SLAC-PUB-742
May 1970
(MISC)

ON DIRECT METHODS FOR SOLVING POISSON'S EQUATIONS*†

B. L. Buzbee

Los Alamos Scientific Laboratory
University of California, Los Alamos, New Mexico 87544

G. H. Golub

Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

and

C. W. Nielson

Los Alamos Scientific Laboratory
University of California, Los Alamos, New Mexico 87544

ABSTRACT

Some efficient and accurate direct methods are developed for solving certain elliptic partial difference equations over a rectangle with Dirichlet, Neumann, or periodic boundary conditions. Generalizations to high dimensions and to L-shaped regions are included.

(Submitted to the SIAM Journal on Numerical Analysis.)

---

*This work was performed under the auspices of the U. S. Atomic Energy Commission.

†This paper is a condensation of the Stanford Computer Science Reports CS 128 and CS 155.

1. Introduction

In many physical applications, one must solve an N x N system of linear algebraic equations,

$$M\underset{\sim}{x} = \underset{\sim}{y} \ , \tag{1.1}$$

where M arises from a finite-difference approximation to an elliptic partial differential equation. For this reason, the matrix M is sparse and the nonzero elements occur very regularly. As an example, let

$$M = \begin{bmatrix} I & F \\ F^T & I \end{bmatrix} \ , \tag{1.2}$$

and partition $\underset{\sim}{x}$ and $\underset{\sim}{y}$ to conform with M. If one can interchange both the rows and the columns of a matrix so that it has the form (M-I), the matrix is said to be two-cyclic.[2] Expanding Eq. (1.1), we have

$$\underset{\sim}{x}_1 + F\underset{\sim}{x}_2 = \underset{\sim}{y}_1 \ ,$$

$$F^T\underset{\sim}{x}_1 + \underset{\sim}{x}_2 = \underset{\sim}{y}_2 \ .$$

Multiplying the first equation by $-F^T$ and adding, we have

$$(I-F^TF)\underset{\sim}{x}_2 = \underset{\sim}{y}_2 - F^T\underset{\sim}{y}_1 \ . \tag{1.3}$$

By this simple device, we have reduced the number of equations. If $(I-F^TF)$ is also two-cyclic, we can again eliminate a number of the variables, and continue until the resulting matrix is no longer two-cyclic. In fact, F has block structure in many applications; and one has the

freedom to specify the number of blocks in it. In these cases, a proper choice of the number of blocks may enable the reduction process to continue until the final $(I-F^TF)$ has block dimension one; the reduction methods to be developed impose such a condition.

Direct methods for solving (1.1) are attractive since in theory they yield the exact solution to the difference equation, whereas commonly used methods seek to approximate the solution by iterative procedures.[12] Based on a suggestion of one of the authors, Hockney[8] has devised an efficient direct method which uses the reduction process. Also, Buneman[2] recently developed an efficient direct method for solving the reduced system of equations. Since these methods offer considerable economy over older techniques,[5] the purpose of this paper is to present a unified mathematical development and generalization of them. Additional generalizations are given by George.[6]

In Section 2 we develop the method of matrix decomposition or discrete separation of variables. In Section 3 we develop the block cyclic reduction process and techniques for solving the reduced systems. In Sections 4, 5, and 6, we apply the results of 2 and 3 to Poisson's equation on a rectangle with Dirichlet, Neumann, and periodic boundary conditions, respectively. Section 7 extends the results of 4, 5, and 6 to higher dimensions; Section 8 extends 2 and 3 to other applications; and Section 9 extends 2 and 3 to "L-shaped" regions. In Section 10, we show that straight-forward applications of the results of 3 can result in severe round-off error in many applications of interest. In Section 11 we develop the Buneman algorithms which are mathematically equivalent to the reduction process of 3, but are not subject to severe round-off.

In Section 11 we apply the Buneman algorithm to Poisson's equation with Dirichlet, Neumann, and periodic boundaries. Finally, in Section 12 we show the stability of the Buneman algorithms.

2. Matrix Decomposition

Consider the system of equations

$$M\underset{\sim}{x} = \underset{\sim}{y} , \tag{2.1}$$

where M is an N x N real symmetric matrix of block tridiagonal form,

$$M = \begin{bmatrix} A & T & & & \\ T & A & . & & \\ & . & . & . & \\ & & . & . & T \\ & & & T & A \end{bmatrix} . \tag{2.2}$$

The matrices A and T are p x p symmetric matrices, and we assume that

$$AT = TA .$$

Such a situation arises for those problems that can be handled by the classical separation-of-variables technique. Indeed, the methods we discuss amount to an efficient computer implementation of the idea of separation of variables carried out on a discretized model of the elliptic differential equation. Since A and T commute and are symmetric, it is well known[1] that there exists an orthogonal matrix Q such that

$$Q^T A Q = \Lambda , \quad Q^T T Q = \Omega , \tag{2.3}$$

and Λ and Ω are real diagonal matrices. The matrix Q is the set of eigenvectors of A and T, and Λ and Ω are the diagonal matrices of eigenvalues of A and T, respectively.

To conform with the matrix M, we write the vectors $\underset{\sim}{x}$ and $\underset{\sim}{y}$ in partitioned form,

$$\underset{\sim}{x} = \begin{bmatrix} \underset{\sim}{x}_1 \\ \hline \underset{\sim}{x}_2 \\ \hline \vdots \\ \hline \underset{\sim}{x}_q \end{bmatrix} \qquad \underset{\sim}{y} = \begin{bmatrix} \underset{\sim}{y}_1 \\ \hline \underset{\sim}{y}_2 \\ \hline \vdots \\ \hline \underset{\sim}{y}_q \end{bmatrix}$$

Furthermore, it is quite natural to write

$$\underset{\sim}{x}_j = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{pj} \end{bmatrix} \qquad \underset{\sim}{y}_j = \begin{bmatrix} y_{1j} \\ y_{2j} \\ \vdots \\ y_{pj} \end{bmatrix} \tag{2.4}$$

System (2.2) may be written

$$A\underset{\sim}{x}_1 + T\underset{\sim}{x}_2 = \underset{\sim}{y}_1 , \tag{2.5a}$$

$$T\underset{\sim}{x}_{j-1} + A\underset{\sim}{x}_j + T\underset{\sim}{x}_{j+1} = \underset{\sim}{y}_j, \; j = 2,3,\ldots,q-1 , \tag{2.5b}$$

$$T\underset{\sim}{x}_{q-1} + A\underset{\sim}{x}_q = \underset{\sim}{y}_q . \tag{2.5c}$$

Using Eq. (2.3), this becomes

$$\left.\begin{aligned} &\Lambda\underset{\sim}{\bar{x}}_1 + \Omega\underset{\sim}{\bar{x}}_2 = \underset{\sim}{\bar{y}}_1 , \\ &\Omega\underset{\sim}{\bar{x}}_{j-1} + \Lambda\underset{\sim}{\bar{x}}_j + \Omega\underset{\sim}{\bar{x}}_{j+1} = \underset{\sim}{\bar{y}}_j , \; (j = 2,3,\ldots,q-1) \\ &\Omega\underset{\sim}{\bar{x}}_{q-1} + \Lambda\underset{\sim}{\bar{x}}_q = \underset{\sim}{\bar{y}}_q . \end{aligned}\right\} , \tag{2.6}$$

where

$$\underset{\sim}{\bar{x}}_j = Q^T \underset{\sim}{x}_j \ , \quad \underset{\sim}{\bar{y}}_j = Q^T \underset{\sim}{y}_j \ , \quad j = 1,2,\ldots,q \ .$$

The components of $\underset{\sim}{\bar{x}}_j$ and $\underset{\sim}{\bar{y}}_j$ are labeled as in Eq. (2.4). Then Eq. (2.6) may be rewritten for $i = 1,2,\ldots,p$

$$\begin{aligned} &\lambda_i \bar{x}_{i1} + \omega_i \bar{x}_{i2} = \bar{y}_{i1} \ , \\ &\omega_i \bar{x}_{ij-1} + \lambda_i \bar{x}_{ij} + \omega_i \bar{x}_{ij+1} = \bar{y}_{ij}, \ (j = 2,\ldots,q-1) \ , \\ &\omega_i \bar{x}_{iq-1} + \lambda_i \bar{x}_{iq} = \bar{y}_{iq} \ . \end{aligned} \tag{2.7}$$

Now let us write

$$\Gamma_i = \begin{bmatrix} \lambda_i & \omega_i & & & \\ \omega_i & \lambda_i & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \omega_i \\ & & & \omega_i & \lambda_i \end{bmatrix}_{q \times q} \ ,$$

$$\underset{\sim}{\hat{x}}_i = \begin{bmatrix} \bar{x}_{i1} \\ \bar{x}_{i2} \\ \cdot \\ \cdot \\ \cdot \\ \bar{x}_{iq} \end{bmatrix} \qquad \underset{\sim}{\hat{y}}_i = \begin{bmatrix} \bar{y}_{i1} \\ \bar{y}_{i2} \\ \cdot \\ \cdot \\ \cdot \\ \bar{y}_{iq} \end{bmatrix} \ ,$$

so that Eq. (2.7) is equivalent to the system of equations,

$$\Gamma_i \hat{\underset{\sim}{x}}_i = \hat{\underset{\sim}{y}}_i \ . \tag{2.8}$$

Thus, the vector $\hat{\underset{\sim}{x}}_i$ satisfies a symmetric tridiagonal system of equations that has a constant diagonal element and a constant super- and sub-diagonal element. After Eq. (2.8) has been solved, it is possible to solve for $\underset{\sim}{x}_j = Q\bar{\underset{\sim}{x}}_j$. Thus the algorithm proceeds as follows.

1. Compute or determine the eigenvectors of A and the eigenvalues of A and T.

2. Compute $\underset{\sim}{y}_j = Q^T \underset{\sim}{y}_j$ $\qquad (j = 1,2,\ldots,q)$.

3. Solve $\Gamma_i \hat{\underset{\sim}{x}}_i = \hat{\underset{\sim}{y}}_i$ $\qquad (i = 1,2,\ldots,p)$.

4. Compute $\underset{\sim}{x}_j = Q\underset{\sim}{x}_j$ $\qquad (j = 1,2,\ldots,q)$.

Hockney[8] has carefully analyzed this algorithm for solving Poisson's equation in a square. He has taken advantage of the fact that in this case the matrix Q is known and that one can use the fast Fourier transform to perform steps 2 and 4. Shintani[11] has given methods for solving for the eigenvalues and eigenvectors in a number of special cases.

A and T need not commute. Assume that T is positive definite and symmetric. It is well known[1] that there is a matrix P such that

$$T = P\,P^T \ , \quad A = P\,\Delta\,P^T \ , \tag{2.9}$$

where Δ is the diagonal matrix of eigenvalues of $T^{-1}A$, and P^{-T} is the matrix of eigenvectors of $T^{-1}A$. Thus, using Eq. (2.9), we modify the algorithm as follows.

1. Compute or determine the eigenvalues and eigenvectors of $T^{-1}A$.

2. Compute $\bar{\underset{\sim}{y}}_j = P^{-1}\underset{\sim}{y}_j$

3. Solve $\Gamma_i \hat{\underset{\sim}{x}}_i = \hat{\underset{\sim}{y}}_i$ where

$$\Gamma_i = \begin{bmatrix} \delta_i & 1 & & & \\ 1 & \delta_i & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & 1 \\ & & & 1 & \delta_i \end{bmatrix} .$$

4. Compute $\underset{\sim}{x}_j = P^{-T}\underset{\sim}{\bar{x}}_j$.

Of course, one should avoid computing $T^{-1}A$ because this would destroy the sparseness of the matrices. Golub et al[7] has proposed an algorithm for solving $A\underset{\sim}{u} = \delta T\underset{\sim}{u}$ when A and T are sparse.

3. Block Cyclic Reduction

In Section 2, we gave a method for which one had to know the eigenvalues and eigenvectors of some matrix. We now give a more direct method for solving the system of Eq. (2.1).

We assume again that A and T are symmetric and that A and T commute. Furthermore, we assume that $q = m-1$ and

$$m = 2^{k+1} ,$$

where k is some positive integer. Let us rewrite Eq. (2.5b) as follows:

$$T\underset{\sim}{x}_{j-2} + A\underset{\sim}{x}_{j-1} + T\underset{\sim}{x}_{j} = \underset{\sim}{y}_{j-1} ,$$

$$T\underset{\sim}{x}_{j-1} + A\underset{\sim}{x}_{j} + T\underset{\sim}{x}_{j+1} = \underset{\sim}{y}_{j} ,$$

$$T\underset{\sim}{x}_{j} + A\underset{\sim}{x}_{j+1} + T\underset{\sim}{x}_{j+1} = \underset{\sim}{y}_{j+1} .$$

Multiplying the first and third equation by T, the second equation by -A, and adding, we have

$$T^2\underset{\sim}{x}_{j-2} + (2T^2-A^2)\underset{\sim}{x}_{j} + T^2\underset{\sim}{x}_{j+2} = T\underset{\sim}{y}_{j-1} - A\underset{\sim}{y}_{j} + T\underset{\sim}{y}_{j+1} .$$

Thus if j is even, the new system of equations involves $\underset{\sim}{x}_j$'s with even indices. Similar equations hold for $\underset{\sim}{x}_2$ and $\underset{\sim}{x}_{m-2}$. The process of reducing the equations in this fashion is known as cyclic reduction. Then Eq. (2.1) may be written as the following equivalent system:

$$\begin{bmatrix} (2T^2-A^2) & T^2 & & & \\ T^2 & (2T^2-A^2) & T^2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & T^2 \\ & & & T^2 & (2T^2-A^2) \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \\ \vdots \\ x_{m-2} \end{bmatrix}$$

$$= \begin{bmatrix} Ty_1 + Ty_3 - Ay_2 \\ Ty_3 + Ty_5 - Ay_4 \\ \vdots \\ Ty_{m-1} + Ty_{m-3} - Ay_{m-2} \end{bmatrix}. \qquad (3.1)$$

and

$$\begin{bmatrix} A & 0 & & & \\ 0 & A & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 0 \\ & & & 0 & A \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ \vdots \\ x_{m-1} \end{bmatrix} = \begin{bmatrix} y_1 - Tx_2 \\ y_3 - Tx_2 - Tx_4 \\ \vdots \\ y_m - Tx_{m-2} \end{bmatrix} \qquad (3.2)$$

Since $m = 2^{k+1}$, and the new system of Eq. (3.1) involves x_j's with even indices, the block dimension of the new system of equations is 2^k-1. Note that once Eq. (3.1) is solved, it is easy to solve for the x_j's with odd indices as evidenced by Eq. (3.2). We shall refer to the system of Eq. (3.2) as the eliminated equations.

Also, note that the algorithm of Section 2 may be applied to System (3.1). Since A and T commute, the matrix $(2T^2-A^2)$ has the same

set of eigenvectors as A and T. Also, if

$$\lambda(A) = \lambda_i \ , \quad \lambda(T) = \omega_i, \quad \text{for } i = 1,2,\ldots,m-1,$$

$$\lambda(2T^2-A^2) = 2\omega_i^2 - \lambda_i^2 \ .$$

Hockney[8] has advocated this procedure.

Since System (3.1) is block tridiagonal and of the form of Eq. (2.2), we can apply the reduction repeatedly until we have one block. However, as noted above, we can stop the process after any step and use the methods of Section 2 to solve the resulting equations.

To define the procedure recursively, let

$$A^{(o)} = A, \ T^{(o)} = T; \ \underset{\sim}{y}_j^{(o)} = \underset{\sim}{y}_j, \ (j = 1,2,\ldots,m-1).$$

Then for $r = 0,1,\ldots,k$,

$$\left.\begin{aligned} A^{(r+1)} &= 2(T^{(r)})^2 - (A^{(r)})^2, \\ T^{(r+1)} &= (T^{(r)})^2, \\ \underset{\sim}{y}_j^{(r+1)} &= T^{(r)} \, \underset{\sim}{y}_{j-2^r}^{(r)} + \underset{\sim}{y}_{j+2^r}^{(r)} - A^{(r)} \, \underset{\sim}{y}_j^{(r)} \ . \end{aligned}\right\} \tag{3.3}$$

At each stage, we have a new system of equations,

$$M^{(r)} \underset{\sim}{z}^{(r)} = \underset{\sim}{f}^{(r)},$$

to solve where

$$M^{(r)} = \begin{bmatrix} A^{(r)} & T^{(r)} & & & \\ T^{(r)} & A^{(r)} & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & T^{(r)} \\ & & & T^{(r)} & A^{(r)} \end{bmatrix}$$

$$\underset{\sim}{z}^{(r)} = \begin{bmatrix} \underset{\sim}{x}_{2^r} \\ \underset{\sim}{x}_{2^r+1} \\ \vdots \\ \underset{\sim}{x}_{j\cdot 2^r} \\ \vdots \end{bmatrix}, \qquad \underset{\sim}{f}^{(r)} = \begin{bmatrix} y_{2^r}^{(r)} \\ y_{2^r}^{(r)} \\ \vdots \\ y_{j\cdot 2^r} \\ \vdots \end{bmatrix}.$$

The eliminated equations are the solution of the block diagonal system

$$N^{(r)} \underset{\sim}{w}^{(r)} = \underset{\sim}{g}^{(r)}, \tag{3.5}$$

where

$$N^{(r)} = \begin{bmatrix} A^{(r-1)} & 0 & & & \\ 0 & A^{(r-1)} & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 0 \\ & & & 0 & A^{(r-1)} \end{bmatrix}.$$

$$\underset{\sim}{w}^{(r)} = \begin{bmatrix} \underset{\sim}{x}_{2^r-2^{r-1}} \\ \underset{\sim}{x}_{2^{r+1}-2^{r-1}} \\ \vdots \\ \underset{\sim}{x}_{j2^r-2^{r-1}} \\ \vdots \end{bmatrix},$$

$$\underset{\sim}{g}^{(r)} = \begin{bmatrix} y^{(r-1)}_{2^r-2^{r-1}} - T\underset{\sim}{x}^{(r-1)}_{2^r} \\ y^{(r-1)}_{2^{r+1}-2^{r-1}} - T\underset{\sim}{x}^{(r-1)}_{2^{r+1}} + \underset{\sim}{x}^{(r-1)}_{2^r} \\ \vdots \\ y^{(r-1)}_{j2^r-2^{r-1}} - T(\underset{\sim}{x}^{(r-1)}_{j2^r} + \underset{\sim}{x}_{(j-1)2^r} \\ \vdots \end{bmatrix}.$$

We can use the methods of Section 2 to solve the system $M^{(r)}\underset{\sim}{z}^{(r)} = \underset{\sim}{f}^{(r)}$, or we can proceed to compute $M^{(r+1)}$ and eliminate half of the unknowns. After k steps, we must solve the system of equations

$$A^{(k)}\ \underset{\sim}{x}_{2^k} = y^{(k)}_{2^k} \quad . \tag{3.6}$$

In either case, we must solve Eq. (3.5) to find the eliminated unknowns, just as in Eq. (3.2). This can be done by

1. direct solution,
2. eigenvalue-eigenvector factorization, or
3. polynomial factorization.

The direct solution is attractive when k is small. One can form the powers of A and T quite easily and solve the resulting equations by Gaussian elimination. Thus, if k = 1, and A and T are tridiagonal matrices, $A^{(1)}$ is a five-diagonal matrix, and for such band matrices it is easy to solve the resulting system of equations.

It is possible to compute the eigenvalue-eigenvector decomposition of $A^{(r)}$ and $T^{(r)}$. Since $A^{(o)} = Q\,\Lambda\,Q^T$ and $T^{(o)} = Q\,\Omega\,Q^T$, we may write

$$A^{(r)} = Q\,\Lambda^{(r)}Q^T \quad \text{and} \quad T^{(r)} = Q\,\Omega^{(r)}Q^T .$$

From Eq. (3.3), it follows that

$$\Lambda^{(r+1)} = 2\left(\Omega^{(r)}\right)^2 - \left(\Lambda^{(r)}\right)^2 ,$$

$$\Omega^{(r+1)} = \left(\Omega^{(r)}\right)^2 .$$

Thus the eigenvalues of $A^{(r)}$ and $T^{(r)}$ can be generated by the simple rule

$$\lambda_i^{(r+1)} = 2\left(\omega_i^{(r)}\right)^2 - \left(\lambda_i^{(r)}\right)^2 , \quad \lambda_i^{(o)} = \lambda_i ,$$

$$\omega_i^{(r+1)} = 2\left(\omega_i^{(r)}\right)^2 , \quad \omega_i^{(o)} = \omega_i ,$$

$$i = 1,2,\ldots,m-1.$$

Hence, the methods of Section 2 can easily be applied to solving the system $M^{(r)}\underset{\sim}{z}^{(r)} = \underset{\sim}{f}^{(r)}$ and $N^{(r)}\underset{\sim}{w}^{(r)} = \underset{\sim}{g}^{(r)}$. Hockney[9] has described this algorithm as the FACR(ℓ) algorithm where ℓ refers to the number of cyclic reductions performed.

From Eq. (3.1), we note that $A^{(1)}$ is a polynomial of degree 2 in A and T. By induction, it is easy to show that $A^{(r)}$ is a polynomial of degree 2^r in the matrices A and T, so that

$$A^{(r)} = \sum_{j=0}^{2^{r-1}} c_{2j}^{(r)} A^{2j} T^{2^r-2j} \equiv P_{2^r}(A,T).$$

We shall proceed to determine the linear factors of $P_{2^r}(A,T)$.

Let

$$p_{2^r}(a,t) = \sum_{j=0}^{2^{r-1}} c_{2j}^{(r)} a^{2j} t^{2^r-2j}, \quad c_{2^r}^{(r)} = -1 .$$

For $t \neq 0$, we make the substitution

$$a/t = -2 \cos \theta . \tag{3.7}$$

From Eq. (3.3), we note that

$$p_{2^{r+1}}(a,t) = 2t^{2^{r+1}} - \left(p_{2^r}(a,t)\right)^2 . \tag{3.8}$$

It is then easy to verify using Eqs. (3.7) and (3.8), that

$$p_{2^r}(a,t) = -2t^{2^r} \cos 2^r\theta ,$$

and, consequently,

$$p_{2^r}(a,t) = 0 \text{ when } a/2t = -\cos\left(\frac{2j-1}{2^{r+1}}\right)\pi \text{ for } j = 1,2,\ldots,2^r.$$

Thus, we may write

$$p_{2^r}(a,t) = -\prod_{j=1}^{2^r} \left[a + 2\,t \cos\left(\frac{2j-1}{2^{r+1}}\right)\pi \right] ,$$

and, hence,

$$A^{(r)} = -\prod_{j=1}^{2^r} \left(A + 2 \cos \theta_j^{(r)} T \right),$$

where $\theta_j^{(r)} = (2j-1)\pi/2^{r+1}$.

Let us write

$$G_j^{(k)} = A + 2 \cos \theta_j^{(k)}\, T .$$

Then, to solve Eq. (3.6), we set $\underset{\sim}{z}_1 = -\underset{\sim}{y}_{2^k}^{(k)}$ and repeatedly solve

$$G_j^{(k)}\, \underset{\sim}{z}_{j+1} = \underset{\sim}{z}_j \quad \text{for } j = 1,2,\ldots,2^k . \tag{3.9}$$

Thus,

$$\underset{\sim}{z}_{2^k+1} = \underset{\sim}{x}_{2^k}$$

If A and T are of band structure, it is simple to solve Eq. (3.9). To determine the solution to the eliminated Eq. (3.5), a similar algorithm may be used with

$$A^{(r)} = -\prod_{j=1}^{2^r} G_j^{(r)} \tag{3.10}$$

The factorization for $A^{(r)}$ may also be used to compute $\underset{\sim}{y}_j^{(r+1)}$ in Eq. (3.3). It is possible, however, to take advantage of the recursive nature of the polynomials $p_{2^r}(a,t)$. Let

$$p_s(a,t) = -2t^s \cos s\theta,$$

where, again, for $t \neq 0$, $a/t = -2 \cos \theta$.
Then a short manipulation shows that

$$p_s(a,t) = -ap_{s-1}(a,t) - t^2 p_{s-2}(a,t), \quad s \geq 2,$$

$$p_o(a,t) = -2, \quad p_1(a,t) = a .$$

Therefore, to compute $A^{(r)}\underset{\sim}{y}_j^{(r)}$ as in Eq. (3.3), we compute the following sequence:

$$\underset{\sim}{\eta}_o = 2\underset{\sim}{y}_j^{(r)} , \quad \underset{\sim}{\eta}_1 = A\underset{\sim}{y}_j^{(r)} ,$$

$$\underset{\sim}{\eta}_s = -A\underset{\sim}{\eta}_{s-1} - T^2\underset{\sim}{\eta}_{s-2} \quad \text{for } s = 2,3,\ldots,2^r .$$

Thus,

$$\underset{\sim}{\eta}_{2^r} = P_{2^r}(A,T)\underset{\sim}{y}_j^{(r)} \equiv A^{(r)}\underset{\sim}{y}_j^{(r)} .$$

We call this method the cyclic odd-even reduction and factorization (CORF) algorithm. In Section 10 we will show that the numerical calculation of $\underset{\sim}{y}_j^{(r)}$ in Eq. (3.3) is subject to severe rounding errors in many cases of interest. Consequently, numerical application of the results of this section must be accompanied by close attention to the results of Section 10. In fact, from a computational viewpoint, the CORF algorithm as developed here is virtually useless; however, the theoretical results

of this section are necessary for the development of the stable, Buneman variants of CORF.

4. Poisson's Equation with Dirichlet Boundary Conditions

It is instructive to apply the results of Section 3 to the solution of the finite-difference approximation to Poisson's equation on a rectangle, R, with specified boundary values. Consider the equation

$$\begin{aligned} u_{xx} + u_{yy} &= f(x,y) \quad \text{for} \quad (x,y)\epsilon R \ , \\ u(x,y) &= g(x,y) \qquad \text{for} \quad (x,y)\epsilon \partial R \ . \end{aligned} \tag{4.1}$$

(Here ∂R indicates the boundary of R.) We assume that the reader is familiar with the general technique of imposing a mesh of discrete points onto R and approximating Eq. (4.1). The equation $u_{xx} + u_{yy} = f(x,y)$ is approximated at (x_i,y_j) by

$$\frac{v_{i-1,j} - 2v_{i,j} + v_{i+1,j}}{(\Delta x)^2} + \frac{v_{i,j-1} - 2v_{i,j} + v_{i,j+1}}{(\Delta y)^2}$$

$$= f_{i,j} \qquad (1 \le i \le n-1,\ 1 \le j \le m-1),$$

with

$$v_{o,j} = g_{o,j}, \quad v_{m,j} = g_{m,j} \quad (1 \le j \le m-1),$$

and

$$v_{i,o} = g_{i,o}, \quad v_{i,m} = g_{i,j} \quad (1 \le i \le n-1).$$

Then v_{ij} is an approximation to $u(x_i,y_j)$, and $f_{i,j} = f(x_i,y_j)$, $g_{i,j} = g(x_i,y_j)$. Hereafter, we assume that

$$m = 2^{k+1} \ .$$

When $u(x,y)$ is specified on the boundary, we have the Dirichlet boundary condition. For simplicity, we shall assume hereafter that $\Delta x = \Delta y$. This leads to the system of equations

$$M_D \underset{\sim}{v} = \underset{\sim}{y} ,$$

Where M_D is of the form of Eq. (2.1) with

$$A = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & 1 \\ & & & 1 & -4 \end{bmatrix}_{(n-1) \times (n-1)} ,$$

$$\text{and } T = I_{n-1} .$$

The matrix I_{n-1} indicates the identity matrix of order (n-1). A and T are symmetric and commute, and, thus, the results of Sections 2 and 3 are applicable. In addition, since A is tridiagonal, the use of the factorization (3.10) is greatly simplified.

5. Neumann Boundary Conditions

When the normal derivative, $\frac{\partial u}{\partial n}$, is specified on the boundary, we have the Neumann boundary condition. Assume that

$$\frac{\partial u}{\partial n} = g(x,y) \text{ when } (x,y) \epsilon \partial R \ .$$

We make the approximation

$$\frac{\partial u}{\partial x} \doteq \frac{u(x+\Delta x,y) - u(x-\Delta x,y)}{2\Delta x} \ , \quad \frac{\partial u}{\partial y} \doteq \frac{u(x,y+\Delta y) - u(x,y-\Delta y)}{2\Delta y} \ .$$

This approximation leads to the matrix equation

$$M_N \underset{\sim}{v} = \underset{\sim}{\chi} \ ,$$

where M_N is of the form

$$\begin{bmatrix} A & 2T & & & \\ T & A & T & & \\ & \ddots & \ddots & \ddots & \\ & & T & \ddots & T \\ & & & 2T & A \end{bmatrix} . \tag{5.1}$$

Here,

$$\begin{bmatrix} -4 & 2 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 2 & -4 \end{bmatrix}_{(n+1) \times (n+1)} , \quad T = I_{n+1} \ .$$

Again A and T commute, but M_N no longer has the structure given by Eq. (2.2). Therefore it is necessary to modify the algorithm of Section 3.

From Eq. (5.1), we see that

$$\begin{aligned} A\underset{\sim}{v}_0 + 2T\underset{\sim}{v}_1 &= \underset{\sim}{y}_0 , \\ T\underset{\sim}{v}_{j-1} + A\underset{\sim}{v}_j + T\underset{\sim}{v}_{j+1} &= \underset{\sim}{y}_j , \qquad j = 1,2,\ldots,m-1, \\ 2T\underset{\sim}{v}_{m-1} + A\underset{\sim}{v}_m &= \underset{\sim}{y}_m . \end{aligned}$$

Performing the cyclic reduction as in Section 3, we have

$$\left.\begin{aligned} (2T^2-A^2)\underset{\sim}{v}_0 + 2T\underset{\sim}{v}_2 &= -A\underset{\sim}{y}_0 + 2T\underset{\sim}{y}_1 , \\ T^2\underset{\sim}{v}_{j-2} + (2T^2-A^2)\underset{\sim}{v}_j + T^2\underset{\sim}{v}_{j+2} &= T(\underset{\sim}{y}_{j-1} + \underset{\sim}{y}_{j+1}) - A\underset{\sim}{y}_j , \\ & \qquad j = 2,4,\ldots,m-2 , \\ 2T\underset{\sim}{v}_{m-2} + (2T^2-A^2)\underset{\sim}{v}_m &= 2T\underset{\sim}{y}_{m-1} - A\underset{\sim}{y}_m . \end{aligned}\right\} \qquad (5.2)$$

The similarity of Eq. (5.2) to Eq. (3.1) should now be evident. Since Eq. (5.2) is of block dimension 2^k+1, after k steps we have the system

$$\begin{bmatrix} A^{(k)} & 2T^{(k)} & 0 \\ T^{(k)} & A^{(k)} & T^{(k)} \\ 0 & 2T^{(k)} & A^{(k)} \end{bmatrix} \begin{bmatrix} \underset{\sim}{v}_0 \\ \underset{\sim}{v}_{2^k} \\ \underset{\sim}{v}_{2^k+1} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{y}_0^{(k)} \\ \underset{\sim}{y}_{2^k}^{(k)} \\ \underset{\sim}{y}_{2^k+1}^{(k)} \end{bmatrix} , \qquad (5.3)$$

and a final reduction yields

$$\left[4\left(T^{(k)}\right)^2 - \left(A^{(k)}\right)^2\right]\underset{\sim}{v}_{2^k} = T\left(\underset{\sim}{y}_0^{(k)} + \underset{\sim}{y}_{2^k+1}^{(k)}\right) - A\underset{\sim}{y}_{2^k}^{(k)} . \qquad (5.4)$$

Equation (5.4) is equivalent to writing

$$P^{(N)}_{2^{k+1}}(A,T)\underset{\sim}{v}_{2^k} = \underset{\sim}{\chi}^{(k+1)}_{2^k} \quad , \tag{5.5}$$

where $P^{(N)}_{2^{k+1}}(A,T)$ is again a polynomial of degree 2^{k+1} in A and T.

Note that from Eq. (3.8),

$$p^{(N)}_{2^{r+1}}(a,t) = 2t^{2^{r+1}} - \left[p^{(N)}_{2^r}(a,t)\right]^2 , \quad r = 0,1,\ldots,k-1 ,$$

and from Eq. (5.4),

$$p^{(N)}_{2^{k+1}}(a,t) = 4t^{2^{k+1}} - \left[p^{(N)}_{2^k}(a,t)\right]^2 .$$

Therefore, since $p_{2^k}(a,t) = -2t^{2^k} \cos 2^k\theta$,

$$p^{(N)}_{2^{k+1}}(a,t) = \left(2t^{2^k} \sin 2^k\theta\right)^2 ,$$

and, thus,

$$p^{(N)}_{2^{k+1}}(a,t) = 0 \quad \text{when } a/2t = -\cos\frac{j\pi}{2^k} \quad \text{for } j = 1,2,\ldots,2^{k+1} .$$

Consequently, we may rewrite Eq. (5.5) as

$$\prod_{j=1}^{2^{k+1}} \left[\mathbf{A} + 2\cos\theta^{(k+1)}_j T\right]\underset{\sim}{v}_{2^k} = -\underset{\sim}{\chi}^{(k+1)}_{2^k} , \tag{5.6}$$

where $\theta^{(k+1)}_j = j\pi/2^k$. Again, $\underset{\sim}{v}_{2^k}$ is determined by solving 2^{k+1} tridiagonal systems. The other components of $\underset{\sim}{v}$ are solved in the manner indicated in Section 3.

Note that the application of matrix decomposition to this problem only requires the modification of the tridiagonal matrices Γ_i in (2.8); i.e., the first super-diagonal and the last sub-diagonal elements must be $2\omega_i$.

It is well known that the solution to Poisson's equation is not unique in this case. Therefore, we would expect the finite-difference approximation to be singular. This is easy to verify by noting that

$$M_N \underset{\sim}{e} = \underset{\sim}{0},$$

where $\underset{\sim}{e}^T = (1,1,\ldots,1)$. In addition, one of the systems of the tridiagonal matrices in Eq. (5.6) is also singular. On a uniform mesh, the eigenvalues of $(A + 2\cos\theta_j T)$ satisfy the equation

$$\lambda_\ell(A + 2\cos\theta_j T) = 4 - 2\cos\left(\frac{\ell\pi}{n}\right) + 2\cos\left(\frac{j\pi}{2^k}\right)$$

$$(\ell = 0,1,2,\ldots,n;\ j = 1,2,\ldots,2^{k+1}).$$

Then $\lambda_0 = 0$ when $j = 2^k$. Similarly, one can show that Γ_0 of matrix decomposition is singular. Normally, the physics of the problem determines the coefficient of the homogeneous solution for the singular case.

6. Periodic Boundary Conditions

In this section we shall consider the problem of solving the finite-difference approximation to Poisson's equation over a rectangle when

$$u(x_o,y) = u(x_n,y) \ , \qquad u(x,y_o) = u(x,y_n) \ . \tag{6.1}$$

The periodic boundary conditions Eq. (6.1) lead to the matrix equation

$$M_P \underset{\sim}{v} = \underset{\sim}{y} \ , \tag{6.2}$$

where

$$M_P = \begin{bmatrix} A & T & 0 & \cdot & \cdot & 0 & T \\ T & A & T & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & & \bigcirc & \cdot \\ \cdot & \bigcirc & \cdot & \cdot & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & & & T & A & T \\ T & 0 & \cdot & \cdot & 0 & T & A \end{bmatrix} \ ,$$

and

$$A = \begin{bmatrix} -4 & 1 & 0 & \cdot & \cdot & 0 & 1 \\ 1 & -4 & 1 & 0 & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & & \bigcirc & \cdot \\ \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \bigcirc & & \cdot & \cdot & \cdot & 0 \\ 0 & & & & 1 & -4 & 1 \\ 1 & 0 & \cdot & \cdot & 0 & 1 & -4 \end{bmatrix}_{n \times n} \ , \qquad T = I_n \ .$$

Note that M_P is "almost" an m block tridiagonal system and, similarly, A is "almost" an n x n tridiagonal matrix. The cyclic reduction can again be performed on Eq. (6.2), and this leads to the reduced system

$$
\begin{aligned}
&(2T^2-A^2)\underset{\sim}{v}_2 + T^2\underset{\sim}{v}_4 + T^2\underset{\sim}{v}_m = T(\underset{\sim}{y}_1 + \underset{\sim}{y}_3) - A\underset{\sim}{y}_2 ,\\
&T^2\underset{\sim}{v}_{j-2} + (2T^2-A^2)\underset{\sim}{v}_j + T^2\underset{\sim}{v}_{j+2} = T(\underset{\sim}{y}_{j-1} + \underset{\sim}{y}_{j+1}) - A\underset{\sim}{y}_j ,\\
&\qquad\qquad j = 2,4,\ldots,m-2 ,\\
&T^2\underset{\sim}{v}_2 + T^2\underset{\sim}{v}_{m-2} + (2T^2-A^2)\underset{\sim}{v}_m = T(\underset{\sim}{y}_1 + \underset{\sim}{y}_{m-1}) - A\underset{\sim}{y}_m .
\end{aligned}
\tag{6.3}
$$

The similarity with the previous cases is again evident. Equation (6.3) has block dimension 2^k. After (k-1) reductions, we have

$$
M_P^{(k-1)} = \begin{bmatrix}
A^{(k-1)} & T^{(k-1)} & 0 & T^{(k-1)} \\
T^{(k-1)} & A^{(k-1)} & T^{(k-1)} & 0 \\
0 & T^{(k-1)} & A^{(k-1)} & T^{(k-1)} \\
T^{(k-1)} & 0 & T^{(k-1)} & A^{(k-1)}
\end{bmatrix} ,
$$

and, finally, after k reductions,

$$
\begin{bmatrix}
A^{(k)} & 2T^{(k)} \\
2T^{(k)} & A^{(k)}
\end{bmatrix}
\begin{bmatrix}
\underset{\sim}{v}_{2^k} \\
\underset{\sim}{v}_{2^k+1}
\end{bmatrix}
=
\begin{bmatrix}
\underset{\sim}{y}_{2^k}^{(k)} \\
\underset{\sim}{y}_{2^k+1}^{(k)}
\end{bmatrix} . \tag{6.4}
$$

From Eq. (6.4), the final equation becomes

$$
\left[4T^{(k)\,2} - A^{(k)\,2}\right]\underset{\sim}{v}_{2^k} = \underset{\sim}{y}_{2^k}^{(k+1)} ,
$$

which is equivalent to

$$P^{(N)}_{2^{k+1}}(A,T)\underset{\sim}{v}_{2^k} = \underset{\sim}{\chi}^{(k+1)}_{2^k} \quad .$$

The analysis of the factorization of $P^{(N)}_{2^{k+1}}(A,T)$ is identical to that of the Neumann case, including the fact that one of the factors of the polynomial must be singular.

Again the application of matrix decomposition to (6.2) is straightforward; however, the resulting Γ_i matrices are no longer tridiagonal since ω_i appears in the (1,n) and (n,1) elements. Standard algorithms for solving tridiagonal systems can be modified to solve these systems such that storage of the full n x n matrix is avoided. As above, one of the Γ_i will be singular.

7. Higher-Dimensional Problems

It is not difficult to extend the applications given in Sections 4, 5, and 6 to higher-dimensional problems. We show this by a simple example. Consider Poisson's equation in three dimensions over the rectangle R:

$$u_{xx} + u_{yy} + u_{zz} = f(x,y,z) \qquad (x,y,z)\epsilon R \ .$$

$$u(x,y,z) = g(x,y,z) \qquad (x,y,z)\epsilon \partial R \ .$$

Again, we assume that the mesh is uniform in each direction so that

$$x_{i+1} = x_i + \Delta x \qquad (i = 0,1,\ldots,n) \ ,$$

$$y_{j+1} = y_j + \Delta y \qquad (j = 0,1,\ldots,m) \ ,$$

$$z_{\ell+1} = z_\ell + \Delta z \qquad (\ell = 0,1,\ldots,p) \ .$$

At the point (x_i,y_j,z_ℓ), we approximate $u(x_i,y_j,z_\ell)$ by $v_{i,j,\ell}$. Let

$$\underset{\sim}{w}_\ell = \begin{bmatrix} \underset{\sim}{v}_{1,\ell} \\ \underset{\sim}{v}_{2,\ell} \\ \vdots \\ \underset{\sim}{v}_{m-1,\ell} \end{bmatrix} , \quad \text{where} \quad \underset{\sim}{v}_{j,\ell} = \begin{bmatrix} v_{1,j,\ell} \\ v_{2,j,\ell} \\ \vdots \\ v_{n-1,j,\ell} \end{bmatrix} .$$

Assume that the usual finite-difference approximation is made to u_{zz} for fixed (x,y,z), namely,

$$u_{zz}(x,y,z) \doteq \frac{u(x,y,z-\Delta z) - 2u(x,y,z) + u(x,y,z+\Delta z)}{(\Delta z)^2} \ .$$

It is then easy to verify that for $\ell = 1,2,\ldots,p-1$,

$$\underset{\sim}{w}_{\ell-1} + H\underset{\sim}{w}_{\ell} + \underset{\sim}{w}_{\ell+1} = \underset{\sim}{f}_{\ell} ,$$

where $\underset{\sim}{w}_{0}$ and $\underset{\sim}{w}_{p}$ are prescribed by the initial conditions and $\underset{\sim}{f}_{\ell}$ is a function of the given data. Thus, again, we have a block tridiagonal matrix, and can use the previously developed methods. Note, also, that H is a block tridiagonal matrix so that one can solve any of the eliminated systems of equations by applying the CORF algorithm repeatedly. Other boundary conditions can be handled in the manner prescribed in Sections 5 and 6.

8. Further Applications

Consider the elliptic equation in self-adjoint form

$$\begin{aligned} &\alpha(x)u_{x\,x} + \beta(y)u_{y\,y} + u(x,y) = q(x,y) \ , \quad (x,y)\epsilon R \ , \\ &u(x,y) = g(x,y) \ , \quad (x,y)\epsilon \partial R \ . \end{aligned} \tag{8.1}$$

Many equations can be transformed to this form. The usual five-point difference equation when $\Delta x = \Delta y$ leads to the following equation:

$$\begin{aligned} &-\alpha_{i+1/2}\, v_{i+1,j} - \alpha_{i-1/2}\, v_{i-1,j} - \beta_{j+1/2}\, v_{i,j+1} - \beta_{j-1/2}\, v_{i,j-1} \\ &+ \left[\alpha_{i+1/2} + \alpha_{i-1/2} + \beta_{j+1/2} + \beta_{j-1/2} - (\Delta x)^2\right] v_{i,j} \doteq -(\Delta x)^2\, q_{i,j} \ , \end{aligned} \tag{8.2}$$

where

$$\alpha_{i\pm 1/2} = \alpha(x_i \pm 1/2\Delta x), \quad \beta_{j\pm 1/2} = \beta(y_j \pm \Delta y),$$

$$q_{i,j} = q(x_i, y_j) \ .$$

If the equations are ordered with

$$\underset{\sim}{v} = \begin{bmatrix} \underset{\sim}{v}_1 \\ \underset{\sim}{v}_2 \\ \cdot \\ \cdot \\ \cdot \\ \underset{\sim}{v}_{m-1} \end{bmatrix} \ , \qquad \underset{\sim}{v}_j = \begin{bmatrix} v_{1,j} \\ v_{2,j} \\ \cdot \\ \cdot \\ \cdot \\ v_{n-1,j} \end{bmatrix} \ ,$$

the linear system of equations $M\underset{\sim}{v} = \underset{\sim}{d}$ will have the block form

$$\begin{bmatrix} A_1 & T_1 & & \\ T_1 & A_2 & T_2 & \\ & \ddots & \ddots & \ddots \\ & & T_{m-2} & A_{m-1} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{m-1} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{m-1} \end{bmatrix} .$$

Here

$$A_j = \left[\beta_{j+1/2} + \beta_{j-1/2} - (\Delta x)^2\right] I +$$

$$+ \begin{bmatrix} \alpha_{1/2} + \alpha_{3/2} & -\alpha_{3/2} & & & \\ -\alpha_{3/2} & \alpha_{3/2} + \alpha_{5/2} & -\alpha_{5/2} & & \\ & \ddots & \ddots & \ddots & \\ & & & -\alpha_{n-3/2} & \\ & & -\alpha_{n-3/2} & \alpha_{n-3/2} + \alpha_{n-1/2} \end{bmatrix} ,$$

$$\equiv \gamma_j I + C ,$$

$$T_j = \beta_{j+1/2} I .$$

Now suppose we have the decomposition

$$Q^T C Q = \Phi ,$$

where $Q^T Q = I$ and $\mathrm{diag}(\Phi) = (\varphi_1, \varphi_2, \ldots, \varphi_{n-1})$. Thus,

$$\lambda_i(A_j) = \gamma_j + \varphi_i \ , \quad (i = 1,2,\ldots,n-1) \ ,$$

$$\equiv \lambda_{i,j} \ .$$

As in Section 2, we define

$$\underset{\sim}{\bar{v}}_j = Q^T \underset{\sim}{v}_j \ , \quad \underset{\sim}{\bar{d}}_j = Q^T \underset{\sim}{d}_j \ ,$$

and after a suitable permutation we are led to the equations

$$\Gamma_i \underset{\sim}{\hat{v}}_i = \underset{\sim}{\hat{d}}_i \ , \quad i = 1,2,\ldots,n-1 \ ,$$

where

$$\Gamma_i = \begin{bmatrix} \lambda_{i,1} & \beta_{3/2} & & & & \\ \beta_{3/2} & \lambda_{i,2} & \beta_{5/2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \beta_{m-3/2} \\ & & & & \beta_{m-3/2} & \lambda_{i,m-1} \end{bmatrix} ,$$

$$\underset{\sim}{\hat{v}}_i = \begin{bmatrix} \bar{v}_{i1} \\ \bar{v}_{i2} \\ \vdots \\ \bar{v}_{i,m-1} \end{bmatrix} , \qquad \underset{\sim}{\hat{d}}_i = \begin{bmatrix} \bar{d}_{i1} \\ \bar{d}_{i2} \\ \vdots \\ \bar{d}_{i,m-1} \end{bmatrix} .$$

Thus, the vector $\hat{\underset{\sim}{v}}_i$ satisfies a symmetric tridiagonal system of equations. Again, once $\hat{\underset{\sim}{v}}_i$ is computed for all i, it is possible to compute $\underset{\sim}{v}$.

Lynch et al[10] have considered a similar methods, but their algorithm requires more operations. Unfortunately, it does not seem possible to use the methods of Section 3 on Eq. (8.2) in this situation.

Now we may write the equivalent to Poisson's equation in two dimensions in cylindrical coordinates as follows:

$$(r\, u_r)_r + r^{-1}\, u_{\theta\theta} = s(r,\theta) \ ,$$

and

$$(r\, u_r)_r + r\, u_{zz} \quad = t(r,z) \ .$$

The matrix A will still be tridiagonal, and T will be a diagonal matrix with positive diagonal elements. We may make the transformation $\tilde{\underset{\sim}{v}}_j = T^{1/2} \underset{\sim}{v}_j$, and are thus led to the equations

$$\tilde{\underset{\sim}{v}}_{j-1} + T^{-1/2}\, A\, T^{-1/2} \tilde{\underset{\sim}{v}}_j + \tilde{\underset{\sim}{v}}_{j+1} = T^{-1/2}\, \underset{\sim}{d}_j \ .$$

Thus, by ordering the equations correctly and by making a simple transformation, one can apply the cyclic reduction and the CORF algorithm to solve the finite-difference approximation to Poisson's equation in cylindrical coordinates.

Another situation in which the methods of Sections 2 and 3 are applicable is in using the nine-point formula to solve the finite-difference approximation to Poisson's equation in the rectangle. In this case, when $\Delta x = \Delta y$,

$$A = \begin{bmatrix} -20 & 4 & & & \\ 4 & -20 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 4 \\ & & & 4 & -20 \end{bmatrix}_{(n-1)\times(n-1)},$$

$$T = \begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 4 \end{bmatrix}_{(n-1)\times(n-1)}.$$

It is easy to verify that $AT = TA$, and that the eigenvalues of A and T are

$$\lambda_i(A) = -20 + 8 \cos \frac{i\pi}{n}, \qquad (i = 1,2,\ldots,n-1),$$

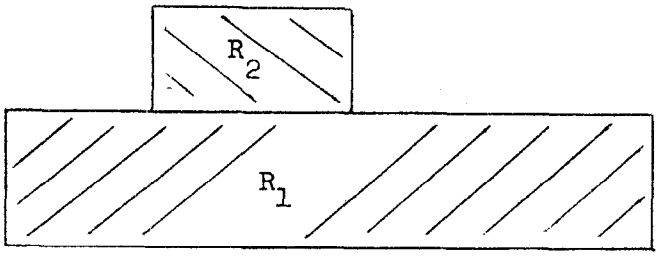
$$\lambda_i(T) = 4 + 2 \cos \frac{i\pi}{n}, \qquad (i = 1,2,\ldots,n-1).$$

Because of the structure of A and T, the fast Fourier transform may be employed when using the methods of Section 2.

We leave as an exercise for the reader the application of the methods in Sections 2 and 3 to the biharmonic equation.

9. Non-Rectangular Regions

In many situations, one wishes to solve an elliptic equation over the region



We shall assume that Dirichlet boundary conditions are given. When Δx is the same throughout the region, one has a matrix equation of the form

$$\left[\begin{array}{c|c} G & \\ \hline & H \end{array}\right] \left[\begin{array}{c} \underset{\sim}{x}^{(1)} \\ \hline \underset{\sim}{x}^{(2)} \end{array}\right] = \left[\begin{array}{c} \underset{\sim}{y}^{(1)} \\ \hline \underset{\sim}{y}^{(2)} \end{array}\right], \qquad (9.1)$$

where

$$G = \begin{bmatrix} A & T & & \\ T & A & \ddots & \\ & \ddots & \ddots & T \\ & & T & A \end{bmatrix}, \quad H = \begin{bmatrix} B & S & & \\ S & B & \ddots & \\ & \ddots & \ddots & S \\ & & S & B \end{bmatrix}. \qquad (9.2)$$

Also, we write

$$\underset{\sim}{x}^{(1)} = \begin{bmatrix} \underset{\sim}{x}_1^{(1)} \\ \underset{\sim}{x}_2^{(1)} \\ \vdots \\ \underset{\sim}{x}_r^{(1)} \end{bmatrix}, \qquad \underset{\sim}{x}^{(2)} = \begin{bmatrix} \underset{\sim}{x}_1^{(2)} \\ \underset{\sim}{x}_2^{(2)} \\ \vdots \\ \underset{\sim}{x}_s^{(2)} \end{bmatrix}. \tag{9.3}$$

We assume again that AT = TA and BS = SB.

From Eq. (9.1), we see that

$$\underset{\sim}{x}^{(1)} = G^{-1}\,\underset{\sim}{y}^{(1)} - G^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ P \end{bmatrix} \underset{\sim}{x}_1^{(2)}, \tag{9.4}$$

and

$$\underset{\sim}{x}^{(2)} = H^{-1}\,\underset{\sim}{y}^{(2)} - H^{-1} \begin{bmatrix} P^T \\ 0 \\ \vdots \\ 0 \end{bmatrix} \underset{\sim}{x}_r^{(1)}. \tag{9.5}$$

Now let us write

$$G\underset{\sim}{z}^{(1)} = \underset{\sim}{y}^{(1)}, \quad H\underset{\sim}{z}^{(2)} = \underset{\sim}{y}^{(2)}, \tag{9.6}$$

$$GW^{(1)} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ P \end{bmatrix}, \qquad HW^{(2)} = \begin{bmatrix} P^T \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{9.7}$$

Then as we partition the vectors $\underset{\sim}{z}^{(1)}$, $\underset{\sim}{z}^{(2)}$ and the matrices $W^{(1)}$ and $W^{(2)}$ as in Eq. (9.3), Eqs. (9.4) and (9.5) become

$$\left.\begin{aligned} \underset{\sim}{x}_j^{(1)} &= \underset{\sim}{z}_j^{(1)} - W_j^{(1)}\, \underset{\sim}{x}_1^{(2)} , \qquad (j = 1,2,\ldots,r) , \\ \underset{\sim}{x}_j^{(2)} &= \underset{\sim}{z}_j^{(2)} - W_j^{(2)}\, \underset{\sim}{x}_r^{(1)} , \qquad (j = 1,2,\ldots,s) . \end{aligned}\right\} . \tag{9.8}$$

For Eq. (9.8), we have

$$\begin{bmatrix} I & W_r^{(1)} \\ W_1^{(2)} & I \end{bmatrix} \begin{bmatrix} \underset{\sim}{x}_r^{(1)} \\ \underset{\sim}{x}_1^{(2)} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{z}_r^{(1)} \\ \underset{\sim}{z}_1^{(2)} \end{bmatrix} .$$

This system of equations is two-cyclic, so we may reduce the system to

$$I - W_r^{(1)}\, W_1^{(2)}\ \ \underset{\sim}{x}_r^{(1)} = \underset{\sim}{z}_r^{(1)} - W_r^{(1)}\ \underset{\sim}{z}_1^{(2)} . \tag{9.9}$$

This system of equations can most easily be solved using Gaussian elimination. Once the two-cyclic system of Eq. (9.9) has been solved, all other components may be computed using Eq. (9.8) or by solving the system

$$G\underset{\sim}{x}^{(1)} = \underset{\sim}{y}^{(1)} - \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ P \end{bmatrix} \underset{\sim}{x}_1^{(2)} ,$$

$$H\underset{\sim}{x}^{(2)} = \underset{\sim}{y}^{(2)} - \begin{bmatrix} P^T \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \underset{\sim}{x}_r^{(1)} .$$

If System (9.1) is to be solved for a number of different right-hand sides, it is best to save the LU decomposition of

$$\left(I - W_r^{(1)} \, W_l^{(2)} \right) . \tag{9.10}$$

Thus, the algorithm proceeds as follows.

1. Solve for $\underset{\sim}{z}_r^{(1)}$ and $\underset{\sim}{z}_l^{(2)}$ using the methods of Section 2 or 3.

2. Solve for $W_r^{(1)}$ and $W_l^{(2)}$ using the methods of Section 2 or 3.

3. Solve Eq. (9.9) using Gaussian elimination. Save the LU decomposition of Eq. (9.10).

4. Solve for the unknown components of $\underset{\sim}{x}^{(1)}$ and $\underset{\sim}{x}^{(2)}$.

10. Accuracy of the CORF algorithm

As will be shown in Section 11, the CORF algorithm and the Buneman algorithms are mathematically identical. The difference between the methods lies in the way the right-hand side is calculated at each stage of the reduction. To the authors' knowledge, this is the only direct method for solving linear equations in which the right-hand side of the equations plays an important rôle in the numerical solution of the equations. In this section, we show the difficulties encountered in using the CORF algorithm. In Section 13, we will prove the stability of the Buneman algorithms.

Recall from Section 3 that it is possible to compute $A^{(r)}\underset{\sim}{y}_j^{(r)}$ by the following algorithm:

$$\underset{\sim}{\eta}_0 = -2\underset{\sim}{y}_j^{(r)} \quad , \quad \underset{\sim}{\eta}_1 = A\underset{\sim}{y}_j^{(r)}$$

$$\underset{\sim}{\eta}_s = -A\underset{\sim}{\eta}_{s-1} - T^2\underset{\sim}{\eta}_{s-2} \qquad \text{for} \quad s = 2,3,\ldots,2^r \tag{10.1}$$

so that

$$\underset{\sim}{\eta}_{2^r} = A^{(r)}\underset{\sim}{y}_j^{(r)} \quad .$$

Because of round-off error, one actually computes the sequence

$$\tilde{\underset{\sim}{\eta}}_0 = -2\underset{\sim}{y}_j^{(r)} \quad , \quad \tilde{\underset{\sim}{\eta}}_1 = A\underset{\sim}{y}_j^{(r)} + \underset{\sim}{\delta}_0$$

$$\tilde{\underset{\sim}{\eta}}_s = -A\tilde{\underset{\sim}{\eta}}_{s-1} - T^2\,\tilde{\underset{\sim}{\eta}}_{s-2} + \underset{\sim}{\delta}_{s-1} \qquad (s = 2,\ldots,2^r) \quad , \tag{10.2}$$

where $\underset{\sim}{\delta}_s$ is the perturbation induced by the roundoff error. Again as in Section 2, we write

$$A = Q \Lambda Q^T \quad , \quad T = Q \Omega Q^T \tag{10.3}$$

where Q is the set of orthonormalized eigenvectors of A and T, and Λ and Ω are the diagonal matrices of eigenvalues of A and T, respectively. Thus, substituting Eq. (10.3) into Eq. (10.2), we have

$$\underset{\sim}{\xi}_0 = -2\underset{\sim}{\bar{y}} \quad , \quad \underset{\sim}{\xi}_1 = -\frac{1}{2} \Lambda \underset{\sim}{\xi}_0 + \underset{\sim}{\tau}_0 \tag{10.4a}$$

$$\underset{\sim}{\xi}_s = -\Lambda \underset{\sim}{\xi}_{s-1} - \Omega^2 \underset{\sim}{\xi}_{s-2} + \underset{\sim}{\tau}_{s-1} \tag{10.4b}$$

where

$$\underset{\sim}{\bar{y}} = Q^T \underset{\sim}{y}_j^{(r)} \quad , \quad \underset{\sim}{\xi}_s = Q^T \underset{\sim}{\eta}_s \quad , \quad \underset{\sim}{\tau}_s = Q^T \underset{\sim}{\delta}_s \quad .$$

Because Λ and Ω are diagonal, we may write an equation for each component of $\underset{\sim}{\xi}_s$; namely,

$$\xi_{j,s+1} + \lambda_j \xi_{j,s} + \omega_j^2 \xi_{j,s-1} = \tau_{j,s} \qquad (j = 1,2,\ldots,p) \quad . \tag{10.5}$$

The solution of Eq. (10.5) can be given explicitly. Consider the characteristic equation

$$\varphi_j(\alpha) \equiv \alpha^2 + \lambda_j \alpha + \omega_j^2 = 0$$

which has roots β_j and γ_j, then

$$\xi_{j,s} = \frac{\beta_j^s - \gamma_j^s}{\beta_j - \gamma_j}\,\xi_{j,1} - \beta_j\gamma_j\,\frac{\beta_j^{s-1} - \gamma_j^{s-1}}{\beta_j - \gamma_j}\,\xi_{j,0}$$

$$+ \sum_{k=1}^{s-1} \frac{\beta_j^{s-k} - \gamma_j^{s-k}}{\beta_j - \gamma_j}\,\tau_{j,k} \qquad \text{when} \quad \beta_j \neq \gamma_j \tag{10.6a}$$

$$= s\beta_j^{s-1}\xi_{j,1} - (s-1)\beta_j^s\xi_{j,0} + \sum_{k=1}^{s-1}(s-k)\beta_j^{s-k-1}\tau_{j,k} \quad \text{when} \quad \beta_j = \gamma_j \,. \tag{10.6b}$$

Let

$$-\lambda_j/2\omega_j = \cos\theta_j \qquad \text{when} \quad |\lambda_j/2\omega_j| \leq 1$$

$$= \cosh z_j \qquad \text{when} \quad |\lambda_j/2\omega_j| \geq 1 \,.$$

Then using the initial conditions of Eq. (10.4a), we may write Eq. (10.6a) as

$$\xi_{j,s} = -2\omega_j^s \cos(s\,\theta_j)\bar{y}_j + \sum_{k=0}^{s-1} \omega_j^{s-k-1}\,\frac{\sin(s-k)\theta_j}{\sin\theta_j}\,\tau_{j,k}$$

$$\text{when} \quad |\lambda_j/2\omega_j| < 1 \tag{10.7a}$$

$$= -2\omega_j^s \cosh(s\,z_j)\bar{y}_j + \sum_{k=0}^{s-1} \omega_j^{s-k-1}\,\frac{\sinh(s-k)z_j}{\sinh z_j}\,\tau_{j,k}$$

$$\text{when} \quad |\lambda_j/2\omega_j| > 1 \,. \tag{10.7b}$$

Note that

$$-2\omega_j^s \times \begin{cases} \cos s\,\theta_j \\ \cosh s\,z_j \end{cases} \equiv p_s(\lambda_j, \omega_j) \qquad (10.8)$$

given in Section 3. Thus

$$\tilde{\eta}_s = P_s(A,T)y_j^{(r)} + \sum_{k=0}^{s-1} Q\, S_{s-k}\, Q^T \delta_k \quad , \qquad (10.9)$$

where

$$\{S_m\}_{ij} = \omega_j^{m-1} \times \begin{cases} \dfrac{\sin m\,\theta_j}{\sin \theta_j} & \text{when } |\lambda_j/2\omega_j| < 1 \text{ and } i = j \\ \dfrac{\sinh m\, z_j}{\sinh z_j} & \text{when } |\lambda_j/2\omega_j| > 1 \text{ and } i = j \end{cases}$$

$$= 0 \qquad \text{for } i \neq j \ .$$

Therefore, if $|\lambda_j/2\omega_j| > 1$, the effect of the round-off error can be catastrophic. However, if $|\lambda_j/2\omega_j| \leq 1$, we see from Eq. (10.9) that $\tilde{\eta}_{2^r}$ may be a good numerical approximation to $A^{(r)}y_j^{(r)}$.

We now apply the above results to Poisson's equation with Dirichlet boundary conditions. For the five-point difference operator with mesh width Δx in the x-direction and Δy in the y-direction, we have

$$\lambda_j = -2[1 + \rho^2(1 - \cos \tfrac{j\pi}{p+1})] \ , \quad \omega_j = 1$$

and

$$\rho = (\Delta x/\Delta y) \text{ or } (\Delta y/\Delta x),$$

depending on how one orders the equations. By inspection

$$|\lambda_j/2\omega_j| > 1$$

for all j, and hence as s increases, equation (10.1) leads to a numerically unstable algorithm. A similar result holds for the nine point difference approximation to Poisson's equation.

Although the above results were obtained under the assumption of (10.1), similar results will be obtained regardless of how $A^{(r)}\ \underset{\sim}{y}_j^{(r)}$ is computed. The problem is that $A^{(r)}$ becomes very ill-conditioned as r increases. For the five point approximation, the ratio of the largest eigenvalue of $A^{(r)}$ to its smallest eigenvalues is

$$\frac{\omega_n^{2^r}\cosh\ (2^r z_n)}{\omega_1^{2^r}\cosh\ (2^r z_1)} \simeq e^{2^r(z_n - z_1)} .$$

Given t-digits of arithmetic, it will generally be impossible to represent $\underset{\sim}{r}_{2^r}$ in t-digits whenever

$$2^r(z_n - z_1) \geq t .$$

As noted in Section 3, Hockney[2,9] has combined one or more steps of odd-even reduction with the Fast Fourier transform to produce a Poisson solver. The above analysis allows one to determine the number of reductions that can be safely performed, and careful attention must be given to it.

11. The Buneman algorithm and variants

In this section, we shall describe in detail the Buneman algorithm[2] and a variation of it. The difference between the Buneman algorithm and the CORF algorithm lies in the way the right hand-side is calculated at each stage of the reduction. Henceforth, we shall assume that in the system of Eqs. (2.5) $T = I_p$, the identity matrix of order p.

Again consider the system of equations as given by Eqs. (2.5) with $q = 2^{k+1}-1$. After one stage of cyclic reduction, we have

$$\underset{\sim}{x}_{j-2} + (2I_p - A^2)\underset{\sim}{x}_j + \underset{\sim}{x}_{j+2} = \underset{\sim}{y}_{j-1} + \underset{\sim}{y}_{j+1} - A\underset{\sim}{y}_j \qquad (11.1)$$

for $j = 2,4,\ldots,q-1$ with $\underset{\sim}{x}_o = \underset{\sim}{x}_{q+1} = \underset{\sim}{0}$, the null vector. Note that the right hand side of Eq. (11.1) may be written as

$$\underset{\sim}{y}_j^{(1)} = \underset{\sim}{y}_{j-1} + \underset{\sim}{y}_{j+1} - A\underset{\sim}{y}_j = A^{(1)}A^{-1}\underset{\sim}{y}_j + \underset{\sim}{y}_{j-1} + \underset{\sim}{y}_{j+1} - 2A^{-1}\underset{\sim}{y}_j \qquad (11.2)$$

where $A^{(1)} = (2I_p - A^2)$.

Let us define

$$\underset{\sim}{p}_j^{(1)} = A^{-1}\underset{\sim}{y}_j \quad , \quad \underset{\sim}{q}_j^{(1)} = \underset{\sim}{y}_{j-1} + \underset{\sim}{y}_{j+1} - 2\underset{\sim}{p}_j^{(1)} \quad .$$

Then

$$\underset{\sim}{y}_j^{(1)} = A^{(1)}\,\underset{\sim}{p}_j^{(1)} + \underset{\sim}{q}_j^{(1)} \quad . \qquad (11.3)$$

After r reductions, we have by Eq. (3.3)

$$\underset{\sim}{y}_j^{(r+1)} = (\underset{\sim}{y}_{j-2^r}^{(r)} + \underset{\sim}{y}_{j+2^r}^{(r)}) - A^{(r)}\, \underset{\sim}{y}_j^{(r)} \quad . \tag{11.4}$$

Let us write in a fashion similar to Eq. (11.3),

$$\underset{\sim}{y}_j^{(r)} = A^{(r)}\, \underset{\sim}{p}_j^{(r)} + \underset{\sim}{q}_j^{(r)} \quad . \tag{11.5}$$

Substituting Eq. (11.5) into Eq. (11.4) and making use of the identity $(A^{(r)})^2 = 2I_p - A^{(r+1)}$ from Eq. (3.3), we have the following relationships:

$$\underset{\sim}{p}_j^{(r+1)} = \underset{\sim}{p}_j^{(r)} - (A^{(r)})^{-1}(\underset{\sim}{p}_{j-2^r}^{(r)} + \underset{\sim}{p}_{j+2^r}^{(r)} - \underset{\sim}{q}_j^{(r)}) \tag{11.6a}$$

$$\underset{\sim}{q}_j^{(r+1)} = \underset{\sim}{q}_{j-2^r}^{(r)} + \underset{\sim}{q}_{j+2^r}^{(r)} - 2\underset{\sim}{p}_j^{(r+1)} \tag{11.6b}$$

for $j = i2^{r+1}$ $(i = 1,2,\ldots,2^{k-r}-1)$ with

$$\underset{\sim}{p}_0^{(r)} = \underset{\sim}{p}_{2^{k+1}}^{(r)} = \underset{\sim}{q}_0^{(r)} = \underset{\sim}{q}_{2^{k+1}}^{(r)} = \underset{\sim}{0} \quad .$$

To compute $(A^{(r)})^{-1}(\underset{\sim}{p}_{j-2^r}^{(r)} + \underset{\sim}{p}_{j+2^r}^{(r)} - \underset{\sim}{q}_j^{(r)})$ in Eq. (11.6a), we solve the system of equations

$$A^{(r)}(\underset{\sim}{p}_j^{(r)} - \underset{\sim}{p}_j^{(r+1)}) = \underset{\sim}{p}_{j-2^r}^{(r)} + \underset{\sim}{p}_{j+2^r}^{(r)} - \underset{\sim}{q}_j^{(r)} \quad ,$$

where $A^{(r)}$ is given by the factorization Eq. (3.10); namely,

$$A^{(r)} = -\prod_{j=1}^{2^r} (A + 2\cos\theta_j^{(r)} I_p) ,$$

$$\theta_j^{(r)} = (2j - 1)\pi/2^{r+1} .$$

After k reductions, one has the equation

$$A^{(k)} \underset{\sim}{x}_{2^k} = A^{(k)} \underset{\sim}{p}_{2^k}^{(k)} + \underset{\sim}{q}_{2^k}^{(k)}$$

and hence

$$\underset{\sim}{x}_{2^k} = \underset{\sim}{p}_{2^k}^{(k)} + (A^{(k)})^{-1} \underset{\sim}{q}_{2^k}^{(k)} .$$

Again one uses the factorization of $A^{(k)}$ for computing $(A^{(k)})^{-1} \underset{\sim}{q}_{2^k}^{(k)}$. To back-solve, we use the relationship

$$\underset{\sim}{x}_{j-2^r} + A^{(r)} \underset{\sim}{x}_j + \underset{\sim}{x}_{j+2^r} = A^{(r)} \underset{\sim}{p}_j^{(r)} + \underset{\sim}{q}_j^{(r)}$$

for $j = i2^r$ $(i = 1,2,\ldots,2^{k+1-r}-1)$ with $\underset{\sim}{x}_0 = \underset{\sim}{x}_{2^{k+1}} = \underset{\sim}{0}$.

For $j = 2^r, 3\cdot 2^r, \ldots, 2^{k+1}-2^r$, we solve the system of equations

$$A^{(r)}(\underset{\sim}{x}_j - \underset{\sim}{p}_j^{(r)}) = \underset{\sim}{q}_j^{(r)} - (\underset{\sim}{x}_{j-2^r} + \underset{\sim}{x}_{j+2^r}) , \qquad (11.7)$$

using the factorization of $A^{(r)}$; hence

$$\underset{\sim}{x}_j = \underset{\sim}{p}_j^{(r)} + (\underset{\sim}{x}_j - \underset{\sim}{p}_j^{(r)}) \quad . \tag{11.8}$$

Thus the Buneman algorithm (variant 1) proceeds as follows:

1. Compute the sequence $\{\underset{\sim}{p}_j^{(r)}, \underset{\sim}{q}_j^{(r)}\}$ by Eq. (11.6) for $r = 1,\ldots,k$ with $\underset{\sim}{p}_j^{(0)} = \underset{\sim}{0}$ for $j = 0,\ldots,2^{k+1}$, and $\underset{\sim}{q}_j^{(0)} = \underset{\sim}{y}_j$ for $j = 1,2,\ldots,2^{k+1}-1$.

2. Back-solve for $\underset{\sim}{x}_j$ using Eqs. (11.7) and (11.8).

It is possible to eliminate the sequence $\{\underset{\sim}{p}_j^{(r)}\}$. From Eq. (11.6b) we note that

$$\underset{\sim}{p}_j^{(r+1)} = \frac{1}{2}\,(\underset{\sim}{q}_{j-2h}^{(r)} + \underset{\sim}{q}_{j+2h}^{(r)} - \underset{\sim}{q}_j^{(r+1)}) \quad , \tag{11.9}$$

where

$$h = 2^{r-1} \quad .$$

Using Eq. (11.9) in Eq. (11.6a) and modifying the subscripts and superscripts appropriately, we have

$$\begin{aligned}\underset{\sim}{q}_j^{(r+1)} = {} & \underset{\sim}{q}_{j-2h}^{(r)} - \underset{\sim}{q}_{j-h}^{(r-1)} + \underset{\sim}{q}_j^{(r)} - \underset{\sim}{q}_{j+h}^{(r-1)} + \underset{\sim}{q}_{j+2h}^{(r)} + \\ & + (A^{(r)})^{-1}[\underset{\sim}{q}_{j-3h}^{(r-1)} - \underset{\sim}{q}_{j-2h}^{(r)} + \underset{\sim}{q}_{j-h}^{(r-1)} - 2\underset{\sim}{q}_j^{(r)} + \\ & + \underset{\sim}{q}_{j+h}^{(r-1)} - \underset{\sim}{q}_{j+2h}^{(r)} + \underset{\sim}{q}_{j+3h}^{(r-1)}]\end{aligned} \tag{11.10}$$

for $j = (2^r, 2^{r+1}, \ldots, 2^{k+1}-2^r)$ with

$$\underset{\sim}{q}_0^{(r)} = \underset{\sim}{q}_{2^{k+1}}^{(r)} = \underset{\sim}{0} \qquad \text{for all } r ,$$

$$\underset{\sim}{q}_j^{(0)} = \underset{\sim}{y}_j \qquad \text{for } j = 1,2,\ldots,2^{k+1}-1 ,$$

$$\underset{\sim}{q}_j^{(1)} = \underset{\sim}{y}_{j-1} + \underset{\sim}{y}_{j+1} - 2A^{-1}\,\underset{\sim}{y}_j \qquad \text{for } j = 2,4,\ldots,2^{k+1}-2 .$$

To solve for $\underset{\sim}{x}_j$, we use the relationships Eqs. (11.7) and (11.9) so that

$$\underset{\sim}{x}_j = \frac{1}{2}\left(\underset{\sim}{q}_{j-h}^{(r-1)} + \underset{\sim}{q}_{j+h}^{(r-1)} - \underset{\sim}{q}_j^{(r)}\right) -$$

$$- \left(A^{(r)}\right)^{-1}\left(\underset{\sim}{x}_{j-2h} + \underset{\sim}{x}_{j+2h} - \underset{\sim}{q}_j^{(r)}\right) . \qquad (11.11)$$

Thus the Buneman algorithm (variant 2) proceeds as follows:

1. Compute the sequence $\{\underset{\sim}{q}_j^{(r)}\}$ by Eq. (11.10) for $r = 1,2,\ldots,k$.

2. Back-solve for $\underset{\sim}{x}_j$ using Eq. (11.11).

Note that the Buneman algorithm (variant 2) requires half the storage that the Buneman algorithm (variant 1) requires. However, the variant 2 algorithm requires approximately twice as many additions.

The $\underset{\sim}{p}_j$'s and $\underset{\sim}{q}_j$'s can be written in terms of the $\underset{\sim}{x}_j$'s. In Section 13, we shall show how this affects the stability of the methods. Note

$$\underset{\sim}{p}_j^{(1)} = A^{-1}\,\underset{\sim}{y}_j = \underset{\sim}{x}_j + A^{-1}(\underset{\sim}{x}_{j-1} + \underset{\sim}{x}_{j+1})$$

and

$$\underset{\sim}{q}_j^{(1)} = \underset{\sim}{y}_{j-1} + \underset{\sim}{y}_{j+1} - 2\underset{\sim}{p}_j^{(1)}$$

$$= \underset{\sim}{x}_{j-2} - (A)^{-1}\,A^{(1)}(\underset{\sim}{x}_{j-1} + \underset{\sim}{x}_{j+1}) + \underset{\sim}{x}_{j+2} \quad .$$

By an inductive argument, it is possible to show that

$$\underset{\sim}{p}_j^{(r)} = \underset{\sim}{x}_j + (-1)^{r+1}\,S^{(r)}\left\{\sum_{k=1}^{2^{r-1}} (\underset{\sim}{x}_{j-(2k-1)} + \underset{\sim}{x}_{j+(2k-1)})\right\} \tag{11.12}$$

and

$$\underset{\sim}{q}_j^{(r)} = \underset{\sim}{x}_{j-2^r} + (-1)^r\,S^{(r)}\,A^{(r)}\left\{\sum_{k=1}^{2^{r-1}} (\underset{\sim}{x}_{j-(2k-1)} + \underset{\sim}{x}_{j+(2k-1)})\right\} + \underset{\sim}{x}_{j+2^r}\,, \tag{11.13}$$

where

$$S^{(r)} = (A^{(r-1)}\,A^{(r-2)} \ldots A^{(0)})^{-1} \quad .$$

12. Applications of the Buneman algorithm to Poisson's equation

As was pointed out in Section 4, matrices of the form of Eqs. (2.5) arise in solving the five-point finite difference approximation to Poisson's equation over a rectangular region with Dirichlet boundary conditions; hence, it is possible to use the methods of Section 11. For the five-point approximation to Poisson's equation over a rectangular region with Neumann or periodic boundary conditions it is necessary to modify the Buneman algorithms.

For the Neumann boundary conditions, we have the system of equations

$$\begin{aligned} A\underset{\sim}{x}_0 + 2\underset{\sim}{x}_1 &= \underset{\sim}{y}_0 \\ \underset{\sim}{x}_{j-1} + A\underset{\sim}{x}_j + \underset{\sim}{x}_{j+1} &= \underset{\sim}{y}_j \qquad (j = 1,2,\ldots,m-1) , \\ 2\underset{\sim}{x}_{m-1} + A\underset{\sim}{x}_m &= \underset{\sim}{y}_m \qquad \text{with } m = 2^{k+1} . \end{aligned}$$

We define

$$\underset{\sim}{y}_j^{(1)} = A^{(1)}\underset{\sim}{p}_j^{(1)} + \underset{\sim}{q}_j^{(1)} \quad \text{for } j = 0,2,4,\ldots,2^{k+1} ,$$

where

$$\underset{\sim}{p}_0^{(1)} = A^{-1}\underset{\sim}{y}_0 , \quad \underset{\sim}{q}_0^{(1)} = 2(\underset{\sim}{y}_1 - \underset{\sim}{p}_0^{(1)}) ,$$

$$\underset{\sim}{p}_j^{(1)} = A^{-1}\underset{\sim}{y}_j , \quad \underset{\sim}{q}_j^{(1)} = \underset{\sim}{y}_{j-1} + \underset{\sim}{y}_{j+1} - 2\underset{\sim}{p}_j^{(1)} \qquad (j = 2,4,\ldots,m-2) ,$$

$$\underset{\sim}{p}_m^{(1)} = A^{-1}\underset{\sim}{y}_m , \quad \underset{\sim}{q}_m^{(1)} = 2(\underset{\sim}{y}_{m-1} - \underset{\sim}{p}_m^{(1)}) .$$

In general then, as in Section 11, we have for $r = 1,2,\ldots,k-1$

$$\underset{\sim}{y}_j^{(r+1)} = A^{(r+1)}\underset{\sim}{p}_j^{(r+1)} + \underset{\sim}{q}_j^{(r+1)} \quad ,$$

where

$$\underset{\sim}{p}_0^{(r+1)} = \underset{\sim}{p}_0^{(r)} - (A^{(r)})^{-1}(2\underset{\sim}{p}_{2^r}^{(r)} - \underset{\sim}{q}_0^{(r)}) \ , \ \underset{\sim}{q}_0^{(r+1)} = 2(\underset{\sim}{q}_{2^r}^{(r)} - \underset{\sim}{p}_0^{(r+1)})$$

$$\underset{\sim}{p}_j^{(r+1)} = \underset{\sim}{p}_j^{(r)} - (A^{(r)})^{-1}(\underset{\sim}{p}_{j-2^r}^{(r)} + \underset{\sim}{p}_{j+2^r}^{(r)} - \underset{\sim}{q}_j^{(r)}) \ , \ \underset{\sim}{q}_j^{(r+1)} = \underset{\sim}{q}_{j-2^r}^{(r)} + \underset{\sim}{q}_{j+2^r}^{(r)} - 2\underset{\sim}{p}_j^{(r+1)} \ ,$$

$$\text{for} \quad j = i2^{r+1} \quad (i = 1,2,\ldots,2^{k-r}-1)$$

$$\underset{\sim}{p}_m^{(r+1)} = \underset{\sim}{p}_m^{(r)} - (A^{(r)})^{-1}(2\underset{\sim}{p}_{m-2^r}^{(r)} - \underset{\sim}{q}_m^{(r)}) \ , \ \underset{\sim}{q}_m^{(r+1)} = 2\underset{\sim}{q}_{m-2^r}^{(r)} - 2\underset{\sim}{p}_m^{(r+1)} \quad .$$

Finally,

$$\underset{\sim}{y}_{2^k}^{(k+1)} = B^{(k+1)}\,\underset{\sim}{p}_{2^k}^{(k+1)} + \underset{\sim}{q}_{2^k}^{(k+1)} \quad , \tag{12.1}$$

where

$$B^{(k+1)} = 4I - (A^{(k)})^2$$

$$\underset{\sim}{p}_{2^k}^{(k+1)} = \underset{\sim}{p}_{2^k}^{(k)} - (A^k)^{-1}(\underset{\sim}{p}_0^{(k)} + \underset{\sim}{p}_{2^{k+1}}^{(k)} - \underset{\sim}{q}_{2^k}^{(k)}) \quad , \tag{12.2}$$

$$\underset{\sim}{q}_{2^k}^{(k+1)} = \underset{\sim}{q}_0^{(k)} + \underset{\sim}{q}_{2^{k+1}}^{(k)} - 4\underset{\sim}{p}_{2^k}^{(k+1)} \quad .$$

From Eq. (5.4), we see that

$$B^{(k+1)}\underset{\sim}{x}_{2^k} = B^{(k+1)}\underset{\sim}{p}_{2^k}^{(k+1)} + \underset{\sim}{q}_{2^k}^{(k+1)}$$

so that

$$\underset{\sim}{x}_{2^k} = \underset{\sim}{p}_{2^k}^{(k+1)} + (B^{(k+1)})^{-}\,\underset{\sim}{q}_{2^k}^{(k+1)} \quad .$$

$(B^{(k+1)})^{-}\underset{\sim}{q}_{2^k}^{(k+1)}$ indicates a solution to the singular system $B^{(k+1)}(\underset{\sim}{x}_{2^k} - \underset{\sim}{p}_{2^k}^{(k+1)}) = \underset{\sim}{q}_{2^k}^{(k+1)}$. The factorization of $B^{(k+1)}$ is given by Eq. (5.6). The back-substitution process proceeds as in Section 11. It is also possible to eliminate the $\underset{\sim}{p}_j^{(r)}$ sequence as was done in the previous section.

For periodic boundary conditions, we have the system of equations

$$\begin{aligned}
A\underset{\sim}{x}_1 + \underset{\sim}{x}_2 + \underset{\sim}{x}_m &= \underset{\sim}{y}_1 \\
\underset{\sim}{x}_{j-1} + A\underset{\sim}{x}_j + \underset{\sim}{x}_{j+1} &= \underset{\sim}{y}_j \quad \text{for} \quad j = 2,3,\ldots,m-1 \ , \\
\underset{\sim}{x}_1 + \underset{\sim}{x}_{m-1} + A\underset{\sim}{x}_m &= \underset{\sim}{y}_m \quad .
\end{aligned}$$

We define

$$\underset{\sim}{y}_j^{(1)} = A^{(1)}\underset{\sim}{p}_j^{(1)} + \underset{\sim}{q}_j^{(1)} \qquad \text{for} \quad j = 2,4,\ldots,2^{k+1} \ ,$$

where

$$\underset{\sim}{p}_2^{(1)} = A^{-1}\underset{\sim}{y}_2 \quad , \quad \underset{\sim}{q}_2^{(1)} = \underset{\sim}{y}_1 + \underset{\sim}{y}_3 - 2\underset{\sim}{p}_2^{(1)} \quad ,$$

$$\underset{\sim}{p}_j^{(1)} = A^{-1}\underset{\sim}{y}_j \quad , \quad \underset{\sim}{q}_j^{(1)} = \underset{\sim}{y}_{j-1} + \underset{\sim}{y}_{j+1} - 2\underset{\sim}{p}_j^{(1)} \quad , \quad (j = 4,6,\ldots,m-2) \ ,$$

$$\underset{\sim}{p}_j^{(1)} = A^{-1}\underset{\sim}{y}_m \quad , \quad \underset{\sim}{q}_m^{(1)} = \underset{\sim}{y}_1 + \underset{\sim}{y}_{m-1} - 2\underset{\sim}{p}_m^{(1)} \quad .$$

In general for $r = 1,2,\ldots,k-1$,

$$\underset{\sim}{y}_j^{(r+1)} = A^{(r+1)}\underset{\sim}{p}_j^{(r+1)} + \underset{\sim}{q}_j^{(r+1)} \qquad \text{for} \quad j = i2^{r+1} \quad (i = 1,2,\ldots,2^{k-r}) \ ,$$

where

$$\underset{\sim}{p}_{2^{r+1}}^{(r+1)} = \underset{\sim}{p}_{2^{r+1}}^{(r)} - (A^{(r)})^{-1}(\underset{\sim}{p}_{2^r}^{(r)} + \underset{\sim}{p}_{3\times 2^r}^{(r)} - \underset{\sim}{q}_{2^{r+1}}^{(r)}), \ \underset{\sim}{q}_{2^{r+1}}^{(r)} = \underset{\sim}{q}_{2^r}^{(r)} + \underset{\sim}{q}_{3\times 2^r}^{(r)} - 2\underset{\sim}{p}_{2^{r+1}}^{(r+1)} \ ,$$

$$\underset{\sim}{p}_j^{(r+1)} = \underset{\sim}{p}_j^{(r)} - (A^{(r)})^{-1}(\underset{\sim}{p}_{j-2^r}^{(r)} + \underset{\sim}{p}_{j+2^r}^{(r)} - \underset{\sim}{q}_j^{(r)}) \ , \quad \underset{\sim}{q}_j^{(r+1)} = \underset{\sim}{q}_{j-2^r}^{(r)} + \underset{\sim}{q}_{j+2^r}^{(r)} - 2\underset{\sim}{p}_j^{(r+1)}$$

$$\underset{\sim}{p}_m^{(r+1)} = \underset{\sim}{p}_m^{(r)} - (A^{(r)})^{-1}(\underset{\sim}{p}_{2^r}^{(r)} + \underset{\sim}{p}_{m-2^r}^{(r)} - \underset{\sim}{q}_m^{(r)}) \ , \quad \underset{\sim}{q}_m^{(r+1)} = \underset{\sim}{q}_{2^r}^{(r)} + \underset{\sim}{q}_{m-2^r}^{(r)} - 2\underset{\sim}{p}_m^{(r+1)} \ .$$

Finally as Eq. (12.1),

$$\underset{\sim}{y}_{2^k}^{(k+1)} = B^{(k+1)}\,\underset{\sim}{p}_{2^k}^{(k+1)} + \underset{\sim}{q}_{2^k}^{(k+1)} \ ,$$

where

$$\underset{\sim}{p}_{2^k}^{(k+1)} = \underset{\sim}{p}_{2^k}^{(k)} - (A^{(k)})^{-1}(2\underset{\sim}{p}_{2^{k+1}}^{(k)} - \underset{\sim}{q}_{2^k}^{(k)}) \quad , \quad \underset{\sim}{q}_{2^k}^{(k+1)} = 2\underset{\sim}{q}_{2^{k+1}}^{(k)} - 4\underset{\sim}{p}_{2^k}^{(k+1)}$$

and $B^{(k+1)}$ is defined by Eq. (12.2). Then

$$B^{(k+1)}\underset{\sim}{x}_{2^k} = B^{(k+1)}\,\underset{\sim}{p}_{2^k}^{(k+1)} + \underset{\sim}{q}_{2^k}^{(k+1)}$$

so that

$$\underset{\sim}{x}_{2^k} = \underset{\sim}{p}_{2^k}^{(k+1)} + (B^{(k+1)})^{-}\,\underset{\sim}{q}_{2^k}^{(k+1)} \quad .$$

The back-substitution process proceeds as in Section 11.

It is possible to express $\underset{\sim}{p}_j^{(r)}$ and $\underset{\sim}{q}_j^{(r)}$ in terms of $\underset{\sim}{x}_j$ as in Eqs. (11.12) and (11.13).

13. Accuracy of the Buneman Algorithms

As was shown in Section 11, the Buneman algorithms consist of generating the sequence of vectors $\{\underset{\sim}{p}_j^{(r)}, \underset{\sim}{q}_j^{(r)}\}$. Let us write, using Eqs. (11.12) and (11.13)

$$\underset{\sim}{p}_j^{(r)} = \underset{\sim}{x}_j^{(r)} + \underset{\sim}{g}_j^{(r)} \tag{13.1a}$$

$$\underset{\sim}{q}_j^{(r)} = \underset{\sim}{x}_{j-2^r} + \underset{\sim}{x}_{j+2^r} - A^{(r)} \underset{\sim}{g}_j^{(r)} \quad , \tag{13.1b}$$

where

$$\underset{\sim}{g}_j^{(r)} = (-1)^{r+1} S^{(r)} \left\{ \sum_{k=1}^{2^{r-1}} (\underset{\sim}{x}_{j-(2k-1)} + \underset{\sim}{x}_{j+(2k-1)}) \right\} \tag{13.2}$$

and

$$S^{(r)} = (A^{(r-1)} \ldots A^{(0)})^{-1} \quad . \tag{13.3}$$

Then

$$\| \underset{\sim}{p}_j^{(r)} - \underset{\sim}{x}_j^{(r)} \|_2 \leq \| S^{(r)} \|_2 \, \| \underset{\sim}{x} \|' \tag{13.4}$$

$$\| \underset{\sim}{q}_j^{(r)} - (\underset{\sim}{x}_{j-2^r} + \underset{\sim}{x}_{j+2^r}) \|_2 \leq \| S^{(r)} A^{(r)} \|_2 \, \| \underset{\sim}{x} \|' \quad , \tag{13.5}$$

where

$\| \underset{\sim}{v} \|_2$ indicates the Euclidean norm of a vector $\underset{\sim}{v}$,

$\| C \|_2$ indicates the spectral norm of a matrix C , and

$\| \underset{\sim}{x} \|' = \sum_{j=1}^{q} \| \underset{\sim}{x}_j \|_2$.

When $T = I_p$, we may redefine the polynomials given in Section 3 in the following way. Let

$$\psi = -a/2 \quad ,$$

and define

$$\psi = \cos\theta \quad \text{for} \quad |\psi| \leq 1$$

$$= \cosh\theta \quad \text{for} \quad |\psi| \geq 1 \quad .$$

Then in a similar fashion to Eq. (10.8),

$$p_{2^k}(a) = -2\cos(2^k \cos^{-1}\psi) \quad \text{for} \quad |\psi| \leq 1$$

$$= -2\cosh(2^k \cosh^{-1}\psi) \quad \text{for} \quad |\psi| \geq 1 \quad .$$

Thus for $A = A^T$,

$$\|S^{(r)}\|_2 = \prod_{j=0}^{r-1} \|(A^{(j)})^{-1}\|_2$$

$$= \prod_{j=0}^{r-1} \max_{\{\lambda_i\}} \left|[p_{2^j}(\lambda_i)]^{-1}\right| \quad ,$$

where $\{\lambda_i\}$ are the eigenvalues of A. Therefore, for $|\lambda_i| \geq 2$,

$$\|S^{(r)}\|_2 = 2^{-r} \prod_{j=0}^{r-1} \max_{\{\theta_i\}} [\cosh 2^j \theta_i]^{-1} \quad ,$$

where

$$\theta_i = \cosh^{-1}(-\lambda_i / 2) \quad .$$

Finally,

$$\|S^{(r)}\ A^{(r)}\|_2 = 2^{-r+1} \times \max_{\{\theta_i\}} \left\{ \left(\prod_{j=0}^{r-1} [\cosh 2^j \theta_i]^{-1} \right) \times \cosh 2^r \theta_i \right\}$$

when $|\lambda_i| \geq 2$.

For the five-point difference approximation to Poisson's equation over a rectangular region with Dirichlet boundary conditions

$$\lambda_i = -2(1 + \rho^2(1 - \cos \frac{i\pi}{p+1})) \quad ,$$

where $\rho = \Delta x/\Delta y$ or $(\Delta y/\Delta x)$ depending on the ordering of the equations. Thus

$$\theta_i = \cosh^{-1}(1 + \rho^2(1 - \cos \frac{i\pi}{p+1}))$$

which implies $\theta_i > 1$ for all i . Then

$$\max_{\{\theta_i\}} [\cosh 2^j \theta_i]^{-1} = [\cosh 2^j \{\cosh^{-1}(1 + \rho^2(1 - \cos \frac{\pi}{p+1}))\}]^{-1} .$$

Thus after some simplification,

$$\|S^{(r)}\|_2 = \frac{e^{-c\theta_1}}{\prod_{j=0}^{r-1} (1 + e^{-2^{j+1}\theta_1})} < e^{-c\theta_1} , \qquad (13.6)$$

where $c = 2^r - 1$, $\cosh \theta_1 = 1 + \rho^2(1 - \cos \frac{\pi}{p})$.

A similar calculation shows

$$\|A^{(r)} S^{(r)}\|_2 < 2 e^{\theta_p} . \qquad (13.7)$$

Therefore from Eq. (13.6) we see that for large r, $\underset{\sim}{p}_j^{(r)}$ will be a good approximation to $\underset{\sim}{x}_j$. And from Eqs. (13.5) and (13.7), we see that

$$\|\underset{\sim}{q}_j^{(r)} - (\underset{\sim}{x}_{j-2^r} + \underset{\sim}{x}_{j+2^r})\|_2 \leq 2 e^{\theta_p} \|\underset{\sim}{x}\|' ,$$

so that the $\|\underset{\sim}{q}_j^{(r)}\|_2$ remains bounded throughout the calculation. This explains why the Buneman algorithms lead to numerically stable results for solving the finite difference approximation to Poisson's equation.

14. Conclusion

Numerous applications require the repeated solution of a Poisson equation. The operation counts given by Dorr[5] indicate that the methods we have discussed should offer significant economies over older techniques; and this has been verified in practice by many users. Computational experiments comparing the Buneman algorithm (variant one), the MD algorithm, the Peaceman-Rachford alternating direction algorithm, and the point successive over-relaxation algorithm are given by Buzbee, et al.[3] We conclude that the method of matrix decomposition, the Buneman algorithms, and Hockney's algorithm (when used with care) are valuable methods.

This paper has benefited greatly from the comments of Dr. F. Dorr, Mr. J. Alan George, Dr. R. Hockney, and Prof. O. Widlund.

REFERENCES

1. Richard Bellman, Introduction to Matrix Analysis, McGraw-Hill, New York, 1960.

2. Oscar Buneman, Stanford University Institute for Plasma Research, Report No. 294, 1969.

3. B. L. Buzbee, G. H. Golub, and C. W. Nielson, "The Method of Odd/Even Reduction and Factorization with Application to Poisson's Equation, Part II," LA-4288, Los Alamos Scientific Laboratory.

4. J. W. Cooley and J. W. Tukey, "An algorithm for machine calculation of complex Fourier series," Math. Comp., Vol. 19, No. 90 (1965), pp. 297-301.

5. F. W. Dorr, "The direct solution of the discrete Poisson equation on a rectangle," to appear in SIAM Review.

6. J. A. George, "An Embedding Approach to the Solution of Poisson's Equation on an Arbitrary Bounded Region," to appear as a Stanford Report.

7. G. H. Golub, R. Underwood, and J. Wilkinson, "Solution of $A\underset{\sim}{x} = \lambda B\underset{\sim}{x}$ when B is positive definite," (to be published).

8. R. W. Hockney, "A fast direct solution of Poisson's equation using Fourier analysis," J. ACM., Vol. 12, No. 1 (1965), pp. 95-113.

9. R. W. Hockney, in Methods in Computational Physics (B. Adler, S. Fernbach and M. Rotenberg, Eds.), Vol. 9, Academic Press, New York and London, 1969.

10. R. E. Lynch, J. R. Rice, and D. H. Thomas, "Direct solution of partial difference equations by tensor product methods," Num. Math., Vol. 6 (1964), pp. 185-199.

11. H. Shintani, "Direct solution of partial difference equations for a rectangle," J. Sci. Hiroshima Univ. Ser. A-I, Vol. 32 (1968), pp. 17-53.

12. R. S. Varga, Matrix Interative Analysis, Prentice Hall, New York, 1962.