Multigrid Application on A New Coupled Elliptic Solver: A Preliminary Study

[1]Lim Lay Ngor, [2]Norhashidah Hj. Mohd. Ali,
Pusat Pengajian Sains Matematik, Universiti Sains Malaysia, 11800 Minden, Pulau Pinang
E-mail: [1]llngor@yahoo.com, [2]shidah@cs.usm.my

ABSTRACT. *A new multigrid scheme using halfsweep four-point explicit decoupled group (EDG) method in solving the steady-state incompressible Navier-Stokes equation is presented. The concept of halfsweep multigrid was initiated by Othman and Abdullah (1997) where promising results was established and confirmed. In this paper, we apply the multigrid V-cycle algorithm on the four point explicit decoupled group (EDG). In Ali and Abdullah [5], a new group iterative scheme, the Explicit Decoupled Group scheme, was developed as a more efficient Navier-Stokes solver on rotated grids compared to the iterative schemes based on the standard five-point formulae and the Alternating Group Explicit scheme due to Sahimi and Evans [10]. Our aim in this work is to investigate the possibility of combining multigrid technique with this second order group iterative scheme to solve the steady-state incompressible Navier-Stokes equation as a way to further improve the convergence of the method. The multigrid formula will be constructed to accelerate the convergence of the iterative process for large Reynolds numbers previously claimed not possible for the original iterative scheme. The preliminary study of the experimental work performed will be reported.*

1. INTRODUCTION. Consider the governing equations for flow fields which describe the two dimensional steady-state Navier-Stokes equations with continuity equation,

$$uu_x + vu_y = -p_x + \frac{1}{Re}\nabla^2 u \quad \ldots\ldots\ldots \quad (1)$$

$$uv_x + vv_y = -p_y + \frac{1}{Re}\nabla^2 v \quad \ldots\ldots\ldots \quad (2)$$

$$u_x + v_y = 0 \quad \ldots\ldots\ldots \quad (3)$$

where u,v are velocities of the flow in x,y directions, p is the pressure and $Re \neq 0$ denotes the non-dimensional Reynolds number. Here, $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the usual Laplacian operator. We can now use the stream-vorticity function approach, i.e., use the stream function ψ and vorticity function ω as dependent variables by defining

$$u = \psi_y \text{ and } v = -\psi_x \quad (4)$$

The vorticity ω is then defined by $\omega = -(u_y - v_x) = -(\psi_{yy} + \psi_{xx}) = -\nabla^2\psi$. (5)

Equations (1) and (2) may then be differentiated and combined to give

$$\nabla^2\omega + Re(\psi_x\omega_y - \psi_y\omega_x) = 0. \quad (6)$$

Equation for p may be obtained from (1) and (2) in a similar way to get

$$\nabla^2 p = 2(\psi_{xx}\psi_{yy} - (\psi_{xy})^2). \quad (7)$$

In short, to solve (1) – (3), we need to solve the coupled system of equations

$$\nabla^2\psi = -\omega \quad (8)$$

$$\nabla^2\omega + Re(\psi_x\omega_y - \psi_y\omega_x) = -c \quad (9)$$

for ψ and ω. Once the stream function ψ is known, the velocities u,v and pressure p can be obtained via (4) and (7). Here, $x,y \in \Omega = (0,L)x(0,L)$ with a set of conditions for ψ and ω prescribed at the boundary while c and Re (the Reynolds number) are non-negative constants. 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 the Poisson problem (8) and convection-diffusion problem (9) successively with $\psi = 0$ and $\omega = 0$ respectively along $\partial\Omega$.

The finite difference approximations of equations (8) and (9) using the *centred difference* formula at the point (x_i, y_j) will result the following:

$$-\psi_{i-1,j}^{(k+1)} - \psi_{i,j-1}^{(k+1)} + 4\psi_{ij}^{(k+1)} - \psi_{i,j+1}^{(k+1)} - \psi_{i+1,j}^{(k+1)} = h^2 \omega_{ij}^{(k)} \tag{10}$$

$$\begin{aligned} &-[1-\sigma(\psi_{i,j-1}^{(k+1)} - \psi_{i,j+1}^{(k+1)})]\omega_{i-1,j}^{(k+1)} - [1+\sigma(\psi_{i-1,j}^{(k+1)} - \psi_{i+1,j}^{(k+1)})]\omega_{i,j-1}^{(k+1)} + 4\omega_{ij}^{(k+1)} \\ &-[1-\sigma(\psi_{i-1,j}^{(k+1)} - \psi_{i+1,j}^{(k+1)})]\omega_{i,j+1}^{(k+1)} - [1+\sigma(\psi_{i,j-1}^{(k+1)} - \psi_{i,j+1}^{(k+1)})]\omega_{i+1,j}^{(k+1)} = h^2 c_{ij}^{(k)}. \end{aligned} \tag{11}$$

here $\sigma = Re/4$ and i,j =1,2,...,n-1.

On the other hand, in discretising (8) and (9) we use the rotated *difference* formula at the point (x_i, y_j) which will result in the following:

$$-\psi_{i-1,j+1}^{(k+1)} - \psi_{i-1,j-1}^{(k+1)} + 4\psi_{ij}^{(k+1)} - \psi_{i+1,j+1}^{(k+1)} - \psi_{i+1,j-1}^{(k+1)} = 2h^2 \omega_{ij}^{(k)} \tag{12}$$

$$\begin{aligned} &-[1-\sigma(\psi_{i-1,j-1}^{(k+1)} - \psi_{i+1,j+1}^{(k+1)})]\omega_{i-1,j+1}^{(k+1)} - [1+\sigma(\psi_{i-1,j+1}^{(k+1)} - \psi_{i+1,j-1}^{(k+1)})]\omega_{i-1,j-1}^{(k+1)} + 4\omega_{ij}^{(k+1)} \\ &-[1-\sigma(\psi_{i-1,j+1}^{(k+1)} - \psi_{i+1,j-1}^{(k+1)})]\omega_{i+1,j+1}^{(k+1)} - [1+\sigma(\psi_{i-1,j-1}^{(k+1)} - \psi_{i+1,j+1}^{(k+1)})]\omega_{i+1,j-1}^{(k+1)} = 2h^2 c_{ij}^{(k)}. \end{aligned} \tag{13}$$

here $\sigma = Re/4$ and i,j =1,2,...,n-1. Observe that if ω is known, then (8) is a linear elliptic equation in ψ, and if ψ is known, then (9) is a linear elliptic equation in ω. Suppose $\psi^{(0)}$ and $\omega^{(0)}$ are the initial guesses, we can use the $\omega^{(0)}$ in (8) to produce $\psi^{(1)}$. Use this $\psi^{(1)}$ in (9) to produce $\omega^{(1)}$. Then use this $\omega^{(1)}$ in (8) 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) an alternating sequence of outer iterates may be generated.

2. THE FOUR POINT EXPLICIT DECOUPLED GROUP OUTER-INNER ITERATIVE METHOD. To construct the *explicit decoupled* group iterative method, we must obtain the following equation from equation (13):

$$\begin{aligned} \omega_{ij}^{(k+1)} = \frac{1}{4}[&(1-\sigma(\psi_{i-1,j-1}^{(k+1)} - \psi_{i+1,j+1}^{(k+1)}))\omega_{i-1,j+1}^{(k+1)} + (1+\sigma(\psi_{i-1,j+1}^{(k+1)} - \psi_{i+1,j-1}^{(k+1)}))\omega_{i-1,j-1}^{(k+1)} \\ &+(1-\sigma(\psi_{i-1,j+1}^{(k+1)} - \psi_{i+1,j-1}^{(k+1)}))\omega_{i+1,j+1}^{(k+1)} + (1+\sigma(\psi_{i-1,j-1}^{(k+1)} - \psi_{i+1,j+1}^{(k+1)}))\omega_{i+1,j-1}^{(k+1)} + 2h^2 c_{ij}^{(k)}]. \end{aligned} \tag{14}$$

Assume that the solution at any group of four points on the solution domain is solved using the rotated equation (14). This will result in a (4x4) system of equations (Ali and Abdullah 1996),

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

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

$$= \begin{bmatrix} \text{rhs}_{ij} \\ \text{rhs}_{i+1,j+1} \\ \text{rhs}_{i+1,j} \\ \text{rhs}_{i,j+1} \end{bmatrix} \qquad (15)$$

This system leads to a *decoupled* system of (2x2) equations, which can be made explicit as follows:

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

and

$$\begin{bmatrix} \tilde{\omega}_{i+1,j} \\ \tilde{\omega}_{i,j+1} \end{bmatrix} = \frac{1}{16-[(1-\sigma(\psi_{i,j-1}-\psi_{i+2,j+1}))(1+\sigma(\psi_{i-1,j}-\psi_{i+1,j+2}))]} \times \begin{bmatrix} 4 & 1-\sigma(\psi_{i,j-1}-\psi_{i+2,j+1}) \\ 1+\sigma(\psi_{i-1,j}-\psi_{i+1,j+2}) & 4 \end{bmatrix} \begin{bmatrix} \text{rhs}_{i+1,j} \\ \text{rhs}_{i,j+1} \end{bmatrix} \qquad (17)$$

The computational molecule for equations (16) and (17) are given in Figures 1(a) and 1(b) respectively:

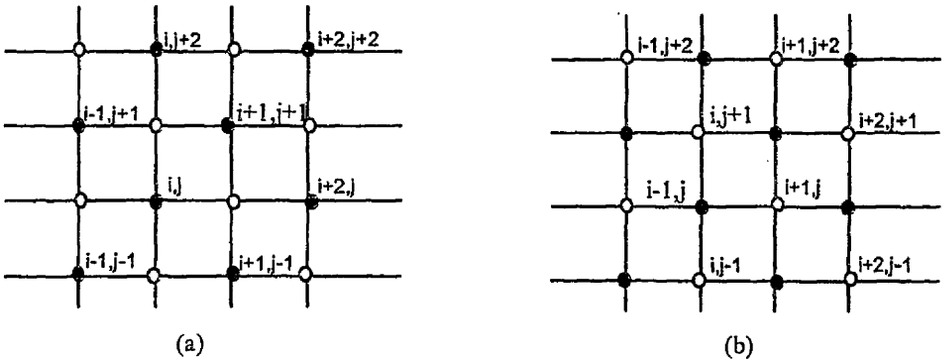

Figure 1. Computational molecule of Equations: (a) (16) and (b) (17).

Note that from both equations and their corresponding computational molecule (Figure 1), iteration of points for the vorticity solutions from equation (16) can be carried out by only involving points of type ● only, while the iterations arise from equation (17) can be implemented by involving points of type O only. This means the

iterative evaluation of points from each group requires contribution of points only from the same group. Therefore, the iterations can be carried out on either one of the two types 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 nodal points. Hence, the four-point *Explicit Decoupled Group* (EDG) iterative scheme corresponds to the generation of iterations on one type of points (say, the points of type O) until a certain convergence criteria is met. After convergence is achieved, evaluate the solutions at the remaining nodal points (points of type ●) using the *centred difference* formula (11). Otherwise, repeat the iteration cycle.

3. MULTIGRID METHOD. From the research, it was observed that the multigrid method is frequently used in the area of computational fluid dynamics and structural mechanics. Thus it is very suitable to use for the steady state Navier-Stokes equation. Multigrid method actually is a smoothing tool for an iterative method, which involves solving the problem on the coarse grids and interpolates the solution back to the fine grids. This means that a series of problems to be solved on a hierarchy of errors with differing mesh sizes is necessary for the use of multigrid method.

In this paper, we will choose the multigrid V-cycle to apply on the equation. Basically, a V-cycle multigrid method consists of smoothing the error using a relaxation scheme (e.g. Gauss-Seidel or Jacobi iterative scheme), solving an approximation to the smooth error equation on a coarse grid, interpolating the error correction to the fine grid, and finally adding the error correction into the approximation. In summary, a multigrid method with the $V(v_1,v_2)$-cycle is the process that goes from the finest grid down to the coarsest grid and back from the coarsest grid up to the finest Here, v_1 is the number of relaxation scheme at each level before projecting the residual to the coarse grid (pre-smoothing), and v_2 is the number of relaxation after interpolating the solution back to the fine grid (post-smoothing). That's why the multigrid method very efficient on correcting the convergence rate of a relaxation method.

4. THE FOUR POINT EDG MULTIGRID ALGORITHM.

Fundamentally, the concept used in designing this scheme is analogous to the halfsweep rotated five-point multigrid method [9]. All the mesh points in solution domain Ω^h are labeled in red O and black □ points as in Figure 2. The red points group is further divided into two groups i.e. red points labeled ● and O (Figure 3(a)).

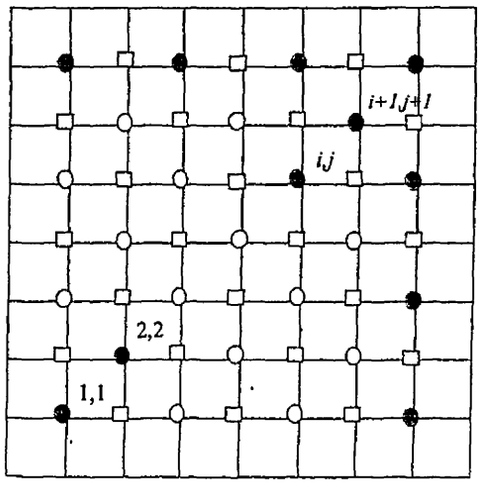

Figure 2. The position of grid points involved in updating u_{ij}

Referring to Figure 2, you will notice that the red points at the last row and column ( ) could not possibly form the computational molecule because the computational molecule needs a pair of points, but there are no suitable points to pair with each red point. Therefore, to evaluate the red points the *rotated difference* stencil (13) is used besides using EDG.

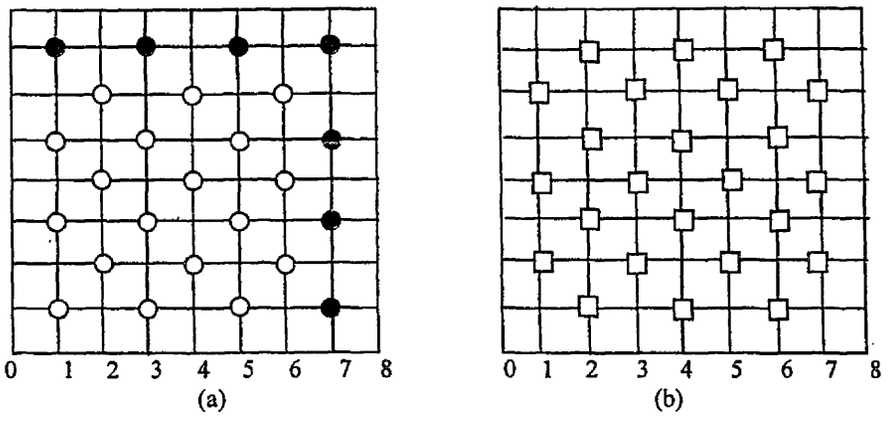

Figure 3. The position of red points (a) and the position of black points (b).

The red points (points ● and O) are iterated until they converge, based on certain convergence criteria. The red points of type O will be iterated using the (16) stencil whilst the red points of type ● (the red points next to the boundaries) will be iterated using the *rotated difference* stencil (13). Only then, can the black points be evaluated directly using the *centred difference* stencil (11). Figure 4 shows the structure of the halfsweep EDG multigrid method with V-cycle on the red points.

In using the halfsweep multigrid method, only the red mesh points (points ● and O) will undergo the process of iterative evaluation using either the (16) or the (13) stencil. The points of type O are evaluated iteratively using 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 halfsweep multigrid algorithm consists of 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

$$r^h = A^h v^h - f^h \tag{18}$$

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

$$\Re_h^{2h} = \frac{1}{8}\begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{19}$$

At the coarser grid, the new residual is defined as

$$r^{2h} = \mathfrak{R}_n^{2h} r^h \qquad (20)$$

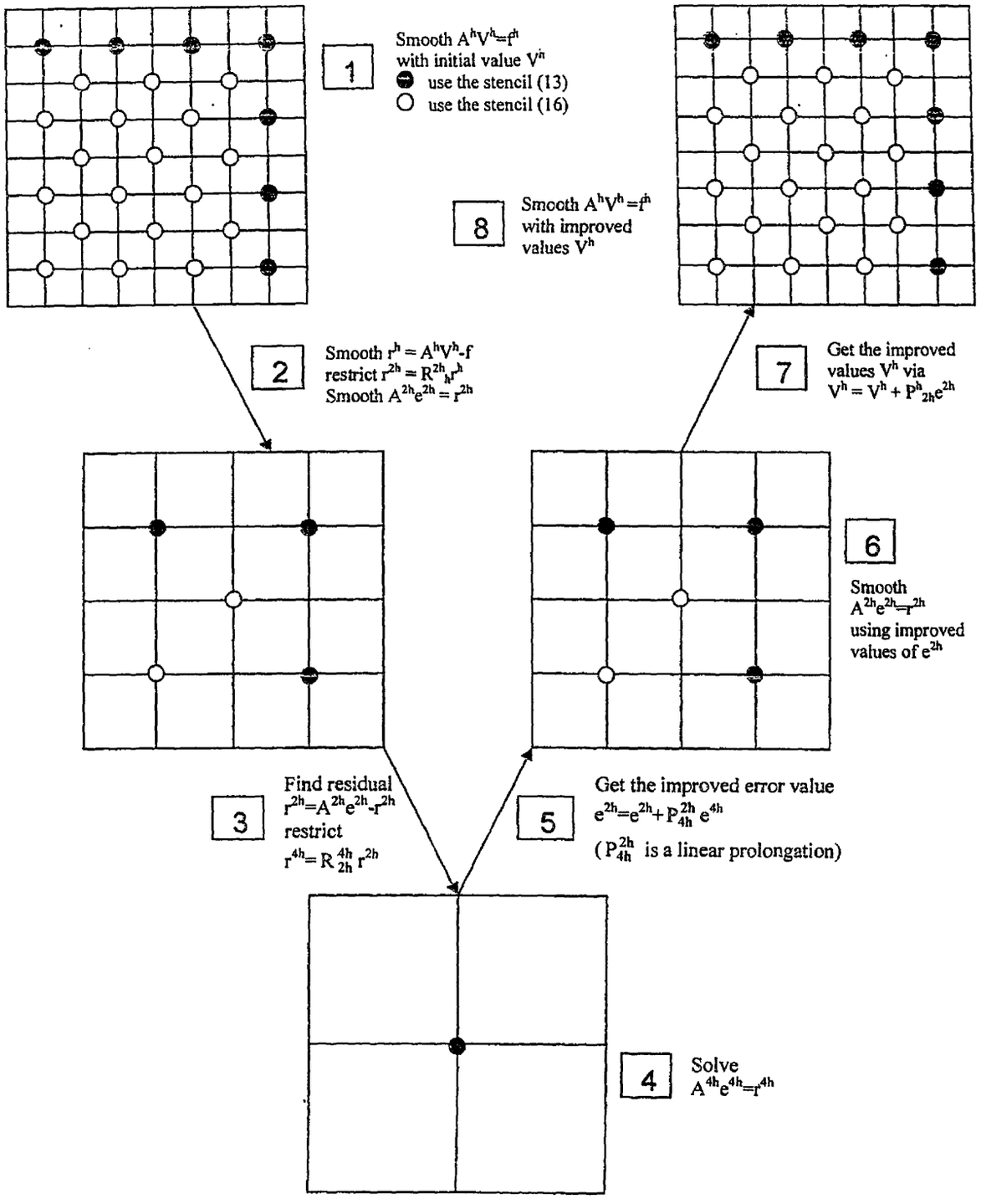

Figure 4. The structure of halfsweep EDG multigrid method with V-cycle.

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

$A^{2h}e^{2h} = r^{2h}$ $\cdot e^{2h}$ is the error value

e^{2h} is found 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 we reach the coarsest grid. The error estimation value is acquires 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

$$v_{2i,2j}^{h} = v_{i,j}^{2h} \qquad \text{for all } i, j \text{ both even or odd.} \tag{21}$$

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

$$v_{4i-2,4j}^{h} = \frac{1}{2}\left(v_{4i-2,4j-2}^{2h} + v_{4i-2,4j+2}^{2h}\right), \tag{22}$$

$$v_{i,2j-1}^{h} = \frac{1}{2}\left(v_{i,j}^{2h} + v_{i,j-1}^{2h}\right) \qquad \text{, for all } i,j = 1,2,...,N-1. \tag{23}$$

and

$$v_{i,j}^{h} = \frac{1}{4}\left(v_{i-1,j-1}^{2h} + v_{i+1,j-1}^{2h} + v_{i-1,j+1}^{2h} + v_{i+1,j+1}^{2h}\right) \qquad \text{for all } i,j \text{ odd.} \tag{24}$$

In this paper, the Gauss-Seidel smoothing scheme was used to smooth the errors or residuals to ensure the convergence of the method. The Gauss-Seidel iteration scheme is the most popular smoothing method and it's a more effective smoother than Jacobi's scheme. These due to the fact that the new updated values for each iteration are used to calculate the next value. 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 red points at each level would be applied with the smoothing scheme where the points will be iterated till they converged. Then obtain the solutions once at the rest of the points (the black points) using the *centred difference* stencil (11). The general algorithm for the explicit decoupled group multigrid method 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 (11). }

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 ν_1 times on $A^h v^h = f^h$ applying Gauss Seidel at the finest domain using stencil (16) or (13).
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 ν_2 times on $A^h v^h = f^h$ applying Gauss Seidel on Ω^h using the stencil (16) or (13).}
}HALFSWEEP_EDG_MULTIGRID Algorithm()
{Flag = 0

WHILE (Flag != 1) DO{

Flag = 1

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

IF $\cdot|v_{ij}^{(k+1)} - v_{ij}^{(k)}| > \varepsilon$ on the red points, set Flag = 0

Iterate ++

Swap $v_{ij}^{(k)} \leftarrow v_{ij}^{(k+1)}$ for all red points}

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

Return v^h as an approximate solution}

5. NUMERICAL EXPERIMENTATION AND RESULTS. Numerical experiment has been carried out using the halfsweep explicit decoupled group multigrid algorithm described previously to solve the following Navier-Stokes problem [10]:

$$\begin{aligned} \nabla^2\psi &= -\omega \\ \nabla^2\omega + \mathrm{Re}(\psi_x\omega_y - \psi_y\omega_x) &= -1 \end{aligned} \qquad (25)$$

with the boundary conditions

$$\begin{aligned} \psi(x,0) = \psi(x,1) = \omega(x,0) = \omega(x,1) = 0, \quad & 0 \le x \le 1, \\ \psi(0,y) = \psi(1,y) = \omega(0,y) = \omega(1,y) = 0, \quad & 0 \le y \le 1, \end{aligned} \qquad (26)$$

The grid spacing used was h=0.1 and the problem were solved for various values of Reynolds number $\mathrm{Re} \ge 1$. For each case, the experimental optimum relaxation parameter γ was chosen to within ± 0.01 which gives the most rapid convergence. Throughout the experiment, the algorithm is run using C++ programming language on different size grids of Ω^h, Ω^{2h},...., Ω^{128h} with $V(1,1)$ cycle. Both 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. The results of the numerical experiment are shown in Table 1, Table 2 and Table 3 respectively. The execution time-versus mesh size were plotted and shown in Figures 5.

TABLE 1. Numerical Solutions Obtained for ψ when x=0.25 and 0.625 (Re = 1).

	y=0.125 (j=1)	y=0.25 (j=2)	y=0.375 (j=3)	y=0.5 (j=4)	y=0.625 (j=5)	y=0.75 (j=6)	y=0.875 (j=7)
x=0.25(i=2)							
EDG	0.00117223	0.00231682	0.00273185	0.00317656	0.00273174	0.00231663	0.00117215
EDG Multigrid	0.00117223	0.00231682	0.00273185	0.00317656	0.00273174	0.00231663	0.00117215
x=0.625 (i=5)							
EDG	0.00164036	0.0027315	0.00379296	0.00378655	0.00379307	0.0027316	0.00164044
EDG Multigrid	0.00164036	0.0027315	0.00379296	0.00378655	0.00379307	0.0027316	0.00164044

TABLE 2. Numerical Solutions Obtained for ω when x=0.25 and 0.625 (Re = 1).

	y=0.125 (j=1)	y=0.25 (j=2)	y=0.375 (j=3)	y=0.5 (j=4)	y=0.625 (j=5)	y=0.75 (j=6)	y=0.875 (j=7)
x=0.25(i=2)							
EDG	0.0290234	0.0467565	0.0558194	0.0588306	0.0558067	0.0467449	0.0290183
EDG Multigrid	0.0290234	0.0467565	0.0558194	0.0588306	0.0558067	0.0467449	0.0290183

x=0.625 (i=5)							
EDG	0.0342003	0.0557937	0.067807	0.0714236	0.0678157	0.0558027	0.034204
EDG Multigrid	0.0342003	0.0557937	0.067807	0.0714236	0.0678157	0.0558027	0.034204

TABLE 3. The experimental results for the EDG outer-inner iterative methods with and without Multigrid Method (Re=1).

Iteration numbers for the Navier-Stokes problem using **Halfsweep Four Point EDG Multigrid Method**					Iteration numbers for the Navier-Stokes problem using **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	Grid size	Time (secs)	Number of outer iteration	Number of inner iteration for ψ	Number of inner iteration for ω
8	3.46	1	1	7	8	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	16	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	32	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	64	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	128	116.44	1	1	6906
		2	6	3			2	5049	71
		3	1	1			3	3	1
							4	1	1

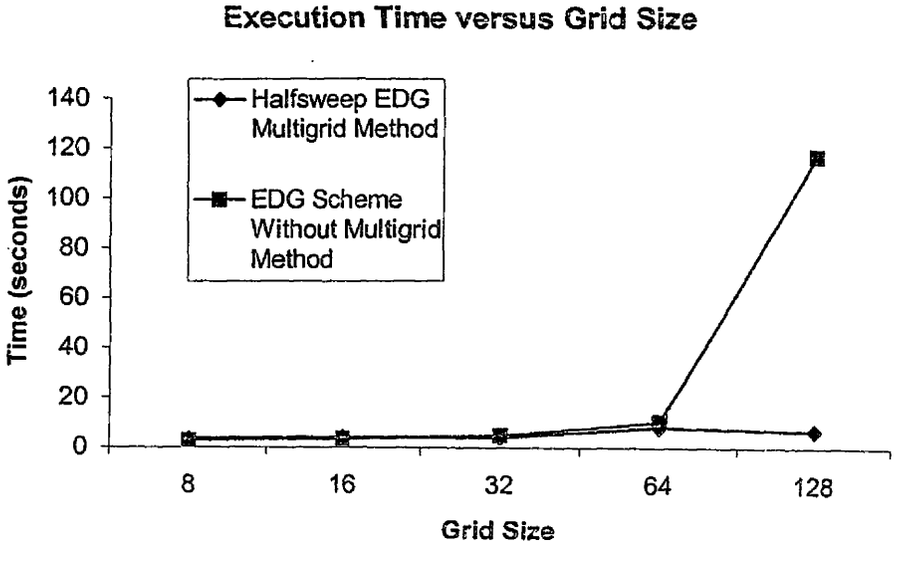

Figure 5. Execution time versus grid size

6. SUMMARY. In this report, we present a new Navier-Stokes solver using the halfsweep Explicit Decoupled Group multigrid algorithm. From the results obtained, we can observe that the Explicit Decoupled Group algorithm with the halfsweep multigrid method is faster than the Explicit Decoupled Group algorithm without any multigrid method, especially when the gird size is increased (Figure 5). In conclusion, the newly developed multigrid method proves to be a viable alternative Navier-Stokes solver.

7. 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] 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.

[3] N.H.M. Ali & A.R. Abdullah, New parallel point iterative solutions for the diffusion-convection equation. *Proceedings of the International Conference on Parallel and Distributed Computing and Networks, Singapore*, 1997, 136-139.

[4] N.H.M. Ali & A.R. Abdullah, Parallel processing in the solution of the 2D diffusion equation. *Proceedings of the Second Asian Technology Conference in Mathematics, Penang, Malaysia*, 1997, 392-404.

[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, 2002, 'A New Nine-Point Multigrid V-Cycle Algorithm', to appear in *SAINS MALAYSIANA*

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

[8] Gupta, M.M., Kouatchou, J. and Zhang, J., 1995, '*Comparison of 2nd and 4th Order Discretizations for Multigrid Poisson Solvers*', http://www.cs.yale.edu directory mgnet/papers/Gupta.

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

[10] 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.* **50**, 1994, pp. 65-87.

[11] J. Sulaiman, & A.R. Abdullah, *Beberapa Kaedah Lelaran Kumpulan Tak Tersirat Nyah-Pasangan Dengan Separuh Sapuan Multigrid Bagi Persamaan Poisson*, submitted for publication.

[12] Yousif, W.S. & Evans, D.J. , Explicit de-coupled group iterative methods and their parallel implementations. *Parallel Algorithms and Applications* 7, 1995, pp. 53-71.

NORHASHIDAH HJ. MOHD. ALI
LIM LAY NGOR
School of Mathematical Sciences,
Universiti Sains Malaysia,
11800 Minden, Pulau Pinang