A Comparative Study of High Order Rotated Group Iterative Schemes on Helmholtz Equation

Norhashidah Hj. Mohd Ali, Teng Wai Ping

Abstract**—**In this paper, we present a high order group explicit method in solving the two dimensional Helmholtz equation. The presented method is derived from a nine-point fourth order finite difference approximation formula obtained from a 45-degree rotation of the standard grid which makes it possible for the construction of iterative procedure with reduced complexity. The developed method will be compared with the existing group iterative schemes available in literature in terms of computational time, iteration counts, and computational complexity. The comparative performances of the methods will be discussed and reported.

Keywords**—**Explicit group method, finite difference, Helmholtz equation, rotated grid, standard grid.

I. Introduction

The Helmholtz equation is an equation of the elliptic type which describes many physical phenomena in science and engineering. Over the last few years, there has been some considerable interest in the development of numerical solutions for the two dimensional Helmholtz [1]–[5], [7]. A half-sweep point iterative method was derived in solving the equation where this method was found to converge faster than the normal full-sweep iterative scheme [2]. The former method uses a skewed difference formula which leads to lower computational complexities since the iterative procedure needs only involved nodes on half of the total grid points in the solution domain and thus reduces the computational complexity. The applications of group strategies were observed in [1], [3] to solve the same equation where these new methods were proven to have even better convergence rates than the point scheme in [2]. All these developed schemes are of second order accuracy and may not be accurate enough if the mesh is not sufficiently refined. In an effort to improve the accuracy of the solutions, Ali and Teng [4] introduced a fourth order group iterative method in solving the two dimensional Helmholtz by applying the nine-point compact finite difference formula in a specific group construction of the nodal points in the solution domain where the method was shown to have a better rate of convergence than the existing nine-point compact scheme. However, they observe a slightly more CPU times in the explicit group (EG) of $O(h^4)$ due to its larger complexity compared to second order EG [2]. In this work, an attempt is made to construct an improved group scheme which will overcome this high complexity problem while still maintaining the high accuracy property.

In this paper, we propose another group iterative scheme with the same order of accuracy as in [4] but with a much improved computing efforts. The four-point EG method is formulated by using a rotated (or skewed) compact nine-point finite difference formula which enables the construction of schemes that involve only half of the nodal points in the iterative processes. The formulation of the proposed method is presented in Section II, followed by the complexity analysis of the scheme in Section III. Numerical experiments comparing the proposed scheme with the existing group methods of the same class will be presented in Section IV. Conclusions are given in Section V.

II. Group Iterative Methods on Rotated Grid

Consider the solution domain $\Omega=(0,1)\mathrm{x}(0,1)$ with Dirichlet conditions defined at the boundary. Many problems related to the steady-state oscillations (e.g. mechanical, thermal) lead to the two dimensional Helmholtz equation of the following form

$$\nabla^2 u + k^2 u = f(x,y), \quad (x,y)\in\Omega. \tag{1}$$

Here, $\nabla^2 = \frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}$ is the normal Laplacian, *u(x,y)* is the solution, *k* is known as a wave number, *f* is assumed to be sufficiently smooth. The solution domain is discretized uniformly in the *x* and *y* directions with the mesh size *h=1/n*, where $x_i = ih, \quad y_j = jh \ \left(i,j=0,1,2,\cdots,n\right)$. The computed solution is denoted as $u_{ij}= u(x_i,y_j)$. Using the second order centered difference formula on the standard grid, (1) can be approximated as

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

which has a truncation error of O(h^2). We denote the standard grid as Ω_h. Applying this formula to any group of four points in the solution domain, [1] gave rise to the second order EG scheme as:

Norhashidah Hj. Mohd Ali is with the School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia (corresponding author, phone: +604-6533960; fax: +604-6570910; e-mail: shidah_ali@usm.my).

Teng Wai Ping is with HELP University, Pusat Bandar Damansara, Kuala Lumpur (e-mail: wp860415@hotmail.com).

Financial support provided by the Fundamental Research Grant Scheme (203/PMATHS/6711321) is gratefully acknowledged.

$$\begin{bmatrix} u_{i,j} \\ u_{i+1,j} \\ u_{i+1,j+1} \\ u_{i,j+1} \end{bmatrix} = \begin{bmatrix} p(p^2-2) & p^2 & 2p & p^2 \\ p^2 & p(p^2-2) & p^2 & 2p \\ 2p & p^2 & p(p^2-2) & p^2 \\ p^2 & 2p & p^2 & p(p^2-2) \end{bmatrix} \quad (3)$$

$$\text{x}\,\frac{1}{p^2(p^2-4)} \begin{bmatrix} u_{i-1,j} + u_{i,j-1} - h^2 f_{i,j} \\ u_{i+2,j} + u_{i+1,j-1} - h^2 f_{i+1,j} \\ u_{i+2,j+1} + u_{i+1,j+2} - h^2 f_{i+1,j+1} \\ u_{i-1,j+1} + u_{i,j+2} - h^2 f_{i,j+1} \end{bmatrix}$$

where $p = (4 - k^2h^2)$. Using a Taylor series expansion, Ali and Teng [4] derive a fourth order compact approximation to (1) as

$$u_{i+1,j+1} + u_{i+1,j-1} + u_{i,-1j+1} + u_{i-1,j-1} +$$
$$\left(4 + \frac{k^2h^2}{2}\right)(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) + (4k^2h^2 - 20)u_{i,j} \quad (4)$$
$$= \frac{h^2}{2}\left(8f_{i,j} + f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}\right)$$

to come up with the EG of $O(h^4)$ where this method was proven to solve (1) with better accuracy compared to the second order group method introduced in [1]. Due to its larger complexity, it was observed that this method incurred more computational time when the grid sizes get larger compared to the second order EG.

A. Explicit Decoupled Group $O(h^2)$

Another type of finite difference formula that can be used to discretize (1) is the second order rotated five-point formula [6], which will result in

$$u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1} + \left(2k^2h^2 - 4\right)u_{i,j} = 2h^2 f_{i,j} \quad (5)$$

This type of grid is denoted as $\Omega_{\sqrt{2}h}$. Based on this formula, [3] derived the explicit decoupled group (EDG) of $O(h^2)$ where they showed that this group method outperformed its pointwise counterpart in terms of time and iteration counts while maintaining its second order accuracy. Because of this promising result, a higher order accurate solutions derived from the rotated grid $\Omega_{\sqrt{2}h}$ is a worthwhile path to follow.

B. EDG $O(h^4)$

To design a higher order scheme, the Taylor series expansion is used to obtain:

$$\delta_x^2 u_{i,j} = u_{xx} + \frac{h^2}{6}u_{x^4} + \frac{h^4}{90}u_{x^6} + O(h^6) = \left(1 + \frac{h^2}{6}\partial_x^2\right)u_{xx} + O(h^4)$$

Therefore,

$$u_{xx} = \left(1 + \tfrac{h^2}{6}\delta_x^2\right)^{-1}\delta_x^2 u_{i,j} \quad (6)$$

Similarly, we can obtain the formula for

$$u_{yy} = \left(1 + \tfrac{h^2}{6}\delta_y^2\right)^{-1}\delta_y^2 u_{i,j} \quad (7)$$

Applying the rotated formulas (6), (7) to (1), we obtain

$$\left(1 + \tfrac{h^2}{6}\delta_x^2\right)^{-1}\delta_x^2 u + \left(1 + \tfrac{h^2}{6}\delta_y^2\right)^{-1}\delta_y^2 u + k^2 u = f + O(h^4) \quad (8)$$

After some rearrangements, the rotated fourth-order nine-point formula with a truncation error on the order $O(h^4)$ is obtained as [7]

$$\left(u_{i+2,j} + u_{i-2,j} + u_{i,j+2} + u_{i,j-2}\right) + (8k^2h^2 - 20)u_{ij} +$$
$$(4 + k^2h^2)\left(u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1}\right) =$$
$$h^2\left(8f_{i,j} + f_{i+1,j+1} + f_{i+1,j-1} + f_{i-1,j+1} + f_{i-1,j-1}\right) \quad (9)$$

The computational molecule of this approximation is of the following form:

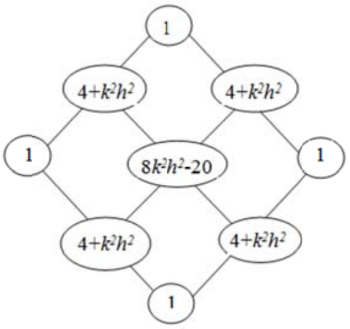

Fig. 1 Computational molecule of approximation (9)

Based on this formula, [7] formulated the pointwise rotated method which is solving (1) by using the multiscale multigrid method combined with the Richardson's extrapolation and obtained promising results. It may be observed that the points involved in this approximation are located on the two diagonal lines from $u(x,y)$, with a mesh spacing of $\sqrt{2}\,h$ between points. Applying (9) to any groups of four points in the solution domain will result in the following (4x4) system of equations:

$$\begin{bmatrix} 20 - 8k^2h^2 & -4 - h^2k^2 & 0 & 0 \\ -4 - h^2k^2 & 20 - 8k^2h^2 & 0 & 0 \\ 0 & 0 & 20 - 8k^2h^2 & -4 - h^2k^2 \\ 0 & 0 & -4 - h^2k^2 & 20 - 8k^2h^2 \end{bmatrix} \begin{bmatrix} u_{ij} \\ u_{i+1,j+1} \\ u_{i+1,j} \\ u_{i,j+1} \end{bmatrix}$$
$$= \begin{bmatrix} S_1 - F_1 \\ S_2 - F_2 \\ S_3 - F_3 \\ S_4 - F_4 \end{bmatrix} + (4 + h^2k^2)\begin{bmatrix} u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1} \\ u_{i+2,j+2} + u_{i+2,j} + u_{i,j+2} \\ u_{i+2,j+1} + u_{i+2,j-1} + u_{i,j-1} \\ u_{i+1,j+2} + u_{i-1,j} + u_{i-1,j+2} \end{bmatrix} \quad (10)$$

where

$$S_1 = u_{i+2,j} + u_{i,j+2} + u_{i-2,j} + u_{i,j-2}$$
$$S_2 = u_{i+3,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i+1,j+3}$$
$$S_3 = u_{i+3,j} + u_{i+1,j+2} + u_{i-1,j} + u_{i+1,j-2}$$
$$S_4 = u_{i+2,j+1} + u_{i,j+3} + u_{i-2,j+1} + u_{i,j-1}$$
$$F_1 = h^2\left(f_{i+1,j+1} + f_{i+1,j-1} + f_{i-1,j+1} + f_{i-1,j-1} + 8f_{i,j}\right)$$
$$F_2 = h^2\left(f_{i+2,j+2} + f_{i+2,j} + f_{i,j} + f_{i,j+2} + 8f_{i+1,j+1}\right)$$

$$F_3 = h^2\left(f_{i+2,j+1} + f_{i+2,j-1} + f_{i,j-1} + f_{i,j+1} + 8f_{i+1,j}\right)$$
$$F_4 = h^2\left(f_{i+1,j+2} + f_{i+1,j} + f_{i-1,j} + f_{i-1,j+2} + 8f_{i,j+1}\right).$$

The system (10) may be transformed into an explicit form by inverting the (4x4) matrix at the left hand side of the system which results in a decoupled form:

$$\begin{bmatrix} u_{ij} \\ u_{i+1,j+1} \end{bmatrix} = \beta \left[\begin{bmatrix} S_1 - F_1 \\ S_2 - F_2 \end{bmatrix} + (4 + h^2k^2) \begin{bmatrix} u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1} \\ u_{i+2,j+2} + u_{i+2,j} + u_{i,j+2} \end{bmatrix} \right] \tag{11}$$

and

$$\begin{bmatrix} u_{i+1,j} \\ u_{i,j+1} \end{bmatrix} = \beta \left[\begin{bmatrix} S_3 - F_3 \\ S_4 - F_4 \end{bmatrix} + (4 + h^2k^2) \begin{bmatrix} u_{i+2,j+1} + u_{i+2,j-1} + u_{i,j-1} \\ u_{i+1,j+2} + u_{i-1,j} + u_{i-1,j+2} \end{bmatrix} \right] \tag{12}$$

where

$$\beta = \frac{1}{384 - 328h^2k^2 + 63h^4k^4} \begin{bmatrix} 20 - 8h^2k^2 & 4 + h^2k^2 \\ 4 + h^2k^2 & 20 - 8h^2k^2 \end{bmatrix}$$

Since it is derived from (8), the truncation error of this scheme is of $O(h^4)$.

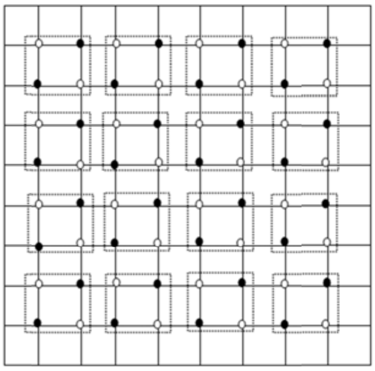

Fig. 2 Groups of two types of points for the EDG $O(h^4)$ method (n=9)

Fig. 2 shows the four points groups that are used in the new method over the whole solution domain for the case n=9. The computational molecule for the evaluation of (11) and (12) are depicted in Figs. 3 (a) and (b) respectively. It may be observed that the evaluation of (11) involves points of type ● only, while (12) can be evaluated involving points of type ○ only. Thus, the calculations of (11) and (12) can be carried out independently. This independency allows us to construct a group scheme where the iterations will involve only one type of points; either the type ● or ○. Without loss of generality, suppose that the iterations are done on the points of type ●. After convergence is achieved, the solution at the other remaining half of the points (of type ○) will be evaluated directly once using the standard formula (2). However, one can also choose to use the formula in (4) for this direct computation to further improve the accuracy of the solutions. Referring to the computational molecule in Fig. 3 and the solution domain in Fig. 2, it is observed that if i = 1 or n – 1, or if j = 1 or n – 1, the formula used in the iterations will involve points which are outside the boundaries of the solution domain. This situation was also encountered in the pointwise rotated method in [7]. To overcome this problem, the rotated five-point formula derived in [7] is used in the evaluation of u at the boundaries.

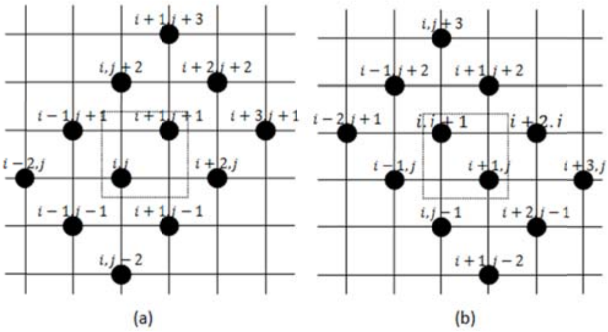

Fig. 3 The computational molecule of EDG $O(h^4)$

The EDG O(h^4) summarized algorithm may then be described as follows:

1. Group the points in the solutions in groups of four as shown in Fig. 2.
2. Perform the iterations on paired points ● in each group using (11) with preferred smoother (e.g. Multigrid, Gauss-Seidel, Successive Over Relaxation, etc.)
3. Check the convergence. If the solutions converge, terminate the iterations. Otherwise, repeat step 2.
4. If it converges, evaluate the solutions at the remaining points ○ using (2).

III. Computational Complexity

Based on the algorithm described in the previous section, the computational complexity of EDG $O(h^4)$ will be calculated in terms of arithmetic operations (+,−,×, ÷) performed in an iteration (excluding the convergence test). The execution time required for each arithmetic operator is assumed to be the same or almost the same. Likewise, the computational complexity for other existing EG methods and the conventional centered difference (CD) scheme are also derived for comparison purposes. Assume that the solution domain is discretized with integer n, then the number of internal mesh points is given by m^2 where m=n-1. There are two main types of internal mesh points namely, iterative points which are points involved in the iteration process, and direct points where solutions are computed directly from a specific formula. If n is even, then there will be ungrouped points near the upper/right boundaries. These types of points will be smoothed by using (9). Table I tabulates the number of arithmetic operations required for the group methods. Here, the symbol $\lceil x \rceil = \min\{n \in \mathbb{Z} | n \geq x\}$, where $\mathbb{Z}$ is the set of integers. Similarly, $\lfloor x \rfloor = max\{n \in \mathbb{Z} | n \leq x\}$.

TABLE I
OPERATION COUNTS OF THE TESTED METHODS

Method	Arithmetic operations in each iteration		After convergence	
	$+/-$	$\times/\div$	$+/-$	$\times/\div$
CD	$4\times m^2$	m^2		
EG $O(h^2)$	$4\times\left[m^2-4\times\left\lfloor\frac{m}{2}\right\rfloor^2\right]$ $+11\times\left\lfloor\frac{m}{2}\right\rfloor^2$	$\left[m^2-4\times\left\lfloor\frac{m}{2}\right\rfloor^2\right]$ $+5\times\left\lfloor\frac{m}{2}\right\rfloor^2$	_	_
EG $O(h^4)$	$8\times\left[m^2-4\times\left\lfloor\frac{m}{2}\right\rfloor^2\right]$ $+23\times\left\lfloor\frac{m}{2}\right\rfloor^2$	$2\times\left[m^2-4\times\left\lfloor\frac{m}{2}\right\rfloor^2\right]$ $+9\times\left\lfloor\frac{m}{2}\right\rfloor^2$	_	_
EDG $O(h^2)$	$7\times\left\lfloor\frac{m}{2}\right\rfloor^2$ $+4\times(\left\lceil\frac{m^2}{2}\right\rceil-2$ $\times\left\lfloor\frac{m}{2}\right\rfloor^2)$	$2\times\left\lfloor\frac{m}{2}\right\rfloor^2$ $+(\left\lceil\frac{m^2}{2}\right\rceil-2\times\left\lfloor\frac{m}{2}\right\rfloor^2)$	$4\times$ $\left\lfloor\frac{m^2}{2}\right\rfloor$	$\left\lceil\frac{m^2}{2}\right\rceil$
EDG $O(h^4)$	$15\times(\left\lfloor\frac{m}{2}\right\rfloor^2-(m-2))$ $+7\times(m-2)$ $+4\times(\left\lceil\frac{m^2}{2}\right\rceil-2\times\left\lfloor\frac{m}{2}\right\rfloor^2)$	$8\times(\left\lfloor\frac{m}{2}\right\rfloor^2-(m-2))$ $+2\times(m-2)$ $+(\left\lceil\frac{m^2}{2}\right\rceil-2\times\left\lfloor\frac{m}{2}\right\rfloor^2)$	$4\times$ $\left\lfloor\frac{m^2}{2}\right\rfloor$	$\left\lceil\frac{m^2}{2}\right\rceil$

IV. NUMERICAL EXPERIMENTS

In order to compare the performances of the proposed method, we conduct several numerical experiments in solving model problem (1) with

$$f(x,y)=(k^2-2\pi^2)\sin(\pi x)\sin(\pi y),\ (x,y)\in\Omega \qquad (13)$$

The solution domain is $\Omega=(0,1)\mathrm{x}(0,1)$ with boundary conditions $u(0,y)=0, u(x,0)=0$ and $u(1,y)=0, u(x,1)=0$. The exact solution can be shown to be [5]

$$u(x,y)=\sin(\pi x)\sin(\pi y) \qquad (14)$$

The execution times (in seconds), the number of iterations (Iter), the maximum absolute errors and the estimated order of accuracy are measured for different grid sizes of 8, 16, 32, 64 and 128. The value of *k* is randomly chosen. The maximum errors are taken as $max\left|u_{i,j}^{(Current\ Iter)}-u_{i,j}^{(Previous\ Iter)}\right|<\varepsilon$. The tolerance used was $\varepsilon=10^{-13}$. The programming language C++ on HP Mini 210-1000 with Windows 7 Starter Edition, processor type is Intel® Atom^{TM} CPU N450 @ 1.66 GHz 1.67 GHz, with installed memory (RAM) of 1GB and 32-bit Operating System type was used throughout the numerical experiments.

To estimate the order of accuracy, we consider two mesh sizes Δ^1 and Δ^2 on Ω^1 and Ω^2 respectively. The estimated order of accuracy for each scheme can be computed for different grid size as [8]

$$Order\,of\ Accuracy=\frac{log\left(\frac{E_1}{E_2}\right)}{log\left(\frac{\Delta^1}{\Delta^2}\right)}. \qquad (15)$$

Here, E_1 and E_2 are the maximum absolute errors of the grids Ω^1 and Ω^2 respectively.

TABLE II
PERFORMANCE OF THE TESTED METHODS

Methods	*n*	*Iter*	*Time (secs)*	*Max error*	*Order of Accuracy*
Centered difference $O(h^2)$	8	70	0.02058	1.6295715e-02	-
	16	198	0.17208	4.0403450e-03	2.0
	32	582	1.67167	1.0080067e-03	2.0
	64	1795	21.045181	2.5187200e-04	2.0
	128	5739	289.70744	6.2959899e-05	2.0
EG $O(h^2)$ [1]	8	51	0.01427	1.6295715e-02	-
	16	130	0.09486	4.0403450e-03	2.0
	32	365	0.99144	1.0080067e-03	2.0
	64	1093	11.64828	2.5187200e-04	2.0
	128	3432	161.12459	6.2959899e-05	2.0
EDG $O(h^2)$ [3]	8	35	0.00716	6.7414825e-02	-
	16	97	0.05167	1.6295715e-02	2.0
	32	277	0.45759	4.0403450e-03	2.0
	64	827	5.43904	1.0080067e-03	2.0
	128	2589	68.20788	2.5187200e-04	2.0
EG $O(h^4)$ [4]	8	57	0.02056	8.2061725e-05	-
	16	129	0.12877	5.1660378e-06	4.0
	32	352	1.27219	3.2344040e-07	4.0
	64	1040	15.28502	2.0223875e-08	4.0
	128	3242	209.53039	1.26432209e-09	4.0
Presented Method EDG $O(h^4)$	8	35	0.00731	5.7194476e-03	-
	16	93	0.06270	2.5809650e-04	4.5
	32	258	0.51509	1.3690194e-05	4.2
	64	760	6.43387	7.8938221e-07	4.1
	128	2350	101.88331	4.7367252e-08	4.1

Table II displays the performances of the implemented numerical schemes. The numerical results obtained by the existing group schemes of the same class, i.e. the EG $O(h^2)$ and EDG $O(h^2)$, which were derived by [1], [3] respectively, and EG $O(h^4)$ [4] are also included in the table. For comparison purposes, the numerical solutions obtained by the conventional second order centered difference scheme as the benchmark solutions are also computed. Fig. 4 depicts the graphical performances of the tested methods in terms of execution timings for different mesh sizes.

V. DISCUSSION OF RESULTS

From Table II, it is clear that between the two group methods which were derived from the rotated grid, EDG $O(h^2)$ requires less computing time than the proposed EDG $O(h^4)$. This is expected since the latter requires more computational effort to produce fourth order accuracy compared to the former which only produces second order accuracy solutions.

Table II shows that among the group methods, the EDG $O(h^2)$ requires the least computing times for each grid size attempted; approximately 65-76% faster than the benchmark centered difference scheme. The new fourth order scheme on the rotated grid, EDG $O(h^4)$, and second order group scheme, EG $O(h^2)$, are about 63-69% and 30-44% faster than the centered difference scheme, respectively. Meanwhile, the fourth order compact scheme on the standard grid Ω_h, EG $O(h^4)$, gives the least improvement in timings with 0.097-27% over the centered point scheme.

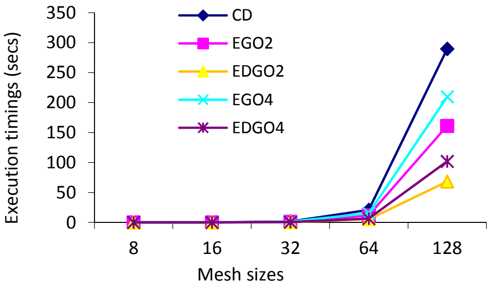

Fig. 4 Execution timings (in secs) for the centered difference (CD) the EG $O(h^2)$ (EGO2), EDG $O(h^2)$ (EDGO2), EG $O(h^4)$ (EGO4), and EDG $O(h^4)$ (EDGO4)

TABLE III
COMPUTING EFFORTS FOR CENTERED DIFFERENCE METHOD

n	Method	
	Centered Five-Point Difference	
	Iter	*Operation count*
8	70	20580
16	198	267300
32	582	3355812
64	1795	42746130
128	5739	555385986

TABLE IV
COMPUTING EFFORTS FOR EG OF ORDER 2 AND EDG OF ORDER 2

n	Methods			
	EG $O(h^2)$		EDG $O(h^2)$	
	Iter	*Operation count*	*Iter*	*Operation count*
8	51	10659	35	4180
16	130	120770	97	50612
32	365	1425325	277	606260
64	1093	17489093	827	7423148
128	3432	222287208	2589	94166004

A summary of the operation counts of the existing methods and the proposed EDG $O(h^4)$ methods are given in Tables III-V. These total operation counts were obtained by combining the results of the experimental number of iterations (shown in Table II) with the theoretical number of operations in each iteration given in Table I. From the experimental results, we can clearly see that the experimental timings of the iterative methods are in agreement with the computing effort analysis.

TABLE V
COMPUTING EFFORTS FOR EG OF ORDER 4 AND EDG OF ORDER 4

n	Methods			
	EG $O(h^4)$ [4]		EDG $O(h^4)$	
	Iter	*Operation count*	*Iter*	*Operation count*
8	57	23826	35	6140
16	129	239682	93	95420
32	352	2749120	258	107320
64	1040	33282080	760	16398560
128	3242	419962196	2350	211944520

In terms of accuracy, the second ordered schemes in general give the least accurate results compared to the schemes of fourth order accuracies. The EG $O(h^4)$ scheme on the standard grid Ω_h gives the best accuracy, followed by the newly formulated EDG $O(h^4)$ which produces relatively comparable accuracy. This is expected since the latter is derived from a formula based on a mesh size $\sqrt{2}\ h$ between points compared to a mesh size h in the former method which incur a very slight decrease in accuracy. However, EDG $O(h^4)$ still manage to produce fourth order accuracy in the solutions. The EDG $O(h^2)$ scheme may give the fastest rate of convergence but it has the least accuracy as n gets larger compared to the other schemes due to its $O(h^2)$ accuracy on the $\Omega_{\sqrt{2}h}$ grid. The EG $O(h^4)$ method, which is the fourth order scheme on the standard grid Ω_h, even though give the best results in terms of accuracy but requires the longest time of execution due to its higher operation counts. However, the newly formulated EDG $O(h^4)$ manage to narrow the gap in the duration of execution time by requiring lesser execution timings than EG $O(h^2)$ but producing higher order accuracy solutions close to the ones obtained by EG $O(h^4)$. In summary, the proposed fourth order scheme on the rotated grid $\Omega_{\sqrt{2}h}$ has proven to be a relatively computationally efficient alternative solver to the Helmholtz equation with high accuracy solution and a fast rate of convergence.

VI. CONCLUSION

A new fourth order iterative method derived from the rotated nine-point discretisation formula, used in conjunction with a specific grouping strategy, is presented in solving the two dimensional Helmholtz equation. Our results indicated that the rotated group scheme, EDG $O(h^4)$, can reduce the computation times quite significantly compared to the fourth order group scheme, EG $O(h^4)$, introduced in s[4]. We have also shown that the proposed group method does give high accuracy numerical solution for the model test problem. The computed results compare well with the solutions obtained from the common existing finite discretisation schemes. It would be worthwhile to investigate the application of this method to other types of partial differential equations such as the time-dependent parabolic equations.

REFERENCES

[1] M.K. Akhir, M. Othman, J. Sulaiman, Z.A. Majid, and M. Suleiman, Numerical solutions of Helmholtz using a new four-point EGMSOR

iterative method equations, *Appl. Math. Science*, Vol. 80, 2011, pp. 3991-4004.

[2] M.K. Akhir, M. Othman, J. Sulaiman, Z.A. Majid, M. Suleiman, Half-sweep iterative method for solving 2D Helmholtz equations, *International Journal of Applied Mathematics & Statistics*, Vol. 29, No. 5, 2012, pp. 101-109.

[3] 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.

[4] N. H. M. Ali and W. P. Teng, New Fourth Order Explicit Group Method in the Solution of the Helmholtz Equation, World Academy of Science, Engineering and Technology, *International Journal of Mathematical, Computational, Physical and Quantum Engineering*, Vol. 9 no. 1, 2015, pp. 7-11.

[5] M. Nabavi, M. H. K. Siddiqui, J. Dargahi, A new 9-point sixth order accurate compact finite difference method for the Helmholtz equation, *Journal of Sound and Vibration*, Vol. 307, 2007, pp. 972-982.

[6] Y. Saad, Iterative Methods for Sparse Linear Systems, second edition, PWS Publishing Company, Boston, 2000.

[7] W. P. Teng and N. H. M. Ali, Higher Order Rotated Iterative Scheme for the 2D Helmholtz Equation, Proceedings of the 21st National Symposium on Mathematical Sciences (SKSM21): Germination of Mathematical Sciences Education and Research towards Global Sustainability, *AIP Proceedings Volume 1605*, 2014, pp. 155-160.

[8] Wang, and J. Zhang, “Sixth order compact scheme combined with multigrid method and extrapolation technique for 2D Poisson equation”, *Journal of Computation Physics*, 228, 2009, pp. 137-146.