International Mathematical Forum, Vol. 12, 2017, no. 16, 771 - 779
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/imf.2017.7654

Explicit Decoupled Group Iterative Method for the Triangle Element Solution of 2D Helmholtz Equations

Mohd Kamalrulzaman Md Akhir and Jumat Sulaiman

Faculty of Science and Natural Resources
Universiti Malaysia Sabah
88400 Kota Kinabalu Sabah Malaysia

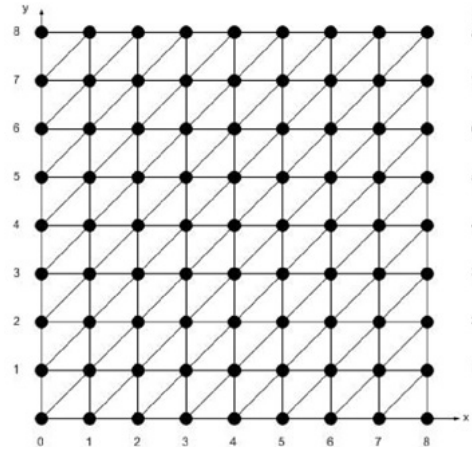
Abstract

Numerical methods are used to solve the two-dimensional Helmholtz equation. The Explicit Decoupled Group (EDG) method is applied to the linear system generated from the first-order triangle finite element approximation. Numerical experiments are conducted to verify the effectiveness of the method.

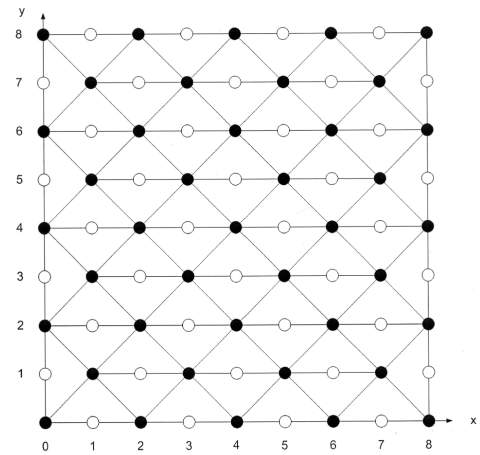
Keywords: Explicit Decoupled Group; Helmholtz; Galerkin Scheme; Triangle element; half-sweep iterations

1 Introduction

Finite element methods allow weighted residual schemes (e.g., subdomain, collocation, least-squares, moments, and Galerkin methods) to be used to gain approximate solutions. Using the first-order triangle finite element approximation based on the Galerkin scheme, this paper describes an Explicit Decoupled Group (EDG) method with the GaussSeidel strategy for solving the two-dimensional (2D) Helmholtz equation. This approach is compared with the Full-Sweep GaussSeidel (FSGS) and Explicit Group (EG) methods. To investigate the effectiveness of EDG, consider the 2D Helmholtz equation



(a)



(b)

Figure 1: (a) and (b) show the definition of the hat function $R_{i,j}(x,y)$, of full- and half-sweep triangle elements at the solution domain.

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - \alpha U = f(x,y), \quad (x,y) \in [a,b] \times [a,b] \tag{1}$$

with the dirichlet boundary conditions

$$\begin{array}{llll} U(x,a) = g_1(x), & a \leq x \leq b, & U(x,b) = g_2(x), & a \leq x \leq b, \\ U(a,y) = g_3(y), & a \leq x \leq b, & U(b,y) = g_4(x), & a \leq y \leq b, \end{array}$$

Here, α is a non-negative constant and $f(x,y)$ is a function with sufficient smoothness. To formulate the full- and half-sweep triangle element approximations for (1), we focus on the uniform node points shown in Figure 1. Based on Figure 1, the solution domain is discretized uniformly in the x and y directions with a mesh size h, defined as

$$\Delta x = \Delta y = h = \frac{b-a}{n}, \quad m = n+1. \tag{2}$$

The full- and half-sweep networks of triangle finite elements are guidelines for the triangle finite element approximations. These approximations form systems of finite element approximations for (1). Using the same concept of half-sweep iterations applied to finite difference methods [1], finite element networks consist of several triangle elements, each with two solid node points of type $\bullet$ (see Figure 1). Consequently, the full- and half-sweep iterative algorithms are applied to such node points until the iterative convergence criterion is satisfied. Approximate solutions at the remaining node points (i.e., points of type ∘) are calculated directly [5, 6, 7].

In the remainder of this paper, we discuss the finite element method based on the Galerkin scheme for discretizing (1), describe the proposed method for solving linear systems generated from triangle element approximations, present some numerical examples, and analyze our results.

2 Half-Sweep Triangle Element Approximations

Using the first-order triangle finite element approximation, a discretization based on the Galerkin scheme gives an approximation of (1). Considering node points of type •, the general approximation, in the form of an interpolation function for an arbitrary triangle element e, is given by [2, 3]

$$\widetilde{U}^{[e]}(x,y) = N_1(x,y)U_1 + N_2(x,y)U_2 + N_3(x,y)U_3 \qquad (3)$$

and the shape functions $N_k(x,y)$, $k = 1,2,3$, can be written as:

$$N_k\left(x,y\right) = \frac{1}{|A|}\left(a_k + b_k x + c_k y\right), \quad k = 1,2,3 \qquad (4)$$

where,

$$|A| = x_1\left(y_2 - y_3\right) + x_2\left(y_3 - y_1\right) + x_3\left(y_1 - y_2\right)$$

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} x_2y_3 - x_3y_2 \\ x_3y_1 - x_1y_3 \\ x_1y_2 - x_2y_1 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} y_2 - y_3 \\ y_3 - y_1 \\ y_1 - y_2 \end{bmatrix}, \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} x_3 - x_2 \\ x_1 - x_3 \\ x_2 - x_3 \end{bmatrix},$$

The first-order partial derivatives of the shape functions can be written as:

$$\left.\begin{array}{rl} \frac{\partial}{\partial x}\left(N_k\left(x,y\right)\right) & = \frac{b_k}{\det A} \\ \frac{\partial}{\partial y}\left(N_k\left(x,y\right)\right) & = \frac{c_k}{\det A} \end{array}\right\}, \quad k = 1,2,3 \qquad (5)$$

Figure 2 illustrates the hat function $R_{r,s}\left(x,y\right)$ in the solution domain [8, 9]. The approximation functions for the full- and half-sweep cases over the entire domain in (1) become:

$$\widetilde{U}(x,y) = \sum_{r=0}^{m}\sum_{s=0}^{m} R_{r,s}(x,y)U_{r,s} \qquad (6)$$

$$\widetilde{f}(x,y) = \sum_{r=0}^{m}\sum_{s=0}^{m} R_{r,s}(x,y)f_{r,s} \qquad (7)$$

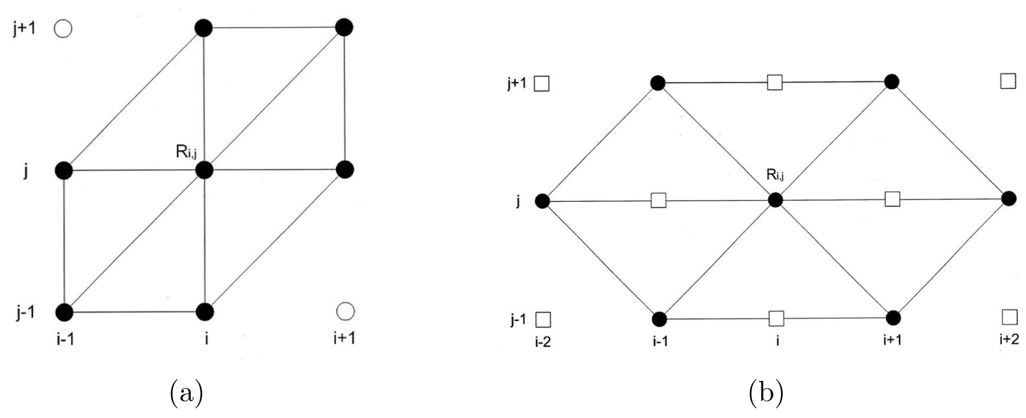


Figure 2: (a) and (b) illustrate the hat function $R_{i,j}(x,y)$, of full- and half-sweep triangle elements, respectively, on the solution domain.

and

$$\widetilde{U}(x,y) = \sum_{r=0,2,4}^{m} \sum_{s=0,2,4}^{m} R_{r,s}(x,y) f_{r,s} + \sum_{r=1,3,5}^{m-1} \sum_{s=1,3,5}^{m-1} R_{r,s}(x,y) f_{r,s} \tag{8}$$

$$\widetilde{f}(x,y) = \sum_{r=0,2,4}^{m} \sum_{s=0,2,4}^{m} R_{r,s}(x,y) f_{r,s} + \sum_{r=1,3,5}^{m-1} \sum_{s=1,3,5}^{m-1} R_{r,s}(x,y) f_{r,s} \tag{9}$$

Similarly, the approximate function $f(x,y)$ for all cases can be easily defined via the hat function. Indeed, (6) and (8) can be denoted as approximate solutions for (1).

To construct the full- and half-sweep linear finite element approximations for (1), we must take account of the Galerkin scheme. Consider the Galerkin residual method [10, 11, 12] to be:

$$\iint_D R_{i,j}(x,y) E_{i,j}(x,y) = 0, \quad i,j = 0,1,2,\ldots,m \tag{10}$$

where, $E(x,y) = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - \alpha U - f(x,y)$ is a residual function. Applying Greens theorem, (10) can be written as

$$\int_{\lambda}\left(-R_{i,j}\left(x,y\right)\frac{\partial u}{\partial y}dx+R_{i,j}\left(x,y\right)\frac{\partial u}{\partial x}dy-\alpha R_{i,j}\left(x,y\right)U\right)$$
$$-\int_a^b\int_a^b\left(\frac{\partial R_{i,j}\left(x,y\right)}{\partial x}\frac{\partial u}{\partial x}+\frac{\partial R_{i,j}\left(x,y\right)}{\partial y}\frac{\partial u}{\partial y}+\alpha R_{i,j}\left(x,y\right)U\right)dxdy=F_{i,j} \quad (11)$$

where,

$$F_{i,j}=\int_a^b\int_a^b R_{i,j}\left(x,y\right)f\left(x,y\right)dxdy.$$

Replacing (5) and imposing the boundary conditions in (1), (11) generates a linear system for all cases. This can be simplified to:

$$-\sum\sum K^{*}_{i,j,r,s}=\sum\sum C^{*}_{i,j,r,s} \quad (12)$$

where,

$$C^{*}_{i,j,r,s}=\int_a^b\int_a^b\left(R_{i,j}\left(x,y\right)R_{r,s}\left(x,y\right)\right)dxdy.$$

Essentially, for the full- and half-sweep cases, (12) can be straightforwardly expressed in stencil form as:
1. Full-sweep:[9]

$$\begin{bmatrix} & \beta_2 & \beta_3\\ \beta_2 & \beta_1 & \beta_2\\ \beta_3 & \beta_2 & \end{bmatrix}U_{i,j}=\beta_0\begin{bmatrix} & 1 & 1\\ 1 & 6 & 1\\ 1 & 1 & \end{bmatrix}f_{i,j} \quad (13)$$

where

$$F_{i,j}=\beta_0\left(f_{i-1,j}+f_{i+1,j}+f_{i,j-1}+f_{i,j+1}+f_{i-1,j-1}+f_{i+1,j+1}+6f_{i,j}\right)$$
$$\beta_0=h^2/12\ ,\ \ \beta_1=-\left(4+\alpha 6\beta_0\right),\ \ \beta_2=1-r\beta_0,\ \ \beta_3=r\beta_0$$

2. Half-sweep:[8]

$$\begin{bmatrix} \gamma_2 & & \gamma_2 & \\ & \gamma_1 & & \gamma_3 \\ \gamma_2 & & \gamma_2 & \end{bmatrix} U_{i,j} = \gamma_0 \begin{bmatrix} 1 & & 1 & \\ & 5 & & 1 \\ 1 & & 1 & \end{bmatrix} f_{i,j}, \quad i = 1$$

$$\begin{bmatrix} & \gamma_2 & & \gamma_2 & \\ \gamma_3 & & \gamma_1 & & \gamma_3 \\ & \gamma_2 & & \gamma_2 & \end{bmatrix} U_{i,j} = \gamma_0 \begin{bmatrix} & 1 & & 1 & \\ 1 & & 6 & & 1 \\ & 1 & & 1 & \end{bmatrix} f_{i,j}, \quad i \neq 1, n$$

$$\begin{bmatrix} & \gamma_2 & & \gamma_2 \\ \gamma_3 & & \gamma_1 & \\ & \gamma_2 & & \gamma_2 \end{bmatrix} U_{i,j} = \gamma_0 \begin{bmatrix} & 1 & & 1 \\ 1 & & 5 & \\ & 1 & & 1 \end{bmatrix} f_{i,j} \tag{14}$$

where

$$F_{i,j} = \gamma_0 \left(f_{i-1,j-1} + f_{i+1,j-1} + f_{i-1,j+1} + f_{i+2,j} + 5f_{i,j} \right), \quad i = 1$$
$$F_{i,j} = \gamma_0 \left(f_{i-1,j-1} + f_{i+1,j-1} + f_{i-1,j+1} + f_{i+1,j+1} + f_{i-2,j} + f_{i+2,j} + 6f_{i,j} \right), \quad i \neq 1,\ n$$
$$F_{i,j} = \gamma_0 \left(f_{i-1,j-1} + f_{i+1,j-1} + f_{i-1,j+1} + f_{i+1,j+1} + f_{i-2,j} + 5f_{i,j} \right), \quad i = n$$
$$\gamma_0 = h^2/6\ ,\ \gamma_1 = -\left(4 + \alpha 6\gamma_0\right),\ \gamma_2 = 1 - r\upsilon_0,\ \gamma_3 = r\gamma_0.$$

In fact, the stencil forms in (13) and (14) implicate seven node points in the approximation equations.

3 Formulation of EDG Method

The EDG method [1] was introduced to solve 2D Poisson equations using a five-point rotated finite difference approximation. An iterative implementation involves only red node points until convergence, with approximate values of the remaining node points computed by direct methods. (For a discussion of this method based on the finite difference approach, see [1, 6, 7]). Let a four-point group be considered to form the (4x4) linear system [7]

$$\left[\begin{array}{cc|cc} \gamma_1 & \gamma_2 & 0 & 0 \\ \gamma_2 & \gamma_1 & 0 & 0 \\ \hline 0 & 0 & \gamma_1 & \gamma_2 \\ 0 & 0 & \gamma_2 & \gamma_1 \end{array}\right] \left[\begin{array}{c} U_{i,j} \\ U_{i+1,j+1} \\ \hline U_{i+1,j} \\ U_{i,j+1} \end{array}\right] = \left[\begin{array}{c} S_1 \\ S_2 \\ \hline S_3 \\ S_4 \end{array}\right] \tag{15}$$

where

$$S_1 = U_{i-1,j-1} + U_{i-1,j+1} + U_{i+1,j-1} - F_{i,j}, \quad S_2 = U_{i,j+2} + U_{i+2,j} + U_{i+2,j+2} - F_{i+1,j+1},$$
$$S_3 = U_{i,j-1} + U_{i+2,j-1} + U_{i+2,j+1} - F_{i+1,j}, \quad S_4 = U_{i,j+1} + U_{i-1,j} + U_{i-1,j+2} - F_{i,j+1}\,.$$

The system in (15) can be rewritten as two (2x2) independent linear systems. The EDG method is

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix}^{(k+1)} = \frac{1}{\gamma_1^2 - \gamma_2^2} \begin{bmatrix} \gamma_1 & -\gamma_2 \\ -\gamma_2 & \gamma_1 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \tag{16}$$

where

$$S_1 = U_{i-1,j-1}^{(k+1)} + U_{i-1,j+1}^{(k+1)} + U_{i+1,j-1}^{(k+1)} - F_{i,j},$$
$$S_2 = U_{i,j+2}^{(k)} + U_{i+2,j}^{(k)} + U_{i+2,j+2}^{(k)} - F_{i+1,j+1}.$$

and

$$\begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix}^{(k+1)} = \frac{1}{\gamma_1^2 - \gamma_2^2} \begin{bmatrix} \gamma_1 & -\gamma_2 \\ -\gamma_2 & \gamma_1 \end{bmatrix} \begin{bmatrix} S_3 \\ S_4 \end{bmatrix} \tag{17}$$

where

$$S_3 = U_{i,j-1}^{(k+1)} + U_{i+2,j-1}^{(k+1)} + U_{i+2,j+1}^{(k+1)} - F_{i+1,j},$$
$$S_4 = U_{i,j+1}^{(k)} + U_{i-1,j}^{(k)} + U_{i-1,j+2}^{(k)} - F_{i,j+1}\,.$$

4 Numerical Results

To compare the iterative methods described in the previous section, numerical experiments were conducted on:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - \alpha U = -(\cos(x+y) + \cos(x-y)) - \alpha \cos(x)\cos(y). \tag{18}$$

$$U(x,0) = \cos x, \qquad U(x,\frac{\pi}{2}) = 0,$$
$$U(0,y) = \cos y, \qquad U(\pi,y) = -\cos y.$$

The exact solution is

$$U(x,y) = \cos(x)\cos(y).$$

The effectiveness of the proposed EDG is compared with that of EG and FSGS. The experiments were conducted on an Intel(R) Core (TM) i7 CPU 860@2.80 GHz with 6 GB memory. The point GS method acts as the control in the comparison of numerical results. The number of iterations, execution time, and maximum/absolute error are used for comparison. Convergence was determined by the tolerance $\varepsilon = 10^{-10}$ and experiments were conducted for mesh sizes of 284, 308, 332, and 356. The numerical results are presented in Table 1.

Table 1: Comparison of the number of iterations, execution time (seconds) and maximum absolute error for the iterative methods

n	Methods	k	t	Abs.Error
284	FSGS	93990	273.61	1.6410e-6
	EG	60812	14.02	5.8423e-7
	EDG	46449	73.06	4.0307e-6
308	FSGS	109305	369.36	1.6046e-6
	EG	70745	197.64	7.1419e-7
	EDG	54099	101.64	3.4318e-6
332	FSGS	125668	522.71	1.6095e-6
	EG	81362	260.94	7.5902e-7
	EDG	62230	137.97	2.9573e-6
356	FSGS	148976	143065	1.6476e-6
	EG	92655	341.31	8.1613e-7
	EDG	70880	191.71	2.5749e-6

Table 2: : The reduction percentages of the EDG and EG methods compared with FSGS method.

Methods	k	t
EG	47.49–47.61	75.40–77.27
EDG	50.52–72.68	81.16–83.65

5 Conclusions

We have presented an application of half-sweep iterations with a block method for solving sparse linear systems generated from the first-order triangle finite element approximation using the Galerkin scheme. The numerical results show that EDG is superior to EG and FSGS in terms of the number of iterations and execution time. This is mainly because of the reduced computational complexitythe implementation of EDG only considers approximately half of all interior node points in the solution domain.

References

[1] A.R. Abdullah, The Four Point Explicit Decoupled Group (EDG) Method: A Fast Poisson Solver, *Intern. J. of Comp. Math.*, **38** (1991), 61-70. https://doi.org/10.1080/00207169108803958

[2] C.A.J. Fletcher, The Galerkin method: An introduction, In *Numerical Simulation of Fluid Motion*, J. Noye (Ed.), North-Holland Publishing

Company, Amsterdam, 1978, 113-170.

[3] C.A.J. Fletcher, *Computational Galerkin Methods*, Springer Series in Computational Physics, Springer-Verlag, New York, 1984. https://doi.org/10.1007/978-3-642-85949-6

[4] D.J. Evans, *Group Explicit Methods for the Numerical Solutions of Partial Differential Equations*, Australia, Gordon and Breach Science Publishers, 1997.

[5] D.M. Young, *Iterative Solution of Large Linear Systems*, London, Academic Press, 1971. https://doi.org/10.1016/c2013-0-11733-3

[6] M.K.M. Akhir, M. Othman, J. Sulaiman, Z.A. Majid and M. Suleiman, Half Sweep Iterative Method for Solving Two-Dimensional Helmholtz Equations, *Int. J. of App. Math. and Stat.*, **29** (2012), 101-109.

[7] M.K.M. Akhir, M. Othman, J. Sulaiman, Z.A. Majid and M. Suleiman, Four Point Explicit Decoupled Group Iterative Method Applied to Two-Dimensional Helmholtz Equation, *Int. J. Math. Anal.*, **20** (2012), 963-974.

[8] M.K.M. Akhir and J. Sulaiman, HSGS Method for the Finite Element Solution of Two-Dimensional Helmholtz Equations, *Global J. of Math.*, **40** (2015), 367-373.

[9] M.K.M. Akhir and J. Sulaiman, Numerical Solutions of Helmholtz equations the Triangle Element using Explicit Group Method, *J. KALAM*, (in press).

[10] O.C. Zienkiewicz, Why finite elements?, ln *Finite Elements in Fluids*, R.H. Gallagher, J.T. Oden, C. Taylor, O.C. Zienkiewicz, (Eds), John Wiley and Sons, London, 1975.

[11] P.E. Lewis and J.P. Ward, *The Finite Element Method: Principles and Applications*, Addison-Wesley Publishing Company, Wokingham, 1991.

[12] R. Vichnevetsky, *Computer Methods for Partial Differential Equations*, Vol. I, New Jersey, Prentice-Hall, 1981.

Received: July 9, 2017; Published: August 16, 2017