

The numerical solution of an elliptic P.D.E. with periodic boundary conditions in a rectangular region by the spectral resolution method

D. J. Evans and S. O. Okolie

ABSTRACT

In this paper a new block matrix factorisation strategy is considered utilising the spectral resolution method for the solution of an elliptic partial differential equation with periodic boundary conditions in a rectangle.

1. INTRODUCTION

Consider the elliptic (Helmholtz's) equation in two space dimensions given by,

$$\nabla^2 U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 2\tau U(x,y) + q(x,y), \quad (1.1)$$

in a rectangular region,

$$R : \begin{cases} 0 \leq x \leq a \\ 0 \leq y \leq b \end{cases} \quad (1.2)$$

enclosed by the boundary region ∂R , and with the periodic boundary conditions in both the x- and y-directions given by,

$$\begin{aligned} U(0,y) &= U(a,y), \\ U(x,0) &= U(x,b), \end{aligned} \quad (1.3)$$

where $q(x,y)$ is a known function in x and y , and τ is any positive constant.

We define the mesh spacings $\Delta x = a/m$ and $\Delta y = b/n$ (m, n are integers) and then super-impose mesh-points $(x_i, y_j) = (i\Delta x, j\Delta y)$ over the interior region R_h and the discrete boundary region, ∂R_h where

$$R_h = \{(x_i, y_j) \mid 1 \leq i \leq m-1, 1 \leq j \leq n-1\},$$

and

$$\partial R_h = \partial R \cap \{(x_i, y_j) \mid 0 \leq i \leq m, 0 \leq j \leq n\}.$$

Further by using the notation $U_{i,j} = U(x_i, y_j)$ and then applying the five point finite difference approximation, i.e.,

$$\begin{aligned} \nabla^2 U_{i,j} &= \frac{1}{(\Delta x)^2} (U_{i-1,j} - 2U_{i,j} + U_{i+1,j}) \\ &+ \frac{1}{(\Delta y)^2} (U_{i,j-1} - 2U_{i,j} + U_{i,j+1}), \end{aligned} \quad (1.4)$$

the given elliptic equation (1.1) at the point (i,j) becomes,

$$U_{i-1,j} + \sigma U_{i,j-1} - 2(1+\sigma+\tau)U_{i,j} + \sigma U_{i,j+1} + U_{i+1,j} = d_{i,j}, \quad (1.5)$$

where

$$\sigma = \left(\frac{\Delta x}{\Delta y}\right)^2 \text{ and } d_{i,j} = (\Delta x)^2 q_{i,j}.$$

We shall here adopt a column-wise ordering of the mesh points as illustrated in figure 1.

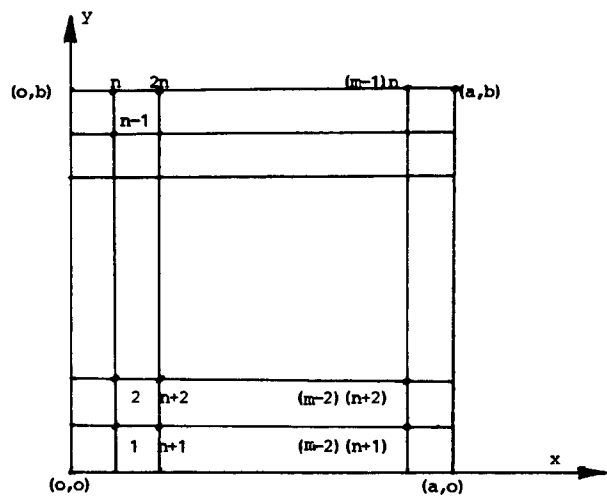


Fig. 1.

Hence, by applying equation (1.5) over the enclosed rectangular region together with the periodic boundary conditions (1.3), we obtain the block matrix system,

$$A\underline{u} = \underline{d}, \quad (1.6)$$

where

$$A = A[I, B, I] = \begin{bmatrix} B & I & & & I \\ I & B & I & & O \\ & & & & \\ & & & & \\ I & & & & B \end{bmatrix}_{(m \times m)} \quad (1.7)$$

\underline{u} the approximate finite difference solution of (1.1) is given by

D. J. Evans and S. O. Okolie, Department of Computer Studies, Loughborough University of Technology, Loughborough, Leicestershire, U.K.

$$\underline{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \quad u_i = \begin{bmatrix} u_{i,1} \\ u_{i,2} \\ \vdots \\ u_{i,n} \end{bmatrix}, \quad i=1,2,\dots,m,$$

and the vector \underline{d} is similarly partitioned as \underline{u} .
The matrix B is an $(n \times n)$ periodic tridiagonal matrix of the form,

$$B = B[\sigma, -2(1+\sigma+\tau), \sigma] = \begin{bmatrix} -2(1+\sigma+\tau) & \sigma & & & \sigma \\ \sigma & -2(1+\sigma+\tau) & \sigma & & 0 \\ & 0 & & & \sigma \\ & & & & \sigma \\ \sigma & & & & -2(1+\sigma+\tau) \end{bmatrix}, \quad (1.8)$$

and I is the $(n \times n)$ identity matrix.

2. THE CYCLIC FACTORISATION METHOD AND ITS VARIANTS

In Evans and Okolie [1] the general cyclic factorisation method was introduced for the solution of the general periodic tridiagonal matrix equation and here we now introduce various simplified variants of that general algorithm.

Symmetric case

Consider the matrix equation,

$$C\underline{u} = \underline{d}, \quad (2.1)$$

where C is a constant term, symmetric cyclic tridiagonal matrix of the form,

$$C = \begin{bmatrix} b & a & & & a \\ a & b & a & & 0 \\ & & & & a \\ & 0 & & & a \\ a & & & & b \end{bmatrix} \quad (2.2)$$

We seek to obtain algorithms that take advantage of the special structure of the coefficient matrix (2.2). First, we consider simplified variants of the generalised cyclic factorisation method of Evans and Okolie [1].

Following the cyclic factorisation method, the matrix C can be decomposed into the product of \bar{P} and \bar{Q} such that

$$C = \bar{P}\bar{Q}, \quad (2.3)$$

where

$$\bar{P} = \begin{bmatrix} 1 & & & & \ell \\ \ell & 1 & & & 0 \\ & & & & 0 \\ & 0 & & & \ell \\ & & & & 1 \end{bmatrix} \quad \text{and} \quad \bar{Q} = \begin{bmatrix} v & a & & & \\ v & a & & & 0 \\ & & & & a \\ & 0 & & & \\ a & & & & v \end{bmatrix}.$$

From (2.3), the elements ℓ, v are related in the following form,

$$\begin{aligned} \ell v &= a, \\ \text{and} \\ v + \ell a &= b. \end{aligned} \quad (2.4)$$

Now either ℓ or v can be expressed as an infinite periodic continued fraction. We consider v , which is then expressed in the form,

$$v = \frac{a^2}{b - \frac{a^2}{b - \frac{a^2}{b - \frac{a^2}{b - \dots}}}} \quad (2.5a)$$

with unit cycle length in the periodicity of the continued fraction.

Now the infinite continued fraction (2.5a) is generated by the linear fractional transformation,

$$\tau^{(1)}(\omega) = \frac{a^2}{b - \omega}, \quad (2.5b)$$

whose fixed points ω_1, ω_2 are given by the roots of the quadratic equation,

$$\omega = \frac{a^2}{b - \omega},$$

$$\text{or } \omega^2 - b\omega + a^2 = 0.$$

Hence the value of v , defined as $\max(\omega_1, \omega_2)$ becomes,

$$\begin{aligned} v &= (b + \sqrt{b^2 - 4a^2})/2, \quad b \geq 2a, \\ \text{and} \\ \ell &= \frac{a}{v} = (b - \sqrt{b^2 - 4a^2})/2a, \quad b \geq 2a. \end{aligned} \quad (2.6)$$

It follows immediately from (2.6) that if $b \geq 2a$, then $\ell \leq 1$.

The values of v and ℓ could simply have been obtained from (2.4) directly without the need to express any of them as an infinite continued fraction, but this has been done to maintain the uniformity of approach with the earlier strategy in the generalised factorisation.

The algorithm is summarised below.

$$\begin{aligned} \text{Step 1. Compute : } \ell &= (b - \sqrt{b^2 - 4a^2})/2a, \quad b \geq 2a, \quad (2.7) \\ \rho &= \ell/a. \end{aligned}$$

Step 2. Compute the following pre-computed coefficients,

$$\begin{aligned} \phi_i &= \ell^i, \quad \text{if } i \text{ is odd} \\ &= -\ell^i, \quad \text{if } i \text{ is even} \\ \gamma_i &= \phi_{n-i+1}. \end{aligned} \quad i = 1, 2, \dots, n. \quad (2.8)$$

Step 3. Calculate the following :-

$$(a) \quad d'_0 = 0, d'_i = d_i - \ell d'_{i-1}, \quad i = 1, 2, \dots, n \quad (2.9a)$$

$$(b) \quad \left. \begin{aligned} y_n &= d'_n / (1 + \phi_n), \\ y_i &= d'_i - \phi_i y_n, \end{aligned} \right\} \quad i = 2, 3, \dots, n-1, \quad (2.9b)$$

$$(c) \quad \left. \begin{aligned} g_n &= \rho y_n, \\ g_i &= \rho y_i - \ell g'_{i+1}, \end{aligned} \right\} \quad i = n-1, n-2, \dots, 1, \quad (2.9c)$$

and

$$(d) \quad \left. \begin{aligned} u_1 &= g'_1 / (1 + \gamma_1), \\ u_i &= g'_i - \gamma_i u_1, \end{aligned} \right\} \quad i = 2, 3, \dots, n \quad (2.9d)$$

where $\underline{u} = (u_1, u_2, \dots, u_n)^T$ is the required solution vector.

Steps 1 and 2 are computed only once during the first solution; thereafter for subsequent solutions only step 3

need be repeated. Thus, the algorithm requires $5n$ multiplications and $4n$ additions, in addition to n precomputed coefficients.

The stability of this algorithm is always guaranteed whenever $|\ell| < 1$, (since ℓ is used as a multiplier in an elimination process) which is easily shown to be satisfied (using (2.7)) whenever the coefficient matrix is strictly diagonally dominant, i.e. $|b| > |2a|$.

If $a = -1$, then $\ell = (-b + \sqrt{b^2 - 4})/2$, $b > 2$, $\rho = -\ell$ and hence (2.9c) becomes,

$$g'_n = -\ell y_n,$$

and

$$g'_i = -\ell(y_i + g'_{i+1}), \quad i = n-1, n-2, \dots, 1.$$

which results in a saving of n extra multiplications to give the required arithmetic operation count as $4n$ multiplications and $4n$ additions for the solution of (2.1) when the coefficient matrix C is of the form,

$$C = \begin{bmatrix} b & -1 & & & -1 \\ -1 & b & -1 & & 0 \\ & & & & \\ 0 & & & & -1 \\ -1 & & & & b \end{bmatrix}.$$

Similarly, if $a = 1$, then

$$\ell = (b - \sqrt{b^2 - 4})/2, \quad b > 2, \quad \ell = \rho.$$

and equation (2.9c) becomes,

$$g'_n = \ell y_n,$$

and

$$g'_i = \ell(y_i - g'_{i+1}), \quad i = n-1, n-2, \dots, 1.$$

These simple factorisation techniques resulting in fast algorithmic procedures can also be readily applied to the cases when C is unsymmetric as well as Toeplitz in form (Okolie [9]).

3. THE CYCLIC BLOCK FACTORISATION METHOD

Now an analogous block equivalent form of the generalised factorisation algorithm can be derived, provided the norm of the matrix B is greater than 2 (see (2.6) for the equivalent condition for the point case).

Here we give briefly the block form of the generalised algorithm which requires only the replacement of the matrix scalar elements 'b' and 'a' by the submatrix B and the identity matrix I respectively in order to obtain a direct solution of the block matrix equation (1.6) by the following procedure :

By defining a submatrix N of order n , (cf. (2.7)), i.e.,

$$N = 0.5[B - (B^2 - 4I)^{1/2}], \quad \|B\|_{\infty} > 2, \quad (3.1)$$

where $W^{1/2}$ is defined as the square root of matrix W (Spath [2]), then the right-hand side vector \underline{d} is modified to be

$$\left. \begin{aligned} d'_1 &= \underline{d}_1 \\ d'_i &= \underline{d}_i - Nd'_{i-1}, \quad i = 2, 3, \dots, m. \end{aligned} \right\} \quad (3.2)$$

Next, we calculate the intermediate solution sub-vectors y_i ($i = 1, \dots, m$) as

$$\left. \begin{aligned} y_m &= (I + \theta N^m)^{-1} \underline{d}'_m, \\ \text{and} \\ y_i &= \underline{d}'_i - \theta N^i y_m, \quad i = m-1, m-2, \dots, 1, \end{aligned} \right\} \quad (3.3)$$

where $\theta = 1$ (i odd) and $\theta = -1$, (i even).

Further by defining the intermediate vectors,

$$\left. \begin{aligned} g_m &= Ny_m \\ \text{and} \\ g_i &= N(y_i - g_{i+1}), \quad i = m-1, m-2, \dots, 1 \end{aligned} \right\} \quad (3.4)$$

we then obtain the final solution vector \underline{u} from the expressions,

$$\left. \begin{aligned} u_1 &= (I + \theta N^m)^{-1} g_1 \\ \text{and} \\ u_i &= g_i - \theta N^{m-i+1} u_1, \quad i = 2, 3, \dots, m. \end{aligned} \right\} \quad (3.5)$$

This block factorisation method (3.1) - (3.5) gives the desired direct solution of equation (1.6) provided that the matrix $(I + \theta N^m)$ is nonsingular. It is also necessary for stability considerations, to have the norm of the matrix N (since N is used as a multiplier in an elimination process) to be less than unity.

The determination of the submatrix N in the form given in (3.1) involves the evaluation of the square root of a positive definite matrix $(B^2 - 4I)$. Various iterative methods, based on the Raphson-Newton scheme or its variants are known to exist for the determination of the square root of such matrices. References for this include Laasonen [3], Spath [2], Babuska et al [4] and Schofield [5]. Also the algorithm involves the calculation of the multiple powers of the matrix N which can be achieved by repeated nested multiplication. However, both the iterative method for the evaluation of the square root of a matrix and the repeated matrix multiplication required to obtain the higher order powers of the matrix N would lead to a grossly inefficient method requiring excessive computing effort and storage. Thus, in the algorithmic solution these computational difficulties must be resolved before the algorithm can become competitive with alternative methods such as the spectral resolution method introduced in the next section.

4. SPECTRAL RESOLUTION METHOD

We consider the matrix equation,

$$A\underline{u} = \underline{d}, \quad (4.1)$$

where A is the $(m \times m)$ block matrix of the form,

$$A = \begin{bmatrix} B & C & & & C \\ C & B & C & & 0 \\ & & & & \\ 0 & & & & C \\ C & & & & C & B \end{bmatrix}_{(m \times m)}, \quad (4.2)$$

and the submatrices, B and C are $(n \times n)$ symmetric real matrices.

The vectors \underline{u} and \underline{d} are written in partitioned form so as to conform with the structure of A and hence have the form,

$$\underline{u} = \begin{bmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \vdots \\ \underline{u}_m \end{bmatrix}, \quad \underline{d} = \begin{bmatrix} \underline{d}_1 \\ \underline{d}_2 \\ \vdots \\ \underline{d}_m \end{bmatrix}, \quad (4.3)$$

where,

$$\underline{u}_j = \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{n,j} \end{bmatrix} \quad \text{and} \quad \underline{d}_j = \begin{bmatrix} d_{1,j} \\ d_{2,j} \\ \vdots \\ d_{n,j} \end{bmatrix}, \quad j = 1, 2, \dots, m. \quad (4.4)$$

We assume that B and C commute, i.e., $BC = CB$, and have a common basis of eigenvectors.

Then, by the well known theorem of Frobenius (see, for example, Varga [6]) there exists an orthogonal matrix Q (i.e. $Q^T = Q^{-1}$) whose columns are the set of eigenvectors of B and C such that,

$$\begin{cases} Q^T B Q = \Lambda \\ Q^T C Q = \Omega \end{cases} \quad (4.5)$$

where Λ and Ω are the diagonal matrices whose elements λ_i, ω_i ($i = 1, 2, \dots, n$) are the eigenvalues of B and C

respectively.

The system (4.1) together with (4.2) and (4.3), may be written as,

$$B \underline{u}_1 + C \underline{u}_2 + C \underline{u}_m = \underline{d}_1, \quad (4.6a)$$

$$C \underline{u}_{j-1} + B \underline{u}_j + C \underline{u}_{j+1} = \underline{d}_j, \quad j = 2, 3, \dots, m-1. \quad (4.6b)$$

and

$$C \underline{u}_1 + B \underline{u}_{m-1} + C \underline{u}_m = \underline{d}_m. \quad (4.6c)$$

By using equation (4.5) we have,

$$B = Q \Lambda Q^T$$

and

$$C = Q \Omega Q^T$$

which, when substituted into (4.6) give the following equations,

$$\Lambda \underline{\bar{u}}_1 + \Omega \underline{\bar{u}}_2 + \Omega \underline{\bar{u}}_m = \underline{\bar{d}}_1, \quad (4.7a)$$

$$\Omega \underline{\bar{u}}_{j-1} + \Lambda \underline{\bar{u}}_j + \Omega \underline{\bar{u}}_{j+1} = \underline{\bar{d}}_j, \quad j = 2, 3, \dots, m-1, \quad (4.7b)$$

$$\Omega \underline{\bar{u}}_1 + \Omega \underline{\bar{u}}_{m-1} + \Lambda \underline{\bar{u}}_m = \underline{\bar{d}}_m, \quad (4.7c)$$

where

$$\begin{cases} \underline{\bar{u}}_j = Q^T \underline{u}_j \\ \underline{\bar{d}}_j = Q^T \underline{d}_j \end{cases} \quad j = 1, 2, \dots, m, \quad (4.8)$$

and $\underline{\bar{u}}_j, \underline{\bar{d}}_j$ are labelled as in (4.4).

Further, we now resolve the equations in (4.7) by re-writing them, for $i = 1, 2, \dots, n$, as

$$\lambda_1 \bar{u}_{i,1} + \omega_1 \bar{u}_{i,2} + \omega_1 \bar{u}_{i,m} = \bar{d}_{i,1}, \quad (4.9a)$$

$$\omega_1 \bar{u}_{i,j-1} + \lambda_1 \bar{u}_{i,j} + \omega_1 \bar{u}_{i,j+1} = \bar{d}_{i,j}, \quad j = 2, 3, \dots, m-1. \quad (4.9b)$$

$$\omega_1 \bar{u}_{i,1} + \omega_1 \bar{u}_{i,m-1} + \lambda_1 \bar{u}_{i,m} = \bar{d}_{i,m}. \quad (4.9c)$$

Now, if we write

$$\Gamma_i = \begin{bmatrix} \lambda_1 & \omega_1 & & & \omega_1 \\ \omega_1 & \lambda_1 & \omega_1 & & 0 \\ & & \ddots & \ddots & \\ \omega_1 & 0 & & \omega_1 & \lambda_1 \end{bmatrix} \quad (m \times m), \quad (4.10)$$

$$\hat{\underline{u}}_i = \begin{bmatrix} \bar{u}_{i,1} \\ \bar{u}_{i,2} \\ \vdots \\ \bar{u}_{i,m} \end{bmatrix}, \quad \text{and} \quad \hat{\underline{d}}_i = \begin{bmatrix} \bar{d}_{i,1} \\ \bar{d}_{i,2} \\ \vdots \\ \bar{d}_{i,m} \end{bmatrix}, \quad (4.11)$$

then the equations in (4.9) are equivalent to the system

$$\Gamma_i \hat{\underline{u}}_i = \hat{\underline{d}}_i, \quad i = 1, 2, \dots, n. \quad (4.12)$$

Thus, the vector $\hat{\underline{u}}_i$ satisfies a symmetric tridiagonal matrix system of equations that has a constant diagonal, super- and sub-diagonal elements as in (4.10) which can be solved in an efficient manner by using the generalised factorisation algorithm.

After solving (4.12) it is then possible to solve for

$$\underline{u}_j = Q \underline{\bar{u}}_j, \quad j = 1, 2, \dots, m. \quad (4.13)$$

The above matrix decomposition algorithm is due to Buzbee et al [8]. If we regard the block vectors \underline{u} and \underline{d} as 2-dimensional arrays, then the above algorithm may be summarised as follows :

Step 1

Compute or determine the eigenvalues of matrices B and C and the eigenvectors of B . These eigensystems are often given by known analytical formulae for certain representations of B and C ; e.g. in the case of solving a Poisson equation in a rectangle, B is tridiagonal and C is diagonal.

Step 2

Compute the vectors,

$$\underline{\bar{d}}_j = Q^T \underline{d}_j, \quad j = 1, 2, \dots, m, \quad (4.14)$$

which is equivalent to multiplying each row of \underline{d} by Q^T .

Step 3

Next, we re-order the array $\underline{\bar{d}}$ by vertical lines instead of horizontal lines to generate the array $\hat{\underline{d}}$ and then solve the tridiagonal systems,

$$\Gamma_i \hat{\underline{u}}_i = \hat{\underline{d}}_i, \quad i = 1, 2, \dots, n. \quad (4.15)$$

Step 4

Finally, we re-order the array $\hat{\underline{u}}$ by horizontal lines instead of vertical lines to generate the array $\underline{\bar{u}}$ and then compute the solution vector,

$$\underline{u}_j = Q \underline{\bar{u}}_j, \quad j = 1, 2, \dots, m. \quad (4.16)$$

If we neglect the computation of the eigensystem (step 1)

then, the operation count for the spectral resolution method is given as shown in table 1. A reduction in the arithmetic operation count is possible if the fast Fourier transfer (Cooley and Tukey [7]) is used to perform steps 2 and 4.

TABLE 1. Summary of arithmetic operation count

Steps	Multiplications (X)	Additions (+)
2	n^2m	n^2m
3	$5nm$	$4nm$
4	n^2m	n^2m
TOTAL	$2n^2m + 5nm$	$2n^2m + 4nm$

REFERENCES

1. EVANS D. J. and OKOLIE S. O. : "A generalised sparse factorisation method for the solution of periodic tridiagonal systems", *Comp. & Maths. with Applic.*, Vol. 5 (1979) 211-216.
2. SPATH H. : "Algorithm 298 : Determination of the square root of a positive definite matrix [F1]", *Comm. A.C.M.*, Vol. 10 (1967) No. 3, p. 182.
3. LAASONEN P. : "On the iterative solution of the matrix equation $AX^2 - I = 0$ ", *Math. Tables Aids Comp.*, Vol. 12 (1958) 109-116.
4. BABUSKA I., PRAGER M. and VITASEK E. : *Numerical processes in differential equations*, John Wiley (1966).
5. SCHOFIELD D. F. : "A note on Lodwin orthogonalisation and the square root of a positive self-adjoint matrix", *Inter. J. of Quantum Chem.*, Vol. VII (1973) 561-568.
6. VARGA R. S. : *Matrix iterative analysis*, Prentice-Hall, Englewood Cliff (1962).
7. COOLEY J. W. and TUKEY J. W. : "Algorithm for machine calculation of complex Fourier series", *Math. Comp.*, Vol. 19 (1965) 297-301.
8. BUZBEE B. L., GOLUB G. H. and NIELSON C. W. : "On direct methods for solving Poisson equations", *SIAM J. Numer. Anal.*, Vol. 7 (1970) No. 4, 627-656.
9. OKOLIE S. O. : *The numerical solution of sparse matrix equations by fast methods and associated computational techniques*, Ph. D. Dissertation, University of Technology, Loughborough (1978).