Spectral collocation method and Darvishi’s preconditionings for Tchebychev-Gauss-Lobatto points

M. T. Darvishi

Department of Mathematics, Razi University
Kermanshah 67149, Iran
and
Islamic Azad University
Branch of Kermanshah, Iran

Abstract

In this paper, we present central, left and right Darvishi’s preconditionings to reduce roundoff error in matrix vector multiplication method. This method is applied to compute derivative(s) of an unknown function in spectral collocation method. We show mentioned preconditionings reduce roundoff error in computing derivatives of functions as well, because errors of derivatives are very small.

Mathematics Subject Classification: 65N55

Keywords: Darvishi’s preconditionings; Differentiation matrix; Spectral methods; Tchebychev points

1 Introduction

A problem frequently encountered by scientists and engineers is the solution of partial differential equations (PDEs). Since these equations often do not have closed form solutions, they are often forced to compute solutions numerically. One of the important methods to solve PDEs numerically is weighted residual methods (WRMs). In these methods we obtain a finite system of equations by discretizing the solution space. The obtained approximate solution is a finite combination of known and selected functions. Weighted residuals method is a generic name of the class of discretization schemes for differential equations.
The key elements of the WRM are the trial functions and the test functions. The trial functions are used as the basis functions for a truncated series expansion of the solution. The test functions are used to ensure that the differential equation is satisfied as closely as possible by the truncated series expansion. A list of the known of WRMs, for example, are: subdomain, collocation, least squares, Bubnou-Galerkin, spectral methods and so on. The subject is described very well in [4, 5, 6]. In this article we briefly explain spectral method or more precisely, spectral collocation method and then we concentrate on computing derivative(s) by matrix vector multiplication method.

2 Spectral collocation method

If a function \( u(x) \) is not periodic, we can approximate it by polynomials in \( x \). However, it is well known that the Lagrange interpolation polynomial based on equally spaced points does not give a satisfactory approximation to general smooth \( u \). In fact, as the number of collocation points increases, interpolation polynomial diverges [6]. This poor behavior of polynomial interpolation can be avoided for smoothly differentiable functions by removing the restriction to equally spaced collocation points. Good results are obtained by relating the collocation points to the structure of classical orthogonal polynomials, like Tchebychev and Legendre polynomials. In the most common spectral collocation Tchebychev method, the interpolation points in the interval \([-1, 1]\) are the Tchebychev-Gauss-Lobatto collocation points \( x_j = \cos\left(\frac{j\pi}{n}\right) \) for \( j = 0, \ldots, n \), which are the extreme of the \( n \)th order Tchebychev polynomials \( T_n(x) = \cos(n \arccos x) \). In order to construct the interpolant of \( u(x) \) at the point \( x \) the following polynomials are defined

\[
g_j(x) = \frac{(-1)^{j+1}(1 - x^2)T_n'(x)}{c_j n^2 (x - x_j)}, \quad j = 0, \ldots, n, \quad (1)
\]

where \( c_0 = c_n = 2 \) and \( c_j = 1 \) for \( j = 1, 2, \ldots, n - 1 \). The interpolation polynomial, \( u_n(x) \), to \( u(x) \) is given by

\[
u_n(x) = \sum_{j=0}^{n} a_j g_j(x). \quad (2)
\]

Then in getting a spectral collocation approximation we have to express the derivatives of \( u_n(x) \) in terms of \( u(x) \) at the collocation points \( x_j \). This can be done by differentiating equation (2), i.e.,

\[
u_n^{(r)}(x) = \sum_{j=0}^{n} g_j^{(r)}(x) u(x_j), \quad r = 1, 2, \ldots, \quad (3)
\]
so that
\[ u_n^{(r)}(x_k) = \sum_{j=0}^{n} d_{kj}^{(r)} u(x_j), \quad r = 1, 2, \ldots, \tag{4} \]
where \( d_{kj}^{(r)} = g_j^{(r)}(x_k) \), are the elements of differentiation matrix \( D_r \). The elements of \( D_r \) can be obtained analytically (for detail see, e.g., [6]). In fact, in equation (4) we use matrix vector multiplication method to find an approximation for the differential operator in terms of the grid point values of \( u_n \), which we explain it in the following section.

3 Matrix vector multiplication method

If \( \vec{u} = \{u(x_i)\} \), is the vector consisting values of \( u(x) \) at the \( n + 1 \) collocation points and \( \vec{u}' = \{u'(x_i)\} \) consists values of the derivatives at the collocation points, then the collocation derivative matrix \( D \), is the matrix mapping \( \vec{u} \mapsto \vec{u}' \). Two matrix multiplications yield \( \vec{u}'' \), the vector containing the second derivatives evaluated at the collocation points. More efficiently the matrix \( D_2 \) maps \( \vec{u} \mapsto \vec{u}'' \). It can be shown that the Tchebychev derivative, when computing the derivative using the matrix vector multiplication method, is a rather ill-conditioned operator, and inaccuracies in the function can be magnified by as much as \( O(n^4) \) where \( n \) is the number of collocation points. Hence, some works to improve the method are studied. These works concentrated on the problem of roundoff error in Tchebychev collocation methods and various algorithms have been suggested to reduce it. The best result for the matrix vector multiplication algorithm managed to reduce the roundoff error from \( O(n^4\varepsilon) \) to \( O(n^3\varepsilon) \), where \( \varepsilon \) is the machine precision (see [7, 8]). This formulation should result in more accurate entries of the matrix.

Some researchers have worked on the problem of reducing roundoff errors in Tchebychev collocation derivative methods. Baltensperger and Trummer [2] demonstrated that naive algorithms for computing these matrices suffer from severe loss of accuracy due to roundoff errors. Breuer and Everson [3] introduced a preconditioning to reduce roundoff error by making the value of the function on the boundaries vanish. In [10], the author attempts to combat roundoff error by preconditioning the problem. Tang and Trummer in [9] use trigonometric identities and a flipping trick to reduce round off errors. Don and Solomonoff [7] attempted to reduce the roundoff error using trigonometric identities for rewriting components of derivative matrix as follows:
\[
d_{kj} = \frac{1}{2} \frac{c_k}{c_j} \sin\left(\frac{\pi}{2n}(k+j)\right) \sin\left(\frac{\pi}{2n}(k-j)\right), \quad k \neq j,
\]
\[
d_{kk} = -\frac{1}{2} \frac{x_k}{\sin^{2}\left(\frac{\pi}{2n}\right)} 
\]
\[
d_{00} = -d_{nn} = \frac{2n^2+1}{6}.
\]

Formula (5), which avoid differencing of nearly-equal numbers, have been introduced to reduce this source of error from \(O(n^4 \varepsilon)\) to \(O(n^3 \varepsilon)\). Don and Solomonoff [7] show that, even with utilization of (5), the error incurred in the evaluation of \(Du\) near \(x = -1\) is significantly larger than at \(x = 1\), even if \(u\) is symmetric (related to the accuracies achieved in evaluating \(\sin(x)\) and \(\sin(\pi - x)\) for small \(x\)). In other words, if \(k\) and \(j\) are small then \(d_{kj}\) can be computed accurately whilst if \(k\) and \(j\) are near \(N\) then the evaluation of \(d_{kj}\) is less accurate. This can be utilized by evaluating \(d_{kj}\) in the upper half of the matrix and then ‘flipping’ to take advantage of the symmetry property \(d_{kj} = -d_{n-k,n-j}\). This formula gives the bottom half of the matrix with smaller cancelation error than equation (5), (see [1]).

4 Preconditionings

In this part central, left and right Darvishi’s preconditionings are described.

Central Darvishi’s preconditionings (CDP). From (5) for \(k \neq j\) we have

\[
|d_{kj}| = \frac{c_k}{2c_j} \frac{1}{\left|\sin(\pi \frac{k+j}{2n}) \sin(\pi \frac{k-j}{2n})\right|}
\]  

(6)
since the value of \(\sin(\pi \frac{k-j}{2n})\) is near zero when \(k\) is near \(j\), hence this causes \(|d_{kj}|\) become large for \(k\) near \(j\). This means entries of derivative matrix \(D\), with large absolute values, are on a band near main diagonal. That is large values of \(|d_{kj}|\) correspond to values of \(k\) near \(j\). Therefore matrix vector multiplication method causes large roundoff error. To reduce roundoff error in \(k\)th node, we define \(h_k(x)\) as follows:

\[
h_k(x) = u(x) - u(x_k),
\]  

(7)
where \(x_k = \cos(\frac{k\pi}{n})\). From \(u(x) = \sum_{j=0}^{n} g_j(x)u_j\) we have

\[
h_k(x) = \sum_{j=0}^{n} g_j(x)h_k(x_j)
\]  

(8)
hence, from (3) the derivative of \(h_k\) at \(x = x_k\) is as follows

\[
h_k'(x_k) = \sum_{j=0}^{n} d_{kj}h_k(x_j)
\]  

(9)
or

\[ u'(x_k) = \sum_{j=0}^{n} d_{kj}(u(x_j) - u(x_k)). \]  

(10)

Therefore, by using this preconditioning, we can reduce the influence of large values of \(|d_{kj}|\), in the matrix vector multiplication method. In the following section we apply matrix vector multiplication method using this preconditioning to compute derivative of some test functions.

**Left and right Darvishi’s preconditionings (LDP and RDP).** As stated in previous part, for \( k \neq j \) as we have equation (6), the entries of the derivative matrix \( D \), with large absolute value, are on a band near main diagonal. In fact the values of \(|d_{kj}|\) in (6) can be very large when \( k \) is near \( j \). Particularly, if \(|k - j| = 1\) the value of \(|d_{kj}|\) is very large. This means large elements of the derivative matrix, in absolute value, except \( d_{00} \) and \( d_{nn} \), are the following elements:

\[
\begin{align*}
|d_{k,k-1}|, & \quad k = 1, \ldots, n, \\
|d_{k,k+1}|, & \quad k = 0, \ldots, n - 1.
\end{align*}
\]

Therefore, if these elements are multiplied by zero, the influence of these big elements in roundoff error, will be vanished. To reduce roundoff error in \( k \)th node, we define the following functions:

\[
\begin{align*}
h_-(x) &= u(x) - u(x_{k-1}), \quad k = 1, \ldots, n, \\
h_+(x) &= u(x) - u(x_{k+1}), \quad k = 0, \ldots, n - 1,
\end{align*}
\]

where \( x_k = \cos(\frac{k\pi}{n}) \), note that \( u'(x) = h'_-(x) = h'_+(x) \). Similar (9), from these functions we have

\[
\begin{align*}
h'_-(x_k) &= \sum_{j=0}^{n} d_{kj}h_-(x_j), \quad k = 1, \ldots, n, \\
h'_+(x_k) &= \sum_{j=0}^{n} d_{kj}h_+(x_j), \quad k = 0, \ldots, n - 1,
\end{align*}
\]

or

\[
\begin{align*}
u'(x_k) &= \sum_{j=0}^{n} d_{kj}(u(x_j) - u(x_{k-1})), \quad k = 1, \ldots, n, \\
u'(x_0) &= \sum_{j=0}^{n} d_{0j}(u(x_j) - u(x_0)),
\end{align*}
\]

and

\[
\begin{align*}
u'(x_k) &= \sum_{j=0}^{n} d_{kj}(u(x_j) - u(x_{k+1})), \quad k = 0, \ldots, n - 1, \\
u'(x_n) &= \sum_{j=0}^{n} d_{nj}(u(x_j) - u(x_n)).
\end{align*}
\]

(11)

In general to compute \( u'(x_k) \) for \( k = 0, \ldots, n \), we propose the following formulas:

\[
\begin{align*}
u'(x_k) &= \sum_{j=0}^{n} d_{kj}(u(x_j) - u(x_{k-1})), \quad k = 1, \ldots, n, \\
u'(x_0) &= \sum_{j=0}^{n} d_{0j}(u(x_j) - u(x_0)),
\end{align*}
\]

and

\[
\begin{align*}
u'(x_k) &= \sum_{j=0}^{n} d_{kj}(u(x_j) - u(x_{k+1})), \quad k = 0, \ldots, n - 1, \\
u'(x_n) &= \sum_{j=0}^{n} d_{nj}(u(x_j) - u(x_n)).
\end{align*}
\]

(12)
We call the preconditioning in (11) as left preconditioning and one in (12) as right preconditioning. The effect of these preconditionings are shown on some test functions in the following section. The numerical results are reported in Tables 1-4, in which the value of maximum error of the first order derivative as a function of number of collocation points, \( n \), is demonstrated.

5 Numerical examples

In this section, we compute first and second derivatives of some test functions by matrix vector multiplication method with and without preconditionings. As Tables 1-4 show the results are very good. For first derivative we measure the error in the numerical approximation \( u_n \) with maximum-error or \( L_\infty \)-error

\[
\| e \|_\infty = \max_{0 \leq k \leq n} |u'(x_k) - u'_n(x_k)|,
\]

for all test functions.

Central, left and right preconditionings are very well for higher derivatives. We use matrix vector multiplication method using our preconditionings to compute second derivatives of the test functions. For second derivative we measure the error in the numerical approximation \( u_n \) with maximum-error or \( L_\infty \)-error

\[
\| e \|_\infty = \max_{0 \leq k \leq n} |u''(x_k) - u''_n(x_k)|,
\]

for all test functions.

Example 1. We compute first derivative of some test functions, namely, 
\[ u(x) = e^{(x^2/0.3)} + \cos(2x), \quad u(x) = \cos(3x), \quad u(x) = \frac{1}{1+x^2} \quad \text{and} \quad u(x) = \frac{\sin(8x)}{(x+1.1)^{3/2}} \]
by matrix vector multiplication method using central preconditioning. Table 1 shows the error of the derivatives for different collocation points with and without preconditioning.

Example 2. We compute first derivative of previous test functions, namely, 
\[ u(x) = e^{(x^2/0.3)} + \cos(2x), \quad u(x) = \cos(3x), \quad u(x) = \frac{1}{1+x^2} \quad \text{and} \quad u(x) = \frac{\sin(8x)}{(x+1.1)^{3/2}} \]
by matrix vector multiplication method using left and right preconditionings. Table 2 shows the error of the derivatives for different collocation points.

Example 3. We compute second derivative of our test functions, namely, 
\[ u(x) = e^{(x^2/0.3)} + \cos(2x), \quad u(x) = \cos(3x), \quad u(x) = \frac{1}{1+x^2} \quad \text{and} \quad u(x) = \frac{\sin(8x)}{(x+1.1)^{3/2}} \]
by matrix vector multiplication method using central preconditioning. Table 3 shows the error of the derivatives for different collocation points with and
without preconditioning.

**Example 4.** We compute second derivative of our test functions, namely,
\[ u(x) = e^{(x^2/0.3)} + \cos(2x), \quad u(x) = \cos(3x), \quad u(x) = \frac{1}{1+x^2} \text{ and } u(x) = \frac{\sin(8x)}{(x+1)^{3/2}} \]
by matrix vector multiplication method using left and right preconditionings.
Table 4 shows the error of the derivatives for different collocation points.

<table>
<thead>
<tr>
<th>n</th>
<th>without Precond.</th>
<th>with Precond.</th>
<th>without Precond.</th>
<th>with Precond.</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>1.32729e-11</td>
<td>5.05906e-12</td>
<td>5.68873e-13</td>
<td>2.32037e-14</td>
</tr>
<tr>
<td>64</td>
<td>8.5904e-11</td>
<td>1.23634e-11</td>
<td>3.41016e-12</td>
<td>5.69544e-14</td>
</tr>
<tr>
<td>128</td>
<td>1.17271e-09</td>
<td>5.40581e-11</td>
<td>3.91078e-11</td>
<td>8.23286e-13</td>
</tr>
<tr>
<td>256</td>
<td>3.89786e-08</td>
<td>1.42165e-10</td>
<td>1.39607e-09</td>
<td>5.01377e-13</td>
</tr>
<tr>
<td>512</td>
<td>4.72096e-08</td>
<td>3.45466e-10</td>
<td>1.68596e-09</td>
<td>3.50092e-12</td>
</tr>
<tr>
<td>1024</td>
<td>2.16787e-06</td>
<td>2.11628e-09</td>
<td>7.77572e-08</td>
<td>3.47093e-11</td>
</tr>
</tbody>
</table>

Table 2. Absolute maximum error of the first order derivatives of four test functions, using left and right preconditionings, for n collocation points.
<table>
<thead>
<tr>
<th>n</th>
<th>( u(x) = e^{(x^2/0.3)} + \cos(2x) )</th>
<th>( u(x) = \cos(3x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>9.20241,-04</td>
<td>5.18561,-11</td>
</tr>
<tr>
<td>32</td>
<td>5.05906,-12</td>
<td>2.28706,-14</td>
</tr>
<tr>
<td>64</td>
<td>1.23634,-11</td>
<td>5.69544,-14</td>
</tr>
<tr>
<td>128</td>
<td>5.40865,-11</td>
<td>8.25562,-13</td>
</tr>
<tr>
<td>256</td>
<td>1.42165,-10</td>
<td>5.01377,-13</td>
</tr>
<tr>
<td>512</td>
<td>3.45466,-10</td>
<td>3.50331,-12</td>
</tr>
<tr>
<td>1024</td>
<td>2.11628,-09</td>
<td>3.47093,-11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n</th>
<th>( u(x) = \frac{1}{1+x^2} )</th>
<th>( u(x) = \frac{\sin(8x)}{(x+1)^{3/2}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1.69912,-05</td>
<td>5.96210,+01</td>
</tr>
<tr>
<td>32</td>
<td>2.55004,-11</td>
<td>1.23353,-02</td>
</tr>
<tr>
<td>64</td>
<td>1.60427,-13</td>
<td>2.22344,-08</td>
</tr>
<tr>
<td>128</td>
<td>3.98126,-13</td>
<td>6.27551,-11</td>
</tr>
<tr>
<td>256</td>
<td>5.72986,-13</td>
<td>5.57804,-10</td>
</tr>
<tr>
<td>512</td>
<td>1.24054,-11</td>
<td>1.55808,-09</td>
</tr>
<tr>
<td>1024</td>
<td>2.95733,-11</td>
<td>5.57020,-09</td>
</tr>
</tbody>
</table>

Table 3. Absolute maximum error of the second derivatives of four test functions, with and without central preconditoning, for \( n \) collocation points.
Table 4. Absolute maximum error of the second order derivatives of four test functions, using left and right preconditionings, for $n$ collocation points.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$u(x) = e^{(x^2/0.3)} + \cos(2x)$</th>
<th>$u(x) = \cos(3x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1.60601,-01</td>
<td>8.96532,-09</td>
</tr>
<tr>
<td>32</td>
<td>9.96788,-01</td>
<td>1.21005,-11</td>
</tr>
<tr>
<td>64</td>
<td>1.79282,-08</td>
<td>1.10173,-10</td>
</tr>
<tr>
<td>128</td>
<td>3.28252,-07</td>
<td>4.17304,-09</td>
</tr>
<tr>
<td>256</td>
<td>1.82367,-06</td>
<td>2.15787,-09</td>
</tr>
<tr>
<td>512</td>
<td>1.11201,-05</td>
<td>1.13023,-07</td>
</tr>
<tr>
<td>1024</td>
<td>9.32096,-04</td>
<td>8.32180,-06</td>
</tr>
<tr>
<td></td>
<td>$u(x) = \frac{1}{1+x^2}$</td>
<td>$u(x) = \frac{\sin(8x)}{(x+11)^{1/2}}$</td>
</tr>
<tr>
<td>16</td>
<td>2.90549,-03</td>
<td>1.14782,+03</td>
</tr>
<tr>
<td>32</td>
<td>1.74246,-08</td>
<td>8.68412,+00</td>
</tr>
<tr>
<td>64</td>
<td>2.04392,-10</td>
<td>6.11535,-05</td>
</tr>
<tr>
<td>128</td>
<td>9.67002,-10</td>
<td>1.98059,-07</td>
</tr>
<tr>
<td>256</td>
<td>1.69621,-08</td>
<td>1.10155,-05</td>
</tr>
<tr>
<td>512</td>
<td>1.02686,-06</td>
<td>9.54697,-05</td>
</tr>
<tr>
<td>1024</td>
<td>1.04288,-05</td>
<td>2.48502,-03</td>
</tr>
</tbody>
</table>

6 Conclusion

We have discussed roundoff errors incurred when calculating the spectral collocation method for Tchebychev-Gauss-Lobatto points and we have suggested three preconditionings to reduce these errors.

References


Received: December 12, 2006