

A Barycentric Interpolation Collocation Method for Linear Nonlocal Boundary Value Problems

Dan Tian¹, Weiya Li¹ & Cec Yulan Wang¹

¹ College of Science, Inner Mongolia University of Technology, Hohhot, China

Correspondence: Weiya Li, College of Science, Inner Mongolia University of Technology, Hohhot, China.
E-mail: liweiya007@126.com

Received: April 27, 2016

Accepted: June 24, 2016

Online Published: August 17, 2016

doi:10.5539/mas.v10n11p140

URL: <http://dx.doi.org/10.5539/mas.v10n11p140>

Abstract

This paper is devoted to the numerical treatment of a class of higher-order multi-point boundary value problems(BVPs). The method is proposed based on the Lagrange interpolation collocation method, but it avoids the numerical instability of Lagrange interpolation. Numerical results obtained by present method compare with other methods show that the present method is simple and accurate for higher-order multi-point BVPs, and it is effective for solving six order or higher order multi-point BVPs.

Keywords: barycentric interpolation, reproducing kernel, multi-point boundary value problems, higher-order

1. Introduction

Collocation method as a numerical calculation method for solving differential equations, it has many merits, such as calculation formula is simple, program implementation is convenient. Use Lagrange interpolation collocation method to solve differential equations, when select too many nodes, Lagrange collocation formula will be not numerical instability, the famous Runge phenomenon illustrates the problem[1]. But barycentric interpolation collocation method has excellent numerical stability[2]. In this paper, we select suitable number of the second kind Chebyshev point as interpolation nodes, use barycentric interpolation collocation method establish the differential matrix of function to solve multi-point boundary value problems(BVPs).

As we know, boundary value problems(BVPs) arise in many fields(see[11-16]). In[3], Henderson and Kunkel proved the uniqueness of solutions for the following linear differential equations with nonlocal boundary conditions:

$$\begin{cases} u^{(m)}(x) + \sum_{i=0}^{m-1} a_i(x)u^{(i)}(x) = f(x) \\ u^{(i-1)}(x_j) = b_{ij}, 1 \leq i \leq m_j, 1 \leq j \leq k \\ u(x_{k+1}) - u(x_{k+2}) = b_m \end{cases} \quad (1)$$

where $a_i(x) \in C[a, b]$, m_j are positive integers such that $\sum_{i=1}^k m_i = m - 1$, $a < x_1 < x_2 < \dots < x_{k+2} < b$, b_m, b_{ij} are real numbers.

In [4],[5], Lin and Wu use the reproducing kernel to solve the following boundary value problems(BVPs). In [6], Li use another reproducing kernel method to solve this problem.

$$\begin{cases} u^{(4)}(x) + \sum_{i=0}^3 a_i(x)u^{(i)}(x) = f(x) \\ u^{(i-1)}(\xi_1) = b_i, 1 \leq i \leq 3, u(\xi_2) - u(\xi_3) = b_4 \end{cases} \quad (2)$$

In this work, we use barycentric interpolation to solve (1), several numerical examples are given to demonstrate the efficiency of the present method. The present method compared with the others methods, reveals that the present method is more effective and convenient.

2. Barycentric Interpolation

2.1 The Differential Matrix Of Barycentric Interpolation

According to the ideas of the collocation method, the all-order derivatives of function at nodes can be approximate as linear weighted sum of the function value at node. Consider the function $u(x)$ which defined in interval $[a, b]$, the values of function $u(x)$ at nodes, $a = x_1 < x_2 < \dots < x_n = b, u_i = u(x_i), i = 1, 2, \dots, n$, and the all-order derivatives of $u(x)$ at the nodes can be expressed as the linear weighted sum of the function value,

$$u_i^{(m)} = u^{(m)}(x_i) = \frac{d^m u(x_i)}{dx^m} = \sum_{j=1}^n D_{ij}^{(m)} u_j, m = 1, 2, \dots \quad (1)$$

Written in matrix form is,

$$u^{(m)} = D^{(m)} u \quad (2)$$

There, $u^{(m)} = [u_1^{(m)}, u_2^{(m)}, \dots, u_n^{(m)}]^T$ is the column vector of the m order derivatives m of unknown function at nodes, matrix $D^{(m)}$ is m order differential matrix of unknown function, the element of $D^{(m)}$, $D_{ij}^{(m)}$ is the weighted coefficient, $u^{(m)} = [u_1^{(m)}, u_2^{(m)}, \dots, u_n^{(m)}]^T$ is the value of unknown function at nodes.

Barycentric interpolation primary function is denoted by $L_j(x)$, the barycentric interpolation of $u(x)$ can be expressed as,

$$u(x) = \sum_{j=1}^n L_j(x) u_j \quad (3)$$

So, the one order and two order derivatives of $u(x)$ can be expressed as,

$$u'(x) = \sum_{j=1}^n L'_j(x) u_j, u''(x) = \sum_{j=1}^n L''_j(x) u_j \quad (4)$$

Barycentric interpolation primary function is,

$$L_j(x) = \frac{\frac{w_j}{x-x_j}}{\sum_{k=1}^n \frac{w_k}{x-x_k}} \quad (5)$$

There, $w_j = \frac{1}{\prod_{j \neq k} x_j - x_k}, j = 1, 2, \dots, n$ is barycentric interpolation weight, it based on the distribution of interpolation nodes. Multiply $x - x_j, (i \neq j)$ at both sides of (5) at the same time, after deformation is,

$$L_j(x) \sum_{k=1}^n w_k \frac{x - x_i}{x - x_k} = w_j \frac{x - x_i}{x - x_j} \quad (6)$$

To facilitate, let,

$$s(x) = \sum_{k=1}^n w_k \frac{x - x_i}{x - x_k} \quad (7)$$

Calculate derivative about x at both sides of (6) at the same time, we get,

$$L'_j(x) s(x) + L_j(x) s'(x) = w_j \left(\frac{x - x_i}{x - x_j} \right)' \quad (8)$$

$$L''_j(x) s(x) + 2L'_j(x) s'(x) + L_j(x) s''(x) = w_j \left(\frac{x - x_i}{x - x_j} \right)'' \quad (9)$$

Calculate derivative about x at both sides of (7),

$$s(x_i) = w_i \quad (10)$$

$$s'(x_i) = \sum_{k \neq i} \frac{w_k}{x_i - x_k} \tag{11}$$

$$s''(x_i) = -2 \sum_{k \neq i} \frac{w_k}{(x_i - x_k)^2} \tag{12}$$

Take formula (10)~(12) into the formula (8) and (9), we know $L_j(x_i) = 0 (i \neq j)$, so we can get,

$$L'_j(x_i) = \frac{\frac{w_j}{w_i}}{x_i - x_j}, \quad j \neq i \tag{13}$$

$$L''_j(x_i) = -2 \frac{\frac{w_j}{w_i}}{x_i - x_j} \left(\sum_{k \neq i} \frac{\frac{w_k}{w_i}}{x_i - x_k} + \frac{1}{x_i - x_j} \right), \quad j \neq i \tag{14}$$

If $i = j$, we know $\sum_{j=1}^n L_j(x) = 1$, calculate derivative about x at both sides, we get $\sum_{j=1}^n L'_j(x) = 0$, so,

$$L'_i(x_i) = - \sum_{j \neq i} L'_j(x_i) \tag{15}$$

$$L''_i(x_i) = - \sum_{j \neq i} L''_j(x_i) \tag{16}$$

Now, we can get one and two order differential matrix,

$$D_{ij}^{(1)} = L'_j(x_i), \quad D_{ij}^{(2)} = L''_j(x_i) \tag{17}$$

Using mathematical induction, we can get the recursion formula of m order differential matrix,

$$\begin{cases} D_{ij}^{(m)} = m(D_{ii}^{(m-1)} D_{ij}^{(1)} - \frac{D_{ij}^{(m-1)}}{x_i - x_j}), \quad i \neq j \\ D_{ii}^{(m)} = - \sum_{j=1, j \neq i}^n D_{ij}^{(m)} \end{cases} \tag{18}$$

2.2 The Barycentric Interpolation Collocation Formula of Multi-Point BVPs

Think about fuction (??)

$$\begin{cases} u^{(m)}(x) + \sum_{i=0}^{m-1} a_i(x)u^{(i)}(x) = f(x) \\ u^{(i-1)}(x_j) = b_{ij}, \quad 1 \leq i \leq m_j, \quad 1 \leq j \leq k \\ u(x_{k+1}) - u(x_{k+2}) = b_m \end{cases} \tag{19}$$

Where $a_i(x) \in C[a, b]$, m_j are positive integers, such that $\sum_{i=1}^k m_i = m - 1, a < x_1 < x_2 < \dots < x_{k+2} < b, b_m, b_{ij}$ are real numbers.

Let interval $[a, b]$ dispersed as $a = x_1 < x_2 < \dots < x_n = b$, let u_1, u_2, \dots, u_n as the value of function $u(x)$ at disperse nodes x_1, x_2, \dots, x_n , using the barycentric interpolation collocation can get approximate function $u(x)$,

$$u(x) = \sum_{j=1}^n L_j(x)u_j \tag{20}$$

Take formula (20) into the differential equation, we can get,

$$\sum_{j=1}^n L_j^{(m)}(x)u_j + \sum_{j=1}^n \sum_{i=0}^{m-1} a_i(x)L_j^{(i)}(x)u_j = f(x) \tag{21}$$

Let (21) accurate established at disperse nodes, we can get n equations,

$$\sum_{j=1}^n L_j^{(m)}(x_k)u_j + \sum_{j=1}^n \sum_{i=0}^{m-1} a_i(x_k)L_j^{(i)}(x_k)u_j = f(x_k), k = 1, 2, \dots, n \quad (22)$$

i.e.

$$\sum_{j=1}^n D_{kj}^{(m)}u_j + \sum_{j=1}^n \sum_{i=0}^{m-1} a_i(x_k)D_{kj}^{(i)}(x_k)u_j = f(x_k), k = 1, 2, \dots, n \quad (23)$$

Writ (23) in matrix form is

$$LU = F$$

There

$$L = D^{(m)} + A_i D^{(i)}, U = [u_1, u_2, \dots, u_n]^T, A_i = \text{diag}[a_i(x_k)] \\ F = [f_1, f_2, \dots, f_n], D^{(m)} = [D_{kj}^{(m)}]_{n \times n}, k, j = 1, 2, \dots, n$$

Take formula (20) into initial conditions,

$$\begin{cases} \sum_{k=1}^n D_{jk}^{(i-1)}u_k = b_{ij} \\ \sum_{j=1}^n L_j(x_{k+1})u_j - \sum_{j=1}^n L_j(x_{k+2})u_j = b_m \end{cases} \quad (24)$$

2.3 Applying Method of Initial Boundary Conditions

Use collocation method to solve the differential equation problem, the key is how to use the initial conditions. There are three method. The first method is displacement method. That is, we use the functions of (24) displace from the first to m functions of (23). The second method is supplemental method. That is, the functions of (24) add after functions of (23).The third method is elimination method. That is, we get u_1, u_n from (24), i.e. use $u_2, u_3 \dots, u_{n-1}$ express u_1 and u_n , and then take them into from the second to $n - 1$ functions of (23). About multi-point BVPs,we always use displacement method and supplemental method.

3. Numerical Experiment

In this section, six numerical examples are studied to demonstrate the accuracy of the present method.

Example 1^[4-6] Considering following fourth-order boundary value problems.

$$\begin{cases} u^{(4)}(x) - e^x u^{(3)}(x) + u(x) = 1 - e^x \cosh x + 2 \sinh x \\ u(\frac{1}{4}) = 1 + \sinh(\frac{1}{4}), u'(\frac{1}{4}) = \cosh(\frac{1}{4}), u''(\frac{1}{4}) = \sinh(\frac{1}{4}) \\ u(\frac{1}{2}) - u(\frac{3}{4}) = \sinh(\frac{1}{2}) - \sinh(\frac{3}{4}) \end{cases}$$

The exact solution is $u_T(x) = 1 + \sinh x$. The numerical results are presented in Table 1. Figure 1.

Example 2^[6] Considering following fifth-order boundary value problems.

$$\begin{cases} u^{(5)}(x) + \sin(2x)u'''(x) - u'(x) + \cos(2x)u(x) = -\sin x. 0 \leq x \leq 1 \\ u(0.1) = \sin(0.1), u'(0.1) = \cos(0.1), \\ u(0.4) = \sin(0.4), u'(0.4) = \cos(0.4), \\ u(0.7) - u(0.9) = \sin(0.7) - \sin(0.9). \end{cases}$$

The exact solution is $u_T(x) = \sin x$. The numerical results are presented in Table 2. Figure 2. Example 3^[7] Considering following fourth-order boundary value problems.

$$\begin{cases} u^{(4)}(x) + u^{(3)}(x) = f(x). 0 \leq x \leq 1 \\ u(0) = 0, u(\frac{1}{4}) = 0, \\ u(\frac{1}{2}) = 0, u(\frac{3}{4}) - u(1) = 0, \end{cases}$$

Table 1. Comparison of absolute errors for Example 1

x	Present method	Present method	Reproducing kernel
x	$u_T(x)$	$u_{21}(x)$	Method in [6]
0.0000	1.0000	1.0000	3.74077×10^{-7}
0.1464	1.1470	1.1470	2.56562×10^{-8}
0.2061	1.2076	1.2076	1.91517×10^{-9}
0.3455	1.3524	1.3524	1.83869×10^{-8}
0.4218	1.4344	1.4344	9.63508×10^{-8}
0.5000	1.5211	1.5211	2.50467×10^{-7}
0.6545	1.7022	1.7022	5.34168×10^{-7}
0.7270	1.7927	1.7927	3.83703×10^{-7}
0.8536	1.9610	1.9610	1.18927×10^{-6}
0.9045	2.0330	2.0330	2.6199×10^{-6}
1.0000	2.1752	2.1752	7.28022×10^{-6}

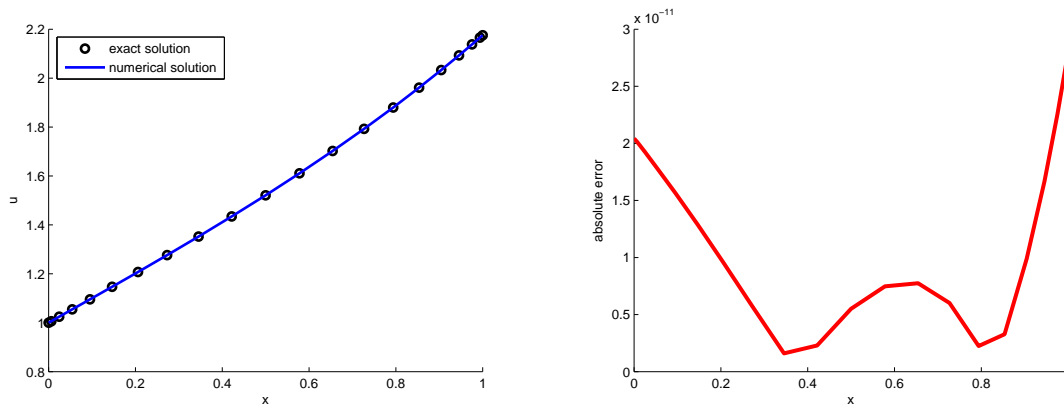


Figure 1. Present numerical method for Example 1, the first picture shows the exact solution and numerical solution, the second picture shows the numerical error when we select 21 nodes

Table 2. Comparison of absolute errors for Example 2

x	Present method	Present method	Reproducing kernel
x	$u_T(x)$	$u_{21}(x)$	Method in[6]
0.0000	0.0000	0.0000	1.51117×10^{-9}
0.1464	0.1459	0.1459	1.46647×10^{-9}
0.2061	0.2047	0.2047	1.08674×10^{-9}
0.3455	0.3387	0.3387	1.60481×10^{-9}
0.4218	0.4094	0.4094	1.82578×10^{-9}
0.5000	0.4794	0.4794	4.63265×10^{-10}
0.6545	0.6088	0.6088	9.86968×10^{-9}
0.7270	0.6646	0.6646	1.68006×10^{-8}
0.8536	0.7536	0.7536	2.04954×10^{-8}
0.9045	0.7861	0.7861	1.31484×10^{-8}
1.0000	0.8415	0.8415	2.92067×10^{-8}

Table 3. Comparison of errors for Example 3

N	Present method	Method in [7]	Method in [7]
	$\ u_T(x) - u(x)\ _2$	$ e^N(0.25) $	$ e^N(0.5) $
16	2.4437×10^{-9}	8.3708×10^{-5}	3.6148×10^{-6}
32	8.6524×10^{-13}	2.1097×10^{-5}	8.7657×10^{-7}
64	6.0749×10^{-11}	5.2848×10^{-6}	2.1744×10^{-7}
128	3.1058×10^{-8}	1.3219×10^{-6}	5.4254×10^{-8}

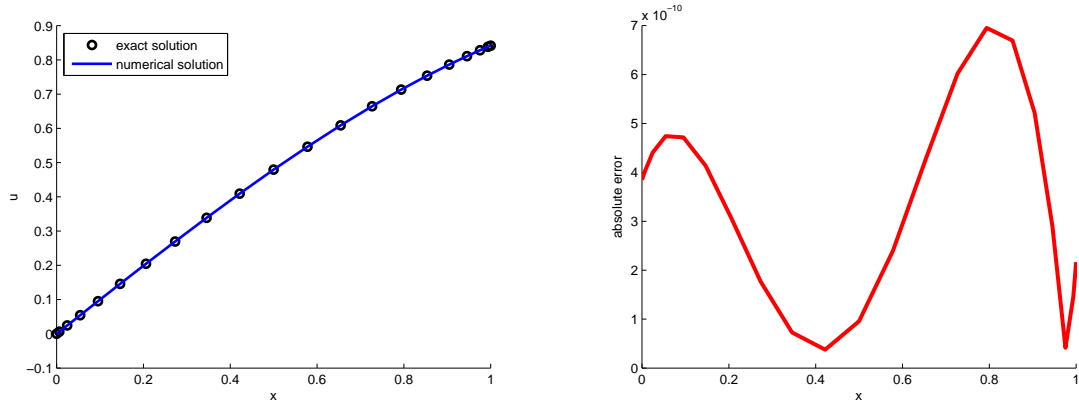


Figure 2. Present numerical method for Example 2, the first picture shows the exact solution and numerical solution, the second picture shows the numerical error when we select 21 nodes

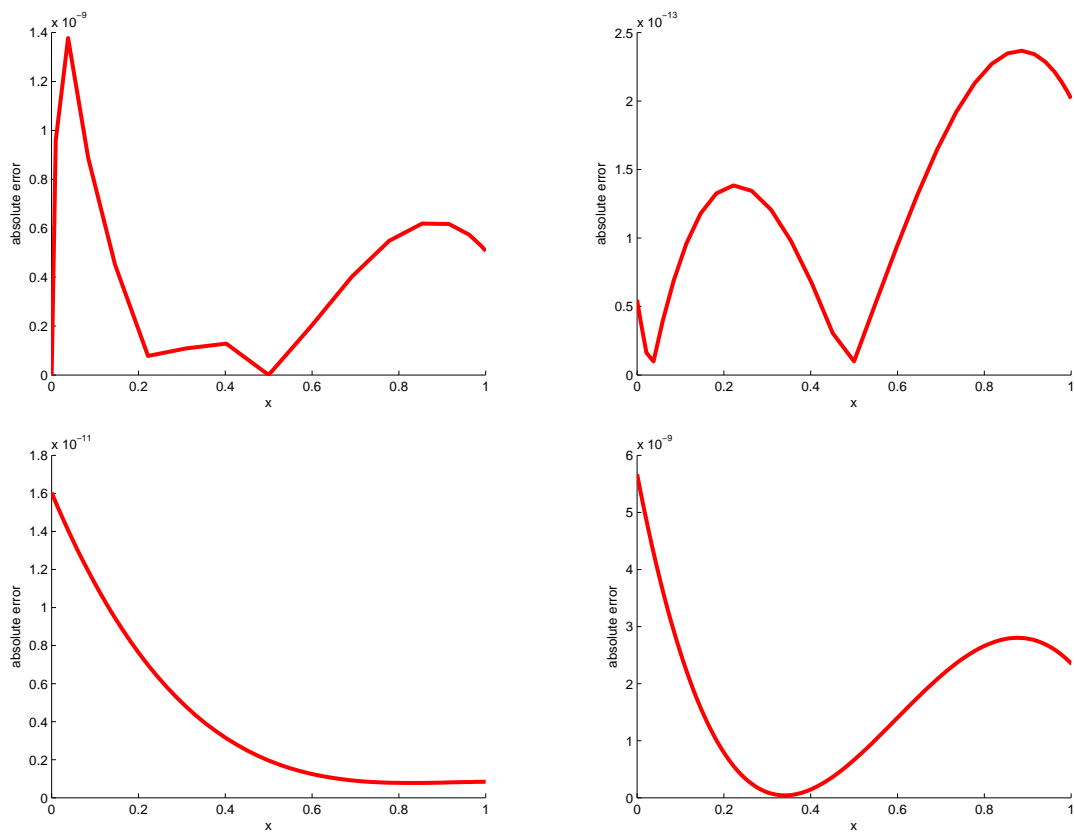


Figure 3. Present numerical method for Example 3, the first picture shows the numerical error when we select 17 nodes, the second picture is 33 nodes, the third picture is 65 nodes, the fourth picture is 129 nodes

The exact solution is $u_T(x) = x(x - \frac{1}{4})(x - \frac{1}{2})e^{-4xln4}$. The numerical results are presented in Table 3. Figure 3.

Example 4^[7] Considering following fourth-order boundary value problems.

$$\begin{cases} u^{(4)}(x) + u^{(3)}(x) = f(x), 0 \leq x \leq 1 \\ u(0) = 0, u'(0) = 0, \\ u(\frac{1}{2}) = 0, u(\frac{3}{4}) - u(1) = 0, \end{cases}$$

The exact solution is $u_T(x) = x^2(x - \frac{1}{2})e^{4ln(\frac{9}{32})x}$. The numerical results are presented in Table 4. Figure 4.

Table 4. Comparison of errors for Example 4

N	Present method $\ u_T(x) - u(x)\ _2$	Method in [7] $ e^N(0) $	Method in [7] $ e^N(0.5) $
64	6.6165×10^{-10}	5.2848×10^{-5}	2.1744×10^{-8}
128	7.1460×10^{-8}	1.3219×10^{-6}	5.4254×10^{-8}

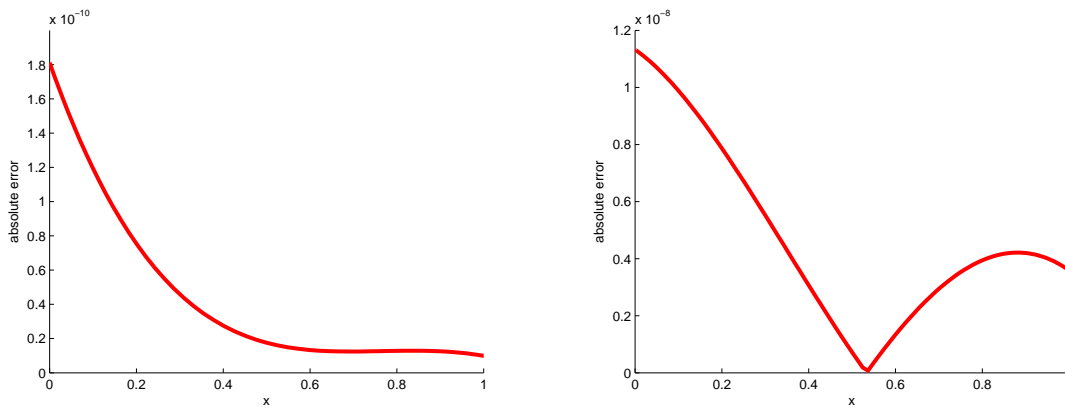


Figure 4. Present numerical method for Example 4, the first picture shows the numerical error when we select 65 nodes, the second picture is 129 nodes

Example 5 Considering following sixth-order boundary value problems.

$$\begin{cases} u^{(6)}(x) + u(x) = 12xcosx + 30sinx, -1 \leq x \leq 1 \\ u(-1) = 0, u(\frac{1}{2}) = -\frac{3}{4}sin(\frac{1}{2}), \\ u'(-1) = 2sin1, u'(\frac{1}{4}) = \frac{1}{2}sin(\frac{1}{4}) - \frac{15}{16}cos(\frac{1}{4}), \\ u''(-1) = -4cos1 - 2sin1, u''(1) = 4cos1 + 2sin1 \end{cases}$$

The exact solution is $u_T(x) = (x^2 - 1)sinx$. The numerical results are presented in Table 5. Figure 5.

Table 5. The numerical results of Example 5

error	$\ u_T(x) - u_{16}(x)\ _2$	$\ u_T(x) - u_{21}(x)\ _2$	$\ u_T(x) - u_{26}(x)\ _2$
absolute error	2.4078×10^{-11}	5.3904×10^{-10}	2.9905×10^{-9}
relative error	2.6464×10^{-11}	5.1308×10^{-10}	2.5460×10^{-9}

Example 6 Considering following eighth-order boundary value problems.

$$\begin{cases} u^{(8)}(x) - u(x) = -48e^x - 16xe^x, 0 \leq x \leq 1 \\ u(0) = 0, u(\frac{1}{2}) = \frac{1}{4}e^{\frac{1}{2}}, \\ u'(0) = 1, u'(\frac{3}{4}) = -\frac{5}{16}e^{\frac{3}{4}}, \\ u''(0) = 0, u''(1) = -4e, \\ u'''(0) = -3, u'''(1) = -9e, \end{cases}$$

The exact solution is $u_T(x) = x(1 - x)e^x$. The numerical results are presented in Table 6. Figure 6.

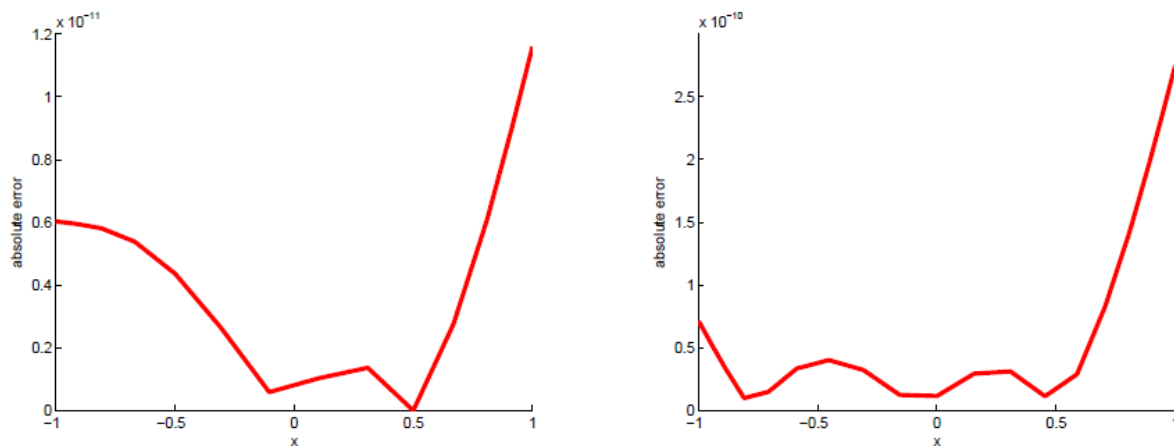


Figure 5. Present numerical method for Example 5, the first picture shows the numerical error when we select 16 nodes, the second picture is 21 nodes

Table 6. The numerical results of Example 6

error	$\ u_T(x) - u_{21}(x)\ _2$	$\ u_T(x) - u_{26}(x)\ _2$
absolute error	6.3861×10^{-9}	1.9445×10^{-8}
relative error	5.4288×10^{-9}	1.4785×10^{-8}

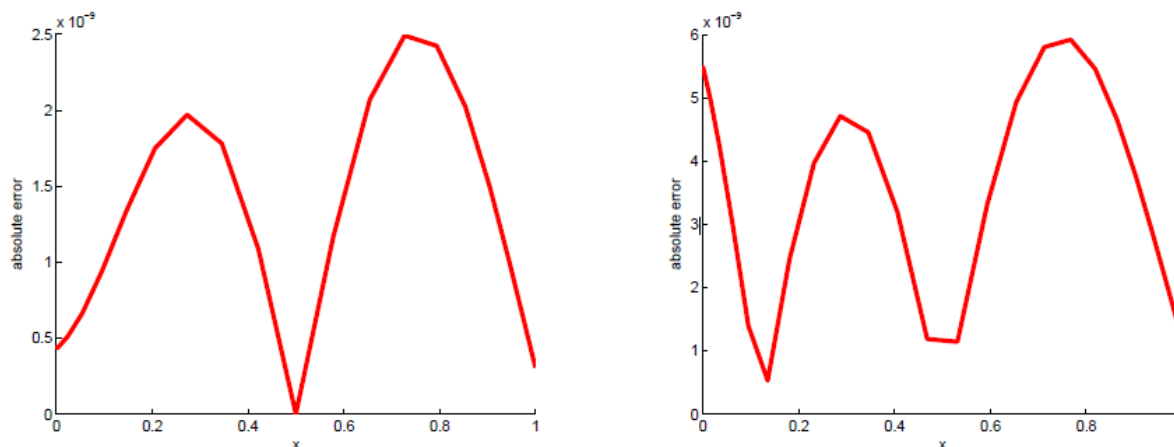


Figure 6. Present numerical method for Example 6, the first picture shows the numerical error when we select 21 nodes, the second picture is 26 nodes

4. Discussion

In this paper, we devote to the numerical treatment of a class of higher-order multi-point solving higher-order multi-point BVPs. The numerical results demonstrate that the method is quite accurate and efficient for linear higher-order multi-point BVPs. This makes it easy to solve the higher-order multi-point BVPs. It is worthy to note that this method can be generalized to more higher order BVPs. All computations are performed by the MatlabR2013a software package.

Acknowledgements

The authors would like to express their thanks to the unknown referees for their careful reading and helpful comments. This paper is supported by the Natural Science Foundation of China (No.11361037), the Natural

Science Foundation of China (No.11361037), the Natural Science Foundation of Inner Mongolia(No.2013MS0109, No.2015MS0118), the Foundation of Inner Mongolia University of Technology (No.X201223).

References

- Abu Arqub, O., & Al-Smadi, M. (2004). Numerical algorithm for solving two-point, second-order periodic boundary value problems for mixed integro-differential equations. *Applied Mathematics and Computation*, 243(2014), 911-922.
- Abu Arqub, O., & Al-Smadi, M. (2013). Solving Fredholm integro-differential equations using reproducing kernel Hilbert space method. *Applied Mathematics and Computation*, 219(2013), 8938-8948.
- Cui, M., & Lin, Y. (2009). Nonlinear Numerical Analysis in the Reproducing Kernel Space. *Nova Science*.
- Hu, Y., & Haiyuan, Yu. (2012). Collocation Methods for High order nonlocal multipoint-boundary value problem. *Xiangtan University*, 2012.
- Jean-Paul, B., & Lloyd, N. T. (2004). Barycentric Lagrange Interpolation. *SIAM Review*, 46(2004), 501-517.
- Johnny, H., & Curtis, J. K. (2008). Uniqueness of solution of linear nonlocal boundary value problems. *Applied Mathematics Letter*, 21(2008), 1053-1056.
- Jiang, W., & Chen, Z. (2014). A collocation method based on reproducing kernel for a modified anomalous subdiffusion equation. *Numerical Methods for Partial Differential Equations*, 30(2014), 289-300.
- Geng, F., Qian, S. P., & Li, S. (2014). A numerical method for singularly perturbed turning point problems with an interior layer. *Journal of Computational and Applied Mathematics*, 255(2014), 97-105.
- Lin, Y., & Lin, J. (2010). Numerical algorithm about a class of linear nonlocal boundary value problems. *Applied Mathematics Letter*, 23 (2010), 997-1002.
- Sim, V. R., & Stuart, C. (2008). Ecological Design, Washington. *Ethical standards of psychologists*. Washington, DC: Island Press.
- Wazwaz, A. M. (2000). A new algorithm for calculating Adomian polynomials for nonlinear operators. *Applied Mathematics and Computation*, 111(2000), 53-69.
- Wazwaz, A. M. (2001). A reliable algorithm for solving boundary value problems for higher-order integro-differential equations. *Applied Mathematics and Computation*, 118(2001), 327-342.
- Wu, B. Y., & Li, X. Y. (2011). A new algorithm for a class of linear nonlocal boundary value problems based on the reproducing kernel method. *Applied Mathematics Letters*, 24(2011), 156-159.
- Yao, H., & Lin, Y. (2009). Solving singular boundary-value problems of higher even-order. *Journal of Computational and Applied Mathematics*, 223(2009), 703-713.
- Zhou, Y., Cui, M., & Yingzhen, Lin. (2009). Numerical algorithm for parabolic problems with non-classical conditions. *Journal of Computational and Applied Mathematics*, 230(2), 770-780.
- Zhi-Yuan, L., & Yu-Lan, W. (2015). Solving a class of linear nonlocal boundary value problems using the reproducing kernel. *Applied Mathematics and computation*, 265(2015), 1098-1105.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).