Group Explicit Multigrid Solutions Of A New Navier-Stokes Solver

27911

*Norhashidah Hj. Mohd. Ali and Lim Lay Ngor

27911

***Abstract*— A new group explicit method, the Explicit Decoupled Group (EDG) method, derived from the *rotated* finite discretisation formula [8], was formulated in solving a coupled system of elliptic equations which represents the two-dimensional steady state Navier-Stokes equations. The method was found to be more efficient compared to the iterative schemes based on the centred five-point difference formulae and the Alternating Group Explicit scheme due to Sahimi and Evans ([2], [4], [5], [12]). However, the new scheme gets to be divergent for very large *Reynold numbers* due to the difficulty with the traditional SuccessiveOverRelaxation (SOR)-type iterative procedure. In this work, we apply the multigrid technique to this new group iterative scheme in solving the steady-state Navier-Stokes equation as a way to further improve the performance of this method. Several experimental work comparing its performance with other schemes based on the common *centred* difference formula will be reported.**

***Index terms*—Explicit Decoupled Group method, multigrid, Navier-Stokes equations**

I. INTRODUCTION

Consider the following coupled system of partial differential equations:

$$\nabla^2\psi = -\omega \qquad (1)$$
$$\nabla^2\omega + \mathrm{Re}(\psi_x\omega_y - \psi_y\omega_x) = -c \qquad (2)$$

where $x,y \in \Omega$ = (0,L)x(0,L) with a set of conditions for ψ and ω prescribed at the boundary. Here, c and Re (the Reynolds number) are non-negative constants and $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the usual Laplacian operator. Note that if Re $\neq$ 0, then the coupled system represents the two dimensional steady state Navier-Stokes equations which describe the basic viscous, incompressible flow problems. ψ and ω are known respectively as the stream and vorticity functions. Suppose we impose the boundary conditions $\psi = 0$ and $\frac{\partial^2\psi}{\partial\eta^2} = 0$, where η is the normal to the boundary $\partial\Omega$ of Ω, then our problem amounts to solving (1) and (2) successively with $\psi = 0$ and $\omega = 0$ respectively along $\partial\Omega$. The aim of this paper is to investigate the versatility of the four-point EDG method by incoorporating the multigrid technique into the scheme, in solving this fundamental problem in fluid dynamics. A brief discussion of the finite difference approximations for (1)-(2) with the specified boundary conditions will be given in Section II. In Section III, the development of the EDG scheme for the vorticity equation (2) will be presented. The derivation of the algorithm for the stream solutions will then readily follow. Section IV describes the multigrid algorithm for solving the coupled system (1)-(2) by incorporating the four-point EDG in its iteration scheme, followed by the numerical experiments Section V, and the concluding remarks is presented in Section VI.

** The authors are staff of School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia.*

II. FINITE DIFFERENCE APPROXIMATIONS

Let us assume that a rectangular grid in the (x,y)-plane with grid spacing h = L/n in both directions and x_i = ih, y_j = jh , i,j = 0,1,...,n are used. Observe that if ω is known, then (1) is a linear *elliptic* equation in ψ, and if ψ is known, then (2) is a linear *elliptic* equation in ω. Suppose $\psi^{(0)}$ and $\omega^{(0)}$ are the initial guesses, we can use the $\omega^{(0)}$ in (1) to produce $\psi^{(1)}$. Use this $\psi^{(1)}$ in (2) to produce $\omega^{(1)}$. Then use this $\omega^{(1)}$ in (1) to produce $\psi^{(2)}$, and this $\psi^{(2)}$ to produce $\omega^{(2)}$ and so on. This indicates that at the grid point (x_i, y_j) the following alternating sequences of outer iterates can be generated [9]:

$$\begin{tikzcd}
\psi_{ij}^{(0)} & \psi_{ij}^{(1)} \arrow[d] & \psi_{ij}^{(2)} \arrow[d] & \psi_{ij}^{(3)} \arrow[d] & \psi_{ij}^{(4)} \arrow[d] & \cdots & \psi_{ij}^{(k)} \arrow[d] & \cdots \\
\omega_{ij}^{(0)} \arrow[ur] & \omega_{ij}^{(1)} \arrow[ur] & \omega_{ij}^{(2)} \arrow[ur] & \omega_{ij}^{(3)} \arrow[ur] & \omega_{ij}^{(4)} & \cdots & \omega_{ij}^{(k)} & \cdots
\end{tikzcd}$$

Fig. 1 : Generation of outer iterates

The finite difference approximations of equations (1) and (2) using the *centred difference* formula at the point (x_i, y_j) will result in the following:

$$-\psi_{i-1,j}^{(k+1)} - \psi_{i,j-1}^{(k+1)} + 4\psi_{ij}^{(k+1)} - \psi_{i,j+1}^{(k+1)} - \psi_{i+1,j}^{(k+1)} = h^2\omega_{ij}^{(k)} \qquad (3)$$

$$\begin{aligned}&-\left[1-\sigma\left(\psi_{i,j-1}^{(k+1)} - \psi_{i,j+1}^{(k+1)}\right)\right]\omega_{i-1,j}^{(k+1)} - \left[1+\sigma\left(\psi_{i-1,j}^{(k+1)} - \psi_{i+1,j}^{(k+1)}\right)\right]\omega_{i,j-1}^{(k+1)} + 4\omega_{ij}^{(k+1)} \qquad (4)\\ &-\left[1-\sigma\left(\psi_{i-1,j}^{(k+1)} - \psi_{i+1,j}^{(k+1)}\right)\right]\omega_{i,j+1}^{(k+1)} - \left[1+\sigma\left(\psi_{i,j-1}^{(k+1)} - \psi_{i,j+1}^{(k+1)}\right)\right]\omega_{i+1,j}^{(k+1)} = h^2 c_{ij}^{(k)},\end{aligned}$$

here σ= Re/4 and i,j =1,2,...,n-1. Another type of approximation that can represent the differential equations (1) and (2) is the cross orientation [6] which can be obtained by rotating the i-plane axis and the j-plane axis clockwise by 45°. With this displacement, equations (3) and (4) become (5) and (6) respectively:

$$-\psi_{i-1,j+1}^{(k+1)} - \psi_{i-1,j-1}^{(k+1)} + 4\psi_{ij}^{(k+1)} - \psi_{i+1,j+1}^{(k+1)} - \psi_{i+1,j-1}^{(k+1)} = 2h^2\omega_{ij}^{(k)} \qquad (5)$$

$$\begin{aligned}&-[1-\sigma(\psi_{i-1,j-1}^{(k+1)} - \psi_{i+1,j+1}^{(k+1)})]\omega_{i-1,j+1}^{(k+1)} - [1+\sigma(\psi_{i-1,j+1}^{(k+1)} - \psi_{i+1,j-1}^{(k+1)})]\omega_{i-1,j-1}^{(k+1)} + 4\omega_{ij}^{(k+1)}\\ &-[1-\sigma(\psi_{i-1,j+1}^{(k+1)} - \psi_{i+1,j-1}^{(k+1)})]\omega_{i+1,j+1}^{(k+1)} - [1+\sigma(\psi_{i-1,j-1}^{(k+1)} - \psi_{i+1,j+1}^{(k+1)})]\omega_{i+1,j-1}^{(k+1)} = 2h^2 c_{ij}^{(k)}.\end{aligned} \qquad (6)$$

Clearly it can be seen that the application of (5)-(6) will result in a large and sparse system with the coefficient matrix being a block matrix depending on the ordering of points taken.

III. THE FOUR POINT EDG FORMULATION

Similar to the single elliptic case [1], we assume that the solution at any group of four points on the solution domain is solved using the *rotated* equation (6). This will result in a (4x4) system of equations

$$\begin{bmatrix} 4 & -[1-\sigma(\psi_{i-1,j+1}-\psi_{i+1,j-1})] & 0 & 0\\ -[1+\sigma(\psi_{i,j+2}-\psi_{i+2,j})] & 4 & 0 & 0\\ 0 & 0 & 4 & -[1-\sigma(\psi_{i,j-1}-\psi_{i+2,j+1})]\\ 0 & 0 & -[1+\sigma(\psi_{i-1,j}-\psi_{i+1,j+2})] & 4\end{bmatrix}\begin{bmatrix}\omega_{ij}\\ \omega_{i+1,j+1}\\ \omega_{i+1,j}\\ \omega_{i,j+1}\end{bmatrix}$$

$$= \begin{bmatrix} [1-\sigma(\psi_{i-1,j-1}-\psi_{i+1,j+1})]\omega_{i-1,j+1} + [1+\sigma(\psi_{i-1,j+1}-\psi_{i+1,j-1})]\omega_{i-1,j-1} + [1+\sigma(\psi_{i-1,j-1}-\psi_{i+1,j+1})]\omega_{i+1,j-1} + 2h^2c_{ij}\\ [1-\sigma(\psi_{ij}-\psi_{i+2,j+2})]\omega_{i,j+2} + [1-\sigma(\psi_{i,j+2}-\psi_{i+2,j})]\omega_{i+2,j+2} + [1+\sigma(\psi_{ij}-\psi_{i+2,j+2})]\omega_{i+2,j} + 2h^2c_{i+1,j+1}\\ [1-\sigma(\psi_{i,j+1}-\psi_{i+2,j-1})]\omega_{i+2,j+1} + [1+\sigma(\psi_{i,j+1}-\psi_{i+2,j-1})]\omega_{i,j-1} + [1+\sigma(\psi_{i,j-1}-\psi_{i+2,j+1})]\omega_{i+2,j-1} + 2h^2c_{i+1,j}\\ [1-\sigma(\psi_{i-1,j+2}-\psi_{i+1,j})]\omega_{i+1,j+2} + [1+\sigma(\psi_{i-1,j+2}-\psi_{i+1,j})]\omega_{i-1,j} + [1-\sigma(\psi_{i-1,j}-\psi_{i+1,j+2})]\omega_{i-1,j+2} + 2h^2c_{i,j+1}\end{bmatrix}$$

$$= \begin{bmatrix}rhs_{i,j}\\ rhs_{i+1,j+1}\\ rhs_{i+1,j}\\ rhs_{i,j+1}\end{bmatrix},$$

which leads to a decoupled system of (2x2) equations whose explicit forms are given by

$$\begin{bmatrix}\tilde{\omega}_{ij}\\ \tilde{\omega}_{i+1,j+1}\end{bmatrix} = \frac{1}{16-[(1-\sigma(\psi_{i-1,j+1}-\psi_{i+1,j-1}))(1+\sigma(\psi_{i,j+2}-\psi_{i+2,j}))]} \times \begin{bmatrix}4 & 1-\sigma(\psi_{i-1,j+1}-\psi_{i+1,j-1})\\ 1+\sigma(\psi_{i,j+2}-\psi_{i+2,j}) & 4\end{bmatrix}\begin{bmatrix}rhs_{ij}\\ rhs_{i+1,j+1}\end{bmatrix} \qquad \text{............(7)}$$

and

$$\begin{bmatrix}\tilde{\omega}_{i+1,j}\\ \tilde{\omega}_{i,j+1}\end{bmatrix} = \frac{1}{16-[(1-\sigma(\psi_{i,j-1}-\psi_{i+2,j+1}))(1+\sigma(\psi_{i-1,j}-\psi_{i+1,j+2}))]} \times \begin{bmatrix}4 & 1-\sigma(\psi_{i,j-1}-\psi_{i+2,j+1})\\ 1+\sigma(\psi_{i-1,j}-\psi_{i+1,j+2}) & 4\end{bmatrix}\begin{bmatrix}rhs_{i+1,j}\\ rhs_{i,j+1}\end{bmatrix}. \qquad \text{............(8)}$$

The computational molecule of Eq. (7) and (8) are given in Figs. 2 and Fig. 3 respectively:

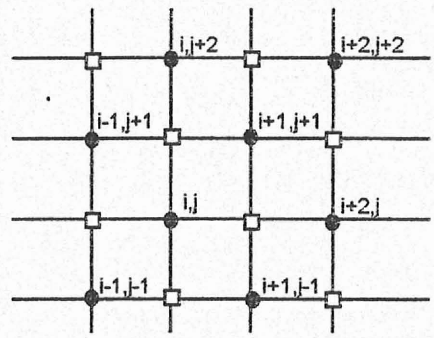

Fig. 2 : Computational molecule of Eq. (7)

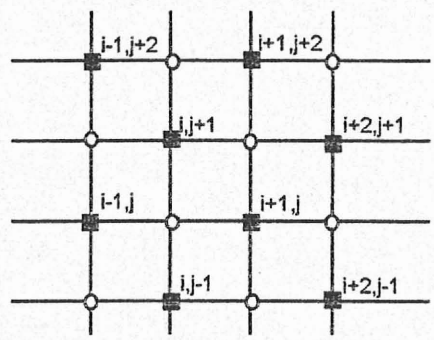

Fig. 3 : Computational molecule of Eq. (8)

Note that for both equations, iterative evaluation of points from each group requires contribution of points only from the same group. This means the iteration of points for the vorticity solutions from Eq. (7) can be carried out by only involving points of type ● only, while the iterations arised from Eq. (8) can be implemented by involving points of type ■ only. Due to this independency, the iterations can be carried out on either one of the two type of points, which means we can expect the execution time to be reduced by nearly *half* since iterations are done on only about *half* of the total nodal points.

In summary, the four-point EDG scheme corresponds to iterating the solutions at approximately *half* of the points in the solution domain using either (7) or (8) until convergence is achieved, i.e., when $\left|\omega_{ij}^{(k+1)} - \omega_{ij}^{(k)}\right| \le \varepsilon$;

here ε is the convergence criterion used. If convergence is achieved, evaluate the solutions at the rest of the nodal points *(points of opposite type)* using the *centred difference* formula (4). Otherwise, repeat the iteration cycle. In the case of n even, the EDGR scheme is adopted, i.e., we assume the uncoupled points are on the right most and top most grid lines. Suppose Eq. (7) is chosen to be used in the iterative evaluation of points, these uncoupled points must be calculated after the iterations on the points ● have converged. The uncoupled points of the same type are calculated using the *rotated* formula of Eq. (6). Only after these points have been calculated, the remaining points of the uncoupled group (of the opposite type) are evaluated using the *centred difference* formula (4).

With boundary conditions specified as before, an algorithm can now be formulated to solve the coupled system (1) and (2):

Step 1 Choose h and construct the number of nodal points as usual for an elliptic problem. Set $\psi_{ij}^{(0)} = \omega_{ij}^{(0)} = 0 = \text{outer\_}\psi_{ij}^{(0)} = \text{outer\_}\omega_{ij}^{(0)}$ as initial approximations for the outer iteration.

Step 2 Generate sequences $\psi^{(k+1)}$ and $\omega^{(k+1)}$ on Ω by the alternating procedure described before for k = 0,1,2,...

Solve the stream-function equation (5) by performing the four-point EDG iterative procedure for a prescribed tolerance ε. (Use the same Eq. (6) but replace ω with ψ, c_{ij} with ω_{ij}, and $\sigma = 0$.)

Solve the vorticity equation (6) by performing the four-point EDG iterative procedure for a prescribed tolerance ε. (Here, use the converged stream-function just obtained previously in the place of ψ, and $\sigma = \text{Re}/4$.)

Store the converged values in $\text{outer\_}\psi_{ij}^{(m)}$ and in $\text{outer\_}\omega_{ij}^{(m)}$ respectively.

Step 3 Check the convergence of the outer iteration process over the whole mesh points for a prescribed convergence criterion δ, i.e., check whether the following condition is achieved,

$$\max\left\{\left|\text{outer\_}\psi_{ij}^{(m+1)} - \text{outer\_}\psi_{ij}^{(m)}\right|, \left|\text{outer\_}\omega_{ij}^{(m+1)} - \text{outer\_}\omega_{ij}^{(m)}\right|\right\} \le \delta .$$

If convergence is achieved, the numerical solution of the given problem is given by the generated $\text{outer\_}\psi_{ij}^{(m+1)}$ and $\text{outer\_}\omega_{ij}^{(m+1)}$. Otherwise, go back to *Step 2*.

IV. MULTIGRID IMPLEMENTATION

A. Multigrid EDG Smoother On Elliptic Equation

Since the work of Hackbusch [10], multigrid methods have been widely applied to the numerical solution of partial differential equations. The application of multigrid EDG technique on Poisson problem was introduced by Mohamed and Abdullah [11]. All the mesh points in solution domain Ω^h are labeled in red O and black □ points as in Fig. 4(a) and (b) respectively. For the case when n is even (for example n = 8), the red points group next to the boundaries are further divided into two groups i.e. red points labeled ● and O.

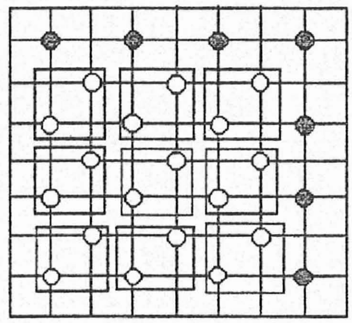

Fig. 4a The circled (red) points

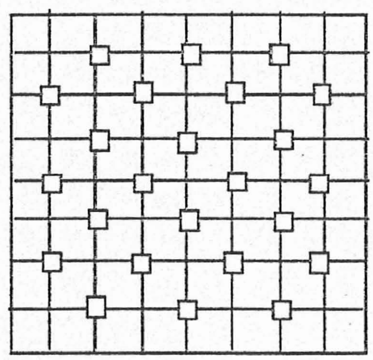

Fig. 4b The square (black) points

The red points (points ● and O) are iterated until they converge, based on a certain convergence criteria. The paired red points of type O will be iterated using the stencil (7) whilst the red points of type ● (the red points next to the boundaries) will be iterated using the *rotated difference* stencil (6). After convergence is achieved, the solutions at the black points (□) will be evaluated directly using the *centred difference* stencil (4).

This means that in using the multigrid EDG method, only the circled mesh points (points ● and O) will undergo the process of iterative evaluation using either the Equation (7) or (6). One observation is that the iterative evaluation on the circled points will require points of the same type. The same goes to points of type

□ (black points). Therefore the iteration over the domain Ω^h can be carried out on either type of points only (either O or □ only). The iterative process consists of different levels of grids; a process that goes from the finest grid down to the coarsest grid and back from the coarsest grid up to the finest.

The EDG multigrid algorithm will involve the basic element of grid transfer and the iterative method for smoothing the errors or residuals. The residual in the domain Ω^h is defined to be

$$r^h = A^h v^h - f^h \qquad (9)$$

The residuals evaluated on the red points at each levels are transferred into the respective red points at the coarser grid using the restricting operator $\Re_h^{2h} : \Omega^h \rightarrow \Omega^{2h}$ defined as

$$\Re_h^{2h} = \frac{1}{8}\begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 0 \\ 1 & 0 & 1 \end{bmatrix} \qquad (10)$$

At the coarser grid, the new residual is defined as

$$r^{2h} = \Re_h^{2h} r^h . \qquad (11)$$

At the coarser grid, a new linear system is established. For example,

$$A^{2h} e^{2h} = r^{2h},$$

e^{2h} is the error value found by using the Gauss-Seidel method as the error smoother, in order to get a better error estimation. It is very important that the residuals are well smoothed before being transferred to the coarser grids. This process will be continued until the coarsest grid is reached. The error estimation value is acquired by solving the resulted equation at the coarsest grid.

On the other hand, the linear prolongation is used to transfer the red points from the coarser grids to the red points at the finer grids using the following prolongation operator $P_{2h}^{h} : \Omega^{2h} \rightarrow \Omega^h$ defined as

$$v^h_{2i,2j} = v^{2h}_{i,j} \qquad (12)$$

for all *i*, *j* both even or odd.

While the bilinear interpolation is applied to interpolate the red points on the fine grid as follows,

$$v^h_{4i-2,4j} = \frac{1}{2}\left(v^{2h}_{4i-2,4j-2} + v^{2h}_{4i-2,4j+2}\right), \qquad (13)$$

$$v^h_{i,2j-1} = \frac{1}{2}\left(v^{2h}_{i,j} + v^{2h}_{i,j-1}\right), \qquad (14)$$

for all $i,j = 1,2,\ldots,N-1$,

$$\text{and } v^h_{i,j} = \frac{1}{4}\left(v^{2h}_{i-1,j-1} + v^{2h}_{i+1,j-1} + v^{2h}_{i-1,j+1} + v^{2h}_{i+1,j+1}\right)$$

for all i,j odd. (15)

At the finest grid, we may use the correction $v^h \leftarrow v^h + P_{2h}^{h} v^{2h}$ to improve the rate of convergence. The blocks of circled points at each level would be applied with the EDG smoothing scheme where the points will be iterated till they converged. After convergence is achieved, the solutions at the rest of the points (the square points) will be evaluated directly once using the *centred difference* stencil (4).

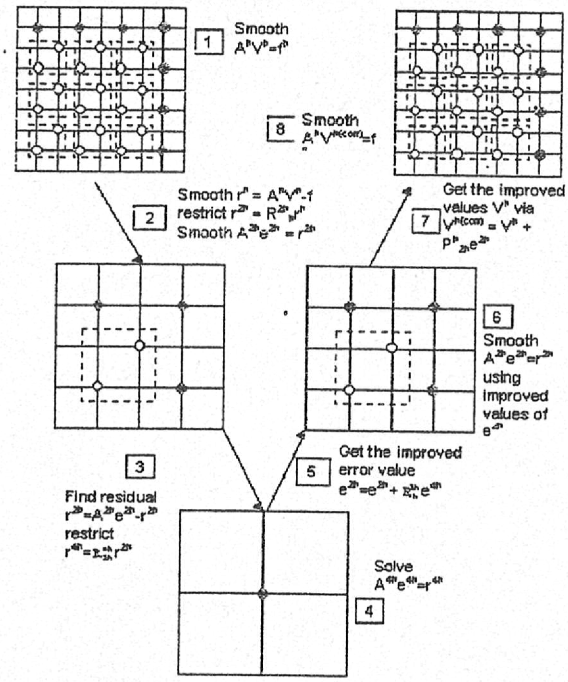

Fig. 5. The structure of halfsweep EDG multigrid method with V-cycle on the circled points

The general algorithm for the EDG multigrid method for an elliptic problem may be described as follows:

BLACK(A^h,v^h,f^h)

{

/* Evaluate only once on the black points */

Smooth $A^h v^h = f^h$ using Gauss-Seidel scheme on the *centred difference* stencil (4).

}

MULTIGRID(i,A^h,v^h,f^h)

{ /* Iterate on the red points until converge */

IF (i = 0) coarsest grid, solve $A^h e^h = r^h$ directly

ELSE

{

Smooth v_1 times on $A^h v^h = f^h$ applying Gauss Seidel at the finest domain using stencil (6) (or (5)).

Compute residuals $r^h \leftarrow f^h - A^h v^h$

Set $e^{2h} \leftarrow 0$ and restrict $r^{2h} \leftarrow R_h^{2h} r^h$.

Get $e^{2h} \leftarrow$ MULTIGRID (i -1,A^{2h},e^{2h},r^{2h}).

Correct errors and transfer to finer grid $v^h \leftarrow v^h + P_{2h}^{h} e^{2h}$

Smooth v_2 times on $A^h v^h = f^h$ applying Gauss Seidel on Ω^h using the stencil (6) (or (5)). } }

```
HALFSWEEP_EDG_MULTIGRID Algorithm()
{
Flag = 0
WHILE (Flag != 1) DO
{
Flag = 1
MULTIGRID(i,A^h,v^h,f^h)
IF |v_ij^(k+1) - v_ij^(k)| > ε on the red points, set Flag = 0
Iterate ++
Swap v_ij^(k) ← v_ij^(k+1) for all red points
}
BLACK(A^h,v^h,f^h)
Return v^h as an approximate solution
}
```

B. *Multigrid EDG Smoother On Navier-Stokes*

An inner-outer iteration procedure for the EDG multigrid solver may be described as the following:

Step 1 Choose h and set $\psi_{ij}^{(0)} = \omega_{ij}^{(0)} = 0 = outer\_\psi_{ij}^{(0)} = outer\_\omega_{ij}^{(0)}$ as initial approximations for the outer iteration.

Step 2 For k = 0,1,2,...
Solve the stream-function equation (5) by performing one multigrid four-point EDG V(1,1)-cycle iterative procedure for a prescribed tolerance ε. (Use the same Eq. (6) but replace ω with ψ, c_{ij} with ω_{ij}, and $\sigma = 0$.)
Solve the vorticity equation (6) by performing one multigrid four-point EDG V(1,1)-cycle iterative procedure for a prescribed tolerance ε .(Here, use the converged stream-function just obtained previously in the place of ψ, and $\sigma = Re/4$.)

Store the converged stream-function values $\psi_{ij}^{(k+1)}$ in $outer\_\psi_{ij}^{(m)}$, and vorticity-function values $\omega_{ij}^{(k+1)}$ in $outer\_\omega_{ij}^{(m)}$.

Step 3 Check the convergence, if both differences of the current and previous values of the converged stream-function and vorticity are less than a prescribed tolerance, then stop and the solutions are stored as $outer\_\psi_{ij}^{(m+1)}$ and $outer\_\omega_{ij}^{(m+1)}$. Otherwise, go back to ***Step 2.***

V. NUMERICAL EXPERIMENT AND RESULTS

Numerical experiments have been carried out using the multigrid algorithm described previously to solve the following Navier-Stokes equations ([2], [4], [5], [12]),

$$\nabla^2\psi = -\omega \qquad (16)$$
$$\nabla^2\omega + Re(\psi_x\omega_y - \psi_y\omega_x) = -1 \qquad (17)$$

with boundary conditions

$$\begin{aligned}\psi(x,0) &= \psi(x,1) = \omega(x,0) = \omega(x,1) = 0, \quad 0 \le x \le 1,\\ \psi(0,y) &= \psi(1,y) = \omega(0,y) = \omega(1,y) = 0, \quad 0 \le y \le 1.\end{aligned} \qquad (18)$$

The problem was solved for various values of Reynolds number Re ≥ 1. Throughout the experiment, the algorithm was executed using C++ programming language on different size grids of $\Omega^h, \Omega^{2h}, \ldots, \Omega^{128h}$. The methods were terminated when the mesh points at the finest grid achieve convergence with tolerance $\delta = \varepsilon = 1.0\text{x}10^{-11}$ for both the outer and inner iterations.

TABLES 1 and 2 list the iteration counts and timings for the EDG method, with and without multigrid, for selected Re = 1 and 1000 respectively. The final computed values of ψ and ω when grid size is h = 1/8 for selected values of x and y for Re = 1 and 1000 are shown in TABLES 3 and 4 respectively. The numerical solutions of the problem for Re = 1 and 1000 are displayed for x = 0.125(0.125)0.875, y = 0.125(0.125)0.875.

TABLE 1. The experimental results for the EDG iterative schemes (Re=1)

	Halfsweep Four Point EDG Multigrid Method				Four Point EDG Scheme Without Multigrid Method			
Grid size	Time (secs)	Number of outer iteration	Number of inner iteration for ψ	Number of inner iteration for ω	Time (secs)	Number of outer iteration	Number of inner iteration for ψ	Number of inner iteration for ω
8	3.46	1	1	7	2.74	1	1	49
		2	6	5		2	42	21
		3	4	3		3	12	3
		4	1	1		4	1	1
16	4.01	1	1	8	3.4	1	1	171
		2	6	5		2	142	45
		3	3	2		3	17	3
		4	1	1		4	1	1
32	4.12	1	1	8	4.95	1	1	596
		2	6	5		2	480	64
		3	3	1		3	22	1
		4	1	1		4	1	1
64	8.13	1	1	8	10.38	1	1	2053
		2	6	4		2	1589	92
		3	2	1		3	10	1
		4	1	1		4	1	1
128	6.75	1	1	7	116.44	1	1	6906
		2	6	3		2	5049	71
		3	1	1		3	3	1
						4	1	1

TABLE 2. The experimental results for the EDG iterative schemes (Re=1000)

Grid size	Halfsweep Four Point EDG Multigrid Method				Four Point EDG Scheme Without Multigrid Method			
	Time (secs)	Number of outer iteration	Number of inner iteration for ψ	Number of inner iteration for ω	Time (secs)	Number of outer iteration	Number of inner iteration for ψ	Number of inner iteration for ω
8	4.78	1	1	7	4.29	1	1	49
		2	6	9		2	42	38
		3	6	8		3	35	33
		4	5	7		4	29	24
		5	4	6		5	19	21
		6	3	5		6	15	16
		7	3	4		7	9	10
		8	2	3		8	3	5
		9	1	1		9	1	2
						10	1	1
16	5.6	1	1	8	5.66	1	1	171
		2	6	9		2	142	126
		3	6	7		3	95	87
		4	5	6		4	54	55
		5	4	5		5	28	28
		6	2	3		6	5	8
		7	1	2		7	1	3
		8	1	1		8	1	3
						9	1	1
32	5.95	1	1	8	6.26	1	1	596
		2	6	7		2	480	318
		3	5	6		3	191	130
		4	4	4		4	46	28
		5	2	2		5	6	7
		6	1	1		6	1	1
64	6.26	1	1	8	1093	1	1	2058
		2	6	6		2	1589	490
		3	4	4		3	198	95
		4	2	2		4	9	9
		5	1	1		5	1	3
						6	1	1
128	8.35	1	1	7	12302	1	1	6906
		2	6	6		2	5049	774
		3	4	3		3	197	38
		4	1	1		4	1	3
						5	1	1

TABLE 3. Numerical solutions of the Navier-Stokes equations for Re = 1 (in the form a(b)=a×10^b) (EDG with Multigrid)

Stream solutions, ψ						
0.731561(-3)	0.117226(-2)	0.164067(-2)	0.161444(-2)	0.164053(-2)	0.117216(-2)	0.73149(-3)
0.117223(-2)	0.231682(-2)	0.273185(-2)	0.317656(-2)	0.273174(-2)	0.231663(-2)	0.117215(-2)
0.164055(-2)	0.273177(-2)	0.379335(-2)	0.378679(-2)	0.379323(-2)	0.273167(-2)	0.164047(-2)
0.161432(-2)	0.317636(-2)	0.378667(-2)	0.438401(-2)	0.378667(-2)	0.317636(-2)	0.161432(-2)
0.164036(-2)	0.27315(-2)	0.379296(-2)	0.378655(-2)	0.379307(-2)	0.27316(-2)	0.164044(-2)
0.117202(-2)	0.231632(-2)	0.273142(-2)	0.317616(-2)	0.273153(-2)	0.231651(-2)	0.11721(-2)
0.731387(-3)	0.117199(-2)	0.164024(-2)	0.161419(-2)	0.164037(-2)	0.117209(-2)	0.731457(-3)
Voticity solutions, ω						
0.195016(-1)	0.290295(-1)	0.342178(-1)	0.357199(-1)	0.342064(-1)	0.290187(-1)	0.194987(-1)
0.290234(-1)	0.467565(-1)	0.558194(-1)	0.588306(-1)	0.558067(-1)	0.467449(-1)	0.290183(-1)
0.342083(-1)	0.558135(-1)	0.678284(-1)	0.714335(-1)	0.678198(-1)	0.558045(-1)	0.342046(-1)
0.357143(-1)	0.588235(-1)	0.714286(-1)	0.756302(-1)	0.714286(-1)	0.588235(-1)	0.357143(-1)
0.342003(-1)	0.557937(-1)	0.67807(-1)	0.714236(-1)	0.678157(-1)	0.558027(-1)	0.34204(-1)
0.290123(-1)	0.467309(-1)	0.557877(-1)	0.588164(-1)	0.558005(-1)	0.467425(-1)	0.290174(-1)
0.194952(-1)	0.290062(-1)	0.341908(-1)	0.357086(-1)	0.342022(-1)	0.29017(-1)	0.194981(-1)

TABLE 4. Numerical solutions of the Navier-Stokes equations for Re = 1000 (in the form a(b)=a×10^b) (EDG with Multigrid)

Stream solutions, ψ						
0.778099(-3)	0.124571(-2)	0.177732(-2)	0.1659(-2)	0.164902(-2)	0.116495(-2)	0.724124(-3)
0.121315(-2)	0.242742(-2)	0.278958(-2)	0.320966(-2)	0.271352(-2)	0.228667(-2)	0.115322(-2)
0.164707(-2)	0.270084(-2)	0.374389(-2)	0.370792(-2)	0.370873(-2)	0.266512(-2)	0.160208(-2)
0.152796(-2)	0.2985(-2)	0.358008(-2)	0.416939(-2)	0.362837(-2)	0.3063(-2)	0.155968(-2)
0.147978(-2)	0.24725(-2)	0.342206(-2)	0.350894(-2)	0.357236(-2)	0.25994(-2)	0.157364(-2)
0.102813(-2)	0.200316(-2)	0.242353(-2)	0.287194(-2)	0.253639(-2)	0.21886(-2)	0.111448(-2)
0.62959(-3)	0.100743(-2)	0.139699(-2)	0.144539(-2)	0.151265(-2)	0.109924(-2)	0.69569(-3)
Voticity solutions, ω						
0.219191(-1)	0.418363(-1)	0.471107(-1)	0.401595(-1)	0.351919(-1)	0.292847(-1)	0.195144(-1)
0.335808(-1)	0.564263(-1)	0.663967(-1)	0.627102(-1)	0.569331(-1)	0.468075(-1)	0.289558(-1)
0.376268(-1)	0.588116(-1)	0.698905(-1)	0.7175(-1)	0.673978(-1)	0.552883(-1)	0.338761(-1)
0.348491(-1)	0.527106(-1)	0.625344(-1)	0.713766(-1)	0.689336(-1)	0.574467(-1)	0.350806(-1)
0.297904(-1)	0.4338(-1)	0.530798(-1)	0.636619(-1)	0.638884(-1)	0.537842(-1)	0.333747(-1)
0.239713(-1)	0.346954(-1)	0.404327(-1)	0.50678(-1)	0.51669(-1)	0.448022(-1)	0.282037(-1)
0.164863(-1)	0.202474(-1)	0.228111(-1)	0.301991(-1)	0.316825(-1)	0.277807(-1)	0.19013(-1)

For comparison purposes, the numerical solutions obtained by the conventional *centred difference* scheme as the benchmark solutions are also given in TABLES 5 and 6. In *Step 2* of the numerical algorithm presented in Section IV *B*, we replaced the EDG inner iterative process with the *centred difference* inner iterative scheme. Since there is no exact solutions available, the symmetry of the solutions obtained in these experiments suggest that they are good approximations to the exact ones.

TABLE 5. Numerical solutions of the Navier-Stokes equations for Re = 1 (in the form a(b)=a×10^b) (Centred Difference with Multigrid)

Stream solutions, ψ						
0.662753(-3)	0.118657(-2)	0.151534(-2)	0.162687(-2)	0.15153(-2)	0.11865(-2)	0.6627(-3)
0.118657(-2)	0.213456(-2)	0.273357(-2)	0.293737(-2)	0.273352(-2)	0.213447(-2)	0.11865(-2)
0.151533(-2)	0.273356(-2)	0.350682(-2)	0.377049(-2)	0.350679(-2)	0.27335(-2)	0.151528(-2)
0.162683(-2)	0.293732(-2)	0.377045(-2)	0.405476(-2)	0.377045(-2)	0.293732(-2)	0.162683(-2)
0.151523(-2)	0.273342(-2)	0.350671(-2)	0.377042(-2)	0.350675(-2)	0.273348(-2)	0.151527(-2)
0.118641(-2)	0.213435(-2)	0.273341(-2)	0.293726(-2)	0.273346(-2)	0.213444(-2)	0.118648(-2)
0.662637(-3)	0.118641(-2)	0.151522(-2)	0.162679(-2)	0.151526(-2)	0.118648(-2)	0.66269(-3)
Voticity solutions, ω						
0.177842(-1)	0.277513(-1)	0.329183(-1)	0.345255(-1)	0.329164(-1)	0.277461(-1)	0.177793(-1)
0.277522(-1)	0.446702(-1)	0.537716(-1)	0.566422(-1)	0.537691(-1)	0.446637(-1)	0.27746(-1)
0.329201(-1)	0.537728(-1)	0.652308(-1)	0.688774(-1)	0.652291(-1)	0.537686(-1)	0.32916(-1)
0.345244(-1)	0.566406(-1)	0.688764(-1)	0.727826(-1)	0.688764(-1)	0.566406(-1)	0.345244(-1)
0.329117(-1)	0.53764(-1)	0.652265(-1)	0.688753(-1)	0.652281(-1)	0.537682(-1)	0.329158(-1)
0.277395(-1)	0.446566(-1)	0.537651(-1)	0.566391(-1)	0.537677(-1)	0.44663(-1)	0.277457(-1)
0.177742(-1)	0.277405(-1)	0.329135(-1)	0.345232(-1)	0.329154(-1)	0.277456(-1)	0.177791(-1)

TABLE 6. Numerical solutions of the Navier-Stokes equations for Re = 1000 (in the form a(b)=a×10^b) (Centred Difference with Multigrid)

Stream solutions, ψ						
0.716677(-3)	0.125155(-2)	0.15582(-2)	0.164967(-2)	0.152556(-2)	0.11903(-2)	0.663752(-3)
0.124946(-2)	0.220781(-2)	0.278089(-2)	0.296062(-2)	0.274199(-2)	0.213606(-2)	0.118616(-2)
0.154171(-2)	0.27577(-2)	0.351602(-2)	0.376789(-2)	0.349885(-2)	0.272552(-2)	0.151055(-2)
0.159791(-2)	0.288906(-2)	0.372256(-2)	0.401407(-2)	0.373935(-2)	0.291652(-2)	0.161645(-2)
0.144204(-2)	0.262832(-2)	0.34141(-2)	0.370039(-2)	0.345838(-2)	0.270352(-2)	0.150101(-2)
0.110107(-2)	0.201572(-2)	0.263222(-2)	0.286336(-2)	0.268411(-2)	0.210474(-2)	0.117261(-2)
0.604982(-3)	0.110732(-2)	0.144919(-2)	0.157943(-2)	0.148413(-2)	0.116798(-2)	0.654125(-3)
Voticity solutions, ω						
0.234049(-1)	0.335044(-1)	0.352454(-1)	0.354748(-1)	0.332977(-1)	0.278922(-1)	0.178275(-1)
0.340247(-1)	0.506646(-1)	0.563774(-1)	0.57731(-1)	0.541988(-1)	0.448188(-1)	0.277927(-1)
0.359543(-1)	0.560764(-1)	0.662435(-1)	0.692479(-1)	0.653227(-1)	0.537664(-1)	0.328997(-1)
0.332058(-1)	0.543845(-1)	0.676472(-1)	0.720695(-1)	0.684527(-1)	0.563996(-1)	0.344145(-1)
0.282148(-1)	0.481501(-1)	0.622662(-1)	0.673053(-1)	0.643939(-1)	0.533396(-1)	0.327338(-1)
0.21859(-1)	0.380123(-1)	0.50337(-1)	0.548668(-1)	0.528529(-1)	0.442064(-1)	0.275559(-1)
0.135384(-1)	0.23001(-1)	0.305775(-1)	0.333472(-1)	0.323193(-1)	0.274523(-1)	0.176583(-1)

TABLES 1 and 2 show that the halfsweep EDG multigrid method with V(1,1)-cycle becomes much faster than the original EDG method as the grid size gets relatively large, with gains in timings over the latter method between 1% to 93% for the case when Re = 1000. It is obvious that as the grid size increases, the iteration count needed for convergence becomes lesser and lesser for the multigrid scheme.

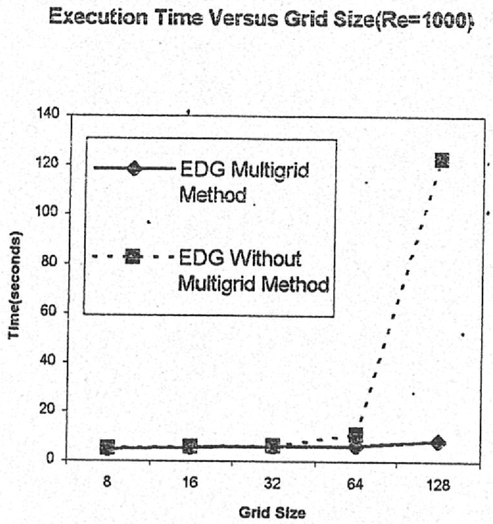

Fig. 6 Execution times of both EDG schemes

TABLE 7. The largest Reynolds number for different grid size and its number of outer iterations (EDG with multigrid)

Halfsweep Four Point EDG Multigrid (for maximum iteration = 150)		
Grid size	The largest Reynolds number for convergence	Number of outer iteration
8	3700	23
16	6500	16
32	12200	11
64	23800	8
128	47000	6
256	92900	5
512	182400	5

TABLE 7 shows the listing of outer iteration numbers for different grid size together with the largest Reynolds number which results in convergence in each case for the EDG multigrid algorithm. TABLE 8 shows the iteration count for convergence for the original EDG scheme without applying the multigrid technique for several grid sizes. For all the cases tested, the maximum number of iterations is set at 150 for both the inner and outer iterative processes. Note that the application of multigrid technique to the EDG method results in the acceleration of convergence in the iteration process such that the method converges for large Reynolds numbers which was not possible previously for the original scheme.

TABLE 8. Iteration numbers for different grid size (original EDG without multigrid)

Four Point EDG Without Multigrid (for maximum iteration = 150)				
Grid size	Re	Number of outer iteration	Number of inner iteration for ψ	Number of inner iteration for ω
8	1	1	1	49
		2	42	21
		3	12	3
		4	1	1
	10	1	1	49
		2	42	27
		3	18	8
		4	4	1
		5	1	1
	100	1	1	49
		2	42	34
		3	25	16
		4	8	11
		5	4	3
		6	1	1
	1000	1	1	49
		2	42	38
		3	35	33
		4	29	24
		5	19	21
		6	15	16
		7	9	10
		8	3	5
		9	1	2
		10	1	1
	3800	1	1	49
		2	42	103
		3	39	31
		4	37	26
		5	35	26
		6	33	24
		7	20	23
		8	28	22
		9	26	20
		10	24	19
		11	21	18
		12	19	16
		13	17	15
		14	15	13
		15	13	12
		16	10	11
		17	8	9
		18	5	8
		19	1	2
		20	1	4
		21	1	4
		22	1	1
	>3800	No convergence		
16	>0	No convergence		
32	>0	No convergence		
64	>0	No convergence		
128	>0	No convergence		
256	>0	No convergence		
512	>0	No convergence		

VI. CONCLUSION

Second order group iterative method derived from *rotated* discretisation formulas have been used in conjunction with the multigrid technique to develop an efficient multigrid solver to the steady-state incompressible Navier-Stokes equations. Our preliminary results indicated that the multigrid scheme can accelerate the original group iteration process quite significantly. We have also shown that the group multigrid method do 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 formula. It would be worthwhile to investigate the parallel implementation of this multigrid method on a parallel or distributed system and the results will be reported soon.

Acknowledgment

The authors acknowledge the Fundamental Research Grant Scheme (304/PMATHS/670020) that has resulted in this article.

References

[1] A.R. Abdullah, The Four Point Explicit Decoupled Group (EDG) Method: A Fast Poisson Solver, *International Journal of Computer Mathematics*, Vol. **38**, 1991, pp. 61-70.

[2] N.H.M.Ali and A.R.Abdullah, "Four point EDG: A Fast Solver for the Navier-Stokes Equation", in *Proceedings of the International Conference on Modelling Simulation and Optimization, Gold Coast, Australia*, 1996, CD Rom-File 242-65.pdf, ISBN: 0-88986-197-8.

[3] A.R. Abdullah and N.H.M. Ali, "The Comparative Study of Parallel Strategies for the Solution of Elliptic PDE's", *Parallel Algorithms and Applications*, Vol. **10**, 1996, pp. 93-103.

[4] N. H. M. Ali and A. R. Abdullah, 'A New Fast Navier-Stokes Solver and Its Parallel Implementation' *Malaysian Journal of Computer Science* **10**(2), 1997, pp. 51-59.

[5] N.H.M.Ali & A.R. Abdullah, New Rotated Iterative Algorithms For The Solution of A Coupled System of Elliptic Equations, *International Journal of Computer Mathematics*, Vol. 74, 1999, pp. 223-251.

[6] N.H.M. Ali, Y. Yunus, & M. Othman, 'A New Nine-Point Multigrid V-Cycle Algorithm', *SAINS MALAYSIANA*, Vol. **31**, 2002, pp. 135-147 (ISSN 0126-6039)

[7] A.J.Chorin, "Numerical Solution of the Navier- Stokes Equations", *Maths. of Comp.*, Vol. 22, 1969, pp. 745-762.

[8] G. Dahlquist and A. Bjorck, *Numerical Methods*. Englewood Cliffs, N.Jersey: Prentice-Hall, 1974.

[9] D. Greenspan, Discrete Numerical Methods in Physics and Engineering, Academic Press Inc., New York, 1974.

[10] W. Hackbush,, 'Parabolic Multi-grid Methods', *Computing Methods in Applied Sciences and Engineering*, VI. North Holland, Amsterdam, 1984

[11] M. Othman, and A.R. Abdullah, The Halfsweeps Multigrid Method As A Fast Multigrid Poisson Solver, *International Journal of Comp. Mathematics*, Vol. **69**, 1997, pp. 319-329

[12] M.S. Sahimi, & D.J. Evans, The numerical solution of a coupled system of elliptic equations using the AGE fractional scheme, *Int. Journal Computer Maths*. Vol. **50**, 1994, pp. 65-87.