New Cubic B-Spline Approximation for Solving Linear Two-Point Boundary-Value Problems

Busrya Latif 1,2, Samsul Ariffin Abdul Karim 3 and Ishak Hashim 2,*

1 Cawangan Negeri Sembilan, Kampus Seremban, Universiti Teknologi MARA, Seremban 70300, Negeri Sembilan, Malaysia; busyra4042@uitm.edu.my
2 Department of Mathematical Sciences, Faculty of Science & Technology, Universiti Kebangsaan Malaysia, UKM, Bangi 43600, Selangor, Malaysia
3 Fundamental and Applied Sciences Department and Centre of Systems Engineering (CSE), Institute of Autonomous System, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, Seri Iskandar 32610, Perak DR, Malaysia; samsul_ariffin@utp.edu.my
* Correspondence: ishak_h@ukm.edu.my

Abstract: In this study, we introduce a new cubic B-spline (CBS) approximation method to solve linear two-point boundary value problems (BVPs). This method is based on cubic B-spline basis functions with a new approximation for the second-order derivative. The theoretical new approximation for a second-order derivative and the error analysis have been successfully derived. We found that the second-order new approximation was $O(h^3)$ accurate. By using this new second-order approximation, the proposed method was $O(h^5)$ accurate. Four numerical problems consisting of linear ordinary differential equations and trigonometric equations with different step sizes were performed to validate the accuracy of the proposed methods. The numerical results were compared with the least squares method, finite difference method, finite element method, finite volume method, B-spline interpolation method, extended cubic B-spline interpolation method and the exact solutions. By finding the maximum errors, the results consistently showed that the proposed method gave the best approximations among the existing methods. We also found that our proposed method involved simple implementation and straightforward computations. Hence, based on the results and the efficiency of our method, we can say that our method is reliable and a promising method for solving linear two-point BVPs.

Keywords: cubic B-spline; two-point boundary value problems; ordinary differential equation; numerical solution; error analysis

1. Introduction

Boundary value problems (BVPs) have been extensively investigated in the fields of physics, chemistry and engineering. Numerous methods have been implemented throughout the years to approximate the solutions of linear and nonlinear two-point BVPs, such as the variational approach, finite difference (FDM), finite element (FEM), finite volume (FVM) and shooting (LSM) [1–3]. Bickley first employed cubic splines to solve a simple two-point BVP [4], followed by Albasiny and Hoskins [5]. Fyfe continued Bickley’s work and established the fourth-order accuracy of the cubic spline interpolation scheme [6]. Since then, spline functions have been frequently used for solving BVPs [7–10].

In 2006, Caglar et al. [11] replaced the cubic spline with a cubic B-spline basis function to solve two-point BVPs and named it the cubic B-spline interpolation method (BSI). Several cubic B-spline based numerical approaches have been widely applied to solve linear and nonlinear BVPs [12–15]. The extended cubic B-spline and cubic trigonometric B-spline
were studied by Hamid et al. [16,17] as solutions to linear two-point BVPs. It was found that the cubic trigonometric B-spline provided better results compared with the cubic BSI method if the problems had trigonometric functions. Heilat and Ismail [18] developed a hybrid cubic B-spline method to deal with nonlinear two-point boundary value problems.

This study focuses on numerically solving linear two-point BVPs using a new cubic B-spline (CBS) approximation. These functions are flexible enough to approximate the solution at any point of the domain with high accuracy while maintaining a high degree of smoothness at the knots. In recent years, the new CBS approximation method has been applied for solving BVPs. Iqbal et al. [19] investigated the numerical solutions of second-order singular BVPs using a new approximation for the second-order derivative. A new approximation for the second-order derivative of the extended cubic B-spline basis was developed by Wasim et al. [20] for solving second-order singular BVPs. Iqbal et al. [21] employed a new CBS approximation for solving several third-order Emden–Flower type equations. Nazir et al. [22] used the new CBS approximation for the numerical solutions of coupled viscous Burgers equations. Abbas et al. [23] proposed a new CBS approximation to approximate the solutions to nonlinear third-order Korteweg–de Vries equations. A new quintic B-spline approximation was studied by Nazir et al. [24] as a method for solving Boussinesq equations. Motivated by these works, this study was conducted to solve linear second-order ordinary differential equations (ODEs) based on a new CBS approximation to figure out whether this method would perform equally well. This paper is organized as follows. In Section 2, we present the typical definition of cubic B-spline basis functions. The formulation of a new approximation to second-order derivatives is developed in Section 3. The numerical method is presented in Section 4. In Section 5, the error analysis is investigated. The results and discussion are demonstrated in Sections 6 and 7, respectively. Section 8 summarizes this paper with a brief conclusion.

2. Cubic B-Spline Functions

Let there be a finite interval \([a, b]\), where \(a = x_0 < x_1 < \ldots < x_N = b\) is divided into equidistant partitions with a mesh point \(x_i = x_0 + ih\), \(i = 0 : 1 : N\) using a step size \(h = \frac{b-a}{N}\), \(N \in \mathbb{Z}^+\). The typical CBS basis function is defined as [23]

\[
B_i(x) = \frac{1}{6h^3} \begin{cases} 
(x-x_{i-2})^3, & x \in [x_{i-2}, x_{i-1}] \\
h^3 + 3h^2(x-x_{i-1}) + 3h(x-x_{i-1})^2 - 3(x-x_{i-1})^3, & x \in [x_{i-1}, x_i] \\
h^3 + 3h^2(x_{i+1}-x) + 3h(x_{i+1}-x)^2 - 3(x_{i+1}-x)^3, & x \in [x_i, x_{i+1}] \\
(x_{i+2}-x)^3, & x \in [x_{i+1}, x_{i+2}] \\
0, & \text{otherwise,}
\end{cases}
\]  

(1)

where \(i = -1 : 1 : N + 1\). For a sufficiently smooth function \(y(x)\), there exists a unique third-degree spline \(Y(x)\) that satisfies the prescribed interpolating conditions:

\[
Y(x_i) = y(x_i), \quad i = 0, 1, 2, \ldots, N \\
Y'(a) = y'(a), \quad Y'(b) = y'(b) \\
Y''(a) = y''(a), \quad Y''(b) = y''(b)
\]

in which

\[
Y(x) = \sum_{i=1}^{N} \sigma_i B_i(x)
\]  

(2)
where \( \sigma_i \) is the set of unknown real coefficients to be computed. The values of \( B_i(x) \) and the first and second derivatives \( B'_i(x) \) and \( B''_i(x) \) at mesh point \( x_i \) are tabulated in Table 1. For simplicity, we denote the CBS approximations \( Y_i(x_i) \), \( Y'_i(x_i) \) and \( Y''_i(x_i) \) as \( Y_j \), \( s_j \) and \( S_j \), respectively. Hence, Equation (2) is a linear combination of \( B_i(x) \) for \( i = -1 : 1 : N + 1 \).

### Table 1. Coefficients of \( B_i(x) \), \( B'_i(x) \) and \( B''_i(x) \) at the nodes.

<table>
<thead>
<tr>
<th></th>
<th>( x_{i-2} )</th>
<th>( x_{i-1} )</th>
<th>( x_i )</th>
<th>( x_{i+1} )</th>
<th>( x_{i+2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B_i(x) )</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( B'_i(x) )</td>
<td>0</td>
<td>(-1/2h)</td>
<td>0</td>
<td>(1/2h )</td>
<td>0</td>
</tr>
<tr>
<td>( B''_i(x) )</td>
<td>0</td>
<td>(1/4h^2 )</td>
<td>(-2/4h^2 )</td>
<td>(1/4h^2 )</td>
<td>0</td>
</tr>
</tbody>
</table>

From Equations (1) and (2) and Table 1, the following relations can be obtained:

\[
Y_j = \sum_{i=-1}^{i=1} \sigma_i B_i(x) = \frac{1}{6} (\sigma_{j-1} + 4\sigma_j + \sigma_{j+1}) \quad (3)
\]

\[
s_j = \frac{1}{2h^4} (\sigma_{j-1} + \sigma_{j+1}) \quad (4)
\]

\[
S_j = \frac{1}{h^4} (\sigma_{j-1} - 2\sigma_j + \sigma_{j+1}) \quad (5)
\]

Using the relations from Equations (3–5), the following expressions can be established [6]:

\[
s_j = y'(x_j) - \frac{1}{180} h^4 y^{(5)}(x_j) + \ldots \quad (6)
\]

\[
S_j = y''(x_j) - \frac{1}{12} h^4 y^{(5)}(x_j) + \frac{1}{360} h^4 y^{(7)}(x_j) + \ldots \quad (7)
\]

Hence, the truncation error in \( S_j \) is \( O(h^2) \). Consequently, a better approximation for \( y''(x) \) can be constructed.

### 3. The New Approximation for \( y''(x) \)

To construct a new approximation for \( y''(x) \), we used Equation (7) to demonstrate the following expression for \( S_{j-1} \) at the knot \( x_j \) for \( j = 1, 2, 3, \ldots, N - 1 \) [25]:

\[
S_{j-1} = y''(x_{j-1}) - \frac{1}{12} h^4 y^{(4)}(x_{j-1}) + \frac{1}{360} h^4 y^{(6)}(x_{j-1}) + \ldots
\]

\[
= y''(x_j) - \frac{5}{12} h^4 y^{(4)}(x_j) - \frac{1}{12} h^4 y^{(5)}(x_j) + \ldots
\]

Similarly, for \( S_{j+1} \) at \( x_i \), we have
Now consider \( \tilde{S}_j \) as the new approximation for \( y''(x) \) such that

\[
\tilde{S}_j = Z_1 S_j + Z_2 S_{j-1} + Z_3 S_{j+1}
\]  

(8)

where \( Z_1, Z_2 \) and \( Z_3 \) are the three parameters to be chosen so that the error order of \( \tilde{S}_j \) is as high as possible. From the linear combination, we have

\[
Z_1 + Z_2 + Z_3 = 1
\]  

(9)

\[
-Z_2 + Z_3 = 0
\]  

(10)

\[
-Z_1 + 5Z_2 + 5Z_3 = 0
\]  

(11)

By solving Equations (9–11) simultaneously, we get \( Z_1 = \frac{5}{6}, \ Z_2 = \frac{1}{12} \) and \( Z_3 = \frac{1}{12} \). Using Equation (5) and the parameters’ values, Equation (8) can be expressed as

\[
\tilde{S}_j = \frac{1}{12h^2} \left( \sigma_{j-2} + 8\sigma_{j-1} - 18\sigma_j + \sigma_{j+2} \right)
\]  

(12)

Next, at the knot \( x_0 \), the linear combination of four neighboring knots can be written as

\[
\tilde{S}_0 = Z_1 S_0 + Z_2 S_1 + Z_3 S_2 + Z_4 S_3
\]  

(13)

where

\[
S_1 = y''(x_0) + h y'''(x_0) + \frac{5}{12} h^2 y^{(4)}(x_0) + \frac{1}{12} h^3 y^{(5)}(x_0) + \ldots
\]

\[
S_2 = y''(x_0) + 2 h y'''(x_0) + \frac{23}{12} h^2 y^{(4)}(x_0) + \frac{7}{6} h^3 y^{(5)}(x_0) + \ldots
\]

\[
S_3 = y''(x_0) + 3 h y'''(x_0) + \frac{53}{12} h^2 y^{(4)}(x_0) + \frac{17}{4} h^3 y^{(5)}(x_0) + \ldots
\]

The linear combination in Equation (13) produces the following four expressions:

\[
Z_1 + Z_2 + Z_3 + Z_4 = 1
\]

\[
Z_2 + 2Z_3 + 3Z_4 = 0
\]

\[-Z_1 + 5Z_2 + 23Z_3 + 53Z_4 = 0\]

\[
Z_2 + 14Z_3 + 51Z_4 = 0
\]

Hence, we get \( Z_1 = \frac{7}{6}, \ Z_2 = -\frac{5}{12}, \ Z_3 = \frac{1}{3} \) and \( Z_4 = -\frac{1}{12} \). By substituting the parameter values, Equation (13) takes the following form:

\[
\tilde{S}_0 = \frac{1}{12h^2} \left( 14\sigma_{-2} - 33\sigma_0 + 28\sigma_1 - 14\sigma_2 + 6\sigma_3 - \sigma_4 \right)
\]

Similarly, the approximation to \( y''(x) \) can be written as

\[
\tilde{S}_n = \frac{1}{12h^2} \left( -\sigma_{n-2} + 4\sigma_{n-3} - 14\sigma_{n-2} + 28\sigma_{n-1} - 33\sigma_n + 14\sigma_{n+1} \right)
\]
Therefore, the new approximation for \( y''(x) \) can be defined as
\[
S_j = \frac{1}{12h^2} \left[ \begin{array}{c}
14\sigma_{j-2} - 33\sigma_{j-1} + 28\sigma_{j+1} - 14\sigma_{j+2} + 6\sigma_{j+3} - \sigma_{j+4}, \\
\sigma_{j+2} - 18\sigma_{j+1} + 8\sigma_j + \sigma_{j-1}, \\
-\sigma_{j-4} + 6\sigma_{j-3} - 14\sigma_{j-2} + 28\sigma_{j-1} - 33\sigma_j + 14\sigma_{j+1}, \\
-\sigma_{j-4} + 6\sigma_{j-3} - 14\sigma_{j-2} + 28\sigma_{j-1} - 33\sigma_j + 14\sigma_{j+1}, \\
\sigma_{j+2} - 18\sigma_{j+1} + 8\sigma_j + \sigma_{j-1}, \\
14\sigma_{j-2} - 33\sigma_{j-1} + 28\sigma_{j+1} - 14\sigma_{j+2} + 6\sigma_{j+3} - \sigma_{j+4},
\end{array} \right]
\]
for \( j = 0 \) \( \ldots \) \( j = N \).

4. Description of the Numerical Method

In this section, the numerical scheme for solving linear second-order ordinary differential equations using the new CBS approximation is discussed. Let us consider the general form of a linear second order of a two-point boundary value problem [26]:
\[
p_1(x)y''(x) + p_2(x)y'(x) + p_3(x)y(x) = f(x)
\]
with boundary conditions
\[
y(a) = \eta_1, \quad y(b) = \eta_2
\]
where \( a \leq x \leq b \), \( \eta_1 \) and \( \eta_2 \) are constants. In addition, \( f(x) \) and \( p_i, i=1,2,3 \) are continuous and sufficiently smooth functions on the interval \( I \) [26]. Assume the CBS solution to Equation (15) is given as
\[
Y(x) = \sum_{i=1}^{N+1} \sigma_i B_i(x)
\]
where \( \sigma_i \) is the constants to be calculated.

Discretizing Equation (15) at the knot \( x_j \) yields the following form:
\[
p_1(x_j)S_{k+1}(x_j) + p_2(x_j)S_{k+2}(x_j) + p_3(x_j)Y_{k+1}(x_j) = f_k(x_j)
\]
where \( k = 0,1,2,\ldots \). By substituting Equations (3), (4) and (14) into Equation (18) for \( j = 0,1,2,\ldots,N-1,N \), we obtain the following \( N+1 \) linear equations involving \( N+3 \) unknowns:
\[
p_1(x_0) \left( \begin{array}{c}
14\sigma_0 - 33\sigma_1 + 28\sigma_2 - 14\sigma_3 + 6\sigma_4 - \sigma_5 \\
-\sigma_4 + \sigma_5 \\
\sigma_3 + 4\sigma_0 + \sigma_1 \\
-\sigma_1 + \sigma_2 \\
-\sigma_2 + 8\sigma_3 - 18\sigma_4 + 8\sigma_5 + \sigma_0 \\
-\sigma_5 + 6\sigma_6 - 14\sigma_7 + 28\sigma_8 - 33\sigma_9 + 14\sigma_{10} \\
-\sigma_9 + 6\sigma_{10} - 14\sigma_{11} + 28\sigma_{12} - 33\sigma_{13} + 14\sigma_{14} \\
\sigma_2 + 4\sigma_3 + \sigma_{11} \\
-\sigma_{11} + \sigma_{12} \\
\sigma_{12} + 4\sigma_{13} + \sigma_{10} \\
-\sigma_8 + 8\sigma_9 - 18\sigma_{10} + 8\sigma_{11} + \sigma_7 \\
14\sigma_0 - 33\sigma_1 + 28\sigma_2 - 14\sigma_3 + 6\sigma_4 - \sigma_5
\end{array} \right)
\]
\[
= f_k(x_0),
\]
\[
= f_k(x_j), \quad j = 1:1:N-1
\]
\[
= f_k(x_N)
\]
In addition, we need two additional equations, which can be obtained from the boundary conditions of Equation (16) as shown below:
\[
\sigma_1 + 4\sigma_2 + \sigma_5 = 6\eta_1
\]
\[ \sigma_{N-1} + 4\sigma_N + \sigma_{N+1} = 6\eta_2 \]

Hence, this will give the following \((N+3) \times (N+3)\) dimensional matrix form:

\[ Ac = b \]

given by

\[
\begin{pmatrix}
1 & 4 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & \ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
q_1 & r_1 & s_1 & t_1 & u_1 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & q_2 & r_2 & s_2 & t_2 & u_2 & \ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & q_{N-2} & r_{N-2} & s_{N-2} & t_{N-2} & u_{N-2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & q_{N-1} & r_{N-1} & s_{N-1} & t_{N-1} & u_{N-1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 1 & 4 & 1 \\
\end{pmatrix}
\]

where

\[
a_1 = 14p_1(x_0) - 6hp_2(x_0) + 2h^2 p_3(x_0),
\]
\[
a_2 = -33p_1(x_0) + 8h^2 p_3(x_0),
\]
\[
a_3 = 28p_1(x_0) + 6hp_2(x_0) + 2h^2 p_3(x_0),
\]
\[
a_4 = -14p_1(x_0),
\]
\[
a_5 = 6p_1(x_0),
\]
\[
a_6 = -6p_1(x_0),
\]
\[
z_1 = -p_1(x_N),
\]
\[
z_2 = 6p_1(x_N),
\]
\[
z_3 = -14p_1(x_N),
\]
\[
z_4 = 28p_1(x_N) - 6hp_2(x_N) + 2h^2 p_3(x_N),
\]
\[
z_5 = -33p_1(x_N) + 8h^2 p_3(x_N),
\]
\[
z_6 = 14p_1(x_N) + 6hp_2(x_N) + 2h^2 p_3(x_N),
\]

and for \(j = 1, 2, \ldots, N-1,\)

\[
q_j = p_1(x_j),
\]
\[
r_j = 8p_1(x_j) - 6hp_2(x_j) + 2h^2 p_3(x_j),
\]
\[
s_j = -18p_1(x_j) + 8h^2 p_3(x_j),
\]
\[
t_j = 8p_1(x_j) + 6hp_2(x_j) + 2h^2 p_3(x_j),
\]

and \(u_j = p_1(x_j).\)

The unknown column vector \(c\) is determined by a well-known Thomas algorithm, and its components are plugged into Equation (17) to obtain the CBS approximation. The solutions to the tridiagonal linear system of equations given in Equation (17) are guaranteed to be unique. This is because the matrix is strictly dominant and nonsingular.
5. Error Analysis

The estimation of a truncation error for the proposed method is presented. By using the new CBS approximations in Equations (3), (4) and (12), the following relations can be obtained:

\[
Y''(x) = \frac{1}{2} \left[ 7Y(x_{j+1}) - 8Y(x_j) + Y(x_{j-1}) \right] + h[Y'(x_{j+1}) + 2Y'(x_j)]
\]

(20)

Furthermore, we have [6]

\[
Y'''(x) = 12 \left[ Y'(x_j) - Y(x_{j+1}) \right] + 6h \left[ Y''(x_j) + Y''(x_{j+1}) \right]
\]

(21)

\[
Y'''(x) = 12 \left[ Y'(x_{j+1}) - Y(x_j) \right] + 6h \left[ Y''(x_{j+1}) + Y''(x_j) \right]
\]

(22)

where \( Y'''(x_{j-1}) \) and \( Y'''(x_{j+1}) \) are the approximate values of \( y'''(x) \) in \([x_{j-1}, x_{j+1}]\) and \([x_{j+1}, x_{j+2}]\), respectively.

Let \( E^\mu \) be the shift operator; in other words, \( E^\mu \left[ Y'(x) \right] = Y'(x_{j+\mu}) \). Then, Equation (19) can be expressed as [6]

\[
hY(x) = 3 \left[ E^\mu E^{-1} y(x) \right] = 3 \left[ E^{-1} + 4 + E \right] y(x)
\]

Thus, we have

\[
hY(x) = 3 \left[ E^{-1} + 4 + E \right] y(x) = 3 \left[ E^{-2} + 4 + E \right] y(x)
\]

(23)

Using \( E = e^{\lambda D}, D = d/dx \), we have

\[
E + E^\lambda = e^{\lambda D} + e^{\lambda D} = \left[ 1 + hD + \frac{h^2 D^2}{2!} + \frac{h^3 D^3}{3!} + \ldots \right] + \left[ 1 - hD + \frac{h^2 D^2}{2!} - \frac{h^3 D^3}{3!} + \ldots \right]
\]

\[
= 2 \left[ 1 + \frac{h^2 D^2}{2!} + \frac{h^3 D^3}{3!} + \ldots \right]
\]

\[
E - E^\lambda = e^{\lambda D} - e^{\lambda D} = \left[ 1 + hD + \frac{h^2 D^2}{2!} + \frac{h^3 D^3}{3!} + \ldots \right] - \left[ 1 - hD + \frac{h^2 D^2}{2!} - \frac{h^3 D^3}{3!} + \ldots \right]
\]

\[
= 2 \left[ hD + \frac{h^2 D^2}{3!} + \frac{h^3 D^3}{5!} + \ldots \right]
\]

Hence, Equation (23) can be written as

\[
Y(x) = 3 \left[ 2 \left( 1 + \frac{h^2 D^2}{2!} + \frac{h^3 D^3}{3!} + \ldots \right) \right] y(x)
\]

\[
= \left[ D + \frac{h^2 D^3}{3!} + \frac{h^3 D^4}{5!} + \ldots \right] y(x)
\]

\[
= \left[ D + \frac{h^2 D^3}{3!} + \frac{h^3 D^4}{5!} + \ldots \right] y(x)
\]

\[
= \left[ D + \frac{h^2 D^3}{3!} + \frac{h^3 D^4}{5!} + \ldots \right] y(x)
\]

\[
= \left[ D + \frac{h^2 D^3}{3!} + \frac{h^3 D^4}{5!} + \ldots \right] y(x)
\]

\[
= \left[ D + \frac{h^2 D^3}{3!} + \frac{h^3 D^4}{5!} + \ldots \right] y(x)
\]

\[
= \left[ D + \frac{h^2 D^3}{3!} + \frac{h^3 D^4}{5!} + \ldots \right] y(x)
\]

\[
= \left[ D + \frac{h^2 D^3}{3!} + \frac{h^3 D^4}{5!} + \ldots \right] y(x)
\]
Therefore, we have
\[ Y'(x_j) = y'(x_j) - \frac{1}{180} h^4 y^{(9)}(x_j) + \ldots \] (24)

Similarly, Equations (20) and (21) give the following:
\[ Y''(x_j) = y''(x_j) + \frac{1}{60} h^3 y^{(5)}(x_j) - \frac{1}{360} h^4 y^{(6)}(x_j) + \frac{1}{1260} h^5 y^{(7)}(x_j) + \ldots \] (25)

\[ Y'''(x_j) = \frac{1}{2} [Y'''(x_j) + Y'''(x_{j-1})] = y'''(x_j) + \frac{1}{12} h^2 y^{(5)}(x_j) + \ldots \] (26)

Thus, the following theorem have been proven.

**Theorem 1.** Let \( y(x) \) be a sufficiently smooth function for \( a \leq x \leq b \) and further assume that \( Y(x) \) is the cubic B-spline approximation defined by Equation (17) for \( y(x) \). Then, at the mesh points \( x_i \) for \( i = 0, 1, 2, \ldots, N \), we have

\[ Y'(x_j) = y'(x_j) - \frac{1}{180} h^4 y^{(9)}(x_j) + \ldots, \]
\[ Y''(x_j) = y''(x_j) + \frac{1}{60} h^3 y^{(5)}(x_j) - \frac{1}{360} h^4 y^{(6)}(x_j) + \frac{1}{1260} h^5 y^{(7)}(x_j) + \ldots, \]
\[ Y'''(x_j) = \frac{1}{2} [Y'''(x_j) + Y'''(x_{j-1})] = y'''(x_j) + \frac{1}{12} h^2 y^{(5)}(x_j) + \ldots. \] □

Let us consider \( e(x) = Y(x) - y(x) \) as the error term. By substituting Equations (24–26) into the Taylor series expansion of \( e(x_j + \theta h) \), we have

\[ e(x_j + \theta h) = \theta(\theta + 1)(5\theta - 2) \frac{y^{(9)}}{360} \frac{h^5 y^{(5)}(x_j)}{360} - \frac{\theta^2}{720} h^6 y^{(6)}(x_j) + \ldots \] (27)

where \( \theta \in [0, 1] \). Hence, it is clear that the proposed new CBS approximation is \( O(h^5) \) accurate.

**6. Results**

In this section, the accuracy of the new CBS approximation method was measured and compared with exact solutions and existing methods. The accuracy was measured using the maximum error, defined as

\[ L_\infty = \max_j |Y(x_j) - y(x_j)|. \]

The numerical computations were performed using Matlab R2018a running on an Intel(R) CORE(TM) i7 CPU 1.30GHz processor, 8.00GB RAM.
6.1. Problem 1

Consider the following linear second-order differential equation [11]:

\[ y''(x) - y'(x) = -e^{-x} - 1, \quad x \in [0, 1] \]

which is subject to the following boundary conditions:

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

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

Table 2 lists the absolute error when \( h = 1/10 \). It is clear that the proposed method agreed with the exact solutions.

<table>
<thead>
<tr>
<th>x</th>
<th>Proposed Method</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.00000000000</td>
<td>0.00000000000</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0593429531</td>
<td>0.0593430340</td>
<td>8.095×10^{-8}</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1101340372</td>
<td>0.1101342072</td>
<td>1.699×10^{-7}</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1510241642</td>
<td>0.1510244089</td>
<td>2.447×10^{-7}</td>
</tr>
<tr>
<td>0.4</td>
<td>0.1804750426</td>
<td>0.1804753456</td>
<td>3.030×10^{-7}</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1967343303</td>
<td>0.1967346701</td>
<td>3.398×10^{-7}</td>
</tr>
<tr>
<td>0.6</td>
<td>0.1978076229</td>
<td>0.1978079724</td>
<td>3.495×10^{-7}</td>
</tr>
<tr>
<td>0.7</td>
<td>0.1814269201</td>
<td>0.1814272455</td>
<td>3.254×10^{-7}</td>
</tr>
<tr>
<td>0.8</td>
<td>0.1450151375</td>
<td>0.1450153975</td>
<td>2.600×10^{-7}</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0856461846</td>
<td>0.0856463238</td>
<td>1.392×10^{-7}</td>
</tr>
<tr>
<td>1.0</td>
<td>1.00000000000</td>
<td>1.00000000000</td>
<td>0</td>
</tr>
</tbody>
</table>

In Table 3, the comparison of maximum errors between the proposed method, FDM, FEM, FVM, BSI, extended cubic B-Spline interpolation using Newton’s method (ECBI(N)) and extended cubic B-Spline interpolation using a built-in function (ECBI(B)) are given with different values of \( N \) to show the impact of \( N \). Evidently, our results were better than the others.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \lambda = 2.9097 \times 10^{-3} )</td>
<td>( \lambda = 2.9375 \times 10^{-3} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>8.24×10^{-5}</td>
<td>6.35×10^{-5}</td>
<td>3.18×10^{-5}</td>
<td>2.9×10^{-4}</td>
<td>7.92×10^{-6}</td>
<td>5.74×10^{-6}</td>
<td>3.50×10^{-7}</td>
</tr>
<tr>
<td>100</td>
<td>8.31×10^{-7}</td>
<td>6.36×10^{-7}</td>
<td>3.18×10^{-7}</td>
<td>2.89×10^{-6}</td>
<td>-</td>
<td>-</td>
<td>3.74×10^{-11}</td>
</tr>
<tr>
<td>1000</td>
<td>8.31×10^{-9}</td>
<td>6.39×10^{-9}</td>
<td>3.18×10^{-9}</td>
<td>2.9×10^{-8}</td>
<td>-</td>
<td>-</td>
<td>6.76×10^{-14}</td>
</tr>
</tbody>
</table>

6.2. Problem 2

Consider the following problem [3]:
having the boundary conditions

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

The true solution is \( y(x) = xe^{x-1} \).

Table 4 displays the absolute error when \( h = 1/10 \). We can say that the proposed method fit well with the exact solutions.

<table>
<thead>
<tr>
<th>( x )</th>
<th>Proposed Method</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.00000000000</td>
<td>0.00000000000</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0406569888</td>
<td>0.0406569660</td>
<td>2.281\times10^{-8}</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0898658490</td>
<td>0.0898658490</td>
<td>5.619\times10^{-8}</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1489756718</td>
<td>0.1489756718</td>
<td>8.0638\times10^{-8}</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2195247518</td>
<td>0.2195247518</td>
<td>9.735\times10^{-8}</td>
</tr>
<tr>
<td>0.5</td>
<td>0.3032654351</td>
<td>0.3032654351</td>
<td>1.052\times10^{-7}</td>
</tr>
<tr>
<td>0.6</td>
<td>0.4021921308</td>
<td>0.4021921308</td>
<td>1.032\times10^{-7}</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5185728443</td>
<td>0.5185728443</td>
<td>8.983\times10^{-8}</td>
</tr>
<tr>
<td>0.8</td>
<td>0.6549846664</td>
<td>0.6549846664</td>
<td>6.398\times10^{-8}</td>
</tr>
<tr>
<td>0.9</td>
<td>0.8143536951</td>
<td>0.8143536951</td>
<td>1.887\times10^{-8}</td>
</tr>
<tr>
<td>1.0</td>
<td>1.00000000000</td>
<td>1.00000000000</td>
<td>0</td>
</tr>
</tbody>
</table>

Problem 2 was studied using LSM, FDM and BSI. As can be seen in Table 5, a comparison of the maximum errors between the proposed method and the mentioned methods is given with different values of \( N \). Obviously, our method generated more accurate results compared with the others.

Table 5. Comparison of maximum errors for Problem 2.

<table>
<thead>
<tr>
<th>( N )</th>
<th>Max-Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3.66\times10^{-7}</td>
</tr>
<tr>
<td>100</td>
<td>4.01\times10^{-11}</td>
</tr>
</tbody>
</table>

6.3. Problem 3

Consider the following equation [27]:

\[ y''(x) + y'(x) - 6y(x) = x, \quad x \in [0, 1] \]

subject to the initial conditions

\[ y(0) = 0, \quad y(1) = 1. \]
The analytical solution is
\[
y(x) = \frac{(43-e^2)e^{-3x}-(43-e^{-3})e^{2x}}{36(e^3-e^{-2})} - \frac{1}{6}x - \frac{1}{36}.
\]

The absolute errors when \( h = 1/20 \) are presented in Table 6. We can say that the proposed method was in good agreement with the exact solutions.

**Table 6. Absolute errors for Problem 3 at \( h = 1/20 \).**

<table>
<thead>
<tr>
<th>( x )</th>
<th>Proposed Method</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000000000</td>
<td>0.0000000000</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0542569644</td>
<td>0.0542570003</td>
<td>3.587 \times 10^{-8}</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1074285058</td>
<td>0.1074285617</td>
<td>5.592 \times 10^{-8}</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1636254812</td>
<td>0.1636255435</td>
<td>6.234 \times 10^{-8}</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2267411540</td>
<td>0.2267412146</td>
<td>6.063 \times 10^{-8}</td>
</tr>
<tr>
<td>0.5</td>
<td>0.3006953149</td>
<td>0.3006953693</td>
<td>5.433 \times 10^{-8}</td>
</tr>
<tr>
<td>0.6</td>
<td>0.3896566891</td>
<td>0.3896567348</td>
<td>4.567 \times 10^{-8}</td>
</tr>
<tr>
<td>0.7</td>
<td>0.4982584629</td>
<td>0.4982584988</td>
<td>3.594 \times 10^{-8}</td>
</tr>
<tr>
<td>0.8</td>
<td>0.6318199531</td>
<td>0.6318199790</td>
<td>2.587 \times 10^{-8}</td>
</tr>
<tr>
<td>0.9</td>
<td>0.7965865545</td>
<td>0.7965865702</td>
<td>1.573 \times 10^{-8}</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>0</td>
</tr>
</tbody>
</table>

The maximum errors between the proposed method and BSI method for \( N = 20, 50 \) and 100 are presented in Table 7. Clearly, our method produced better approximations compared with BSI.

**Table 7. Comparison of maximum errors for Problem 3.**

<table>
<thead>
<tr>
<th>( N )</th>
<th>( \text{Max-Norm} )</th>
<th>BSI [27]</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>9.53 \times 10^{-5}</td>
<td>6.23 \times 10^{-8}</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>1.52 \times 10^{-5}</td>
<td>1.63 \times 10^{-9}</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>3.81 \times 10^{-6}</td>
<td>1.02 \times 10^{-10}</td>
<td></td>
</tr>
</tbody>
</table>

6.4. Problem 4

Consider the following equation [27]:
\[
y''(x) + 2y'(x) + 5y(x) = 6 \cos(2x) - 7 \sin(2x), \quad x \in \left[0, \frac{\pi}{4}\right]
\]
with boundary conditions
\[
y(0) = 4, \quad y\left(\frac{\pi}{4}\right) = 1
\]

The analytical solution is
\[
y(x) = 2(1 + e^{-x}) \cos(2x) + \sin(2x).
\]

In Table 8, the absolute errors for \( h = 1/20 \) are provided. Clearly, the proposed method worked well with the exact solutions.
Table 8. Absolute errors for Problem 4 at $h = 1/20$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>Proposed Method</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.0000000000</td>
<td>4.0000000000</td>
<td>0</td>
</tr>
<tr>
<td>$\frac{\pi}{40}$</td>
<td>3.9579782897</td>
<td>3.9579782444</td>
<td>$4.523 \times 10^{-8}$</td>
</tr>
<tr>
<td>$\frac{\pi}{20}$</td>
<td>3.8367443738</td>
<td>3.8367442991</td>
<td>$7.468 \times 10^{-8}$</td>
</tr>
<tr>
<td>$\frac{3\pi}{40}$</td>
<td>3.6439387932</td>
<td>3.6439387036</td>
<td>$8.959 \times 10^{-8}$</td>
</tr>
<tr>
<td>$\frac{\pi}{10}$</td>
<td>3.3876357138</td>
<td>3.3876356206</td>
<td>$9.313 \times 10^{-8}$</td>
</tr>
<tr>
<td>$\frac{\pi}{8}$</td>
<td>3.0762425517</td>
<td>3.0762424637</td>
<td>$8.804 \times 10^{-8}$</td>
</tr>
<tr>
<td>$\frac{3\pi}{20}$</td>
<td>2.7184121103</td>
<td>2.7184120337</td>
<td>$7.666 \times 10^{-8}$</td>
</tr>
<tr>
<td>$\frac{7\pi}{40}$</td>
<td>2.3229629854</td>
<td>2.3229629245</td>
<td>$6.091 \times 10^{-8}$</td>
</tr>
<tr>
<td>$\frac{\pi}{5}$</td>
<td>1.8988043202</td>
<td>1.8988042779</td>
<td>$4.227 \times 10^{-8}$</td>
</tr>
<tr>
<td>$\frac{9\pi}{40}$</td>
<td>1.4548614961</td>
<td>1.4548614743</td>
<td>$2.189 \times 10^{-8}$</td>
</tr>
<tr>
<td>$\frac{\pi}{4}$</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>0</td>
</tr>
</tbody>
</table>

The maximum errors between the proposed method and the BSI method for $N = 20, 50$ and 100 are demonstrated in Table 9. Again, our method produced better approximations compared with BSI.

Table 9. Comparison of maximum errors for Problem 4.

<table>
<thead>
<tr>
<th>$N$</th>
<th>Max-Norm BSI [27]</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>$6.30 \times 10^{-5}$</td>
<td>$9.31 \times 10^{-8}$</td>
</tr>
<tr>
<td>50</td>
<td>$1.01 \times 10^{-5}$</td>
<td>$2.40 \times 10^{-8}$</td>
</tr>
<tr>
<td>100</td>
<td>$2.52 \times 10^{-6}$</td>
<td>$1.50 \times 10^{-10}$</td>
</tr>
</tbody>
</table>

7. Discussion

The numerical results, exact solution and absolute error for each problem at specific values of subintervals are presented in Tables 2, 4, 6 and 8. Additionally, for each problem, the details of the maximum errors of the existing methods at different values of $N$ are given in Tables 3, 5, 7 and 9. From our derivations, the truncation error of the proposed method was $O(h^5)$ accurate. On the other hand, LSM produced truncation errors that were $O(h^4)$ accurate, while FDM, FEM, FVM, BSI and ECBI were $O(h^2)$ accurate. Due to this fact, the
approximation obtained by the proposed method was more accurate compared with the others. One potential extension to the present study is the generation of optimal trajectories, obtained from variational calculus to optimal control problems when the boundary conditions are given.

8. Conclusions

This study set out to numerically solve linear two-point BVPs. The method presented was based on a CBS basis function that used the new approximation for the second-order derivative. Four examples were considered and compared with the exact solutions to validate the accuracy of the proposed method. Investigation of the proposed method as a solution to the four problems also showed that it gave more precise and accurate results compared with FDM, FEM, FVM, ECBI(N), ECBI(B), BSI and LSM by calculating the maximum error. We found that the errors decreased, which led to higher accuracy as the step size increased. By performing the error analysis, we found that the proposed method provided fifth-order accuracy. Hence, it was concluded that the proposed method was effective to solve linear two-point BVPs.

Author Contributions: Conceptualization, B.L., S.A.A.K. and I.H.; methodology, B.L., S.A.A.K. and I.H.; software, B.L. and S.A.A.K.; validation, B.L., S.A.A.K. and I.H.; formal analysis, B.L., S.A.A.K. and I.H.; investigation, B.L.; writing—original draft preparation, B.L.; writing—review and editing, B.L., S.A.A.K. and I.H.; supervision, S.A.A.K. and I.H.; funding acquisition, I.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by I.H.’s UKM grant number GP-2020-K006388.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the anonymous referees for their expert advice and suggestions, which helped to improve this paper. Latif, B. is thankful to the Ministry of Higher Education (MOHE) of Malaysia and Universiti Teknologi MARA (UiTM) for the full scholarship (SLAB).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a, b$</td>
<td>End points of the interval</td>
</tr>
<tr>
<td>$h$</td>
<td>Step size</td>
</tr>
<tr>
<td>$B_i(x)$</td>
<td>Cubic b-spline basis function</td>
</tr>
<tr>
<td>$B'_i(x)$</td>
<td>First derivative of the cubic b-spline basis function</td>
</tr>
<tr>
<td>$B''_i(x)$</td>
<td>Second derivative of the cubic b-spline basis function</td>
</tr>
<tr>
<td>$\sigma_i$</td>
<td>Unknown real coefficients</td>
</tr>
<tr>
<td>$x_i$</td>
<td>Mesh point</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of mesh points</td>
</tr>
<tr>
<td>$Y(x_i)$</td>
<td>Cubic b-spline approximation, $Y_j$</td>
</tr>
<tr>
<td>$Y'(x_i)$</td>
<td>First-order cubic b-spline approximation, $S_j$</td>
</tr>
<tr>
<td>$Y''(x_i)$</td>
<td>Second-order cubic b-spline approximation, $S_j$</td>
</tr>
</tbody>
</table>
\( O(h^2) \) Truncation error (order two)

\( \tilde{S}_j \) New approximation for \( y''(x) \)

\( Z_1, Z_2, Z_3 \) Parameters

\( \eta_1, \eta_2 \) Constants

\( f(x) \) Continuous function

\( A \) Coefficient matrix of order \( n+3 \)

\( c \) Unknown column vector

\( b \) Column matrix of order \( n+3 \)

\( E^\mu \left( Y'(x_j) \right) \) Operator notation of \( Y'(x_{j+\mu}) \)

\( Z \) Integer

\( D \) Derivative with respect to \( x, \frac{d}{dx} \)

\( O(h^6) \) Truncation error (order six)

\( O(h^4) \) Truncation error (order four)

\( O(h^5) \) Truncation error (order five)

\( e(x) \) Error term

\( L_\infty \) Maximum error

References