	
1	1.221402698	1.221402749	1.221402756	1.221402757	1.221402758
2	1.491824589	1.491824689	1.491824696	1.491824697	1.491824698
3	1.822118795	1.822118828	1.822118832	1.822118833	1.822118834
4	2.225540897	2.225540919	2.225540926	2.225540927	2.225540928
5	2.718281786	2.718281819	2.718281826	2.718281827	2.718281828

**Table 6-** Maximum absolute errors, Example 3

x	y(x)		y <sup>(1)</sup> (x)		y <sup>(2)</sup> (x)	
	Our method	Omotayo Taiwo [7]	Our method	Omotayo . Taiwo [7]	Our method	OmotayoTaiwo [7]
	h = 0.2	h = 0.2	h = 0.2	h = 0.2	h = 0.2	h = 0.2
0.2	$7.705 \times 10^{-6}$	$1.960 \times 10^{-5}$	$5.596 \times 10^{-5}$	$5.058 \times 10^{-3}$	$6.832 \times 10^{-5}$	$6.618 \times 10^{-4}$
0.4	$1.315 \times 10^{-6}$	$3.211 \times 10^{-5}$	$9.175 \times 10^{-5}$	$7.390 \times 10^{-3}$	$1.123 \times 10^{-4}$	$1.214 \times 10^{-3}$
0.6	$1.731 \times 10^{-6}$	$3.568 \times 10^{-5}$	$1.323 \times 10^{-4}$	$1.062 \times 10^{-2}$	$1.574 \times 10^{-4}$	$1.485 \times 10^{-3}$
0.8	$9.288 \times 10^{-6}$	$2.683 \times 10^{-5}$	$2.618 \times 10^{-4}$	$1.505 \times 10^{-2}$	$1.301 \times 10^{-4}$	$1.211 \times 10^{-3}$
1	0	0	$1.132 \times 10^{-4}$	$1.568 \times 10^{-2}$	0	0

## 5. Conclusion

For addressing fourth order boundary value problems utilizing non-polynomial tension Splines, a Non-polynomial tension Spline technique has been considered. The method is straightforward.

## References

- [1] M. Ghaffari and M. Razzaghi, "A new quartic spline approach for solving fourth-order boundary value problems," *Applied Mathematics and Computation*, vol. 268, pp. 179–188, 2015.
- [2] A. Korkmaz, "Non-polynomial spline methods for solving second-order boundary-value problems," *Numerical Algorithms*, vol. 72, pp. 789–807, 2016.
- [3] A. Gupta and S. Singh, "Convergence analysis of cubic spline methods for high-order boundary-value problems," *Journal of Computational and Applied Mathematics*, vol. 319, pp. 184–198, 2017.
- [4] J. Rashidinia and R. Jalilian, "Improved quintic spline solutions for fourth-order differential equations," *Numerical Methods for Partial Differential Equations*, vol. 34, pp. 1251–1268, 2018.
- [5] S. S. Siddiqi and G. Akram, "Advanced non-polynomial spline techniques for fifth-order boundary-value problems," *Applied Mathematics and Computation*, vol. 339, pp. 240–252, 2019.
- [6] M. A. Noor, K. I. Noor, and M. Waseem, "Quadratic and cubic non-polynomial spline methods for solving nonlinear boundary value problems," *Computational and Applied Mathematics*, vol. 40, pp. 1–18, 2021.
- [7] O. A. Taiwo and O. M. Ogunlaran, "A non-polynomial spline method for solving linear fourth-order boundary-value problems, *Journal of the Physical Sciences* Vol. 6(13),2011,pp . 3246-3254
- [8] H. Brunner, "Spline collocation methods for high-order differential equations: Recent developments," *Applied Numerical Mathematics*, vol. 181, pp. 45–63, 2023.