Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 12 (17): 1817-1824, 2012
ISSN 1812-5654 / DOI: 10.3923/jas.2012.1817.1824

Quarter-sweep Modified SOR Iterative Algorithm and Cubic Spline Basis for the Solution of Second Order Two-point Boundary Value Problems

N.I.M. Fauzi and J. Sulaiman
Mathematics with Economics Programme, School of Science and Technology,
Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia

Abstract: The aim of this study is to describe the formulation of Quarter-Sweep Modified Successive Over-Relaxation (QSMSOR) iterative method using cubic polynomial spline scheme for solving second order two-point linear boundary value problems. To solve the problems, a linear system will be constructed via discretization process by using cubic spline approximation equation. Then the generated linear system has been solved using the proposed QSMSOR iterative method to show the superiority over Full-Sweep Modified Successive Over-Relaxation (FSMSOR) and Half-Sweep Modified Successive Over-Relaxation (HSMSOR) methods. Computational results are provided to illustrate the effectiveness of the proposed method.

Key words: Two-point boundary value problems, quarter-sweep approach, modified successive over-relaxation iteration, cubic spline scheme

INTRODUCTION

Presently, seeking the approximate and/or exact solutions of two-point Boundary Value Problems (BVPs) play an essential role in science and engineering problems. For instance, modeling of chemical reactions and heat transfer can be governed by these problems. In addition to that, obtaining accurate and fast numerical solution of two-point BVPs is of great importance due to its wide application in scientific and engineering research. Therefore, many numerical methods intensively have been proposed to solve two-point boundary value problems such as finite difference, finite element and finite volume methods (Fang *et al*., 2002), extended ADM (EADM) (Jang, 2008), Differential Quadrature method (Sari, 2008), Precise Time Integration (PTI) method (Chen *et al*., 2006), shooting method (Ha, 2001), using Legendre polynomials and function approximation (Hoseini *et al*., 2009), Galerkin and Collocation methods (Mohsen and Gamel, 2008; Tasci, 2003) and using interpolation (Sophianopoulos and Asteris, 2004). In this study, however, discretization schemes based on cubic spline scheme was used to discretize the problems. Precisely, the cubic spline scheme will be taken into account in constructing a cubic spline approximation equation towards two-point linear boundary value problems.

Actually based on previous studies of spline solutions, the development and analysis of the methods towards two-point boundary value problems have also been discussed by many researchers (Albasiny and Hoskins, 1969; Aziz and Khan, 2002; Ramadan *et al*., 2007; Rashidinia *et al*., 2008). These spline schemes are used to discretize two-point boundary value problems and then derive their corresponding spline approximation equations. Then each of these approximation equations yields a large and sparse linear system. Since the attributes of linear systems are large and sparse, iterative methods are the natural options for efficient solutions. As a matter of fact, various iterative methods also have been studied to yield fast numerical solution of linear systems (Young, 1971; Hackbusch, 1995; Saad, 2003; Jin *et al*., 2010). Apart from those iterative methods, the finding of the half-sweep iteration concept has been introduced by Abdullah (1991) via., Explicit Decoupled Group (EDG) iterative method in solving two-dimensional Poisson equations. The essential characteristic of this concept is that the half-sweep iterative method includes the reduction technique in order to reduce the computational complexity of linear systems generated from corresponding approximation equations. Following to this concept, further investigations have been extensively conducted by Yousif and Evans (1995), Akhir *et al*. (2011), Aruchunan and Sulaiman (2011) and Hasan *et al*. (2011) for demonstrating the capability of the half-sweep iteration. Beside these one-stage iteration methods, combinations between half-sweep iteration concept with

Corresponding Author: N.I.M. Fauzi, Mathematics with Economics Programme, School of Science and Technology, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia

two-stage iterative methods namely HSIADE (Sulaiman *et al.*, 2004), HSAM (Muthuvalu and Sulaiman, 2011) and HSGM (Muthuvalu and Sulaiman, 2012) have also been constructed and implemented for solving linear systems. They pointed out that these proposed two-stage iterative methods are one of most efficient iterative methods in solving any system of linear equations. Due to the low computational complexity, again, this concept has been used to develop the formulation of various multigrid methods (Othman *et al.*, 2000; Sulaiman *et al.*, 2008). In addition to that, Hasan *et al.* (2005, 2007) have established a family of FDTD methods using this concept in solving wave propagation problems. They pointed out that the proposed FDTD method is more superior compared to the standard FDTD method. For robotic problems, Saudi and Sulaiman (2009, 2010) have used this concept to solve the robotic path planning.

Differently from the half-sweep iteration approach, Othman and Abdullah (2000) have expanded this approach to initiate the Modified Explicit Group (MEG) method based on the quarter-sweep approach. It is proved that this method is one of most efficient block iterative methods in solving any linear systems as compared with EG and EDG iterative methods. Also many studies have been carried out to demonstrate the capability of the quarter-sweep iteration (Jha and Srivastava, 2008; Sulaiman *et al.*, 2005, 2009, 2010; Muthuvalu and Sulaiman, 2010).

Due to the advantage of quarter-sweep approach, the main purpose of this paper is to examine the efficiency of Modified Successive Over-Relaxation (MSOR) iterative methods namely FSMSOR, HSMSOR and QSMSOR for solving two-point boundary value problems by using spline approximation equation based on cubic spline scheme.

Consider two-point linear boundary value problem be defined in general form as:

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

subject to the boundary conditions:

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

where l(x), f(x) and g(x) are continuous on the interval [a, b], through A_i, i = 1,2 are finite real constants and l(x), f(x) and g(x) are known functions.

Based on Fig. 1, we use the finite grid network to facilitate us in formulating the full-, half-and quarter-sweep cubic spline approximation equations. Later, the implementations of full-, half-and quarter-sweep iterative methods will be executed to solve the linear system generated by using the corresponding spline approximation equation. However, we just consider to obtain approximate values onto node points only until convergence test will be figured out. Meanwhile, the approximation solutions for the remaining points can be computed by using direct method (Abdullah, 1991; Ibrahim and Abdullah, 1995).

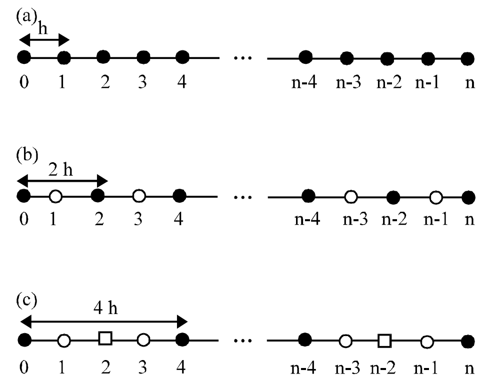

Fig. 1(a-c): Distribution of uniform solid node points for (a) Full-, (b) Half- and (c) Quarter-sweep cases

DERIVATION OF QUARTER-SWEEP CUBIC SPLINE APPROXIMATION EQUATION

By using full-, half- and quarter-sweep cubic spline approximation equations for solving problem (1), a finite set of grid points x_i, i = 1,2,…,n-1, n is established by partitioning the interval [a, b] into (n+1) uniformly subinterval:

$$x_i = a + ih, \quad x_0 = a, \quad x_n = b, \quad h = \frac{(b-a)}{n+1}$$

Let y(x) be the exact solution of problem (1) and S_i be an approximation to $y_i = y(x_i)$ determine by the segments of $Q_i(x)$ are passing through to the points (x_i, S_i) and (x_{i+1}, S_{i+1}). The spline approximation in general form can be expressed as:

$$S_k(x) = \sum_{m=0}^{n} C_m (x - x_k)^m \tag{2}$$

for n = 1, 2, …, n where C_m are the coefficient which must be determined in each interval, while suppose that n refer to the order of spline. Then let the cubic polynomial spline from Eq. 2 be defined as $Q_i(x)$ which is denoted in general form as:

$$Q(x) = a_i(x-x_i)^3+b_i(x-x_i)^2+c_i(x-x_i)+d_i \quad (3)$$

for i = 0p, 1p, 2p, ..., n-p, n with a_i, b_i and c_i are constant coefficients. The value of p which equals to 1, 2 and 4, represents for the full-, half- and quarter-sweep cases, respectively. In order to get the expression of three coefficients, a_i, b_i and c_i, the cubic spline segments Q_i in terms of S_{i+p} and M_{i+p} can be manipulated to derive by the following conditions:

$$\left.\begin{array}{c} Q_i(x_{i+p}) = S_{i+p} \\ Q_i(x_i) = S_{i-p} \\ Q_i''(x_{i+p}) = M_{i+p} \end{array}\right\} \quad (4)$$

Then the following expressions can be obtained by the straightforward substitution:

$$\left.\begin{array}{c} a_i = \dfrac{M_i - M_{i-p}}{6hp} \\ b_i = \dfrac{M_{i-p}}{2} \\ c_i = \left(\dfrac{S_i - S_{i-p}}{ph}\right) - \dfrac{ph}{6}\left(M_i + 2M_{i+p}\right) \\ d_i = S_{i-p} \end{array}\right\} \quad (5)$$

with i = 0p, 1p, 2p, ..., n-p, n. Now by mean of the expressions in Eq. 5 and the continuity conditions:

$$Q_{i-p}^{(k)}(x_i) = Q_i^{(k)}(x_i)$$

where k = 1, 2, we get the following relations for i = 0p, 1p, 2p, ..., n-p:

$$S_{i-p} - 2S_i + S_{i+p} = \frac{(ph)^2}{6}\left(M_{i-p} + 4M_i + M_{i+p}\right) \quad (6)$$

From Eq. 1, we substitute $M_i = y''(x_i)$ and obtain the following expression:

$$M_i = -l_i y_i' - f_i y_i + g_i \quad (7)$$

where $l_i = l(x_i)$, $f_i = f(x_i)$ and $g_i = g(x_i)$. According to Eq. 6, we can write the expression in Eq. 7 at the grid points x_k, k = i-p, i, i+p as follows:

$$\left.\begin{array}{c} M_{i-p} = -l_{i-p}y_{i-p}' - f_{i-p}y_{i-p} + g_{i-p} \\ M_i = -l_i y_i' - f_i y_i + g_i \\ M_{i+p} = -l_{i+p}y_{i+p}' - f_{i+p}y_{i+p} + g_{i+p} \end{array}\right\} \quad (8)$$

To approximate the first order derivative of y with respect to x in Eq. 8, we consider the following second order finite difference approximation equations:

$$\left.\begin{array}{c} y_{i+p}' \cong \dfrac{3y_{i+p} - 4y_i + y_{i-p}}{2ph} \\ y_i' \cong \dfrac{y_{i+p} - y_{i-p}}{2ph} \\ y_{i-p}' \cong \dfrac{-y_{i+p} + 4y_i - 3y_{i-p}}{2ph} \end{array}\right\} \quad (9)$$

Substituting Eq. 9 into 8, we have the following expression:

$$\left.\begin{array}{c} M_{i-p} = -l_{i-p}\left(\dfrac{-y_{i+p} + 4y_i - 3y_{i-p}}{2ph}\right) \\ - f_{i-p}y_{i-p} + g_{i-p}, \\ M_i = -l_i\left(\dfrac{y_{i+p} - y_{i-p}}{2ph}\right) - f_i y_i + g_i \\ M_{i+p} = -l_{i+p}\left(\dfrac{3y_{i+p} - 4y_i + y_{i-p}}{2ph}\right) \\ - f_{i+p}y_{i+p} + g_{i+p} \end{array}\right\} \quad (10)$$

Again substituting Eq. 10 into 6, we obtain the following:

$$\left(\left(\frac{n+1}{p}\right) - 1\right)$$

cubic spline approximation equations in case of full-, half- and quarter-sweep for i = 1p, 2p, 3p, ..., n-p can be stated as:

$$\alpha_{ip}S_{i-p} + \beta_{ip}S_i + \sigma_{ip}S_{i+p} = F_{ip} \quad (11)$$

Where:

$$\alpha_{ip} = 6 - 3\frac{(ph)}{2}l_{i-p} + \frac{(ph)}{2}l_{i+p} - 2(ph)l_{i+p} + (ph)^2 f_{i-p}$$
$$\beta_{ip} = 4(ph)^2 f_i - 2(ph)l_{i+p} + 2(ph)l_{i-p} - 12$$
$$\sigma_{ip} = 6 - \frac{(ph)}{2}\left(l_{i-p} - 2l_i - 3l_{i+p}\right) + (ph)^2 f_{i+p}$$
$$F_{ip} = (ph)^2\left(g_{i-p} + 4g_i + g_{i+p}\right)$$

Let us rewrite the linear system (11) in matrix form as follows:

$$AS = F \quad (12)$$

Where:

$$A = \begin{bmatrix} \beta_{1p} & \sigma_{1p} & & & & \\ \alpha_{2p} & \beta_{2p} & \sigma_{2p} & & & \\ & \alpha_{3p} & \beta_{3p} & \sigma_{3p} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \alpha_{n-2p} & \beta_{n-2p} & \sigma_{n-2p} \\ & & & & \alpha_{n-p} & \beta_{n-p} \end{bmatrix}_{\left(\left(\frac{n+1}{p}\right)-1\right)\times\left(\left(\frac{n+1}{p}\right)-1\right)}$$

$$\underset{\sim}{S} = [S_{1P}, \quad S_{2P}, \quad S_{3P}, \quad \cdots, \quad S_{n-P}]^T$$

$$\underset{\sim}{F} = \left[F_{1P} - \alpha_{1p}, \quad F_{2P}, \quad F_{3P}, \quad \cdots, \quad F_{n-P} - \beta_{n-p}\right]^T$$

FORMULATION OF MODIFIED SUCCESSIVE OVER-RELAXATION ITERATIVE ALGORITHMS

As aforementioned in the second section, the coefficient matrix, A of linear systems in Eq. 12 is sparse and large. Consequently, iterative methods are proposed being as the natural options for efficient solutions of sparse linear system. In this section, we show on how to construct FSMSOR, HSMSOR and QSMSOR iterative methods being applied to solve linear systems (12).

To derive the formulation for FSMSOR, HSMSOR and QSMSOR iterative methods, let the coefficient matrix, A in Eq. 12 be decomposed as:

$$A = D+L+U \tag{13}$$

where, L, D and U are lower triangular, diagonal and upper triangular matrices, respectively. By using the decomposition in Eq. 13, therefore, the general scheme for SOR method can be stated as (Young, 1971):

$$\underset{\sim}{S}^{(k+1)} = (1-\omega)\underset{\sim}{S}^{(k)} + \omega(D+L)^{-1}\left(-U\underset{\sim}{S}^{(k)} + \underset{\sim}{F}\right) \tag{14}$$

where, ω and $\underset{\sim}{S}^{(k)}$ represent as a relaxation factor and an unknown vector at the kth iteration respectively. In fact, the formulation of the QSMSOR iterative scheme is the same to the SOR iteration scheme by considering the implementation of red-black ordering strategy through the use of two different weighted parameters, ω and ω'. For example, parameter ω' was performed on the order of red and the next parameter ω was applied to black rule. Based on Eq. 14, QSMSOR iterative scheme can be expressed as (Young, 1971; Kincaid and Young, 1972):

$$\left.\begin{aligned} \underset{\sim}{S}^{(k+1)} &= (1-\omega)\underset{\sim}{S}^{(k)} + \omega(D+L)^{-1}\left(-U\underset{\sim}{S}^{(k)} + \underset{\sim}{F}\right) \\ \underset{\sim}{S}^{(k+1)} &= (1-\omega')\underset{\sim}{S}^{(k)} + \omega'(D+L)^{-1}\left(-U\underset{\sim}{S}^{(k)} + \underset{\sim}{F}\right) \end{aligned}\right\} \tag{15}$$

As taking $\omega = \omega'$, the schematic of Eq. 15 is known as SOR iterative method with the Red-Black ordering strategy. Therefore by determining values of matrices D, L and U as stated in Eq. 13, the general algorithm of FSMSOR, HSMSOR and QSMSOR iterative methods would be generally described in Algorithm 1:

Algorithm 1: FSMSOR, HSMSOR and QSMSOR schemes

Initialize $S_i^{(0)} \leftarrow 0$, $l_i \leftarrow 0$, $\varepsilon \leftarrow 10^{-10}$

For i = 1p, 2p, 3p, …, n-p, calculate:

$$\alpha_{ip} = 6 - 3\frac{(ph)}{2}l_{i-p} + \frac{(ph)}{2}l_{i+p} - 2(ph)l_{i+p} + (ph)^2 f_{i-p},$$

$$\beta_{ip} = 4(ph)^2 f_i - 2(ph)l_{i+p} + 2(ph)l_{i-p} - 12,$$

$$\sigma_{ip} = 6 - \frac{(ph)}{2}\left(l_{i-p} - 2l_i - 3l_{i+p}\right) + (ph)^2 f_{i+p},$$

$$F_{ip} = (ph)^2\left(g_{i-p} + 4g_i + g_{i+p}\right),$$

For i 1p, 2p, …, n-p, calculate:

$$S_i^{(k+1)} \leftarrow (1-\omega)S_i^{(k)} + \frac{\omega}{\beta_{ip}}\left(F_i - \alpha_{ip}S_{i-p}^{(k+1)} - \sigma_i S_{i+p}^{(k)}\right)$$

For i = 2p, 4p, …, n-2p, calculate:

$$S_i^{(k+1)} \leftarrow (1-\omega')S_i^{(k)} + \frac{\omega 1}{\beta_{ip}}\left(F_i - \alpha_{ip}S_{i-p}^{(k+1)} - \sigma_i S_{i+p}^{(k)}\right)$$

Check the convergence test, $|S_i^{(k+1)} - S_i^{(k)}| \le \varepsilon$. If yes, go to step (vi). Otherwise go back to step (iii)

Display approximate solutions

NUMERICAL EXPERIMENTS

Here, we have applied QSMSOR methods on the following three problems, whose exact solutions are known to us and have compared the results with the FSMSOR and HSMSOR methods in order to demonstrate the effectiveness of QSMSOR method based on cubic spline approach. For the sake of comparison, there are three parameters considered in numerical comparison namely number of iterations, execution time and maximum absolute error. Throughout the numerical simulations, the initial value of vector $\underset{\sim}{S}^{(0)} = 0$ is used in all proposed iterative methods and the iterations were terminated stopped when the absolute error tolerance, $\varepsilon = 10^{-10}$ was achieved at several different values of n.

Example 1 (Mohsen and Gamel, 2008):

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

The exact solution for Eq. 16 is given by:

$$y(x) = \cos h(2x-1) - \cosh 1$$

Example 2:

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

Table 1: Comparison of No. of iterations K, the execution time (seconds) and maximum absolute errors for the iterative methods using example 1

	FSMSOR			HSMSOR			QSMSOR		
M	K	Time	Max. error	K	Time	Max. error	K	Time	Max. error
512	1151	0.40	3.7506e-07 ω = 1.986607 ω' = 1.986606	619	0.25	1.9102e-06 ω = 1.9731160 ω' = 1.9730980	313	0.15	2.3402e-05 ω = 1.9468600 ω' = 1.9467950
1024	2140	0.53	2.6125e-07 ω = 1.9929013 ω' = 1.9929010	1151	0.31	3.7506e-07 ω = 1.98660700 ω' = 1.9866055	619	0.22	5.8685e-06 ω = 1.97311600 ω' = 1.9731000
2048	4262	0.96	3.5322e-07 ω = 1.9966811 ω' = 1.9966808	2139	0.51	2.6246e-07 ω = 1.99290146 ω' = 1.9929013	1151	0.34	1.4696e-07 ω = 1.98660690 ω' = 1.9866055
4096	9062	2.40	6.9302e-07 ω = 1.99850760 ω' = 1.99850754	4556	1.11	6.5438e-07 ω = 1.99701170 ω' = 1.9970111	2140	0.64	3.6743e-07 ω = 1.99290138 ω' = 1.9929010
8192	14226	17.97	2.1216e-06 ω = 1.99118364 ω' = 1.99113627	9062	3.26	6.9302e-07 ω = 1.99850760 ω' = 1.9985075	4820	2.25	2.3799e-07 ω = 1.99693529 ω' = 1.9969350

Table 2: Comparison of number of iterations K, the execution time (seconds) and maximum absolute errors for the iterative methods using example 2

	FSMSOR			HSMSOR			QSMSOR		
M	K	Time	Max. error	K	Time	Max. error	K	Time	Max. error
512	1716	0.55	2.6479e-06 ω = 1.988040 ω'= 1.988005	847	0.43	1.0607e-05 ω = 1.97609700 ω'= 1.87609070	437	0.26	7.6748e-05 ω = 1.95398000 ω'= 1.9538850
1024	3467	0.89	6.6156e-07 ω = 1.993972 ω'= 1.993944	1715	0.47	2.6478e-06 ω = 1.98805000 ω'= 1.98800400	847	0.42	1.9185e-05 ω = 1.97609700 ω'= 1.9760907
2048	6936	2.43	1.6998e-07 ω = 1.996963 ω'= 1.996946	3467	0.75	6.6199e-07 ω = 1.99397600 ω'= 1.9939200	1716	0.56	4.7923e-06 ω = 1.9880500 ω'= 1.9880000
4096	13831	4.10	5.6481e-08 ω = 1.988468 ω'= 1.998464	6935	1.53	1.6917e-07 ω = 1.99696100 ω'= 1.99695230	3467	0.88	1.1981e-06 ω = 1.99397600 ω'= 1.9939320
8192	27530	19.84	5.2709e-08 ω = 1.988468 ω'= 1.999224	13831	6.64	5.7577e-08 ω = 1.98846900 ω'= 1.99846200	6935	2.78	3.0319e-07 ω = 1.99696100 ω'= 1.9969523

The exact solution for Eq. 17 is given by:

$$y(x) = \cosh(2x\text{-}1)\text{-}\cosh\text{-}1$$

Example 3 (Aziz and Khan, 2002):

$$\varepsilon y'' = y+\cos^2(\pi x)+2\varepsilon\pi^2\cos(2\pi x) \qquad (18)$$

And the exact solution for Eq. 18 is given by:

$$y(x) = \exp(-(1-x)/\sqrt{\varepsilon}) + \exp(-x/\sqrt{\varepsilon})/[1+\exp(-1/\sqrt{\varepsilon})] - \cos^2(\pi x)$$

According to above three examples, the results of numerical experiments obtained from implementation of the FSMSOR, HSMSOR and QSMSOR iterative methods have been tabulated in Table 1-3, whereas, Table 4 indicates depreciation percentage of number of iterations and execution time.

COMPUTATIONAL COMPLEXITY ANALYSIS

To compare the computational complexity of three proposed MSOR iterative methods, we need to calculate an estimation amount of the computational works needed for implementation of FSMSOR, HSMSOR and QSMSOR methods. The computational work is evaluated by analysing arithmetic operation achieved per iteration for each of proposed MSOR iterative methods. Based on Algorithm 1, it can be seen that there are:

$$4\left(\left(\frac{n+1}{p}\right)-1\right)$$

addition/subtraction (ADD/SUB) and:

$$5\left(\left(\frac{n+1}{p}\right)-1\right)$$

Multiplication/division (MUL/DIV) operations in computing a value for each node point in the solution of

Table 3: Comparison of number of iterations K, the execution time (seconds) and maximum absolute errors for the iterative methods using example 3

	FSMSOR			HSMSOR			QSMSOR		
M	K	Time	Max. error	K	Time	Max. error	K	Time	Max. error
512	177	0.15	5.8344e-05 ω = 1.888000 ω′ = 1.888000	91	0.07	2.3362e-04 ω = 1.788400 ω′ = 1.786800	47	0.02	3.6164e-03 ω = 1.62060 ω′ = 4.61800
1024	365	0.31	1.4582e-05 ω = 1.945550 ω′ = 1.945230	173	0.15	5.8344e-05 ω = 1.884390 ω′ = 1.884363	91	0.05	9.3603e-04 ω = 1.78840 ω′ = 1.78673
2048	627	0.34	3.6454e-06 ω = 1.969841 ω′ = 1.969837	342	0.25	1.4582e-05 ω = 1.942500 ω′ = 1.942480	176	0.19	2.3841e-04 ω = 1.88794 ω′ = 1.88786
4096	1191	0.66	9.1197e-07 ω = 1.984841 ω′ = 1.984838	665	0.43	3.6450e-06 ω = 1.970950 ω′ = 9.1197e-07	341	0.34	6.0181e-05 ω = 1.94241 ω′ = 1.94247
8192	2323	1.81	2.2939e-07 ω = 1.992398 ω′ = 1.992396	1191	0.85	9.1197e-07 ω = 1.984841 ω′ = 1.984838	665	0.52	1.5120e-05ω ω = 1.97095 ω′ = 1.97092

Table 4: Depreciation percentage of number of iterations and execution time for HSMSOR and QSMSOR iterative methods compared to FSMSOR using three examples

	HSMSOR (%)		QSMSOR (%)	
Examples	No. of iterations	Execution time	No. of iterations	Execution time
1	36.30-49.81	37.50-81.86	66.11-72.99	56.60-87.47
2	49.77-50.64	21.82-69.14	74.53-75.57	52.72-85.99
3	44.17-52.60	26.47-53.33	71.28-75.06	44.11-86.67

Table 5: Total number of arithmetic operations per iteration for FSMSOR, HSMSOR and QSMSOR methods

	Arithmetic operation	
Methods	ADD/SUB	MUL/DIV
FSMSOR	4n	5n
HSMSOR	$4\left(\left(\frac{n+1}{2}\right)-1\right)$	$5\left(\left(\frac{n+1}{2}\right)-1\right)$
QSMSOR	$4\left(\left(\frac{n+1}{4}\right)-1\right)$	$5\left(\left(\frac{n+1}{4}\right)-1\right)$

cubic spline approximation equation. Based on the order of coefficient matrix, A, the total number of arithmetic operation per iteration for the FSMSOR, HSMSOR and QSMSOR iterative methods in solving Eq. 1 has been recorded in Table 5. Clearly it shows that the computational complexity of HSMSOR and QSMSOR methods is lesser than FSMSOR method.

CONCLUSION

The efficiency of the family of MSOR iterative methods namely FSMSOR, HSMSOR and QSMSOR with the corresponding cubic spline approximation equations was employed successfully for solving two-point linear boundary value problems. Clearly it can be observed that the general cubic spline approximation equation in Eq. 11 has been used to form linear system which has been solved by the proposed MSOR iterative methods. Through numerical experiments results from Table 1-4, it clearly shows that QSMSOR method managed to converge faster than FSMSOR and HSMSOR methods. This is because of the QSMSOR method has less computational complexity, see in Table 5.

REFERENCES

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. Othman, J. Sulaiman, Z.A. Majid and M. Suleiman, 2011. Half-sweep modified successive overrelaxation for solving iterative method two-dimensional Helmholtz equations. Aust. J. Bas. Applied Sci., 5: 3033-3039.

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

Aruchunan, E. and J. Sulaiman, 2011. Half-sweep conjugate gradient method for solving first order linear fredholm integro-differential equations. Aust. J. Bas. Applied Sci., 5: 38-43.

Aziz, T. and A. Khan, 2002. A spline method for second-order singularly perturbed boundary-value problems. J. Comput. Applied Math., 147: 445-452.

Chen, B., L. Tong and Y. Gu, 2006. Precise time integration for linear two-point boundary value problems. Applied Math. Comput., 175: 182-211.

Fang, Q., T. Tsuchiya and T. Yamamoto, 2002. Finite difference, finite element and finite volume methods applied to two-point boundary value problems. J. Comput. Applied Math., 139: 9-19.

Ha, S.N., 2001. A nonlinear shooting method for two-point boundary value problems. Int. J. Comput. Math. Appl., 42: 1411-1420.

Hackbusch, W., 1995. Iterative Solution of Large Sparse Systems of Equations. Springer-Verlag, New York.

Hasan, M.K., J. Sulaiman, S.A.A. Karim and M. Othman, 2011. Development of some numerical methods applying complexity reduction approach for solving scientific problem. J. Applied Sci., 11: 1255-1260.

Hasan, M.K., M. Othman, Z. Abbas, J. Sulaiman and F. Ahmad, 2007. Parallel solution of high speed low order FDTD on 2d free space wave propagation. Comput. Sci. Appl., 4706: 13-24.

Hasan, M.K., M. Othman, Z. Abbas, J. Sulaiman and R. Johari, 2005. The HSLO (3)-FDTD with direct-domain and temporary-domain approaches on infinite space wave propagation. Proceedings of the 13th IEEE International Conference on Networks and 7th IEEE Malaysia International Conference on Communication, November 16-18, 2005, IEEE Computer Society, Kuala Lumpur, Malaysia, pp: 1002-1007.

Hoseini, S.M., M. Hoseini and H. Orooji, 2009. Numerical solution of two-point boundary value problem in linear differential equation. Applied Math. Sci., 3: 1493-1499.

Ibrahim, A. and A.R. Abdullah, 1995. Solving The two-dimensional diffusion equation by the four point Explicit Decoupled Group (EDG) iterative method. Int. J. Comput. Mathe., 58: 253-256.

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

Jha, N. and P.R. Srivastava, 2008. Application of QSAGE solver for the computation of state variable models in software test process. Int. J. Mathe. Model. Simul. Appl., 1: 305-312.

Jin, Y., G. Jin and T. Ma, 2010. New parallel three-level iterative method for diffusion equation. Inform. Technol. J., 9: 1184-1189.

Kincaid, D.R. and D.M. Young, 1972. The modified successive over relaxation method with fixed parameters. Mathe. Comput., 26: 705-717.

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

Muthuvalu, M.S. and J. Sulaiman, 2010. Quarter-sweep iteration for first kind linear Fredholm integral equations. Int. J. Open Prob. Compt. Math., 3: 357-367.

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.

Muthuvalu, M.S. and J. Sulaiman, 2012. Half-Sweep geometric mean iterative method for the repeated simpson solution of second kind linear fredholm integral equations. Proyecciones J. Math., 31: 65-79.

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

Othman, M., J. Sulaiman and A.R. Abdullah, 2000. A parallel halfsweep multigrid algorithm on the shared memory multiprocessors. Malaysian J. Comput. Sci., 13: 1-6.

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.

Rashidinia, J., R. Mohammadi and R. Jalilian, 2008. Cubic spline method for two-point boundary value problems. IUST Int. J. Eng. Sci., 19: 39-43.

Saad. Y., 2003. Iterative Methods for Sparse Linear Systems. 2nd Edn., Society for Industrial and Applied Mathematics, Philadelpha.

Sari, M., 2008. Differential quadrature method for singularly perturbed two-point boundary value problems. J. Applied Sci., 8: 1091-1096.

Saudi, A. and J. Sulaiman, 2009. Path planning for mobile robot with Half-Sweep Successive Over-Relaxation (HSSOR) iterative method. Proceedings of the Symposium on Progress in Information and Communication Technology, December 7-8, 2009, Kuala Lumpur, Malaysia, pp: 57-62.

Saudi, A. and J. Sulaiman, 2010. Red-black strategy for mobile robot path planning. Proc. Int. Multi Conf. Eng. Comput. Sci., 3: 2215-2220.

Sophianopoulos, D.S. and P.G. Asteris, 2004. Interpolation based numerical procedure for solving two-point nonlinear boundary value problems. Int. J. Nonlinear Sci. Numer. Simul., 5: 67-78.

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

Sulaiman, J., M.K. Hasan and M. Othman, 2005. The half-sweep iterative alternating decomposition explicit (HSIADE) method for diffusion equations. Comput. Inform. Sci., 3314: 57-63.

Sulaiman, J., M. Othman and M.K. Hasan, 2008. Half-Sweep Algebraic Multigrid (HSAMG) Method Applied to Diffusion Equations. In: Modeling, Simulation and Optimization of Complex Processes, Bock, H.G., H.X. Phu, R. Rannacher and J.P. Schloder (Eds.). Springer, Germany, ISBN-13: 9783540794080, pp: 547-556.

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.

Sulaiman, J., M.K. Hasan, M. Othman and S.A.A. Karimd, 2010. MEGSOR iterative method for the triangle element solution of 2D Poisson equations. Procedia Comput. Sci., 1: 377-385.

Tasci, F., 2003. On the approximative solution of boundary value problems by collocation. J. Applied Sci., 3: 210-215.

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

Yousif, W.S. and D.J. Evans, 1995. Explicit de-coupled group iterative methods and their parallel implementations. Paral. Algorithms Appli., 7: 53-71.