Applying the MEDG Iterative Method to Nonlinear Steady Burgers' Equation: A Numerical Approach

Dr. Fatima Al-Farsi
Associate Professor, Department of Biotechnology, Sultan Qaboos University, Oman

KEYWORDS: Modified Explicit Decoupled Group Iterative Method, Burgers' Equation.

ABSTRACT

Explicit Decoupled Group (EDG) iterative methods were found more superior than common existing methods based on the centered five-point difference schemes for solving two dimensional Burgers' Equation. In this article, the new formulation of the Modified Explicit Decoupled Group (MEDG) scheme from the rotated finite difference discretization to the numerical solution of the nonlinear steady two dimensional Burgers' Equation is presented. The numerical experiments carried out confirm the superiority of the proposed method over the latter in terms of number of iterations and execution time.

INTRODUCTION

Several group iterative methods have been developed for solving many types of partial differential equations for the last 15 years, but this quest is still going on ([1], [2], [3], [4], [5]).

The suitability of the four-point Explicit Decoupled Group (EDG) iterative method in solving a two-dimensional steady-state Navier-Stokes equation was demonstrated in Ali and Abdullah [6] where the iterative group scheme derived from the rotated five-point formula converges more rapidly than the standard point scheme based on the centred difference formula. Ali and Abdullah [7] introduced Explicit Decoupled Group (EDG) iterative schemes based on the rotated finite difference discretisation in solving the two dimensional steady Burgers' equation. The numerical experiments of this work yield very encouraging results.

This paper will contribute to improve the proposed EDG iterative schemes to the most superior Group iterative method namely Modified Explicit Decoupled Group (MEDG) [8], for solving the two dimensional steady Burgers' equation.

Consider steady two dimensional Burgers' equation [7] as the following:

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}-\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)=0 \qquad (1)$$

$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}-\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)=0 \qquad (2)$$

with Dirichlet boundary conditions on *u* and v. This equation is considered to be a simplified form of the Navier-Stokes equation, where the pressure term is neglected. Here, Re is the Reynolds number. For our discussion, the solutions u and v that satisfy these equations are sought in the interior region S as shown in figure

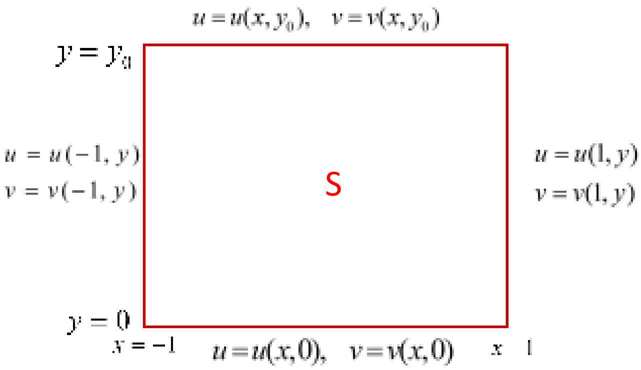

Figure 1: Solution domain of Burgers' equation.

Let n be a fixed positive integer. Determine the grid size h = 2/n so that a uniformly spaced square network ($\Delta x = \Delta y = h$) with $x_i = -1 + ih,\ y = jh, i, j = 0,1,2,...,n,$ is imposed on S. By using the centred difference approximation and neglecting the error terms, equations (1) and (2) can be discretised at the grid points (x_i, y_j) by the following finite difference equations:

$$[\frac{h \operatorname{Re}(u_{i+1,j} - u_{i-1,j}) + 8}{2 \operatorname{Re} h^2}]u_{ij} = \frac{(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1})}{\operatorname{Re} h^2} - v_{ij}(\frac{u_{i,j+1} - u_{i,j-1}}{2h}) \tag{3}$$

$$[\frac{h \operatorname{Re}(v_{i,j+1} - v_{i,j-1}) + 8}{2 \operatorname{Re} h^2}]v_{ij} = \frac{(v_{i-1,j} + v_{i+1,j} + v_{i,j-1} + v_{i,j+1})}{\operatorname{Re} h^2} - u_{ij}(\frac{v_{i+1,j} - v_{i-1,j}}{2h}) \tag{4}$$

It can be seen that if v is known, then we can solve (3) iteratively for u, while if u is known, we can solve (4) iteratively for v, and vice versa. By the same manner of the schemes presented for the Navier-Stokes problem, we can devise a similar algorithm by first making initial guesses u_{ij}^0 and v_{ij}^0, and then generate an alternating sequence of outer iterates. The iteration is continued until for some k such that $\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right| < \varepsilon$ and $\left|v_{ij}^{(k+1)} - v_{ij}^{(k)}\right| < \varepsilon$ for some given tolerance (ε). The solutions $u_{ij}^{(k+1)}$ and $v_{ij}^{(k+1)}$ generated are then taken to be the numerical solutions of the given problem ([6], [9], [10]).

The outline of this paper is as follows: An overview of the formulation of EDG iterative method for solving the two dimensional steady Burgers' equation will be given in Section 2. The formulation of MEDG method for solving the mentioned Burgers' equation is presented in Section 3. The numerical results are discussed to show the efficiency of the proposed MEDG method in Section 4 and the concluding remarks is given in Section 5.

EDG iterative method for solving the two dimensional steady Burgers' equation

Another type of approximation that can represent the Burgers' equation under study is the cross orientation which can be obtained by rotating the i-plane axis and the j-plane axis clockwise by 45° as the following:

$$[\frac{h \operatorname{Re}(u_{i+1,j-1} - u_{i-1,j+1} + u_{i+1,j+1} - u_{i-1,j-1}) + 8}{2 \operatorname{Re} h^2}]u_{ij} + (\frac{v_{ij}}{4h} - \frac{1}{2 \operatorname{Re} h^2})u_{i-1,j+1} +$$
$$(\frac{-v_{ij}}{4h} - \frac{1}{2 \operatorname{Re} h^2})u_{i+1,j-1} + (\frac{-v_{ij}}{4h} - \frac{1}{2 \operatorname{Re} h^2})u_{i-1,j-1} + (\frac{v_{ij}}{4h} - \frac{1}{2 \operatorname{Re} h^2})u_{i+1,j+1} = 0$$

(5)

$$[\frac{h \operatorname{Re}(v_{i+1,j+1} - v_{i-1,j-1} - v_{i+1,j-1} + v_{i-1,j-1}) + 8}{4 \operatorname{Re} h^2}]u_{ij} + (\frac{-u_{ij}}{4h} - \frac{1}{2 \operatorname{Re} h^2})v_{i-1,j+1} +$$
$$(\frac{u_{ij}}{4h} - \frac{1}{2 \operatorname{Re} h^2})v_{i+1,j-1} + (\frac{-u_{ij}}{4h} - \frac{1}{2 \operatorname{Re} h^2})v_{i-1,j-1} + (\frac{u_{ij}}{4h} - \frac{1}{2 \operatorname{Re} h^2})v_{i+1,j+1} = 0$$

(6)

The four-point EDG for solving the problem (1)-(1) can now formulated by using the above rotated finite difference approximation (5)-(6). Without loss of generality, assume the generation of $v_{ij}^{(k+1)}$ is done first using equation (2.2) followed by the generation of $u_{ij}^{(k+1)}$ using equation (5).Using the equation (6) for v_{ij} any group of four points on a discretised solution domain can be solved resulting in a (4×4) system of equations as the following [7]:

$$A \times \begin{bmatrix} v_{ij} \\ v_{i+1,j+1} \\ v_{i+1,j} \\ v_{i,j+1} \end{bmatrix} = \begin{bmatrix} r\,hs_{ij} \\ r\,hs_{i+1,j+1} \\ r\,hs_{i+1,j} \\ r\,hs_{i,j+1} \end{bmatrix} \tag{7}$$

Where

$$A=\begin{bmatrix} \frac{h\,\mathrm{Re}(v_{i+1,j+1}-v_{i-1,j-1}+v_{i+1,j-1}+v_{i-1,j+1})+8}{4\,\mathrm{Re}\,h^2} & cu_{ij}-d & 0 & 0 \\ -cu_{i+1,j+1}-d & \frac{h\,\mathrm{Re}(v_{i+2,j+2}-v_{i,j}+v_{i+2,j}-v_{i,j+2})+8}{4\,\mathrm{Re}\,h^2} & 0 & 0 \\ 0 & 0 & \frac{h\,\mathrm{Re}(v_{i+2,j+1}-v_{i,j-1}-v_{i+2,j-1}+v_{i,j+1})+8}{4\,\mathrm{Re}\,h^2} & -cu_{i+1,j}-d \\ 0 & 0 & cu_{i,j+1}-d & \frac{h\,\mathrm{Re}(v_{i+1,j+2}-v_{i-1,j}-v_{i+1,j}+v_{i-1,j+2})+8}{4\,\mathrm{Re}\,h^2} \end{bmatrix}$$

$$r\,hs_{ij}=(cu_{ij}+d)v_{i-1,j+1}+(-cu_{ij}+d)v_{i-1,j-1}+(cu_{ij}+d)v_{i-1,j-1},$$
$$r\,hs_{i+1,j+1}=(cu_{i+1,j+1}+d)v_{i,j+2}+(-cu_{i+1,j+1}+d)v_{i+2,j}+(-cu_{i+1,j+1}+d)v_{i+2,j+2},$$
$$r\,hs_{i+1,j}=(-cu_{i+1,j}+d)v_{i+2,j+1}+(-cu_{i+1,j}+d)v_{i+2,j-1}+(cu_{i+1,j}+d)v_{i,j-1},$$
$$r\,hs_{i,j+1}=(cu_{i,j+1}+d)v_{i-1,j+2}+(-cu_{i,j+1}+d)v_{i+1,j+2}+(cu_{i,j+1}+d)v_{i-1,j},$$
$$c=\frac{1}{4h},\quad d=\frac{1}{2\,\mathrm{Re}\,h^2}\ .$$

The system (7) leads to a decoupled system of (2×2) equations which can be made explicit as follows:

$$\begin{bmatrix} v_{ij} \\ v_{i+1,j+1} \end{bmatrix}^{(k+1)}=\frac{(4\,\mathrm{Re}\,h^2)^2}{[\mathrm{Re}\,h(v^k_{i+1,j+1}-v_{i-1,j-1}-v_{i+1,j-1}+v_{i-1,j+1})+8][\mathrm{Re}\,h(v^k_{i+2,j+2}-v^k_{i,j}-v_{i+2,j}+v_{i,j+2})+8]}$$
$$\times\begin{bmatrix} \frac{\mathrm{Re}\,h(v_{i+2,j+2}-v^k_{ij}-v_{i+2,j}+v_{i,j+2})+8}{4\,\mathrm{Re}\,h^2} & -cu_{ij}+d \\ cu_{i+1,j+1}+d & \frac{\mathrm{Re}\,h(v^k_{i+1,j+1}-v_{i-1,j-1}-v_{i+1,j-1}+v_{i-1,j+1})+8}{4\,\mathrm{Re}\,h^2} \end{bmatrix}\begin{bmatrix} rhs_{ij} \\ rhs_{i+1,j+1} \end{bmatrix} \qquad (8)$$

and

$$\begin{bmatrix} v_{i+1,j} \\ v_{i,j+1} \end{bmatrix}^{(k+1)}=\frac{(4\,\mathrm{Re}\,h^2)^2}{[\mathrm{Re}\,h(v^k_{i+1,j+1}-v_{i-1,j-1}-v_{i+1,j-1}+v_{i-1,j+1})+8][\mathrm{Re}\,h(v_{i+2,j+2}-v^k_{i,j}-v_{i+2,j}+v_{i,j+2})+8]+(h\,\mathrm{Re}\,u_{i+1,j+1}+2)(h\,\mathrm{Re}\,u_{ij}-2)}$$
$$\times\begin{bmatrix} \frac{\mathrm{Re}\,h(v_{i+1,j+2}-v^k_{i+1,j}-v_{i-1,j}+v_{i-1,j+2})+8}{4\,\mathrm{Re}\,h^2} & cu_{i+1,j}+d \\ -cu_{i,j+1}+d & \frac{\mathrm{Re}\,h(v^k_{i,j+1}-v_{i,j-1}-v_{i+2,j-1}+v_{i+2,j+1})+8}{4\,\mathrm{Re}\,h^2} \end{bmatrix}\begin{bmatrix} rhs_{i+1,j} \\ rhs_{i,j+1} \end{bmatrix}$$

(9)

By the same manner, from the generation of u_{ij} using equation (5), a (4×4) system of equations can be formed as the following:

$$B\times\begin{bmatrix} u_{ij} \\ u_{i+1,j+1} \\ u_{i+1,j} \\ u_{i,j+1} \end{bmatrix}=\begin{bmatrix} r\,hs_{ij} \\ r\,hs_{i+1,j+1} \\ r\,hs_{i+1,j} \\ r\,hs_{i,j+1} \end{bmatrix} \qquad (10)$$

Where

$$A=\begin{bmatrix} \frac{h\,\mathrm{Re}(u_{i+1,j-1}-u_{i-1,j+1}+u_{i+1,j+1}-u_{i-1,j-1})+8}{4\,\mathrm{Re}\,h^2} & cv_{ij}-d & 0 & 0 \\ -cv_{i+1,j+1}-d & \frac{h\,\mathrm{Re}(u_{i+2,j}-u_{i,j+2}+u_{i+2,j+2}-u_{ij})+8}{4\,\mathrm{Re}\,h^2} & 0 & 0 \\ 0 & 0 & \frac{h\,\mathrm{Re}(u_{i+2,j-1}-u_{i,j+1}+u_{i+2,j+1}-u_{i,j-1})+8}{4\,\mathrm{Re}\,h^2} & cv_{i+1,j}-d \\ 0 & 0 & -cv_{i,j+1}-d & \frac{h\,\mathrm{Re}(u_{i+1,j}-u_{i-1,j+2}+u_{i+1,j+2}-u_{i-1,j})+8}{4\,\mathrm{Re}\,h^2} \end{bmatrix},$$

$$rhs_{ij} = (-cv_{ij} + d)u_{i-1,j+1} + (cv_{ij} + d)u_{i+1,j-1} + (cv_{ij} + d)u_{i-1,j-1},$$
$$rhs_{i+1,j+1} = (-cv_{i+1,j+1} + d)u_{i,j+2} + (cv_{i+1,j+1} + d)u_{i+2,j} + (-cv_{i+1,j+1} + d)u_{i+2,j+2},$$
$$rhs_{i+1,j} = (-cv_{i+1,j} + d)u_{i+2,j+1} + (cv_{i+1,j} + d)u_{i+2,j-1} + (cv_{i+1,j} + d)u_{i,j-1},$$
$$rhs_{i,j+1} = (-cv_{i,j+1} + d)u_{i-1,j+2} + (-cv_{i,j+1} + d)u_{i+1,j+2} + (cv_{i,j+1} + d)u_{i-1,j}.$$

The system (10) leads to a decoupled system of (2×2) equations whose explicit forms can be obtained as follows:

$$\begin{bmatrix} u_{ij} \\ u_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{(4\,\mathrm{Re}\,h^2)^2}{[\mathrm{Re}\,h(u_{i+1,j-1} - u_{i-1,j+1} + u^k_{i+1,j+1} - u_{i-1,j-1}) + 8][\mathrm{Re}\,h(u_{i+2,j} - u^k_{i,j} - u_{i,j+2} + u_{i+2,j+2}) + 8] + (\mathrm{Re}\,h\,v_{i+1,j+1} + 2)(\mathrm{Re}\,h\,v_{ij} - 2)} \times \begin{bmatrix} \dfrac{\mathrm{Re}\,h(u_{i+2,j} - u_{i,j+2} + u_{i+2,j+2} - u_{ij}) + 8}{4\,\mathrm{Re}\,h^2} & -cv_{ij} + d \\ cv_{i+1,j+1} + d & \dfrac{\mathrm{Re}\,h(u^k_{i+1,j-1} - u_{i-1,j+1} + u_{i+1,j+1} - u_{i-1,j-1}) + 8}{4\,\mathrm{Re}\,h^2} \end{bmatrix} \begin{bmatrix} rhs_{ij} \\ rhs_{i+1,j+1} \end{bmatrix}$$

(11)

and

$$\begin{bmatrix} u_{i+1,j} \\ u_{i,j+1} \end{bmatrix}^{(k+1)} = \frac{(4\,\mathrm{Re}\,h^2)^2}{[h\,\mathrm{Re}(u_{i+2,j-1} + u_{i+2,j+1} - u^k_{i,j+1} - u_{i,j-1}) + 8][h\,\mathrm{Re}(u^k_{i+1,j} - u_{i-1,j+2} - u_{i-1,j} + u_{i+1,j+2}) + 8] + (h\,\mathrm{Re}\,v_{i,j+1} + 2)(h\,\mathrm{Re}\,u_{i+1,j} - 2)} \times \begin{bmatrix} \dfrac{\mathrm{Re}\,h(u_{i+1,j} - u_{i-1,j+2} + u_{i+1,j+2} - u_{i-1,j}) + 8}{4\,\mathrm{Re}\,h^2} & -cv_{i+1,j} + d \\ cv_{i,j+1} + d & \dfrac{\mathrm{Re}\,h(u_{i+2,j-1} - u_{i,j+1} + u_{i+2,j+1} - u_{i,j-1}) + 8}{4\,\mathrm{Re}\,h^2} \end{bmatrix} \begin{bmatrix} rhs_{i+1,j} \\ rhs_{i,j+1} \end{bmatrix}$$

(12)

It can be seen that the computational molecule for this scheme is similar to the ones described in the EDG method for problems in [11]. From Figure 1, it may be observed that the computational molecule for this scheme is similar to the ones described in the EDG method for problems in [11]. The computational molecule of equations (8), (9), (2.7) and (2.8) are given in figure 1.

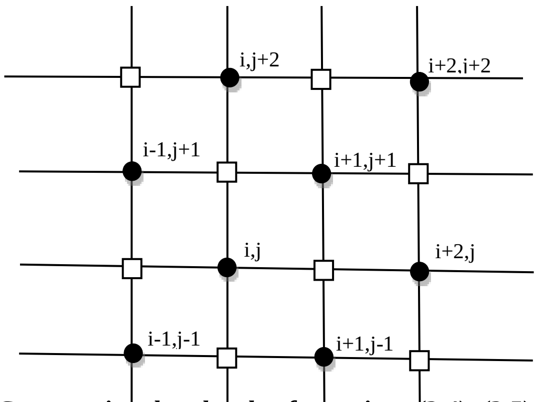

Figure 1 Computational molecule of equations (2.4), (2.5), (2.7) and (2.8)

Note that for both equations, iterative evaluation of points from each group requires contribution of points only from the same group. This means the iterative evaluation of equations (8) and (11) involve points of type ● only, while the iterations arise from equations (9) and (12) 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 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 total nodal points [9]. After convergence is achieved, the solution at the other half of the points is evaluated directly once using the centred difference formulae (3) or (4).

The formulation of MEDG method for solving the Burgers' equation

The Modified Explicit Decoupled Group method is modification of the EDG method described by considering the points with grid spacing 2h = 2L/n. When i-plane axis and the j-plane axis are rotated clockwise 45^0 with grid spacing 2h, then equations (8) and (9) become as the following equations:

$$\begin{bmatrix} v_{ij} \\ v_{i+2,j+2} \end{bmatrix}^{(k+1)} = \frac{(4\operatorname{Re}h^2)^2}{[\operatorname{Re}h(v^k_{i+2,j+2} - v_{i-2,j-2} - v_{i+2,j-2} + v_{i-2,j+2}) + 8][\operatorname{Re}h(v^k_{i+4,j+4} - v^k_{i,j} - v_{i+4,j} + v_{i,j+4}) + 8]} \times \begin{bmatrix} \frac{\operatorname{Re}h(v_{i+4,j+4} - v^k_{ij} - v_{i+4,j} + v_{i,j+4}) + 8}{4\operatorname{Re}h^2} & -cu_{ij} + d \\ cu_{i+2,j+2} + d & \frac{\operatorname{Re}h(v^k_{i+2,j+2} - v_{i-2,j-2} - v_{i+2,j-2} + v_{i-2,j+2}) + 8}{4\operatorname{Re}h^2} \end{bmatrix} \begin{bmatrix} rhs_{ij} \\ rhs_{i+2,j+2} \end{bmatrix}$$

$$\begin{bmatrix} v_{i+2,j} \\ v_{i,j+2} \end{bmatrix}^{(k+1)} = \frac{(4\operatorname{Re}h^2)^2}{[\operatorname{Re}h(v^k_{i+2,j+2} - v_{i-2,j-2} - v_{i+2,j-2} + v_{i-2,j+2}) + 8][\operatorname{Re}h(v_{i+4,j+4} - v^k_{i,j} - v_{i+4,j} + v_{i,j+4}) + 8] + (h\operatorname{Re}u_{i+2,j+2} + 2)(h\operatorname{Re}u_{ij} - 2)} \times \begin{bmatrix} \frac{\operatorname{Re}h(v_{i+2,j+4} - v^k_{i+2,j} - v_{i-2,j} + v_{i-2,j+4}) + 8}{4\operatorname{Re}h^2} & cu_{i+2,j} + d \\ -cu_{i,j+2} + d & \frac{\operatorname{Re}h(v^k_{i,j+2} - v_{i,j-2} - v_{i+4,j-2} + v_{i+4,j+2}) + 8}{4\operatorname{Re}h^2} \end{bmatrix} \begin{bmatrix} rhs_{i+2,j} \\ rhs_{i,j+2} \end{bmatrix}$$

Also, equations (11) and (12) become as the following

$$\begin{bmatrix} u_{ij} \\ u_{i+2,j+2} \end{bmatrix}^{(k+1)} = \frac{(4\operatorname{Re}h^2)^2}{[\operatorname{Re}h(u_{i+2,j-2} - u_{i-2,j+2} + u^k_{i+2,j+2} - u_{i-2,j-2}) + 8][\operatorname{Re}h(u_{i+4,j} - u^k_{i,j} - u_{i,j+4} + u_{i+4,j+4}) + 8] + (\operatorname{Re}hv_{i+2,j+2} + 2)(\operatorname{Re}hv_{ij} - 2)} \times \begin{bmatrix} \frac{\operatorname{Re}h(u_{i+4,j} - u_{i,j+4} + u_{i+4,j+4} - u_{ij}) + 8}{4\operatorname{Re}h^2} & -cv_{ij} + d \\ cv_{i+2,j+2} + d & \frac{\operatorname{Re}h(u^k_{i+2,j-2} - u_{i-2,j+2} + u_{i+2,j+2} - u_{i-2,j-2}) + 8}{4\operatorname{Re}h^2} \end{bmatrix} \begin{bmatrix} rhs_{ij} \\ rhs_{i+2,j+2} \end{bmatrix}$$

$$\begin{bmatrix} u_{i+2,j} \\ u_{i,j+2} \end{bmatrix}^{(k+1)} = \frac{(4\operatorname{Re}h^2)^2}{[h\operatorname{Re}(u_{i+4,j-2} + u_{i+4,j+2} - u^k_{i,j+2} - u_{i,j-2}) + 8][h\operatorname{Re}(u^k_{i+2,j} - u_{i-2,j+4} - u_{i-2,j} + u_{i+2,j+4}) + 8] + (h\operatorname{Re}v_{i,j+2} + 2)(h\operatorname{Re}u_{i+2,j} - 2)} \times \begin{bmatrix} \frac{\operatorname{Re}h(u_{i+2,j} - u_{i-2,j+4} + u_{i+2,j+4} - u_{i-2,j}) + 8}{4\operatorname{Re}h^2} & -cv_{i+2,j} + d \\ cv_{i,j+2} + d & \frac{\operatorname{Re}h(u_{i+4,j-2} - u_{i,j+2} + u_{i+4,j+2} - u_{i,j-2}) + 8}{4\operatorname{Re}h^2} \end{bmatrix} \begin{bmatrix} rhs_{i+2,j} \\ rhs_{i,j+2} \end{bmatrix}$$

NUMERICAL RESULTS AND DISCUSSION

Numerical experiments have been carried out to solve the Burgers' Equations (1)-(2) with the exact solution

$$u = \frac{-2(a_2 + a_4 y + \lambda a_5 \cos\lambda y(e^{\lambda(x-x_0)} - e^{-\lambda(x-x_0)}))}{\operatorname{Re}(a_1 + a_2 x + a_3 y + a_4 xy + a_5(e^{\lambda(x-x_0)} + e^{-\lambda(x-x_0)})\cos\lambda y)}$$

$$v = \frac{-2(a_3 + a_4 x - \lambda a_5 \sin\lambda y(e^{\lambda(x-x_0)} + e^{-\lambda(x-x_0)}))}{\operatorname{Re}(a_1 + a_2 x + a_3 y + a_4 xy + a_5(e^{\lambda(x-x_0)} + e^{-\lambda(x-x_0)})\cos\lambda y)}, \quad -1 \le x \le 1,\ 0 \le y \le 2$$

with the boundary conditions satisfying the exact solutions. Here, $a_1, a_2, a_3, a_4, a_5, \lambda$ and x_0 can be chosen to produce different behavior of exact solutions [12]. To compare the numerical results among EDG iterative method, MEDG iterative method with the previous work [7], we randomly chose $a_1 = a_2 = 1.0,\ a_3 = a_4 = 0.0,\ a_5 = x_0 = 1.0$

and $\lambda = 0.3$ for $\operatorname{Re} = 10, 100$ and 1000. Throughout the experiment, a tolerance of $\delta = \varepsilon = 10^{-11}$ was used as the termination criteria for both the outer and inner iterations. 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 EDG and MEDG iterative methods respectively.

The results from MEDG scheme portray similar behavior as the EDG. However, it can be seen that the MEDG requires only about 60-72% of the time required by the EDG system. Furthermore, the iteration count for the MEDG system increases at a slower rate than the EDG system.

CONCLUSION

In this paper, we derive new Modified Explicit Decoupled Group (MEDG) for solving two dimensional steady Burgers' Equation. The MEDG scheme has shown improvements in the number of iterations and the execution time experimentally. Hence, we conclude that the proposed MEDG iterative method is superior to EDG method for solving the two dimensional Burgers' Equation. However the estimation of implementing the family of explicit group methods for solving another type of Partial Differential Equations such as the Fractional Partial Differential Equations (FPDEs) remains a challenging task and therefore it will be a worthwhile effort to venture more into this group iterative method.

Table 1. Iterative count and Elapsed Time for 4-points EDG outer-inner scheme

N	Re	Ave-Abs. Error for u	Ave-Abs. Error for v	Number of outer iterates	Number of inner iteration for v	Number of inner iteration for u	Time (secs)
	10	1.38E-07	1.16E-07	1 2 3 4	49 33 22 1	51 30 17 1	22.23
41	100	1.36E-08	1.05E-08	1 2 3 4	41 30 17 1	44 23 9 1	20.08
	1000	1.38E-09	1.65E-09	1 2 3 4	33 25 12 1	31 19 1 1	18.74
	10	6.77E-08	4.62E-08	1 2 3 4	86 53 28 1	79 47 18 1	79.68
67	100	6.69E-09	4.39E-09	1 2 3 4	53 25 12 1	41 19 7 1	59.69
	1000	6.85E-10	4.18E-10	1 2 3 4	49 24 10 1	38 18 1 1	48.56
	10	3.37E-08	1.85E-08	1 2 3 4	151 122 41 1	143 87 23 1	146.44
111	100	3.22E-09	1.47E-09	1 2 3 4	124 81 35 1	117 66 1 1	126.32
	1000	3.09E-10	1.14E-10	1 2 3 4	99 62 24 1	107 43 1 1	115.47

Table 2. Iterative count and Elapsed Time for 4-points MEDG outer-inner scheme

N	Re	Ave-Abs. Error for u	Ave-Abs. Error for v	Number of outer iterates	Number of inner iteration for v	Number of inner iteration for u	Time (secs)
41	10	1.21E-07	1.08E-07	1 2 3 4	37 23 15 1	39 21 13 1	16.05
	100	1.18E-08	1.03E-08	1 2 3 4	29 20 13 1	32 15 7 1	14.24
	1000	1.29E-09	1.48E-09	1 2 3 4	22 19 8 1	24 15 1 1	12.35
67	10	7.01E-08	4.14E-08	1 2 3 4	56 43 26 1	58 35 14 1	47.58
	100	6.98E-09	4.24E-09	1 2 3 4	31 18 9 1	35 13 5 1	31.14
	1000	6.97E-10	4.05E-10	1 2 3 4	27 14 8 1	29 17 1 1	29.22
111	10	3.46E-08	1.75E-08	1 2 3 4	98 83 36 1	104 77 18 1	105.73
	100	3.39E-09	1.41E-09	1 2 3 4	76 50 16 1	81 51 1 1	92.84
	1000	3.19E-10	1.08E-10	1 2 3 4	65 41 16 1	73 32 1 1	86.58

ACKNOWLEDGEMENTS

The author gratefully acknowledges Qassim University, represented by the Deanship of Scientific Research, on the material support for this research under the number (1123-cos-2016-12-s) during the academic year 1438 AH/2017AD).

REFERENCES

1. M. Othman and A. R. Abdullah, "An Efficient Four Points Modified Explicit Group Poisson Solver," International Journal Of Computer Mathematics, VOL. 76, pp. 203 - 217, 2000.
2. N. H. M. Ali, A. R. Abdullah and J. L. Kok, "A comparative study of explicit group iterative solvers on a cluster of workstations," Parallel Algorithms and Applications, VOL. 19, NO. 4, pp. 237 - 255, 2004.
3. A. M. Saeed and N. H. M. Ali, "On the Convergence of the Preconditioned Group Rotated Iterative Methods In The Solution of Elliptic PDEs," Applied Mathematics & Information Sciences, VOL. 5,1, pp. 65 - 73, 2011.
4. A. M. Saeed and, "Efficient Group Iterative Method for Solving the Biharmonic Equation," British Journal of Mathematics & Computer Science, VOL. 9, 3, pp. 237-245, 2015.

5. A. M. Saeed and, "Solving Poisson's Equation Using Preconditioned Nine-Point Group SOR Iterative Method ," International Journal of Mathematics and Statistics Invention (IJMSI), VOL. 4,8, pp. 20-26, 2016.
6. N. H. M. Ali and A. R. Abdullah, "New Rotated Iterative Algorithms for the Solution of a coupled system of elliptic equations," International Journal of Computer Mathematics, VOL. 74, pp. 223-251, 1999.
7. N. H. M. Ali and A. R. Abdullah, "Rotated Outer-Inner Iterative Schemes for The Solution of Burgers' Equation," Matematika, VOL. 21, 1, pp. 61 - 77, 2005.
8. N. H. M. Ali and K. F. Ng, "A New Iterative Elliptic Solver On A Distributed PC Cluster,", Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 47-53, Dec. 2008.
9. 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, VOL. 10,M11, pp. 137-149, 2011.
10. N. H. M. Ali and A. M. Saeed, "Preconditioned Modified Explicit Decoupled Group for the Solution of Steady State Navier-Stokes Equation," Applied Mathematics & Information Sciences, VOL. 7, 5, pp. 1837-1844, 2013.
11. A. R. Abdullah, "The Four Point Explicit Decoupled Group (EDG) Method: A Fast Poisson Solver," International Journal Of Computer Mathematics, VOL. 38, pp. 61 - 70, 1991.
12. C. A. J. Fletcher, Computational Techniques for Fluid Dynamics 1, Springer-Verlag, Berlin, 1991.