

Modified Explicit Decoupled Group Scheme On Helmholtz Equation

Norhashidah Hj. Mohd Ali
 School of Mathematical Sciences
 Universiti Sains Malaysia
 11800 Penang, Malaysia
 Email: shidah@cs.usm.my

Farhana Aziz
 School of Mathematical Sciences
 Universiti Sains Malaysia
 11800 Penang, Malaysia
 Email: farhana.aziz85@gmail.com

Abstract—In this paper, the formulation of a new group iterative method called the Modified Explicit Decoupled Group method in solving the two dimensional Helmholtz equation is described. The method is derived using a combination of the five-point finite difference approximation on the rotated grid stencil together with the five-point centred difference approximation on the standard grid stencils. Numerical experimentations of this new formulation shows significant improvement in computational complexity and execution timings over the original Explicit Decoupled Group method [2].

Keywords—explicit decoupled group; Helmholtz equation; finite difference; rotated grid

I. INTRODUCTION

Over the last few decades, fast explicit group methods for solving partial differential equations using finite difference schemes derived from skewed (rotated) difference operators have been extensively investigated [1,2,3,4,5,6,10,11]. In particular, the formulation of the Explicit Decoupled Group (EDG) method was presented in [1] as an addition to the family of four-point explicit group methods in solving the Poisson equation which was shown to be more economical computationally than the Explicit Group (EG) scheme due to Yousif and Evans [9]. The method uses a skewed difference formula which leads to lower computational complexities since the iterative procedure need only involve nodes on half of the total grid points in the solution domain and thus a reduced system of linear equations is attained. In 2007, a modification to the EDG method was introduced in the iterative process by combining the the *rotated* five-point finite difference approximation on the $\Omega_{\sqrt{2}h}$ grid together with the five-point centred difference approximation on the Ω_h and Ω_{2h} grids and was shown to have a better rate of convergence than the original EDG method in solving the Poisson equation. In this paper, we extend the idea of the modification to the EDG method to investigate whether this strategy is able to produce a more improved scheme in solving the two-dimensional Helmholtz equation. The derivation of the relevant finite difference formulas in solving the Helmholtz equation is presented in Section II followed by the formulation of the MEDG method in Section III. Section IV presents the analysis

of the computing efforts of the explicit group methods. Section V describes the numerical experiments carried out and the results obtained, followed by the conclusions in Section VI.

II. FINITE DIFFERENCE DISCRETIZATIONS

Consider the two dimensional Helmholtz equation as follows:

$$u_{xx} + u_{yy} - \alpha u = f(x, y) \quad (1)$$

with Dirichlet boundary conditions

$$u(x, y) = g(x, y), \quad (x, y) \in \partial\Omega.$$

Here α is a non-negative constant and $f(x, y)$ is a function in a bounded region Ω in \mathfrak{R}^2 . Many problems related to steady-state oscillations (e.g. mechanical, thermal) lead to the two dimensional Helmholtz equation. The simplest finite difference formula to approximate (1) is the five-point difference approximation formula:

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - (4 + h^2\alpha)u_{i,j} = h^2 f_{i,j}. \quad (2)$$

We assume that a rectangular grid in the (x,y) plane with grid spacing h in both directions with $x_i = ih, y_j = jh$ is used and $u_{i,j} = u(x_i, y_j)$ with $i, j = 0, 1, 2, \dots, n$. Another type of approximation that can represent the Helmholtz equation is based on the cross orientation operator which can be obtained by rotating the i-plane axis and the j-plane axis clockwise by 45° [7]. This operator may be expressed in coordinates rotated 45° with respect to the original mesh and the spacing between points becomes $\sqrt{2}h$. This will result in the *rotated* (skewed) five-point approximation formula:

$$u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1} - (4 + 2h^2\alpha)u_{i,j} = 2h^2 f_{i,j} \quad (3)$$

By applying central difference approximation to $2h$ -spaced grid points, (1) may be approximated by a $2h$ -spaced centred difference formula:

$$u_{i+2,j} + u_{i-2,j} + u_{i,j+2} + u_{i,j-2} - (4 + 4h^2\alpha)u_{i,j} = 4h^2 f_{i,j} \quad (4)$$

Financial support provided by the Fundamental Research Grant Scheme #203/PMATHS/6711188 for the completion of this article is gratefully acknowledged.

Using the two dimensional Taylor series expansion, a rotated $\sqrt{8}h$ -spaced difference formula can be obtained as:

$$u_{i+2,j+2} + u_{i+2,j-2} + u_{i-2,j+2} + u_{i-2,j-2} - (4 + 8h^2\alpha)u_{i,j} = 8h^2 f_{i,j} \quad (5)$$

Using the combination of finite difference approximation formulas (1)-(5), we shall now describe the formulation of the proposed explicit group method in the next section.

III. FORMULATION OF THE EXPLICIT GROUP METHODS

A. Explicit Decoupled Group (EDG) Method

The EDG method was formulated by applying (3) to any group of four points in the solution domain where the following (4x4) system of equations is obtained [2]:

$$\begin{bmatrix} 4 + 2h^2\alpha & -1 & 0 & 0 \\ -1 & 4 + 2h^2\alpha & 0 & 0 \\ 0 & 0 & 4 + 2h^2\alpha & -1 \\ 0 & 0 & -1 & 4 + 2h^2\alpha \end{bmatrix} \begin{bmatrix} u_{i,j} \\ u_{i+1,j+1} \\ u_{i+1,j} \\ u_{i,j+1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad (6)$$

where

$$\begin{aligned} b_1 &= u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} - 2h^2 f_{i,j} \\ b_2 &= u_{i+2,j+2} + u_{i,j+2} + u_{i+2,j} - 2h^2 f_{i+1,j+1} \\ b_3 &= u_{i+2,j+1} + u_{i,j-1} + u_{i+2,j-1} - 2h^2 f_{i+1,j} \\ b_4 &= u_{i+1,j+2} + u_{i-1,j} + u_{i-1,j+2} - 2h^2 f_{i,j+1} \end{aligned} \quad (7)$$

The (4x4) matrix in (6) can be inverted to give

$$\begin{bmatrix} u_{i,j} \\ u_{i+1,j+1} \\ u_{i+1,j} \\ u_{i,j+1} \end{bmatrix} = \frac{1}{\beta} \begin{bmatrix} 4 + 2h^2\alpha & 1 & 0 & 0 \\ 1 & 4 + 2h^2\alpha & 0 & 0 \\ 0 & 0 & 4 + 2h^2\alpha & 1 \\ 0 & 0 & 1 & 4 + 2h^2\alpha \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad (8)$$

which can be decoupled as

$$\begin{bmatrix} u_{i,j} \\ u_{i+1,j+1} \end{bmatrix} = \frac{1}{\beta} \begin{bmatrix} 4 + 2h^2\alpha & 1 \\ 1 & 4 + 2h^2\alpha \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (9)$$

and

$$\begin{bmatrix} u_{i+1,j} \\ u_{i,j+1} \end{bmatrix} = \frac{1}{\beta} \begin{bmatrix} 4 + 2h^2\alpha & 1 \\ 1 & 4 + 2h^2\alpha \end{bmatrix} \begin{bmatrix} b_3 \\ b_4 \end{bmatrix} \quad (10)$$

where $\beta = (4 + 2h^2\alpha)^2 - 1$.

From Fig. 1, it may be observed that the evaluation of (9) involves points of type ● (including the ungrouped ○ points) only, while (10) can be evaluated involving points of type ▲ only. Thus, the calculations of (9) and (10) can be carried out independently which may save the execution time by nearly half if the iteration is carried out on only one type of points; either the type ● or ▲. Fig. 1 shows the various types of points involved if iteration is carried out on ● points (grouped) using (9) and on ○ points (ungrouped) using (3) for $n=14$. After

convergence is achieved, the solutions at the other remaining half of the points (▲) are computed directly once using the standard five-point formula (2).

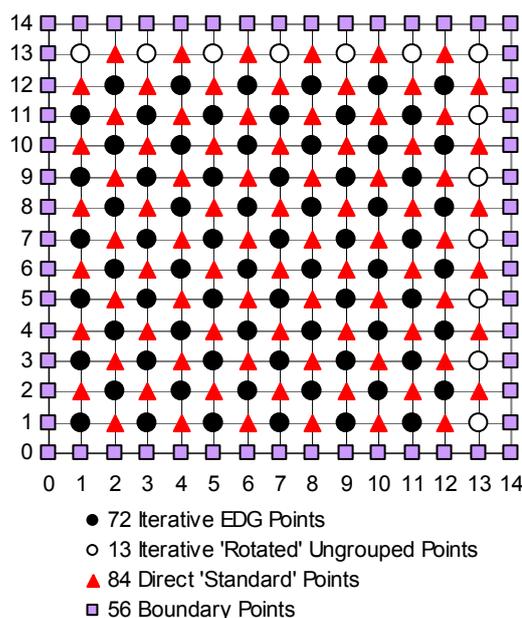


Fig. 1. Types of points involved in EDG method for $n = 14$.

B. Modified Explicit Decoupled Group (MEDG) Method

To formulate the MEDG method, we apply the rotated $\sqrt{8}h$ -spaced difference formula to groups of four points as shown in Fig. 2 and produce the following (4x4) system of equations

$$\begin{bmatrix} 4 + 8h^2\alpha & -1 & 0 & 0 \\ -1 & 4 + 8h^2\alpha & 0 & 0 \\ 0 & 0 & 4 + 8h^2\alpha & -1 \\ 0 & 0 & -1 & 4 + 8h^2\alpha \end{bmatrix} \begin{bmatrix} u_{i,j} \\ u_{i+2,j+2} \\ u_{i+2,j} \\ u_{i,j+2} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \quad (11)$$

where

$$\begin{aligned} c_1 &= u_{i-2,j-2} + u_{i-2,j+2} + u_{i+2,j-2} - 8h^2 f_{i,j} \\ c_2 &= u_{i+4,j+4} + u_{i,j+4} + u_{i+4,j} - 8h^2 f_{i+2,j+2} \\ c_3 &= u_{i+4,j+2} + u_{i,j-2} + u_{i+4,j-2} - 8h^2 f_{i+2,j} \\ c_4 &= u_{i+2,j+4} + u_{i-2,j} + u_{i-2,j+4} - 8h^2 f_{i,j+2} \end{aligned} \quad (12)$$

which can be inverted and rewritten in explicit forms

$$\begin{bmatrix} u_{i,j} \\ u_{i+2,j+2} \\ u_{i+2,j} \\ u_{i,j+2} \end{bmatrix} = \frac{1}{\gamma} \begin{bmatrix} 4 + 8h^2\alpha & 1 & 0 & 0 \\ 1 & 4 + 8h^2\alpha & 0 & 0 \\ 0 & 0 & 4 + 8h^2\alpha & 1 \\ 0 & 0 & 1 & 4 + 8h^2\alpha \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \quad (13)$$

where $\gamma = (4 + 8h^2\alpha)^2 - 1$. This system can be decoupled as a system of (2x2) equations

$$\begin{bmatrix} u_{i,j} \\ u_{i+2,j+2} \end{bmatrix} = \frac{1}{\gamma} \begin{bmatrix} 4 + 8h^2\alpha & 1 \\ 1 & 4 + 8h^2\alpha \end{bmatrix} \begin{bmatrix} u_{i-2,j-2} + u_{i+2,j-2} + u_{i-2,j+2} - 8h^2 f_{i,j} \\ u_{i+4,j+4} + u_{i+4,j} + u_{i+4,j+4} - 8h^2 f_{i+2,j+2} \end{bmatrix} \quad (14)$$

and

$$\begin{bmatrix} u_{i+2,j} \\ u_{i,j+2} \end{bmatrix} = \frac{1}{\gamma} \begin{bmatrix} 4+8h^2\alpha & 1 \\ 1 & 4+8h^2\alpha \end{bmatrix} \begin{bmatrix} u_{i,j-2} + u_{i+4,j-2} + u_{i-4,j+2} - 8h^2 f_{i+2,j} \\ u_{i-2,j+4} + u_{i-2,j} + u_{i+2,j+4} - 8h^2 f_{i,j+2} \end{bmatrix} \quad (15)$$

It can be observed that the evaluation of (14) and (15) can be performed independently. Fig. 3 shows the discretization points of a unit square domain with $n=14$ and the various types of points involved. It is obvious that the evaluation of (14) involves only points of type ● and the evaluation of (15) involves only points of type ▲. In this paper we solve points of type ● iteratively using (14) until convergence after which the points of type ▲ is computed directly once using the standard $2h$ spaced five-point formula (4). The remaining in-between points of type □ are also computed directly once using the rotated five-point difference formula (3), and followed by points of type ◇ using the standard five-point difference formula of (2).

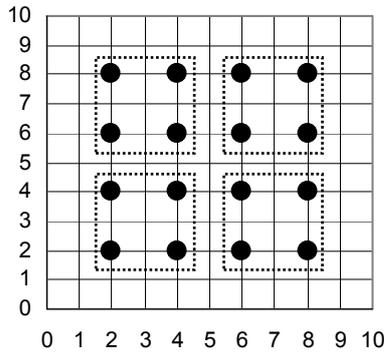


Fig. 2. Group of four points with $2h$ spacing

We define the four point MEDG method combined with Successive Over-Relaxation (SOR) scheme as follows:

1. Divide the solution domain into five types of points as shown in Fig. 3 (for the case $n=14$).
2. Group all the $2h$ spaced ● points into two-points groups as shown in Fig. 4.
3. Iterate on the points ● in each group using

$$\begin{bmatrix} \hat{u}_{i,j}^{(k+1)} \\ \hat{u}_{i+2,j+2}^{(k+1)} \end{bmatrix} = \frac{1}{\gamma} \begin{bmatrix} 4+8h^2\alpha & 1 \\ 1 & 4+8h^2\alpha \end{bmatrix} \begin{bmatrix} u_{i-2,j-2}^{(k+1)} + u_{i+2,j-2}^{(k+1)} + u_{i-2,j+2}^{(k)} - 8h^2 f_{i,j} \\ u_{i,j+4}^{(k)} + u_{i+4,j}^{(k)} + u_{i+4,j+4}^{(k)} - 8h^2 f_{i+2,j+2} \end{bmatrix}$$

and implement the SOR relaxation scheme

$$\begin{bmatrix} u_{i,j}^{(k+1)} \\ u_{i+2,j+2}^{(k+1)} \end{bmatrix} = \begin{bmatrix} u_{i,j}^{(k)} \\ u_{i+2,j+2}^{(k)} \end{bmatrix} + \omega \left(\begin{bmatrix} \hat{u}_{i,j}^{(k+1)} \\ \hat{u}_{i+2,j+2}^{(k+1)} \end{bmatrix} - \begin{bmatrix} u_{i,j}^{(k)} \\ u_{i+2,j+2}^{(k)} \end{bmatrix} \right)$$

4. If solution converged, go to step 5. Otherwise, repeat the iteration step 3.
5. Evaluate the remaining points in this sequence:

a. points of type ▲

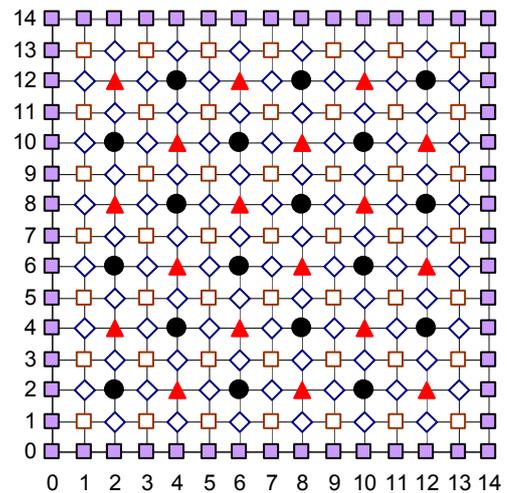
$$u_{i,j} = \frac{1}{4+4h^2\alpha} (u_{i+2,j} + u_{i-2,j} + u_{i,j+2} + u_{i,j-2} - 4h^2 f_{i,j})$$

b. points of type □

$$u_{i,j} = \frac{1}{4+2h^2\alpha} (u_{i+1,j+1} + u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} - 2h^2 f_{i,j})$$

c. points of type ◇

$$u_{i,j} = \frac{1}{4+h^2\alpha} (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - h^2 f_{i,j})$$



- 18 Iterative MEDG Points ($2h$)
- ▲ 18 Direct 'Standard' Points ($2h$)
- 49 Direct 'Rotated' Points (h)
- ◇ 84 Direct 'Standard' Points (h)
- 56 Boundary Points

Fig. 3. Types of points in MEDG for $n = 14$

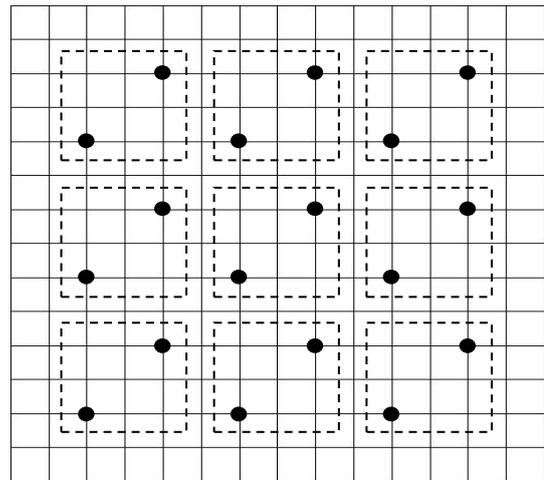


Fig. 4. Groups of iterative points in MEDG for $n = 14$

IV. COMPUTATIONAL COMPLEXITY ANALYSIS

Assume that the solution domain is discretized with even integer n , then the number of internal mesh points is given by m^2 where $m=n-1$. Basically, there are two types of internal mesh points namely, iterative points which are points involved in the iteration process, and direct points which solutions are computed directly once after the iteration process converged. TABLE I lists the number of points for the various internal mesh points for the proposed MEDG method as well as for the original EDG method due to [2].

Next we estimate the computational complexity of MEDG method in terms of arithmetic operations performed in an iteration (excluding the convergence test). From (14) we obtain

$$\begin{bmatrix} u_{i,j} \\ u_{i+2,j+2} \end{bmatrix} = \frac{1}{\gamma} \begin{bmatrix} 4 + 8h^2\alpha & 1 \\ 1 & 4 + 8h^2\alpha \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (16)$$

where

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} u_{i-2,j-2} + u_{i+2,j-2} + u_{i-2,j+2} - 8h^2 f_{i,j} \\ u_{ij+4} + u_{i+4,j} + u_{i+4,j+4} - 8h^2 f_{i+2,j+2} \end{bmatrix} \quad (17)$$

Thus, the updated value u_{ij} is in the form

$$\tilde{u}_{i,j} = p * b_1 + q * b_2 \quad (18)$$

and the SOR iterative scheme is given by

$$u_{i,j}^{(k+1)} = u_{i,j}^{(k)} + \omega(\tilde{u}_{i,j} - u_{i,j}^{(k)}). \quad (19)$$

The updates in the iterations are done in groups of two points with the values of b_1 and b_2 calculated only once before the computation of those two points of u . This results in 6 additions (*adds*) and 4 multiplications (*mults*) for a single point, provided the constants p , q and $8h^2$ are computed and stored beforehand. Similarly it can be shown that the computational cost of a single iterative grouped point in the EDG method is 6 *adds*+4 *mults*. The cost of computing an iterative ungrouped point in EDG method using the rotated five-point difference formula is 6 *adds*+3 *mults* while the direct solution after convergence requires 4 *adds*+2 *mults* per point. Hence, the number of arithmetic operations required in an iteration and in the direct solution after convergence for the EDG and MEDG methods in terms of m are given in TABLE II.

TABLE I. NUMBER OF DIFFERENT TYPES OF POINTS INVOLVED IN EDG SOR [2] AND MEDG SOR METHODS

Types of points	EDG	MEDG
Iterative grouped points	$(m-1)^2/2$	$(m-1)^2/8$
Iterative ungrouped points	m	None
Total iterative points, ip	$(m^2+1)/2$	$(m-1)^2/8$
Direct $2h$ spaced 'standard' points	None	$(m-1)^2/8$
Direct h spaced 'rotated' points	None	$(m+1)^2/4$
Direct h spaced 'standard' points	$(m^2-1)/2$	$(m^2-1)/2$
Total direct points, dp	$(m^2-1)/2$	$(7m^2+2m-1)/8$
Total internal points, $ip+dp$	m^2	m^2

V. NUMERICAL EXPERIMENTS

To test the effectiveness of the proposed method, several numerical experiments were conducted. The proposed MEDG SOR method was implemented to solve the following model [8]:

TABLE II. COMPUTING EFFORTS OF EDG SOR [2] AND MEDG SOR METHODS

Method	Per Iteration		After convergence	
	<i>adds</i>	<i>mults</i>	<i>adds</i>	<i>mults</i>
EDG	$3(m-1)^2 + 6m$	$2(m-1)^2 + 3m$	$2(m^2-1)$	(m^2-1)
MEDG	$3(m-1)^2/4$	$(m-1)^2/2$	$(7m^2+2m-1)/2$	$(7m^2+2m-1)/4$

$$u_{xx} + u_{yy} - \alpha u = 6 - \alpha(2x^2 + y^2), \quad (x, y) \in [0, 1] \times [0, 1] \quad (20)$$

with Dirichlet boundary conditions satisfying the exact solution $u(x, y) = 2x^2 + y^2$. The tolerance used was $\epsilon = 1.0 \times 10^{-10}$ and the acceleration parameter, ω_{opt} , was chosen between 1 to 2 (up to ± 0.0001 accuracy) which give the least number of iterations. Different grid sizes of $n = 42, 114, 142, 214$ and 242 were chosen to record the timings in seconds, the iteration counts and the average absolute errors of all the group methods described in Section III. The values of n were chosen randomly such that $(n-2)\%4$ equals 0 to ensure that there are no ungrouped points for the MEDG method (here, % is the remainder operator). All of the experiments were carried out on a netbook with AMD E-450 APU 1.65GHz, 2GB RAM running on Windows 7 Starter Edition using Microsoft Visual C++ 2010 Express. For comparison purposes, the classical point SOR method derived from the standard centred difference formula was also included in the experiments. The numerical results for the point SOR (P SOR), EDG SOR and MEDG SOR are shown in Table III.

TABLE III. PERFORMANCE COMPARISON BETWEEN POINT SOR, EDG SOR [2] AND MEDG SOR

Mesh Size	Method	Relaxation Factor ω_{opt}	Number of Iterations	Execution Time (Seconds)	Average Absolute Error
42	P SOR	1.8299	168	0.061	8.228E-11
	EDG SOR	1.7516	86	0.015	7.741E-12
	MEDG SOR	1.5638	43	0.004	5.701E-12
114	P SOR	1.9350	457	1.010	9.814E-13
	EDG SOR	1.8999	230	0.237	5.382E-11
	MEDG SOR	1.8105	115	0.042	8.540E-12
142	P SOR	1.9474	569	2.025	1.166E-12
	EDG SOR	1.9191	286	0.313	9.544E-12
	MEDG SOR	1.8448	143	0.069	8.495E-12
214	P SOR	1.9632	856	6.930	7.315E-10
	EDG SOR	1.9456	429	1.520	9.045E-12
	MEDG SOR	1.8934	212	0.223	1.815E-10
242	P SOR	1.9673	968	15.4	8.741E-10
	EDG SOR	1.9515	482	3.626	9.702E-11
	MEDG SOR	1.9050	241	0.582	2.080E-10

TABLE IV. REDUCTION PERCENTAGES OF THE NUMBER OF ITERATIONS AND EXECUTION TIME FOR THE EDG SOR AND MEDG SOR METHODS COMPARED TO PSOR METHOD.

Method	Number of Iteration (%)	Execution Time (%)
EDG SOR	48.81 - 50.21	75.41 – 84.54
MEDG SOR	74.40 – 75.23	93.44 – 96.78

It is clearly seen from the results tabulated in Table III, the MEDG SOR method is the fastest amongst the tested methods. From the reduction percentages of execution time (in seconds) in Table IV, it is shown that MEDG SOR reduces the execution time about 93% to 97% of the PSOR method, followed by the EDG SOR by about 75% to 85% of the PSOR method. In terms of number of iterations, the MEDG SOR requires about a quarter of the iterations required by PSOR method while the EDG SOR method requires about a half of PSOR. Among the three methods, the MEDG SOR method requires the least computational effort whilst maintaining the same degree of accuracy.

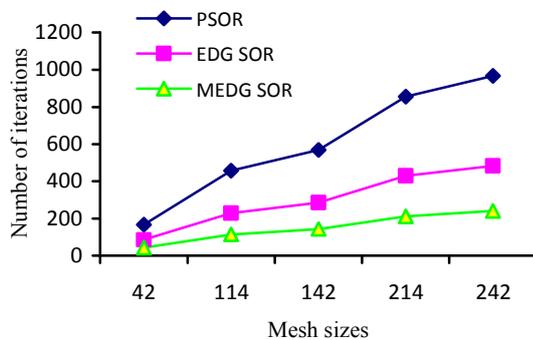


Fig. 5. Number of iterations for the three methods for different mesh sizes

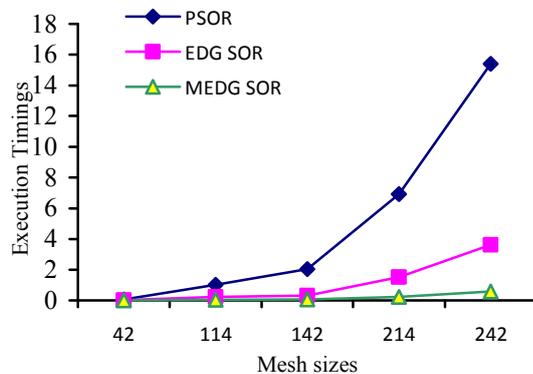


Fig. 6. Execution timings (in secs) for the three methods for different mesh sizes

TABLE V tabulates the total computing efforts in terms of operation counts for the two group methods for different grid sizes n . The total number of arithmetic operations for these methods were obtained by combining the results from the experimental number of iterations shown in TABLE III with the number of operations required in each iteration by each

method. Here, k is the number of iterations from each method obtained from TABLE III. From the experimental results, we can clearly see that the MEDG SOR method shows better execution timings than the EDG SOR method which is in agreement with its analysis of computing efforts.

TABLE V. COMPUTING EFFORTS FOR EDG SOR AND MEDG SOR.

n	Methods			
	EDG SOR		MEDG SOR	
	k	Operation count	k	Operation count
42	86	724,774	43	94,886
114	230	14,697,814	115	1,870,406
142	286	28,450,574	143	3,608,086
214	429	97,363,377	212	12,148,666
242	482	140,035,698	241	17,657,286

VI. CONCLUSIONS

In this paper, a new explicit group method was formulated to solve the two dimensional Helmholtz equation. The proposed MEDG SOR method outperforms the original EDG SOR [2] in terms of number of iterations and execution timings. It is also able to maintain almost as good accuracy as the existing point SOR method. Overall, the MEDG SOR serves as viable alternative solver to the Helmholtz equation.

REFERENCES

- [1] A. R. Abdullah, The Four Point Explicit Decoupled Group (EDG) Method : A Fast Poisson Solver. *International Journal of Computer Math.*, vol. 38, 1991, pp.61-70.
- [2] M.K. Akhir, M. Othman, Z. Abdul Majid, and M. Suleiman, Four Point Explicit Decoupled Group Iterative Method Applied to Two-Dimensional Helmholtz Equation. *International Journal of Math. Analysis*, vol. 6, no. 20, 2012, pp. 963-974.
- [3] N. H. M. Ali and A. R. Abdullah, New Rotated Iterative Algorithms For The Solution of A Coupled System of Elliptic Equations, *International Journal of Computer Mathematics*, vol. 72, 1999, pp. 223 - 251.
- [4] N. H. M. Ali and D. J. Evans, Preconditioned Rotated Iterative Methods in the Solution of Elliptic PDE's, *International Journal of Computer Mathematics*, vol. 81, no. 9, 2004, pp. 1163-1174.
- [5] N. H. M. Ali and D. J. Evans, Preconditioned Variational Methods On Rotated Finite Difference Discretisation, *International Journal of Computer Mathematics*, vol. 81, no.10, 2004, pp. 1265-1279.
- [6] N. H. M. Ali, and K. F. Ng, Modified Explicit Decoupled Group Method in The Solution of 2-D Elliptic PDEs. *Proceedings of the 12th WSEAS International Conference on Applied Mathematics, Cairo, Egypt*, p.162-167, 29-31 Dec. 2007.
- [7] G. Dahlquist, and A. Björck, *Numerical methods*. Englewood Cliffs, N.J.: Prentice-Hall, 1974.
- [8] D.J. Evans, Group Explicit Iterative Methods for Solving Large Linear Systems. *International Journal of Computer Math.*, vol. 17, 1985, pp. 81-108.
- [9] W. S. Yousif, and D. J. Evans, Explicit Group Over-relaxation Methods for Solving Elliptic Partial Differential Equations, *Math. Computer Simulation.*, vol. 28, 1986, pp.453-466
- [10] W. S. Yousif and D. J. Evans, Explicit DeCoupled Group Iterative Methods And Their Parallel Implementations, *Parallel Algorithms and Applications*, vol. 7, 1995, pp. 53-71.
- [11] J. Zhang, J. Kouatchou, and L. Ge, A Family of Fourth Order Difference Schemes On Rotated Grid for Two Dimensional Convection-Diffusion Equation, *Mathematics and Computers in Simulation*, vol. 59, 2002, pp. 413-429.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US