Preconditioned Modified Explicit Decoupled Group for the Solution of Steady State Navier-Stokes Equation

Norhashidah Hj. Mohd Ali [1,*] *and Abdulkafi Mohammed Saeed* [2,*]

[1] School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Pulau Pinang, Malaysia
[2] Department of Mathematics , College of Science, Qassim University, Saudi Arabia

Abstract: Combining iterative schemes with suitable preconditioners may improve the rate of the convergence of the methods. However, the real difficulty lies in the construction of the correct preconditioners applied to the formulated schemes. In this paper, the Modified Explicit Decoupled Group Successive Over-Relaxation method is formulated to solve the two dimensional steady-state Navier-Stokes equations. A new block splitting preconditioned matrix is applied to the formulated scheme as an effort to accelerate the convergence rate. Numerical experiments are carried out to confirm the effectiveness of the preconditioner in terms of accuracy and execution timings. Comparison with its unpreconditioned counterpart will also be reported.

Keywords: Preconditioning, steady-state Navier-Stokes equation, Modified Explicit Decoupled Group (MEDG) method

1 Introduction

The finite difference method is the oldest discretisation technique for the governing partial differential equations (PDEs) and has long been a dominating means for solving fluid flow problems. Having the advantages of simplicity, robustness and versatility, numerical techniques based on finite differences are still playing important role for solving PDEs in the current researches. In many applications, the stream-vorticity function form of the Navier Stokes equations provides better insight into the physical mechanisms driving the flow than the primitive variable formulation in terms of velocities of the flow u, v in x, y directions and the pressure p. The stream-vorticity function formulation is also useful for numerical work since it avoids some problems resulting from the discretisation of the continuity equation. The governing equations for flow fields which describe the two dimensional steady-state Navier-Stokes equations with continuity equation are described in [1]. Using the stream-vorticity function approach, the Navier-Stokes equation may be represented as a coupled system of elliptic partial differential equations as shown below

$$\nabla^2 \psi = -\omega \qquad (1)$$

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

where $x, y \in \Omega = (0, L) \times (0, L)$ 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 $\partial \Omega$. Over the last few decades, there have been some interests in designing new methods for solving these stream-vorticity formulations ([2], [3], [4], [5], [6]). In particular, Ali and Abdullah [4] derived a group explicit method derived from a skewed five-point finite difference approximations where the proposed method was shown to be more efficient in CPU timings than the existing schemes based on the standard five-point difference stencil. With the realization that preconditioning technique is essential for the

* Corresponding author e-mail: shidah@cs.usm.my, abdelkafe@yahoo.com

improvement in the performance of iterative methods, research on preconditioners has moved to center stage in recent years ([7], [8], [9], [10], [11], [12]). Recently, Ali and Saeed [13] proposed a $(I+K)$-type preconditioning matrix in block form and applied it to the original system obtained from the four point Explicit Decoupled Group (EDG) method for solving the coupled system of elliptic partial differential Eqs. (1)-(2) with promising results.

In this paper, we shall formulate a new group explicit scheme namely the Modified Explicit Decoupled Group (MEDG) method, in solving this fundamental problem in fluid dynamic (1)-(2). We also introduce the application of a suitable preconditioner to this proposed scheme in an effort to further accelerate the convergence rate of the method. The outline of this paper is as follows: the formulation of the MEDG iterative method for solving the coupled system of elliptic equations will be given in Section 2. A specific preconditioned block splitting matrix applied to the proposed MEDG formula will be described in Section 3, followed by its convergence analysis in Section 4. The numerical results are presented to verify the efficiency of the preconditioned MEDG method in Section 5 and the concluding remarks is given in Section 6.

2 Modified Explicit Decoupled Group Formulation

Let us assume that a rectangular grid in the (x,y)-plane with grid spacing $h = L/n$ in both directions $x_i = ih$ and $y_j = jh$, $i, j = 0,1,...,n$ is 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 $\psi^{(1)}$. Again we can use this $\psi^{(1)}$ in (2) to produce $\omega^{(1)}$. Then we can use this $\omega^{(1)}$ in (1) to produce $\psi^{(2)}$, and so on. This indicates that at the grid point (x_i, y_j) the following alternating sequences of outer iterates can be generated as in **Fig. 1**.
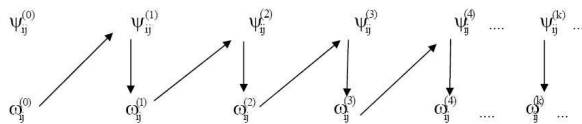

Fig. 1: Generation of outer iterates

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

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

$$-[1 - \sigma(\psi_{i,j-1}^{(k+1)} - \psi_{i,j+1}^{(k+1)})]\omega_{i-1,j}^{(k+1)} - [1 + \sigma(\psi_{i-1,j}^{(k+1)} - \psi_{i+1,j}^{(k+1)})]\omega_{i,j-1}^{(k+1)} + 4\omega_{ij}^{(k+1)}$$
$$-[1 - \sigma(\psi_{i-1,j}^{(k+1)} - \psi_{i+1,j}^{(k+1)})]\omega_{i,j+1}^{(k+1)} - [1 + \sigma(\psi_{i,j-1}^{(k+1)} - \psi_{i,j+1}^{(k+1)})]\omega_{i+1,j}^{(k+1)} = h^2c_{ij}^{(k)}. \quad (4)$$

Here $\sigma = \text{Re}/4$ and $i,j = 1,2,,n-1$. By rotating the i-plane axis and the j-plane axis clockwise by 45^0, Eqs. (3) and (4) become (5) and (6) respectively [4]:

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

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

We now consider the points with grid spacing $2h = 2L/n$. The application of centered difference equation on these $2h$ spaced points will transform the Eqs. (3) and (4) to the following equations:

$$-\psi_{i-2,j}^{(k+1)} - \psi_{i,j-2}^{(k+1)} + 4\psi_{i,j}^{(k+1)} - \psi_{i,j+2}^{(k+1)} - \psi_{i+2,j}^{(k+1)} = 4h^2\omega_{ij}^{(k)} \quad (7)$$

$$-[1 - \sigma(\psi_{i,j-2}^{(k+1)} - \psi_{i,j+2}^{(k+1)})]\omega_{i-2,j}^{(k+1)} - [1 + \sigma(\psi_{i-2,j}^{(k+1)} - \psi_{i+2,j}^{(k+1)})]\omega_{i,j-2}^{(k+1)} + 4\omega_{i,j}^{(k+1)}$$
$$-[1 - \sigma(\psi_{i-2,j}^{(k+1)} - \psi_{i+2,j}^{(k+1)})]\omega_{i,j+2}^{(k+1)} - [1 + \sigma(\psi_{i,j-2}^{(k+1)} - \psi_{i,j+2}^{(k+1)})]\omega_{i+2,j}^{(k+1)} = 4h^2c_{ij}^{(k)}. \quad (8)$$

Rotating the i-j axis clockwise 45^0, Eqs. (7) and (8) can be transformed into (9) and (10) respectively:

$$-\psi_{i-2,j+2}^{(k+1)} - \psi_{i-2,j-2}^{(k+1)} + 4\psi_{i,j}^{(k+1)} - \psi_{i+2,j+2}^{(k+1)} - \psi_{i+2,j-2}^{(k+1)} = 8h^2\omega_{ij}^{(k)} \quad (9)$$

$$-[1 - \sigma(\psi_{i-2,j-2}^{(k+1)} - \psi_{i+2,j+2}^{(k+1)})]\omega_{i-2,j+2}^{(k+1)} - [1 + \sigma(\psi_{i-2,j+2}^{(k+1)} - \psi_{i+2,j-2}^{(k+1)})]\omega_{i-2,j-2}^{(k+1)} + 4\omega_{ij}^{(k+1)}$$
$$-[1 - \sigma(\psi_{i-2,j+2}^{(k+1)} - \psi_{i+2,j-2}^{(k+1)})]\omega_{i+2,j+2}^{(k+1)} - [1 + \sigma(\psi_{i-2,j-2}^{(k+1)} - \psi_{i+2,j+2}^{(k+1)})]\omega_{i+2,j-2}^{(k+1)} = 8h^2c_{ij}^{(k)}. \quad (10)$$

Clearly, it can be seen that the application of (9)-(10) will result in a large and sparse system with the coefficient matrix being a block matrix depending on the ordering of points taken. Assume that the solution at any group of four points on the solution domain is achieved using the rotated equation (10). This will result in a (4x4) system of equations which lead to a decoupled system of (2x2) equations whose explicit forms are given by

$$\begin{bmatrix} \tilde{\omega}_{ij} \\ \tilde{\omega}_{i+2,j+2} \end{bmatrix} = \frac{1}{16 - [(1 - \sigma(\psi_{i-2,j+2} - \psi_{i+2,j-2}))(1 + \sigma(\psi_{i,j+4} - \psi_{i+4,j}))]}$$
$$\times \begin{bmatrix} 4 & 1 - \sigma(\psi_{i-2,j+2} - \psi_{i+2,j-2}) \\ 1 + \sigma(\psi_{i,j+2} - \psi_{i+4,j}) & 4 \end{bmatrix} \begin{bmatrix} rhs_{ij} \\ rhs_{i+2,j+2} \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} \tilde{\omega}_{i+2,j} \\ \tilde{\omega}_{i,j+2} \end{bmatrix} = \frac{1}{16 - [(1 - \sigma(\psi_{i,j-2} - \psi_{i+4,j+2}))(1 + \sigma(\psi_{i-2,j} - \psi_{i+2,j+4}))]}$$
$$\times \begin{bmatrix} 4 & 1 - \sigma(\psi_{i,j-2} - \psi_{i+4,j+2}) \\ 1 + \sigma(\psi_{i-2,j} - \psi_{i+2,j+4}) & 4 \end{bmatrix} \begin{bmatrix} rhs_{i+2,j} \\ rhs_{i,j+2} \end{bmatrix} \quad (12)$$

The computational molecule of Eqs. (11) and (12) are given in **Fig. 2** and **Fig.3** respectively. 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. (11) can be carried out by involving points of type ● only, while the iterations arise from Eq. (12) can be implemented by involving points of type ■

Fig. 2: Computational molecule of Eq.(11)

Fig. 3: Computational molecule of Eq. (12)

only. Due to this independency, the iterations can be carried out on either one of the two types of points with grid spacing *2h*, which means we can expect that the execution time to be reduced compared to the EDG method introduced in Ali and Saeed [13] .

In theory, if we apply Eq. (11) or (12) to each of the groups in natural ordering, then this will lead to the formation of a system of equations in the form:

$$A\tilde{\omega} = b \qquad (13)$$

whose coefficient matrix *A* is defined as

$$A = \begin{bmatrix} D_2 & V_2 & & & \\ L_6 & D_6 & V_6 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & V_{N-8} \\ & & & L_{N-4} & D_{N-4} \end{bmatrix}_{\frac{(N-2)^2}{2} \times \frac{(N-2)^2}{2}} \qquad (14)$$

with $D_j = \begin{bmatrix} R_{2,j} & Q_{2,j} & & & \\ U_{6,j} & R_{6,j} & Q_{6,j} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & Q_{N-8,j} \\ & & & U_{N-4,j} & R_{N-4,j} \end{bmatrix}_{(N-2)\times(N-2)}$,

$$L_j = \begin{bmatrix} S_{2,j} & & & \\ E_{6j} & S_{6j} & & \\ & \ddots & \ddots & \\ & & \ddots & \ddots \\ & & & E_{N-4,j} & S_{N-4,j} \end{bmatrix}_{(N-2)\times(N-2)},$$

$$V_j = \begin{bmatrix} F_{2,j} & G_{2,j} & & \\ & F_{6,j} & G_{6,j} & \\ & & \ddots & \ddots \\ & & & \ddots & G_{N-8,j} \\ & & & & F_{N-4,j} \end{bmatrix}_{(N-2)\times(N-2)}$$

where

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

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

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

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

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

$$U_{ij} = \begin{bmatrix} 0 & -[1-\sigma(\psi_{i-2,j-2} - \psi_{i+2,j+2})] \\ 0 & 0 \end{bmatrix}, \text{ and}$$

$$G_{ij} = \begin{bmatrix} 0 & 0 \\ -[1-\sigma(\psi_{i,j+4} - \psi_{i+4,j})] & 0 \end{bmatrix}, \quad \text{for} \quad i,j = 1(2)N\text{ - }4.$$

$$\tilde{\omega} = \begin{bmatrix} \tilde{\omega}_2 \\ \tilde{\omega}_6 \\ \vdots \\ \tilde{\omega}_{N-8} \\ \tilde{\omega}_{N-4} \end{bmatrix}_{\frac{(N-2)^2}{2}}, \tilde{\omega}_i = \begin{bmatrix} \tilde{\omega}_{2,i} \\ \tilde{\omega}_{6,i} \\ \vdots \\ \tilde{\omega}_{N-4,i} \end{bmatrix}_{(N-2)} \quad \text{for} \quad i=1(2)N\text{-}4,$$

$$\tilde{\omega}_{i,j} = \begin{bmatrix} \omega_{i,j} \\ \omega_{i+2,j+2} \end{bmatrix}, \quad \text{for} \quad i,j=1(2)N\text{-}4.$$

$$b = \begin{bmatrix} \tilde{v}_2 \\ \tilde{v}_6 \\ \vdots \\ \tilde{v}_{N-8} \\ \tilde{v}_{N-4} \end{bmatrix}_{\frac{(N-2)^2}{2}}, \tilde{v}_i = \begin{bmatrix} \tilde{v}_{2,i} \\ \tilde{v}_{6,i} \\ \vdots \\ \tilde{v}_{N-4,i} \end{bmatrix}_{(N-2)} \quad \text{for} \quad i=1(2)N\text{-}4.$$

$$\tilde{v}_{i,j} = \begin{bmatrix} v_{i,j} \\ v_{i+2,j+2} \end{bmatrix} \quad \text{for} \quad i,j=1(2)N\text{-}4.$$

In order to obtain the formula of the MEDG SOR method, we first need to derive the formula of the MEDG Gauss Seidel scheme as the following:

Let $b_1 = 1 - \sigma(\psi_{i-2,j+2} - \psi_{i+2,j-2})$, $b_2 = 1 + \sigma(\psi_{i,j+4} - \psi_{i+4,j})$,
$b_3 = 1/(16 - b_1 * b_2)$, $b_4 = (1 - \sigma(\psi_{i-2,j-2} - \psi_{i+2,j+2}))\omega_{i-2,j+2} + (1 + \sigma(\psi_{i-2,j+2} - \psi_{i+2,j-2}))\omega_{i-2,j-2} + (1 + \sigma(\psi_{i-2,j-2} - \psi_{i+2,j+2}))\omega_{i+2,j-2} + 8h^2c$.
$b_5 = (1 - \sigma(\psi_{ij} - \psi_{i+4,j+4}))\omega_{i,j+4} + (1 - \sigma(\psi_{i,j+4} - \psi_{i+4,j}))\omega_{i+4,j+4} + (1 + \sigma(\psi_{ij} - \psi_{i+4,j+4}))\omega_{i+4,j} + 8h^2c$.

The $(k+1)^{th}$ iterates of the Gauss-Seidel iterative scheme is defined by
$$\tilde{\omega}_{ij}^{(k+1)} = b_3 * (4 * b_4 + b_1 * b_5) \qquad \text{and}$$
$$\tilde{\omega}_{i+2,j+2}^{(k+1)} = b_3 * (b_2 * b_4 + 4 * b_5).$$
The $(k+1)^{th}$ iterates of the SOR iterative scheme are

given by

$$\omega_{ij}^{(k+1)} \;=\; \omega_{ij}^{(k)} \;+\; \gamma(\tilde{\omega}_{ij}^{(k+1)} \;-\; \omega_{ij}^{(k)}) \qquad \text{and}$$

$$\omega_{i+2,j+2}^{(k+1)} = \omega_{i+2,j+2}^{(k)} + \gamma(\tilde{\omega}_{i+2,j+2}^{(k+1)} - \omega_{i+2,j+2}^{(k)}).$$

In summary, the four-point MEDG SOR scheme corresponds to iterating the solutions at approximately half of the points in the solution domain using either (11) or (12) by implementing the relaxation procedure $\omega^{(k+1)} = \omega^{(k)} + \gamma(\tilde{\omega}^{(k+1)} - \omega^{(k+1)})$ until convergence is achieved, i.e., when $\left| \omega_{ij}^{(k+1)} - \omega_{ij}^{(k)} \right| \leq \varepsilon$; here γ is the optimum acceleration parameter and ε is the tolerance used. If convergence is achieved, then the solutions are evaluated at the rest of the nodal points (points of opposite type) using the centred difference formula (8). Otherwise, the iteration cycle is repeated.

3 The proposed preconditioned MEDG SOR iterative method

In this section we will apply the preconditioner $(\bar{P} = I + \bar{K})$ to the linear system resulting from the MEDG scheme with

$$\bar{K} = \begin{bmatrix} J_1 & & & \\ & J_1 & & \\ & & \ddots & \\ & & & J_1 \end{bmatrix}_{\frac{(N-2)^2}{2} \times \frac{(N-2)^2}{2}}, J_1 = \begin{bmatrix} \tilde{0} & & \\ \tilde{0} & \tilde{0} & \\ & \ddots & \ddots \\ & & -U_{N-4,j} & \tilde{0} \end{bmatrix}_{(N-2)\times(N-2)}$$

where $U_{N-4,j}$ defined as in Section 2.
Therefore, the preconditioner, $\bar{P} = I + \bar{K}$ matrix will become

$$I + \bar{K} = \begin{bmatrix} J_2 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_2 \end{bmatrix}_{\frac{(N-2)^2}{2} \times \frac{(N-2)^2}{2}}, J_2 = \begin{bmatrix} I_0 & & \\ \tilde{0} & I_0 & \\ & \ddots & \ddots \\ & & -U_{N-4,j} & I_0 \end{bmatrix}_{(N-2)\times(N-2)}.$$

Here I_0 is a 2×2 identity matrix.

Now, obtain $\bar{A}$ by multiplying $I + \bar{K}$ into A in Eq. (14)

$$\bar{A} = (I + \bar{K})A = \begin{bmatrix} J_2 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_2 \end{bmatrix} \begin{bmatrix} D_2 & V_2 & & \\ L_6 & D_6 & \ddots & \\ & \ddots & \ddots & V_{N-8} \\ & & L_{N-4} & D_{N-4} \end{bmatrix}$$

$$= \begin{bmatrix} M_{2,2} & V_2 & & \\ M_{4,6} & M_{2,6} & \ddots & \\ & \ddots & \ddots & V_{N-8} \\ & & M_{4,j} & M_{2,j} \end{bmatrix}_{\frac{(N-2)^2}{2} \times \frac{(N-2)^2}{2}},$$

Therefore, we can rewrite the system $\bar{A}\tilde{w} = \bar{b}$ as:

$$\begin{bmatrix} M_{2,2} & V_2 & & & & \\ M_{4,6} & M_{2,6} & V_6 & & & \\ & M_{4,10} & M_{2,10} & V_{10} & & \\ & & \ddots & \ddots & \ddots & \\ & & & M_{4,N-8} & M_{2,N-8} & V_{N-8} \\ & & & & M_{4,j} & M_{2,j} \end{bmatrix} \begin{bmatrix} \tilde{\omega}_2 \\ \tilde{\omega}_6 \\ \tilde{\omega}_{10} \\ \vdots \\ \tilde{\omega}_{N-8} \\ \tilde{\omega}_{N-4} \end{bmatrix} = \begin{bmatrix} J_2\tilde{v}_2 \\ J_2\tilde{v}_6 \\ J_2\tilde{v}_{10} \\ \vdots \\ J_2\tilde{v}_{N-8} \\ J_2\tilde{v}_{N-4} \end{bmatrix}. \quad (15)$$

From Eq. (15) we obtain:

$$M_{2,j}\tilde{\omega}_j + V_2\tilde{\omega}_{j+4} = J_2\tilde{v}_j \qquad for \qquad j = 2. \quad (16)$$

$$M_{4,j}\tilde{\omega}_{j-4} + M_{2,j}\tilde{\omega}_j + V_6\tilde{\omega}_{j+4} = J_2\tilde{v}_j \qquad for \qquad j = 6(2)N - 8. \quad (17)$$

$$M_{4,j}\tilde{\omega}_{j-4} + M_{2,j}\tilde{\omega}_j = J_2\tilde{v}_j \qquad for \qquad j = N - 4. \quad (18)$$

From Eq. (16), we will obtain three preconditioned MEDG schemes which can be transformed into preconditioned ME DG SOR schemes as:

For $i = 2, 6(2)$ N-8 and N-4,$j = 2$

$$X\omega_{i,j}^{(k+1)} = \gamma(4F_1 + b_1F_2 + F_3\omega_{i+4,j} + F_4\omega_{i,j+4} + F_5\omega_{i+4,j+4}) + X(1-\gamma)\omega_{i,j}^{(k)}$$

$$X\omega_{i+2,j+2}^{(k+1)} = \gamma(b_2F_1 + 4F_2 + 4b_6\omega_{i+4,j} + 4b_7\omega_{i,j+4} + 4b_2\omega_{i+4,j+4}) + X(1-\gamma)$$
$$\omega_{i+2,j+2}^{(k)}, \quad (19)$$

$$X\omega_{i,j}^{(k+1)} = \gamma(2h^2c_{i,j} + b_1F_2 + F_6\omega_{i-2,j+2} + F_4\omega_{i,j+4} + b_1b_6\omega_{i+4,j} + F_5\omega_{i+4,j+4})$$
$$+ X(1-\gamma)\omega_{i,j}^{(k)}$$

$$X\omega_{i+2,j+2}^{(k+1)} = \gamma(2h^2c_{i,j}b_2 + 4F_2 + F_7\omega_{i-2,j+2} + 4b_7\omega_{i,j+4} + 4b_6\omega_{i+4,j} + 4b_8\omega_{i+4,j+4})$$
$$+ X(1-\gamma)\omega_{i+2,j+2}^{(k)} \quad (20)$$

and

$$Y\omega_{i,j}^{(k+1)} = \gamma(F_8 + F_6F_9 + 4F_{10}\omega_{i-4,j} + F_{11}\omega_{i-2,j+2} + F_{12}\omega_{i,j+4}) + Y(1-\gamma)\omega_{i,j}^{(k)}$$

$$Y\omega_{i+2,j+2}^{(k+1)} = \gamma(F_{13} + \{[1 - \sigma(\psi_{i-2,j-2} - \psi_{i+2,j+2})][1 + \sigma(\psi_{N-4,j+4} - \psi_{N,j})]\}F_{14} - \{[1 + \sigma(\psi_{N-4,j+4} - \psi_{N,j})] - F_{10}\}\omega_{i-4,j} - F_{15}\omega_{i-2,j+2} + b_7[4 - F_{10}]\omega_{i,j+4}) + Y(1-\gamma)\omega_{i+2,j+2}^{(k)}. \quad (21)$$

Similarly, From Eqs. (17) and (18), we will obtain six preconditioned MEDG schemes which can be also transformed into preconditioned MEDG SOR schemes.

In the next section, we will discuss the convergence properties of the above preconditioned MEDG SOR method.

4 Convergence Analysis of the Proposed Method

Preconditioned version of iterative methods should exhibit better convergence rates than its unpreconditioned version. This fact motivates us to present the convergence properties related to the improved convergence of the proposed preconditioned MEDG iterative method in solving the coupled equations (1)-(2). Several properties are needed to prove our main conjecture of this work.
For the sake of our discussion, we define the spectral radius of a matrix, denoted by $\rho(.)$, as the largest of the moduli of the eigenvalues of the iteration matrix. It is well known that a matrix is said to be diagonally dominant if in every row of the matrix, the magnitude of the diagonal entry in that row is larger than or equal to the sum of the magnitudes of all the other (non-diagonal) entries in that row.
More precisely, the matrix A is diagonally dominant if $|a_{ii}| \geqslant \sum_{j\neq i} |a_{ij}|$ for all i, where a_{ij} denotes the entry in the ith row and j^{th} column.

Lemma 4.1. The Successive OverRelaxation (SOR)

method is convergent for the acceleration factor when A is symmetric and positive definite or diagonally dominant. [14].

Theorem 4.1. Let $A \in \mathbb{R}^{n \times n}$ be nonsingular. Then there exist nonsingular matrices P such that PA is diagonal dominant. [15].

Corollary 4.1. The Jacobi method, the Gauss-Seidel method and the SOR method are convergent for all nonsingular linear systems in the sense of preconditioned version. [15].

The matrix A in (13) can be decomposed into $A = D - E - F$, where $D = D_j = diag(A_{11}, A_{22}, ..., A_{ii})$ and

$$E = L_j = E_{ij} = \begin{cases} -A_{ij} & for \quad j < i \\ 0 & for \quad j \geqslant i \end{cases}, \quad F = V_j = F_{ij} = \begin{cases} -A_{ij} & for \quad j > i \\ 0 & for \quad j \leqslant i \end{cases}$$

are block matrices consisting of the block diagonal, strict block lower triangular, and strict block upper triangular parts of A. Here, the diagonal entries A_{ii} are nonsingular. The block Jacobi iteration matrix is $B_J(A) = D^{-1}(E + F) = L + U$, where $L = D^{-1}E$, $U = D^{-1}F$ while the block Gauss-Seidel iteration matrix is $B_{GS}(A) = (I - L)^{-1}U$. The Block Successive Over-Relaxation method (BSOR) iteration matrix is therefore

$$G_{\ell_\gamma} = (I - \gamma L)^{-1}\{(1-\gamma)I + \gamma U\}. \quad (22)$$

The preconditioned system (15) can be written as:

$$(I + \bar{K})A\tilde{w} = (I + \bar{K})b. \quad (23)$$

Hence,

$$\bar{A}\tilde{w} = \bar{b} \quad (24)$$

where: $\bar{A} = (I + \bar{K})A = I - L - \bar{K}L - (U - \bar{K} + \bar{K}U)$ and $\bar{b} = (I + \bar{K})b$.

Similarly, the matrix $\bar{A}$ in (24) can be decomposed into $\bar{A} = I - \bar{L} - \bar{U}$, where $\bar{L} = L + \bar{K}L$ and $\bar{U} = U - \bar{K} + \bar{K}U$.

The SOR iteration matrix will result in an Enhanced Block Successive Over-Relaxation iteration matrix (EBSOR) and is given by

$$\tilde{G}_{\ell_\gamma} = \{I - \gamma\bar{L}\}^{-1}[(1-\gamma)I + \gamma\bar{U}]. \quad (25)$$

The result of Lemma 4.1, Theorem 4.1 and Corollary 4.1 enable us to prove the following theorem:

Theorem 4.2. Let $A = I - L - U$ and $\bar{A} = I - \bar{L} - \bar{U}$ be the matrices of the linear system (12) and (15) respectively. If the matrix A of (12) is diagonally dominant and $0 < \gamma < 2$, then the block SOR iteration matrices G_{ℓ_γ} and $\tilde{G}_{\ell_\gamma}$ in (22) and (25) respectively satisfied,

$$\rho(G_{\ell_\gamma}) < 1 \quad and \quad \rho(\tilde{G}_{\ell_\gamma}) < \rho(G_{\ell_\gamma}).$$

Proof.

By Lemma 4.1 and since the block SOR iteration matrix G_{ℓ_γ} in (22) is diagonally dominant, we can conclude that $\rho(G_{\ell_\gamma}) < 1$ which proved the first part of the theorem. Also from Theorem 4.1 and Corollary 4.1, it is clear that $\rho(\tilde{G}_{\ell_\gamma}) < 1$.

Now, because of the matrix A of (12) is a diagonally dominant and the iteration matrix G_{ℓ_γ} given by: $G_{\ell_\gamma} = (I - \gamma L)^{-1}\{(1-\gamma)I + \gamma U\}$, from [9] there exists a positive vector y such that $G_{\ell_\gamma}y = y\rho(G_{\ell_\gamma})$, or equivalently:

$$\{(1-\gamma)I + \gamma U\}y = y\rho(G_{\ell_\gamma})(I - \gamma L). \quad (26)$$

Therefore, we can get:

$$[\gamma\rho(G_{\ell_\gamma})L + \gamma U]y = [\rho(G_{\ell_\gamma}) - 1 + \gamma)y. \quad (27)$$

Hence,

$$\tilde{G}_{\ell_\gamma}y - y\rho(G_{\ell_\gamma}) = y\{I - \gamma\bar{L}\}^{-1}[(1-\gamma)I + \gamma\bar{U} - \rho(G_{\ell_\gamma})\{I - \gamma\bar{L}\}]$$
$$= y\{I - \gamma\bar{L}\}^{-1}[(1 - \gamma - \rho(G_{\ell_\gamma}))I + \gamma\rho(G_{\ell_\gamma})\bar{L} + \gamma\bar{U}]. \quad (28)$$

Obviously, from Eqs. (27) and (28), we can see that for $0 < \gamma < 2$, if $\rho(G_{\ell_\gamma}) < 1$ then $\rho(\tilde{G}_{\ell_\gamma}) < \rho(G_{\ell_\gamma})$. Thus, the proof is complete.

The spectral radius of the preconditioned matrix is smaller than the unpreconditioned matrix which implies that we should expect the rate of convergence of the preconditioned version to be faster than its unpreconditioned counterpart.

5 Numerical Experimentation and Results

To illustrate the results obtained in Section 4 and confirm the superiority of the proposed preconditioned scheme, numerical experiments have been carried out to solve the following Navier-Stokes equations [2],

$$\nabla^2\psi = -\omega$$
$$\nabla^2\omega + Re(\psi_x\omega_y - \psi_y\omega_x) = -1 \quad (29)$$

with the boundary conditions

$$\psi(x,0) = \psi(x,1) = \omega(x,0) = \omega(x,1) = 0, \ 0 \leqslant x \leqslant 1,$$
$$\psi(0,y) = \psi(1,y) = \omega(0,y) = \omega(1,y) = 0, \ 0 \leqslant y \leqslant 1. \quad (30)$$

and the exact solution is

$$\psi^*(x,y) = x^2(x-1)^2y^2(y-1)^2,$$
$$\omega^*(x,y) = -\nabla\psi^*(x,y). \quad (31)$$

The grid spacing used was h = 0.1 and the problem was solved for various values of Reynolds number $Re \geqslant 1$. For each case, the experimental optimum relaxation parameter γ was chosen to within 0.01 which gives the most rapid convergence. Throughout the experiment, a tolerance of $\delta = \varepsilon = 10^{-11}$ was used as the termination criteria for both the outer and inner iterations. The computer processing unit was Intel(R) Core(TM) 2Duo with memory of 3Gb and the software used to implement and generate the results was Developer C++ Version 4.9.9.2. Tables 1 and 2 list the iteration counts and timings for both the original and preconditioned MEDG SOR methods for selected Re ranging from 1 to 1000 and from 5000 to 7682 respectively. The results from preconditioned MEDG SOR scheme portray similar behavior as the original MEDG SOR. However, it can be seen that the preconditioned MEDG requires only about 50-55 $\%$ of the time required by the original method. We can also observe that the proposed preconditioned MEDG SOR scheme has shown considerable improvement in the number of iterations and execution time compared to that of the preconditioned scheme displayed in Ali and Saeed [13].

It is observed that our model problem is same as the problem in [6] when $Re \neq 0$ and $\in_1 = 0$ in Eq. (2). Tables 3 and 5 display the values of the computed values of the streamfunction ψ and the values of the exact solution $\psi*$. Tables 4 and 6 display the values of the computed values of the vorticity ω and the values of the exact solution $\omega*$. Comparisons between Tables 3 and 5 show good agreement in the results. Similarly, Tables 4 and 6 also show the computed values are very close to the exact solutions.

In order to check the efficiency and accuracy of our proposed preconditioned iterative method, we extended our experiments to larger mesh points with Re=1.0 and the error estimations are displayed in Table 7. This table shows the discrete L_2 norm $\|\psi^* - \psi\|_{L_2}$ where ψ denotes the computational solution on an n×n grid (h=1/n) and ψ^* is the exact solution. It is observed that the results behave in similar manner as in [6].

6 Conclusion

In this paper, we derive a new preconditioned Modified Explicit Decoupled Group Successive Over-Relaxation (MEDG SOR) for solving a coupled system of elliptic equations which represents the stream-vorticity formulation of the steady-state Navier-Stokes equation. The preconditioned schemes have shown improvements in the number of iterations and the execution times experimentally. The significance of this study is that the proposed preconditioner is able to substantially reduce the operation counts of the original method which confirms the more favourable spectral properties of the preconditioned scheme as proven in Theorem 4.2 in solving the two dimensional steady-state Navier-Stokes

equation for certain relaxation parameters lying in an optimum range. Hence, we conclude that the proposed preconditioner is suitable to be implemented on the newly developed MEDG SOR method resulting in a viable alternative solver for the stream-vorticity formulation of the steady-state Navier-Stokes equation.

Table 1: Iteration numbers and timings for the MEDG and Preconditioned MEDG outer-inner iterative methods for selected Re ranging from 1 to 1000

	Unpreconditioned MEDG SOR Method						Preconditioned MEDG SOR Method				
Re	Time (secs)	value of γ	No.of outer iter.(m)	No.of inner iter. of ψ	No.of inner iter. of ω	Re	Time (secs)	value of γ	No.of outer iter.(m)	No.of inner iter. of ψ	No.of inner iter. of ω
1	0.56	1.23	1	1	18	1	0.28	1.39	1	1	9
			2	16	10				2	8	7
			3	5	1				3	3	1
			4	1	1				4	1	1
10	0.59	1.23	1	1	18	10	0.30	1.39	1	1	9
			2	16	13				2	10	8
			3	7	2				3	6	2
			4	1	1				4	1	1
100	0.88	1.23	1	1	18	100	0.45	1.39	1	1	9
			2	18	16				2	11	14
			3	10	5				3	9	6
			4	3	2				4	3	1
			5	1	1				5	1	1
1000	2.48	1.34	1	1	24	1000	1.26	1.47	1	1	13
			2	21	22				2	11	10
			3	15	17				3	8	7
			4	9	10				4	6	5
			5	4	4				5	2	1
			6	1	1				6	1	1

Table 2: Iteration numbers and timings for the MEDG and Preconditioned MEDG outer-inner iterative methods for elected Re ranging from 5000 to 7682

	Unpreconditioned MEDG SOR Method						Preconditioned MEDG SOR Method				
Re	Time (secs)	value of γ	No.of outer iter.(m)	No.of inner iter. of ψ	No.of inner iter. of ω	Re	Time (secs)	value of γ	No.of outer iter.(m)	No.of inner iter. of ψ	No.of inner iter. of ω
5000	7.31	0.67	1	1	33	5000	3.86	0.90	1	1	17
			2	31	33				2	14	15
			3	23	31				3	11	12
			4	22	23				4	7	9
			5	13	16				5	3	4
			6	5	10				6	1	2
			7	2	4				7	1	1
			8	1	2						
			9	1	1						
7127	14.45	0.58	1	1	42	7682	7.94	0.73	1	1	21
			2	40	41				2	19	20
			3	30	36				3	17	19
			4	29	30				4	15	16
			5	20	23				5	12	14
			6	9	17				6	7	9
			7	5	9				7	3	5
			8	3	6				8	1	2
			9	1	3				9	1	1
			10	1	2						
			11	1	1						
>7127		Diverge				>7682		Diverge			

Table 3: Computed values of ψ at (x,y) where [x=0.2, 0.7, 0.9 and y=0.2, 0.3, 0.5, 0.7, 0.9] (Re=1000) for preconditioned MEDG method

x \ y	0.2	0.3	0.5	0.7	0.9
0.2	0.15855E-02	0.21304E-02	0.25861E-02	0.21363E-02	0.84822E-03
0.7	0.21403E-02	0.28861E-02	0.35054E-02	0.28853E-02	0.11508E-02
0.9	0.25544E-02	0.31334E-02	0.38264E-02	0.32436E-02	0.45543E-02

Table 4: Computed values of ω at (x,y) where [x=0.2, 0.7, 0.9 and y=0.2, 0.3, 0.5, 0.7, 0.9] (Re=1000) for preconditioned MEDG method

x \ y	0.2	0.3	0.5	0.7	0.9
0.2	0.352061E-01	0.434996E-01	0.506427E-01	0.443278E-01	0.210487E-01
0.7	0.434886E-01	0.556857E-01	0.645341E-01	0.556858E-01	0.267081E-01
0.9	0.642755E-01	0.676967E-01	0.776526E-01	0.637758E-01	0.347131E-01

Table 5: Exact values of $\psi*$ at (x,y) where [x=0.2, 0.7, 0.9 and y=0.2, 0.3, 0.5, 0.7, 0.9] (Re=1000)

x \ y	0.2	0.3	0.5	0.7	0.9
0.2	0.15849E-02	0.21283E-02	0.25857E-02	0.21345E-02	0.84772E-03
0.7	0.21411E-02	0.28856E-02	0.35122E-02	0.28848E-02	0.11505E-02
0.9	0.25573E-02	0.31346E-02	0.38343E-02	0.32444E-02	0.45621E-02

Table 6: Exact values of $\omega*$ at (x,y) where [x=0.2, 0.7, 0.9 and y=0.2, 0.3, 0.5, 0.7, 0.9] (Re=1000)

x \ y	0.2	0.3	0.5	0.7	0.9
0.2	0.349051E-01	0.434764E-01	0.506387E-01	0.443266E-01	0.210386E-01
0.7	0.429877E-01	0.556848E-01	0.644636E-01	0.556849E-01	0.267025E-01
0.9	0.642688E-01	0.676886E-01	0.776495E-01	0.636957E-01	0.346842E-01

Table 7: Error estimates of different mesh size h

h	$\|\psi^* - \psi\|_{L_2}$
$\frac{1}{14}$	0.855749E-05
$\frac{1}{22}$	7.588931E-06
$\frac{1}{30}$	3.911702E-06

Acknowledgement

References

[1] T. Kerh, J. J. Lee and L. C. Wellford, Predictions of Confined Shear Flows over a Wall Obstacle, International Journal of Modeling and Simulation, **15**, 23-29 (1995).

[2] M. S. Sahimi and D. J. Evans, The numerical solution of a coupled system of elliptic equations using the AGE fractional scheme, International Journal of Computer Mathematics. **50**, 65-87 (1994).

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

[4] 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, **72**, 233-251 (1999).

[5] J. Wang, W. Zhong and J. Zhang, High order compact computation and nonuniform grids for streamfunction vorticity equations, Applied Mathematics and Computation, **179**, 108-120 (2006).

[6] F. Fairag and M. S. Sahimi, The alternating group explicit (AGE) iterative method for solving a Ladyzhenskaya model for stationary incompressible viscous flow, International Journal of Computer Mathematics, **85**, 287-305 (2008).

[7] T. Kohno, H. Kotakemori and H. Niki, Improving the modified gauss-seidel method for z-matrices, Linear Algebra Appl., **267**, 113-123 (1997).

[8] W. Li and W. Sun, Modified gauss-seidel type methods and jacobi type methods for z-matrix, Linear Algebra Appl., **317**, 227-240 (2000).

[9] M. M. Martins, D. J. Evans, and W. Yousif, Further results on the preconditioned SOR method, International Journal of Computer Mathematics, **77**, 603-610 (2001).

[10] A. Hadjidimos, D. Noutsos and M. Tzoumas, More on modifications and improvements of classical iterative schemes for m-matrices, Linear Algebra Appl., **364**, 253-279 (2003).

[11] Q. Liu, G. Chen and J. Cai, Convergence analysis of the preconditioned gauss-seidel method for h-matrices, Computers and Mathematics with Applications. **56**, 2048-2053 (2008).

[12] A. M. Saeed and N. H. M. Ali, Preconditioned Modified Explicit Decoupled Group Method In The Solution Of Elliptic PDEs, Applied Mathematical Sciences, **4**, 1165-1181 (2010).

[13] N. H. M. Ali and A. M. Saeed , New Preconditioned Group Methods for the Solution of a Coupled System of Partial Differential Equations, International Journal of Mathematics and Computation, **10**, 137-149 (2011).

[14] D. Young, Iterative solutions of large linear systems (Academic Press, New York), (1971).

[15] J. Y. Yuan, Preconditioned diagonal dominant matrices, Applied Mathematics and Computation, **114**, 255-262 (2000).

Dr. Norhashidah Hj. Mohd. Ali obtained her PhD in Industrial Computing from Universiti Kebangsaan Malaysia. She received her MSc in Applied Mathematics from Virginia Tech, U.S.A. and her BSc. (High Honors) in Mathematics from Western Illinois University, U.S.A. She is an experienced researcher in the areas of Numerical Partial Differential Equations and Parallel Numerical Algorithms. She is an Associate Professor attached to the School of Mathematical Sciences, Universiti Sains Malaysia, Penang.

Dr. Abdulkafi M. Saeed obtained his Ph. D. in Partial Differential Equations (PDEs) from Universiti Sains Malaysia. He received his Master degree in Applied Mathematics from Hyderabad Central University in India. He has taught for four years in Hodeidah University in Yemen, then, started his Post-Doctoral in the School of Mathematical Sciences, Universiti Sains Malaysia, Penang for one year. Has experience in Applied Mathematics with emphasis in PDEs. His current research interests include the numerical study of solutions of PDEs that model physical phenomena in Fluid Dynamics and Electromagnetic field. He has joined several conferences in connection with research collaboration. He is an Assistant Professor attached to the Department of Mathematics, College of Science, Qassim University, Saudi Arabia.