A Nonclassical Collocation Method For Solving Two-Point Boundary Value Problems Over Infinite Intervals

M. Maleki and M. Tavassoli Kajani

Department of Mathematics, Mobarakeh Branch, Islamic Azad University, Isfahan, Iran.

Abstract: A collocation method for solving two-point boundary value problems posed on an infinite interval involving a second order linear differential equation is proposed. By reducing the infinite interval to a finite interval that is large and approximating the variable using Lagrange interpolation, the resulting boundary value problem is reduced to an algebraic system by using nonclassical pseudospectral method. The applications are demonstrated via test examples.

Key words: Collocation method, Two-point boundary value problem, infinite interval.

INTRODUCTION

During the last few years much progress has been made in the numerical treatment of initial value problems and boundary value problems over infinite intervals (see for example (Ibibola, 2010; Ravi Kanth, 2003) and references therein). Typically, these problems arise very frequently in fluid dynamics, aerodynamics, quantum mechanics, electronics, and other domains of science. A few notable examples are the Von Karman swirling flows (Lentini, 1980; Markowich, 1982), combined forced and free convection over a horizontal plate (Schneider, 1979) and eigenvalue problems for the Schrodinger equation (Lentini, 1980). In many cases, the domain of the governing equations of these problems is infinite or semi-infinite so that the special treatment is required for these so called infinite interval problems. The analytical solutions for these problems are not readily attainable and thus the problem is brought to the problem of finding efficient computational algorithms for obtaining numerical solution.

In the present paper, we consider the linear two-point boundary value problem of the form

\[ Ly(x) = y''(x) + P(x)y'(x) + Q(x)y(x) = R(x), \]

with

\[ y(a) = b, \]

\[ y(\infty) = c, \quad \text{or} \quad \lim_{x \to \infty} y(x) = c, \]

where \( P(x), Q(x) \) and \( R(x) \) are continuous functions and \( Q(x) > 0 \). We assume that (1.1)-(1.3) have a unique solution \( y(x) \) to be determined. Before computing the solution, we plummet the infinite interval to a finite but large one, so that a finite point represent infinity. This is standard approach of solving such problems that are posed on infinite intervals and named domain truncation. The boundary condition at infinity is replaced with the same conditions at a finite value \( L \). This provides very accurate results provided that \( L \) is sufficiently large. De Hoog and Weiss (De Hoog, F.R., 1979) proposed an analytical transformation of the independent variable that reduces the original problem to a boundary value problem over a finite interval. Usually, that produces a singularity of the second kind at the origin and must be solved by suitable difference methods.

In this paper, we introduce a new nonclassical collocation method for solving (1.1)-(1.3). This method consists of reducing the solution of (1.1)-(1.3) to a set of linear algebraic equations by first expanding \( y(x) \) in terms of Lagrange interpolating polynomials based on a set of nonclassical Gauss-Lobatto (NGL) nodes. These nodes, which arise from nonclassical orthogonal polynomials based on an arbitrary weight function over interval \([a, L]\) are presented. (1.1) is then collocated at these NGL collocation points to evaluate the unknown coefficients, which are the values of the function \( y(x) \) at these collocation points.

This paper is organized as follows: the following section is devoted to the generation of NGL collocation points and function approximation. In section 3, we explain our method and in section 4 we give our numerical findings by considering two examples to demonstrate the accuracy and applicability of the proposed technique.
2. Nonclassical Pseudospectral Method:

2.1. NGL Points and Weights:

In classical pseudospectral methods (Elnagar, 1995; Elnagar, 1998) the classical Gauss-Lobatto collocation points are based on Chebyshev or Legendre polynomials and lie on the closed interval [-1,1]. In the present work, we consider the generation of the NGL collocation points based on nonclassical orthogonal polynomials with respect to arbitrary weights in the interval \([a,L]\).

Let \(N+1\) be the number of collocation points and \(P_N(t)\) be the \(N\)th-degree nonclassical orthogonal polynomial with respect to the weight \(w(t)\) which can be obtained from the following three-term recurrence relation (Shizgal, B., 1981).

\[
P_{k+1}(x) = (x - \alpha_k)P_k(x) - \beta_k P_{k-1}(x), \quad k = 0,1,2,...
\]

\(P_{N+1}(x) = 0, \quad P_0(x) = 1.\)  

The recurrence coefficients in (2.1) are given in (Shizgal, 1996) by

\[
\alpha_k = \frac{\int_a^L w(x)P_k^2(x)dx}{\int_a^L w(x)P_k(x)dx}, \quad k = 0,1,2,...
\]

\[
\beta_k = \frac{\int_a^L w(x)P_k^2(x)dx}{\int_a^L w(x)P_k(x)dx}.
\]

The NGL collocation points \(x_j\) and weights \(w_j\) for \(j = 0,1,...,N\) are obtained by the method outlined by Golub (Golub, G.H., 1973). The tridiagonal Jacobi-Lobatto matrix of order \(N+1\) is defined by

\[
J_{N+1}^L = \begin{pmatrix}
\alpha_0 & \sqrt{\beta_1} & & \\
\sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & \\
& \sqrt{\beta_2} & \alpha_2 & \sqrt{\beta_3} & \\
& & \ddots & \ddots & \ddots & \\
& & & \sqrt{\beta_N} & \alpha_{N+1} & \sqrt{\beta_N} \\
& & & & \sqrt{\beta_N} & \alpha_{N+1}
\end{pmatrix},
\]

where \(\alpha_N^L, \beta_N^L\) are the solution of the \(2 \times 2\) linear system

\[
\begin{pmatrix}
P_n(a) & P_{n-1}(a) \\
P_n(L) & P_{n-1}(L)
\end{pmatrix}\begin{pmatrix}
\alpha_N^L \\
\beta_N^L
\end{pmatrix} = \begin{pmatrix}
\alpha(a) \\
\beta(a)
\end{pmatrix} = \begin{pmatrix}
LP_n(a) \\
LP_{n-1}(a)
\end{pmatrix}.
\]

Theorem 1. (Golub, 1973) The Gauss-Lobatto nodes \(x_0 = a,x_1,...,x_{N-1},x_N = L\) are the eigenvalues of \(J_{N+1}^L\), and the Gauss-Lobatto weights \(w_j\) are given by

\[w_j = \beta_0(v_j)^2, \quad j = 0,1,...,N,\]

where \(v_j\) is the normalized eigenvector of \(J_{N+1}^L\) corresponding to the eigenvalue \(x_j\) (i.e. \(v_j^T v_j = 1\)) and \(v_j\) its first component.

2.2 Function Approximation:

In order to interpolate a function \(f \in L^2[a,L]\) at the point \(x \in [a,L]\), we use the Lagrange interpolation of degree \(N\) of the form

\[f(x) \equiv F_N(x) = \sum_{j=0}^{N} L_j(x)f(x_j), \quad x \in [a,L]\]

(2.2)

where \(x_j, \quad j = 0,1,...,N,\) are a set of NGL collocation points (interpolation nodes) in \([a,L]\) and

\[L_j(x) = \prod_{k=0, k \neq j}^{N} \frac{x-x_k}{x_j-x_k}, \quad j = 0,1,...,N.\]
Differentiating the series of (2.2), two times, and evaluating the result at the collocation points \( x_k \) gives

\[
F_N^{(m)}(x_k) = \sum_{j=0}^{N} \frac{d^m}{dx^m}[L_j(x)]_{x=x_k} f(x_j) = \sum_{j=0}^{N} D_{kj}^{(m)} f(x_j), \quad k = 0, 1, \ldots, N, \quad m = 1, 2, \tag{2.3}
\]

where the coefficients \( D_{kj}^{(m)} \) are entries of an \((N+1) \times (N+1)\) square differentiation matrix.

### 3. Solution of Two-Point Boundary Value Problems:

In order to solve (1.1)-(1.3) with a nonclassical collocation method, we first choose an interval \([a, L]\). \( L \) is any positive integer. Let

\[
y^N(x) = \sum_{j=0}^{N} L_j(x) y(x_j) = y^T L(x), \tag{3.1}
\]

where \( y^T = [y(x_0), y(x_1), \ldots, y(x_N)] \) and \( L(x) = [L_0(x), L_1(x), \ldots, L_N(x)]^T \). Differentiating (3.1) and using (2.3), we obtain

\[
y'^N(x) = \sum_{j=0}^{N} L'_j(x) y(x_j) = y^T D^{(1)}(x), \tag{3.3}
\]

\[
y''^N(x) = \sum_{j=0}^{N} L''_j(x) y(x_j) = y^T D^{(2)}(x), \tag{3.3}
\]

where \( D^{(1)}(x) = [L'_0(x), L'_1(x), \ldots, L'_N(x)]^T \) and \( D^{(2)}(x) = [L''_0(x), L''_1(x), \ldots, L''_N(x)]^T \). By substituting (3.1) – (3.3) in (1.1), we get

\[
Ly^N(x) = y^T D^{(2)}(x) + P(x)y^T D^{(1)}(x) - Q(x)y^T L(x) = R(x), \tag{3.4}
\]

which can be rewritten as

\[
Ly^N(x) = y^T [D^{(2)}(x) + P(x)D^{(1)}(x) - Q(x)L(x)] = R(x). \tag{3.5}
\]

We now collocate (3.5) at \( N - 1 \) NGL collocation points \( x_j, j = 1, \ldots, N - 1, \) as

\[
Ly^N(x_j) = y^T [D^{(2)}(x_j) + P(x_j)D^{(1)}(x_j) - Q(x_j)L(x_j)] = R(x_j), \tag{3.6}
\]

The sets of NGL collocation points are defined on the intervals \([0, L]\). In addition, using (2.2), we get

\[
L(x_j) = e_{j+1},
\]

where \( e_{j+1} \) is an \((N+1) \times 1\) vector whose \((j+1)\)th component is one and other components are zero.

Furthermore, using (3.1), we approximate the boundary conditions in (1.2) – (1.3) as follows:

\[
y^N(a) = y(a) = b, \tag{3.7}
\]

\[
y^N(L) = y(L) = c, \tag{3.8}
\]

where \( L \) is chosen so that the computed solution approximates the actual solution.

Using (3.6) – (3.8), we obtain a system of \( N + 1 \) linear equations. Solving this system and substituting the obtained values of vector \( y^T \) in (3.1) the approximate solutions can be obtained.

### 4. Computational Results:

In this section, we have implanted the present method on two examples. The numerical results are compared with exact results and other results in the literature.

There are many orthogonal weights functions \( w(x) \) that can be used (Maleki, 2010; Chen, 2001). In this paper we use only two weights. The cases are summarized in Table 1.

<table>
<thead>
<tr>
<th>Case</th>
<th>( w(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 1 + x + x^2 )</td>
</tr>
<tr>
<td>2</td>
<td>( e^{-x} )</td>
</tr>
</tbody>
</table>
**Example 1:**

Consider the boundary value problem

\[ Ly(x) = -y'' - 2y' + 2y = e^{-2x}, \quad (4.1) \]

with

\[ y(0) = 1, \quad (4.2) \]

\[ y(x) = 0. \quad (4.3) \]

This problem has earlier been considered in (Kadalbajoo, 1984) and (Ravi Kanth, 2003), and its exact solution is

\[ y_e(x) = \frac{1}{2}(e^{-(1+\sqrt{5})x} + e^{-2x}), \]

The boundary value problem given by (4.1) - (4.3) has been solved using nonclassical collocation method for the cases given in Table 1. The numerical results are presented in Tables 2-3 and compared with the method in (Ravi Kanth, 2003) and the exact solutions. Note that in (Ravi Kanth, 2003), \( N \) defines domain truncation \([0, N]\).

Define

\[ E_N = \max \left\{ \left| y_e(x) - y^N(x) \right| : 0 \leq x \leq L \right\}, \]

where \( y^N(x) \) and \( y_e(x) \) are approximated and exact solution, respectively. In table 4, the maximum absolute errors \( E_N \) for different values of \( N \) and \( L \) are given. Table 4 shows that in this method by changing the weight function \( w(x) \) the obtained results can be improved.

**Table 2:** Computational results of \( y(x) \) for example 1 (\( L = 8 \)).

<table>
<thead>
<tr>
<th>( x )</th>
<th>Exact ( \times 10^1 )</th>
<th>Method in (Ravi Kanth, 2003) ( N = 8, h = \frac{1}{64} )</th>
<th>Present Case 1, ( N = 15 )</th>
<th>Present Case 1, ( N = 20 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.100000</td>
<td>0.100000 \times 10^1</td>
<td>0.100000 \times 10^1</td>
<td>0.100000 \times 10^1</td>
</tr>
<tr>
<td>1.0</td>
<td>0.100210 \times 10^1</td>
<td>0.100220 \times 10^1</td>
<td>0.100209 \times 10^1</td>
<td>0.100210 \times 10^1</td>
</tr>
<tr>
<td>2.0</td>
<td>0.112759 \times 10^1</td>
<td>0.112776 \times 10^1</td>
<td>0.112733 \times 10^1</td>
<td>0.112759 \times 10^1</td>
</tr>
<tr>
<td>3.0</td>
<td>0.137723 \times 10^1</td>
<td>0.137748 \times 10^1</td>
<td>0.137634 \times 10^1</td>
<td>0.137723 \times 10^1</td>
</tr>
<tr>
<td>4.0</td>
<td>0.176704 \times 10^1</td>
<td>0.176737 \times 10^1</td>
<td>0.174839 \times 10^1</td>
<td>0.176706 \times 10^1</td>
</tr>
<tr>
<td>5.0</td>
<td>0.232839 \times 10^1</td>
<td>0.232828 \times 10^1</td>
<td>0.230441 \times 10^1</td>
<td>0.232850 \times 10^1</td>
</tr>
<tr>
<td>6.0</td>
<td>0.311011 \times 10^1</td>
<td>0.311051 \times 10^1</td>
<td>0.290983 \times 10^1</td>
<td>0.310653 \times 10^1</td>
</tr>
<tr>
<td>7.0</td>
<td>0.418238 \times 10^1</td>
<td>0.417943 \times 10^1</td>
<td>0.647676 \times 10^1</td>
<td>0.400832 \times 10^1</td>
</tr>
<tr>
<td>7.5</td>
<td>0.153582 \times 10^2</td>
<td>0.153582 \times 10^2</td>
<td>0.14919 \times 10^2</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3:** Computational results of \( y(x) \) for example 1 (\( L = 10 \)).

<table>
<thead>
<tr>
<th>( x )</th>
<th>Exact ( \times 10^1 )</th>
<th>Method in (Ravi Kanth, 2003) ( N = 10, h = \frac{1}{64} )</th>
<th>Present Case 1, ( N = 20 )</th>
<th>Present Case 1, ( N = 26 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.100000 \times 10^1</td>
<td>0.100000 \times 10^1</td>
<td>0.100000 \times 10^1</td>
<td>0.100000 \times 10^1</td>
</tr>
<tr>
<td>1.0</td>
<td>0.100210 \times 10^1</td>
<td>0.100220 \times 10^1</td>
<td>0.100210 \times 10^1</td>
<td>0.100210 \times 10^1</td>
</tr>
<tr>
<td>2.0</td>
<td>0.112759 \times 10^1</td>
<td>0.112776 \times 10^1</td>
<td>0.112759 \times 10^1</td>
<td>0.112759 \times 10^1</td>
</tr>
<tr>
<td>3.0</td>
<td>0.137723 \times 10^1</td>
<td>0.137748 \times 10^1</td>
<td>0.137723 \times 10^1</td>
<td>0.137723 \times 10^1</td>
</tr>
<tr>
<td>4.0</td>
<td>0.176704 \times 10^1</td>
<td>0.176737 \times 10^1</td>
<td>0.176704 \times 10^1</td>
<td>0.176704 \times 10^1</td>
</tr>
<tr>
<td>5.0</td>
<td>0.232839 \times 10^1</td>
<td>0.232828 \times 10^1</td>
<td>0.232839 \times 10^1</td>
<td>0.232839 \times 10^1</td>
</tr>
<tr>
<td>6.0</td>
<td>0.311011 \times 10^1</td>
<td>0.311051 \times 10^1</td>
<td>0.290983 \times 10^1</td>
<td>0.310653 \times 10^1</td>
</tr>
<tr>
<td>7.0</td>
<td>0.418238 \times 10^1</td>
<td>0.417943 \times 10^1</td>
<td>0.647676 \times 10^1</td>
<td>0.400832 \times 10^1</td>
</tr>
<tr>
<td>7.5</td>
<td>0.153582 \times 10^2</td>
<td>0.153582 \times 10^2</td>
<td>0.14919 \times 10^2</td>
<td></td>
</tr>
</tbody>
</table>

**Example 2:**

A second example

\[ Ly(x) = -y'' + (1 + \frac{1}{x})y' = \frac{1}{x^2}, \quad (4.4) \]

with

\[ y(1) = 0.0, \quad (4.5) \]

\[ y(\infty) = 0.0. \quad (4.6) \]
Table 4: Maximum absolute error $E_N$ for example 1.

<table>
<thead>
<tr>
<th>Case</th>
<th>$E_N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 8$, case 1 and</td>
<td></td>
</tr>
<tr>
<td>$N = 10$</td>
<td>$2.5 \times 10^{-3}$</td>
</tr>
<tr>
<td>$N = 15$</td>
<td>$1.0 \times 10^{-5}$</td>
</tr>
<tr>
<td>$N = 20$</td>
<td>$5.6 \times 10^{-8}$</td>
</tr>
<tr>
<td>$L = 8$, case 2 and</td>
<td></td>
</tr>
<tr>
<td>$N = 10$</td>
<td>$1.8 \times 10^{-2}$</td>
</tr>
<tr>
<td>$N = 15$</td>
<td>$4.5 \times 10^{-5}$</td>
</tr>
<tr>
<td>$N = 20$</td>
<td>$8.6 \times 10^{-8}$</td>
</tr>
<tr>
<td>$L = 10$, case 1 and</td>
<td></td>
</tr>
<tr>
<td>$N = 10$</td>
<td>$5.0 \times 10^{-3}$</td>
</tr>
<tr>
<td>$N = 15$</td>
<td>$6.4 \times 10^{-5}$</td>
</tr>
<tr>
<td>$N = 20$</td>
<td>$2.0 \times 10^{-7}$</td>
</tr>
<tr>
<td>$N = 26$</td>
<td>$1.0 \times 10^{-9}$</td>
</tr>
<tr>
<td>$L = 10$, case 2 and</td>
<td></td>
</tr>
<tr>
<td>$N = 10$</td>
<td>$1.1 \times 10^{-4}$</td>
</tr>
<tr>
<td>$N = 15$</td>
<td>$6.5 \times 10^{-4}$</td>
</tr>
<tr>
<td>$N = 20$</td>
<td>$1.5 \times 10^{-6}$</td>
</tr>
<tr>
<td>$N = 26$</td>
<td>$1.3 \times 10^{-9}$</td>
</tr>
</tbody>
</table>

This example has earlier been considered in (Robertson, T.N., 1971) and later in (Ravi Kanth, 2003). For this problem we consider intervals of the form $[1, L]$. We applied the method presented in this paper and solved (4.4) - (4.6) and then evaluated the different values of $y(x)$, which also were evaluated in (Ravi Kanth, 2003) by using a forth order finite difference method.

The computational results are presented in Tables 5-6, and a comparison is made with the method outlined in (Ravi Kanth, 2003).

Table 5: Computational results of $y(x)$ for example 2 ($L = 16$).

<table>
<thead>
<tr>
<th>$x$</th>
<th>Method in [2] $N = 15$, $h = 1/32$</th>
<th>Present Case 1, $N = 15$</th>
<th>Present Case 1, $N = 25$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>$0.000000 \times 10^1$</td>
<td>$0.000000 \times 10^0$</td>
<td>$0.000000 \times 10^0$</td>
</tr>
<tr>
<td>3.0</td>
<td>$0.109831 \times 10^1$</td>
<td>$0.109817 \times 10^1$</td>
<td>$0.109817 \times 10^1$</td>
</tr>
<tr>
<td>5.0</td>
<td>$0.390872 \times 10^1$</td>
<td>$0.390871 \times 10^1$</td>
<td>$0.390871 \times 10^1$</td>
</tr>
<tr>
<td>7.0</td>
<td>$0.119537 \times 10^1$</td>
<td>$0.119545 \times 10^1$</td>
<td>$0.119545 \times 10^1$</td>
</tr>
<tr>
<td>9.0</td>
<td>$0.545541 \times 10^1$</td>
<td>$0.551633 \times 10^1$</td>
<td>$0.551633 \times 10^1$</td>
</tr>
<tr>
<td>11.0</td>
<td>$0.255120 \times 10^1$</td>
<td>$0.293328 \times 10^1$</td>
<td>$0.293328 \times 10^1$</td>
</tr>
</tbody>
</table>

Table 6: Computational results of $y(x)$ for example 2 ($L = 26$).

<table>
<thead>
<tr>
<th>$x$</th>
<th>Method in [2] $N = 25$, $h = 1/32$</th>
<th>Present Case 1, $N = 20$</th>
<th>Present Case 1, $N = 30$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>$0.000000 \times 10^1$</td>
<td>$0.000000 \times 10^0$</td>
<td>$0.000000 \times 10^0$</td>
</tr>
<tr>
<td>3.0</td>
<td>$0.852347 \times 10^1$</td>
<td>$0.852345 \times 10^1$</td>
<td>$0.852345 \times 10^1$</td>
</tr>
<tr>
<td>5.0</td>
<td>$0.390872 \times 10^1$</td>
<td>$0.390875 \times 10^1$</td>
<td>$0.390875 \times 10^1$</td>
</tr>
<tr>
<td>7.0</td>
<td>$0.119572 \times 10^1$</td>
<td>$0.119571 \times 10^1$</td>
<td>$0.119571 \times 10^1$</td>
</tr>
<tr>
<td>9.0</td>
<td>$0.545541 \times 10^1$</td>
<td>$0.551633 \times 10^1$</td>
<td>$0.551633 \times 10^1$</td>
</tr>
<tr>
<td>11.0</td>
<td>$0.255120 \times 10^1$</td>
<td>$0.293328 \times 10^1$</td>
<td>$0.293328 \times 10^1$</td>
</tr>
<tr>
<td>15.0</td>
<td>$0.844953 \times 10^1$</td>
<td>$0.103544 \times 10^2$</td>
<td>$0.103442 \times 10^2$</td>
</tr>
</tbody>
</table>
Conclusion:
A nonclassical collocation method has been used for the approximate solution of two-point boundary value problems over infinite intervals. The orthogonal weight function $W$ and the interval of definition of orthogonal polynomials can be chosen arbitrarily which make this method computationally very attractive, because a variety range of Gauss-Lobatto collocation points can be utilized. Examples show the efficiency and accuracy of this method.

REFERENCES