QUARTIC NON-POLYNOMIAL SPLINE SOLUTION FOR SOLVING TWO-POINT BOUNDARY VALUE PROBLEMS BY USING CONJUGATE GRADIENT ITERATIVE METHOD

Hynichearry Justine, Jackel Vui Ling Chew, Jumat Sulaiman
Mathematics with Economics Programme, University Malaysia Sabah
Sabah, Malaysia
cherryjust20@gmail.com, jackelchew93@gmail.com, jumat@ums.edu.my
Received: 23 November 2016; accepted: 10 February 2017

Abstract. Solving two-point boundary value problems has become a scope of interest among many researchers due to its significant contributions in the field of science, engineering, and economics which is evidently apparent in many previous literary publications. This present paper aims to discretize the two-point boundary value problems by using a quartic non-polynomial spline before finally solving them iteratively with Conjugate Gradient (CG) method. Then, the performances of the proposed approach in terms of iteration number, execution time and maximum absolute error are compared with Gauss-Seidel (GS) and Successive Over-Relaxation (SOR) iterative methods. Based on the performances analysis, the two-point boundary value problems are found to have the most favorable results when solved using CG compared to GS and SOR methods.

MSC 2010: 34B05
Keywords: two-point boundary value problems, quartic non-polynomial spline, Conjugate Gradient, Successive Over-Relaxation, Gauss-Seidel

1. Introduction

Numerical methods have numerous significances in the field of sciences, economics, and engineering, and one of them when it comes to the solution of two-point boundary value problems which involve finding an approximate solution iteratively, as it will be time-consuming to solve them with analytical method. Some of the contributions of numerical methods related to the two-point boundary value problems include the modelling of chemical reactions and the modelling of heat transfer, such as in rocket thrust chamber liners and in the fuel elements for nuclear reactors as discussed by Ozisik [1]. On the other hand, Goffe [2] mentioned the modelling of the growth theory, capital theory, investment theory, resource economics and labor economics in the field of economics. Prior to these, many researchers had attempted to initiate different methods in order to accelerate the approximate solution when solving the problems and this can be abundantly
found in previous literary publications. Some of the methods that were readily apparent are the Newton-EGMSOR method [3], the EADM method [4], the shooting method [5], the PTI method [6], the nonlinear shooting method [7], the mean weight method [8], the finite difference, the finite element and the finite volume method [9] and the spline method [10, 11]. Despite all these methods, the solution in this paper was given focus based on the discretization of the quartic non-polynomial spline together with the Conjugate Gradient (CG) iterative method.

Moreover, there are many other iterative methods which are thoroughly discussed by Kelly [12], Burgerscentrum [13], Hestenes and Stiefel [14], Saad [15], Hackbush [16] and Young [17, 18]. According to Ibrahim and Abdullah [19], and Yousif and Evans [20], there are several numbers of the iterative methods family with a different concept, and they emphasized the concept of block iteration. In addition to that, Ul-Islam et al. [21], Ramadan et al. [22], Siddiqi et al. [23] have provided the basis for this paper at a different degree of splines to solve the two-point boundary value problems. In regards to the advantages of the CG iterative method and the spline approach as highlighted in [14] and [21-23] respectively, this present paper aims to develop a solution for the problems by using a quartic non-polynomial spline together with the CG iterative method. As for comparison purposes, Successive Over-Relaxation (SOR) and Gauss-Seidel (GS) were set as control methods so that the performances of the CG method can be determined in respect to its iteration number, execution time and maximum absolute error.

2. Two-point boundary value problems

Generally, the two-point boundary value problems can be expressed as Eq. (1) and subject to boundary conditions (2) as follows:

$$y'' + f(x)y' + g(x)y = g(x), \quad x \in [a, b], \quad (1)$$

$$y(a) = A_1, \quad y(b) = A_2 \quad (2)$$

where $f(x)$, $q(x)$ and $g(x)$ are known functions restricted by boundary $[a, b]$ and $A_i, i = 1, 2$ is a constant. The solution for problem (1) cannot be obtained through a random selection of functions $f(x)$, $q(x)$ and $g(x)$ due to the restrictions held by the boundary conditions (2). Furthermore, the process for discretization of problem (1) is made simpler by assuming positive integer $m = 2^p, p \geq 2$ and letting the solution domain, $[a, b]$ be divided uniformly into a uniform separation of nodes set or subinterval, $m$, as shown in Figure 1. Then, the length of the uniform subintervals, $\Delta x$ can be defined as:

$$\Delta x = \frac{b-a}{m} = h, \quad n = m - 1. \quad (3)$$
Moreover, the solution domain in Figure 1 was used to develop a uniform grid of \( a \) the network as shown in Figure 2 for the derivation of the spline function, and the grid points in the solution domain were labeled as \( x_i = a + ih, \ i = 0, 1, 2, \ldots, m \) with function \( y(x) \) denoted as \( y(x_i) = y_i \). The formulation and implementation of the GS, SOR and CG iterative method were then conducted based on the interior grid points until the convergence test is satisfied.

3. Quartic non-polynomial spline approximation equation

The general form of the non-polynomial spline can be expressed as follows:

\[
S(x) = Q_i(x), \quad x \in [x_i, x_{i+1}], \quad i = 0, 1, 2, \ldots, n,
\]

and it was used to discretize the problem (1) so that the approximation equation can be constructed as a system of linear equations in a matrix form. This discretization process was conducted by assuming \( y(x) \) as the exact solution for problem (1) and \( S_i \) as the quartic non-polynomial spline approximation to \( y_i = y(x_i) \) obtained from the mixed splines \( Q(x) \) as shown in Figure 2 which passing through the points \( (x_i, S_i) \) and \( (x_{i+1}, S_{i+1}) \). Based on Eq. (4), the quartic non-polynomial spline can be expressed in \( Q(x) \) as:
where \( a_i, b_i, c_i, d_i \) and \( e_i \) are constants for \( i = 0, 1, 2, \ldots, n \) and \( k \) is a free parameter \([19]\). The function \( Q_i(x) \) interpolates \( y(x) \) at the points \( x_i \) by depending on \( k \) and reducing to quartic spline in \([a,b]\) as \( k \to 0 \).

Then, in order to obtain the necessary conditions for all the constants \( a_i, b_i, c_i, d_i \) and \( e_i \), the function \( Q_i(x) \) has been satisfied at \( x_i \) and \( x_{i+1} \), at boundary conditions (2) and at the continuity of the common nodes at \((x_i, y_i)\) of first, second and third derivatives. Before deriving the expression for all the coefficients of (6) in terms of \( F_i, F_{i+1}, D_i, D_{i+1}, S_i \) and \( S_{i+1} \), we first define the function \( Q_i(x) \) at second and fourth derivatives as:

\[
Q_i(x_i) = S_i, \quad Q_i(x_{i+1}) = S_{i+1}, \quad Q_i''(x_i) = D_i, \quad Q_i''(x_{i+1}) = D_{i+1},
\]

\[
Q_i^{(iv)}(x_i) = \frac{1}{2} (F_i + F_{i+1})
\]

After performing a straightforward calculation, all the values (7) of constant \( a_i, b_i, c_i, d_i \) and \( e_i \) were obtained as follows:

\[
a_i = \frac{1}{2} k^2 F_i + \frac{1}{2k^2} F_{i+1} - \frac{1}{2k^2} D_i + \frac{1}{2k^2} D_{i+1},
\]

\[
b_i = \frac{1}{2} \cos \theta \sin \theta F_i + \frac{1}{2k^2} \cos \theta \sin \theta F_{i+1} + \frac{1}{2k^2} \sin \theta D_i - \frac{1}{2k^2} \cos \theta D_{i+1},
\]

\[
c_i = \frac{1}{4k^2} F_i + \frac{1}{4k^2} F_{i+1} + \frac{1}{4} D_i + \frac{1}{4} D_{i+1},
\]

\[
d_i = \frac{1}{h} S_{i+1} - \frac{1}{h} S_i - \frac{h}{4k^2} F_i - \frac{h}{4k^2} F_{i+1} - \frac{4 - h^2 k^2}{4h k^2} D_i - \frac{4 - h^2 k^2}{4h k^2} D_{i+1},
\]

\[
e_i = S_i - \frac{1}{2k^2} F_i - \frac{1}{2k^2} F_{i+1} + \frac{1}{2k^2} D_i - \frac{1}{2k^2} D_{i+1},
\]

where \( \theta = k h \) and \( i = 0, 1, 2, \ldots, N \).

Now that all the values of constant \( a_i, b_i, c_i, d_i \) and \( e_i \) were obtained, we then use the continuity conditions (2) of the quartic spline \( S_i \) at its first and third derivatives at the point \((x_i, S_i)\) where the two quartics \( Q_i^{m}(x) \) and \( Q_{i+1}^{m}(x) \) join, and this relation can be written as:

\[
Q_i^{m}(x) = Q_{i+1}^{m}(x),
\]

where the degree of the derivative is \( m = 0, 1 \).
Based on Eqs. (5) and (7), the relation at the first derivative (8) and the third derivative (9) can be expressed in the following form:

\[
F_{i+1} + 2F_i + F_{i-1} = \frac{4hk^2 + 4hk^2 \cos \theta - 8k \sin \theta - 2h^2 k^3 \sin \theta}{-2h + 2h \cos \theta + h^2 k \sin \theta} D_i - \frac{2hk^2 - 2hk^2 \cos \theta + 4k \sin \theta - h^2 k^3 \sin \theta}{-2h + 2h \cos \theta + h^2 k \sin \theta} D_{i+1} + \frac{2hk^2 - 2hk^2 \cos \theta + 4k \sin \theta - h^2 k^3 \sin \theta}{-2h + 2h \cos \theta + h^2 k \sin \theta} S_{i+1} + \frac{8k^3 \sin \theta}{-2h + 2h \cos \theta + h^2 k \sin \theta} S_i + \frac{4k^3 \sin \theta}{-2h + 2h \cos \theta + h^2 k \sin \theta} S_{i-1} \tag{8}
\]

Upon subtraction of Eqs. (8) and (9), it yields the following equation:

\[
F_{i+1} + 2F_i + F_{i-1} = -\frac{2k^2(1 + \cos \theta)}{1 - \cos \theta} D_i + \frac{k^2(1 + \cos \theta)}{1 - \cos \theta} D_{i+1} + \frac{k^2(1 + \cos \theta)}{1 - \cos \theta} D_{i-1} \tag{9}
\]

Then, a system of linear equations is constructed based on (10) in the following form:

\[
Ay = F \tag{11}
\]

where

\[
A = \begin{bmatrix}
b_1 & c_1 & \\
\vdots & \ddots & \ddots \\
a_{n-1} & b_{n-1} & c_{n-1} \\
a_n & b_n & \end{bmatrix}
\]

\[
y = [y_1, y_2, \ldots, y_{n-1}, y_n] \text{ and } F = [F_1 - a_1 y_0, F_2, \ldots, F_{n-1}, F_n - c_n y_{n+1}]^\top.
\]
4. Algorithm of CG method

The CG iterative method was first discussed by Hestenes and Stiefel [14] as mentioned earlier, and this iterative method will converge in less than or equal to the size of the matrix itself in the absence of round-off error, with the assumption that matrix $A$ in (11) is symmetric and positive definite. In fact, this method surpassed the Gauss elimination method. In addition to that, the CG method is much simpler to code when it comes to computer programming and requires less storage space due to its ability to maintain the particular matrix throughout the implementation and improvement which occur at each step of the estimations. In other words, the original data can be used to its maximum. Owing to these advantages of the CG method, the present paper aims to examine its performances in comparison with another two iterative methods which are Successive Over Relaxation (SOR) and Gauss-Seidel (GS) when solving the two-point boundary value problems using the quartic non-polynomial scheme.

By referring to (11), the CG method can be formulated by computing the sequence of $n$ vectors $p_0, p_1, \ldots, p_{n-1}$ which are elements of $\mathbb{R}^n$ that satisfy the following conditions

$$p_i^T A p_j = 0, \quad i \neq j$$

and at the same time matrix $A$ is assumed as $n \times n$ symmetrical matrix. As for method GS and SOR, the formulation begins by decomposing the matrix $A$ as

$$A = D + L + U$$

where $D$ is a diagonal matrix, $L$ is a lower triangular matrix and $U$ is upper triangular matrix. Upon imposing (13) onto (11), the formulation of GS and SOR methods can be written as (14) and (15) respectively.

$$y^{(k+1)} = (1 - \omega) y^{(k)} + \omega (D + L)^{-1} \left(-U y^{(k)} + F \right)$$

$$y^{(k+1)} = -(D + L)^{-1} U y + (L + D)^{-1} F$$

To facilitate the convergence rate of the SOR method, the value of the parameter $\omega$ must be obtained first through several computer programs in the range of $1 \leq \omega \leq 2$. The optimal value of the parameter $\omega$ is selected based on the smallest iterations number. As for the GS method, the value of the parameter is equal to one if we reduce (14) to the GS method. Since both the GS and SOR methods are implemented for comparison purpose only, therefore only the algorithm for the CG method is presented.
The Algorithm of CG Method

i. Initialize $x_0$.

ii. Compute the residual $r_0 = f - Ax_0$ and choose a direction of $p_0 = r_0 - f^T A x_0$.

iii. Obtain the new $x_i$, $r_i = f - Ax_i$ and the direction $p_i$ then compute the new estimate $x_{i+1}$ and its residual $r_{i+1}$ by using the formulas

$$\alpha_i = \frac{r_i^T r_i}{p_i^T A p_i}, \quad x_{i+1} = x_i + \alpha_i p_i, \quad r_{i+1} = r_i - \alpha_i A p_i$$

iv. Next find the direction of $p_{i+1}$ by using the formulas and repeat step (iii)

$$p_{i+1} = r_{i+1} - \beta_i p_i \quad \text{where} \quad \beta_i = \frac{r_{i+1}^T r_{i+1}}{r_i^T r_i}$$

v. Check the convergence. If yes, go to step (vi). Otherwise go back to step (iii).

vi. Display the approximate solutions.

4. Numerical experiment

In order to verify the performances of the CG iterative method, a numerical experiment is conducted by solving the following two-point boundary value problems [8].

**Problem 1**

$$y'' - 4y = 4 \cosh(1), \quad x \in [0, 1] \quad (16)$$

given that the exact solution for problem (16) is

$$y(x) = \cosh(2x - 1) - \cosh(1).$$

**Problem 2**

$$-\frac{d^2 y}{dx^2} = 9 \sin(3x), \quad x \in [0, 1] \quad (17)$$

with its exact solution given by

$$\sin(3x)$$

The analysis and results of the performances in terms of iterations number (Iter), execution time (Second) and maximum absolute error (MAE) for Problem 1 and Problem 2 are presented and discussed in the next section.
5. Result and discussion

Based on the numerical experiment, the performances results of Problem 1 and Problem 2 are successfully tabulated in Table 1 and 2, respectively. Both tables show that as the matrix sizes increasing, the iterations number generated by the three iterative methods are also increasing. This is due to the accumulated round-off error that occurred at every iteration. However, it can be observed that the CG iterative method performs better than SOR and GS iterative methods as the matrix sizes are being incremented, and this is evidently presented through the difference of iterations number, execution time and maximum absolute error at different matrix sizes (128, 256, 512, 1024 and 2048).

Other than lesser iterations number, the CG iterative method also requires shorter execution time in order to iterate and converge to the exact solution, when solving the two-point boundary value problems. In fact, by going down the tables, the performances of the CG iterative method can be seen improving over SOR and GS methods for different matrix sizes especially the accuracy which given by maximum absolute error. This indicates that CG iterative method can cope with the accumulated round-off error better than SOR and GS method when solving the two-point boundary value problems together with the quartic non-polynomial spline scheme. Hence, it can be stated that CG iterative method has better performances compared to SOR and GS iterative methods when solving the problems.

Table 1

<table>
<thead>
<tr>
<th>Matrix Sizes Method</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Iterations Number</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GS</td>
<td>18173</td>
<td>66139</td>
<td>238353</td>
<td>848604</td>
<td>2975185</td>
</tr>
<tr>
<td>SOR</td>
<td>382</td>
<td>807</td>
<td>1438</td>
<td>2821</td>
<td>5367</td>
</tr>
<tr>
<td>CG</td>
<td>65</td>
<td>129</td>
<td>257</td>
<td>513</td>
<td>1025</td>
</tr>
<tr>
<td><strong>Execution Time</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GS</td>
<td>14.40</td>
<td>49.09</td>
<td>168.88</td>
<td>662.96</td>
<td>83318.09</td>
</tr>
<tr>
<td>SOR</td>
<td>1.52</td>
<td>1.45</td>
<td>3.95</td>
<td>6.32</td>
<td>8.75</td>
</tr>
<tr>
<td>CG</td>
<td>0.16</td>
<td>0.54</td>
<td>1.44</td>
<td>2.31</td>
<td>2.62</td>
</tr>
<tr>
<td><strong>Maximum Absolute Error</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GS</td>
<td>1.1788e-07</td>
<td>4.7242e-07</td>
<td>1.8899e-06</td>
<td>7.5601e-06</td>
<td>3.0241e-05</td>
</tr>
<tr>
<td>SOR</td>
<td>4.0171e-10</td>
<td>5.8395e-09</td>
<td>4.3379e-09</td>
<td>9.6408e-09</td>
<td>1.8447e-08</td>
</tr>
<tr>
<td>CG</td>
<td>1.1833e-10</td>
<td>7.2046e-12</td>
<td>1.8359e-13</td>
<td>2.1296e-12</td>
<td>8.1399e-12</td>
</tr>
</tbody>
</table>
Quartic non-polynomial spline solution for solving two-point boundary value problems …

Table 2

Comparison of GS, SOR and CG iterative methods in terms of iterations number (Iter), execution time (Seconds) and maximum absolute errors (MAE) for Problem 2

<table>
<thead>
<tr>
<th>Method</th>
<th>Matrix Sizes</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS</td>
<td>25950</td>
<td>94591</td>
<td>341534</td>
<td>1218827</td>
<td>4286118</td>
<td></td>
</tr>
<tr>
<td>SOR</td>
<td>735</td>
<td>976</td>
<td>2703</td>
<td>5174</td>
<td>9181</td>
<td></td>
</tr>
<tr>
<td>CG</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Matrix Sizes</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS</td>
<td>37.06</td>
<td>88.40</td>
<td>325.63</td>
<td>1220.80</td>
<td>6231.12</td>
<td></td>
</tr>
<tr>
<td>SOR</td>
<td>1.68</td>
<td>2.59</td>
<td>4.12</td>
<td>7.35</td>
<td>13.76</td>
<td></td>
</tr>
<tr>
<td>CG</td>
<td>0.57</td>
<td>1.14</td>
<td>1.60</td>
<td>1.99</td>
<td>4.21</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Matrix Sizes</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS</td>
<td>1.6485e-07</td>
<td>6.6388e-07</td>
<td>2.6560e-06</td>
<td>1.0624e-05</td>
<td>4.2498e-05</td>
<td></td>
</tr>
<tr>
<td>SOR</td>
<td>2.6789e-09</td>
<td>6.9686e-10</td>
<td>1.4228e-08</td>
<td>2.8422e-08</td>
<td>5.1772e-08</td>
<td></td>
</tr>
<tr>
<td>CG</td>
<td>1.0363e-09</td>
<td>6.4801e-11</td>
<td>4.1241e-12</td>
<td>1.1072e-13</td>
<td>5.3167e-12</td>
<td></td>
</tr>
</tbody>
</table>

6. Conclusion

In conclusion, an approximate equation to solve two-point boundary value problems was successfully developed based on a quartic non-polynomial spline so that a system of linear equations can be constructed. Then, this linear system was solved by using three iterative methods which are the GS, SOR and CG iterative methods. Based on the results of performances experiment, the CG method was found to be superior compared to the GS and SOR method, and it is evidently proven through the comparison shown by the CG method in terms of iterations number, execution time and maximum absolute error at different respective matrix sizes. Therefore, it can be summarized that, the approximate solution obtained from the discretization of two-point boundary value problems by using the quartic non-polynomial scheme to form a linear system is best solved with the CG iterative method.

References


