MKSOR ITERATIVE METHOD WITH CUBIC B-SPLINE APPROXIMATION FOR SOLVING TWO-POINT BOUNDARY VALUE PROBLEMS

M. N. Suardi, N. Z. F. M. Radzuan and J. Sulaiman

Mathematics with Economics Programmes, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia

Published online: 17 October 2017

ABSTRACT

In this study, two-point boundary value problems have been discretized by using cubic B-spline discretization scheme to derive the cubic B-spline approximation equations that corresponds. Then, this approximation equation is used to develop system of cubic B-spline approximation equations. To get the numerical solutions, there are three iterative methods such as Gauss-Seidel (GS), Successive Over Relaxation (SOR) and Modified Kaudd Successive Over Relaxation (MKSOR) used to solve the generated systems of linear equations. For the purpose of comparison, the GS iterative method has been designated as a control method for the SOR and MKSOR iterative methods. Three examples of problems also have been considered to test the effectiveness of these proposed iterative methods. From the numerical results, MKSOR iterative method is superior method in terms of number of iterations and computational time.

Keywords: cubic B-spline approximation; two-point boundary value problem; MKSOR iteration.

Author Correspondence, e-mail: jumat@ums.edu.my
doi: http://dx.doi.org/10.4314/jfas.v9i5s.42

1. INTRODUCTION
The B-spline method has been founded by a Frenchman who is a mathematician and engineer, Pierre Bezier. This method is essentially based on the theories that have been developed by P. De Casteljau but Pierre Bezier has fixed the loopholes that exist in order to be a strong theory in the early 1960s [1]. Thus, the B-spline curve can be defined as

\[ y(t) = \sum_{p=0}^{N} C_p \cdot \beta_{p,d}(x), \quad 0 \leq t \leq 1 \]  

(1)

where \( C_p \) is the control point and \( \beta_{p,d}(t) \) is a B-spline basis functions. B-spline function can also be expressed as [2]

\[ \beta_{p,d}(t) = \frac{t-t_p}{t_{p+d-1}-t_p} \beta_{p,d-1}(t) + \frac{t_{p+d}-t_p}{t_{p+d+1}-t_p} \beta_{p+1,d-1}(t) \]  

(2)

with condition

\[ \beta_{p,0}(t) = \begin{cases} 1, & t \in [t_p, t_{p+1}] \\ 0, & \text{otherwise} \end{cases} \]  

(3)

Numerical solution in solving two-point boundary value problems is important to explain many problems involving science, physics and engineering phenomena. Therefore, various numerical methods have been developed to solve and explain all of these problems. As a result, some researchers used Sinc-Galerkin method and modifications decomposition [3], Adomain decomposition method [4] and hybrid Galerkin method [5]. Apart from these methods, the shooting method based on the initial boundary approach to solve two-point boundary value problems [6], the spline solution based on quadratic and cubic spline schemes [7-8] and B-spline method [9]. However, this paper focuses on obtaining the cubic B-spline solution over cubic B-spline approximation linear equations via GS, SOR and MKSORiterative methods. The intention to describe the efficiency results for these three iterative methods, firstly, let us consider two-point boundary value problems being defined as

\[ y'' + f(t)y' + g(t)y = r(t), \quad x \in [t_0, t_N] \]  

(4)

subject to two boundary conditions

\[ y(x_0) = a, \quad y(x_N) = b \]  

(5)

with \( y(t_0) \) is the initial boundary represented by \( a \) and \( y(x_N) \) is the end boundary represented by \( b \) for two-point boundary value problem [7].
2. CUBIC B-SPLINE APPROXIMATION EQUATIONS

In this section, the process of discretization must be imposed to derive B-spline approximation equations for constructing a system of linear equations. However, this paper proposes the discretization of problem (4) through the cubic B-spline discretization scheme. Prior to that, let us consider the cubic B-spline function can be defined as [10]

\[
\beta(t) = \begin{cases} 
\frac{t-p}{t_{p+2}-t_p} \beta_0(t) + \frac{t-p}{t_{p+2}-t_{p+1}} \beta_{1,0}(t) + \frac{t-p}{t_{p+2}-t_{p+1}} \beta_{1,0}(t) & \text{if } t \in [t_p, t_{p+1}] \\
\frac{t-p}{t_{p+3}-t_p} \beta_1(t) + \frac{t-p}{t_{p+3}-t_{p+1}} \beta_{2,1}(t) + \frac{t-p}{t_{p+3}-t_{p+1}} \beta_{2,1}(t) & \text{if } t \in [t_{p+1}, t_{p+2}] \\
\frac{t-p}{t_{p+4}-t_p} \beta_2(t) + \frac{t-p}{t_{p+4}-t_{p+1}} \beta_{3,2}(t) + \frac{t-p}{t_{p+4}-t_{p+1}} \beta_{3,2}(t) & \text{if } t \in [t_{p+2}, t_{p+3}] \\
\end{cases}
\]

Then, simplify Equation (6), the following are the formulation of the cubic B-spline functions at the several different intervals

\[
\beta(t) = \begin{cases} 
(t-t_p)^3, & \text{if } t \in [t_p, t_{p+1}] \\
k_1, & \text{if } t \in [t_{p+1}, t_{p+2}] \\
k_2, & \text{if } t \in [t_{p+2}, t_{p+3}] \\
(t_{p+4} - t)^3, & \text{if } t \in [t_{p+3}, t_{p+4}] \\
\end{cases}
\]

where

\[
k_1 = h^3 + 3h^2(t - t_{p+1}) + 3h(t - t_{p+1})^2 + 3(t - t_{p+1})^3,
\]

\[
k_2 = h^3 + 3h^2(t_{p+3} - t) + 3h(t_{p+3} - t)^2 + 3(t - t_{p+3})^3.
\]

Then, simplify Equation (6), the following are the formulation of the cubic B-spline functions at the several different intervals

\[
\beta(t) = \begin{cases} 
(t-t_{p-1})^3, & \text{if } t \in [t_{p-1}, t_p] \\
k_3, & \text{if } t \in [t_p, t_{p+1}] \\
k_3, & \text{if } t \in [t_{p+1}, t_{p+2}] \\
(t_{p+3} - t)^3, & \text{if } t \in [t_{p+2}, t_{p+3}] \\
\end{cases}
\]

where
\( k_3 = h^3 + 3h^2(t - t_p) + 3h(t - t_p)^2 + 3(t - t_p)^3, \)

\( k_4 = h^3 + 3h^2(t_{p+2} - t) + 3h(t_{p+2} - t)^2 + 3(t_{p+2} - t)^3. \)

\[
\beta_{p-2,3}(t) = \begin{cases} 
(t - t_{p-2})^3, & t \in [t_{p-2}, t_{p-1}] \\
k_5, & t \in [t_{p-1}, t_p] \\
k_6, & t \in [t_p, t_{p+1}] \\
(t_{p+2} - t)^3, & t \in [t_{p+1}, t_{p+2}] 
\end{cases}
\] (9)

where

\( k_5 = h^3 + 3h^2(t - t_{p-1}) + 3h(t - t_{p-1})^2 + 3(t - t_{p-1})^3, \)

\( k_6 = h^3 + 3h^2(t_{p+1} - t) + 3h(t_{p+1} - t)^2 + 3(t_{p+1} - t)^3. \)

\[
\beta_{p-3,3}(t) = \begin{cases} 
(t - t_{p-3})^3, & t \in [t_{p-3}, t_{p-2}] \\
k_7, & t \in [t_{p-2}, t_{p-1}] \\
k_8, & t \in [t_{p-1}, t_p] \\
(t_{p+1} - t)^3, & t \in [t_p, t_{p+1}] 
\end{cases}
\] (10)

where

\( k_7 = h^3 + 3h^2(t - t_{p-2}) + 3h(t - t_{p-2})^2 + 3(t - t_{p-2})^3, \)

\( k_8 = h^3 + 3h^2(t_p - t) + 3h(t_p - t)^2 + 3(t_p - t)^3. \)

By considering the cubic B-spline function being expressed in Equations (7)-(10), the function can be stated as

\[
\begin{align*}
\beta_{p,3}(t_p) &= 0 \\
\beta_{p-1,3}(t_p) &= \frac{1}{6} \\
\beta_{p-2,3}(t_p) &= \frac{4}{6} \\
\beta_{p-3,3}(t_p) &= \frac{1}{6}
\end{align*}
\] (11)

the first derivative of functions (11) at the \( t = t_p \) can be shown as

\[
\begin{align*}
\beta'_{p,3}(t_p) &= 0 \\
\beta'_{p-1,3}(t_p) &= \frac{1}{2h} \\
\beta'_{p-2,3}(t_p) &= \frac{1}{6h} \\
\beta'_{p-3,3}(t_p) &= -\frac{1}{2h}
\end{align*}
\] (12)
and the second derivative as
\[
\begin{align*}
\beta''_{p,3}(t_p) &= 0 \\
\beta''_{p-1,3}(t_p) &= \frac{1}{h^2} \\
\beta''_{p-2,3}(t_p) &= -\frac{2}{h^2} \\
\beta''_{p-3,3}(t_p) &= \frac{1}{h^2}
\end{align*}
\] (13)

Consider the problem in Equation (4) and then the derivation of the approximation, Equation (1) can be rewritten as
\[y(t) = C_{-3} \cdot \beta_{-3,3}(t) + C_{-2} \cdot \beta_{-2,3}(t) + C_{-1} \cdot \beta_{-1,3}(t) + C_0 \cdot \beta_{0,3}(t) + C_1 \cdot \beta_{1,3}(t) + C_2 \cdot \beta_{2,3}(t) + C_3 \cdot \beta_{3,3}(t) + C_4 \cdot \beta_{4,3}(t) + C_5 \cdot \beta_{5,3}(t) + C_6 \cdot \beta_{6,3}(t) + C_7 \cdot \beta_{7,3}(t) + C_8 \cdot \beta_{8,3}(t) + C_9 \cdot \beta_{9,3}(t) + R_p\] (14)

for the case of \(n = 8\) and \(C_i\) are unknown coefficients. Then, let us impose the first derivative and the second derivative over Equation (14) and substitute into Equation (4), cubic B-spline approximation equation of problem (14) is acquired and can be stated as
\[\alpha_p \cdot C_{p-3} + \beta_p \cdot C_{p-2} + \gamma_p \cdot C_{p-1} = R_p\] (15)

where
\[
\alpha_p = \frac{1}{h^2} - \frac{p_p}{2h} + \frac{q_p}{6},
\]
\[
\beta_p = -\frac{2}{h^2} + \frac{p_p}{6h} + \frac{4q_p}{6},
\]
\[
\gamma_p = \frac{1}{h^2} + \frac{p_p}{2h} + \frac{q_p}{6}.
\]

for \(p = 1, 2, 3, \ldots, 8\). Furthermore, the approximation Equation (15) will be used to carry out a system of linear equations in matrix form generally as
\[
A \theta = R
\] (16)

where
\[
A = \begin{bmatrix}
\alpha_0 & \beta_0 & \gamma_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \alpha_1 & \beta_1 & \gamma_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \alpha_2 & \beta_2 & \gamma_2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \alpha_3 & \beta_3 & \gamma_3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \alpha_4 & \beta_4 & \gamma_4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \alpha_5 & \beta_5 & \gamma_5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha_6 & \beta_6 & \gamma_6 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_7 & \beta_7 & \gamma_7 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_8 & \beta_8 \\
\end{bmatrix}
\]
\[
\theta = [\theta_{-2} \theta_{-1} \theta_0 \theta_1 \theta_2 \theta_3 \theta_4 \theta_5 \theta_6]^T,
\]
Clearly, \( A, \theta \) and \( R \) are known respectively as the coefficient matrix, unknown vector and known vector. In order to obtain an approximate solution of linear equations, the coefficients matrix, \( A \) in Equation (16) must fulfill the positive definite, \( |a_{pp}| \geq \sum_{p \neq q} |a_{pq}| \).

### 3. FORMULATION OF ITERATIVE METHODS

By referring the first section, the system of linear Equation (16) will be solved through GS, SOR and MKSOR iterative methods. To facilitate us for the following discussion, formulation of GS, SOR and MKSOR iterative methods will be presented in matrix and/or iterative form.

#### 3.1. GS Iteration Scheme

Systems of linear Equation (16) can be manipulated to produce a variety of iteration matrix schemes. Therefore, by manipulating the coefficient matrix, \( A \) in linear system (16), let the matrix \( A \) be expressed as

\[
A = L + D + V
\]

(17)

where \( L \) is strictly lower matrix, \( D \) is strictly upper matrix and \( V \) is diagonal matrix respectively. By using the matrix decomposition in Equation (17), the linear Equation (16) can be rewritten as

\[
(L + D + V)\theta = R
\]

(18)

Furthermore, the GS iteration scheme can be constructed in the iterative form generally as

\[
\theta^{r+1} = -(L + D)^{-1}V\theta^{r} + (L + D)^{-1}R
\]

(19)

or the general formula for the iterative method is given as

\[
\theta_{p}^{(r+1)} = \frac{1}{a_{pq}} \left( R_{p} - \sum_{q=1}^{p-1} a_{pq} \theta^{(r+1)}_{q} - \sum_{q=p+1}^{N} a_{pq} c_{q}^{(r)} \right)
\]

(20)

with \( p = -2, -1, 0, 1, ..., N - 2 \).

#### 3.2. SOR Iteration Scheme

Young studies [11-14] also introduced the SOR iterative method. This method improves the GS iterative method by adding the relaxation parameter, \( \omega \) which aims to accelerate the convergence rate and reduce error approximation solution. The value of \( \omega \) does not depend on the value of \( i \) and \( k \) but the range value of \( \omega \) is given as 1 \( \leq \omega < 2 \) [15]. The numerical
solution for SOR method shows more accurate if the selected value of $\omega$ is the optimal value. The general formula for the SOR method is given as [16]

$$\theta_p^{(r+1)} = (1 - \omega)\theta_p^{(r)} + \frac{\omega}{a_{pp}} \left( R_p - \sum_{q=1}^{p-1} a_{pq} \theta_q^{(r+1)} - \sum_{q=p+1}^{N} a_{pq} \theta_q^{(r)} \right)$$

for $p = -2, -1, 0, 1, ..., N - 2$. As considering $\omega = 1$, the SOR method will perform as GS method.

### 3.3. MKSOR Iteration Scheme

Due to the advantage of the SOR iterative method, the formulation of an MKSOR iterative scheme is based on SOR iterative scheme but this method has been modified to form a new method [17]. MKSOR method has considered the implementation of the red-black ordering strategy, by using two relaxation parameters, $\omega_1^*$ and $\omega_2^*$. For example, the first parameter ”$\omega_1^*$” performed on the red and the second parameter “$\omega_2^*$” is also applied to black rule. The general formula for the MKSOR method can be declared as

$$\theta_p^{(r+1)} = \frac{1}{1 + \omega_1^*} \left( \theta_p^{(r)} + \frac{\omega_1^*}{a_{pp}} \left( R_p - \sum_{q=1}^{p-1} a_{pq} \theta_q^{(r+1)} - \sum_{q=p+1}^{N} a_{pq} \theta_q^{(r)} \right) \right)$$

for $p = -2, 0, 2, 4, ..., N - 2$. Based on Equations (22) and (23), algorithm 1 explains the implementation of MKSOR iteration scheme.

Algorithm 1: MKSOR scheme

i. Set initial value $\theta^{(0)} = 0$.

ii. Calculate the coefficient matrix, A.

iii. Calculate the vector, R.

iv. For $p = -2, 0, 2, 4, ..., N - 2$, calculate Equation (22).

v. For $p = -1, 1, 3, 5, ..., N - 1$, calculate Equation (23).
vi. Check the convergence test, \( |\theta_p^{(r+1)} - \theta_p^{(r)}| < \varepsilon = 10^{-10} \). If converge, go to step (vii).

Otherwise, repeat from step (iv).

vii. Show numerical solution.

If taking \( \omega_1^* = \omega_2^* \), then the MKSOR iterative method can be reduced to the Red-Black KSOR iterative method.

4. RESULTS AND DISCUSSION

Three examples of two-point boundary value problems have been considered to examine the effectiveness of GS, SOR and MKSOR iterative methods by considering the cubic B-spline approximation equation. Comparison of these three iterative methods will be measured based on three parameters which is number of iterations (Iter), computational time in seconds (Time) and maximum error (Error). Then, the implementation of these three proposed iterative methods has considered the tolerance error at different grid sizes that is constant in which its value is given as \( \varepsilon = 10^{-1} \).

4.1. Problem 1 [18]

\[
y''' - y' = -e^{(t-1)-1}, \ t \in [0,1]
\]  
(24)

The analytical solution for problem (24) is

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

4.2. Problem 2 [19]

\[-y''' - 2y' + 2y = e^{-2t}, \ t \in [0,1]
\]  
(25)

The analytical solution for problem (25) is

\[
y(t) = \frac{1}{2} e^{-(1+\sqrt{3})} + \frac{1}{2} e^{-2t}, \ t \in [0,1]
\]

4.3. Problem 3 [5]

\[
y''' - 4y = \text{kosh}(1), \ t \in [0,1]
\]  
(26)

The analytical solution for problem (26) is

\[
y(t) = \text{kosh}(2t - 1) - \text{kosh}(1), \ t \in [0,1]
\]

Based on these three problems in Equations (24), (25) and (26), all results of numerical experiments were also recorded in Tables 1, 2 and 3. After analyzing the numerical results...
are obtained in these tables, clearly show that the SOR and MKSOR iterative methods have less number of iterations and more faster in term of computational time than the GS method at different values of grid sizes, m = 32, 64, 128, 256, 512 and 1024.

**Table 1.** Result of the number of iterations (Iter), computational time (Time) and maximum absolute error (Error) for problem 1

<table>
<thead>
<tr>
<th>M</th>
<th>GS</th>
<th>SOR</th>
<th>MKSOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>1701</td>
<td>103.0</td>
<td>95.0</td>
</tr>
<tr>
<td></td>
<td>(w=1.8215)</td>
<td>(w=-2.2170)</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>6248</td>
<td>206.0</td>
<td>178.0</td>
</tr>
<tr>
<td></td>
<td>(w=1.9101)</td>
<td>(w=-2.1028)</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>22753</td>
<td>392.0</td>
<td>353.0</td>
</tr>
<tr>
<td></td>
<td>(w=1.9542)</td>
<td>(w=-2.0475)</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>82043</td>
<td>770.0</td>
<td>663.0</td>
</tr>
<tr>
<td></td>
<td>(w=1.99768)</td>
<td>(w=-2.0247)</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>292276</td>
<td>1526.0</td>
<td>1346.0</td>
</tr>
<tr>
<td></td>
<td>(w=1.9879)</td>
<td>(w=-2.0116)</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>1025490</td>
<td>6334.0</td>
<td>2448</td>
</tr>
<tr>
<td></td>
<td>(1.9910)</td>
<td>(w=-2.0061)</td>
<td></td>
</tr>
</tbody>
</table>

**Time (Second)**

<table>
<thead>
<tr>
<th></th>
<th>GS</th>
<th>SOR</th>
<th>MKSOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>0.22</td>
<td>0.05</td>
<td>0.03</td>
</tr>
<tr>
<td>64</td>
<td>0.65</td>
<td>0.06</td>
<td>0.05</td>
</tr>
<tr>
<td>128</td>
<td>2.18</td>
<td>0.15</td>
<td>0.12</td>
</tr>
<tr>
<td>256</td>
<td>7.04</td>
<td>0.30</td>
<td>0.24</td>
</tr>
<tr>
<td>512</td>
<td>26.67</td>
<td>0.33</td>
<td>0.28</td>
</tr>
<tr>
<td>1024</td>
<td>112.06</td>
<td>1.02</td>
<td>0.56</td>
</tr>
</tbody>
</table>

**Error**

<table>
<thead>
<tr>
<th></th>
<th>GS</th>
<th>SOR</th>
<th>MKSOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From the numerical results as obtained in Table 1, it shows that the number of iterations of MKSOR iterative method has declined approximately by 84.42-99.76% as compared with GS method. In addition, MKSOR iterative method is faster than GS iterative method in term of computational time where the range is 86.36-99.50%. It means that the MKSOR iterative method have less number of iterations and faster in computational time than GS and SOR iterative methods.

**Table 2.** Result of the number of iterations (Iter), computational time (Time) and maximum absolute error (Error) for problem 2

<table>
<thead>
<tr>
<th>M</th>
<th>GS</th>
<th>SOR</th>
<th>MKSOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>1415</td>
<td>126.0</td>
<td>115.0</td>
</tr>
<tr>
<td></td>
<td>(w=1.7950)</td>
<td>(w=-2.2536)</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>5225</td>
<td>249.0</td>
<td>229.0</td>
</tr>
<tr>
<td></td>
<td>(w=1.8911)</td>
<td>(w=-2.1201)</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>19143</td>
<td>493.0</td>
<td>454.0</td>
</tr>
<tr>
<td></td>
<td>(w=1.9436)</td>
<td>(w=-2.0585)</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>69532</td>
<td>977.0</td>
<td>902.0</td>
</tr>
<tr>
<td></td>
<td>(w=1.9711)</td>
<td>(w=-2.0290)</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>249932</td>
<td>1933.0</td>
<td>1790.0</td>
</tr>
<tr>
<td></td>
<td>(w=1.9853)</td>
<td>(w=-2.0145)</td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>886861</td>
<td>5070.0</td>
<td>3547.0</td>
</tr>
<tr>
<td></td>
<td>(w=1.9910)</td>
<td>(w=-2.0073)</td>
<td></td>
</tr>
</tbody>
</table>
From the numerical results are observed in Table 2, it can be concluded that the MKSOR iterative method has less the number of iterations by 91.87-99.60% as compared with GS method. Other than that, in term of computational time, the MKSOR iterative method is faster than GS iterative method with the range is 87.88-99.29%. It shows that the MKSOR iterative method is much better than GS and SOR iterative methods for solving the second problem involving two-point boundary value problems.

Table 3. Result of the number of iterations (Iter), computational time (Time) and maximum absolute error (Error) for problem 3
From the numerical results as recorded in Table 3, it can be observed that the MKSOR iterative method has a lesser amount of the number of iterations by 93.44-99.72% as compared with GS method. Similar in term of computational time, implementations of MKSOR iterative method with the range 89.29-99.41% are faster than GS iterative method. It means that the MKSOR iterative method obtains less number of iterations and faster in computational time than GS and SOR iterative methods.
Table 4. Depreciation percentage of the number of iterations (Iter) and computational time (Time) for the SOR and MKSOR compared with GS iterative method

<table>
<thead>
<tr>
<th>Problem</th>
<th>Iter</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SOR</td>
<td>MKSOR</td>
</tr>
<tr>
<td>Problem 1</td>
<td>93.94-99.48%</td>
<td>94.42-99.76%</td>
</tr>
<tr>
<td></td>
<td>77.27-99.08%</td>
<td>86.36-99.50%</td>
</tr>
<tr>
<td>Problem 2</td>
<td>91.10-99.43%</td>
<td>91.87-99.60%</td>
</tr>
<tr>
<td></td>
<td>80.30-99.08%</td>
<td>87.88-99.29%</td>
</tr>
<tr>
<td>Problem 3</td>
<td>92.77-99.94%</td>
<td>93.44-99.72%</td>
</tr>
<tr>
<td></td>
<td>82.14-99.11%</td>
<td>89.29-99.41%</td>
</tr>
</tbody>
</table>

Based on the numerical results as obtained in Table 4 with GS iterative method as a control, it can be observed that MKSOR iterative method has reduced number of iterations approximately 91.87%-99.76% and computational time approximately 86.36%-99.50%. Therefore, it is proven that the MKSOR iterative method is more efficient in terms of number of iterations and computational time as compared with GS and SOR methods.

5. CONCLUSION

In conclusion, the cubic B-spline approximation equation with GS, SOR and MKSOR iterative methodsto solve two-point boundary value problems has been studied. The numerical results with the three selected problems indicated that the proposed MKSOR iterative method requires much lesser number of iterations and computational time in obtaining approximate solution for the proposed problems as compared to the other two proposed iterative methods. Overall, the three proposed iterative methods are good in term of accuracy. However, the overall numerical results recorded were obtained through iterative methods based on the concept of full sweep. Therefore, further studies should be continued in the review of the half-sweep iteration concept[20-21] and the quarter-sweep iteration concept[22]. Apart from these three proposed iterative methods which are categorized as a family of one-step iterative methods, further observations should be made to investigate the efficiency of the two-step iteration family such as AM [23], AGE, IADE [24] and QSAM [25]with the B-spline approximation approach.
6. REFERENCES


How to cite this article: