Journal of Fundamental and Applied Sciences

ISSN 1112-9867

Available online at http://www.jfas.info

Research Article

Special Issue

# 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-splineapproximation; 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

$$y(t) = \sum_{p=0}^{N} C_p \cdot \beta_{p,d}(x), \qquad 0 \le t \le 1 \tag{1}$$

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]

$$\beta_{p,d}(t) = \frac{t-t_p}{t_{p+d-1}-t_p}\beta_{p,d-1}(t)\frac{t_{p+d}-t_p}{t_{p+d}-t_{p+1}}\beta_{p+1,d-1}(t) \tag{2}$$

with condition

$$\beta_{p,0}(t) = \begin{cases} 1 & ,t \in [t_p, t_{p+1}] \\ 0 & ,\text{otherwise} \end{cases} \tag{3}$$

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 iterativemethods, firstly,let us consider two-point boundary value problems being defined as

$$y'' + f(t)y' + g(t)y = r(t)\,, x \in [t_0, t_N] \tag{4}$$

subjectto two boundary conditions

$$y(x_0) = a, \;\; y(x_N) = b \tag{5}$$

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 boundaryvalue problem [7].

## 2. CUBIC B-SPLINE APPROXIMATION EQUATIONS

In this section, the processof discretization must be imposed toderive B-spline approximation equation 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]

$$\beta_{p,3}(t) = \frac{t-t_p}{t_{p+3}-t_p}\left[\frac{t-t_p}{t_{p+2}-t_p}\left[\frac{t-t_p}{t_{p+1}-t_p}\beta_{p,0}(t) + \frac{t_{p+2}-t}{t_{p+2}-t_{p+1}}\beta_{p+1,0}(t)\right] + \frac{t_{p+3}-t}{t_{p+3}-t_{p+1}}\left[\frac{t-t_{p+1}}{t_{p+2}-t_{p+1}}\beta_{p+1,0}(t) + \frac{t_{p+3}-t}{t_{p+3}-t_{p+2}}\beta_{p+2,0}(t)\right]\right]$$

$$+\frac{t_{p+4}-t}{t_{p+4}-t_{p+1}}\left[\frac{t-t_{p+1}}{t_{p+3}-t_{p+1}}\left[\frac{t-t_{p+1}}{t_{p+1}-t_p}\beta_{p+1,0}(t) + \frac{t_{p+3}-t}{t_{p+3}-t_{p+2}}\beta_{p+2,0}(t)\right] + \frac{t_{p+4}-t}{t_{p+4}-t_{p+2}}\left[\frac{t-t_{p+2}}{t_{p+3}-t_{p+2}}\beta_{p+2,0}(t) + \frac{t_{p+4}-t}{t_{p+4}-t_{p+3}}\beta_{p+3,0}(t)\right]\right] \quad (6)$$

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

$$\beta_{p,3}(t) = \frac{1}{6h^3}\begin{cases} (t-t_p)^3, & t \in [t_p, t_{p+1}] \\ k_1, & t \in [t_{p+1}, t_{p+2}] \\ k_2, & t \in [t_{p+2}, t_{p+3}] \\ (t_{p+4}-t)^3, & t \in [t_{p+3}, t_{p+4}] \end{cases} \quad (7)$$

where

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

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

$$\beta_{p-1,3}(t) = \frac{1}{6h^3}\begin{cases} (t-t_{p-1})^3, & t \in [t_{p-1}, t_p] \\ k_3, & t \in [t_p, t_{p+1}] \\ k_4, & t \in [t_{p+1}, t_{p+2}] \\ (t_{p+3}-t)^3, & t \in [t_{p+2}, t_{p+3}] \end{cases} \quad (8)$$

where

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

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

$$\beta_{p-2,3}(t) = \frac{1}{6h^3}\begin{cases} (t - t_{p-2})^3, & t \in [t_{p-2}, t_{p-1}] \\ k_5, & t \in [t_{p-1}, t_p] \\ k_6, & t \in [t_p, t_{p+1}] \\ (t_{p+2} - t)^3, & t \in [t_{p+1}, t_{p+2}] \end{cases} \quad (9)$$

where

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

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

$$\beta_{p-3,3}(t) = \frac{1}{6h^3}\begin{cases} (t - t_{p-3})^3, & t \in [t_{p-3}, t_{p-2}] \\ k_7, & t \in [t_{p-2}, t_{p-1}] \\ k_8, & t \in [t_{p-1}, t_p] \\ (t_{p+1} - t)^3, & t \in [t_p, t_{p-1}] \end{cases} \quad (10)$$

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 functionbeing expressed in Equations (7)-(10), the function can be stated as

$$\left.\begin{aligned} \beta_{p,3}(t_p) &= 0 \\ \beta_{p-1,3}(t_p) &= \frac{1}{6} \\ \beta_{p-2,3}(t_p) &= \frac{4}{6} \\ \beta_{p-3,3}(t_p) &= \frac{1}{6} \end{aligned}\right\} \quad (11)$$

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

$$\left.\begin{aligned} \beta'_{p,3}(t_p) &= 0 \\ \beta'_{p-1,3}(t_p) &= \frac{1}{2h} \\ \beta'_{p-2,3}(t_p) &= \frac{1}{6h} \\ \beta'_{p-3,3}(t_p) &= -\frac{1}{2h} \end{aligned}\right\} \quad (12)$$

and the second derivative as

$$\left.\begin{aligned} \beta''_{p,3}(t_p) &= 0 \\ \beta''_{p-1,3}(t_p) &= \frac{1}{h^2} \\ \beta''_{p-2,3}(t_p) &= -\frac{2}{h^2} \\ \beta''_{p-3,3}(t_p) &= \frac{1}{h^2} \end{aligned}\right\} \quad (13)$$

Consider the problem in Equation (4) and then the derivation of the approximation, Equation (1) can be rewritten as

$$y(t) = C_{-3}\cdot\beta_{-3,3}(t) + C_{-2}\cdot\beta_{-2,3}(t) + C_{-1}\cdot\beta_{-1,3}(t) + C_0\cdot\beta_{0,3}(t) + C_1\cdot\beta_{1,3}(t) + C_2\cdot \beta_{2,3}(t) + C_3\cdot\beta_{3,3}(t) + C_4\cdot\beta_{4,3}(t) + C_5\cdot\beta_{5,3}(t) + C_6\cdot\beta_{6,3}(t) + C_7\cdot\beta_{7,3}(t) \quad (14)$$

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

$$\alpha_p \cdot C_{p-3} + \beta_p \cdot C_{p-2} + \gamma_p \cdot C_{p-1} = R_p \quad (15)$$

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

$$A\theta = \underline{R} \quad (16)$$

where

$$A = \begin{bmatrix} \alpha_0 & \beta_0 & \gamma_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_1 & \beta_1 & \gamma_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_2 & \beta_2 & \gamma_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_3 & \beta_3 & \gamma_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_4 & \beta_4 & \gamma_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_5 & \beta_5 & \gamma_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha_6 & \beta_6 & \gamma_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_7 & \beta_7 & \gamma_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_8 & \beta_8 & \gamma_8 \end{bmatrix}$$

$\theta = [\theta_{-2} \quad \theta_{-1} \quad \theta_0 \quad \theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6]^T$,

$$\underline{R} = [R_0 - \alpha \quad R_1 \quad R_2 \quad R_3 \quad R_4 \quad R_5 \quad R_6 \quad R_7 \quad R_8 - \beta]^T.$$

Clearly, A,$\theta$ and$\underline{R}$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, Ain Equation (16) must fulfill the positive definite,$[a_{pp}] \geq \sum_{p \neq q}[a_{pq}]$.

## 3. FORMULATION OF ITERATIVE METHODS

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

### 3.1.GS Iteration Scheme

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

$$A = L + D + V \tag{17}$$

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

$$(L + D + V)\theta = \underline{R} \tag{18}$$

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

$$\theta^{r+1} = -(L + D)^{-1}V\theta^r + (L + D)^{-1}\underline{R} \tag{19}$$

or the general formula for the iterative method is given as

$$\theta_p^{(r+1)} = \frac{1}{a_{pq}}\left(R_p - \sum_{q=1}^{p-1} a_{pq}\,\theta_q^{(r+1)} - \sum_{q=p+1}^{N} a_{pq}\,c_q^{(r)}\right) \tag{20}$$

with$p = -2, -1, 0, 1, \dots, N - 2$.

### 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]

$$\theta_p^{(r+1)} = (1-\omega)\theta_p{}^{(r)} + \frac{\omega}{a_{pp}}\left(R_p - \sum_{q=1}^{p-1} a_{pq}\,\theta_q{}^{(r+1)} - \sum_{q=p+1}^{N} a_{pq}\,\theta_q{}^{(r)}\right) \tag{21}$$

for $p = -2, -1, 0, 1, \dots, N-2$. 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 thismethod 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 firstparameter ”$\omega_1^*$” 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

$$\theta_p^{(r+1)} = \frac{1}{(1+\omega_1^*)}\left[\theta_p{}^{(r)} + \frac{\omega_1^*}{a_{pp}}\left(R_p - \sum_{q=1}^{p-1} a_{pq}\,\theta_q{}^{(r+1)} - \sum_{q=p+1}^{N} a_{pq}\,\theta_q{}^{(r)} - a_{pq}\theta_q{}^{(r+1)}\right)\right] \tag{22}$$

for $p = -2, 0, 2, 4, \dots, N-2$.

$$\theta_p^{(r+1)} = \frac{1}{(1+\omega_2^*)}\left[\theta_p{}^{(r)} + \frac{\omega_2^*}{a_{pp}}\left(R_p - \sum_{q=1}^{p-1} a_{pq}\,\theta_q{}^{(r+1)} - \sum_{q=p+1}^{N} a_{pq}\,\theta_q{}^{(r)} - a_{pq}\theta_q{}^{(r+1)}\right)\right] \tag{23}$$

for $p = -1, 1, 3, 5, \dots, N-1$. 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 $p = -2, 0, 2, 4, \dots, N-2$, calculate Equation (22).

v. For $p = -1, 1, 3, 5, \dots, N-1$, calculate Equation (23).

vi. Check the convergence test, $\left|\theta_p^{(r+1)} - \theta_p^{(r)}\right| < \varepsilon = 10^{-10}$. If converge, go to step (vii). Otherwise,repeat from step (iv).

vii. Show numerical solution.

Iftaking$\omega_1^* = \omega_2^*$, then the MKSOR iterative method can be reduced to the Red-BlackKSOR iterative method.

## 4. RESULTS AND DISCUSSION

Three examples of two-point baundary value problems have been considered to examinate the effectivenessof 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]

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

The analytical solution for problem (24) is

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

### 4.2. Problem 2 [19]

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

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]

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

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 areobtained 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

| | **Iter** | | |
|---|---|---|---|
| **M** | **GS** | **SOR** | **MKSOR** |
| 32 | 1701 | 103.0 | 95.0 |
| | | (w=1.8215) | (w=-2.2170) |
| 64 | 6248 | 206.0 | 178.0 |
| | | (w=1.9101) | (w=-2.1028) |
| 128 | 22753 | 392.0 | 353.0 |
| | | (w=1.9542) | (w=-2.0475) |
| 256 | 82043 | 770.0 | 663.0 |
| | | (w=1.99768) | (w=-2.0247) |
| 512 | 292276 | 1526.0 | 1346.0 |
| | | (w=1.9879) | (w=-2.0116) |
| 1024 | 1025490 | 6334.0 | 2448 |
| | | (1.9910) | (w=-2.0061) |
| | **Time (Second)** | | |
| | **GS** | **SOR** | **MKSOR** |
| 32 | 0.22 | 0.05 | 0.03 |
| 64 | 0.65 | 0.06 | 0.05 |
| 128 | 2.18 | 0.15 | 0.12 |
| 256 | 7.04 | 0.30 | 0.24 |
| 512 | 26.67 | 0.33 | 0.28 |
| 1024 | 112.06 | 1.02 | 0.56 |
| | **Error** | | |
| | **GS** | **SOR** | **MKSOR** |

| | | | |
|---|---|---|---|
| 32 | 2.8273e-5 | 2.8283e-5 | 2.8284e-5 |
| 64 | 7.0299e-6 | 7.0687e-6 | 7.0759e-6 |
| 128 | 1.6075e-6 | 1.7644e-6 | 1.7516e-6 |
| 256 | 2.1305e-7 | 4.3928e-7 | 4.6000e-7 |
| 512 | 2.4807e-6 | 1.1671e-7 | 7.8049e-8 |
| 1024 | 1.0334e-5 | 1.7631e-8 | 8.8889e-8 |

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

| | | **Iter** | |
|---|---|---|---|
| **M** | **GS** | **SOR** | **MKSOR** |
| 32 | 1415 | 126.0 | 115.0 |
| | | (w=1.7950) | (w=-2.2536) |
| 64 | 5225 | 249.0 | 229.0 |
| | | (w=1.8911) | (w=-2.1201) |
| 128 | 19143 | 493.0 | 454.0 |
| | | (w=1.9436) | (w=-2.0585) |
| 256 | 69532 | 977.0 | 902.0 |
| | | (w=1.9711) | (w=-2.0290) |
| 512 | 249932 | 1933.0 | 1790.0 |
| | | (w=1.9853) | (w=-2.0145) |
| 1024 | 886861 | 5070.0 | 3547.0 |
| | | (w=1.9910) | (w=-2.0073) |
| | | **Time (Second)** | |

| | GS | SOR | MKSOR |
|---|---|---|---|
| 32 | 0.28 | 0.04 | 0.03 |
| 64 | 0.66 | 0.13 | 0.08 |
| 128 | 2.14 | 0.15 | 0.12 |
| 256 | 6.39 | 0.26 | 0.24 |
| 512 | 23.83 | 0.37 | 0.32 |
| 1024 | 102.47 | 0.94 | 0.73 |
| **Error** | | | |
| | **GS** | **SOR** | **MKSOR** |
| 32 | 9.5775e-5 | 9.5768e-5 | 9.5768e-5 |
| 64 | 2.3959e-5 | 2.3929e-5 | 2.3929e-5 |
| 128 | 6.1035e-6 | 5.9823e-6 | 5.9819e-6 |
| 256 | 1.9892e-6 | 1.4978e-6 | 1.4966e-6 |
| 512 | 2.3987e-6 | 3.7911e-7 | 3.7647e-7 |
| 1024 | 8.2376e-6 | 1.2311e-7 | 9.9733e-8 |

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

| **Iter** | | | |
|---|---|---|---|
| **M** | **GS** | **SOR** | **MKSOR** |
| 32 | 1341 | 97.0 (w=1.7941) | 88.0 (w=-2.2590) |
| 64 | 4953 | 193.0 (w=1.8944) | 167.0 (w=-2.1164) |

| | | | |
|---|---|---|---|
| 128 | 18173 | 382.0 (w=1.9439) | 313.0 (w=-2.0590) |
| 256 | 66139 | 724.0 (w=1.9717) | 581.0 (w=-2.0290) |
| 512 | 238353 | 1438.0 (w=1.9858) | 1158.0 (w=-2.0144) |
| 1024 | 848604 | 4739.0 (w=1.9910) | 2357.0 (w=-2.0068) |
| | **Time (Second)** | | |
| | **GS** | **SOR** | **MKSOR** |
| 32 | 0.28 | 0.05 | 0.03 |
| 64 | 0.57 | 0.09 | 0.04 |
| 128 | 1.84 | 0.18 | 0.15 |
| 256 | 6.05 | 0.26 | 0.24 |
| 512 | 22.41 | 0.33 | 0.29 |
| 1024 | 97.2 | 0.86 | 0.57 |
| | **Error** | | |
| | **GS** | **SOR** | **MKSOR** |
| 32 | 1.2400e-4 | 1.2401e-4 | 1.2401e-4 |
| 64 | 3.0963e-5 | 3.0992e-5 | 3.0986e-5 |
| 128 | 7.6296e-6 | 7.7471e-6 | 7.7569e-6 |
| 256 | 1.4644e-6 | 1.9383e-6 | 1.9708e-6 |
| 512 | 1.4058e-6 | 4.8750e-7 | 5.1340e-7 |
| 1024 | 7.4391e-6 | 9.9535e-8 | 6.2049e-8 |

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

| | | SOR | MKSOR |
|---|---|---|---|
| Problem 1 | Iter | 93.94-99.48% | 94.42-99.76% |
| | Time | 77.27-99.08% | 86.36-99.50% |
| Problem 2 | Iter | 91.10-99.43% | 91.87-99.60% |
| | Time | 80.30-99.08% | 87.88-99.29% |
| Problem 3 | Iter | 92.77-99.94% | 93.44-99.72% |
| | Time | 82.14-99.11% | 89.29-99.41% |

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

[1] Choi JW, Curry R E, Elkaim GH. Minimizing the maximum curvature of quadratic bezier curve with a tetragonal concave polygonal boundary constraint. Computer Aided Design,2012, 44(4):311-319

[2] Botella I, Shariff K. B-spline method in fluid dynamics. International Journal of Computational Fluid Dynamics, 2003, 17(12):133-149

[3] El-Gamel M. Comparison of the solution obtained by adomian decomposition and wavelet-galerkin methods of boundary value problems. Applied Mathematics and Computation,2007, 186(1):652-664

[4] Jang B. Two-point boundary value problems by extended adomian decomposition method. Computational and Applied Mathematics, 2007, 219(1):253-262

[5] Mohsen A, Gamel ME. On the galerkin and collocation methods for two point boundary value problems using sinc bases.Computer and Mathematics with Applications,2008, 56(4):930-941

[6] Lin Y, Enszer J A, StadtherrM A. Enclosing all solutions of two-point boundary value problems for ODEs.Computer and Chemical Engineering, 2008, 32(8):1714-1725

[7] Albasiny E L, Hoskin W D. Cubic spline solution to two point boundary value problems. Computer Journal,1969, 12(2):151-153

[8] RamadanM A, LashienI F, ZahraW K. Polynomial and nonpolynomial spline approaches to the numerical solution of second order boundary value problems. Applied Mathematics and Computation,2007, 184(2):476-484

[9] Nazan C, Hikmet C. B-spline methods for solving linear system of second order boundary value problems.Computers and Mathematics with Application, 2008, 57(5):757-762

[10]Chang J, Yang Q, Zhao L. Comparison of B-spline method and finite difference method to solve BVP of linear ODEs.Journal of Computers,2011, 6(10):2149-2155

[11]Young D M. Iterative methods for solving partial difference equations of elliptic type. Transactions of the American Mathematical Society,1954, 76(1):92-111

[12]Young D.M. Iterative solution of large linear systems. London: Academic Press, 1971

[13]Young D M. Second-degree iterative methods for the solution of large linear systems. Journal of Approximation Theory,1972,5(2):137-148

[14]Young D M. Iterative solution of linear systems arising from finite element techniques. InJ. R. Whiteman (Eds.),The mathematics of finite elements and applications II.London: Academic Press, 1976, pp. 439-464

[15]Smith G. D. Numerical solution of partial differential equations: Finite difference methods. Oxford: Clarendon Press, 1978

[16]Youssef I K, Meligy S A. Boundary value problem on triangular domains and MKSOR method. Applied and Computation Mathematics, 2014,3(3):90-99

[17]Youssef I K, Taha A A. On the modified successive overrelaxation method. Applied Mathematics and Computation,2013, 219(9):4601-4613

[18]Caglar H N, Caglar S H, Elfaituri K. B-spline interpolation compared with finite difference, finite element and finite volume methods which applied to two point boundary value problems. Applied Mathematics and Computation,2006,175(1):72-79

[19]Robertson T N. The linear two-point boundary value problem on an infinite interval. Mathematics of Computation, 1971, 25(115):475-481

[20]Abdullah A R.The four point explicit decoupled group (EDG) method: A fast Poisson solver. InternationalJournal Computer Mathematics,1991, 38(1-2):61-70

[21]Ibrahim A,Abdullah AR. Solving the two dimensional diffusion equation by the four point explicitdecoupled group (EDG) iterative method. International Journal Computer Mathematics, 1995, 58(3-4):253-256

[22]Othman M, Abdullah A R. An effcient multigrid Poisson solver. International Journal of Computer Mathematics,1999,71(4):541-553

[23]Ruggiero V, Galligani E. An iterative method for large sparse linear systems on a vector computer.Computers and Mathematics with Applications,1990, 20(1):25-28

[24]Sahimi M S, Ahmad A, Bakar A A. The iterative alternating decomposition explicit (IADE) method to solve the heat conduction equation. International Journal of Computer Mathematics, 1993,47(3-4):219-229

[25]Sulaiman J, Othman M, Hasan M K.A new quarter-sweep arithmetic mean (QSAM) method to solve diffusion equations. Chamchuri Journal of Mathematics, 2009, 1(2):89-99

**How to cite this article**:
Suardi M N, Radzuan N Z F M, and Sulaiman J. MKSOR Iterative Method with Cubic B-Spline Approximation for Solving Two-Point Boundary Value Problems. J. Fundam. Appl. Sci., 2017, *9(5S), 594-608*.