World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:5, No:12, 2011

New Explicit Group Newton's Iterative Methods for the Solutions of Burger's Equation

Tan K. B. and Norhashidah Hj. M. Ali

***Abstract*—**In this article, we aim to discuss the formulation of two explicit group iterative finite difference methods for time-dependent two dimensional Burger's problem on a variable mesh. For the non-linear problems, the discretization leads to a non-linear system whose Jacobian is a tridiagonal matrix. We discuss the Newton's explicit group iterative methods for a general Burger's equation. The proposed explicit group methods are derived from the standard point and rotated point Crank-Nicolson finite difference schemes. Their computational complexity analysis is discussed. Numerical results are given to justify the feasibility of these two proposed iterative methods.

***Keywords*—**Standard point Crank-Nicolson (CN), Rotated point Crank-Nicolson (RCN), Explicit Group (EG), Explicit Decoupled Group (EDG).

I. Introduction

THIS paper is an extension of our previous work in [1] on the explicit group methods for time-dependent two dimensional convection-diffusion equations. The explicit group (EG) iterative scheme was first introduced by Yousif and Evans [2] in solving the two dimensional elliptic problems by grouping the mesh points into smaller size groups of points. In 1991, Abdullah [3] developed the explicit decoupled group (EDG) iterative scheme by using the same ideas of explicit group on rotated grid. These explicit group methods have been extensively investigated over the years in solving various types of equations [4] – [7]. These methods are easy to implement and require lesser computational cost and they are suitable to be used on parallel computers. In [1] we have applied the EG and EDG iterative schemes to solve the two dimensional convection-diffusion problems. We also discussed the stability and consistency of EG and EDG schemes.

Burger's equation is an important partial differential equation in fluid mechanics. It has been widely used for various applications, such as mathematical model of turbulence [8], shock wave [9] and traffic flow. There are many numerical methods for solving Burger's equation, such as finite difference scheme, the Eulerian-Lagrangian method, variational iteration method, the lattice Boltzmann method and so on.

In this paper, we introduce the explicit group Newton's iterative methods for the solution of the time-dependent two dimensional Burger's equation. In the next section, we will describe the formulation of these explicit group methods followed by the computational complexity analysis in Section III. The Numerical experiments and results are presented in Section IV. The concluding remark is given in Section V.

K. B. Tan is a doctoral candidate at the School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia (phone: +6-016-4210353; e-mail: avery701040@gmail.com).

Norhashidah Hj. M. Ali is an Associate Professor at the School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia (phone: +6-04-6533960; e-mail: shidah@cs.usm.my).

II. The Explicit Group Newton's Iterative Methods

A. Problem statement Review Stage

In this paper, we introduce the explicit group Newton's iterative methods for the Burger's equation based on finite difference approximation. We consider the time-dependent two-dimensional Burger's equation as follows:

$$\frac{\partial u}{\partial t}+u\left(\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}\right)=\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \tag{1}$$

subject to the initial conditions

$$u(x,y,0)=u_0(x,y),\quad (x,y)\in D. \tag{2}$$

The boundary conditions

$$u(x,y,t)=f(x,y,t),(x,y,t)\in \partial D\times(0,T] \tag{3}$$

where $D=[a,b]\times[c,d]$ is a rectangular domain and ∂D is its boundary.

We define that

$$F=0.5\mathrm{u}^2. \tag{4}$$

Then, Equation (1) is equivalent to

$$\frac{\partial u}{\partial t}+\left(\frac{\partial F}{\partial x}+\frac{\partial F}{\partial y}\right)=\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \tag{5}$$

or

$$\frac{\partial u}{\partial t}+0.5\left(\frac{\partial u^2}{\partial x}+\frac{\partial u^2}{\partial y}\right)=\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \tag{6}$$

B. Standard Point Crank-Nicolson (CN) Newton's Iterative Method

The common Crank-Nicolson finite difference formula in discretizing (6) is:

$$u_{i,j}^{k+1}-u_{i,j}^{k}+\frac{\Delta t}{4}\left(\frac{u_{i+1,j}^{k+1}u_{i+1,j}^{k+1}-u_{i-1,j}^{k+1}u_{i-1,j}^{k+1}}{2\Delta x}+\frac{u_{i+1,j}^{k}u_{i+1,j}^{k}-u_{i-1,j}^{k}u_{i-1,j}^{k}}{2\Delta x}\right.$$
$$\left.+\frac{u_{i,j+1}^{k+1}u_{i,j+1}^{k+1}-u_{i,j-1}^{k+1}u_{i,j-1}^{k+1}}{2\Delta y}+\frac{u_{i,j+1}^{k}u_{i,j+1}^{k}-u_{i,j-1}^{k}u_{i,j-1}^{k}}{2\Delta y}\right) \quad (7)$$
$$=\frac{\Delta t}{2\,\mathrm{Re}}\left(\frac{u_{i+1,j}^{k+1}-2u_{i,j}^{k+1}+u_{i-1,j}^{k+1}}{\Delta x^2}+\frac{u_{i+1,j}^{k}-2u_{i,j}^{k}+u_{i-1,j}^{k}}{\Delta x^2}\right.$$
$$\left.+\frac{u_{i,j+1}^{k+1}-2u_{i,j}^{k+1}+u_{i,j-1}^{k+1}}{\Delta y^2}+\frac{u_{i,j+1}^{k}-2u_{i,j}^{k}+u_{i,j-1}^{k}}{\Delta y^2}\right)$$

This equation is equivalent to

$$(1+Sx+Sy)u_{i,j}^{k+1}-\frac{Sx}{2}\left(u_{i+1,j}^{k+1}+u_{i-1,j}^{k+1}\right)-\frac{Sy}{2}\left(u_{i,j+1}^{k+1}+u_{i,j-1}^{k+1}\right)$$
$$+\frac{Cx}{8}\left[\left(u_{i+1,j}^{k+1}\right)^2-\left(u_{i-1,j}^{k+1}\right)^2\right]+\frac{Cy}{8}\left[\left(u_{i,j+1}^{k+1}\right)^2-\left(u_{i,j-1}^{k+1}\right)^2\right] \quad (8)$$
$$=(1-Sx-Sy)u_{i,j}^{k}+\frac{Sx}{2}\left(u_{i+1,j}^{k}+u_{i-1,j}^{k}\right)+\frac{Sy}{2}\left(u_{i,j+1}^{k}+u_{i,j-1}^{k}\right)$$
$$-\frac{Cx}{8}\left[\left(u_{i+1,j}^{k}\right)^2-\left(u_{i-1,j}^{k}\right)^2\right]-\frac{Cy}{8}\left[\left(u_{i,j+1}^{k}\right)^2-\left(u_{i,j-1}^{k}\right)^2\right]$$

where $Sx=\frac{1}{\mathrm{Re}}\frac{\Delta t}{\Delta x^2}$, $Sy=\frac{1}{\mathrm{Re}}\frac{\Delta t}{\Delta y^2}$, $Cx=\frac{\Delta t}{\Delta x}$ and $Cy=\frac{\Delta t}{\Delta y}$. Using the classic Newton's iteration method, equation (8) can be written as the root finding problem for the function of g as

$$g_{i,j}^{p}=(1+Sx+Sy)u_{i,j}^{k+1,p}-\frac{Sx}{2}\left(u_{i+1,j}^{k+1,p}+u_{i-1,j}^{k+1,p}\right)-\frac{Sy}{2}\left(u_{i,j+1}^{k+1,p}+u_{i,j-1}^{k+1,p}\right)$$
$$+\frac{Cx}{8}\left[\left(u_{i+1,j}^{k+1,p}\right)^2-\left(u_{i-1,j}^{k+1,p}\right)^2\right]+\frac{Cy}{8}\left[\left(u_{i,j+1}^{k+1,p}\right)^2-\left(u_{i,j-1}^{k+1,p}\right)^2\right] \quad (9)$$
$$-\left\{(1-Sx-Sy)u_{i,j}^{k}+\frac{Sx}{2}\left(u_{i+1,j}^{k}+u_{i-1,j}^{k}\right)+\frac{Sy}{2}\left(u_{i,j+1}^{k}+u_{i,j-1}^{k}\right)\right.$$
$$\left.-\frac{Cx}{8}\left[\left(u_{i+1,j}^{k}\right)^2-\left(u_{i-1,j}^{k}\right)^2\right]-\frac{Cy}{8}\left[\left(u_{i,j+1}^{k}\right)^2-\left(u_{i,j-1}^{k}\right)^2\right]\right\}$$

The Newton's iteration [10] at each time step, is given by

$$\left(\frac{\partial g_{i,j}^{p}}{\partial u_{i,j}^{k+1,p}}\right)\left(w_{i,j}\right)=-g_{i,j}^{p} \quad (10)$$

where

$$w_{i,j}=u_{i,j}^{k+1,p+1}-u_{i,j}^{k+1,p},\ p=0,1,2,... \quad (11)$$

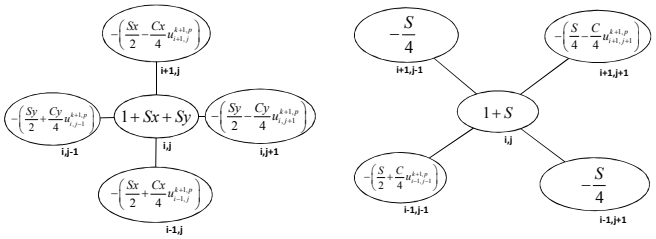
and $\left(\frac{\partial g_{i,j}^{p}}{\partial u_{i,j}^{k+1,p}}\right)$ = Jacobian matrix, evaluated using term by term discretization of (9). The standard point Crank-Nicolson Newton's iterative formula can be shown to be the following form:

$$(1+Sx+Sy)w_{i,j}-\left(\frac{Sx}{2}-\frac{Cx}{4}u_{i+1,j}^{k+1,p}\right)w_{i+1,j}-\left(\frac{Sx}{2}+\frac{Cx}{4}u_{i-1,j}^{k+1,p}\right)w_{i-1,j} \quad (12)$$
$$-\left(\frac{Sy}{2}-\frac{Cy}{4}u_{i,j+1}^{k+1,p}\right)w_{i,j+1}-\left(\frac{Sy}{2}+\frac{Cy}{4}u_{i,j-1}^{k+1,p}\right)w_{i,j-1}=-g_{i,j}^{p}$$

The computational molecule of CN is shown in Fig. 1 (a).

The Newton's Iterative Algorithm for standard point Crank-Nicolson (CN) method at each time step is:

1. Do until convergence
2. compute g function
3. Do until convergence
4. solve for w in system Aw=b
5. End Do
6. $u^{p+1} = w + u^{p}$
7. End Do



(a) (b)

Fig. 1 (a) Computational Molecule of (12); (b) Computational Molecule of (16)

C. Rotated Point Crank-Nicolson (RCN) Newton's Iterative Method

Using rotated point finite difference approximation for (6), the difference equation becomes

$$u_{i,j}^{k+1}+u_{i,j}^{k}+\frac{\Delta t}{2}\left(\frac{u_{i+1,j+1}^{k+1}u_{i+1,j+1}^{k+1}-u_{i-1,j-1}^{k+1}u_{i-1,j-1}^{k+1}}{4h}\right.$$
$$\left.+\frac{u_{i+1,j+1}^{k}u_{i+1,j+1}^{k}-u_{i-1,j-1}^{k}u_{i-1,j-1}^{k}}{4h}\right) \quad (13)$$
$$=\frac{\Delta t}{2\,\mathrm{Re}}\left(\frac{u_{i+1,j+1}^{k+1}+u_{i-1,j-1}^{k+1}+u_{i-1,j+1}^{k+1}+u_{i+1,j-1}^{k+1}-4u_{i,j}^{k+1}}{2h^2}\right.$$
$$\left.+\frac{u_{i+1,j+1}^{k}+u_{i-1,j-1}^{k}+u_{i-1,j+1}^{k}+u_{i+1,j-1}^{k}-4u_{i,j}^{k}}{2h^2}\right)$$

where h = Δx = Δy. On simplification, the following equation is obtained

$$(1+S)u_{i,j}^{k+1}-\frac{S}{4}\left(u_{i+1,j-1}^{k+1}+u_{i-1,j+1}^{k+1}+u_{i+1,j+1}^{k+1}+u_{i-1,j-1}^{k+1}\right)$$
$$+\frac{C}{8}\left(u_{i+1,j+1}^{k+1}u_{i+1,j+1}^{k+1}-u_{i-1,j-1}^{k+1}u_{i-1,j-1}^{k+1}\right) \quad (14)$$
$$=(1-S)u_{i,j}^{k}+\frac{S}{4}\left(u_{i+1,j-1}^{k}+u_{i-1,j+1}^{k}+u_{i+1,j+1}^{k}+u_{i-1,j-1}^{k}\right)$$
$$-\frac{C}{8}\left(u_{i+1,j+1}^{k}u_{i+1,j+1}^{k}-u_{i-1,j-1}^{k}u_{i-1,j-1}^{k}\right)$$

where $S=\frac{1}{\mathrm{Re}}\frac{\Delta t}{h^2}$ and $C=\frac{\Delta t}{h}$. The function g for the rotated point

approximation is:

$$g_{i,j}^{p}=(1+S)u_{i,j}^{k+1}-\frac{S}{4}\left(u_{i+1,j-1}^{k+1}+u_{i-1,j+1}^{k+1}+u_{i+1,j+1}^{k+1}+u_{i-1,j-1}^{k+1}\right)+\frac{C}{8}\left(u_{i+1,j+1}^{k+1}u_{i+1,j+1}^{k+1}-u_{i-1,j-1}^{k+1}u_{i-1,j-1}^{k+1}\right)-\left\{(1-S)u_{i,j}^{k}+\frac{S}{4}\left(u_{i+1,j-1}^{k}+u_{i-1,j+1}^{k}+u_{i+1,j+1}^{k}+u_{i-1,j-1}^{k}\right)-\frac{C}{8}\left(u_{i+1,j+1}^{k}u_{i+1,j+1}^{k}-u_{i-1,j-1}^{k}u_{i-1,j-1}^{k}\right)\right\} \tag{15}$$

The RCN Newton's iterative equation is as below

$$(1+S)w_{i,j}-\left(\frac{S}{4}-\frac{C}{4}u_{i+1,j+1}^{k+1,p}\right)w_{i+1,j+1}-\left(\frac{S}{4}+\frac{C}{4}u_{i-1,j-1}^{k+1,p}\right)w_{i-1,j-1}-\frac{S}{4}\left(w_{i+1,j-1}+w_{i-1,j+1}\right)=-g_{i,j}^{p} \tag{16}$$

The computational molecule of RCN is shown in Fig. 1 (b). Fig. 2 represents the mesh points on solution domain of RCN Newton's iterative method for the case n = 9. We can see that the evaluations at the points ● involve only the same type of points; similarly for the points of type O. Thus, the iterations can be generated involving one type of points only. For the numerical experiment in Section IV, we chose the points of type ● to be involved in the iterations. After a certain convergence criteria is achieved, the solutions at the points of O will be evaluated directly using (9) and (12).

The Newton's Iterative Algorithm of rotated point Crank-Nicolson (RCN) scheme at each time step is as follows:

1. *Do until convergence*
2. *compute g function at the points (●) only*
3. *Do until convergence*
4. *solve for w in system Aw=b at the points (●) only*
5. *End Do*
6. *compute $u^{p+1} = w + u^{p}$ at the points (●) only*
7. *End Do*
8. *compute g function using CN at the points (O)*
9. *compute w at points (O)*
10. *compute u^{p+1} at points (O)*

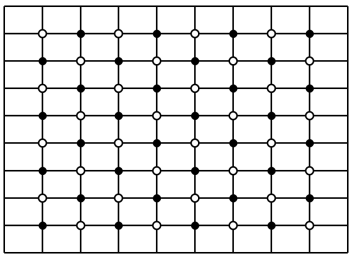


Fig. 2 Grid point on x-y plane for n=9

D.Explicit Group (EG) Newton's Iterative Method

To formulate the EG Newton's iterative scheme for the Burger's equation, we apply (12) to any group of four points on the solution domain in the inner iteration at each time step. This will result in a (4x4) system of equations as follows

$$\begin{bmatrix} a_1 & a_2 & 0 & a_3 \\ a_4 & a_1 & a_5 & 0 \\ 0 & a_6 & a_1 & a_7 \\ a_8 & 0 & a_9 & a_1 \end{bmatrix}\begin{bmatrix} w_{i,j} \\ w_{i+1,j} \\ w_{i+1,j+1} \\ w_{i,j+1} \end{bmatrix}=\begin{bmatrix} RHS_{i,j} \\ RHS_{i+1,j} \\ RHS_{i+1,j+1} \\ RHS_{i,j+1} \end{bmatrix} \tag{17}$$

where

$$\begin{aligned} &a_1=1+Sx+Sy,\\ &a_2=-\left(\frac{Sx}{2}-\frac{Cx}{4}u_{i+1,j}^{k+1,p}\right),\ a_3=-\left(\frac{Sy}{2}-\frac{Cy}{4}u_{i,j+1}^{k+1,p}\right),\\ &a_4=-\left(\frac{Sx}{2}+\frac{Cx}{4}u_{i,j}^{k+1,p}\right),\ a_5=-\left(\frac{Sy}{2}-\frac{Cy}{4}u_{i+1,j+1}^{k+1,p}\right),\\ &a_6=-\left(\frac{Sy}{2}+\frac{Cy}{4}u_{i+1,j}^{k+1,p}\right),\ a_7=-\left(\frac{Sx}{2}+\frac{Cx}{4}u_{i,j+1}^{k+1,p}\right),\\ &a_8=-\left(\frac{Sy}{2}+\frac{Cy}{4}u_{i,j}^{k+1,p}\right),\ a_9=-\left(\frac{Sx}{2}-\frac{Cx}{4}u_{i+1,j+1}^{k+1,p}\right), \end{aligned} \tag{18}$$

and

$$\begin{aligned} RHS_{i,j}&=\left(\frac{Sx}{2}+\frac{Cx}{4}u_{i-1,j}^{k+1,p}\right)w_{i-1,j}+\left(\frac{Sy}{2}+\frac{Cy}{4}u_{i,j-1}^{k+1,p}\right)w_{i,j-1}-g_{i,j}^{p}\\ RHS_{i+1,j}&=\left(\frac{Sx}{2}-\frac{Cx}{4}u_{i+2,j}^{k+1,p}\right)w_{i+2,j}+\left(\frac{Sy}{2}+\frac{Cy}{4}u_{i+1,j-1}^{k+1,p}\right)w_{i+1,j-1}-g_{i+1,j}^{p}\\ RHS_{i+1,j+1}&=\left(\frac{Sx}{2}-\frac{Cx}{4}u_{i+2,j+1}^{k+1,p}\right)w_{i+2,j+1}+\left(\frac{Sy}{2}-\frac{Cy}{4}u_{i+1,j+2}^{k+1,p}\right)w_{i+1,j+2}-g_{i+1,j+1}^{p}\\ RHS_{i,j+1}&=\left(\frac{Sx}{2}+\frac{Cx}{4}u_{i-1,j+1}^{k+1,p}\right)w_{i-1,j+1}+\left(\frac{Sy}{2}-\frac{Cy}{4}u_{i,j+2}^{k+1,p}\right)w_{i,j+2}-g_{i,j+1}^{p} \end{aligned} \tag{19}$$

The computational molecule of EG is shown in Fig. 3. The explicit form of (17) is obtained as

$$\begin{bmatrix} w_{i,j} \\ w_{i+1,j} \\ w_{i+1,j+1} \\ w_{i,j+1} \end{bmatrix}=\frac{1}{\text{const}}\begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ b_5 & b_6 & b_7 & b_8 \\ b_9 & b_{10} & b_{11} & b_{12} \\ b_{13} & b_{14} & b_{15} & b_{16} \end{bmatrix}\begin{bmatrix} RHS_{i,j} \\ RHS_{i+1,j} \\ RHS_{i+1,j+1} \\ RHS_{i,j+1} \end{bmatrix} \tag{20}$$

where

$$\begin{aligned} &b_1=a_1\left(a_1^2-a_5a_6-a_7a_9\right),b_2=-a_2\left(a_1^2-a_7a_9\right)-a_3a_6a_9,\\ &b_3=a_1\left(a_2a_5+a_3a_9\right),b_4=-a_3\left(a_1^2-a_5a_6\right)-a_2a_5a_7,\\ &b_5=-a_4\left(a_1^2-a_7a_9\right)-a_5a_7a_8,b_6=a_1\left(a_1^2-a_3a_8-a_7a_9\right),\\ &b_7=-a_5\left(a_1^2-a_3a_8\right)-a_3a_4a_9,b_8=a_1\left(a_3a_4+a_5a_7\right)\\ &b_9=a_1\left(a_4a_6+a_7a_8\right),b_{10}=-a_6\left(a_1^2-a_3a_8\right)-a_2a_7a_8,\\ &b_{11}=a_1\left(a_1^2-a_2a_4-a_3a_8\right),b_{12}=-a_7\left(a_1^2-a_2a_4\right)-a_3a_4a_6,\\ &b_{13}=-a_8\left(a_1^2-a_5a_6\right)-a_4a_6a_9,b_{14}=a_1\left(a_2a_8+a_6a_9\right),\\ &b_{15}=-a_9\left(a_1^2-a_2a_4\right)-a_2a_5a_8,b_{16}=a_1\left(a_1^2-a_2a_4-a_5a_6\right),\\ &\text{const}=a_1^2\left(a_1^2-a_2a_4-a_5a_6-a_3a_8-a_7a_9\right)\\ &\qquad+a_3a_5a_6a_8-a_2a_5a_7a_8-a_3a_4a_6a_9+a_2a_4a_7a_9 \end{aligned} \tag{21}$$

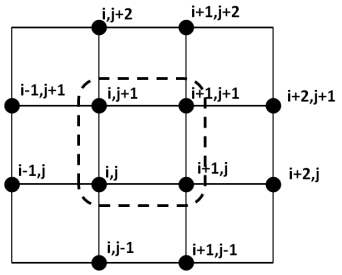


Fig. 3 Computational Molecule of (17)

E. Explicit Decoupled Group (EDG) Newton's Iterative Method

For EDG Newton's iterative scheme, we apply (17) to any group of four points on the solution domain in the inner iteration at each time step. Similarly, this will also result in a (4x4) system of equation:

$$\begin{bmatrix} a_1 & -a_2 & 0 & 0 \\ -a_3 & a_1 & 0 & 0 \\ 0 & 0 & a_1 & -a_4 \\ 0 & 0 & -a_4 & a_1 \end{bmatrix} \begin{bmatrix} w_{i,j} \\ w_{i+1,j+1} \\ w_{i+1,j} \\ w_{i,j+1} \end{bmatrix} = \begin{bmatrix} RHS_{i,j} \\ RHS_{i+1,j+1} \\ RHS_{i+1,j} \\ RHS_{i,j+1} \end{bmatrix} \tag{22}$$

where

$$a_1 = 1+S, a_2 = \frac{S}{4} - \frac{C}{4}u_{i+1,j+1}^{k+1,p}, a_3 = \frac{S}{4} + \frac{C}{4}u_{i,j}^{k+1,p}, a_4 = -\frac{S}{4} \tag{23}$$

and

$$\begin{aligned} RHS_{i,j} &= \left(\frac{S}{4} + \frac{C}{4}u_{i-1,j-1}^{k+1,p}\right)w_{i-1,j-1} + \frac{S}{4}\left(w_{i+1,j-1} + w_{i-1,j+1}\right) - g_{i,j}^{p} \\ RHS_{i+1,j+1} &= \left(\frac{S}{4} - \frac{C}{4}u_{i+2,j+2}^{k+1,p}\right)w_{i+2,j+2} + \frac{S}{4}\left(w_{i+2,j} + w_{i,j+2}\right) - g_{i+1,j+1}^{p} \\ RHS_{i+1,j} &= \left(\frac{S}{4} + \frac{C}{4}u_{i,j-1}^{k+1,p}\right)w_{i,j-1} + \left(\frac{S}{4} - \frac{C}{4}u_{i+2,j+1}^{k+1,p}\right)w_{i+2,j+1} + \frac{S}{4}w_{i+2,j-1} - g_{i+1,j}^{p} \\ RHS_{i,j+1} &= \left(\frac{S}{4} + \frac{C}{4}u_{i-1,j}^{k+1,p}\right)w_{i-1,j} + \left(\frac{S}{4} - \frac{C}{4}u_{i+1,j+2}^{k+1,p}\right)w_{i+1,j+2} + \frac{S}{4}w_{i-1,j+2} - g_{i,j+1}^{p} \end{aligned} \tag{24}$$

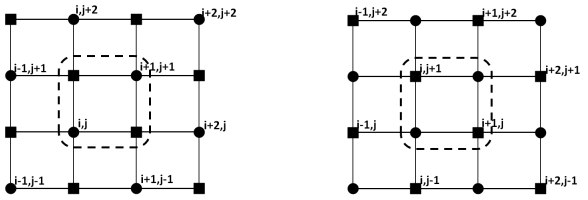
This system leads to two explicit decoupled systems of 2x2 equations

$$\begin{bmatrix} w_{i,j} \\ w_{i+1,j+1} \end{bmatrix} = \frac{1}{a_1^2 - a_2 a_3} \begin{bmatrix} a_1 & a_2 \\ a_3 & a_1 \end{bmatrix} \begin{bmatrix} RHS_{i,j} \\ RHS_{i+1,j+1} \end{bmatrix} \tag{25}$$

and

$$\begin{bmatrix} w_{i+1,j} \\ w_{i,j+1} \end{bmatrix} = \frac{1}{a_1^2 - a_4^2} \begin{bmatrix} a_1 & a_4 \\ a_4 & a_1 \end{bmatrix} \begin{bmatrix} RHS_{i+1,j} \\ RHS_{i,j+1} \end{bmatrix} \tag{26}$$

Fig. 4 shows the computational molecule of equations (25) and (26). The iterative evaluation of (25) involves points of type ● only, while (26) involves points of type ■ only. Therefore, the implementation of (25) and (26) can be carried out independently. We can save the execution time if the iteration over the solution domain is only carried out on either type of points (● or ■). Suppose we choose to iterate on points of type ●. The EDG Newton's iterative scheme corresponds to iterations on these points using the group formula (25) until convergence test is satisfied. After a convergence criterion is achieved, the solutions at the remaining points of type ■ are evaluated directly once using the formulas of CN (9) and (12).



(a) (b)

Fig. 4 (a) Computational Molecule of (25); (b) Computational Molecule of (26)

III. Computational Complexity Analysis

In this section, we analyze the computational complexity of the four Newton's iterative methods. The estimation is based on the arithmetic operations performed at every iteration. Assume that the solution domain is discretized with mesh size n, then the number of internal mesh points is given by m^2 where m = n – 1. There are two types of internal mesh points, which are the iterative and direct points. Iterative points are the points which are involved in the iteration process, while direct points are the points that are computed directly once after the iteration convergence criteria is achieved. Table 1 shows the number of arithmetic operations required for every iteration of each method (excluding the convergence test and direct solutions). To further study the complexity analysis of the explicit group methods, please refer to [2] – [3].

TABLE I
Computational Complexity for Four Newton's Iterative Methods

Methods	Additional	Multiplication
Standard Point CN	9 m^2	8 m^2
Rotated Point CN	3 m^2	3 m^2
EG	21.75 m^2	18 m^2
EDG	4 m^2	3.25 m^2

IV. Numerical Experiments and Discussion of Results

In order to compare the performances of these Newton's iterative methods described in Section II, we carried out the numerical experiments on a PC with Core 2 Duo 2.8 GHz, 2 GB of RAM with Window XP SP3 operating system using Cygwin C. The convergence criteria used was the maximum absolute error test with the error tolerance equal to $\varepsilon = 10^{-10}$.

We solve the following two-dimensional Burger's equation

$$\frac{\partial u}{\partial t} + u\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right) = \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{27}$$

where $D:[0,2]\times[0,2]$.The initial and boundary conditions are defined so that they satisfy the exact solution [11]:

$$u(x,y,t)=\frac{1}{1+\exp(\mathrm{Re}(x+y-t)/2)},(x,y)\in D,t\geq 0 \quad (28)$$

Tables II and III in APPENDIX show the numerical results of all four methods described in Section II when 1/RE = 1.0. Amongst the point methods, the rotated point scheme (RCN) is faster than the traditional CN point scheme as the number of grid points increases due to its lower computational complexities. It can be observed that the accuracies of EG and EDG are as good as the CN and RCN respectively, but they require less execution times to achieve the results. The computing timing of EG is only 56 – 84 % of CN. The computing timing of EDG is only 62 – 91 % of RCN. The results of these four methods when 1/RE = 0.1 are shown in Tables 4 and 5 in APPENDIX. The EDG scheme is the fastest method. Figs. 5 and 6 display the execution times of these four methods when 1/RE=1.0 and 0.1 respectively.

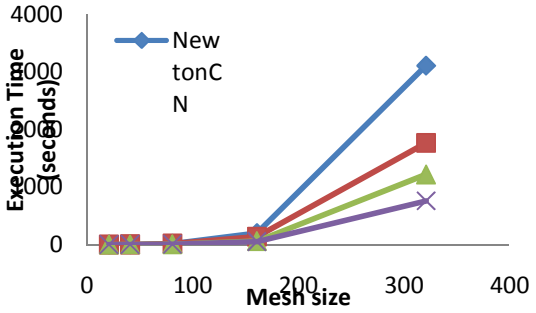


Fig. 5 Execution Times of standard point CN, rotated point CN, EG and EDG Newton's iterative methods when t = 2, 1/Re = 1, dt=1/1000

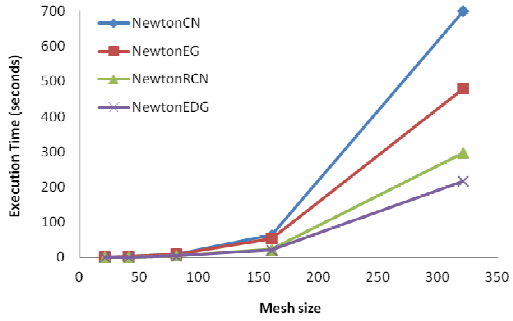


Fig. 6 Execution Times of standard point CN, rotated point CN, EG and EDG Newton's iterative methods when t = 2, 1/Re = 0.10, dt=1/1000

The experimental results of these methods with various number of 1/RE (0.1, 0.01 and 0.005) are shown in Table 6 on APPENDIX. When 1/RE is decreased, the shock of the solution is sharper and steeper. Our Newton's iterative methods are still accurate and non-oscillatory since the maximum error increases only at a small part localized to the middle part of the domain along the line x+y=2.

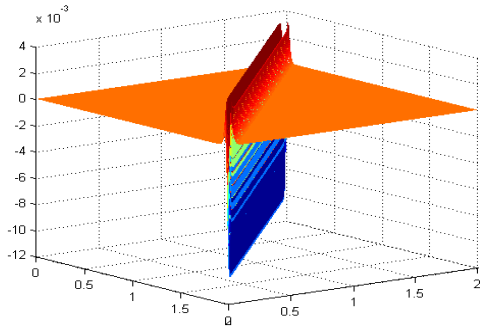


Fig. 7 Surface plot of the difference between the numerical results of EG and exact solutions at t = 2, 1/Re = 0.005, Δt = 0.005, Δx = 2/401

V. Conclusion

We have presented two explicit group Newton's iterative methods derived from the standard point and rotated point Crank-Nicolson approximations to solve the time-dependent two dimensional Burger's equation. It is observed that the computational cost for EDG is the least compared with the other three methods. However, when 1/RE is getting smaller, a finer mesh is preferred, as the maximum error increases when 1/RE decreases. The computation cost increases as the grid sizes increase. The implementation of parallel technology and domain decomposition methods will be investigated and reported soon.

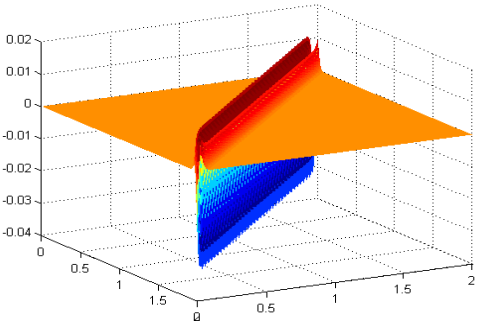


Fig. 8 Surface plot of the difference between the numerical results of EDG and exact solutions at t = 2, 1/Re = 0.005, Δt = 0.005, Δx = 2/401

Appendix

TABLE II
Numerical results of CN and RCN at 1/RE=1.0

Space step	CN		RCN	
	Maximum Error	Time (seconds)	Maximum Error	Time (seconds)
2/21	0.0000061424	0.406	0.0000246707	0.217
2/41	0.0000016167	2.310	0.0000064729	0.986
2/81	0.0000004153	17.777	0.0000016577	6.198
2/161	0.0000001163	198.231	0.0000004231	60.783
2/321	0.0000000958	3104.663	0.0000001329	1216.607

TABLE III
NUMERICAL RESULTS OF EG AND EDG AT 1/RE=1.0

Space step	EG		EDG	
	Maximum Error	Time (seconds)	Maximum Error	Time (seconds)
2/21	0.0000061424	0.344	0.0000246707	0.219
2/41	0.0000016167	2.201	0.0000064729	0.900
2/81	0.0000004146	13.906	0.0000016573	5.686
2/161	0.0000001087	133.466	0.0000004216	49.335
2/321	0.0000000549	1760.950	0.0000001247	754.048

TABLE IV
NUMERICAL RESULTS OF CN AND RCN AT 1/RE=0.1

Space step	CN		RCN	
	Maximum Error	Time (seconds)	Maximum Error	Time (seconds)
2/21	0.005942000	0.407	0.023384424	0.187
2/41	0.001548691	1.735	0.006122375	0.873
2/81	0.000400472	9.246	0.001600764	4.294
2/161	0.000101567	62.936	0.000404833	22.984
2/321	0.000025769	699.855	0.000102175	297.523

TABLE V
NUMERICAL RESULTS OF EG AND EDG AT 1/RE=0.1

Space step	EG		EDG	
	Maximum Error	Time (seconds)	Maximum Error	Time (seconds)
2/21	0.005942001	0.421	0.023384424	0.188
2/41	0.001548691	1.841	0.006122375	0.842
2/81	0.000400472	9.269	0.001600764	3.642
2/161	0.000101567	52.714	0.000404833	20.858
2/321	0.000025768	479.061	0.000102175	216.505

TABLE VI
MAXIMUM ERROR OF FOUR METHODS WHEN $\Delta T = 0.005$, $\Delta X = 2/401$, $T = 2.0$

1/RE	Maximum Error			
	CN	EG	RCN	EDG
0.1	0.000023766	0.000023766	0.000072648	0.000072648
0.01	0.002657666	0.002657666	0.008096495	0.008096405
0.005	0.010284811	0.010284811	0.031722012	0.031722012

ACKNOWLEDGMENT

The authors acknowledge the School of Mathematical Sciences KPI Incentive Grant Scheme (1002/PMATHS/ARSP13000) for the support of this research.

REFERENCES

[1] K. B. Tan, N. H. M. Ali and C-H. Lai, "Explicit Group Methods in the Solution of the 2-D Convertion-Diffusion Equations," *Proceedings of the World Congress on Engineering 2010*, pp. 1799 – 1804.

[2] W. S. Yousif, and D. J. Evans, "Explcit Group Over-Relaxation Methods for Solving Elliptic Partial Differential Equations," *Mathematics and Computers in Simulation*, vol. 28, pp. 453 – 466, 1986.

[3] 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.

[4] W. S. Yousif, and D. J. Evans, "Explicit De-coupled Group Iterative Methods and Their Parallel Implementations," *Parallel Algorithms and Applications*, vol. 7, pp. 53 – 71, 1995.

[5] M. Othman, and A. R. Abdullah, "An Efficient Four Points Modified Explicit Group Poisson Solver," *International Journal of Computer Mathematics*, vol. 76(2), pp. 203 – 217, 2000.

[6] N. H. M. Ali, The Design and Analysis of Some Parallel Algorithms for the Iterative Solution of Partial Differential Equations. PhD Thesis, Fakulti Teknologi dan Sains Maklumat, Universiti Kebangsaan Malaysia, 1998, pp. 160 – 216.

[7] N. H. M. Ali, and K. F. Ng, "Modified Explicit Decoupled Group Method in The Solution of 2-D Elliptic PDES," *Proceedings of the 12th WSEAS International Conference on Applied Mathematics*, pp. 162 – 167, 2007.

[8] J. M. Burger, "A mathematical model illustrating the theory of turbulence," *Advances in Applied Mechanics*, vol. 1, pp. 171 – 199, 1948.

[9] J. D. Cole, "On a quasilinear parabolic equations occurring in aerodynamics," *Quarterly of Applied Mathematics*, vol. 9, pp. 225 – 236, 1951.

[10] D. J. Evans, "Iterative methods for solving non-linear two point boundary value problems," *International Journal of Computer Mathematics*, Vol. 72, pp. 395 – 401, 1999.

[11] W. Liao, "A fourth-order finite-difference method for solving the system of two-dimensional Burgers' equations," *International Journal for Numerical Methods in Fluids*, vol. 64, pp. 565 – 590, 2010.