

Original Research Article

Efficient Group Iterative Method for Solving the Biharmonic Equation

Author details:

Abdulkafi Mohammed Saeed^{1*}

¹ *Department of Mathematics, College of Science, Qassim University, Saudi Arabia*

* *Corresponding author: E-mail address: abdelkafe@yahoo.com*

ABSTRACT

The biharmonic equations arise in many applications such as elasticity, fluid mechanics, and many other areas. In this paper, the combination of Explicit Decoupled Group (EDG) method with Successive Over Relaxation (SOR) is proposed for solving the biharmonic equation by reducing this equation into a coupled second order Poisson equations. Thus, this pair of Poisson equations can be easily solved using finite difference method, which discretizes the solution domain into a finite number of grids. The sparse linear system derived is usually solved by iterative methods which always take advantage of the existence of zeros in the coefficient matrix. However, such methods yield high number of iterations for convergence especially if the number of grid points is very large. To overcome of this problem, EDG SOR method formulated to accelerate the rate of convergence for the solution of these iterative methods. The numerical experiments carried out confirm the superiority of the introduced method over the classical standard five point SOR formula in terms of number of iterations and execution time.

Keywords: *Explicit Decoupled Group (EDG) method; Successive Over Relaxation Method (SOR); Biharmonic equation; Coupled second order Poisson equations*

1 INTRODUCTION

The biharmonic equation is a fourth order elliptic partial differential equation (pde) frequently arises in linear elasticity problems. Due to the existence of the fourth order derivative, the analytical solution for such equation is usually difficult to obtain. Consider the Dirichlet problem of the biharmonic equation defined on domain Ω and boundary $\partial\Omega$

$$\begin{aligned}\nabla^4 u(x, y) &= f(x, y), & (x, y) \in \Omega \\ u(x, y) &= g_1(x, y), & (x, y) \in \Omega \\ u_n(x, y) &= g_2(x, y), & (x, y) \in \Omega\end{aligned}\tag{1.1}$$

where
$$\nabla^4 = \frac{\partial^4}{\partial x^4} + 2 \frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4},$$

with the square boundary $\partial\Omega$, $u_n = \frac{\partial u}{\partial \hat{n}}$ is the normal derivative of u on $\partial\Omega$ and $\hat{n}$ is the unit normal vector. Several approximation methods carried out for the numerical solution of

Equation (1.1) [1, 2, 3, 4, 5, 6]. A popular technique, which was introduced by Smith [5] is to split (1.1) into a coupled of Poisson equations

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = u, \quad (1.2)$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \quad (1.3)$$

which may be solved by classical Poisson solvers such as the standard five point formula. Applying such formula on each grid point in the solution domain leads to a sparse linear system, which is preferably, solved by iterative methods since such method takes advantage of the existence of many zero elements in the matrix coefficient. However, this method suffers from slow convergence when the grid size grows. Abdullah [7] introduced the Explicit Decoupled Group (EDG) for solving such system of equations 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 [8]. The outline of this work is as follows: Section 2 gives an overview of the construction of EDG SOR iterative method. We apply the proposed EDG SOR formula to the coupled Poisson equations in section 3. Section 4 presented the numerical experimentation and results. The concluding remark is given in section 5.

2 Construction of EDG SOR Iterative Method

Poisson equation (1.3) may be approximated at the point (x_i, y_j) in many ways. Assume that a rectangular grid in the xy plane with equal grid spacing h in both directions with $x_i = ih, y_j = jh$ ($i, j = 0, 1, \dots, N$) are used, where $u_{i,j} = u(x_i, y_j)$ and $h = 1/N$. By neglecting terms of $O(h^2)$, we obtain the simplest approximation for (1.2) which is known as the standard five-point difference formula:

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

Another type of approximation that can represent the Poisson equation under study is the cross orientation which can be obtained by rotating the xy -plane clockwise by 45° Abdullah [7]. This will result in the rotated (skewed) five-point approximation formula:

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

In the (EDG) method, the grid points are gathered into groups which can consists of only 2 grid points. Each value for u of every grid point is approximated by the rotated five-point formula. These values are calculated with a sequence from left to right and then upwards. Hence, the iteration over the solution domain is only carried out on half the mesh points. Once convergence is achieved, the solution at the other half of the points is obtained directly once using the standard five-point difference formula [7, 9, 10, 11, 12].

Let us assume that the solution at any four points on the solution domain is solved using rotated five-point equation above. This results in a (4×4) system of equations

$$\begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & 0 & 0 \\ 0 & 0 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} u_{i,j} \\ u_{i+1,j+1} \\ u_{i+1,j} \\ u_{i,j+1} \end{bmatrix} = \begin{bmatrix} u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} - 2h^2 f_{i,j} \\ u_{i,j+2} + u_{i+2,j} + u_{i+2,j+2} - 2h^2 f_{i+1,j+1} \\ u_{i,j-1} + u_{i+2,j-1} + u_{i+2,j+1} - 2h^2 f_{i+1,j} \\ u_{i-1,j+2} + u_{i-1,j} + u_{i+1,j+2} - 2h^2 f_{i,j+1} \end{bmatrix} \quad (2.1)$$

This system lead to a decoupled system of two equations whose explicit form are given by

74

$$\begin{bmatrix} u_{i,j} \\ u_{i+1,j+1} \end{bmatrix} = \frac{1}{15} \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} - 2h^2 f_{i,j} \\ u_{i,j+2} + u_{i+2,j} + u_{i+2,j+2} - 2h^2 f_{i+1,j+1} \end{bmatrix} \quad (2.2)$$

76 and

$$\begin{bmatrix} u_{i,j} \\ u_{i+1,j+1} \end{bmatrix} = \frac{1}{15} \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} u_{i,j-1} + u_{i+2,j-1} + u_{i+2,j+1} - 2h^2 f_{i+1,j} \\ u_{i-1,j+2} + u_{i-1,j} + u_{i+1,j+2} - 2h^2 f_{i,j+1} \end{bmatrix} \quad (2.3)$$

78

	17		18		19		20	
	13		14		15		16	
	9		10		11		12	
	5		6		7		8	
	1		2		3		4	

79

80

81

Fig. 1. Natural group ordering

Theoretically, the application of either equation (2.2) or (2.3) to each of the group in Fig.1 with natural group ordering will result in a system of equations [9, 10]

84

85

$$A\tilde{u} = \tilde{b} \quad (2.4)$$

86

87

where:

88

$$A = \begin{bmatrix} R_0 & R_1 & & & \\ R_2 & R_0 & R_1 & & \\ & R_2 & R_0 & \ddots & \\ & & \ddots & \ddots & R_1 \\ & & & R_2 & R_0 \end{bmatrix}_{\frac{(N-1)^2}{2} \times \frac{(N-1)^2}{2}}, R_0 = \begin{bmatrix} R_{00} & R_{01} & & & \\ R_{02} & R_{00} & \ddots & & \\ & \ddots & \ddots & \ddots & R_{01} \\ & & & R_{02} & R_{00} \end{bmatrix}_{(N-1) \times (N-1)},$$

89

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

90

$$R_1 = \begin{bmatrix} R_{01} & R_{01} & & & \\ & R_{01} & \ddots & & \\ & & \ddots & \ddots & \\ & & & R_{01} & \\ & & & & R_{01} \end{bmatrix}_{(N-1) \times (N-1)}, R_2 = \begin{bmatrix} R_{02} & & & & \\ R_{02} & R_{02} & & & \\ & \ddots & \ddots & \ddots & \\ & & R_{02} & R_{02} & \\ & & & R_{02} & R_{02} \end{bmatrix}_{(N-1) \times (N-1)}$$

91

$$\tilde{u} = \begin{bmatrix} u_{\%}^1 \\ u_{\%}^2 \\ \vdots \\ u_{\%}^M \\ u_{\%}^{N-4} \\ u_{\%}^{N-2} \end{bmatrix}_{\frac{(N-1)}{2}}, \tilde{u}_{\%} = \begin{bmatrix} u_{\%}^{1,i} \\ u_{\%}^{2,i} \\ \vdots \\ u_{\%}^{N-2,i} \end{bmatrix}_{(N-1)} \quad \text{for } i = 1(2) N-2, \quad u_{\%}^{i,j} = \begin{bmatrix} u_{i,j} \\ u_{i+1,j+1} \end{bmatrix} \quad \text{for } i, j = 1(2) N-2$$

$$92 \quad \underline{b} = \begin{bmatrix} v_{\sim 1} \\ v_{\sim 3} \\ \vdots \\ v_{\sim N-4} \\ v_{\sim N-2} \end{bmatrix}_{\frac{(N-1)^2}{2}}, \quad \underline{v}_{\sim i} = \begin{bmatrix} v_{\sim 1,i} \\ v_{\sim 3,i} \\ \vdots \\ v_{\sim N-2,i} \end{bmatrix}_{(N-1)} \quad \text{for } i = 1(2)N-2, \quad v_{\sim i,j} = \begin{bmatrix} v_{i,j} \\ v_{i+1,j+1} \end{bmatrix} \quad \text{for } i,j = 1(2)N-2$$

$$93 \quad \text{For } i = 1, \quad \underline{v}_{1,i} = \begin{bmatrix} v_{1,i} \\ v_{2,i+1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}h^2 f_{1,i} + \frac{1}{4}u_{0,i-1} + \frac{1}{4}u_{2,i-1} + \frac{1}{4}u_{0,i+1} \\ -\frac{1}{2}h^2 f_{2,i+1} \end{bmatrix}$$

$$94 \quad \underline{v}_{k,i} = \begin{bmatrix} v_{k,i} \\ v_{k+1,i+1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}h^2 f_{k,i} + \frac{1}{4}u_{k-1,i-1} + \frac{1}{4}u_{k+1,i-1} \\ -\frac{1}{2}h^2 f_{k+1,i+1} \end{bmatrix} \quad \text{for } k = 3(2)N-4$$

$$95 \quad \underline{v}_{N-2,i} = \begin{bmatrix} v_{N-2,i} \\ v_{N-1,i+1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}h^2 f_{N-2,i} + \frac{1}{4}u_{N-3,i-1} + \frac{1}{4}u_{N-1,i-1} \\ -\frac{1}{2}h^2 f_{N-1,i+1} + \frac{1}{4}u_{N,i} + \frac{1}{4}u_{N,i+2} \end{bmatrix}$$

$$96 \quad \text{For } i = 3(2)N-4, \quad \underline{v}_{1,i} = \begin{bmatrix} v_{1,i} \\ v_{2,i+1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}h^2 f_{1,i} + \frac{1}{4}u_{0,i-1} + \frac{1}{4}u_{0,i+1} \\ -\frac{1}{2}h^2 f_{2,i+1} \end{bmatrix}$$

$$97 \quad \underline{v}_{k,i} = \begin{bmatrix} v_{k,i} \\ v_{k+1,i+1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}h^2 f_{k,i} \\ -\frac{1}{2}h^2 f_{k+1,i+1} \end{bmatrix} \quad \text{for } k = 3(2)N-4$$

$$98 \quad \underline{v}_{N-2,i} = \begin{bmatrix} v_{N-2,i} \\ v_{N-1,i+1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}h^2 f_{N-2,i} \\ -\frac{1}{2}h^2 f_{N-1,i+1} + \frac{1}{4}u_{N,i} + \frac{1}{4}u_{N,i+2} \end{bmatrix}$$

$$99 \quad \text{For } i = N-2, \quad \underline{v}_{\sim 1,i} = \begin{bmatrix} v_{1,i} \\ v_{2,i+1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}h^2 f_{1,i} + \frac{1}{4}u_{0,i-1} + \frac{1}{4}u_{0,i+1} \\ -\frac{1}{2}h^2 f_{2,i+1} + \frac{1}{4}u_{1,i+2} + \frac{1}{4}u_{3,i+2} \end{bmatrix}$$

$$100 \quad \underline{v}_{\sim k,i} = \begin{bmatrix} v_{k,i} \\ v_{k+1,i+1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}h^2 f_{k,i} \\ -\frac{1}{2}h^2 f_{k+1,i+1} + \frac{1}{4}u_{k,i+2} + \frac{1}{4}u_{k+2,i+2} \end{bmatrix} \quad \text{with } k = 3(2)N-4$$

$$101 \quad \underline{v}_{\sim N-2,i} = \begin{bmatrix} v_{N-2,i} \\ v_{N-1,i+1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}h^2 f_{N-2,i} \\ -\frac{1}{2}h^2 f_{N-1,i+1} + \frac{1}{4}u_{N-2,i+2} + \frac{1}{4}u_{N,i} + \frac{1}{4}u_{N,i+2} \end{bmatrix}$$

102 The EDG formula is hence written as the following

$$103 \quad u_{i,j} = \frac{1}{15}(4F_1 + F_2), \quad u_{i+1,j+1} = \frac{1}{15}(F_1 + 4F_2) \quad (2.5)$$

$$104 \quad \text{where: } F_1 = u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} - 2h^2 f_{i,j}, \quad F_2 = u_{i,j+2} + u_{i+2,j} + u_{i+2,j+2} - 2h^2 f_{i+1,j+1}$$

105 As can be seen, the matrix A is a block tridiagonal matrix. Therefore the SOR method
 106 applied on this system will be converge Smith [5]. In order to obtain the formula of EDG SOR
 107 method, we first need to derive the formulas of EDG Jacobi and EDG Gauss Seidel method
 108 respectively.

109 The iterative scheme for EDG Jacobi method is given by

$$110 \quad u_{i,j}^{(k+1)} = \frac{1}{15}(4F_1 + F_2), \quad u_{i+1,j+1}^{(k+1)} = \frac{1}{15}(F_1 + 4F_2)$$

$$111 \quad \text{where: } F_1 = \frac{1}{15}(u_{i-1,j-1}^{(k)} + u_{i+1,j-1}^{(k)} + u_{i-1,j+1}^{(k)} - 2h^2 f_{i,j}), \quad F_2 = \frac{1}{15}(u_{i,j+2}^{(k)} + u_{i+2,j}^{(k)} + u_{i+2,j+2}^{(k)} - 2h^2 f_{i+1,j+1})$$

112 The iterative scheme for EDG Gauss Seidel method is given by

$$113 \quad u_{i,j}^{(k+1)} = \frac{1}{15}(4F_1 + F_2), \quad u_{i+1,j+1}^{(k+1)} = \frac{1}{15}(F_1 + 4F_2) \quad (2.6)$$

$$114 \quad \text{where: } F_1 = \frac{1}{15}(u_{i-1,j-1}^{(k+1)} + u_{i+1,j-1}^{(k+1)} + u_{i-1,j+1}^{(k+1)} - 2h^2 f_{i,j}), \quad F_2 = \frac{1}{15}(u_{i,j+2}^{(k)} + u_{i+2,j}^{(k)} + u_{i+2,j+2}^{(k)} - 2h^2 f_{i+1,j+1})$$

115 Hence, the iterative scheme for EDG SOR method is given by

$$116 \quad u_{i,j}^{(k+1)} = \frac{1}{15}w(4F_1 + F_2) + (1-w)u_{i,j}^{(k)}, \quad u_{i+1,j+1}^{(k+1)} = \frac{1}{15}w(F_1 + 4F_2) + (1-w)u_{i+1,j+1}^{(k)} \quad (2.7)$$

117 where F_1, F_2 as shown in (2.6).

118

119 3 THE PROPOSED METHOD FOR SOLVING COUPLED POISSON EQUATIONS

120

121 Throughout this paper, we are interested in the domain of a unit square $(0 \times 1) \times (0 \times 1)$. This
 122 domain will be discretized into equally spaced grid of length h . Hence we have N^2 grids

123 and $(N-1)^2$ internal points, where $h = \frac{1}{N}$.

124 Consider a typical dirichlet biharmonic problem

125

$$126 \quad \nabla^4 u(x, y) = 0, \quad (x, y) \in (0 \times 1) \times (0 \times 1)$$

$$127 \quad u(x, y) = 0, \quad (x, y) \in \partial\Omega \quad (3.1)$$

$$128 \quad u_n(0, y) = u_n(1, y) = u_n(x, 0) = 0, u_n(x, 1) = 1, \quad 0 < x, y < 1$$

129

130 As in the coupled Poisson equations in (1.2) and (1.3), we need to approximate the
 131 boundary condition for v , by using the normal derivatives of u . Since the boundary of the
 132 domain is a square, we have $v = u_{nn}$ on the boundaries. Using centered approximations for
 133 u_{nn} , we have

$$134 \quad \left. \begin{aligned} v_{0,j} &= (2/h^2)u_{1,j} \\ v_{N,j} &= (2/h^2)u_{N,j} \end{aligned} \right\}, j = 1, \dots, N$$

$$135 \quad \left. \begin{aligned} v_{i,0} &= (2/h^2)u_{i,1} \\ v_{i,N+1} &= 2/h + (2/h^2)u_{i,N} \end{aligned} \right\}, i = 1, \dots, N \quad (3.2)$$

136 Solving (1.2) and (1.3) by the standard five point formula, together with the boundary
 137 conditions specified by (3.1) and (3.2) leads to the following equations

$$138 \quad Au = h^2 v, \quad (3.3)$$

$$Av = -(2/h^2)Mu - (2/h)K$$

139 where

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

$$A = \begin{bmatrix} D & I & & \\ I & D & & \\ & & \ddots & I \\ & & I & D \end{bmatrix}_{(N-1)^2 \times (N-1)^2}, D = \begin{bmatrix} -4 & 1 & & \\ 1 & -4 & & \\ & & \ddots & 1 \\ & & 1 & -4 \end{bmatrix},$$

$$M = \begin{bmatrix} S + I & & & \\ & S + I & & \\ & & \ddots & \\ & & & S + I \end{bmatrix}_{(N-1)^2 \times (N-1)^2}, S = \begin{bmatrix} 1 & & & \\ & & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}_{(N-1) \times (N-1)}$$

and K is the vector containing known values. By applying SOR iterative scheme for solving (3.3), we obtained

$$\begin{aligned} Au_{k+1} &= h^2 \bar{v}_k, & \bar{u}_{k+1} &= w_1 u_{k+1} + (1 - w_1) \bar{u}_k, \\ Av_{k+1} + (2/h^2)Mu_{k+1} &= \frac{k}{h^2}, & \bar{v}_{k+1} &= w_2 v_{k+1} + (1 - w_2) \bar{v}_k \end{aligned} \quad (3.5)$$

As mentioned before, the SOR scheme (3.5) above suffers from slow convergence when the number of grid points grow. Here we solve the pair of equations in (3.3) using the EDG SOR formula (2.4) by the same manner above.

4 NUMERICAL EXPERIMENTATION AND RESULTS

In this section we check the effectiveness of the proposed iterative method using the modal problem (3.1). For the purpose of comparison, we use a tolerance of $\varepsilon = 10^{-7}$ as the termination criteria and the range of the acceleration parameter is $1 \leq w \leq 2$. The computer processing unit is Intel(R) Core(TM) i5 with memory of 4Gb and the software used to implement and generate the results was Developer C++ Version 4.9.9.2.

The results are summarized in Table 1 which showed the comparison between the EDG formula and the classical standard five point formula(original system) when solving the biharmonic equation with SOR iterative scheme. We can easily observe that the number of iterations and elapsed time significantly reduced when using the new EDG SOR method for solving such equation.

Table 1. Comparison of number of iterations (k) and elapsed time (t) between the original system and EDG SOR system

Original system			EDG SOR system	
N	k	t	k	t
15	65	10.86	40	9.42
25	113	39.78	62	23.34
45	186	59.70	106	45.54
60	213	67.62	178	57.84
85	238	75.18	207	66.18

*Elapsed time measured in seconds

167

168

According to Smith [5], for numerical scheme with tolerance value of 10^{-q} , the number of iterations k is bounded by

$$k \geq \frac{q}{(-\log_{10} \rho)} \quad (4.1)$$

where ρ is the spectral radius of the iteration matrix which must be satisfied $0 < \rho < 1$ for the convergence of these types of iterative methods. Clearly k decreases as ρ decreases.

The iterative matrix for the SOR method case is given by

$$H(w) = (I - wL)^{-1}[(1-w)I + wU] \quad (4.2)$$

where L and U are strictly lower and upper triangular matrices respectively for matrix A in (2.1). Table 2 shows the comparison of the spectral radius of the iteration matrix between the original and the EDG SOR systems.

Table 2. Comparison of the spectral radius of the iteration matrix (ρ) between the original system and EDG SOR system

N	Original system	EDG SOR system
	spectral radius(ρ)	Spectral radius(ρ)
15	0.562	0.374
25	0.731	0.534
45	0.785	0.588
60	0.813	0.692
85	0.875	0.713

Clearly it can be seen that the spectral radius of the EDG SOR system is smaller compared to the original system, thus justifying our findings.

5 CONCLUSION

In this paper, we derive new iterative method which called Explicit Decoupled Group Successive Over-Relaxation (EDG SOR) for solving a biharmonic equation. The new schemes have shown improvements in the number of iterations and the execution time experimentally. Hence, we conclude that the proposed group iterative method is suitable for solving a coupled of Poisson equations and it is able to accelerate the rate of convergence of the solution.

ACKNOWLEDGEMENTS

The author acknowledges the Project No. (2787) in Qassim University for the completion of this article.

REFERENCES

- [1] McLaurin J. W. A general coupled equation approach for solving the biharmonic boundary value problem. SIAM J. Num. Anal. 1974; 11:14-33.
- [2] Greenbaum A. , Greengard L., Mayo A. On the numerical-solution of the biharmonic equation in the plane. PHYSICA D. 1992; 60:216-225.
- [3] Mayo A, Greenbaum A. Fast parallel iterative solution of Poisson's and the biharmonic equations on irregular regions. SIAM J. Sci. Stat. Comput. 1992; 13:101-118.

- 209
210 [4] Smith J. The coupled equation approach to the numerical solution of the biharmonic
211 equation by finite differences. SIAM Journal of Numerical Analysis. 1968; 5(2): 323-
212 339.
213
214 [5] Smith G.D. Numerical solution of partial differential equations. New York: Clarendon
215 Press-Oxford; 1985.
216
217 [6] Stephenson J.W. Single cell discretizations of order two and four for biharmonic
218 problems. Journal of Computational Phys. 1984; 55(1): 65-80.
219
220 [7] Abdullah A. R. The Four Point Explicit Decoupled Group (EDG) Method: A Fast
221 Poisson Solver. International Journal of Comp. Math. 1991; 38: 61-70.
222
223 [8] Yousif W. S., Evans D. J. Explicit Group Over-relaxation Methods for Solving Elliptic
224 Partial Differential Equations, Mathematics and Computers in Simulation. 1986; 28:
225 453-466.
226
227 [9] Saeed A. M, Ali N. H. M. Preconditioned $(I + \bar{S})$ Group Iterative Methods on Rotated
228 Grids. European Journal in Scientific Research. 2009; 37(2):278-287.

229 [10] Saeed A. M, Ali N. H. M. On the Convergence of the Preconditioned Group Rotated
230 Iterative Methods In The Solution of Elliptic PDEs. Applied Mathematics &
231 Information Sciences. 2011; 5(1): 65-73.

232 [11] Saeed A. M, Ali N. H. M. Accelerated Solution Of Two Dimensional Diffusion
233 Equation. World Applied Sciences Journal. 2014; 32(9): 1906-1912.

234 [12] Saeed A. M. Fast Iterative Solver For The 2-D Convection-Diffusion Equations.
235 Journal of Advances In Mathematics. 2014; 9(6): 2773-2782.
236