

An Economic Elliptic Equation Solver for Rectangular Regions

P.K. SHARMA* and MANOJ KUMAR AGARWAL**

*Department of Mathematics, D.A.V. (P.G.) College, and

**Wildlife Institute of India, Chandrabani, Dehradun, India

ABSTRACT. An algorithm is presented which solves the system of simultaneous linear equations generated after the application of finite differences to the elliptic equation for rectangular regions. The subroutine made is arranged to take minimum storage and minimum execution time. It is tested extensively on three examples whose exact solutions are known apriori. The solutions obtained are compared with the exact results and the results obtained by some authors previously. The present results are found in good agreement with all these results.

1. Introduction

A variety of physical phenomena are governed by elliptic partial differential equations, e.g., Laplace's equation, Poisson's equation and Navier-Stokes equation.

Let the elliptic partial differential equation be of the form

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + a(x, y) \frac{\partial u}{\partial x} + b(x, y) \frac{\partial u}{\partial y} + c(x, y)u = d(x, y) \quad (1)$$

where a, b, c and d are functions of two space coordinates (x, y) . We wish to obtain the numerical solution for the function u at the nodes of rectangular grid (Fig. 1) defined over the domain D which is given by

$$D = \{(x, y) : X_1 \leq x \leq X_2, Y_1 \leq y \leq Y_2\}$$

The boundary conditions for domain D are of the form

$$\left. \begin{aligned} u(X_1, y) = g_1(y) \quad , \quad u(X_2, y) = g_2(y) \\ u(x, Y_1) = g_3(x) \quad , \quad u(x, Y_2) = g_4(x) \end{aligned} \right] \quad (2)$$

$$\left. \begin{aligned} \frac{\partial u(X_1, y)}{\partial x} = h_1(y) \quad , \quad u(X_2, y) = h_2(y) \\ u(x, Y_1) = h_3(x) \quad , \quad u(x, Y_2) = h_4(x) \end{aligned} \right] \quad (3)$$

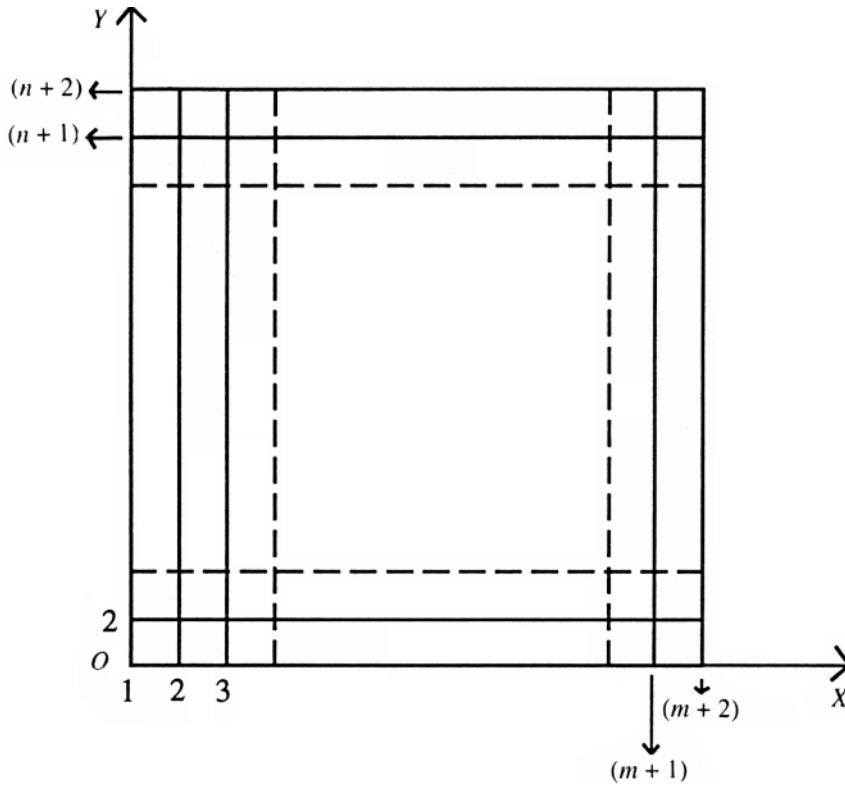


FIG. 1.

Equation (1) with boundary conditions (2) or (3) arises in many scientific and engineering applications. For examples Poisson's equation in cartesian coordinates and cylindrical coordinates, Laplace's equation and Navier-Stokes equations all are elliptic partial differential equations and extensively applied in heat and mass transfer, fluid flow problems and potential problems.

Finite difference approximation of Equation (1) gives rise to block tridiagonal system of linear equations whose solution is a problem in itself. Iterative methods like Jacobi's and Gauss-Seidel to solve such a system sometimes fail to converge because diagonal-dominance property does not always hold good. Young^[1] has described several iterative techniques to solve linear system of equations which are sufficiently large. A second approach is to employ direct methods to solve such a system. Fox^[2] has described some direct schemes like factorization technique and Gaussian elimination to solve such a system. Gupta and Manohar^[3] have used LEQT1B, LEQT2B solvers or GELB solver. However, these solvers take a little more storage and computer time, since they do not take block tridiagonal property of the system into consideration. Linger^[4] has applied a

semi-direct method to solve Poisson's equation. Elsner and Mehrmann^[5] have dealt in detail on the convergence conditions of block iterative methods. The block tridiagonal system of linear equations is a sparse coefficient matrix system and it is possible to take the advantage of the sparseness in order to reduce both computation time and storage requirements. Duff^[6], Erisman and Reid^[7] have dealt in detail with the direct methods for sparse matrices. Jennings^[8] has also given methods like elimination using submatrices to deal with a sparse structure of such type.

The present work also employs a direct elimination technique to solve a block tridiagonal system of linear equations. The program is prepared in such a manner that it does not require zeros or units to be stored so that operation counts and memory requirements are sufficiently reduced and round-off errors are taken care off.

2. Structure of System of Equations

The rectangular region (Fig. 1) is divided into intervals so that there are in total $(m + 2)$ and $(n + 2)$ nodes along X and Y directions and out of these m and n are interior nodes (at which value of function u is required). The finite difference representation at node (i, j) in Equation (1) is expressed as

$$\begin{aligned}
 u_{ij} &= u(x_i, y_j) \\
 \left. \begin{aligned}
 \left(\frac{\partial^2 u}{\partial x^2}\right)_{(ij)} &\approx \frac{u_{i+1j} - 2u_{ij} + u_{i-1j}}{h^2} \\
 \left(\frac{\partial^2 u}{\partial y^2}\right)_{(ij)} &\approx \frac{u_{ij+1} - 2u_{ij} + u_{ij-1}}{k^2} \\
 \left(\frac{\partial u}{\partial x}\right)_{(ij)} &\approx \frac{u_{i+1j} - u_{i-1j}}{2h} \\
 \left(\frac{\partial u}{\partial y}\right)_{(ij)} &\approx \frac{u_{ij+1} - u_{ij-1}}{2k}
 \end{aligned} \right\} \quad (4)
 \end{aligned}$$

where h and k are mesh-widths along x and y directions respectively.

Above finite difference approximation to Equation (1) gives rise to following block tridiagonal system of equations.

$$\begin{bmatrix}
 A_1 & B_1 & & & \\
 C_2 & A_2 & B_2 & & \\
 & \ddots & \ddots & \ddots & \\
 & & C_{n-1} & A_{n-1} & B_{n-1} \\
 & & & C_n & A_n
 \end{bmatrix}
 {}_{mn \times mn}
 \begin{bmatrix}
 \bar{u}_1 \\
 | \\
 | \\
 | \\
 | \\
 | \\
 \bar{u}_{n-1} \\
 \bar{u}_n
 \end{bmatrix}
 {}_{mn \times 1}
 =
 \begin{bmatrix}
 V_1 \\
 | \\
 | \\
 | \\
 | \\
 | \\
 | \\
 |
 \end{bmatrix}
 \quad (5)$$

where A_j 's, $j = 1(1)n$ are tridiagonal matrices B_j 's and C_j 's, $j = 1(1)n$ are diagonal matrices given by

$$A_j = \begin{bmatrix} \beta_{1j} & \gamma_{1j} & & & \\ \alpha_{2j} & \beta_{2j} & \gamma_{2j} & & \\ & & \ddots & \ddots & \\ & & & \beta_{m-1j} & \gamma_{m-1j} \\ & & & \alpha_{mj} & \beta_{mj} \end{bmatrix}_{m \times m} \quad (6)$$

$$B_j = \text{diag}[\delta_{1j}, \delta_{2j}, \dots, \delta_{mj}]_{m \times m} \quad (7)$$

$$C_j = \text{diag}[\eta_{1j}, \eta_{2j}, \dots, \eta_{mj}]_{m \times m} \quad (8)$$

where $j = 1, 2, \dots, n$.

Also α_{ij} , β_{ij} , γ_{ij} , δ_{ij} and η_{ij} are defined as:

$$\begin{aligned} \alpha_{ij} &= 1/h^2 - a_{ij}/2h \\ \beta_{ij} &= c_{ij} - 2/h^2 - 2/k^2 \\ \gamma_{ij} &= 1/h^2 + a_{ij}/2h \\ \delta_{ij} &= 1/k^2 + b_{ij}/2k \\ \eta_{ij} &= 1/k^2 - b_{ij}/2k \end{aligned} \quad (9)$$

\bar{u}_j , $j = 1(1)n$ is unknown column vector defined as:

$$\bar{u}_j = [u_{1j}, u_{2j}, \dots, u_{mj}]^T \quad (10)$$

V_j , $j = 1(1)n$ is right hand side modified after necessary adjustment due to given boundary conditions.

$$V_j = [V_{1j}, V_{2j}, \dots, V_{mj}]^T \quad (11)$$

3. Method of Solution

The resulting structure of system of Equations (5) can be written as

$$A\bar{U} = V \quad (12)$$

Coefficient matrix A is a block tridiagonal system having n blocks, where each block is of the border $(m \times m)$. Let R_{ij} , $i = 1(1)m$, $j = 1(1)m$, $j = 1(1)n$ denotes i th row of j th block. Elimination algorithm is as follows:

Step 1

For $j = 1$

Do $R_{i,1} = R_{i,1} - (\alpha_{i,1}/\beta_{i-1,1}) * (R_{i-1,1})$ for $i = 2$ to m

Step 2

For $j = 2$

(a) $i = 1$

(b) (i) $k = i$
 $R_{ij} = R_{ij} - (\eta_{ij} / \beta_{kj-1}) * R_{kj-1}$

By this operation, first entry of first row of second block becomes zero and second entry of first row of second block becomes nonzero, say $\hat{\eta}_{ij}$,

(ii) For $k = k + 1$ upto m do
 $\bar{R}_{ij} = \bar{R}_{ij} - (\hat{\eta}_{ij} / \bar{\beta}_{kj-1}) * \bar{R}_{kj-1}$

where “-” refers to modified entries.

- (c) $i = i + 1$
 (i) $R_{ij} = R_{ij} - (\eta_{ij} / \bar{\beta}_{kj-1}) * \bar{R}_{kj-1}$
 (ii) for $k = i$ to m (just like 2(b))
 $\bar{R}_{ij} = \bar{R}_{ij} - (\hat{\eta}_{ij} / \bar{\beta}_{kj-1}) * \bar{R}_{kj-1}$
 (iii) $\bar{R}_{ij} = \bar{R}_{ij} - (\alpha_{ij} / \bar{\beta}_{i-1j}) * \bar{R}_{i-1j}$
 (d) Again go to step 2(c) upto $i = m$.

Step 3

- (i) Put $j = j + 1$ and repeat steps 2(a) to 2(d).
 (ii) Continue this process upto $j = n$.

Step 4

Now the system is reduced to upper-triangular form and by back-substitution process, we may obtain values of unknowns u_{ij} , $i = 1(1)m, j = 1(1)n$.

4. Storage System and Operation Counts

The subroutine has been built using the above algorithm keeping in view the minimum required storage. Storage system is such that it does not require zeros or units to be stored. Only two single-subscripted arrays A and U are required for storage, where A requires $(m + 1)*m*n$ words and U $m*n$ words, where m and n are number of interior points along X and Y directions respectively.

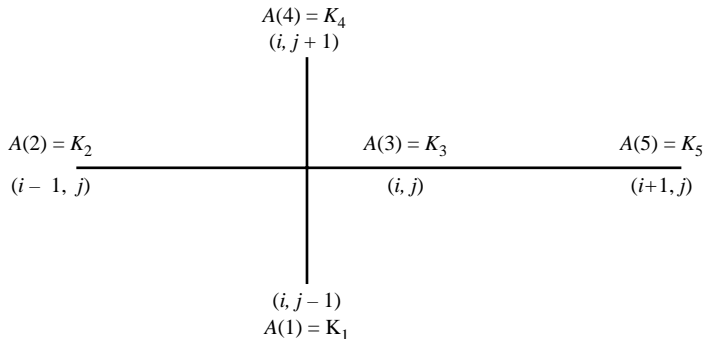


FIG. 2

In general six words per row (*i.e.*, per difference equation) are required to specify six coefficients in the following equation, which is obtained by applying finite difference approximation to Equation (1) at node (i, j)

$$K_1 u_{ij-1} + K_2 u_{i-1j} + K_3 u_{ij} + K_4 u_{ij+1} + K_5 u_{i+1j} = K_6$$

The coefficients K_1 to K_5 are stored in first five locations of A as shown in Fig. 2 and right hand side K_6 in $A(6)$ for node $i = 1, j = 1$. Next K_1, K_2, \dots, K_6 for node $i = 2, j = 1$ are stored from $A(7)$ to $A(12)$. The six words starting with $6^*m^*(n-1) + 6^*(m-1) + 1$ gives K_1, K_2, \dots, K_6 for last node $i = m, j = n$. The remaining storage of A is used as scratch storage. Solution is finally obtained in array U . First m entries of U provides values of unknown variable at the nodes of first row, next m entries at the nodes of second row. Thus last m entries starting with $m^*(n+1) + 1$ give solution at the points of last row. Total storage required is $(m+2)^*m^*n$.

The algorithm requires approximately $4m^3n$ addition operations and $3m^3n + 10m^2n$ multiplicative operations, thus totalling $m^2n(7m+10)$ arithmetic operations.

4.1 Test Examples

The following examples are tested using the present algorithm.

Example 1:

The equation solved is of Poisson's type in cartesian coordinates

$$\frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial y^2} = -\omega \quad (13)$$

The structure is given by (5)

$$\text{where } \alpha_{ij} = \gamma_{ij} = \delta_{ij} = \eta_{ij} = 1 \\ \beta_{ij} = -4$$

$$(a) \quad \text{Here, } \omega = 2\exp(x+y) \quad (14) \\ \text{Region: } \{(x, y), 0 \leq x \leq 1, 0 \leq y \leq 1\}$$

Boundary conditions are

$$\Psi = -[\exp(x+y) + (x-y) / Re] \quad (15)$$

on x -axis, y -axis, $x = 1$ and $y = 1$.

Analytical solution for the Equation (13) is

$$\Psi = -[\exp(x+y) + (x-y) / Re] \quad (16)$$

$$(b) \quad \Psi = \log(x^2 + y^2) \quad (17) \\ \text{Region: } \{(x, y), 0 \leq x \leq 1, 0 \leq y \leq 1\}$$

Boundary conditions are

$$\Psi = -((x^2 + y^2) / 4)[\log(x^2 + y^2) - 2] \quad (18)$$

on x-axis, y-axis, $x = 1$ and $y = 1$.

Analytical solution is same as (18).

Example 2

Poisson's equation is cylindrical coordinates is

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial z^2} = f(r, z) \tag{19}$$

Structure is given again by (5)

where $\alpha_{ij} = 1 - h/2r_i, 1 \leq i \leq m$

$$\begin{aligned} \beta_{ij} &= -2[1 + (h/k)^2] \\ \gamma_{ij} &= 1 + h/2r_i, 1 \leq i \leq M \\ \delta_{ij} &= \eta_{ij} = (h/k)^2 \end{aligned}$$

Domain in which equation to be solved is

$$\{(r, z) : 0.2 \leq r \leq 1, 0 \leq z \leq \}$$

Right hand side is as follows

$$f(r, z) = -4r^2 \exp(-2z) \tag{20}$$

Boundary conditions are specified as

$$\begin{aligned} u(0.2, z) &= 0.96 e^{-2z} \\ u(1, z) &= 0 \\ u(r, 0) &= 1 - r^2 \\ u(r, 2) &= e^{-4}(1 - r^2) \end{aligned} \tag{21}$$

Exact solution for this equation is

$$u(r, z) = e^{-2z}(1 - r^2) \tag{22}$$

Example 3

Navier-Stokes equation in its stream function vorticity form is

$$\nabla^2 \omega - Re(u\omega_x + v\omega_y) = 0 \tag{23}$$

where

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}$$

Here ψ denotes stream function and ω denotes vorticity of the flow. To determine a flow, Equation (23) is generally solved coupled with continuity Equation (13). Equation (23) is a nonlinear elliptic partial differential equation, the degree of nonlinearity increases with the value of Re , which denotes Reynolds number.

After applying finite difference formulae to Equation (23) on any rectangular region, structure is provided by (5) where

$$\begin{aligned}\alpha_{ij} &= 1 + (Re/4) (\Psi_{i,j+1} - \Psi_{i,j-1}) \\ \beta_{ij} &= -4 \\ \gamma_{ij} &= 1 - (Re/4) (\Psi_{i,j+1} - \Psi_{i,j-1}) \\ \delta_{ij} &= 1 + (Re/4) (\Psi_{i+1,j} - \Psi_{i-1,j}) \\ \eta_{ij} &= 1 - (Re/4) (\Psi_{i+1,j} - \Psi_{i-1,j})\end{aligned}$$

To demonstrate the applicability of our algorithm on Equation (23), following example is considered:

$$\text{Region } D = \{(x, y) : 1 \leq x \leq 2, 1 \leq y \leq 2\}$$

Values of Ψ are assumed as predetermined and given by

$$\Psi = -((x^2 + y^2)/4[\log(x^2 + y^2) - 2]) \quad (24)$$

Boundary conditions are of Dirichlet type so that values of ω are regarded as being known in advance and specified as

$$\omega = \log(x^2 + y^2) \quad (25)$$

on all the boundaries

Analytical solution is same as given by Equation (25).

5. Numerical Results and Conclusion

Results for all the examples are computed taking 11×11 and 21×21 node points in the given domain.

Results of Example 1(a) for some arbitrarily chosen points are given in Tables 1 and 2 along with exact results for $Re = 100$ and $Re = 1000$ respectively. Results for Example 1(b) are presented in Table 3 along with exact results and results obtained by Sharma and Agarwal^[9]. Sharma and Agarwal^[9] have applied a very efficient direct method, viz. Hockney's method^[10] to solve the resulting system of equations. Method in detail is discussed in Ref. [9]. The results obtained by present algorithm compare well with those as is obvious from Table 3.

TABLE 1. Results of Example 1(a) at some randomly chosen node points for $Re = 100$.

Node points 11×11			Node points 21×21		
Points	Computed results	Exact results	Points	Computed results	Exact results
(0.1, 0.1)	-1.22144	-1.2214028	(0.1, 0.05)	-1.16234	-1.1623342
(0.3, 0.2)	-1.64988	-1.6497213	(0.9, 0.25)	-3.16472	-3.1646929
(0.5, 0.5)	-2.71862	-2.7182818	(0.5, 0.5)	-2.71837	-2.7182818
(0.8, 0.7)	-4.48295	-4.4826891	(0.25, 0.7)	-2.58126	-2.5812097
(0.9, 0.9)	-6.04975	-6.0496475	(0.9, 0.95)	-6.35933	-6.3593195

TABLE 2. Results of Example 1(a) at some randomly chosen node points for $Re = 1000$.

Node points 11×11			Node points 21×21		
Points	Computed results	Exact results	Points	Computed results	Exact results
(0.1, 0.1)	- 1.22144	- 1.2214028	(0.1, 0.05)	- 1.16189	- 1.1618842
(0.3, 0.2)	- 1.64898	- 1.6488213	(0.9, 0.25)	- 3.15887	- 3.1588429
(0.5, 0.5)	- 2.71862	- 2.7182818	(0.5, 0.5)	- 2.71837	- 2.7182818
(0.8, 0.7)	- 4.48205	- 4.4817891	(0.25, 0.7)	- 2.58531	- 2.5852597
(0.9, 0.9)	- 6.04975	- 6.0496475	(0.9, 0.95)	- 6.35978	- 6.3597695

TABLE 3. Results of Example 1(b) at some randomly chosen node points.

Node points 11×11				Node points 21×21			
Points	Exact results	Computed results	Results of Sharma & Agarwal ^[9]	Points	Exact results	Computed results	Results of Sharma & Agarwal ^[9]
(0.1, 0.1)	0.0295601	0.0297723	0.02977231	(0.05, 0.1)	0.0199438	0.0199810	0.01998104
(0.2, 0.3)	0.1313071	0.131478	0.13147763	(0.25, 0.9)	0.4660006	0.466003	0.46600230
(0.5, 0.5)	0.3366433	0.336791	0.33679135	(0.5, 0.5)	0.3366433	0.336680	0.33668001
(0.7, 0.8)	0.5304735	0.530537	0.53053658	(0.7, 0.25)	0.3581998	0.358210	0.35821047
(0.9, 0.9)	0.6146174	0.614634	0.61463382	(0.95, 0.9)	0.625938319	0.625941	0.62594041

Example 2 is an illustration of Poisson’s equation in cylindrical coordinates and is considered in detail by Mittal and Gahlaut^[11]. They have derived some higher order finite difference schemes and compared their solutions with exact solutions. Table 4 provides a comparative study of our results and one of the higher order schemes of Mittal and Gahlaut^[11] along with exact results. Our results are in good agreement with both of these.

TABLE 4. Results of Example (2) at some randomly chosen node points.

Node points 11×11			Node points 21×21			
Points	Exact results	Computed results	Points	Exact results	Computed results	Results of Mittal & Gahlaut ^[11]
(0.36, 0.4)	0.39109593	0.391845	(0.36, 0.4)	0.39109593	0.391288	0.39128716
(0.52, 0.8)	0.14730370	0.147891	(0.52, 0.8)	0.14730370	0.147452	0.14745197
(0.76, 0.8)	0.08528108	0.0856590	(0.76, 0.8)	0.08528108	0.0853767	0.08537642
(0.68, 1.2)	0.04876997	0.0490204	(0.68, 1.2)	0.04876997	0.0488329	0.048832753
(0.84, 1.6)	0.01200039	0.0120594	(0.84, 1.6)	0.01200039	0.0120152	0.01201513

Example 3 gives an illustration to solve Navier-Stokes equation in cartesian coordinates. This example was first considered by Richards and Crane^[12] and later on solved by a fast direct solver BLTRI by Mittal and Sharma^[13]. In Ref [13], a measure to check

the overall accuracy of the algorithm at all points is given. This is called Maximum Modulus Error (*MME*) and is defined as

$$MME = \max_{i,j} |u_{ij} - u(x_i, y_j)|$$

TABLE 5. Results of Example 3 at some randomly chosen node points $Re = 100$.

Node points 11×11			Node points 21×21		
Points	Exact results	Computed results	Points	Exact results	Computed results
(1.1, 1.1)	0.88376754	0.883817	(1.1, 1.1)	0.83832919	0.838339
(1.2, 1.3)	1.141033	1.14134	(1.25, 1.9)	1.6433561	1.64339
(1.5, 1.5)	1.5040775	1.50446	(1.5, 1.5)	1.5040775	1.50417
(1.7, 1.8)	1.8131947	1.81329	(1.7, 1.25)	1.4934657	1.49352
(1.9, 1.9)	1.976855	1.97686	(1.95, 1.9)	2.0031678	2.00317

TABLE 6. Results of Example 3 at some randomly chosen node points $Re = 1000$.

Node points 11×11			Node points 21×21		
Points	Exact results	Computed results	Points	Exact results	Computed results
(1.1, 1.1)	0.883252	0.883252	(1.05, 1.1)	0.83832919	0.838302
(1.2, 1.3)	1.141033	1.14162	(1.25, 1.9)	1.6433561	1.64340
(1.5, 1.5)	1.5040775	1.50444	(1.5, 1.5)	1.5040775	1.50419
(1.7, 1.8)	1.8131947	1.81317	(1.7, 1.25)	1.4934657	1.49354
(1.9, 1.9)	1.976855	1.97670	(1.95, 1.9)	2.0031678	2.00317

where u_{ij} is numerically determined value, $u(x_i, y_j)$ corresponding analytical value and maximum is taken over all the internal node-points. In Ref [13], *MME* for Example 3 is given as 0.000402 and 0.00579 for $Re = 100$ when mesh is 11×11 and 21×21 respectively. Similarly for $Re = 1000$, values of *MME* are 0.000705 and 0.00162 for 11×11 and 21×21 mesh respectively. We have computed *MME* for all the three examples and results are reported in Table 7. For Example 3, *MME* values obtained by present algorithm is better than those of Mittal and Sharma^[13]. Another drawback in the scheme of Mittal and Sharma^[13] was the introduction of roundoff errors with increase in number of node-points and that is the reason for the deterioration of results for 21×21 mesh in comparison to 11×11 mesh. The present algorithm is along the established norms, since here by increasing node-points, results are significantly improved as is clear from Tables 1 to 6 and *MME* is also decreased as shown by Table 7.

TABLE 7. Maximum modulus error for Examples 1 to 3.

Examples	Re	Node points 11×11	Node points 21×21
1(a)	100	0.000353813	0.0000863075
	1000	0.000353813	0.0000860691
1(b)	–	0.00021219	0.00005351
2	–	0.00087428	0.00022417
3	100	0.000401855	0.000103116
	1000	0.000704169	0.000165939

Conclusively, the present algorithm gives accurate results to the problems after applying finite difference schemes to elliptic partial differential equations and along with it is also economical in computer storage.

Acknowledgement

The authors gratefully acknowledge the computational and other facilities provided by Wildlife Institute of India, Dehra Dun, in carrying out the present work.

References

- [1] **Young, D.M.**, *Iterative Solution of Large Linear System*, Academic Press, New York, pp. 63-105 (1971).
- [2] **Fox, L.**, *Numerical Solution of Ordinary and Partial Differential Equations*, Pergamon Press, pp. 276-283 (1962).
- [3] **Gupta, M.M.** and **Manohar, R.P.**, On the use of central difference scheme for Navier-Stokes equation, *Int. J. Num. Meth. Engg.*, **15**: 557-573 (1980).
- [4] **Linger, W.**, A quasi-direct fast Poisson solver for general regions, *J. Comput. Appl. Math.*, **28**: 25-45 (1989).
- [5] **Elsner, L.** and **Mehrmann, V.**, Convergence of block iterative methods for linear system arising in the numerical solution of Euler equation, *Numer. Math.*, **59**: 541-559 (1991).
- [6] **Duff, I., Erisman, A.** and **Reid, J.**, *Direct Methods for Sparse Matrices*, Oxford University Press, pp. 1-56 (1987).
- [7] **Erisman, A.** and **Reid, J.**, *Direct Methods for Sparse Matrices*, Oxford University Press, Oxford, pp. 43-76 (1981).
- [8] **Jennings, A.**, *Matrix Computation for Engineers and Scientists*, John Wiley & Sons, pp. 145-168 (1977).
- [9] **Sharma, P.K.** and **Agarwal, M.K.**, Fast finite difference direct solver for Poisson's equation, *Indian J. Phy. Nat. Sci.*, **10**, Sec. B: 22-28 (1989).
- [10] **Hockney, R.W.**, A fast direct solver of Poisson's equation using Fourier analysis, *J. Assoc. Comput. Mach.*, **12**: 95-113 (1965).
- [11] **Mittal, R.C.** and **Gahlaut, S.**, High order finite difference scheme to solve Poisson's equation in cylindrical symmetry, *Communication in Applied Numerical Methods*, **3**: 457-461 (1987).
- [12] **Richards, C.W.** and **Crane, C.M.**, The accuracy of finite difference scheme for the solution of the Navier-Stokes equations, *Appl. Math. Modelling*, **3**: 205-211 (1979).
- [13] **Mittal, R.C.** and **Sharma, P.K.**, A numerical treatment of two dimensional Navier-Stokes equations, *Math. Phy. Sci.*, **19**: 169-177 (1985).

طريقة اقتصادية لحل المعادلة الإهليلجية للمناطق مستطيلة الشكل

بي. كيه. شارما* و مانوج كومار أجاروال**
 * قسم الرياضيات ، الكلية التقنية الهندية ، دهرادون - الهند
 ** معهد الحياة الفطرية الهندية ، دهرادون - الهند

المستخلص . تم عرض خوارزمية تقوم بحل نظام المعادلات الخطية الآتية المتولد من تطبيق طريقة الفروق المحدودة على المعادلة الإهليلجية للمناطق مستطيلة الشكل . وتم ترتيب برنامج الحاسب المستنبط ، بحيث يشغل أقل ما يمكن من السعة التخزينية ، ومن وقت التنفيذ . ويجري اختبار هذا البرنامج بصورة مكثفة على ثلاثة أمثلة لها حلول تامة الدقة معلومة سلفاً . وتقارن الحلول الناتجة (هنا) مع الحلول تامة الدقة ، وكذلك مع النتائج التي حصل عليها بعض الباحثين من قبل . وتتفق النتائج الحالية جيداً مع كافة النتائج الأخرى .