Applied Mathematical Sciences, Vol. 4, 2010, no. 24, 1165 - 1181

Preconditioned Modified Explicit Decoupled Group Method in the Solution of Elliptic PDEs

Abdulkafi Mohammed Saeed

School of Mathematical Sciences, Universiti Sains Malaysia
11800 USM, Pulau Pinang
abdelkafe@yahoo.com

Norhashidah Hj. Mohd Ali

School of Mathematical Sciences, Universiti Sains Malaysia
11800 USM, Pulau Pinang
shidah@cs.usm.my

Abstract

Many physical phenomena in static field problems particularly in the electromagnetic field and the incompressible potential flow field are described by elliptic partial differential equations (pdes). Improved techniques using explicit group methods derived from the standard and skewed (rotated) finite difference operators have been developed over the last few years in solving the linear systems that arise from the discretization of these elliptic pdes (Ali et al., 2004; Evans and Yousif, 1990; Othman and Abdullah, 2000; Yousif and Evans, 1995). The convergence rates of these iterative methods depend on the spectral properties of the coefficient matrices resulted from these group discretization formulas. The formulation of suitable preconditioners which can improve the convergence rates of these iterative schemes are crucial to the development of these new group methods. This paper is concerned with the application of suitable preconditioning technique to a recently developed scheme, the Modified Explicit Decoupled Group (MEDG) iterative method due to Ali and Ng (2008), for solving the two dimensional elliptic pdes. Numerical experiments are conducted on each developed non-preconditioned and preconditioned schemes for comparison purposes. The results show the improvements in the convergence rate and the efficiency of the newly formulated preconditioned iterative scheme.

Keywords: Preconditioning method, Modified Explicit Decoupled Group (MEDG) method, Successive Over-Relaxation (SOR) method

1 Introduction

Solving partial differential equations (pdes) are usually at the heart of most scientific and engineering applications. The Explicit Decoupled Group (EDG) scheme was developed by Abdullah [1] as a more efficient Poisson solver on rotated grids by using small fixed size group strategy which was shown to be more economical computationally than the Explicit Group (EG) scheme due to Yousif and Evans ([4], [12]). Othman and Abdullah [8] subsequently modified the formulation of the EG method by altering the ordering of grid points taken in the iterative process to come up with the modified four-point EG where this method (MEG) was shown to be more superior in timings than both the original methods. In a recent paper, another explicit group method was proposed, namely the Modified Explicit Decoupled Group (MEDG) method ([2], [3]) as an addition to this family of four-point explicit group methods in solving Poisson equation. This method has been shown to be the fastest method among the four methods due to its lowest computational complexity. Since the reliability and robustness of iterative methods may now be improved by the use of preconditioning techniques, hence further efforts are being taken to combine the MEDG method with appropriate preconditioners as a way to further improve the performance of the method. A good choice of preconditioner can have a crucial impact on the efficiency and robustness of the resulting preconditioned iterative solver. Finding a good preconditioner inevitably combines a lot of intuition with rigorous definitions. Several methods have been developed on the preconditioned iterative methods for the last 15 years, but this quest is still going on ([6], [7], [11]). Saeed and Ali [9] introduced the preconditioner that improves the convergence rate of the Explicit Decoupled Group (EDG) and the numerical experiments yield very encouraging results. This paper is concerned with the application of suitable preconditioning techniques to the Modified Explicit Decoupled Group (MEDG) iterative method due to Ali and Ng [3] for solving the elliptic partial differential equation. Preconditioning strategies which improve the rate of convergence of these iterative methods are investigated. For a linear system which is obtained from the four point MEDG finite difference approximation, the $I + \overline{S}$ -type preconditioning matrix is employed in conjunction with the original system, where $\overline{S}$ is obtained by taking the first upper diagonal groups of iteration matrix of the original system. The paper is organised in six sections: Section 2 gives an overview of the Modified Explicit Decoupled Group (MEDG) method. In Section 3, we briefly describe the preconditioner $(I + \overline{S})$. We formulate this preconditioner in block formulation to the MEDG iterative method in Section 4. The Successive Over-Relaxation (SOR) was the accelarator used in the iterative method. The numerical results are presented in Section 5 in order to show the efficiency of the preconditioned MEDG SOR method. Finally, the concluding remarks are given in Section 6.

2 Modified Explicit Decoupled Group SOR (MEDG SOR)

Consider the following two dimensional Poisson equation

$$u_{xx} + u_{yy} = f(x, y), \quad x, y \in \Omega \tag{1}$$

with a Dirichlet boundary condition on a unit square solution domain $[0 \leq x,\ y \leq 1]$. Let Ω be discretized uniformly in both x and y directions with a mesh size $h = 1/n$ where n is a positive even integer. The solutions at the $(n-1)^2$ internal mesh points (x, y) can be approximated by various finite difference schemes. By using the centered difference equation, we will obtain the h-spaced standard five-point difference formula as follows:

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = 2h^2 f_{i,j}\ . \tag{2}$$

By rotating the x-y axis clockwise 45^o and applying the centered difference formula, we will then achieve the following $\sqrt{2}h - spaced$ rotated five-point difference formula

$$u_{i+1,j+1} + u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} - 4u_{i,j} = 2h^2 f_{i,j}\ . \tag{3}$$

The Modified Explicit Decoupled Group method is modification of the EDG method described by considering the points at grid size of $2h = 2/n$. The application of centered difference equation on these $2h$ spaced points results in the following standard five-point formula (with spacing $2h$)

$$u_{i+2,j} + u_{i-2,j} + u_{i,j+2} + u_{i,j-2} - 4u_{i,j} = 4h^2 f_{i,j}\ . \tag{4}$$

When the $x - y$ axis is rotated clockwise 45^o and the centered difference equation is applied on these points, it will result in the following rotated five-point difference formula (with spacing $2h$)

$$u_{i+2,j+2} + u_{i-2,j-2} + u_{i-2,j+2} + u_{i+2,j-2} - 4u_{i,j} = 8h^2 f_{i,j}\ . \tag{5}$$

Now by applying Eq. (5) to groups of four points as shown in Figure 1 and we produce the following (4×4) system of equations

$$\begin{pmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & 0 & 0 \\ 0 & 0 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} u_{i,j} \\ u_{i+2,j-2} \\ u_{i+2,j} \\ u_{i,j+2} \end{pmatrix} = \begin{pmatrix} u_{i-2,j-2} + u_{i+2,j-2} + u_{i-2,j+2} - 8h^2 f_{i,j} \\ u_{i,j+4} + u_{i+4,j} + u_{i+4,j+4} - 8h^2 f_{i+2,j+2} \\ u_{i,j-2} + u_{i+4,j-2} + u_{i+4,j+2} - 8h^2 f_{i+2,j} \\ u_{i-2,j+4} + u_{i-2,j} + u_{i+2,j+4} - 8h^2 f_{i,j+2} \end{pmatrix} \tag{6}$$

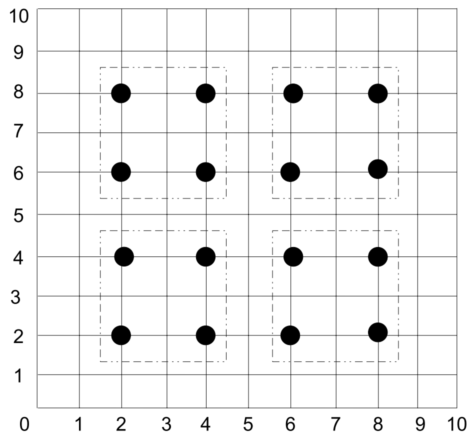

Figure 1: Groups of four points with 2h spacing

which can be inverted and rewritten in explicit forms of a decoupled system of (2×2) equations as :

$$\begin{pmatrix} u_{i,j} \\ u_{i+2,j+2} \end{pmatrix} = \frac{1}{15} \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} u_{i-2,j-2} + u_{i+2,j-2} + u_{i-2,j+2} - 8h^2 f_{i,j} \\ u_{i,j+4} + u_{i+4,j} + u_{i-4,j+4} - 8h^2 f_{i+2,j+2} \end{pmatrix} \tag{7}$$

and

$$\begin{pmatrix} u_{i+2,j} \\ u_{i,j+2} \end{pmatrix} = \frac{1}{15} \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} u_{i,j-2} + u_{i+4,j-2} + u_{i-4,j+2} - 8h^2 f_{i+2,j} \\ u_{i-2,j+4} + u_{i+2,j} + u_{i+2,j+4} - 8h^2 f_{i,j+2} \end{pmatrix} \tag{8}$$

Similar to the original EDG method, the evaluation of Eq. (7) and Eq. (8) can be performed independently. Figure 2 shows the discretization points of a unit square domain with n=14 and the various types of points involved. It is obvious that the evaluation of Eq. (7) involves only points of type $\bullet$ and Eq. (8) only points of type $\triangle$. The points of type $\bullet$ solved iteratively using Eq. (7) until convergence after which the points of type $\triangle$ is computed directly once using the standard 2h spaced five-point formula of Eq. (4). The remaining in-between points of type $\square$ are also computed directly once using the rotated five-point difference formula of Eq. (3), and followed by points of type $\diamond$ using the standard five-point difference formula of Eq. (2).

This will lead to the formation of a system of equations in the form $\mathrm{A}\tilde{u} = \tilde{b}$ where:

$$A = \begin{pmatrix} R_0 & R_1 & & & \\ R_2 & R_0 & R_1 & & \\ & R_2 & R_0 & \ddots & \\ & & \ddots & \ddots & R_1 \\ & & & R_2 & R_0 \end{pmatrix}_{\frac{(N-2)^2}{2} \times \frac{(N-2)^2}{2}}, R_0 = \begin{pmatrix} R_{00} & R_{01} & & \\ R_{02} & R_{00} & \ddots & \\ & \ddots & \ddots & R_{01} \\ & & R_{02} & R_{00} \end{pmatrix}_{(N-2)\times(N-2)},$$

$$R_{00} = \begin{pmatrix} 1 & -\frac{1}{4} \\ -\frac{1}{4} & 1 \end{pmatrix}, R_{01} = \begin{pmatrix} 0 & 0 \\ -\frac{1}{4} & 0 \end{pmatrix}, R_{02} = R_{01}^T,$$

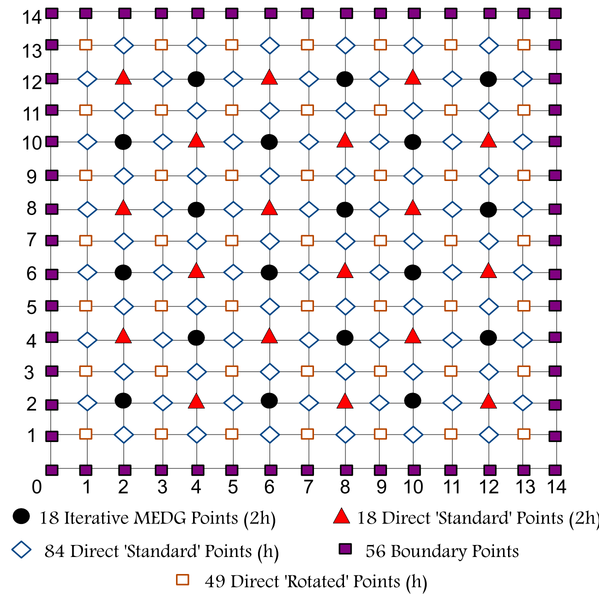

Figure 2: Types of discretized points in MEDG method for n=14

$$R_1 = \begin{pmatrix} R_{01} & R_{01} & & \\ & R_{01} & \ddots & \\ & & \ddots & R_{01} \\ & & & R_{01} \end{pmatrix}_{(N-2)\times(N-2)}, R_2 = \begin{pmatrix} R_{02} & & & \\ R_{02} & R_{02} & & \\ & \ddots & \ddots & \\ & & R_{02} & R_{02} \end{pmatrix}_{(N-2)\times(N-2)},$$

$$\tilde{u} = \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_3 \\ \vdots \\ \tilde{u}_{N-8} \\ \tilde{u}_{N-4} \end{pmatrix}_{\frac{(N-2)}{2}}, \tilde{u}_i = \begin{pmatrix} \tilde{u}_{2,i} \\ \tilde{u}_{6,i} \\ \vdots \\ \tilde{u}_{N-4,i} \end{pmatrix}_{(N-2)} \quad \text{for } i = 2(2)N-4,$$

$$\tilde{u}_{i,j} = \begin{pmatrix} u_{i,j} \\ u_{i+2,j+2} \end{pmatrix} \text{ for } i,j = 2(2)N-4$$

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

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

For $i = 2$,

$$\tilde{v}_{2,i} = \begin{pmatrix} v_{2,i} \\ v_{4,i+2} \end{pmatrix} = \begin{pmatrix} -2h^2 f_{2,i} + \frac{1}{4}u_{0,i-2} + \frac{1}{4}u_{4,i-2} + \frac{1}{4}u_{0,i+2} \\ -2h^2 f_{4,i+2} \end{pmatrix}$$
$$\tilde{v}_{k,i} = \begin{pmatrix} v_{k,i} \\ v_{k+2,i+2} \end{pmatrix} = \begin{pmatrix} -2h^2 f_{k,i} + \frac{1}{4}u_{k-2,i-2} + \frac{1}{4}u_{k+2,i-2} \\ -2h^2 f_{k+2,i+2} \end{pmatrix} \text{for } k = 6(2)N-8$$
$$\tilde{v}_{N-4,i} = \begin{pmatrix} v_{N-4,i} \\ v_{N-2,i+2} \end{pmatrix} = \begin{pmatrix} -2h^2 f_{N-4,i} + \frac{1}{4}u_{N-6,i-2} + \frac{1}{4}u_{N-2,i-2} \\ -2h^2 f_{N-2,i+2} + \frac{1}{4}u_{N,i} + \frac{1}{4}u_{N,i+4} \end{pmatrix}$$

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

$$\tilde{v}_{2,i} = \begin{pmatrix} v_{2,i} \\ v_{4,i+2} \end{pmatrix} = \begin{pmatrix} -2h^2 f_{2,i} + \frac{1}{4}u_{0,i-2} + \frac{1}{4}u_{0,i+2} \\ -2h^2 f_{4,i+2} \end{pmatrix}$$
$$\tilde{v}_{k,i} = \begin{pmatrix} v_{k,i} \\ v_{k+2,i+2} \end{pmatrix} = \begin{pmatrix} -2h^2 f_{k,i} \\ -2h^2 f_{k+2,i+2} \end{pmatrix} \quad \text{for} \quad k = 6(2)N-8$$
$$\tilde{v}_{N-4,i} = \begin{pmatrix} v_{N-4,i} \\ v_{N-2,i+2} \end{pmatrix} = \begin{pmatrix} -2h^2 f_{N-2,i} \\ -2h^2 f_{N-2,i+2} + \frac{1}{4}u_{N,i} + \frac{1}{4}u_{N,i+4} \end{pmatrix}$$

For $i = N-4$,

$$\tilde{v}_{2,i} = \begin{pmatrix} v_{2,i} \\ v_{4,i+2} \end{pmatrix} = \begin{pmatrix} -2h^2 f_{2,i} + \frac{1}{4}u_{0,i-2} + \frac{1}{4}u_{0,i+2} \\ -2h^2 f_{4,i+2} + \frac{1}{4}u_{2,i+4} + \frac{1}{4}u_{6,i+4} \end{pmatrix}$$
$$\tilde{v}_{k,i} = \begin{pmatrix} v_{k,i} \\ v_{k+2,i+2} \end{pmatrix} = \begin{pmatrix} -2h^2 f_{k,i} \\ -2h^2 f_{k+2,i+2} + \frac{1}{4}u_{k,i+4} + \frac{1}{4}u_{k+4,i+4} \end{pmatrix} \quad \text{for} \quad k = 6(2)N-8$$
$$\tilde{v}_{N-4,i} = \begin{pmatrix} v_{N-4,i} \\ v_{N-2,i+2} \end{pmatrix} = \begin{pmatrix} -2h^2 f_{N-4,i} \\ -2h^2 f_{N-2,i-2} + \frac{1}{4}u_{N-4,i+4} + \frac{1}{4}u_{N,i} + \frac{1}{4}u_{N,i+4} \end{pmatrix}$$

The line with $\underline{u}_{i,j}$ in (6) can be written as

$$R_{00}\underline{u}_{i,j} + R_{01}\underline{u}_{i,j-4} + R_{01}\underline{u}_{i-4,j} + R_{00}\underline{u}_{i-4,j-4} + R_{01}\underline{u}_{i+4,j} + R_{11}\underline{u}_{i,j+4} + R_{11}\underline{u}_{i+4,j+4} = \underline{v}_{i,j}$$

By rewriting in the explicit form, we will get:

$$\underline{u}_{i,j} = -R_{00}^{-1}R_{01}\underline{u}_{i-4,j-4} - R_{00}^{-1}R_{01}\underline{u}_{i,j-4} - R_{00}^{-1}\underline{u}_{i-4,j} - R_{00}^{-1}R_{01}\underline{u}_{i+4,j}$$

$$-R_{00}^{-1}R_{11}\underline{u}_{i,j+4} - R_{00}^{-1}R_{11}\underline{u}_{i+4,j+4} + R_{00}^{-1}\underline{v}_{i,j}$$

Since

$$R_{00}^{-1} = \frac{4}{15}\begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}, -R_{00}^{-1}R_{01} = \frac{1}{15}\begin{pmatrix} 1 & 0 \\ 4 & 0 \end{pmatrix}, -R_{00}^{-1}R_{02} = \frac{1}{15}\begin{pmatrix} 0 & 4 \\ 0 & 1 \end{pmatrix},$$

The MEDG formula is hence written as the following:

$$u_{i,j} = \frac{1}{15}(4F_1 + F_2),\ u_{i+2,j+2} = \frac{1}{15}(F_1 + 4F_2)$$

where:

$$F_1 = u_{i-2,j-2} + u_{i+2,j-2} + u_{i-2,j+2} - 8h^2 f_{i,j},$$
$$F_2 = u_{i,j+4} + u_{i+4,j} + u_{i+4,j+4} - 8h^2 f_{i+2,j+2}$$

As can be seen, the matrix A is a block tridiagonal matrix. Therefore the SOR method applied on this system will converge [10]. In order to obtain the formula of MEDG SOR method, we first need to derive the formulas of MEDG Jacobi and MEDG Gauss Seidel method respectively.

The iterative scheme for MEDG Jacobi method is given by

$$u_{i,j} = \frac{1}{15}(4F_1 + F_2),\ u_{i+2,j+2} = \frac{1}{15}(F_1 + 4F_2)$$

where:

$$F_1 = u^{(k)}_{i-2,j-2} + u^{(k)}_{i+2,j-2} + u^{(k)}_{i-2,j+2} - 8h^2 f_{i,j},$$
$$F_2 = u^{(k)}_{i,j+4} + u^{(k)}_{i+4,j} + u^{(k)}_{i+4,j+4} - 8h^2 f_{i+2,j+2}$$

The iterative scheme for MEDG Gauss Seidel method is given by

$$u_{i,j} = \frac{1}{15}(4F_1 + F_2),\ u_{i+2,j+2} = \frac{1}{15}(F_1 + 4F_2)$$

where:

$$\begin{aligned} F_1 &= u^{(k+1)}_{i-2,j-2} + u^{(k+1)}_{i+2,j-2} + u^{(k+1)}_{i-2,j+2} - 8h^2 f_{i,j}, \\ F_2 &= u^{(k)}_{i,j+4} + u^{(k)}_{i+4,j} + u^{(k)}_{i+4,j+4} - 8h^2 f_{i+2,j+2} \end{aligned} \tag{9}$$

Hence, the iterative scheme for MEDG SOR method is given by

$$u^{(k+1)}_{i,j} = \frac{1}{15}w(4F_1 + F_2) + (1-w)u^{(k)}_{i,j},$$
$$u^{(k+1)}_{i+2,j+2} = \frac{1}{15}w(F_1 + 4F_2) + (1-w)u^{(k)}_{i+2,j+2}$$

where F_1, F_2 are shown in Eq. (9) .

3 Preconditioners

Consider the linear system

$$A\tilde{u} = \tilde{b} \tag{10}$$

which is obtained by solving the Poisson equation using specific finite difference schemes. When a preconditioner is applied to the linear system Eq. (31), a new system will be obtained such that

$$PA\tilde{u} = P\tilde{b} \tag{11}$$

The preconditioner P of Gunawardena et al.[5] eliminates the elements of the first upper codiagonal of A where $P = I + \overline{S}$, I is the identity matrix which has the same dimension as A while $\overline{S}$ is the elements of the first upper diagonal of A,

$$\overline{S} = \begin{pmatrix} 0 & -a_{12} & 0 & \cdots & 0 \\ 0 & 0 & -a_{23} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & \cdots & -a_{n-1,n} \\ 0 & 0 & 0 & \ldots & 0 \end{pmatrix},$$

and the system become

$$(I + \overline{S})A\tilde{u} = (I + \overline{S})\tilde{b}$$

In the next section we formulate the block formulation of the above preconditioner to suit the structure of the MEDG SOR iterative method.

4 Preconditioned Modified Explicit Decoupled Group SOR (MEDG SOR)

For MEDG method, the matrix A, vectors $\tilde{u}$ and $\tilde{b}$ are of equal definitions as in Section (2). Since MEDG is a group method, $\overline{S}$ is obtained by taking the first upper diagonal groups of R_0 in Section (2) as the following:

$$\overline{S} = \begin{pmatrix} s_1 & & & \\ & s_1 & & \\ & & \ddots & \\ & & & s_1 \end{pmatrix}_{\frac{(N-2)^2}{2} \times \frac{(N-2)^2}{2}}, \quad s_1 = \begin{pmatrix} \tilde{0} & -R_{01} & & \\ & \tilde{0} & \ddots & \\ & & \ddots & -R_{01} \\ & & & \tilde{0} \end{pmatrix}_{(N-2)\times(N-2)}$$

Therefore, the preconditioner, $I + \overline{S}$ matrix will become

$$I + \overline{S} = \begin{pmatrix} s_2 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_2 \end{pmatrix}_{\frac{(N-2)^2}{2} \times \frac{(N-2)^2}{2}}, \quad s_2 = \begin{pmatrix} I_0 & -R_{01} & & \\ & I_0 & \ddots & \\ & & \ddots & -R_{01} \\ & & & I_0 \end{pmatrix}_{(N-2)\times(N-2)}$$

Here I_0 is a (2×2) identity matrix.

Now, obtain $\overline{A}$ by multiplying $I + \overline{S}$ with A.

$$\overline{A} = (I+\overline{S})A = \begin{pmatrix} s_2 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_2 \end{pmatrix} \begin{pmatrix} R_0 & R_1 & & \\ R_2 & R_0 & \ddots & \\ & \ddots & \ddots & R_1 \\ & & R_2 & R_0 \end{pmatrix}$$

$$= \begin{pmatrix} M_1 & R_1 & & \\ M_2 & M_1 & \ddots & \\ & \ddots & \ddots & R_1 \\ & & M_2 & M_1 \end{pmatrix}_{\frac{(N-2)^2}{2}\times\frac{(N-2)^2}{2}}$$

$$M_1 = s_2R_0 = \begin{pmatrix} M_{01} & M_{02} & & \\ R_{02} & M_{01} & \ddots & \\ & \ddots & \ddots & M_{02} \\ & & R_{02} & R_{00} \end{pmatrix}_{(N-2)\times(N-2)}, M_{01} = \begin{pmatrix} 1 & -\frac{1}{4} \\ -\frac{1}{4} & \frac{15}{16} \end{pmatrix},$$

$$M_{02} = \begin{pmatrix} 0 & 0 \\ 0 & -\frac{1}{16} \end{pmatrix}, \quad M_2 = s_2R_2 = \begin{pmatrix} M_{21} & M_{22} & & \\ R_{02} & M_{21} & \ddots & \\ & \ddots & \ddots & M_{22} \\ & & R_{02} & R_{02} \end{pmatrix}_{(N-2)\times(N-2)},$$

$$M_{21} = \begin{pmatrix} 0 & -\frac{1}{4} \\ 0 & -\frac{1}{16} \end{pmatrix}, \quad M_{22} = \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{16} \end{pmatrix}$$

R_{00}, R_{01} *and* R_{02} are defined earlier as in Section (2).

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

$$\begin{pmatrix} M_1 & R_1 & & & & \\ M_2 & M_1 & R_1 & & & \\ & M_2 & M_1 & R_1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & M_2 & M_1 & R_1 \\ & & & & M_2 & M_1 \end{pmatrix} \begin{pmatrix} \tilde{u}_2 \\ \tilde{u}_6 \\ \tilde{u}_{10} \\ \vdots \\ \tilde{u}_{N-8} \\ \tilde{u}_{N-4} \end{pmatrix} = \begin{pmatrix} s_2\tilde{v}_2 \\ s_2\tilde{v}_6 \\ s_2\tilde{v}_{10} \\ \vdots \\ s_2\tilde{v}_{N-8} \\ s_2\tilde{v}_{N-4} \end{pmatrix} \tag{12}$$

From Eq. (12),we will obtain

$$M_1\tilde{u}_j + R_1\tilde{u}_{j+4} = s_2\tilde{v}_j \qquad \text{for} \qquad j = 2 \tag{13}$$

$$M_2\tilde{u}_{j-4} + M_1\tilde{u}_j + R_1\tilde{u}_{j+4} = s_2\tilde{v}_j \qquad \text{for} \quad j = 6(2)N-8 \tag{14}$$

$$M_2\tilde{u}_{j-4} + M_1\tilde{u}_j = s_2\tilde{v}_j \qquad\qquad \text{for} \quad j = N-4 \tag{15}$$

Now, from Eq. (13) we can get:

For $i = 2$,

$$\begin{aligned}&\tilde{u}_{i,j} = M_{01}^{-1}\tilde{v}_{i,j} - M_{01}^{-1}R_{01}\tilde{v}_{i+4,j}\\ &-M_{01}^{-1}M_{02}\tilde{u}_{i+4,j} - M_{01}^{-1}R_{01}\tilde{u}_{i,j+4} - M_{01}^{-1}R_{01}\tilde{u}_{i+4,j+4}\end{aligned} \tag{16}$$

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

$$\begin{gathered}\tilde{u}_{i,j} = M_{01}^{-1}\tilde{v}_{i,j} - M_{01}^{-1}R_{01}\tilde{v}_{i+4,j} - M_{01}^{-1}R_{02}\tilde{u}_{i-4,j} - M_{01}^{-1}M_{02}\tilde{u}_{i+4,j}\\ -M_{01}^{-1}R_{01}\tilde{u}_{i,j+4} - M_{01}^{-1}R_{01}\tilde{u}_{i+4,j+4}\end{gathered} \tag{17}$$

For $i = N-4$,

$$\tilde{u}_{i,j} = R_{00}^{-1}\tilde{v}_{i,j} - R_{00}^{-1}R_{02}\tilde{u}_{i-4,j} - R_{00}^{-1}R_{01}\tilde{u}_{i,j+4} \tag{18}$$

The equation Eq. (16) can be written in matrix form as:

$$\begin{aligned}&14\begin{pmatrix} u_{i,j} \\ u_{i+2,j+2}\end{pmatrix} = \begin{pmatrix} 15 & 4 \\ 4 & 16\end{pmatrix}\begin{pmatrix} -2h^2 f_{i,j} + \frac{1}{4}u_{i-2,j-2} + \frac{1}{4}u_{i-2,j+2} + \frac{1}{4}u_{i+2,j-2} \\ -2h^2 f_{i+2,j+2}\end{pmatrix} - \begin{pmatrix} -1 & 0 \\ -4 & 0\end{pmatrix}\\ &\begin{pmatrix} -2h^2 f_{i+4,j} + \frac{1}{4}u_{i+2,j-2} + \frac{1}{4}u_{i+6,j-2} \\ -2h^2 f_{i+6,j+2}\end{pmatrix} - \begin{pmatrix} 0 & -\frac{1}{4} \\ 0 & -1\end{pmatrix}\begin{pmatrix} u_{i+4,j} \\ u_{i+5,j+1}\end{pmatrix} - \begin{pmatrix} -1 & 0 \\ -4 & 0\end{pmatrix}\begin{pmatrix} u_{i,j+4} \\ u_{i+1,j+5}\end{pmatrix}\\ &-\begin{pmatrix} -1 & 0 \\ -4 & 0\end{pmatrix}\begin{pmatrix} u_{i+4,j+4} \\ u_{i+5,j+5}\end{pmatrix}\end{aligned} \tag{19}$$

From equation Eq. (19), we will obtain a scheme as the following:

$$14u_{i,j} = \frac{15}{4}F_1 + \frac{1}{4}F_2 \ , \ 14u_{i+2,j+2} = F_1 + F_2 \tag{20}$$

where

$F_1 = u_{i-2,j-2} + u_{i-2,j+2} + u_{i+2,j-2} - 8h^2 f_{i,j}$

$F_2 = u_{i+2,j-2} + u_{i+6,j-2} + u_{i+5,j+1} + 4u_{i,j+4} + 4u_{i+4,j+4} - 32h^2 f_{i+2,j+2} - 8h^2 f_{i+4,j}$

Using the same manner as above, from Eq. (17) we can obtain a scheme as the following:

$$14u_{i,j} = \frac{15}{4}F_1 + \frac{1}{4}F_2 + \frac{15}{4}u_{i-3,j+1}, \quad 14u_{i+2,j+2} = F_1 + F_2 + u_{i-3,j+1} \tag{21}$$

where F_1 and F_2 are same expression as Eq. (19).

Also from Eq. (18), we can obtain a scheme as the following:

$$15u_{i,j} = 4F_1 + C_1 \ , \quad 15u_{i+2,j+2} = F_1 + C_2 \tag{22}$$

where:

$$C_1 = 4u_{i-3,j+1} + u_{i,j+4} - 8h^2 f_{i+2,j+2} \ , \qquad C_2 = u_{i-3,j+1} + 4u_{i,j+4} - 32h^2 f_{i+2,j+2}$$

and F_1 is same expression as Eq. (19).

We use the same way as above and from the Eq. (14) to get a formula as the following:

For i = 2,

$$\begin{aligned} &14u_{i,j} = \frac{15}{4}F_1 + \frac{1}{4}F_2 + \frac{7}{2}u_{i+1,j-3} - \frac{1}{4}u_{i+5,j-3}, \\ &14u_{i+2,j+2} = F_1 + F_2 - u_{i+5,j-3} \end{aligned} \tag{23}$$

For i = 6(2)N-8,

$$\begin{aligned} &14u_{i,j} = \frac{15}{4}F_1 + \frac{1}{4}F_2 + \frac{15}{4}(u_{i-3,j-3} + u_{i-3,j+1}) + \frac{1}{4}(16u_{i+1,j-3} - u_{i+5,j-3}), \\ &14u_{i+2,j+2} = F_1 + F_2 + u_{i-3,j-3} + u_{i-3,j+1} - u_{i+5,j-3} + 2u_{i+1,j-3} \end{aligned} \tag{24}$$

For i = N-4,

$$\begin{aligned} &15u_{i,j} = 4F_1 + C_1 + 4u_{i-3,j-3} + \frac{17}{4}u_{i+1,j-3}, \\ &15u_{i+2,j+2} = F_1 + C_2 + u_{i-3,j-3} + 2u_{i+1,j-3} \end{aligned} \tag{25}$$

where $F_1\&F_2$ are same expressions as Eq. (20) and $C_1\&C_2$ are same expressions as Eq. (22).

Also from the Eq. (15) we can get a formula as the following:

For i = 2,

$$\begin{aligned}14u_{i,j} &= \frac{15}{4}F_1 + \frac{1}{4}F_2 + \frac{7}{2}u_{i+1,j-3} - \frac{1}{4}u_{i+5,j-3} - u_{i,j+4} - u_{i+4,j+4} \\ 14u_{i+2,j+2} &= F_1 + F_2 - 4(u_{i,j+4} + u_{i+4,j+4}) - u_{i+5,j-3}\end{aligned} \tag{26}$$

For i=6(2)N-8,

$$\begin{aligned}&14u_{i,j} = \frac{15}{4}F_1 + \frac{1}{4}F_2 + \frac{15}{4}(u_{i-3,j-3} + u_{i-3,j+1}) - (u_{i,j+4} + u_{i+4,j+4}) \\ &-\frac{1}{4}u_{i+5,j-3} + 4u_{i+1,j-3}, \\ &14u_{i+2,j+2} = F_1 + F_2 - 4(u_{i,j+4} + u_{i+4,j+4}) + 2u_{i+1,j-3} + u_{i-3,j-3} \\ &+u_{i-3,j+1} - u_{i+5,j-3}\end{aligned} \tag{27}$$

For i = N-4,

$$\begin{aligned}15u_{i,j} &= 4F_1 + C_1 + \frac{1}{4}(16u_{i-3,j-3} + 17u_{i+1,j-3}) - u_{i,j+4}, \\ 15u_{i+2,j+2} &= F_1 + C_2 - 4u_{i,j+4} + u_{i-3,j-3} + 2u_{i+1,j-3}\end{aligned} \tag{28}$$

where $F_1\&F_2$ are same expressions as Eq. (20) and $C_1\&C_2$ are same expression as Eq. (22).

Finally, we have nine iterative cases for the preconditioned MEDG method and the difference equations can be transformed into the MEDG SOR schemes as:

For $i = 2, 3(2)N-8\&N-4,\ j = 2$

$$\begin{aligned}14u_{i,j}^{(k+1)} &= w[\frac{15}{4}F_1 + \frac{1}{4}F] + 14(1-w)u_{i,j}^{(k)}, \\ 14u_{i+2,j+2}^{(k+1)} &= w(F_1 + F_2) + 14(1-w)u_{i+2,j+2}^{(k)}\end{aligned}$$

$$\begin{aligned}14u_{i,j}^{(k+1)} &= w[\frac{15}{4}F_1 + \frac{1}{4}F_2 + \frac{15}{4}u_{i-3,j+1})] + 14(1-w)u_{i,j}^{(k)}, \\ 14u_{i+2,j+2}^{(k+1)} &= w(F_1 + F_2 + u_{i-3,j+1}) + 14(1-w)u_{i+2,j+2}^{(k)}\end{aligned}$$

$$\begin{aligned}15u_{i,j}^{(k+1)} &= w(4F_1 + C_1) + 15(1-w)u_{i,j}^{(k)}, \\ 15u_{i+2,j+2}^{(k+1)} &= w(F_1 + C_2) + 15(1-w)u_{i+2,j+2}^{(k)}\end{aligned}$$

Similarly, for i = 2, 6(2) N-8& N-4,   j = 6(2) N-8

$$14u_{i,j}^{(k+1)} = w(\frac{15}{4}F_1 + \frac{1}{4}F_2 + \frac{7}{2}u_{i+1,j-3} - \frac{1}{4}u_{i+5,j-3}) + 14(1-w)u_{i,j}^{(k)},$$
$$14u_{i+2,j+2}^{(k+1)} = w(F_1 + F_2 - u_{i+5,j-3}) + 14(1-w)u_{i+2,j+2}^{(k)}$$

$$14u_{i,j}^{(k+1)} = w[\frac{15}{4}F_1 + \frac{1}{4}F_2 + \frac{15}{4}(u_{i-3,j-3} + u_{i-3,j+1}) + \frac{1}{4}(16u_{i+1,j-3} - u_{i+5,j-3})],$$
$$+14(1-w)u_{i,j}^{(k)}$$
$$14u_{i+2,j+2}^{(k+1)} = w(F_1 + F_2 + u_{i-3,j-3} + u_{i-3,j+1} - u_{i+5,j-3} + 2u_{i+1,j-3}) + 14(1-w)u_{i+2,j+2}^{(k)}$$

$$15u_{i,j}^{(k+1)} = w[4F_1 + C_1 + 4u_{i-3,j-3} + \frac{17}{4}u_{i+1,j-3}] + 15(1-w)u_{i,j}^{(k)},$$
$$15u_{i+2,j+2}^{(k+1)} = w(F_1 + C_2 + u_{i-3,j-3} + 2u_{i+1,j-3}) + 15(1-w)u_{i+2,j+2}^{(k)}$$

And for i = 2, 6(2) N-8 & N-4, j =N-4

$$14u_{i,j}^{(k+1)} = w(\frac{15}{4}F_1 + \frac{1}{4}F_2 + \frac{7}{2}u_{i+1,j-3} - \frac{1}{4}u_{i+5,j-3} - u_{i,j+4} - u_{i+4,j+4}) + 14(1-w)u_{i,j}^{(k)},$$
$$14u_{i+2,j+2}^{(k+1)} = w[F_1 + F_2 - 4(u_{i,j+4} + u_{i+4,j+4}) - u_{i+5,j-3}] + 14(1-w)u_{i+2,j+2}^{(k)}$$

$$14u_{i,j}^{(k+1)} = w[\frac{15}{4}F_1 + \frac{1}{4}F_2 + \frac{15}{4}(u_{i-3,j-3} + u_{i-3,j+1}) - (u_{i,j+4} + u_{i+4,j+4}) - \frac{1}{4}u_{i+5,j-3} + 4u_{i+1,j-3}]$$
$$+14(1-w)u_{i,j}^{(k)},$$
$$14u_{i+2,j+2}^{(k+1)} = w[F_1 + F_2 - 4(u_{i,j+4} + u_{i+4,j+4}) + 2u_{i+1,j-3} + u_{i-3,j-3} + u_{i-3,j+1} - u_{i+5,j-3}]$$
$$+14(1-w)u_{i+2,j+2}^{(k)}$$

$$15u_{i,j}^{(k+1)} = w[4F_1 + C_1 + \frac{1}{4}(16u_{i-3,j-3} + 17u_{i+1,j-3}) - u_{i,j+4}] + 15(1-w)u_{i,j}^{(k)},$$
$$15u_{i+2,j+2}^{(k+1)} = w[F_1 + C_2 - 4u_{i,j+4} + u_{i-3,j-3} + 2u_{i+1,j-3}] + 15(1-w)u_{i+2,j+2}^{(k)}$$

where:

$$F_1 = u_{i-2,j-2} + u_{i-2,j+2} + u_{i+2,j-2} - 8h^2 f_{i,j},$$
$$F_2 = u_{i+2,j-2} + u_{i+6,j-2} + u_{i+5,j+1} + 4u_{i,j+4} + 4u_{i+4,j+4} - 32h^2 f_{i+2,j+2} - 8h^2 f_{i+4,j}$$

And

$$C_1 = 4u_{i-3,j+1} + u_{i,j+4} - 8h^2 f_{i+2,j+2}\,,\ C_2 = u_{i-3,j+1} + 4u_{i,j+4} - 32h^2 f_{i+2,j+2}\ .$$

Notice that the preconditioned formulas above are more complicated in terms of formulation but the effectiveness of this preconditioned method will be shown in the next section.

5 Numerical Experimentation and Results

In order to confirm the superiority of the preconditioned MEDG SOR iterative formula against non-preconditioned MEDG SOR iterative formula, experiments were carried out on the following model problem

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = (x^2 + y^2)e^{xy}, \quad (x, y) \in \Omega = [0, 1] \times [0, 1] \tag{29}$$

with Dirichlet boundary conditions satisfying the exact solution $u(x, y) = e^{xy}$, $(x, y) \in \partial\Omega$, $\partial\Omega$ is the boundary of Ω. The theoretical optimum relaxation factor ω_0 for implementing the group SOR iterative scheme can be computed from the formula

$$\omega_0 = \frac{2}{1 + \sqrt{1 - \rho^2(B)}} \tag{30}$$

where $\rho(B)$ is the spectral radius of the group Jacobian iterative matrix which can be estimated for the MEDG method as $\rho(B) \approx 1 - 4\sqrt{2}\pi^2 h^2$ [5]. The theoretical number of iteration to converge with the error tolerance tolerance ϵ can then be estimated as

$$k_0 \approx \frac{\ln \varepsilon}{\ln(\omega_0 - 1)} \tag{31}$$

From the boundary conditions given, we have a square solution domain. The values of u are calculated using different mesh sizes, 42, 74, 114,186 and 242. The value of tolerance is set to be $\varepsilon = 5 \times 10^{-6}$. The computer processing unit is Intel(R) Core(TM) 2Duo with memory of 3Gb and the software used to implement and generate the results is Developer C++ Version 4.9.9.2. Table 1 shows a comparison of the results for the preconditioned system and non-preconditioned system. The results show the corresponding number of iterations (k), value of optimum w obtained, and the maximum error (e).

From Table 1 above, it is shown that the preconditioner has succeeded in reducing the number of iterations. To illustrate this further, Figure 3 shows the comparison between the unpreconditioned and preconditioned methods. The execution time for the preconditioned system has been significantly reduced. The timings obtained as shown in Table 1 show that the execution times of the preconditioned MEDG SOR is only about 40% of the original MEDG SOR.

The convergence of the iteration methods relies on the spectral radius, which is defined as the largest of the moduli of the eigenvalues of the iteration matrix. It is stated and proven that a linear system with smaller value of spectral radius will have better convergence rate. Thus, the spectral radius of the coefficient matrix of the original system and the preconditioned system will be compared in order to justify the performance and suitability of the preconditioner. Since there are no special theoretical formulas that can be used to determine the spectral radiuses of the preconditioned matrices, therefore, we use Matlab software to estimate the values of the spectral radius. Table 2 shows a

Table 1: Comparison of the execution times, number of iterations and over relaxation parameter for MEDG SOR method with and without preconditioner

N	System without preconditioner				System with preconditioner			
	w	k	t	e	w	k	t	e
42	1.651	24	0	5.00E-06	1.544	19	0	3.47E-06
74	1.785	35	0.016	4.57E-06	1.611	32	0.007	3.09E-06
114	1.880	51	0.034	2.08E-06	1.639	43	0.015	3.55E-06
186	1.908	81	0.064	4.80E-06	1.684	59	0.025	4.24E-06
242	1.961	114	0.146	3.73E-06	1.691	84	0.061	2.18E-06

t is the execution time of the computer with corresponding w in seconds(s).

Table 2: Comparison of spectral radius between the original and the preconditioned linear systems

N	Original linear system	Preconditioned linear system
42	0.6286	0.4501
74	0.8442	0.5372
114	0.9213	0.6113
186	0.9412	0.6807
242	0.9601	0.7922

comparison of the spectral radius between the original and the preconditioned systems. Clearly it can be seen that the spectral radius of the preconditioned system is smaller compared to the original system, thus justifying our findings.

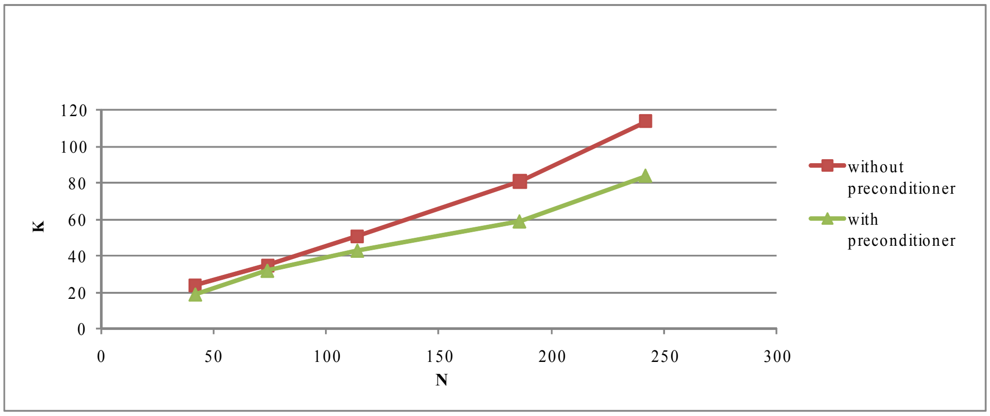

Figure 3: Comparison of the number of iterations(k)between the preconditioned and original systems for the MEDG SOR

6 Conclusions

In this work, we present the formulation of the preconditioner $(I+\overline{S})$ in block formulation for the MEDG SOR iterative method. We discover that the resulted preconditioned system has complicated terms in its formulation, nevertheless, this preconditioned schemes have shown improvements in the number of iterations and the execution time. Hence, we conclude that the proposed preconditioner is suitable to be implemented on the MEDG SOR method and is able to accelerate the rate of convergence of this method.

ACKNOWLEDGEMENTS. This research was supported by the Universiti Sains Malaysia Research University Grant (1001/PMATHS/817027).

References

[1] A.R. Abdullah, The Four Point Explicit Decoupled Group (EDG) Method : A Fast Poisson Solver, *International Journal of Comp. Math.* **38** (1991), 61-70.

[2] N. H. M. Ali and K. F. Ng, Modified explicit decoupled group method in the solution of 2-D elliptic PDEs, *12th WSEAS International Conference on Applied Mathematics, Cairo, Egypt,* 29-31 December(2007).

[3] N.H.M. Ali and K.F. Ng, A New Iterative Elliptic Solver On A Distributed PC Cluster, *Proceeding of the The Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies, Dunedin, New Zealand,* 1-4 December(2008), 47-53 .

[4] D.J. Evans and W.S. Yousif, The Implementation of The Explicit Block Iterative Methods On the Balance 8000 Parallel Computer, *Parallel Computing,* **16** (1990), 81-97.

[5] A. Gunawardena, S. Jain, and L. Snyder, Modified iterative methods for consistent linear systems, *Linear Algebra Appl.,* **154/156** (1991), 123-143.

[6] Y. Li, C. Li, and S. Wu, Improving AOR method for consistent linear systems, *Appl. Math. Comput,* **186** (2007), 379- 388.

[7] M.M. Martins, D.J. Evans, and W.S. Yousif, Further results on the preconditioned SOR method, *International Journal of Computer Mathematics,* **77(4)** (2001), 603-610.

[8] M. Othman and A.R. Abdullah, An Efficient Four Points Modified Explicit Group Poisson Solver, *International Journal of Computer Mathematics,* **76** (2000),203-217.

[9] A.M. Saeed and N.H.M. Ali, Preconditioned $(I+\overline{S})$ Group Iterative Methods on Rotated Grids, *Proceedings of the 5th Asian Mathematical Conference,Kuala Lumpur, Malaysia,* 22-26 June 2009.

[10] G.D. Smith, Numerical solution of partial differential equations, *Oxford, UK: Clarendon Press* 1985.

[11] M. Usui, T. Kohno, and H. Niki, On the preconditioned SOR method, *International. Journal of Computer Mathematics,* **59(1)** (1995), 123-130.

[12] W.S. Yousif and D.J. Evans, Explicit Group Over-relaxation Methods for Solving Elliptic Partial Differential Equations, *Mathematics and Computers in Simulation,* **28** (1986), 453-466.

Received: August, 2009