THE METHOD OF ODD/EVEN REDUCTION AND FACTORIZATION WITH APPLICATION TO POISSON'S EQUATION, PART II

BY

B. L. BUZBEE

G. H. GOLUB

C. W. NIELSON

STAN-CS-70-155
MARCH, 1970

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

THE METHOD OF ODD/EVEN REDUCTION AND FACTORIZATION WITH APPLICATION TO POISSON'S EQUATION, PART II*

by

B. L. Buzbee+, G. H. Golub, C. W. Nielson+

To be issued simultaneously as Los Alamos Scientific Laboratory Report LA 4288.

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

+ Los Alamos Scientific Laboratory, University of California.

The preparation of this manuscript was supported in part by the Office of Naval research (NR 044 211), the National Science Foundation (GJ 798), and the Atomic Energy Commission (Stanford PA #18).

Abstract

In this paper, we derive and generalize the methods of Buneman for solving elliptic partial difference equations in a rectangular region. We show why the Buneman methods lead to numerically accurate solutions whereas the CORF algorithm may be numerically unstable. Several numerical examples are given and discussed.

Introduction

In the first part of this report, we described several direct methods for solving linear equations arising from elliptic partial difference equations. In this part, we develop the Buneman algorithms which are closely related to the Cyclic Odd/Even Reduction and Factorization (CORF) algorithm which was derived in the first part. We then show why the CORF algorithm is numerically unstable whereas the Buneman algorithms yield numerically accurate results. Finally, we describe some numerical examples and compare the time and accuracy of several methods for solving them.

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:

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

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

so that

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

Because of roundoff error, one actually computes the sequence

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

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

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 (10.3) into (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

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

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

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

The solution of (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

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

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

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

Let

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

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

Then using the initial conditions (10.4a), we may write (10.6a) as follows:

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

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

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

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

Note that

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

given in Section 3. Thus

$$\tilde{\eta}_s = P_s(A,T)\underset{\sim}{y}_j^{(r)} + \sum_{k=0}^{s-1} Q\, S_{s-k}\, Q^T \underset{\sim}{\delta}_k \tag{10.9}$$

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 roundoff error can be catastrophic. However, if $|\lambda_j/2\omega_j| \leq 1$, we see from (10.9) that $\tilde{\underset{\sim}{\eta}}_{2^r}$ may be a good numerical approximation to $A^{(r)}\underset{\sim}{y}_j^{(r)}$.

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

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

depending on how one orders the equations. By inspection

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

for all j ; and hence for large s , equation (10.1) leads to a numerically unstable algorithm. A similar result holds for the nine point difference approximation to Poisson's equation. Using the five point approximation with uniform mesh and any number of grid points, equation (10.9) predicts severe loss of accuracy for more than five contractions on a CDC 6600; and this has actually been observed. As noted in Section 3, Hockney [6, 7] has combined one or more steps of CORF with the fast Fourier transform to produce a Poisson solver. For such a use of CORF one must pay careful attention to the above results.

The cyclic odd/even reduction method can be used successfully for solving tridiagonal systems of equations. In that situation, one must make provision for the fact that overflow can occur during the reduction stages.

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 equations (2.5) $T = I_p$, the identity matrix of order p .

Again consider the system of equations as given by (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 (11.1) may be written as follows:

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

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 (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 (11.3),

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

Substituting (11.5) into (11.4) and making use of the identity $(A^{(r)})^2 = 2I_P - A^{(r+1)}$ from (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 (11.6a) we solve the system of equations

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

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

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

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

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

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

For $j = 2^r, 3 \cdot 2r, \ldots, 2k+l-2r$, 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

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

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

1) Compute the sequence $\{\underset{\sim}{p}_j^{(r)}, \underset{\sim}{q}_j^{(r)}\}$ by (11.6) for $r = 1,\ldots,k$ with $\underset{\sim}{p}_j^{(0)} = \underline{0}$ 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) Backsolve for $\underset{\sim}{x}_j$ using (11.7) and (11.8).

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

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

where

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

Using (11.9) in (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

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

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

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

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

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

$$- \left(A^{(r)}\right)^{-1}\left(\underset{\sim}{x}_{j-2h} + \underset{\sim}{x}_{j+2h} - \underset{\sim}{q}_j^{(r)}\right) . \qquad (111.11)$$

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

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

2) Backsolve for $\underset{\sim}{x}_j$ using (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 p_j's and 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

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

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

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 (2.5) arise in solving the five point finite difference approximation to Poisson's equation over a rectangular region with Dirichlet boundary conditions and 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

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

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

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

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

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

where

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

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

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

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

Finally,

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

where

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

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

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

From (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

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

$(B^{(k+1)})^{-}\underset{\sim}{q}_{3^k}^{(k+1)}$ 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 (5.6). The backsubstitution 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

$$A\underset{\sim}{x}_1 + + \underset{\sim}{x}_2 + \underset{\sim}{x}_m = \underset{\sim}{y}_1$$

$$\underset{\sim}{x}_{j-1} + A\underset{\sim}{x}_j + \underset{\sim}{x}_{j+1} = \underset{\sim}{y}_j \quad \text{for } j = 2,3,\ldots,m-1 ,$$

$$\underset{\sim}{x}_1 + \underset{\sim}{x}_{m-1} + A\underset{\sim}{x}_m = \underset{\sim}{y}_m .$$

We define

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

where

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

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

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

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

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

where

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

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

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

Finally as (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 (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 backsubstitution 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 equations (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 (11.12) and (11.13)

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

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

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

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

Then

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

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

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

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

and define

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

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

Then in a similar fashion to (10.8),

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

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

Thus for $A = A^T$,

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

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

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

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

where

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

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

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

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

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

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

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

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

A similar calculation shows

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

Therefore from (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 (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. Conclusions

The Appendix contains the results of some numerical experiments involving the application of the Buneman algorithm (variant 1), the method of matrix decomposition, the method of point successive over-relaxation (cf. [10]), and the Peaceman-Rachford alternating direction method (cf. [11]) to the five point finite difference approximation to Laplace's equation over a rectangle with Dirichlet boundary conditions. In these experiments the Buneman algorithm was the most efficient and accurate; however, the method of matrix decomposition was competitive in several cases. We conclude, therefore, that the Buneman algorithm and the method of matrix decomposition are useful methods in the situations where they apply.

Acknowledgment

The authors are very pleased to thank Mr. J. Alan George for his many helpful suggestions.

Appendix

Numerical Experiments

In order to gain computational experience with the methods of matrix decomposition (MD) and the Buneman algorithm (variant 1), it was decided to apply the algorithms to the five point difference approximation of Laplace's equation with Dirichlet boundary conditions. In addition, in order to compare these methods with established methods, it was decided to apply the methods of point successive over-relaxation (SOR) (cf. [10]) and Peaceman-Rachford alternating direction method (PR) (cf. [11]) to the same problems. We did not attempt to-determine which method is best in general. Those interested in operation counts, variations of these direct procedures, and customizing the direct procedures for particular problems are referred to [4] and [7].

The following problems were chosen so that the computed error could be detemined exactly:

Problem #1, $u = 1$;

Problem #2, $u = \cos(x)\cosh(y)$;

Problem #3, $u = e^{x}(\sin(y) + \cos(y))$;

Problem #4, $u = x^{5} - 10x^{3}y^{2} + 5xy^{4}$.

Let

$\bar{u}$ = <computed solution of the difference equation>

and

$$d = \max_{(\text{mesh})}\{|\bar{u}|, 1.0\} ;$$

the tabulated error is

$$\max_{(\text{mesh})} \left|\frac{\bar{u} - u}{d}\right| .$$

One should note that in many cases the tabulated error is the truncation error of the difference equation.

Each of these problems was solved on the following meshes (includes boundary points):

Mesh #1 20 by 129 ,

Mesh #2 40 by 129 ,

Mesh #3 80 by 129 ,

Mesh #4 129 by 129 ,

Let $\rho = \Delta x/\Delta y$. Each of the four problems was solved on each of the four meshes for five values of ρ :

i	Δx	Δy	ρ_i
1	.025	.00025	.01
2	.025	.0025	.1
3	.025	.025	1.0
4	.0025	.025	10.0
5	.00025	.025	100.0

Thus each problem was solved on a total of twenty rectangular regions These regions were chosen such that the lower left-hand corner of the rectangle was always at the origin. The following is a table of the coordinates of the upper right-hand corners:

	Mesh #1	Mesh #2	Mesh #3	Mesh #4
ρ_1	(.5, .032)	(1., .032)	(2., .032)	(4., .032)
ρ_2	(.5, .322)	(1., .322)	(2., .322)	(4., .322)
ρ_3	(.5, 3.225)	(1., 3.225)	(2., 3.225)	(4., 3.225)
ρ_4	(.05, 3.225)	(.1, 3.225)	(.2, 3.225)	(.4, 3.225)
ρ_5	(.005, 3.225)	(.01, 3.225)	(.02, 3.225)	(.04, 3.225)

We define

$V(p,i,j)$ = max {solution of prob #p on region with ρ_i and mesh #j}

- min [solution of prob #p on region with ρ_i and mesh #j}

Note $V(1,r,j) = 0$ for all i and j . The following tables give V for the other problems:

$V(2,i,j)$

	Mesh #1	Mesh #2	Mesh #3	Mesh #4
ρ_1	.1	.42	1.37	2.0
ρ_2	.15	.47	1.44	2.1
ρ_3	11.1	11.4	16.4	24.0
ρ_4	11.0	11.0	11.	11.
ρ_5	11.0	11.0	11.	11.

$V(3,i,j)$

	Mesh #1	Mesh #2	Mesh #3	Mesh #4
ρ_1	.59	1.64	6.22	23.6
ρ_2	.95	2.24	7.84	29.2
ρ_3	3.84	6.32	17.2	58.5
ρ_4	2.56	2.7	2.97	3.36
ρ_5	2.46	2.47	2.5	2.53

V(4,i,j)

	Mesh #1	Mesh #2	Mesh #3	Mesh #4
ρ_1	.018	.77	28.2	323.0
ρ_2	.07	.10	28.3	323.0
ρ_3	219.5	400.0	556.0	1733.0
ρ_4	22.9	48.2	98.2	158.0
ρ_5	2.29	4.82	9.9	16.1

For the above rectangles, the optimum relaxation factor is given by

$$\omega_b(i,j) = \frac{2}{1 + \sqrt{1 - B_{ij}^2}}$$

where

$$B_{ij} = \frac{\rho_i^2 \cos\left(\frac{\pi}{N_j - 1}\right) + \cos\left(\frac{\pi}{128}\right)}{2(\rho_i^2 + 1)}$$

and N_j is the number of grid points in the x-direction of the j-th mesh.

The initial guess for SOR and PR was the zero vector.

The iteration process was terminated when

$$\max_{\substack{\text{(entire mesh)} \\ |U^n| > 10^{-5}}} \left| \frac{U^{n+1} - U^n}{U^n} \right| < 10^{-4}$$

Optimum PR parameters were determined by Wachspress's algorithm [11]

for cycles of length 2^k . Convergence required

$$\max_{\text{(complete cycle)}} \left\{ \max_{\substack{\text{(entire mesh)} \\ |U^n| > 10^{-5}}} \left| \frac{U^{n+1} - U^n}{U^n} \right| \right\} < 10^{-4} \quad .$$

Because of this convergence criterion, a short cycle was desirable. After some experimentation, it was decided to use a cycle length of four exclusively.

All problems were run on a CDC 6600 (about 14 decimal digits of accuracy); -the RR, MD, and Buneman programs all used the same tridiagonal system solver. The Q matrix and eigenvalues required by MD were computed with the QR algorithm for symmetric tridiagonal matrices. The matrix multiplications $(Q^T y)$ required by MD were performed with a machine language inner product routine which is quite efficient and which accumulates the inner products in double precision. It should be noted that for problems with uniform mesh spacing, these matrix products may be performed with the fast Fourier transform; and this makes MD competitive in speed with the Buneman algorithm. However, MD is capable of handling more general problems such as those with non-uniform mesh spacings; in these cases Q must be computed and the matrix products performed. Thus the MD routine used in this study gives an indication of the kind of performance one might expect with these more general problems. Note also that the matrix multiplication $(Q^T y)$ requires $0(qp^2)$ operations. The Buneman algorithm requires a total of $0(qp \log_2 p)$ operations. Thus as p becomes small, MD approaches the Buneman algorithm in speed.

The following tables of computation times are normalized by the time required for the Buneman algorithm on Mesh #1:

Computation times for the Buneman algorithm (variant 1) and MD

	Mesh #1	Mesh #2	Mesh #3	Mesh #4
Buneman	1.0	2.08	4.31	6.96
MD	1.18	3.65	14.9	41.0

Computation time for PR

(These times are averages over all four problems.)

	Mesh #1	Mesh #2	Mesh #3	Mesh #4
ρ_1	2.56	5.17	9.43	15.2
ρ_2	4.85	10.3	20.6	30.1
ρ_3	5.44	12.6	31.2	47.7
ρ_4	2.56	7.61	15.2	32.5
ρ_5	2.25	4.49	8.58	13.5

Computation time for SOR

(These times are averages over all four problems.)

	Mesh #1	Mesh #2	Mesh #3	Mesh #4
ρ_1	40.9	89.0		
ρ_2	45.1	97.9		
ρ_3	16.1	60.1	(not run)	(not run)
ρ_4	7.58	32.9		
ρ_5	7.22	28.8		

Relative error for the Buneman algorithm, Mesh #1

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	4(-11)	7(-9)	6(-9)	3(-7)
ρ_2	2(-11)	5(-7)	4(-7)	2(-5)
ρ_3	5(-13)	2(-6)	2(-6)	4(-7)
ρ_4	2(-13)	1(-8)	1(-8)	2(-9)
ρ_5	2(-13)	1(-10)	1(-10)	2(-11)

Relative error for MD, Mesh #1

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	5(-7)	5(-7)	4(-7)	3(-7)
ρ_2	1(-7)	5(-7)	4(-7)	2(-5)
ρ_3	1(-9)	2(-6)	2(-6)	4(-7)
ρ_4	2(-10)	1(-8)	1(-8)	2(-9)
ρ_5	8(-10)	7(-10)	7(-10)	6(-10)

Relative error for PR, Mesh #1

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	2(-11)	7(-9)	6(-9)	3(-7)
ρ_2	2(-8)	5(-7)	4(-7)	2(-5)
ρ_3	7(-8)	2(-6)	2(-6)	4(-7)
ρ_4	2(-9)	1(-8)	1(-8)	2(-9)
ρ_5	8(-13)	1(-10)	1(-10)	2(-11)

Relative error for SOR, Mesh #1

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	4(-4)	4(-4)	4(-4)	3(-6)
ρ_2	1(-3)	1(-3)	1(-3)	2(-5)
ρ_3	5(-4)	1(-4)	1(-4)	4(-7)
ρ_4	3(-4)	3(-4)	3(-4)	9(-6)
ρ_5	3(-4)	3(-4)	3(-4)	5(-5)

Relative error for the Buneman algorithm, Mesh #2

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	4(-11)	7(-9)	6(-9)	7(-7)
ρ_2	3(-11)	6(-7)	4(-7)	5(-5)
ρ_3	2(-12)	5(-6)	7(-6)	2(-6)
ρ_4	3(-13)	5(-8)	6(-8)	8(-9)
ρ_5	7(-13)	6(-10)	6(-10)	8(-11)

Relative error for MD, Mesh #2

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	1(-7)	8(-8)	1(-7)	8(-7)
ρ_2	5(-8)	6(-7)	5(-7)	5(-5)
ρ_3	2(-9)	5(-6)	7(-6)	2(-6)
ρ_4	2(-9)	6(-8)	6(-8)	8(-9)
ρ_5	5(-10)	1(-9)	1(-9)	3(-10)

Relative error for PR, Mesh #2

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	2(-11)	7(-9)	7(-9)	7(-7)
ρ_2	4(-8)	6(-7)	4(-7)	5(-4)
ρ_3	4(-6)	4(-6)	7(-6)	2(-6)
ρ_4	8(-10)	6(-8)	6(-8)	8(-9)
ρ_5	3(-12)	6(-10)	6(-10)	7(-6)

Relative error for SOR, Mesh #2

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	4(-4)	4(-4)	3(-4)	2(-4)
ρ_2	7(-4)	5(-4)	3(-4)	5(-5)
ρ_3	1(-3)	6(-4)	2(-4)	2(-6)
ρ_4	5(-4)	4(-4)	6(-5)	7(-7)
ρ_5	4(-4)	4(-4)	4(-4)	8(-5)

Relative error for the Buneman algorithm, Mesh #3

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	4(-11)	7(-9)	6(-9)	5(-8)
ρ_2	3(-11)	6(-7)	4(-7)	5(-6)
ρ_3	1(-11)	8(-6)	1(-5)	1(-5)
ρ_4	4(-13)	2(-7)	2(-7)	3(-8)
ρ_5	2(-12)	2(-9)	3(-9)	3(-9)

Relative error for MD, Mesh #3

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	1(-9)	8(-9)	7(-9)	5(-8)
ρ_2	1(-9)	6(-7)	4(-7)	5(-6)
ρ_3	8(-10)	8(-6)	1(-5)	1(-5)
ρ_4	8(-10)	2(-7)	2(-7)	3(-8)
ρ_5	8(-10)	3(-9)	3(-9)	7(-10)

Relative error for PR, Mesh #3

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	1(-11)	7(-9)	6(-9)	5(-8)
ρ_2	6(-8)	6(-7)	4(-7)	4(-6)
ρ_3	3(-6)	8(-6)	1(-5)	1(-5)
ρ_4	2(-7)	3(-7)	2(-7)	3(-8)
ρ_5	1(-11)	3(-9)	3(-9)	3(-10)

Relative error for the Buneman algorithm, Mesh #4

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	4(-11)	7(-9)	6(-9)	8(-9)
ρ_2	3(-11)	6(-7)	4(-7)	7(-7)
ρ_3	3(-11)	8(-6)	2(-5)	1(-5)
ρ_4	1(-12)	5(-7)	6(-7)	8(-8)
ρ_5	4(-12)	6(-9)	7(-9)	8(-10)

Relative error for MD, Mesh #4

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	7(-7)	5(-7)	4(-7)	4(-7)
ρ_2	2(-9)	6(-7)	4(-7)	7(-7)
ρ_3	2(-9)	8(-6)	2(-5)	1(-5)
ρ_4	3(-9)	4(-7)	6(-7)	8(-8)
ρ_5	3(-9)	9(-9)	9(-9)	2(-9)

Relative error for PR, Mesh #4

	Prob 1	Prob 2	Prob 3	Prob 4
ρ_1	1(-11)	7(-9)	6(-9)	8(-9)
ρ_2	6(-8)	6(-7)	4(-7)	7(-7)
ρ_3	1(-5)	8(-6)	2(-5)	1(-5)
ρ_4	6(-8)	4(-7)	6(-7)	8(-8)
ρ_5	3(-11)	6(-9)	7(-9)	8(-10)

References

[ll] Wachspress, E. L., Iterative Solution of Elliptic Systems.
Prentice Hall, Englewood Cliffs, N. J., Chapter 6.