World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:7, No:12, 2013

New High Order Group Iterative Schemes in the Solution of Poisson Equation

Sam Teek Ling, Norhashidah Hj. Mohd. Ali

***Abstract*—**We investigate the formulation and implementation of new explicit group iterative methods in solving the two-dimensional Poisson equation with Dirichlet boundary conditions. The methods are derived from a fourth order compact nine point finite difference discretization. The methods are compared with the existing second order standard five point formula to show the dramatic improvement in computed accuracy. Numerical experiments are presented to illustrate the effectiveness of the proposed methods.

***Keywords*—**Explicit group iterative method, finite difference, fourth order compact, Poisson equation.

I. Introduction

We study the finite difference discretization schemes for approximating the solution of the two-dimensional Poisson equation given by

$$u_{xx}+u_{yy}=f\left(x,y\right) \tag{1}$$

defined in a unit square domain Ω with Dirichlet boundary conditions. Assume (1) as our model problem and then discretize Ω with uniform mesh size $h=\frac{1}{n}$ in both x and y coordinate directions, where $x_i=ih$, $y_j=jh$ $\left(i,j=0,1,2,\cdots,n\right)$. In the sequel, we use the index pair $\left(i,j\right)$ to represent the mesh point $\left(x_i,y_j\right)$. There are various ways to discretize (1). The most familiar scheme is based on the fourth order compact (nine point) formula,

$$\begin{aligned}&u_{i+1,j+1}+u_{i+1,j-1}+u_{i-1,j+1}+u_{i-1,j-1}\\&\quad+4\left(u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}\right)-20u_{i,j}\\&\quad=\frac{h^2}{2}\left(8f_{i,j}+f_{i+1,j}+f_{i-1,j}+f_{i,j+1}+f_{i,j-1}\right).\end{aligned} \tag{2}$$

Equation (2) is generally called *Mehrstellenverfahren* and has been known for many years [5], [6]. This popular finite difference approximation was developed by Collatz [5] and implemented by Houstis and Papatheodorou in a Fortran Program FFT9 [3].

The second order group iterative method that Evans proposed in 1985 is called the Explicit Group (EG) method [1]. The Modified Explicit Group (MEG) iterative method for solving large linear systems was initiated by Othman and Abdullah [8]. Further investigations on group iterative methods have been extensively conducted by Evans and Yousif [2], [11], Martins, Yousif and Evans [7], Othman and Abdullah [9]. The fourth order EG and MEG methods may be constructed by using (2). The MEG method is found to be more superior in execution timings than the EG method.

The aim of this paper is to study the performance of higher order group iterative methods derived from the nine point formula (2). The paper is organised in four sections. In Section II, we show the formulation of the algorithms for group iterative methods. Numerical experiments on the specific Poisson equation have been carried out and the results are shown in Section III. Concluding remarks are given in Section IV.

II. Explicit Group Iterative Methods

Applying (2) to groups of four points will result in the following (4×4) system

$$\begin{bmatrix}20&-4&-4&-1\\-4&20&-1&-4\\-4&-1&20&-4\\-1&-4&-4&20\end{bmatrix}\begin{bmatrix}u_{i,j}\\u_{i+1,j}\\u_{i,j+1}\\u_{i+1,j+1}\end{bmatrix}=\begin{bmatrix}b_{i,j}\\b_{i+1,j}\\b_{i,j+1}\\b_{i+1,j+1}\end{bmatrix} \tag{3}$$

where

$$\begin{aligned}b_{i,j}&=4u_{i-1,j}+4u_{i,j-1}+u_{i-1,j-1}+u_{i+1,j-1}+u_{i-1,j+1}\\&\quad-\frac{h^2}{2}\left(8f_{i,j}+f_{i+1,j}+f_{i-1,j}+f_{i,j+1}+f_{i,j-1}\right)\\b_{i+1,j}&=4u_{i+2,j}+4u_{i+1,j-1}+u_{i+2,j+1}+u_{i,j-1}+u_{i+2,j-1}\\&\quad-\frac{h^2}{2}\left(8f_{i+1,j}+f_{i+2,j}+f_{i,j}+f_{i+1,j+1}+f_{i+1,j-1}\right)\\b_{i,j+1}&=4u_{i-1,j+1}+4u_{i,j+2}+u_{i+1,j+2}+u_{i-1,j}+u_{i-1,j+2}\\&\quad-\frac{h^2}{2}\left(8f_{i,j+1}+f_{i+1,j+1}+f_{i-1,j+1}+f_{i,j+2}+f_{i,j}\right)\\b_{i+1,j+1}&=4u_{i+2,j+1}+4u_{i+1,j+2}+u_{i+2,j+2}+u_{i+2,j}+u_{i,j+2}\\&\quad-\frac{h^2}{2}\left(8f_{i+1,j+1}+f_{i+2,j+1}+f_{i,j+1}+f_{i+1,j+2}+f_{i+1,j}\right)\end{aligned}$$

The system (3) can be inverted to produce a four-point EG equation

T. L. Sam is with Institut Matematik Kejuruteraan, Universiti Malaysia Perlis, Kampus Pauh Putra, 02600 Pauh, Perlis, Malaysia (corresponding author; e-mail: tlsam@unimap.edu.my).

Norhashidah Hj. Mohd. Ali is with School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Minden, Pulau Pinang, Malaysia (e-mail: shidah@cs.usm.my).

$$\begin{bmatrix} u_{i,j} \\ u_{i+1,j} \\ u_{i,j+1} \\ u_{i+1,j+1} \end{bmatrix} = \frac{1}{2079}\begin{bmatrix} 116 & 28 & 28 & 17 \\ 28 & 116 & 17 & 28 \\ 28 & 17 & 116 & 28 \\ 17 & 28 & 28 & 116 \end{bmatrix}\begin{bmatrix} b_{i,j} \\ b_{i+1,j} \\ b_{i,j+1} \\ b_{i+1,j+1} \end{bmatrix}. \tag{4}$$

The algorithm of the EG method which represents the fourth order compact nine point formula (2) in solving Poisson problem (1) is illustrated in Algorithm I.

ALGORITHM I
ALGORITHM OF THE EG METHOD

```
$S_{01} = u_{i-1,j-1} - \frac{h^2}{2}\left(8f_{i,j} + f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}\right)$
$S_{02} = u_{i,j-1} + u_{i-1,j}$
$S_{03} = u_{i+1,j-1} + u_{i-1,j+1}$
$S_{04} = u_{i+2,j} + u_{i,j+2}$
$S_{05} = u_{i+2,j+1} + u_{i+1,j+2}$
$S_{06} = u_{i+2,j+2} - \frac{h^2}{2}\left(8f_{i+1,j+1} + f_{i+2,j+1} + f_{i,j+1} + f_{i+1,j+2} + f_{i+1,j}\right)$
$S_{07} = u_{i,j-1} + u_{i+2,j+1}$
$S_{08} = u_{i+1,j-1} + u_{i+2,j}$
$S_{09} = u_{i+2,j-1} - \frac{h^2}{2}\left(8f_{i+1,j} + f_{i+2,j} + f_{i,j} + f_{i+1,j+1} + f_{i+1,j-1}\right)$
$S_{10} = u_{i-1,j} + u_{i+1,j+2}$
$S_{11} = u_{i-1,j+1} + u_{i,j+2}$
$S_{12} = u_{i-1,j+2} - \frac{h^2}{2}\left(8f_{i,j+1} + f_{i+1,j+1} + f_{i-1,j+1} + f_{i,j+2} + f_{i,j}\right)$
$T_{01} = S_{01} + S_{06}$
$T_{02} = S_{09} + S_{12}$

$u_{i,j} = \frac{1}{2079}\left(116S_{01} + 492S_{02} + 228S_{03} + 28T_{02} + 129S_{04} + 96S_{05} + 17S_{06}\right)$
$u_{i+1,j} = \frac{1}{2079}\left(28T_{01} + 228S_{07} + 492S_{08} + 116S_{09} + 129S_{10} + 96S_{11} + 17S_{12}\right)$
$u_{i,j+1} = \frac{1}{2079}\left(28T_{01} + 129S_{07} + 96S_{08} + 17S_{09} + 228S_{10} + 492S_{11} + 116S_{12}\right)$
$u_{i+1,j+1} = \frac{1}{2079}\left(17S_{01} + 96S_{02} + 129S_{03} + 28T_{02} + 228S_{04} + 492S_{05} + 116S_{06}\right)$
```

We modify the EG method by considering points at grid size $2h = \frac{2}{n}$. The construction of the MEG method is similar to the original EG method. We first discretize (1) using the same formula (2) with grid spacing $2h$ which leads to the following formula,

$$\begin{aligned} & u_{i+2,j+2} + u_{i+2,j-2} + u_{i-2,j+2} + u_{i-2,j-2} \\ & \quad + 4\left(u_{i+2,j} + u_{i-2,j} + u_{i,j+2} + u_{i,j-2}\right) - 20u_{i,j} \\ & \quad = 2h^2\left(8f_{i,j} + f_{i+2,j} + f_{i-2,j} + f_{i,j+2} + f_{i,j-2}\right). \end{aligned} \tag{5}$$

Now we use (5) to groups of four points and generate the following (4×4) system

$$\begin{bmatrix} 20 & -4 & -4 & -1 \\ -4 & 20 & -1 & -4 \\ -4 & -1 & 20 & -4 \\ -1 & -4 & -4 & 20 \end{bmatrix}\begin{bmatrix} u_{i,j} \\ u_{i+2,j} \\ u_{i,j+2} \\ u_{i+2,j+2} \end{bmatrix} = \begin{bmatrix} c_{i,j} \\ c_{i+2,j} \\ c_{i,j+2} \\ c_{i+2,j+2} \end{bmatrix} \tag{6}$$

with

$$\begin{aligned} c_{i,j} &= 4u_{i-2,j} + 4u_{i,j-2} + u_{i-2,j-2} + u_{i+2,j-2} + u_{i-2,j+2} \\ &\quad - 2h^2\left(8f_{i,j} + f_{i+2,j} + f_{i-2,j} + f_{i,j+2} + f_{i,j-2}\right) \\ c_{i+2,j} &= 4u_{i+4,j} + 4u_{i+2,j-2} + u_{i+4,j+2} + u_{i,j-2} + u_{i+4,j-2} \\ &\quad - 2h^2\left(8f_{i+2,j} + f_{i+4,j} + f_{i,j} + f_{i+2,j+2} + f_{i+2,j-2}\right) \\ c_{i,j+2} &= 4u_{i-2,j+2} + 4u_{i,j+4} + u_{i+2,j+4} + u_{i-2,j} + u_{i-2,j+4} \\ &\quad - 2h^2\left(8f_{i,j+2} + f_{i+2,j+2} + f_{i-2,j+2} + f_{i,j+4} + f_{i,j}\right) \\ c_{i+2,j+2} &= 4u_{i+4,j+2} + 4u_{i+2,j+4} + u_{i+4,j+4} + u_{i+4,j} + u_{i,j+4} \\ &\quad - 2h^2\left(8f_{i+2,j+2} + f_{i+4,j+2} + f_{i,j+2} + f_{i+2,j+4} + f_{i+2,j}\right) \end{aligned}$$

The system (6) can be inverted to give a four-point MEG equation

$$\begin{bmatrix} u_{i,j} \\ u_{i+2,j} \\ u_{i,j+2} \\ u_{i+2,j+2} \end{bmatrix} = \frac{1}{2079}\begin{bmatrix} 116 & 28 & 28 & 17 \\ 28 & 116 & 17 & 28 \\ 28 & 17 & 116 & 28 \\ 17 & 28 & 28 & 116 \end{bmatrix}\begin{bmatrix} c_{i,j} \\ c_{i+2,j} \\ c_{i,j+2} \\ c_{i+2,j+2} \end{bmatrix}. \tag{7}$$

The algorithm of the MEG method which computes the approximate solution of Poisson problem (1) is mentioned in Algorithms II. Figs. 1-3 show the discretization points of a unit square domain with $n = 14$ and the various types of points involved. It is obvious that MEG method involved only one quarter of the interior points in its iteration process. After convergence is achieved, the remaining points are solved directly once.

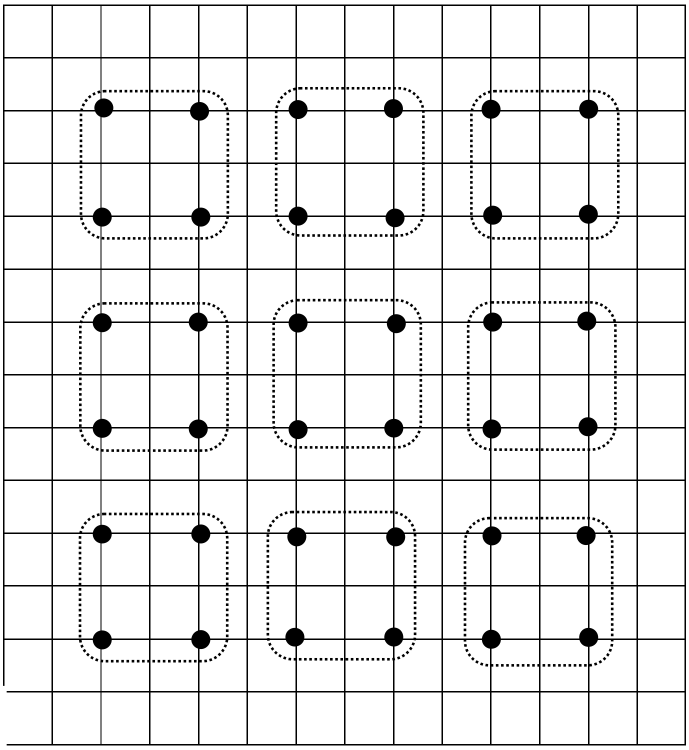

Fig. 1 MEG Step 1: Iterate on the points of type ● using MEG formula

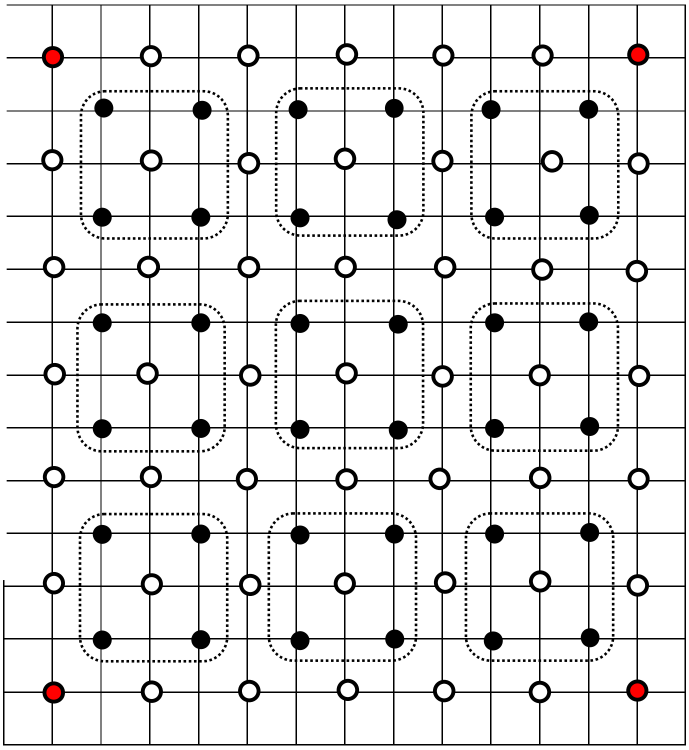

Fig. 2 MEG Step 2: Evaluation at 🔴 and ○ points directly once using Rotated Fourth Order formula (with spacing h)

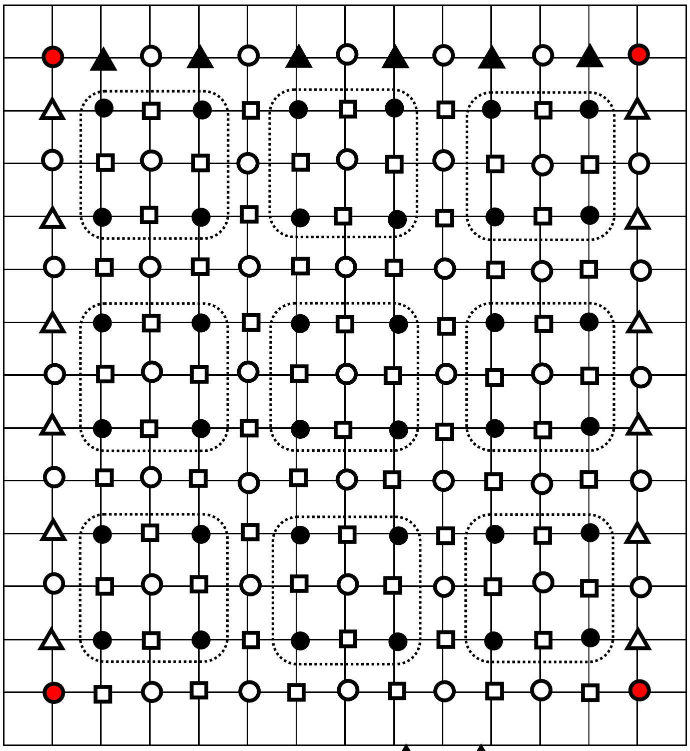

Fig. 3 MEG Step 3: Evaluation at □ , △ and ▲ points directly once using the Compact Fourth Order formula (with spacing h)

ALGORITHM II
ALGORITHM OF THE MEG METHOD

Iterate:

●

MEG:

$$S_{01} = u_{i-2,j-2} - 2h^2\left(8f_{i,j} + f_{i+2,j} + f_{i-2,j} + f_{i,j+2} + f_{i,j-2}\right)$$
$$S_{02} = u_{i,j-2} + u_{i-2,j}$$
$$S_{03} = u_{i+2,j-2} + u_{i-2,j+2}$$
$$S_{04} = u_{i+4,j} + u_{i,j+4}$$
$$S_{05} = u_{i+4,j+2} + u_{i+2,j+4}$$
$$S_{06} = u_{i+4,j+4} - 2h^2\left(8f_{i+2,j+2} + f_{i+4,j+2} + f_{i,j+2} + f_{i+2,j+4} + f_{i+2,j}\right)$$
$$S_{07} = u_{i,j-2} + u_{i+4,j+2}$$
$$S_{08} = u_{i+2,j-2} + u_{i+4,j}$$
$$S_{09} = u_{i+4,j-2} - 2h^2\left(8f_{i+2,j} + f_{i+4,j} + f_{i,j} + f_{i+2,j+2} + f_{i+2,j-2}\right)$$
$$S_{10} = u_{i-2,j} + u_{i+2,j+4}$$
$$S_{11} = u_{i-2,j+2} + u_{i,j+4}$$
$$S_{12} = u_{i-2,j+4} - 2h^2\left(8f_{i,j+2} + f_{i+2,j+2} + f_{i-2,j+2} + f_{i,j+4} + f_{i,j}\right)$$
$$T_{01} = S_{01} + S_{06}$$
$$T_{02} = S_{09} + S_{12}$$

$$u_{i,j} = \frac{1}{2079}\left(116S_{01} + 492S_{02} + 228S_{03} + 28T_{02} + 129S_{04} + 96S_{05} + 17S_{06}\right)$$
$$u_{i+2,j} = \frac{1}{2079}\left(28T_{01} + 228S_{07} + 492S_{08} + 116S_{09} + 129S_{10} + 96S_{11} + 17S_{12}\right)$$
$$u_{i,j+2} = \frac{1}{2079}\left(28T_{01} + 129S_{07} + 96S_{08} + 17S_{09} + 228S_{10} + 492S_{11} + 116S_{12}\right)$$
$$u_{i+2,j+2} = \frac{1}{2079}\left(17S_{01} + 96S_{02} + 129S_{03} + 28T_{02} + 228S_{04} + 492S_{05} + 116S_{06}\right)$$

After convergence is achieved:

Rotated Fourth Order: (with spacing *h*)

○

Boundary Down:

$$u_{i-2,j}+4u_{i,j-1}+\frac{18}{4}u_{i+1,j+1}+\frac{13}{4}u_{i+1,j-1}+\frac{17}{4}u_{i-1,j+1}+3u_{i-1,j-1}$$
$$+\frac{1}{4}\left(u_{i+3,j+1}+u_{i+3,j-1}+u_{i+1,j+3}+u_{i-1,j+3}\right)-21u_{i,j}$$
$$=h^2\left(8f_{i,j}+f_{i+1,j+1}+f_{i+1,j-1}+f_{i-1,j+1}+f_{i-1,j-1}-f_{i,j-1}+\frac{1}{2}\left(f_{i+2,j}+f_{i,j+2}\right)\right)$$

Boundary Up:

$$u_{i-2,j}+u_{i,j-2}+4u_{i,j+1}+\frac{13}{4}u_{i+1,j+1}+3u_{i-1,j+1}+\frac{17}{4}u_{i+1,j-1}+4u_{i-1,j-1}$$
$$+\frac{1}{4}\left(u_{i+3,j-1}+u_{i+3,j+1}\right)-21u_{i,j}$$
$$=h^2\left(8f_{i,j}+f_{i+1,j+1}+f_{i+1,j-1}+f_{i-1,j+1}+f_{i-1,j-1}-f_{i,j+1}+\frac{1}{2}f_{i+2,j}\right)$$

Boundary Left:

$$u_{i,j-2}+4u_{i-1,j}+\frac{18}{4}u_{i+1,j+1}+\frac{17}{4}u_{i+1,j-1}+\frac{13}{4}u_{i-1,j+1}+3u_{i-1,j-1}$$
$$+\frac{1}{4}\left(u_{i+3,j+1}+u_{i+3,j-1}+u_{i+1,j+3}+u_{i-1,j+3}\right)-21u_{i,j}$$
$$=h^2\left(8f_{i,j}+f_{i+1,j+1}+f_{i+1,j-1}+f_{i-1,j+1}+f_{i-1,j-1}-f_{i-1,j}+\frac{1}{2}\left(f_{i+2,j}+f_{i,j+2}\right)\right)$$

Boundary Right:

$$u_{i-2,j}+u_{i,j-2}+4u_{i+1,j}+\frac{13}{4}u_{i+1,j+1}+3u_{i+1,j-1}+\frac{17}{4}u_{i-1,j+1}+4u_{i-1,j-1}$$
$$+\frac{1}{4}\left(u_{i+1,j+3}+u_{i-1,j+3}\right)-21u_{i,j}$$
$$=h^2\left(8f_{i,j}+f_{i+1,j+1}+f_{i+1,j-1}+f_{i-1,j+1}+f_{i-1,j-1}-f_{i+1,j}+\frac{1}{2}f_{i,j+2}\right)$$

●

Corner Left-Down:

$$4\left(u_{i-1,j}+u_{i,j-1}\right)+\frac{18}{4}u_{i+1,j+1}+2u_{i-1,j-1}+\frac{13}{4}\left(u_{i+1,j-1}+u_{i-1,j+1}\right)$$
$$+\frac{1}{4}\left(u_{i+3,j+1}+u_{i+3,j-1}+u_{i+1,j+3}+u_{i-1,j+3}\right)-22u_{i,j}$$
$$=h^2\begin{pmatrix}8f_{i,j}+f_{i+1,j+1}+f_{i+1,j-1}+f_{i-1,j+1}+f_{i-1,j-1}-f_{i-1,j}-f_{i,j-1}\\+\frac{1}{2}\left(f_{i+2,j}+f_{i,j+2}\right)\end{pmatrix}$$

Corner Right -Down:

$$u_{i-2,j}+4\left(u_{i+1,j}+u_{i,j-1}\right)+\frac{13}{4}u_{i+1,j+1}+2u_{i+1,j-1}+\frac{17}{4}u_{i-1,j+1}+3u_{i-1,j-1}$$
$$+\frac{1}{4}\left(u_{i+1,j+3}+u_{i-1,j+3}\right)-22u_{i,j}$$
$$=h^2\left(8f_{i,j}+f_{i+1,j+1}+f_{i+1,j-1}+f_{i-1,j+1}+f_{i-1,j-1}-f_{i+1,j}-f_{i,j-1}+\frac{1}{2}f_{i,j+2}\right)$$

Corner Left-Up:

$$u_{i,j-2}+4\left(u_{i-1,j}+u_{i,j+1}\right)+\frac{13}{4}u_{i+1,j+1}+2u_{i-1,j+1}+\frac{17}{4}u_{i+1,j-1}+3u_{i-1,j-1}$$
$$+\frac{1}{4}\left(u_{i+3,j-1}+u_{i+3,j+1}\right)-22u_{i,j}$$
$$=h^2\left(8f_{i,j}+f_{i+1,j+1}+f_{i+1,j-1}+f_{i-1,j+1}+f_{i-1,j-1}-f_{i-1,j}-f_{i,j+1}+\frac{1}{2}f_{i+2,j}\right)$$

Corner Right-Up:

$$u_{i-2,j}+u_{i,j-2}+4\left(u_{i+1,j}+u_{i,j+1}\right)+4u_{i-1,j-1}+3\left(u_{i+1,j-1}+u_{i-1,j+1}\right)+2u_{i+1,j+1}-22u_{i,j}$$
$$=h^2\left(8f_{i,j}+f_{i+1,j+1}+f_{i+1,j-1}+f_{i-1,j+1}+f_{i-1,j-1}-f_{i+1,j}-f_{i,j+1}\right)$$

○

Inner:

$$u_{i-2,j}+u_{i,j-2}+\frac{18}{4}u_{i+1,j+1}+\frac{17}{4}\left(u_{i+1,j-1}+u_{i-1,j+1}\right)+4u_{i-1,j-1}$$
$$+\frac{1}{4}\left(u_{i+3,j+1}+u_{i+3,j-1}+u_{i+1,j+3}+u_{i-1,j+3}\right)-20u_{i,j}$$
$$=h^2\left(8f_{i,j}+f_{i+1,j+1}+f_{i+1,j-1}+f_{i-1,j+1}+f_{i-1,j-1}+\frac{1}{2}\left(f_{i+2,j}+f_{i,j+2}\right)\right)$$

Compact Fourth Order: (with spacing *h*)

□

Boundary Down & Inner:

$$u_{i+1,j-1}+u_{i-1,j-1}+\frac{1}{4}\left(u_{i+2,j+1}+u_{i+1,j+2}+u_{i-1,j+2}+u_{i-2,j+1}\right)+4u_{i,j-1}$$
$$+\frac{9}{2}u_{i,j+1}+\frac{17}{4}\left(u_{i+1,j}+u_{i-1,j}\right)-20u_{i,j}$$
$$=\frac{h^2}{2}\left(8f_{i,j}+f_{i+1,j}+f_{i-1,j}+f_{i,j+1}+f_{i,j-1}+\frac{1}{2}\left(f_{i+1,j+1}+f_{i-1,j+1}\right)\right)$$

△

Boundary Left:

$$u_{i+1,j-1}+u_{i-1,j+1}+u_{i-1,j-1}+\frac{1}{4}\left(u_{i+2,j+1}+u_{i+1,j+2}\right)+4\left(u_{i-1,j}+u_{i,j-1}\right)$$
$$+\frac{17}{4}\left(u_{i+1,j}+u_{i,j+1}\right)-20u_{i,j}$$
$$=\frac{h^2}{2}\left(8f_{i,j}+f_{i+1,j}+f_{i-1,j}+f_{i,j+1}+f_{i,j-1}+\frac{1}{2}\left(f_{i+1,j+1}\right)\right)$$

Boundary Right:

$$u_{i+1,j+1}+u_{i+1,j-1}+u_{i-1,j-1}+\frac{1}{4}\left(u_{i-1,j+2}+u_{i-2,j+1}\right)+4\left(u_{i+1,j}+u_{i,j-1}\right)$$
$$+\frac{17}{4}\left(u_{i-1,j}+u_{i,j+1}\right)-20u_{i,j}$$
$$=\frac{h^2}{2}\left(8f_{i,j}+f_{i+1,j}+f_{i-1,j}+f_{i,j+1}+f_{i,j-1}+\frac{1}{2}\left(f_{i-1,j+1}\right)\right)$$

▲

Boundary Up:

$$u_{i+1,j+1}+u_{i+1,j-1}+u_{i-1,j+1}+u_{i-1,j-1}+4\left(u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}\right)-20u_{i,j}$$
$$=\frac{h^2}{2}\left(8f_{i,j}+f_{i+1,j}+f_{i-1,j}+f_{i,j+1}+f_{i,j-1}\right)$$

III. Numerical Experiments and Results

In this section, numerical results for the scheme presented in the previous sections are given. The Successive Over-Relaxation (SOR) was the accelerator used in the iterative methods. The theoretical optimum relaxation factor ω_{opt} for the SOR iterative scheme can be computed from

$$\omega_{\text{opt}}=\frac{2}{1+\sqrt{1-\rho^2\left(B\right)}} \tag{8}$$

where ρ (B) is the spectral radius of the Jacobian iterative matrix [10]. The rate of convergence for the iterative methods is relied on the spectral radius. The smaller ρ, the faster convergence [4].

All numerical experiments were carried out on a computer with processor Intel(R) Core(TM) 2 Quad CPU Q9400 @ 2.66GHz and 3.00GB of main memory (RAM). Our code is written in C++ programming language. We used the following two test problems on a unit square to test the performance of the higher order explicit group methods for different values of n.

Problem I: The problem can be written as

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=-9\pi^2\left(sin\left(3\pi x\right)+cos\left(3\pi y\right)\right),\quad (x,y)\in\Omega=[0,1]\times[0,1], \tag{9}$$

where the boundary conditions are

$$u\left(x,0\right)=sin\left(3\pi x\right)+1,\ \ u\left(x,1\right)=sin\left(3\pi x\right)-1,$$
$$u\left(0,y\right)=cos\left(3\pi y\right),\ \ u\left(1,y\right)=cos\left(3\pi y\right).$$

The analytic solution of (9) is

$$u\left(x,y\right)=sin\left(3\pi x\right)+cos\left(3\pi y\right).$$

Problem II: We choose the following equation:

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=\frac{\left(2x^2+2y^2\right)\left(e^x+e^y\right)}{\left(1+xy\right)^3}+\frac{e^x+e^y}{1+xy}-\frac{2xe^y+2ye^x}{\left(1+xy\right)^2},\quad (x,y)\in\Omega=[0,1]\times[0,1], \tag{10}$$

which has the Dirichlet boundary conditions:

$$u\left(x,0\right)=e^x+1,\ \ u\left(x,1\right)=\frac{e^x+e}{1+x},$$
$$u\left(0,y\right)=1+e^y,\ \ u\left(1,y\right)=\frac{e+e^y}{1+y}.$$

The exact solution of this problem is

$$u\left(x,y\right)=\frac{e^x+e^y}{1+xy}.$$

To ensure a fair comparison is attained, all the iterations were initiated from the same initial values $u_{ij}^{(0)}$ and the computations were terminated when the same convergence test was satisfied with tolerance ε = 10^{-12}. The programs terminated when the Euclidean norm (2-norm) of the residual vector is reduced by 10^{-12}. The maximum absolute error reported is the maximum absolute error between the computed solution at convergence and the exact solution over all grid points. The results show the number of iteration (k), value of ω, maximum absolute error (e), execution time (t) in seconds and the order of accuracy (m) [12].

TABLE I
PERFORMANCE COMPARISON ON PROBLEM I

Methods	***n***	***ω***	***k***	***t***	***e***	***m***
Second Order (Standard Five Point)	8	1.45	39	0.00	2.53516e-001	-
	16	1.68	79	0.00	6.47435e-002	1.96927
	32	1.82	158	0.03	1.60236e-002	2.01454
	64	1.91	325	0.19	3.99585e-003	2.00362
	128	1.95	651	1.56	9.99040e-004	1.99989
	256	1.98	1535	15.91	2.49720e-004	2.00023
Compact Fourth Order (Nine Point)	8	1.43	38	0.00	1.75785e-002	-
	16	1.66	75	0.00	1.12369e-003	3.96750
	32	1.81	151	0.01	6.95049e-005	4.01499
	64	1.90	298	0.22	4.33286e-006	4.00372
	128	1.95	610	1.79	2.70819e-007	3.99992
	256	1.97	1368	17.05	1.74676e-008	3.95458
Fourth Order EG (Nine Point)	8	1.33	28	0.00	1.75785e-002	-
	16	1.57	54	0.00	1.12369e-003	3.96750
	32	1.76	109	0.01	6.95049e-005	4.01499
	64	1.87	214	0.16	4.33284e-006	4.00373
	128	1.93	451	1.39	2.70827e-007	3.99987
	256	1.97	973	12.67	1.69233e-008	4.00029
Fourth Order MEG (Nine Point)	8	1.12	16	0.00	3.26688e-001	-
	16	1.33	28	0.00	1.75785e-002	4.21603
	32	1.57	54	0.00	1.12369e-003	3.96750
	64	1.76	109	0.02	7.27410e-005	3.94933
	128	1.87	214	0.16	4.59156e-006	3.98571
	256	1.93	451	1.37	2.87502e-007	3.99734

We first compare the Second Order Standard Five Point formula (*FPF*) with the Fourth Order Compact Nine Point formula (*NPF*). We also investigate the computational cost (CPU time) required for computing an approximate solution with a given accuracy. Tables I and II contain the results.

TABLE II
PERFORMANCE COMPARISON ON PROBLEM II

Methods	***n***	***ω***	***k***	***t***	***e***	***m***
Second Order (Standard Five Point)	8	1.45	39	0.00	4.28755e-004	-
	16	1.68	80	0.00	1.11086e-004	1.94848
	32	1.82	164	0.03	2.80023e-005	1.98806
	64	1.91	320	0.19	7.01780e-006	1.99645
	128	1.95	684	1.62	1.75520e-006	1.99938
	256	1.98	1536	15.94	4.38912e-007	1.99963
Compact Fourth Order (Nine Point)	8	1.44	38	0.00	9.29910e-006	-
	16	1.66	78	0.00	5.84471e-007	3.99189
	32	1.82	157	0.03	3.65783e-008	3.99807
	64	1.90	310	0.22	2.17991e-009	4.06865
	128	1.95	616	1.83	1.30212e-010	4.06533
	256	1.98	1403	17.85	9.33387e-012	3.80224
Fourth Order EG (Nine Point)	8	1.33	28	0.00	9.29909e-006	-
	16	1.57	55	0.00	5.84479e-007	3.99187
	32	1.76	109	0.02	3.65789e-008	3.99807
	64	1.87	215	0.17	2.28563e-009	4.00035
	128	1.93	480	1.51	1.18647e-010	4.26784
	256	1.97	978	12.84	9.29701e-012	3.67377
Fourth Order MEG (Nine Point)	8	1.12	16	0.00	1.61035e-004	-
	16	1.33	28	0.00	1.18586e-005	3.76337
	32	1.57	55	0.00	8.57168e-007	3.79021
	64	1.76	109	0.02	6.48277e-008	3.72490
	128	1.87	215	0.16	4.66644e-009	3.79622
	256	1.93	480	1.45	3.17236e-010	3.87869

NPF achieves significantly better accuracy than *FPF* for the same values of *n*. The CPU times are comparable, which *NPF* is remarkably better than *FPF*. If we seek a required accuracy, *NPF* converges with less iterations and far fewer execution time by doing calculations. For example from Table I, *FPF* achieves a maximum absolute error around 3.99585×10^{-3} with the cost is 0.19 CPU seconds. The computational cost of *NPF* is 0.00 CPU seconds for a similar value of error. In all cases, *NPF* costs less CPU time than *FPF*, and achieves higher accuracy.

According to Tables I and II, we note that the errors of all fourth order iterative schemes (*NPF*, EG and MEG) decay by a factor of 16 and the errors of *FPF* decrease by a factor of 4 when *n* is doubled. The maximum errors of *NPF* and EG do not vary very much. EG is going faster than *NPF* to obtain the same accuracy.

From the results obtained, it can be observed that the MEG method performs better than all iterative methods in terms of number of iterations and execution times in all of the cases tested. MEG converges the fastest among the iterative methods tested which is due to its lower computational complexity in the iterative process. The iterative process for the MEG scheme is carried on one quarter of the total nodal points, while the remaining points are solved directly once after convergence is achieved.

The values of ω can be used to calculate ρ by applying (8). We rearrange (8) to obtain ρ in the form

$$\rho = \sqrt{1-\left(\frac{2}{\omega}-1\right)^{2}}. \qquad (11)$$

The numerical results of ρ are shown in Table III.

TABLE III
PERFORMANCE COMPARISON ON SPECTRAL RADIUS, ρ

	Problem I			Problem II		
Methods	***n***	***ω***	***ρ***	***n***	***ω***	***ρ***
Compact Fourth Order (Nine Point)	8	1.43	0.91712	8	1.44	0.92128
	16	1.66	0.97880	16	1.66	0.97880
	32	1.81	0.99448	32	1.82	0.99510
	64	1.90	0.99861	64	1.90	0.99861
	128	1.95	0.99967	128	1.95	0.99967
	256	1.97	0.99988	256	1.98	0.99995
Fourth Order EG (Nine Point)	8	1.33	0.86384	8	1.33	0.86384
	16	1.57	0.96176	16	1.57	0.96176
	32	1.76	0.99066	32	1.76	0.99066
	64	1.87	0.99758	64	1.87	0.99758
	128	1.93	0.99934	128	1.93	0.99934
	256	1.97	0.99988	256	1.97	0.99988
Fourth Order MEG (Nine Point)	8	1.12	0.61859	8	1.12	0.61859
	16	1.33	0.86384	16	1.33	0.86384
	32	1.57	0.96176	32	1.57	0.96176
	64	1.76	0.99066	64	1.76	0.99066
	128	1.87	0.99758	128	1.87	0.99758
	256	1.93	0.99934	256	1.93	0.99934

Clearly it has been seen that for the same values of *n*, the spectral radius of MEG method is the smallest among three iterative methods. The spectral radius of EG method is slightly fewer than *NPF* in both test problems, thus verifying our findings in CPU times. Smaller ρ will result in faster convergence rate [4].

IV. CONCLUSIONS

Through numerical experiments, we have shown that the higher order of point and group iterative methods can be applied successfully with SOR in solving simpler type of partial differential equations. Among all iterative methods, MEG requires the least CPU times to converge, while *NPF* appears to be the most expensive in terms of execution times. *NPF* and EG give much higher accuracy than *FPF* and MEG. All of our tests show that EG is much more efficient than *NPF* in terms of CPU times and number of iterations.

ACKNOWLEDGMENT

Financial support provided by the School of Maths KPI Incentive Grant and Fundamental Research Grant Scheme (203/PMATHS/6711321) for the completion of this article are gratefully acknowledged.

REFERENCES

[1] D. J. Evans, “Group Explicit Iterative Methods for Solving Large Linear Systems,” *Int. J. Computer Math.*, 17, pp. 81-108, 1985.

[2] D. J. Evans, and W. S. Yousif, “Explicit Group Over-Relaxation Iterative Methods for Solving Elliptic Partial Differential Equations in 3-Space Dimensions,” *Int. J. Comp. Math.*, 17, pp. 323-340, 1986.

[3] E. N. Houstis, and T. S. Papatheodorou, “Algorithm 543, FFT9: Fast Solutions of Helmholtz-Type Partial Differential Equations,” *ACM Trans. Math. Software 5*, 3, pp. 490-493, Sept. 1979.

[4] J. H. Ferziger, and M. Peric, *Computational Methods for Fluid Dynamics*, 2nd ed. Berlin: Springer-Verlag, 1999.

[5] L. Collatz, *The Numerical Treatment of Differential Equation*. Berlin: Springer-Verlag, 1960.

[6] L. V. Kantorovich, and V. I. Krylov, *Approximate Methods of Higher Analysis*. New York: Wiley, 1964.

[7] M. M. Martins, W. S. Yousif, and D. J. Evans, “Explicit Group AOR Method for Solving Elliptic Partial Differential Equations,” *Neural, Parallel and Science Computation*, vol. 10, no. 4, pp. 411-422, 2002.

[8] M. Othman, and A. R. Abdullah, “An Efficient Four Points Modified Explicit Group Poisson Solver,” *Int. J. Comp. Math.*, 76, pp. 203-217, 2000.

[9] M. Othman, and A. R. Abdullah, “A Parallel Four Points Modified Explicit Group Poisson Solver,” *J. Paral. Emer. Distri. Sys.*, 19, pp. 1-9, 2004.

[10] W. S. Yousif, *Iterative Methods for Sparse Linear Systems*, 2nd ed. USA: Society for Industrial and Applied Mathematics, 2003.

[11] W. S. Yousif, and D. J. Evans, “Explicit Group Over-Relaxation Methods for Solving Elliptic Partial Differential Equations,” *Mathematics and Computers in Simulation*, 28, pp. 453-466, 1986.

[12] Y. Wang, and J. Zhang, “Sixth Order Compact Scheme Combined with Multigrid Method and Extrapolation Technique for 2D Poisson Equation”, *Journal of Computation Physics*, 228, pp. 137-146, 2009.