Journal of Engineering and Applied Sciences 14 (3): 693-700, 2019
ISSN: 1816-949X

Performance of Quarter-Sweep SOR Iteration with Cubic B-Spline Scheme for Solving Two-Point Boundary Value Problems

Mohd Norfadli Suardi, Nurul Zafira Farhana Mohd Radzuan and Jumat Sulaiman
Faculty of Science and Natural Resources, Universiti Malaysia Sabah,
88400 Kota Kinabalu, Sabah, Malaysia

Abstract: In this study, we deals with cubic B-spline method to solve two-point boundary value problem. The cubic B-spline approximation equation based on quarter-sweep concept are used to discretize the proposed problem and construct the linear system. The linear system are solved via. the family of SOR iterative methods which is Full-Sweep Gauss-Seidel (FSGS), Full-Sweep Successive Over Relexation (FSSOR), Half-Sweep Successive Over Relexation (HSSOR) and Quarter-Sweep Successive Over Relexation (QSSOR) iterative methods. The performance for the proposed iterative methods are recorded with compared three parameters such as number of iterations, execution time and maximum error. The QSSOR is superior method as compared with FSGS, FSSOR and HSSOR iterative method based on the numerical solution are obtained.

Key words: Cubic B-spline, scheme, QSSOR, iterations, two-point boundary, value problem

INTRODUCTION

Now a days, two-point boundary value problems are used widely to solve many phenomenain science, physics and engineering problems. Due to the advantages, there are many researchers have been interested and give more attention to solve these problems (Robertson, 1971; Wang and Guo, 2008; Aarao *et al*., 2010). Automatically, the various methods are used for solving two-point boundary value problems such as Sinc-Galerkin method and modifications decomposition (Gamel, 2007) A domain decomposition method (Jang, 2008) and hybrid Galerkin method (Mohsen and El-Gamel, 2008). The other methods are used is shooting method (Lin *et al*., 2008) the family of spline and B-spline methods (Ramadan *et al*., 2007). The two-point boundary value problems are defined as follows:

$$y''f(x)y'+g(x)y=r(x),\ x\in[x_0,\ x_n] \tag{1}$$

with the boundary conditions:

$$y(x_0)=a,\ y(x_n)=b \tag{2}$$

where, a and b are assumed as left and right boundary, respectively (Albasiny and Hoskins, 1969).

B-spline method has been used to solve one dimensional problem in partial differential equations where it can give the accurate numerical solutions (Viswanadham and Koneru, 1993; Gardner and Gardner, 1995; Wu and Zhang, 2011). Actually, the idea of B-spline was introduced by Schoenberg. In early 1960s, Pierre Bezier was improved and upgraded the basic idea of B-spline which proposed by P. De Calteljau (Schoenberg, 1946; Choi *et al*., 2012). Also, the quarter-sweep concept is apllied into B-spline method. The utmost imposing this concept is to reduce the convergence rate. Figure 1 shows the full, half and quarter-sweep concepts. As we can see, the difference of all these concept is the value of h are used. For example, the subinterval lenght of the full-sweep iteration is used h = 1 in Fig. 1a. The value of the subinterval lenght for half-sweep and quarter-sweep iterations are 2 and 4 h, respectively. However, the process of computation are same for all these concepts where they will compute all node points of type • only with the attention to get the approximate solution until the convergence criterion is reached. Meanwhile, the direct methods are used to solve the remaining node points (Abdullah, 1991).

In order to get numerical solution in this study, first of all, we consider B-spline function of the form (Choi *et al*., 2012; Suardi *et al*., 2017a, b):

$$y(x)=\sum_{i=0}^{n}C_i\cdot\beta_{i,\,d}(x),\ 0\le x\le 1 \tag{3}$$

where, C_i and $\beta_{i,\,d}(x)$ are assumed as the control point and B-spline basis functions, respectively. The third degree B-spline function can be defined as (Botella and Shariff, 2003):

Corresponding Author: Mohd Norfadli Suardi, Faculty of Science and Natural Resources, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia

$$\beta_{i,\,d}\left(x\right)=\frac{x\text{-}x_i}{x_{i+d-1}\text{-}x_i}\beta_{i,\,d-1}\left(x\right)\frac{x_{i+d}\text{-}x_i}{x_{i+d}\text{-}x_{i+1}}\beta_{i+1,\,d-1}\left(x\right) \quad (4)$$

with the condition:

$$\beta_{i,\,0}\left(x\right)=\begin{cases}1, & x\in\left[x_i,\,x_{i+1}\right]\\ 0, & \text{otherwise}\end{cases} \quad (5)$$

After that we need to descretize the proposed Eq. 1 using cubic B-spline discretization scheme to get the approximation equation of the proposed problem. The approximate equation will lead us to construct the tridiagonal linear system. Then, we will choose the iterative methods to solve the problem that are considered. In this study, the FSGS, FSSOR, HSSOR and QSSOR iterative methods are chosen. The reason for all these iterative methods are selected is that the iterative methods are the natural option for the linear system which has the characteristics large and sparse to obtain the efficient solution (Young, 1971; Hackbusch, 1995; Saad, 1996). The implementation of the iterative methods with quarter-sweep concept is to reduce their convergence rate. Actually the half-sweep iteration concept has been initiated by Abdullah (1991) which he using the Explicit Decoupled Group (EDG) method to solve two-dimensional poisson equations. According to that there are many studies are applied to the half-sweep iteration (Akhir *et al*., 2011; Muthuvalu and Sulaiman, 2011; Dahalan *et al*., 2014; Alibubin *et al*., 2016; Chew and Sulaiman, 2016; Eng *et al*., 2017). Motivated with the half-sweep concept findings, the quarter-sweep iteration has been introduced by Othman and Abdullah (2000) for solving two-dimensional poisson equation via. Modified Explicit Group (MEG). For the quarter-sweep concept these are many researchers that have been used this concept (Othman *et al*., 2014; Sulaiman *et al*., 2004, 2009).

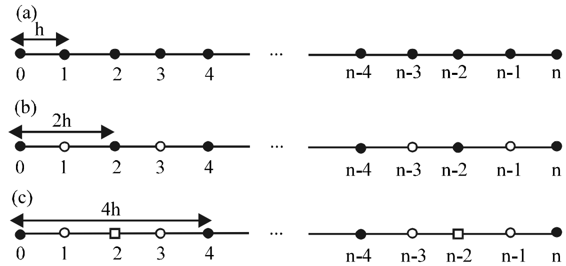

Fig. 1: The distribution of uniform points for; a) Full-sweep; b) Half-sweep and c) Quarter-sweep iteration

MATERIALS AND METHODS

Quarter-sweep cubic B-spline approximation equations: In this study, we discuss the way to derive the two-point boundary value problems using the cubic B-spline discretization scheme. These steps lead to construct the linear system of problem Eq. 1 getting the cubic B-spline approximation equation. First of all, we consider $d = 3$ in Eq. 4 or known as third degree cubic B-spline, the function can be given as (Chang *et al*., 2011):

$$\begin{aligned}\beta_{i,\,3}\left(x\right)=&\frac{x\text{-}x_i}{x_{i+12}\text{-}x_i}\left[\frac{x\text{-}x_i}{x_{i+8}\text{-}x_i}\left[\frac{x\text{-}x_i}{x_{i+4}\text{-}x_i}\beta_{i,\,0}\left(x\right)+\frac{x_{i+8}\text{-}x}{x_{i+8}\text{-}x_{i+4}}\beta_{i+4,\,0}\left(x\right)\right]+\frac{x_{i+12}\text{-}x}{x_{i+12}\text{-}x_{i+4}}\left[\frac{x\text{-}x_{i+4}}{x_{i+8}\text{-}x_{i+4}}\beta_{i+4,\,0}\left(x\right)+\frac{x_{i+12}\text{-}x}{x_{i+12}\text{-}x_{i+8}}\beta_{i+8,\,0}\left(x\right)\right]\right]+\\&\frac{x_{i+16}\text{-}x}{x_{i+16}\text{-}x_{i+4}}\left[\frac{x\text{-}x_{i+4}}{x_{i+12}\text{-}x_{i+4}}\left[\frac{x\text{-}x_{i+4}}{x_{i+4}\text{-}x_i}\beta_{i+4,\,0}\left(x\right)+\frac{x_{i+12}\text{-}x}{x_{i+12}\text{-}x_{i+8}}\beta_{i+8,\,0}\left(x\right)\right]+\frac{x_{i+16}\text{-}x}{x_{i+16}\text{-}x_{i+8}}\left[\frac{x\text{-}x_{i+8}}{x_{i+12}\text{-}x_{i+8}}\beta_{i+8,\,0}\left(x\right)+\frac{x_{i+16}\text{-}x}{x_{i+16}\text{-}x_{i+12}}\beta_{i+12,\,0}\left(x\right)\right]\right]\end{aligned} \quad (6)$$

Then, the cubic B-spline function in Eq. 6 at points x_i, x_{i-4}, x_{i-8} and x_{i-12} where $h = b\text{-}a/n$ can be derived and simplified to:

$$\beta_{i,\,3}\left(x\right)=\frac{1}{6h^3}\begin{cases}\left(x\text{-}x_i\right)^3, & x\in\left[x_i,\,x_{i+4}\right]\\ k_1, & x\in\left[x_{i+4},\,x_{i+8}\right]\\ k_2, & x\in\left[x_{i+8},\,x_{i+12}\right]\\ \left(x_{i+16}\text{-}x\right)^3, & x\in\left[x_{i+12},\,x_{i+16}\right]\end{cases} \quad (7)$$

Where:

$$k_1=h^3+3h^2\left(x\text{-}x_{i+4}\right)+3h\left(x\text{-}x_{i+4}\right)^2+3\left(x\text{-}x_{i+4}\right)^3$$
$$k_2=h^3+3h^2\left(x_{i+12}\text{-}x\right)+3h\left(x_{i+12}\text{-}x\right)^2+3\left(x_{i+12}\text{-}x\right)^3$$

$$\beta_{i-4,\,3}\left(x\right)=\frac{1}{6h^3}\begin{cases}\left(x\text{-}x_{i-4}\right)^3, & x\in\left[x_{i-4},\,x_i\right]\\ k_3, & x\in\left[x_i,\,x_{i+4}\right]\\ k_4, & x\in\left[x_{i+4},\,x_{i+8}\right]\\ \left(x_{i+12}\text{-}x\right)^3, & x\in\left[x_{i+8},\,x_{i+12}\right]\end{cases} \quad (8)$$

Where:

$$k_3=h^3+3h^2\left(x\text{-}x_i\right)+3h\left(x\text{-}x_i\right)^2+3\left(x\text{-}x_i\right)^3$$
$$k_4=h^3+3h^2\left(x_{i+8}\text{-}x\right)+3h\left(x_{i+8}\text{-}x\right)^2+3\left(x_{i+8}\text{-}x\right)^3$$

$$\beta_{i-8,\,3}\left(x\right)=\frac{1}{6h^3}\begin{cases}\left(x\text{-}x_{i-8}\right)^3, & x\in\left[x_{i-8},\,x_{i-4}\right]\\ k_5, & x\in\left[x_{i-4},\,x_i\right]\\ k_6, & x\in\left[x_i,\,x_{i+4}\right]\\ \left(x_{i+8}\text{-}x\right)^3, & x\in\left[x_{i+4},\,x_{i+8}\right]\end{cases} \quad (9)$$

Where:

$$k_5 = h^3+3h^2(x-x_{i-4})+3h(x-x_{i-4})^2+3(x-x_{i-4})^3$$
$$k_6 = h^3+3h^2(x_{i+4}-x)+3h(x_{i+4}-x)^2+3(x_{i+4}-x)^3$$

$$\beta_{i-12,3}(x) = \frac{1}{6h^3}\begin{cases}(x-x_{i-12})^3, & x\in[x_{i-12}, x_{i-8}]\\ k_7, & x\in[x_{i-8}, x_{i-4}]\\ k_8, & x\in[x_{i-4}, x_i]\\ (x_{i+4}-x)^3, & x\in[x_i, x_{i-4}]\end{cases} \quad (10)$$

Where:

$$k_7 = h^3+3h^2(x-x_{i-8})+3h(x-x_{i-8})^2+3(x-x_{i-8})^3$$
$$k_8 = h^3+3h^2(x_i-x)+3h(x_i-x)^2+3(x_i-x)^3$$

By substituting $x = x_i$ into Eq. 7-10, the value for each piecewise function can be stated as:

$$\left.\begin{aligned}\beta_{i,3}(x_i) &= 0\\ \beta_{i-4,3}(x_i) &= \frac{1}{6}\\ \beta_{i-8,3}(x_i) &= \frac{4}{6}\\ \beta_{i-12,3}(x_i) &= \frac{1}{6}\end{aligned}\right\} \quad (11)$$

By applying the first derivative concept into Eq. 7-10, the first derivative functions at point $x = x_i$ can be written as:

$$\left.\begin{aligned}\beta'_{i,3}(x_i) &= 0\\ \beta'_{i-4,3}(x_i) &= \frac{1}{8h}\\ \beta'_{i-8,3}(x_i) &= 0\\ \beta'_{i-12\,3}(x_i) &= \frac{1}{8h}\end{aligned}\right\} \quad (12)$$

Since, the steps to get the Eq. 13 as same as the steps to get Eq. 12, the second derivative of the Eq. 7-10 are obtained as:

$$\left.\begin{aligned}\beta''_{i,3}(x_i) &= 0\\ \beta''_{i-4,3}(x_i) &= \frac{1}{16h^2}\\ \beta''_{i-8,3}(x_i) &= \frac{2}{16h^2}\\ \beta''_{i-12,3}(x_i) &= \frac{1}{16h^2}\end{aligned}\right\} \quad (13)$$

Using the Eq. 3, we substituted and derive the approximation function, thus, we have cubic B-spline approximation equation:

$$\begin{aligned}y(x) = {} & C_{-12}.\beta_{-12,3}(x)+C_{-8}.\beta_{-8,3}(x)+C_{-4}.\beta_{-4,3}(x)+C_0.\beta_{-0\,3}(x)+\\ & C_4.\beta_{4,3}(x)+C_8.\beta_{8,3}(x)+C_{12}.\beta_{12,3}(x)+C_{16}.\beta_{16,3}(x)+\\ & C_{20}.\beta_{20,3}(x)+C_{24}.\beta_{24,3}(x)+C_{26}.\beta_{26,3}(x)\end{aligned} \quad (14)$$

where, C_i is unknown coefficients with I = -12, -8, -4, ..., n-4. Then, by substituting the all the value in Eq. 11-13 into the proposed problem with consider the Eq. 14, the simply cubic B-spline approximation equation easily get:

$$\alpha_i.C_{i-12}+\beta_i.C_{i-8}+\gamma_i.C_{i-4} = r_i \quad (15)$$

Where:

$$\alpha_i = \frac{1}{4h^2}-\frac{p_i}{4h}+\frac{q_i}{6}$$
$$\beta_i = -\frac{2}{4h^2}+\frac{4q_i}{6}$$
$$\gamma_i = \frac{1}{4h^2}-\frac{p_i}{4h}+\frac{q_i}{6}$$

For I = 0, 4, 8, ..., n-12. After that the approximate solution is leads to construct the linear system that can be seen as:

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

Where:

$$A = \begin{bmatrix}\alpha_0 & \beta_0 & \gamma_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & \alpha_4 & \beta_4 & \gamma_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & \alpha_8 & \beta_8 & \gamma_8 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & \alpha_{12} & \beta_{12} & \gamma_{12} & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & \alpha_{16} & \beta_{16} & \gamma_{16} & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & \alpha_{20} & \beta_{20} & \gamma_{20} & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha_{24} & \beta_{24} & \gamma_{24} & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_{28} & \beta_{28} & \gamma_{28} & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_{32} & \beta_{32} & \gamma_{32}\end{bmatrix}$$

$$\underline{C} = [c_{-8}\ c_{-4}\ c_0\ c_4\ c_8\ c_{12}\ c_{16}\ c_{20}\ c_{24}]^T$$
$$\underline{R} = [r_0-\alpha\ r_4\ r_8\ r_{12}\ r_{16}\ r_{20}\ r_{24}\ r_{28}\ r_{32}-\beta]^T$$

Clearly, A represents as the coefficient matrix, $\underline{C}$ is an unknown vector and $\underline{R}$ is a known vector. In fact, the coefficients matrix, A of linear system (Eq. 16) should be considered positive definite condition $[a_{ii}] \geq \sum_{i\neq j}[a_{ij}]$ to get approximate solution based on studied by Young (1971).

RESULTS AND DISCUSSION

Formulation of quarter-sweep successive over relaxation: In this study, the formulation of the family of SOR iterative methods will be presented. The idea to choose the iterative methods as linear solver based on the study by Young (1971), Hackbusch (1995) and Saad (1996). They mentioned that the iterative methods are the best linear solver to solve the linear system which has large and sparse matrix. As we can see that the matrix A in linear system (Eq. 16) is large and sparse, the family of SOR iterative methods are chosen as linear solver to solve the linear system (Eq. 16).

The SOR iterative method was introduced by Young (1954, 1971, 1972, 1976). Basically, this method is the improvement of GS iterative method. The aim of SOR iterative method is to accelerate the convergence rate and reduce error approximation solution by adding the relaxation parameter, ω .The range of ω is $1 \leq \omega < 2$ and the optimum value of ω will lead to the numerical solution of SOR iterative method becomes more accurate.

As mentioned earlier in first paragraph in this study, the matrix A is large and spare, so, let define matrix A in the summation of three matrices form as follows:

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

Where:

D = A Diagonal matrix of matrix
A, L and U = Strictly Lower matrix and strictly Upper matrix, respectively

By taking Eq. 17 into Eq. 16, we can define the linear system as:

$$\left(L+D+U\right)\underline{C} = \underline{R} \tag{18}$$

By referring Eq. 18, the general scheme of QSSOR is written as (Youssef and Meligy 2014; Radzuan *et al*., 2017; Suardi *et al*., 2017a, b):

$$c_i^{(k+1)} = \left(1-\omega\right)c_i^{(k)} + \frac{\omega}{a_{ii}}\begin{pmatrix} r_i - \sum_{j=1}^{i-1} a_{ij}c_j^{(k+1)} \\ -\sum_{j=i+1}^{n} a_{ij}c_j^{(k)} \end{pmatrix} \quad (19$$

For i = 0, 4, 8, ..., n. Thus, the implementation of QSSOR iterative method are shows in algorithm 1.

Algorithm 1; QSSOR iterative method:

I. Set initial value $c^{(0)} = 0$
ii. Calculate the coefficient matrix and vector, $\underline{R}$
iii. For i = 0, 4, 8, ..., n, calculate

$$c_i^{(k+1)} = \left(1-\omega\right)c_i^{(k)} + \frac{\omega}{a_{ii}}\begin{pmatrix} r_i - \sum_{j=1}^{i-1} a_{ij}c_j^{(k+1)} \\ -\sum_{j=i+1}^{n} a_{ij}c_j^{(k)} \end{pmatrix}$$

iv. Check the convergence test, $|c_i^{(k+1)}-c_i^{(k)}| \leq \varepsilon = 10^{-10}$. If yes, go to step (v). Otherwise go back to step (iii)
v. Display numerical solution

Numerical examples: In order to observe the performance of the proposed iterative methods, we consider three numerical examples which are tested at several grid sizes and solve via. selected iterative methods such as FSGS, FSSOR, HSSOR and QSSOR iterative methods. The FSGS iterative method acts as control parameter. Then, three comparison parameters are taken during implementation of proposed methods like number of Iterations (Iter), execution Time in sec (Time) and maximum error (Error). The tolerance error is constant and set up as $\varepsilon = 10^{-10}$ for each different grid size. Example 1 (Cagler *et al*., 2006; Caglar and Caglar, 2009). Suppose the two point boundary value problem is as:

$$y''-y' = e^{(x-1)-1}, \ x \in \left[0, 1\right] \tag{20}$$

The exact solution for example 1 is known as. The numerical results for Eq. 20 can be observed in Table 1. Figure 2 and 3 are illustrated the results based on Table 1 in term of number of iterations and execution time, example 2 (Robertson, 1971). Consider two-point boundary value problem in form:

$$-y''-2y'+2y = e^{-2x}, \ x \in \left[0, 1\right] \tag{21}$$

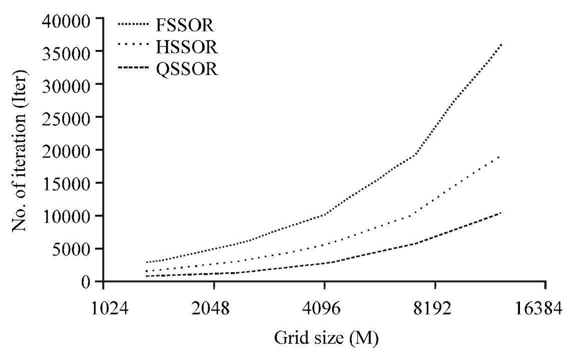

Fig. 2: The number of iteration for example 1

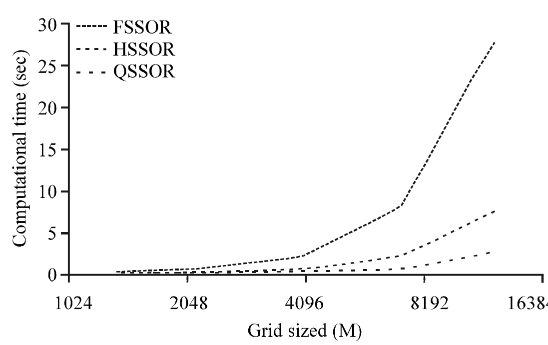

Fig. 3: The execution time for example 1

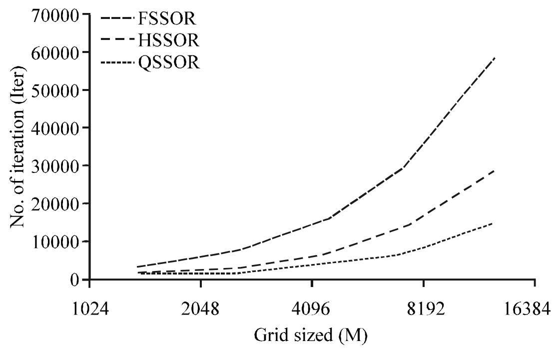

Fig. 4: The number of iteration for example 2

Table 1: Comparison of the number of iterations, execution time (sec) and the maximum absolute error on iterative methods, for example 1

M/Method	Iter	Time (sec)	Error
1024			
FSGS	1025490	109.60	1.03e-05
FSSOR	2946	0.57	3.05e-08
HSSOR	1526	0.33	1.16e-07
QSSOR	769	0.26	4.39e-07
2048			
FSGS	3527433	501.08	4.14e-05
FSSOR	5792	1.14	1.49e-08
HSSOR	2946	0.58	3.11e-08
QSSOR	1526	0.40	1.16e-07
4096			
FSGS	11811520	3214.48	1.66e-04
FSSOR	10245	2.78	9.42e-08
HSSOR	5792	1.10	1.49e-08
QSSOR	2946	0.53	3.11e-08
8192			
FSGS	38052999	15288.39	6.63e-04
FSSOR	19073	8.41	1.77e-07
HSSOR	10245	2.75	9.42e-08
QSSOR	5792	1.13	1.49e-08
16384			
FSGS	115439220	58347.49	2.65e-03
FSSOR	36021	27.47	2.90e-07
HSSOR	19073	7.73	1.77e-07
QSSOR	10245	2.88	9.42e-08

where the exact solution is defined as $y(x) = 1/2e^{-(1+\sqrt{3})}+1/2e^{-2x}$. The results are recorded in Table 2 for the Eq. 21. The number of iterations and execution time in Table 2 are illustrated in Fig. 4 and 5. Example 3 (Mohsen and El-Gamel, 2008). We consider one-dimensional two-point boundary value problem defines as follows:

$$-y''-4y = \cosh(1),\ x \in [0, 1] \quad (22)$$

The exact solution of this example is y(x) = cosh (2x-1)-cosh(1). Table 1 shows the numerical results are recorded for Eq. 22 at different grid sizes. Figure 6 and 7 are presented the graph of number of iterations and execution time for example 3.

The reduction percentages for all performance of iterative methods were obtained and can be shown in Table 4. From the results were carried out in Table 1-3.

Table 2: Comparison of the number of iterations, execution time (sec) and the maximum absolute error on iterative methods, for example 2

M/Methods	Iter	Time (sec)	Error
1024			
FSGS	886861	100.44	8.24e-06
FSSOR	3823	0.66	1.05e-07
HSSOR	1933	0.40	1.01e-06
QSSOR	977	0.26	1.50e-06
2048			
FSGS	3095807	482.89	3.26e-05
FSSOR	7553	1.66	4.97e-08
HSSOR	3823	0.75	2.62e-07
QSSOR	1933	0.37	3.79e-07
4096			
FSGS	10576347	3099.27	1.30e-04
FSSOR	14905	4.40	6.29e-08
HSSOR	7553	1.57	8.77e-08
QSSOR	3823	0.81	1.0e-07
8192			
FSGS	35077202	16192.67	5.21e-04
FSSOR	29377	14.44	1.23e-07
HSSOR	14905	4.56	7.21e-08
QSSOR	7553	1.56	4.97e-08
16384			
FSGS	111394765	56824.70	2.09e-03
FSSOR	57831	49.66	2.69e-07
HSSOR	29377	14.11	1.26e-07
QSSOR	14333	4.44	6.24e-09

Table 3: Comparison of the number of iterations, execution time (sec) and the maximum absolute error on iterative methods, for example 3

M/Method	Iter	Time (sec)	Error
1024			
FSGS	848604	96.58	7.44e-06
FSSOR	2613	0.54	1.35e-07
HSSOR	1354	0.31	4.90e-07
QSSOR	720	0.25	1.94e-06
2048			
FSGS	2975185	466.09	3.02e-05
FSSOR	5218	1.33	3.44e-08
HSSOR	2613	0.50	1.35e-07
QSSOR	1354	0.36	4.90e-07
4096			
FSGS	10223821	2999.24	1.21e-04
FSSOR	9886	3.55	2.66e-08
HSSOR	5218	1.19	3.44e-08
QSSOR	2613	0.59	1.35e-07
8192			
FSGS	34187618	15930.80	4.84e-04
FSSOR	17413	9.54	2.16e-07
HSSOR	9886	3.11	2.66e-08
QSSOR	5218	1.06	3.44e-08
16384			
FSGS	109919813	57115.58	1.94e-03
FSSOR	36776	33.21	3.63e-07
HSSOR	17413	8.34	2.16e-07
QSSOR	9202	2.79	8.34e-08

and Fig. 2-7, the combination between SOR iterative method and quarter-sweep approach need less number of iterations to solve two-point boundary value problems at all grid sizes are considered. Inline the results in term of execution time shows that the QSSOR iterative method is the fastest method to solve the proposed problem among the others iterative methods. Also, the reduction percentage of QSSOR iterative method is the higher

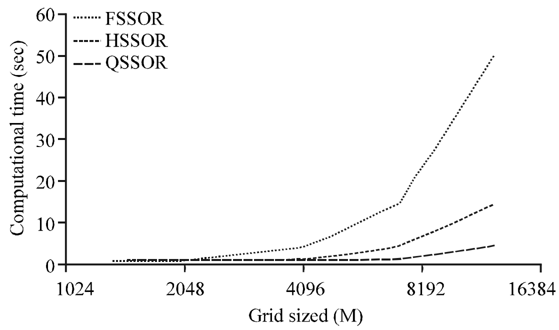

Fig. 5: The execution time for example 2

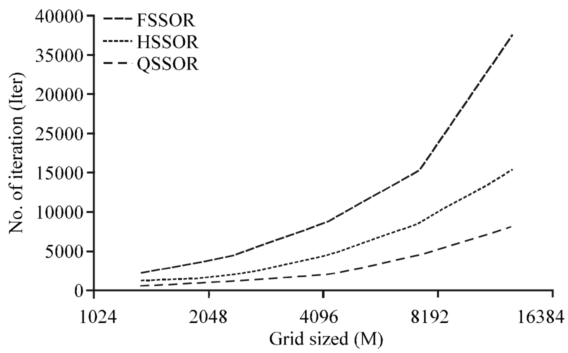

Fig. 6: The number of iteration for example 3

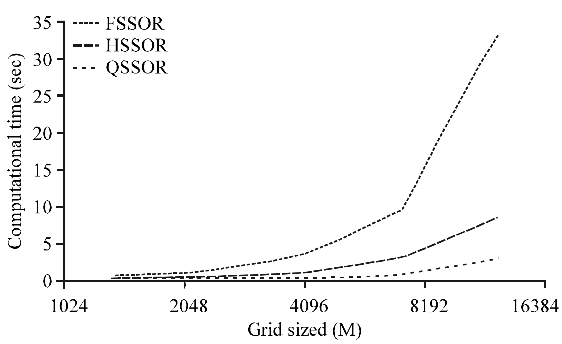

Fig. 7: The execution time for example 3

Table 4: Reduction percentage of the number of iterations and computational time for the FSKSOR, HSSOR and QSSOR compared with FSGS method

Examples	Iter (%)	Time (%)
Example 1		
FSSOR	99.71-99.99	99.48-99.95
HSSOR	99.85-99.99	99.70-99.99
QSSOR	99.93-99.99	99.97-99.99
Example 2		
FSSOR	99.57-99.95	99.34-99.91
HSSOR	99.78-99.97	99.60-99.98
QSSOR	99.89-99.99	99.74-99.99
Example 3		
FSSOR	99.69-99.97	99.44-99.94
HSSOR	99.84-99.98	88.68-99.99
QSSOR	99.92-99.99	99.74-99.99

reduction percentage for all problem are tested can be observed in Table 4. Clearly, QSSOR iterative method requires lesser number of iteration and execution time for solving two-point boundary value problems.

CONCLUSION

The study presented the cubic B-spline method in solving two-point boundary value problems. The combination between cubic B-spline approach and quarter-sweep concept are applied to discretize the proposed problem. Then, the numerical examples are tasted via. the family of SOR iterative methods are used as linear solver. Based on the numerical solutions, we concluded that QSSOR iterative method is superior in term of number of iterations and execution time than FSGS, FSSOR and HSSOR iterative methods.

RECOMMENDATIONS

Apart of this finding, the two-step iteration with B-spline approach such as AM (Ruggiero and Galligani, 1990), IADE (Sahimi *et al*., 1993) and QSAM (Sulaiman *et al*., 2009) can be consider for the future research.

ACKNOWLEDGEMENT

The researchers are grateful for the fund received from Universiti Malaysia Sabah upon publication of this study (GUG0135-1/2017).

REFERENCES

Aarao, J., B.H. Bradshaw-Hajek, S.J. Miklavcic and D.A. Ward, 2010. The Extended-domain-eigenfunction method for solving elliptic boundary value problems with annular domains. J. Phys. A. Math. Theor., 43: 185-202.

Abdullah, A.R., 1991. The four point Explicit Decoupled Group (EDG) method: A fast poisson solver. Int. J. Comput. Math., 38: 61-70.

Akhir, M.K.M., M.Y.H.M. Othman, J. Sulaiman, Z.A. Majid and M. Suleiman, 2011. Half-sweep modified successive over relaxation for solving Two-dimensional Helmholtz equations. Aust. J. Basic Appl. Sci., 5: 3033-3039.

Albasiny, E.L. and W.D. Hoskins, 1969. Cubic spline solutions to two-point boundary value problems. Comput. J., 12: 151-153.

Alibubin, M.U., A. Sunarto, M.K.M. Akhir and J. Sulaiman, 2016. Performance analysis of Half-sweep SOR iteration with rotated nonlocal arithmetic mean scheme for 2D nonlinear elliptic problems. Global J. Pure Appl. Math., 12: 3415-3424.

Botella, O. and K. Shariff, 2003. B-spline methods in fluid dynamics. Intl. J. Comput. Fluid Dyn., 17: 133-149.

Caglar, H., N. Caglar and K. Elfaituri, 2006. B-spline interpolation compared with finite difference, finite element and finite volume methods which applied to Two-point boundary value problems. Appl. Math. Comput., 175: 72-79.

Caglar, N. and H. Caglar, 2009. B-spline method for solving linear system of Second-order boundary value problems. Comput. Math. Appl., 57: 757-762.

Chang, J., Q. Yang and L. Zhao, 2011. Comparison of B-spline method and finite difference method to solve BVP of linear ODES. J. Comput., 6: 2149-2155.

Chew, J.V.L. and J. Sulaiman, 2016. Half-Sweep newton-gauss-seidelforimplicit finite difference solution of 1D nonlinear porous medium equations. Global J. Pure Appl. Math., 12: 2745-2752.

Choi, J.W., R.E. Curry and G.H. Elkaim, 2012. Minimizing the maximum curvature of quadratic Bezier curves with a tetragonal concave polygonal boundary constraint. Comput. Aided Des., 44: 311-319.

Dahalan, A.A., J. Sulaiman and M.S. Muthuvalu, 2014. Performance of HSAGE method with Seikkala derivative for 2-D fuzzy poisson equation. Applied Math. Sci., 8: 885-899.

Eng, J.H., A. Saudi and J. Sulaiman, 2017. HSAOR iteration for poisson image blending problem via rotated Five-point laplacian operater. Global J. Pure Appl. Math., 13: 5447-5459.

Gamel, E.M., 2007. Comparison of the solutions obtained by Adomian decomposition and Wavelet-galerkin methods of Boundary-value problems. Appl. Math. Comput., 186: 652-664.

Gardner, L.R.T. and G.A. Gardner, 1995. A two dimensional Bi-cubic B-spline finite element: Used in a study of MHD-duct flow. Comput. Methods Appl. Mech. Eng., 124: 365-375.

Hackbusch, W., 1995. Iterative Solution of Large Sparse Systems of Equations. Springer, New York, USA., ISBN:9783540940647, Pages: 429.

Jang, B., 2008. Two-point boundary value problems by the extended adomian decomposition method. J. Comput. Applied Math., 219: 253-262.

Lin, Y., J.A. Enszer and M.A. Stadtherr, 2008. Enclosing all solutions of Two-point boundary value problems for ODEs. Comput. Chem. Eng., 32: 1714-1725.

Mohsen, A. and M. El-Gamel, 2008. On the galerkin and collocation methods for two-point boundary value problems using sinc bases. Comput. Math. Applic., 56: 930-941.

Muthuvalu, M.S. and J. Sulaiman, 2011. Half-Sweep arithmetic mean method with composite trapezoidal scheme for solving linear fredholm integral equations. Applied Math. Comput., 217: 5442-5448.

Othman, M. and A.R. Abdullah, 2000. An efficient four points modified explicit group poisson solver. Int. J. Comput. Math., 76: 203-217.

Othman, M., A.R. Abdullah and D.J. Evans, 2014. A parallel four points modified explicit group algorithm on shared memory multiprocessors. Parallel Algorithms Appl., 19: 1-9.

Radzuan, N.Z.F.M., M.N. Suardi and J. Sulaiman, 2017. KSOR iterative method with quadrature scheme for solving system of Fredholm integral equations of second kind. J. Fundam. Appl. Sci., 9: 609-623.

Ramadan, M.A., I.F. Lashien and W.K. Zahra, 2007. Polynomial And nonpolynomial spline approaches to the numerical solution of second order boundary value problems. Applied Math. Comput., 184: 476-484.

Robertson, T.N., 1971. The linear Two-point Boundary-value problem on an infinite interval. Math. Comput., 25: 475-481.

Ruggiero, V. and E. Galligani, 1990. An iterative method for large sparse systems on a vector computer. Comput. Mathematics Appl., 20: 25-28.

Saad, Y., 1996. Iterative Methods for Sparse Linear Systems. PWS Publishing Company, Cleveland, New York, USA., ISBN:9780534947767, Pages: 447.

Sahimi, M.S., A. Ahmad and A.A. Bakar, 1993. The iterative alternating decomposition explicit method to solve the heat conduction equation. Int. J. Comput. Math., 47: 219-229.

Schoenberg, I.J., 1946. Contributions to the problem of approximation of equidistant data by analytic functions Part B: On the problem of osculatory interpolation: A second class of analytic approximation formulae. Q. Appl. Math., 4: 112-141.

Suardi, M.N., N.Z.F.M. Radzuan and J. Sulaiman, 2017a. Cubic B-spline solution for Two-point boundary value problem with AOR iterative method. J. Phys. Conf. Ser., 890: 1-6.

Suardi, M.N., N.Z.F.M. Radzuan and J. Sulaiman, 2017b . MKSOR iterative method with cubic B-spline approximation for solving Two-point boundary value problems. J. Fundam. Appl. Sci., 9: 594-608.

Sulaiman, J., M. Othman and M.K. Hasan, 2004. Quarter-sweep iterative alternating decomposition explicit algorithm applied to diffusion equations. Int. J. Comput. Math., 81: 1559-1565.

Sulaiman, J., M. Othman and M.K. Hasan, 2009. A new Quarter-Sweep Arithmetic Mean (QSAM) method to solve diffusion equations. Chamchuri J. Math., 1: 89-99.

Viswanadham, K.K. and S.R. Koneru, 1993. Finite element method for One-dimensional and Two-dimensional time dependent problems with B-splines. Comput. Methods Appl. Mech. Eng., 108: 201-222.

Wang, Y.M. and B.Y. Guo, 2008. Fourth-order compact finite difference method for Fourth-order nonlinear elliptic boundary value problems. J. Comput. Appl. Math., 221: 76-97.

Wu, J. and X. Zhang, 2011. Finite element method by using quartic Bsplines. Numer. Methods Partial Differ. Equations, 27: 818-828.

Young, D., 1954. Iterative methods for solving partial difference equations of elliptic type. Trans. Am. Math. Soc., 76: 92-111.

Young, D.M., 1971. Iterative Solution of Large Linear Systems. Academic Press, London.

Young, D.M., 1972. Second-degree iterative methods for the solution of large linear systems. J. Approximation Theory, 5: 137-148.

Young, D.M., 1976. Iterative Solution of Linear Systems Arising from Finite Element Techniques. In: The Mathematics of Finite Elements and Applications II, Whiteman, J.R. (Ed.). Academic Press, London, UK., pp: 439-464.

Youssef, I.K. and S.A. Meligy, 2014. Boundary value problem on triangular domains and MKSOR method. Appl. Comput. Math., 3: 90-99.