International Journal of Pure and Applied Mathematics
Volume 80 No. 5 2012, 751-764
ISSN: 1311-8080 (printed version)
url: http://www.ijpam.eu

ijpam.eu

AN EFFICIENT QUARTER-SWEEP MODIFIED SOR ITERATIVE METHOD FOR SOLVING HELMHOLTZ EQUATION

Mohd Kamalrulzaman Md Akhir[1 §], Mohamed Othman[2], Zanariah Abdul Majid[3], Mohamed Suleiman[4], Jumat Sulaiman[5]

[1,2,3,4]Institute For Mathematical Research
Universiti Putra Malaysia
43400, UPM Serdang Selangor, MALAYSIA

[5]Schools of Science and Technology
Universiti Malaysia Sabah
Locked Bag 2073, 88999 Kota Kinabalu Sabah, MALAYSIA

Abstract: The aim of this article is to describe the formulation of quarter-sweep Modified Successive Over-relaxation (QSMSOR) method using the finite difference approach for solving two-dimensional Helmholtz equation. The concept of QSMSOR method is inspired via the combination between quarter-sweep iterative and modified successive over-relaxation (MSOR) method. In addition, the formulation and the implementation of the QSMSOR method are also presented. Some illustrative examples are given to benchmark the effectiveness of the proposed method.

Key Words: Helmholtz equation, full-, half-and-quarter-sweep iterations, modified successive over-relaxation (MSOR) method, finite difference method

Received: September 25, 2012

§Correspondence author

1. Introduction

Over several past decades, many problems in engineering and science involve partial differential equation On the other hand, the applications of partial differential equation are encountered in many fields such as time harmonic acoustic and electromagnetic fields, optical waveguide, acoustic wave scattering noise reduction in silencer, water wave propagation and radar scattering problems (see Nabavi et al [16]). However, it is still very difficult to gain any solution in solving these problems, either analytically or numerically. As a matter of fact, mesh based methods such as finite element, finite volume and finite differences have been used widely to obtain the numerical solution which can be used to discretize and construct approximation equations for approximating the proposed problems. Next, these approximation equations will be used to generate the corresponding systems of linear algebraic equations. Due to the large scale of linear systems, many studies on various iterative methods have been proposed to speed up the convergence rate in solving any system of linear equations. Thus, Young [5-6], Belytschko et al [21] and Saad [24] had elaborated and discussed the concept of various iterative methods.

Recently, Akhir et al [12] proposed a Half-Sweep Modified Successive Over-Relaxation (HSMSOR) method by combining the concept of the half-sweep iterations and the modified SOR method to solve two-dimensional Helmholtz equation, mainly on optical waveguide problem. Actually, the half-sweep iterative method was initiated by Abdullah [1] via Explicit Decoupled Group (EDG) method for solving two-dimensional Poisson equation. Following to that, related works on half-sweep iteration can be found in Ibrahim and Abdullah [2], Akhir et al [15], Othman and Abdullah [17] and Yousif and Evans [23]. Motivated by this finding, Othman and Abdullah [18] extended the concept of half sweep iteration by introducing quarter-sweep iteration via the Modified Explicit Group (MEG) iterative method to solve two-dimensional Poisson equation. Further studies to verify the effectiveness of the quarter-sweep iteration have been carried out; see (Othman and Abdullah [19], and Sulaiman et al [10-11]).

In this paper, we examine the applications of quarter-sweep iterative concepts with the Modified Successive Over-Relaxation (MSOR) iterative method to solve two-dimensional Helmholtz equation. Again, the concept of this method is the extension of the HSMSOR method, which is inspired by Akhir et al (2010). To begin the derivation of QSMSOR method, let us consider the two-dimensional Helmholtz equation given by

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - \alpha U = f\left(x, y\right), \quad (x, y) \in D = [a, b] \times [a, b]. \qquad (1)$$

subject to the dirichlet boundary condition and statisfy the exact solution $U(x, y) = G(x, y)$, $(x, y) \in D = \partial D$. Where $f(x, y)$ is given function with suffi- icient smoothness and α is the nonnegative constant. Before implementing a formulation of the finite difference approximation equation, let us consider that the solution domain, D in Figure 1 needs to be partitioned uniformly in both x and y directions with fixed mesh size $h = 1/n$, where n is arbitrary positive integer.

In order to obtain finite grid networks for approximate finite approximate equation over equation (1), let us refer to Figure 1. Figure 1 acts as a guide for the implementation of the proposed computational algorithms. For that reason, the implementation of all points iterative algorithms will be applied onto the node points of the same type until the iterative convergence fixed is achieved.

The outline of this article is organized as follows. In Section 2, the formulation of the full-,half- and quarter-sweep finite difference approximation equations will be elaborated. The latter section of this article will discuss the formulations and derivation of the FSMSOR, HSMSOR and QSMSOR methods, and some numerical results will be shown in fourth section to assert the performance of the proposed methods. Besides that, analysis on computational complexity is mentioned in Section 5. Meanwhile, conclusions and open problems are given in Section 6 and 7 respectively.

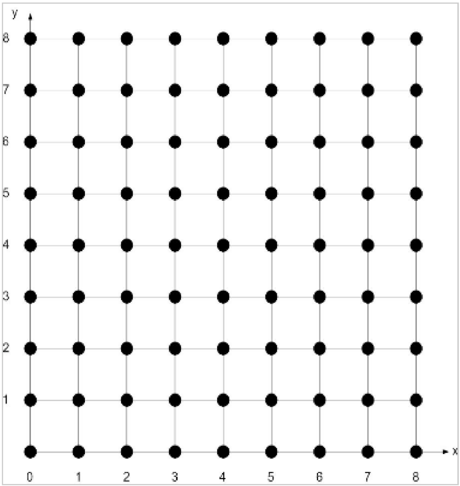
2. Quarter-Sweep Finite Difference Approximations Equations

As mentioned in the first section, the application of the QSMSOR method will be investigated to solve two-dimensional Helmholtz equation by the central difference finite difference approximation equations. In performing various iterative schemes such as full-, half- and quarter-sweep iterative methods, the finite grid networks show the distribution of uniform node points as a guide for development and implementations for the proposed algorithms.

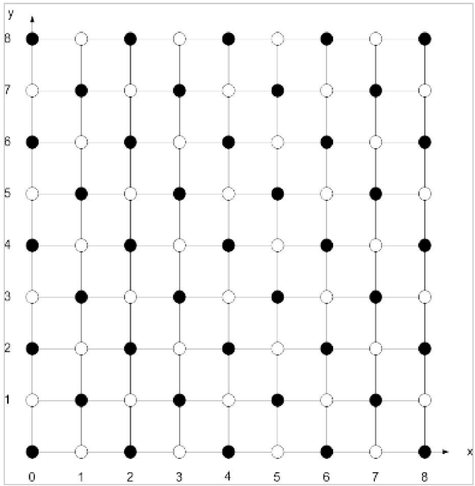
In order to derive full-, half- and quarter-sweep finite difference approximation equations for problem (1), this section is restricted to implement the second order finite difference schemes based on central difference method. Generally, when equation (1) is solved by the finite difference scheme the most commonly used approximation leads to the following formula

$$U_{i-p,j} + U_{i+p,j} + U_{i,-p} + U_{i,j+p} - \left(4 + \alpha p h^2\right) U_{i,j} = ph^2 f_{i,j}, \quad (3)$$

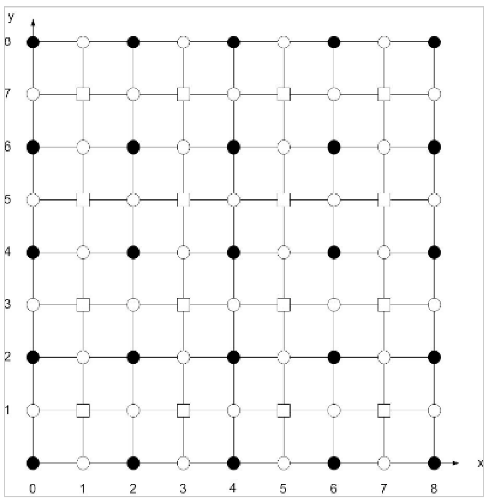
The value of p, which corresponds to 1 and 2, represents the case of full- and quarter-sweep method respectively. Apart from equation (**??**), the standard five



(a) Full-Sweep



(b) Half-Sweep



(c) Quarter-Sweep

Figure 1: The distribution of uniform nodal points for three sweep cases at $m = 8$

points rotated finite difference approximation can be obtained by rotating x-y

45^{o} (Dahqusit and Bjork [22]) can be formed by the following transformation:

$$\begin{gathered} i, j\pm \to 1j \pm 1, \\ i \pm 1, j\pm \to i \pm 1j \pm 1, \\ \Delta x, \Delta y \to \sqrt{(\Delta x)^2 + (\Delta x)^2} = \sqrt{2h},\ \Delta x = \Delta y. \end{gathered}$$

Based on the above transformation, the five point rotated finite difference approximation can be expressed as

$$U_{i+1,j+1} + U_{i-1,j-1} + U_{i+1,j-1} + U_{i-1,j+1} - \left(4 + 2\alpha h^2\right) U_{i,j} = 2ph^2 f_{i,j}, \quad (4)$$

equation (**??**) and equation (**??**) have truncation of order $0(h^2)$. Now, it can be clearly seen that the application to each internal mesh points has resulted a large and sparse system of linear algebraic equations, which can generally be stated as

$$Au = b, \quad (5)$$

where A and b are a square nonsingular matrix and a column matrix, respectively. While u is a column matrix shows the solution. The solution of linear system of linear algebraic equation (5) can be obtained by direct or iterative methods. Since the equation is large and sparse, the iterative method is suitable to solve this type of problem.

3. Derivation of the Quarter-Sweep MSOR Iterative Method

The idea of MSOR method is without doubt one of the efficient methods in solving a system of linear equations. The concept of the MSOR method was first considered by DeVogelaere [8], but it was mainly Young [5-6], Taylor [20], and Akhir et al [12-15] that has been extensively studied for demonstrating the capability of MSOR methods for solving a system linear equations According to Kincaid and Young [7], MSOR method is best applied in solving the system of linear equations when the two matrixes A are two-cyclic consistently ordered of equation (excluding red and black equations). Thus, the general schemes for MSOR iterative method can be written as follows

$$\begin{aligned} u^{(k+1)} &= (D - \theta_r L)^{-1} + \left[(\theta_r U - (1-\theta_r)D)\ u^{(k)} + \theta_r\ f\right], \\ u^{(k+1)} &= (D - \theta_b L)^{-1} + \left[(\theta_b U - (1-\theta_b)D)\ u^{(k)} + \theta_b\ f\right]. \end{aligned} \quad (6)$$

where $A = D - L - U$ in which D, L and U are the diagonal, negative lower triangulation and negative upper triangulation matrices, respectively. Here, the relaxation parameter θ_r is for "red" equations and θ_b is for the "black" equations. The performance of the MSOR method can be improved drastically with a proper choice of the relaxation parameter. Relaxation parameter can be calculated practically by consecutively choosing a value with some precision until the optimal value obtained. For the choice $\theta_r = \theta_b = 1$, MSOR method will reduce Gauss-Seidel (GS) method. Note that when and $\theta_r = \theta_b$ will result in the MSOR method coincides with the original Successive Over-Relaxation (SOR) method with red black ordering.

By determining the value of matrices D, L and U as method stated in (5), the general algorithm for FSSMOR, HSMSOR and QSMSOR methods to solve problem (1) would be generally described in Algorithm 1.

In this method, the Ω is divided into three types of points (i.e. •, ◯ and □) as shown in Figure 1 (c). The solutions on any points • can only be implemented by only involving the same type of point. The QSMSOR algorithm may be describes as follows

1. Divide the solution domain into three types of points (i.e. •, ◯ and □) as in Figure 1 (c). Compute the values of $4h^2$.

2. Iterate the intermediate solution u of point type • using equation (3)

2.1 $U_{i-2,j} + U_{i+2,j} + U_{i,-2} + U_{i,j+2} - \left(4 + \alpha (2h)^2\right) U_{i,j} = 4h^2 f_{i,j},$

3. Implement the relaxation parameter for θ_r and θ_b:

$$u^{(k+1)} = (D - \theta_r L)^{-1} + \left[(\theta_r U - (1-\theta_r)D)\ u^{(k)} + \theta_r\ f\right],$$
$$u^{(k+1)} = (D - \theta_b L)^{-1} + \left[(\theta_b U - (1-\theta_b)D)\ u^{(k)} + \theta_b\ f\right],$$

3. Check the convergence. If converge evaluate the rest of points (i.e. • and ◯) using,

3.1. $U_{i+1,j+1} + U_{i-1,j-1} + U_{i+1,j-1} + U_{i-1,j+1} - \left(4 + 2\alpha h^2\right) U_{i,j} = 2h^2 f_{i,j},$

3.2. $U_{i-1,j} + U_{i+1,j} + U_{i,-1} + U_{i,j+1} - \left(4 + \alpha h^2\right) U_{i,j} = h^2 f_{i,j}.$

respectively. Otherwise repeat the iteration cycle (i.e., go to step 2)

4. Stop

4. Numerical Experiments

In order to verify the effectiveness of the proposed methods, several numerical tests were conducted. The computer language used for the programming is C++, and the programs performed on CPU Intel Core i7 Processor@2.80Ghz with memory is 4. 00G. In comparison, the Full-Sweep Gauss-Seidel (FSGS) method acts as the control of comparison of the numerical results. Three criteria such as the number of iterations, execution time and maximum absolute error will be considered in comparison for FSGS. In the following examples, the convergence test considered the tolerance error$\varepsilon = 10^{-10}$.

Example 1 (Evans, 1985)

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - \alpha U = 6 - \alpha\left(2x^2 + y^2\right), \quad (x, y) \in D = [0,1] \times [0,1].$$

and the exact solution of the problem given by

$$U(x, y) = 2x^2 + y^2.$$

Example 2 (Evans *et al*, 2003)

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - 2U = \left(12x^2 + 3x^4\right)\sin y, \quad (x, y) \in D = [0,1] \times [0, 2\pi].$$

with the exact solution

$$U(x, y) = x^4 \sin y.$$

Result of numerical experiments, which were obtained from implementations of the FSGS, FSMSOR, HSMSOR and QSMSOR methods for Example 1 and 2, has been recorded in Table 2-4.

5. Computational Complexity Analysis

The computational effort measured by number of computer operations needed to obtained a (sufficient accurate) solution by the three methods discussed for solving problem (1) can be estimated. Assume the solution domain is large with m^2 number of internal mesh points with $m = n-1$. In their iterative process, the FSMSOR and HSMSOR methods requires $(m-1)^2$ and $(m-1)^2 \big/ 2$ internal mesh points respectively. While QSMSOR method require $(m-1)^2 \big/ 4$ internal mesh points.

Methods	Arithmetic Operation	
	(ABB/SUB)	(MUL/DIV)
FSMSOR	$(10k)\, m^2$	$(8k)\, m^2$
HSMSOR	$(5k + 2)\, m^2$	$(4k + 1)\, m^2$
QSMSOR	$\left(\frac{5}{2}k + \frac{3}{2}\right) m^2$	$\left(2k + \frac{3}{2}\right) m^2$

Table 1: Total number of arithmetic operations per iteration for FSMSOR, HSMSOR and QSMSOR methods

Note that our estimate on this computational complexity is based on the arithmetic operations performed per iteration and execution time for the additions/subtraction (ADD/SUB) and multiplications/divisions (MUL/DIV) operations. Hence the number of operations of operations required (excluding red and black equations, convergence test and direct solution) for FSMSOR, HSMSOR and QSMSOR methods as described in Section 3 are respectively given as follows in Table 1,

Note. k is the number of iterations and m^2 represents $(m - 1)$.

6. Results and Discussion

In the previous section, we present formulation of full-, half, and quarter-sweep approximation equations based on the second order finite difference method can easily generate a system of linear algebraic equations as shown in equation 5. From Tables 2 and 3, clearly show that by applying MSOR methods can reduce the number of iterations compared to FSGS method. Table 4 shows a decrement percentages number of iterations for FSMSOR, HSMSOR and QSMSOR methods. Through the surveillance in Tables 2 and 3, found that application of the quarter-sweep concepts reduced the execution time of the iterative method. Meanwhile, decrement percentages of the execution time for FSMSOR, HSMSOR and QSMSOR methods compared with FSGS method have been summarized in Table 4. In addition, the accuracy approximate solutions for FSMSOR, HSMSOR and QSMSOR methods are in good agreement compared with the FSGS method.

7. Conclusions and Future Research

As a conclusion, the numerical results prove that QSMSOR iterative method is a better method compared with the HSMSOR, FSMSOR and FSGS methods in the sense of the complexity and execution time. This is due to the computational complexity of the QSMSOR method is approximately 25% and 50% less than FSMSOR and HSMSOR methods respectively; refer Table 1. For our future works, this study will be extended to investigate the effectiveness of the QSMSOR method for solving other multidimensional partial differential equations.

Mesh size	**Methods**	**Numbers of Iterations**	**Execution Times (Seconds)**	**Maximum Absolute Error**
	FSGS	1326	0.26	6.8176e-10
	FSMSOR	340	0.16	1.4690e-11
		(θ_r =1.61 & θ_b =1.62)		
32	**HSMSOR**	169	0.08	5.0787e-10
		(θ_r =1.86 & θ_b =1.87)		
	QSMSOR	79	0.03	1.9967e-11
		(θ_r =1.59 & θ_b =1.60)		
	FSGS	4910	1.38	2.7407e-10
	FSMSOR	534	0.22	2.5031e-11
		(θ_r =1.81 & θ_b =1.82)		
64	**HSMSOR**	257	0.13	5.6165e-10
		(θ_r =1.90 & θ_b =1.91)		
	QSMSOR	133	0.06	3.2468e-11
		(θ_r =1.79 & θ_b =1.80)		
	FSGS	18085	15.43	1.1004e-10
	FSMSOR	1155	0.98	5.8664e-11
		(θ_r =1.89 & θ_b =1.90)		
128	**HSMSOR**	551	0.51	2.6136e-10
		(θ_r =1.89 & θ_b =1.88)		
	QSMSOR	263	0.24	9.3119e-11
		(θ_r =1.89 & θ_b =1.90)		
	FSGS	66177	210.11	4.4067e-10
	FSMSOR	4510	14.07	1.5182e-11
		(θ_r =1.88 & θ_b =1.89)		
256	**HSMSOR**	2112	4.71	7.1276E-10
		(θ_r =1.93 & θ_b =1.94)		
	QSMSOR	1190	2.35	7.4974e-11
		(θ_r =1.96 & θ_b =1.97)		

Table 2: Comparison of a number of iterations, execution times (seconds) and maximum absolute error for the iterative methods (Example 1) at $\alpha = 10$.

Mesh size	**Methods**	**Numbers of Iterations**	**Execution Times (Seconds)**	**Maximum Absolute Error**
	FSGS	1694	0.34	9.7880e-3
	FSMSOR	305	0.15	1.4115e-3
		(θ_r =1.92 &θ_b =1.93)		
32	**HSMSOR**	149	0.06	1.4095e-2
		(θ_r =1.85 &θ_b =1.86)		
	QSMSOR	70	0.03	1.4152e-3
		(θ_r =1.69 &θ_b =1.70)		
	FSGS	6175	1.56	9.7789e-3
	FSMSOR	533	0.23	1.4114e-3
		(θ_r =1.85 &θ_b =1.86)		
64	**HSMSOR**	263	0.14	1.4108e-2
		(θ_r =1.91 &θ_b =1.92)		
	QSMSOR	136	0.07	1.4125e-3
		(θ_r =1.83 &θ_b =1.84)		
	FSGS	22340	16.81	9.7906e-3
	FSMSOR	1055	0.89	1.4116e-3
		(θ_r =1.92 &θ_b =1.91)		
128	**HSMSOR**	510	0.45	1.4115e-2
		(θ_r =1.92 &θ_b =1.91)		
	QSMSOR	290	0.23	1.4113e-3
		(θ_r =1.92 &θ_b =1.91)		
	FSGS	80028	225.82	9.7923e-3
	FSMSOR	2537	7.84	1.4116e-3
		(θ_r =1.95 & θ_b =1.96)		
256	**HSMSOR**	1287	4.36	1.4119e-2
		(θ_r =1.95 &θ_b =1.96)		
	QSMSOR	674	2.28	1.4117e-3
		(θ_r =1.97 &θ_b =1.96)		

Table 3: Comparison of a number of iterations, execution times (seconds) and maximum absolute error for the iterative methods (Example 2).

Example	**Method**	**Numbers of Iterations (%)**	**Execution Time (%)**
	FSMSOR	74.36-93.91	38.46-93.65
1	**HSMSOR**	87.25-96.95	69.23-97.76
	QSMSOR	94.04-98.55	88.46-99.34
	FSMSOR	81.99-96.83	55.88-96.53
2	**HSMSOR**	91.20-98.39	82.35-98.07
	QSMSOR	95.87-99.03	89.10-99.41

Table 4: Decrement percentages of the number of iterations and execution time for FSMSOR, HSMSOR and QSMSOR methods compared with the FSGS.

References

[1] A.R. Abdullah, The four point explicit decoupled group (EDG) method: A fast poisson solver, *Int. J. Comp. Math.*, **38** (1991), 61-70.

[2] A. Ibrahim, A.R. Abdullah, Solving the two-dimensional diffusion equation by the four point explicit decoupled group (EDG) iterative method, *Int. J. Comp. Math.*, **58** (1995), 253-256.

[3] D.J. Evans, Group explicit iterative methods for solving large linear systems, *Int. J. Comp. Math.*, **17** (1985), 81-108.

[4] D.J. Evans, M. Ergut, H. Bulut, The numerical solution of multidimensional partial differential equations by decomposition method, *Int. J. Comp. Math.*, **80** (2003), 1189-1198.

[5] D.M. Young, *Iterative Solution of Large Linear Systems*, London, Academic Press (1971).

[6] D.M. Young, Second-degree iterative methods for the solution of large linear systems, *J. Approx.*, **5** (1972), 37-148.

[7] D.R. Kincaid, D.M. Young, The modified successive over relaxation method with fixed parameters, *J. Math. Comp.*, **26** (1972), 705-717.

[8] D. Vogelaere, Over-relaxations, Abstract #539-5, *Notices Amer. Math. Soc.*, **5** (1958), 147.

[9] J. Sulaiman, M.K. Hasan, M. Othman, The half-sweep iterative alternating decomposition explicit (HSIADE) method for diffusion equations, in computational and information science, *J. Zhang* (E-ds: J.H. He, Y.Fu), Berlin, Springer-Verlag (2004), 57-63.

[10] J. Sulaiman, M.K. Hasan, M. Othman, Quarter-sweep iterative alternating decomposition explicit algorithm applied to diffusion equations, *Int. J. Comp. Math.*, **81** (2004), 1559-1565.

[11] J. Sulaiman, M.K. Hasan, M. Othman, A new quarter-sweep arithmetic mean (QSAM) method to solve diffusion equations, *Chamchuri J. Math.*, **2** (2009), 93-103.

[12] M.K. Akhir, M. Othman, J. Sulaiman, Z.A Majid, Solving two-dimensional Helmholtz equations by modified SOR iterative method, *Proceedings of Fundamental Science (FSC)*, 18-19 September (2009), 88-89.

[13] M.K. Akhir, M.Othman, J.Sulaiman, Z.A. Majid, A new half-sweep modified SOR for solving two-dimensional Helmholtz equation, In: *Proceedings of International Conference on Intelligent Network and Computing (IC-NIC)*, 26-28 November (2010), 416-419.

[14] M.K. Akhir, M. Othman, J. Sulaiman, Z.A. Majid, M. Suleiman, Numerical solutions of Helmholtz using a new four point EGMSOR iterative method equations, *Appl. Math. Science*, **80** (2011), 3991-4004.

[15] M.K. Akhir, M. Othman, J. Sulaiman, Z.A. Majid, M. Suleiman, The four point-EDGMSOR iterative method for solution of 2D Helmholtz equations, In: *Communications in Computer and Information Science*, A. Abd Manaf., Berlin, Springer-Verlag, **253** (2011), 218-227.

[16] M. Nabavi, M.H.K Sidiqui, J. Daraghi, A new 9-point sixth-order accurate compact finite difference methods for Helmholtz equation, *J. Sound Vibrat*, **307** (2007), 972-982.

[17] M. Othman, A.R. Abdullah, The halfsweeps multigrid method as a fast multigrid Poisson solver, *Int. J. Comp. Math.*, **69** (1998), 219-229.

[18] M. Othman, A.R. Abdullah, An efficient four points modified explicit group Poisson solver, *Int. J. Comp. Math.*, **76** (2000), 972-982.

[19] M. Othman, A.R. Abdullah, A parallel four point modified explicit group iterative algorithm on shared memory multiprocessor, *Int. J. Paral. Emer. Dsitri. Sys.*, **19** (2004), 1-9.

[20] P.J. Taylor, A generalization of systematic relaxation methods for consistently ordered, *J. Numer. Math.*, **13** (1969), 377-395.

[21] T. Belytschko, Y. Kongraus, D. Organ, D. Fleming, P. Krystal, Meshless methods: An overview and recent developments, *J. Comput. Meth. Appl., Mecha. Eng.*, **139** (2007), 3-47.

[22] V. Dahlquist, A. Bjorck, *Numerical Method*, New York, Prentice Hall (1974).

[23] W.S. Yousif, D.J. Evans, Explicit de-coupled group iterative methods and their implementations, *Int. J. Paral. Emer. Dsitri. Sys.*, **7** (1995), 53-71.

[24] Y. Saad, *Iterative Method for Sparse Linear Systems*, Thomas, Boston (1996).