December 6, 1973

To the Graduate Council:

I am submitting herewith a thesis written by Eayo Ouseph entitled "A Comparison of Certain Algorithms for Determining the Chebyshev Solution of an Overdetermined System of Equations." I recommend that it be accepted for nine quarter hours of credit in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Randall E Cline
Major Professor

We have read this thesis and recommend its acceptance:

B.C. Gonzalez

Robert J. Plemmons

Accepted for the Council:

Vice Chancellor for
Graduate Studies and Research

A COMPARISON OF CERTAIN ALGORITHMS FOR DETERMINING THE CHEBYSHEV SOLUTION OF AN OVERDETERMINED SYSTEM OF EQUATIONS

A Thesis

Presented to

the Graduate Council of

The University of Tennessee

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Eayo Ouseph

March 1974

ACKNOWLEDGMENTS

The author wishes to express his great appreciation to Professor Randal E. Cline for suggesting this problem and for his patient guidance in the preparation of this paper. His systematic correction and competent criticism have been most helpful. The author would also like to express his sincere appreciation to Professor Robert J. Plemmons and to Professor Ralph C. Gonzales for their services on the Master's degree committee.

1136723

ABSTRACT

The purpose of this thesis is to compare the Golub-Bartells algorithm and the Meicler algorithm for determining the Chebyshev solution of an overdetermined system of equations. Computer programs for both the algorithms are included.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
II.	THE GOLUB-BARTELLS ALGORITHM	5
III.	THE MEICLER ALGORITHM	15
IV.	NUMERICAL RESULTS AND CONCLUSIONS	23
LIST OF REFERENCES		29
APPENDIXES		31
A.	COMPUTER PROGRAM FOR GOLUB-BARTELLS ALGORITHM	32
B.	COMPUTER PROGRAM FOR MEICLER ALGORITHM	44
C.	NUMERICAL EXAMPLES	54
VITA		67

CHAPTER I

INTRODUCTION

Given a linear algebraic system of equations $Ax = b$, it is well known that there is either no exact solution, a unique solution or infinitely many solutions. When there is not exact solution, that is, the system of equations is inconsistent, one frequently wishes to determine an approximate solution which is "best" in some sense. In such a case, the criterion generally employed is to require that the approximate solution minimize the residual in some l_p norm, where for any positive number p, $1 \leq p \leq \infty$, the l_p norm of a vector y is defined as

$$||y||_p = (\Sigma|y_i|^p)^{1/p} \quad .$$

Then for any l_p norm, a best approximate solution $x = \hat{x}$, say, of $Ax = b$ is an approximate solution which minimizes the norm $||A\hat{x} - b||_p$.

Throughout the literature, the l_p norms most frequently considered are when $p = 1$, 2, and ∞. Although there appears to be no uniform terminology for $p = 1$, the case $p = 2$ is called the least squares solution and $p = \infty$ is called the Chebyshev or minimax solution since $||x||_\infty = \text{Max } |x_i|$.

It is the purpose of this thesis to compare a number of techniques for determining a Chebyshev solution of an inconsistent system of equations $Ax = b$. In particular, the method recently suggested by Bartells and Golub [1] is to be compared with the method suggested by Meicler [2].

Chapter II contains a summary of the mathematics and the computational method used in the Steifel exchange algorithm and in the modification proposed by Bartells and Golub. In Chapter III the Meicler algorithm is developed in detail. Finally in Chapter IV, these methods are compared in terms of computational stability, time of execution, storage requirements, etc.

Throughout this paper an overdetermined system of equations is denoted by

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i = 1, 2, \ldots, m \qquad (1)$$

where n is the number of unknowns and m ($m > n$) is the number of equations. For these considerations, the elements a_{ij} ($i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$) and b_i ($i = 1, 2, \ldots, m$) are assumed to be real numbers. When (1) is written in matrix notation as $Ax = b$ with $A = (a_{ij})$, columns of A will be denoted by $A^{(j)}$ ($j = 1, \ldots, n$). Also, to satisfy the requirements necessary for each algorithm to

be considered in this paper, rows of A will be assumed to satisfy the Haar Condition, that is, each nxn submatrix of A is nonsingular. Corresponding to any real inconsistent system of equations Ax = b there exists one or more vectors $\hat{x} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n)^T$ such that:

$$\underset{1\leq i\leq m}{\text{Max}} \left| \sum_{j=1}^{n} a_{ij}\hat{x}_j - b_i \right| \leq \underset{1\leq i\leq m}{\text{Max}} \left| \sum_{j=1}^{n} a_{ij}x_j - b_i \right|$$

for all $x\epsilon E^n$ (real n - dimensional Euclidean space). Such a vector $\hat{x}$ is called the Chebyshev (or minimax) solution.

For the system of equations (1) we define the residual for any row as

$$r_i(x) = \sum_{j=1}^{n} a_{ij}x_j - b_i$$

Also, any n + 1 by n subsystem of (1) of the line form

$$\begin{bmatrix} A_{i_1} \\ \cdot \\ \cdot \\ \cdot \\ A_{i_{n+1}} \end{bmatrix} \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} b_{i_1} \\ \cdot \\ \cdot \\ \cdot \\ b_{i_{n+1}} \end{bmatrix}$$

where, by assumption, the rank of the matrix is n, is called a reference subsystem and the row vectors $A_{i_1}, \ldots, A_{i_{n+1}}$ are called the reference set. Now, if $x = (x_1 \ \ldots \ x_n)^T$

is a Chebyshev solution for this subsystem, then

$$|\varepsilon| = |r_{i_1}(x)| = \ldots |r_{i_{n+1}}(x)| = \inf_{y \in E^n} \max_{1 \le j \le m} |r_{ij}(y)|$$

is called the reference deviation.

For any scalar x, the function Sgn(x) is defined as

$$\mathrm{Sgn}(x) = \begin{cases} 1, & \text{if } x > 0 , \\ 0, & \text{if } x = 0 , \\ -1, & \text{if } x < 0 . \end{cases}$$

Finally, the Moore-Penrose generalized inverse of any matrix A, is written as A^+, where A^+ is the unique solution to the equations,

$$AA^+A = A ,$$
$$A^+AA^+ = A^+ ,$$
$$(A^+A)^* = AA^+$$

and

$$(A^+A)^* = A^+A ,$$

where A^* denotes the conjugate transpose of A.

CHAPTER II

THE GOLUB-BARTELLS ALGORITHM

To develop the methods to be explored in this thesis, we state, without formal proofs, a number of results (see, for example, Cheney [3], Chapter 2).

Theorem 2.1

The system of linear equations (1) is consistent if and only if b lies in the linear span of vectors $A^{(j)}$. If the system is consistent, it has a unique solution if and only if the $A^{(j)}$ are linearly independent.

This theorem states that the linear independence among the columns of A implies the linear independence among the b_i. In other words, for any m - tuple c, $\sum_{i=1}^{m} c_i A^{(i)} = 0$ implies $\sum_{i=1}^{m} c_i b_i = 0$, if the system $Ax = b$ is consistent.

To characterize the minimax solutions to (1), we consider the system of linear equalities,

$$r_i(x) = \sum_{j=1}^{n} a_{ij}x_j - b_i \quad (i = 1, 2, \ldots, m) \tag{2}$$

Let $v = (v_1, \ldots, v_n)$ be an n - tuple. Without loss of generality, let us assume that the vectors r_i have been ordered so that $\max_{1 \le i \le m} |r_i(v)| = |r_1(v)|$ and $|r_1(v)| \ge |r_2(v)| \ge \ldots$

$\geq |r_m(v)|$. Under these assumptions, the following theorem characterizes the minimax solutions to (1).

Theorem 2.2 (Chacterization Theorem)

The following statements are equivalent.

1. There is no vector z for which

$$\max_{1\leq i\leq m} |r_i(z)| < \max_{1\leq i\leq m} |r_i(v)|$$

2. $$\sum_{i=1}^{k} u_i \operatorname{Sgn}(r_i(v))A_i = 0$$

The characteristic theorem can be used to establish an essential criterion for the minimax solution.

Theorem 2.3

Every minimax solution of the system (1) is a minimax solution of an appropriate subsystem consisting of n + 1 equations.

The algorithms to be considered herein for finding a minimax solution reduce to a search for an appropriate reference deviation of the entire system of equations in (1). In the view of the above theorem, the case where m = n + 1 is of special interest.

To begin, we assume, without loss of generality, that the equations have been ordered so that the first n rows,

$A_1, \ldots, A_n$ are linearly independent. Thus scalars $\lambda_1, \ldots, \lambda_{n+1}$ can be found with $\lambda_{n+1} \neq 0$ such that

$$\sum_{i=1}^{n+1} \lambda_i A_i = 0 \quad .$$

Let $\mathrm{Sgn}(\lambda_i) = S_i$ for $i = 1, \ldots n+1$ and calculate

$$\varepsilon = - \sum_{j=1}^{n+1} \lambda_i b_i \Big/ \sum_{j=1}^{n+1} |\lambda_j|$$

Thus we can solve the system of equations

$$\begin{bmatrix} A_1 \\ \cdot \\ \cdot \\ \cdot \\ A_n \end{bmatrix} \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{bmatrix} + \varepsilon \begin{bmatrix} S_1 \\ \cdot \\ \cdot \\ \cdot \\ S_n \end{bmatrix}$$

To see that $x = (x_1, \ldots, x_n)^T$ is, in fact, the minimax solution to the system of $n + 1$ equations, we first note that

$$r_i(x) = \sum_{j=1}^{n} a_{ij} x_j - b_i = S_i \varepsilon, \quad \text{for } i = 1, 2, \ldots, n$$

and

$$r_{n+1}(x) = S_{n+1} \varepsilon \quad .$$

But from the linear dependence of the A_i,

$$0 = \sum_{i=1}^{n+1} \lambda_i A_i = \sum_{i=1}^{n+1} |\lambda_i| \, S_i A_i \quad ,$$

and so

$$\begin{aligned} 0 &= \mathrm{Sgn}(\varepsilon) \cdot 0 \\ &= \sum_{i=1}^{n+1} |\lambda_i| \, S_i \, \mathrm{Sgn}(\varepsilon) \, A_i \\ &= \sum_{i=1}^{n+1} |\lambda_i| \, \mathrm{Sgn}\,(r_i(x)) \, A_i \end{aligned}$$

Thus, by the Characterization Theorem 2.2, we have that $x = (x_1, \ldots, x_n)^T$ is the Chebyshev solution to the system of n + 1 equations.

Returning to the general case where $m > n + 1$, we give a summary of the Steifel exchange algorithm as modified by Bartells and Golub. (The Steifel algorithm is discussed in detail in Cheney [3]).

To begin, we start with a subsystem of n + 1 equations and modify it by row interchanges so that the reference derivation is increased at each step. Let $A_1, \ldots, A_{n+1}$ be a reference set and $x = (x_1, \ldots, x_n)^T$ be the Chebyshev solution to the corresponding reference subsystem. Then scalars $\varepsilon, \lambda_1, \ldots, \lambda_{n+1}$ are computed, as described above, such that

$$\text{(a)} \quad \sum_{i=1}^{n+1} \lambda_i A_i = 0 \quad ,$$

$$\text{(b)} \quad \varepsilon = - \sum_{i=1}^{n+1} \lambda_i b_i \Bigg/ \sum_{j=1}^{n+1} |\lambda_j|$$

and the residuals

$$\text{(c)} \quad r_i(x) = S_i \varepsilon \quad ,$$

where $S_i = \text{Sgn}(\lambda_i)$ for $i = 1, 2, \ldots, n+1$. Obviously, x may not be the Chebyshev solution to the entire system since there may exist an $\alpha \varepsilon \{n+2, \ldots, m\}$ for which $|r_\alpha(x)| > |\varepsilon|$. In this case, we proceed as follows to replace one row of the reference subsystem by some row A_α. Note first that row A_α can be expressed as

$$A_\alpha = \sum_{i=1}^{n+1} \rho_i A_i$$

where $\rho_1, \ldots, \rho_{n+1}$ are scalars. Now to determine which row should be deleted from the reference subsystem, we impose two conditions to ensure that the exchange produces a greater reference deviation. First assume that $\lambda_i \neq 0$ for all $i = 1, 2, \ldots, n+1$. Now let $\beta \varepsilon \{1, 2, \ldots, n+1\}$ be any index for which

$$\frac{\sigma_\alpha S \rho_\beta}{\lambda_\beta} = \underset{1 \le i \le n+1}{\text{Max}} \frac{\sigma_\alpha S \rho_i}{\lambda_i}$$

where $\sigma_\alpha = \text{Sgn}(\sigma_\alpha(s))$ and $S = \text{Sgn}(\varepsilon)$, and assume that $(A_1, \ldots, A_{\beta-1} A_\alpha A_{\beta+1}, \ldots, A_{n+1})$ is the reference set. It will be shown now that with this set we can move in the direction of increasing ε. Suppose we form the solution $x' = (x_1', \ldots, x_n')$ of the subsystem

$$\begin{bmatrix} A_1 \\ \vdots \\ A_{\beta-1} \\ A'_{\beta+1} \\ \vdots \\ A_{n+1} \\ A_\alpha \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_{\beta-1} \\ b_{\beta+1} \\ \vdots \\ b_\alpha \end{bmatrix}$$

Thus scalars ε', λ_1', $\ldots$, λ_{n+1}' are computed as described above such that:

$$\text{(a)} \qquad \sum_{i=1, i \neq \beta}^{n+1} \lambda_i' A_i + \lambda_\alpha' A_\alpha = 0$$

$$(b) \qquad \varepsilon' \equiv \frac{\sum_{i=1}^{n+1}{}_{i\neq\beta} \lambda_i' b_i + \lambda_\alpha' b_\alpha}{|\lambda_\alpha'| + \sum_{i=1}^{n+1}{}_{i\neq\beta} |\lambda_i|}$$

and

$$(c) \qquad r_i(x') = S_i'\varepsilon' \text{ for } i = 1, \ldots, \beta-1, \beta+1, \ldots, n+1, \alpha$$

where $S_i' = \mathrm{Sgn}(\lambda_i')$.

Define

$$k = |\lambda_\alpha'| + \sum_{k+1}^{n+1}{}_{k\neq\beta} |\lambda'_k|$$

and

$$c = \frac{|\lambda'_\alpha|}{k}$$

then

$$|\varepsilon'| = c|r_\alpha(x)| + (1 - c)|\varepsilon| \quad .$$

By hypothesis, $\lambda_i \neq 0$ for $i = 1, \ldots, \beta-1, \beta+1, \ldots, n+1, \alpha$, and hence $c > 0$. Since $|r_\alpha(x)| > \varepsilon$, the strictness of the inequality, along with the two conditions imposed on the selection of row A_β, ensures that if an initial reference set is chosen and subsequently modified by exchange of non-reference set rows of A for rows in the reference set such that the assumed conditions hold at each exchange, then the process will converge to a Chebyshev solution of the entire system.

We now present an algorithmic outline of the Bartells-Golub method:

1. Let $i_1, \ldots, i_{n+1}$ be n+1 indices such that the matrix

$$P = \left[\begin{array}{c|c} A_{i_1} & b_{i_1} \\ \vdots & \vdots \\ A_{i_{n+1}} & b_{i_{n+1}} \end{array}\right]$$

is nonsingular. If there exists no such set of indices, terminate.

2. Perform Gaussian reduction of P^T into the product of a lower triangular matrix L and an upper triangular matrix U.

3. Solve

$$P^T \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{n+1} \end{bmatrix} = LU \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}$$

so that $\sum_{i=1}^{n+1} \lambda_i b_i = -1$ and $\sum_{i=1}^{n+1} \lambda_i A_i = 0$.

4. Determine the current reference deviation ε by $\varepsilon = 1 \Big/ \sum_{i=1}^{n+1} |\lambda_i|$. If ε is less than any ε previously calculated for the current data, go to Step 9.

5. Solve

$$P \begin{bmatrix} x_1 \\ \vdots \\ x_{n+1} \end{bmatrix} = \varepsilon \begin{bmatrix} \mathrm{Sgn}(\lambda_1) \\ \vdots \\ \mathrm{Sgn}(\lambda_{n+1}) \end{bmatrix}$$

6. Calculate the residual for each row not included in the reference set, that is, compute $r_j(x) = \sum_{k=1}^{n} a_{jk}x_k - b_j$ for $j \notin \{i_1, \ldots, i_{n+1}\}$. Find α for which $|r_\alpha(x)|$ is maximal. Then row A_α is to enter the reference subsystem. If $|r_\alpha(x)| \leq \varepsilon$, then $x = (x_1, \ldots, x_n)^T$ is a Chebyshev solution; go to Step 9.

7. Solve $P^T\mu = \left[\begin{array}{c} A_\alpha^T \\ \hline b_\alpha \end{array}\right]$.

8. To determine which row should now leave the reference system, choose $\beta \in \{1, \ldots, n+1\}$ such that $(\mu_\beta/(\lambda_\beta)) \cdot (\mathrm{Sgn}(r_\alpha(x))$ is maximal. Then α will replace β in the reference subsystem. Go to Step 2.

9. Restore the preceding set of indices $i_1, \ldots, i_{n+1}$ and reconstruct the preceding LU decomposition.

10. Iteratively refine the solution to the system

$$P \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \varepsilon \begin{bmatrix} \mathrm{Sgn}(\lambda_1) \\ \vdots \\ \mathrm{Sgn}(\lambda_{n+1}) \end{bmatrix}.$$

Calculate residuals $r_j(x)$ for $j \in \{i_1, \ldots, i_{n+1}\}$. If $|r_\alpha(x)| = \underset{j}{\mathrm{Max}} |r_j(x)| \leq |\varepsilon|$, then $(x_1, \ldots, x_n)^T$ gives the Chebyshev solution to the entire system. If this check is not successful and the last refined value of ε is greater than the current value of ε, give $(x_1, \ldots, x_n)$ as a doubtful solution. Otherwise go to Step 7.

CHAPTER III

THE MEICLER ALGORITHM

Meicler [2] has developed an algorithm to obtain the Chebyshev solution of an overdetermined system of equations $Ax = b$. In his method, the Chebyshev solution is obtained from the sign vector of the least squares solution of the same reference system.

Since there were a number of errors in the Meicler paper, complete proofs for certain theorems where corrections are necessary are included in the following discussion.

As in the previous section, we first consider the case of a reference subsystem A_J with $m = n + 1$.

It is well known that the residual from the least squares solution of $Ax = b$ is given by $d = (AA^+ - I)b$ where A^+ is the Moore-Penrose generalized inverse of A. If A_J and b are partitioned as $A_J = \begin{pmatrix} D \\ v \end{pmatrix}$ and $b = \begin{pmatrix} b \\ b_v \end{pmatrix}$, then the following theorem can be used to calculate the least squares residual without actually calculating the generalized inverse of A.

Theorem 3.1

If $A_J = \begin{pmatrix} D \\ v \end{pmatrix}$ and $b = \begin{pmatrix} b \\ b_v \end{pmatrix}$, then the least squares residual is $d = \alpha(b_v - wb_D)\begin{pmatrix} T \\ -1 \end{pmatrix}$ where $w = vD^{-1}$ and $\alpha = 1/(1+ww^T)$.

The reference deviation ε for the reference subsystem can be obtained from the following theorem.

Theorem 3.2

With the same notation used in Theorem 3.1,

$$\varepsilon = \frac{|b_v - wb_D|}{1 + \sum_{i=1}^{n} |w_i|}$$

Next let us consider the general case where $m > n + 1$. As in the previously discussed algorithms, we again start with $n + 1$ rows of A, and modify the reference subsystem one row at a time. The following theorem, due to Cheney [3], gives the minimax residual in terms of the least squares residual of the reference subsystem.

Theorem 3.3

The minimax residual is given by $r = \varepsilon\sigma$ where σ is the sign vector of the least squares residual and ε is the reference deviation of the reference subsystem.

The following theorems provide the basis for calculating the new reference deviation ε'.

When a row, say p, enters the reference subsystem, one of the following two cases occurs:

Either (1) the last row of A_J changes or (2) any other row of A_J is changed. In the first case, the new w, say w'

is given by $w' = p\ D^{-1}$. In the second case, the following theorem can be used to update the reference deviation after interchanging p and any row of D.

Theorem 3.4

Let D be a nonsingular matrix and $C_1, \ldots, C_n$ columns of its inverse. Let D' be the matrix obtained by replacing the j^{th} row of D by a vector p. If $\lambda = \langle p^T, C_j \rangle \neq 0$ then D' is nonsingular and the columns of its inverse are given by

$$C_j' = \frac{1}{\lambda} C_j$$

and

$$C_i' = C_i - \langle p^T, C_i \rangle C_j', \quad i \neq j \quad .$$

From this theorem it immediately follows that the new w, that is, w', obtained by interchanging p and the j^{th} row of D, is given by

$$w_j' = \lambda^{-1} w_j$$

and

$$w_i' = w_i - \langle p^T, C_i \rangle w_j', \quad i \neq j \quad .$$

Also, $w'b_{D'}$, after this interchange, is given by Lemma 3.1

Lemma 3.1

$$w'b_D' = wb_D + w_j'[b_p - pcb_D]$$

Corrected forms of three results from Meider's paper are given in Lemmas 3.2, 3.3 and 3.6.

Lemma 3.2

$$\sum_{i=1}^{n} |w_i,| = |w_j'| + \sum_{i=1, i\neq j}^{n} |w_i - w_j'\gamma_i|$$

where

$$\gamma_i = \langle p^T, C_i \rangle \ , \ i = 1, 2, \ldots, n.$$

Proof

Since

$$\sum_{i=1}^{n} |w_i'| = |w_j'| + \sum_{i=1, i\neq j}^{n} |w_i'|$$

with $w_i' = w_i - \gamma_i w_j'$,

then

$$\sum_{i=1}^{n} |w_i'| = |w_j'| + \sum_{i=1, i\neq j}^{n} |w_i - w_j'\gamma_i| \ .$$

Next we calculate the minimax solution of the reference subsystem from the sign vector of the least squares solution of the same n + 1 equations.

Lemma 3.3

The minimax solution of any subsystem A_J is given by $x = D^{-1}(b_D + \varepsilon\sigma)$ where σ is the sign vector of the first n components of the corresponding least squares solution.

Proof

By Theorem 3.3 the Chebyshev residual is given by $\varepsilon\sigma$. Since $Dx = b_D x + \varepsilon\sigma$ has an exact solution, $x = D^{-1}(b_D + \varepsilon\sigma)$ as asserted.

We can now calculate the minimax residual for each row p not included in the reference subsystem.

Lemma 3.4

The minimax residual $r_p(x)$ for any row p of A is given by

$$r_p(x) = px - bp = pD^{-1}b_D - bp + \varepsilon pD^{-1}\sigma \quad .$$

The following lemma ensures that there exists a row j belonging to the reference subsystem A_J such that the exchange of A_J and p will yield a greater reference deviation ε'.

Lemma 3.5

If $|r_p| > \varepsilon$, the current reference deviation, then there exists an index $j \varepsilon J$ such that the exchange of p and a_j will ensure a greater reference deviation ε'.

We have the following lemma to calculate ε'.

Lemma 3.6

$$\varepsilon' = \frac{|N_v - w_j' N_p|}{1 + \sum_{i=1}^{n} |w_i'|}$$

where

$$N_v = b_v - wb_D \quad \text{and} \quad N_p = b_p - pCb_D \ .$$

Proof

Clearly,

$$\varepsilon' = \frac{|b_v - w'b_D'|}{1 + \sum_{i=1}^{n} |w_i'|}$$

where $w'b_D' = wb_D + w_j'[b_p - pCb_D]$ by Lemma 3.4.

Therefore,

$$\varepsilon' = \frac{|b_v - wb_D + w_j'[b_p - pCb_D]|}{1 + \sum_{i=1}^{n} |w_i'|}$$

or

$$\varepsilon' = \frac{|N_v - w_j' N_p|}{1 + \sum_{i=1}^{n} |w_i'|} \ .$$

We now summarize an implementation of the Meicler algorithm.

Detailed Implementation of the Meicler Algorithm

1. Select (n+1) indices $J = \{i_1, \ldots, i_{n+1}\}$ so that $A_J = \begin{pmatrix} D \\ v \end{pmatrix}$ where D is nxn nonsingular and v is the i_{n+1}-th row of A.

2. Since any vector p not in D will generate an ε, calculate

$$\underset{j \geq n+1}{\varepsilon_j} = \frac{|b_v - wb_D|}{1 + \sum_{i=1}^{n} |w_i'|}$$

where $w = A_j D^{-1}$.

3. Find the max ε_j, call it ε', the current Chebyshev amplitude.

4. For each row not included in D calculate the residual using $r_p = pD^{-1}b_D - b_p + \varepsilon' pD^{-1}\sigma$.

5. For $p \notin \{i_1, \ldots i_n\}$ find the row for which $|\gamma_p|$ is maximum. Call that row A_α. Row A_α is to enter the reference subsystem. If Max $|\gamma_p| \leq \varepsilon$, then stop.

6. Find the row which should leave the basis using the equation

$$\varepsilon' = \frac{|N_v - \beta N_p|}{1 + \sum_{i=1}^{n} |w_i'|}$$

where $\beta = w_j'$. Calculate for each row in D, the ε that may be expected if row A_α replaces that row. (Thus there will be n different values of ε'.)

7. For $\beta \varepsilon \{i_1, \ldots, i_n\}$ determine an index β such that $|\varepsilon_\beta'|$ is maximum. Row A_β should now leave the reference set. Go to Step 2.

CHAPTER IV

NUMERICAL RESULTS AND CONCLUSIONS

Computer programs were written in FORTRAN IV to implement the Golub-Bartells algorithm and the Meicler algorithm. In the Golub-Bartells method, starting with any reference subsystem of the given overdetermined system of equations, the exchange method produces a new reference subsystem at the expense of solving three sets of n + 1 linear equations of the form $Py = q_1$, $P^T x = q_2$, and $Pz = q_3$, where q_1, q_2, and q_3 are three specified vectors. The method of solution is to make an LU decomposition of P^T and to solve the six triangular system of equations by back substitution. This can be done with $\frac{n^3}{3} + 4n^2 + 0(n)$ operations of multiplication and division. In the LU decomposition of P^T, pivot selection using row interchanges is employed and, consequently, the algorithm is almost always numerically stable in practice (see comments to this effect in [4] and [5].) Golub and Bartells [1] have calculated that the total number of operations performed at each exchange step would be $\frac{n^3}{3} + \frac{29}{6} n^2 + 0(n)$, where they assume that all the rows in the nonreference set are equally likely to appear in the reference subsystem at any exchange step.

The Meicler algorithm requires the calculation of D^{-1} which can be done in n^3 operations. Calculation of least squares residual involves $n^2 + 2n + 1$ operations and the Chebyshev residual itself can be computed in $n^2 + 1$ operations. The residual corresponding to any row of the matrix can be calculated in $2n^2 + n + 2$ operations. Thus an exchange step involves $n^3 + (3 + m)n^2 + 0(n)$ operations. This, obviously, is more than the number of operations necessary in an exchange step in Golub-Bartells algorithm.

The requirement that the system of equations satisfy the Haar condition puts severe restrictions on the type of problems which can be solved by the methods discussed in this paper. For the purpose of testing the programs, the examples given in Cheney (3) and Meicler were used. In addition, in another experiment to compare the two algorithms, a Haar system of 7 equations in 4 unknowns was constructed in the following way. Observe that if

$$A = \begin{bmatrix} I \\ A_1 \\ A_2 \\ A_3 \end{bmatrix}$$

with I the 4x4 identity matrix and $A_1 = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]$, $A_2 = [\beta_1, \beta_2, \beta_3, \beta_4]$, $A_\beta = [\gamma_1, \gamma_2, \gamma_3, \gamma_4]$,

then

$$A_1 = \sum_{i=1}^{4} \alpha_i e_i$$

$$A_2 = \sum_{i=1}^{4} \beta_i e_i$$

and

$$A_3 = \sum_{i=1}^{4} \gamma_i e_i$$

where e_1, e_2, e_3, and e_4 are the rows of I. Then clearly A satisfies the Haar condition if and only if each submatrix of

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \end{bmatrix}$$

is nonsingular.

Hence, to extend this procedure to obtain larger matrices A satisfying the Haar condition, observe that if we add another row, say A_4, to A then for A to satisfy the Haar condition, each submatrix of

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\ \delta_1 & \delta_2 & \delta_3 & \delta_4 \end{bmatrix}$$

must now be nonsingular, where $a_4 = \sum_{i=1}^{4} \delta_i e_i$. The extensions of this technique to general m by n matrices $A = \begin{pmatrix} I \\ V \end{pmatrix}$ is apparent by noting that every submatrix of V must be nonsingular. Thus arbitrary systems which satisfy the Haar condition are difficult to obtain. However, when the Haar condition is satisfied, the Chebyshev solution will be unique and, as mentioned by Rice [6], it is "the best of the best solutions".

Numerical Examples

1. Given the system of equations Ax = b with

$$A = \begin{bmatrix} 2 & 1 \\ 3 & 1 \\ 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 6.9 \\ 7.2 \\ 7 \\ 3 \\ 1 \end{bmatrix},$$

then $x = (2.0, 2.0)^T$ by the two algorithms.

2. On the other hand, for

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 2 \\ 1 & -1 & 2 & 4 \\ 3 & 1 & 7 & 11 \end{bmatrix}, \quad b = \begin{bmatrix} 2.2 \\ 3.1 \\ 1 \\ 1 \\ 7.8 \\ 6.5 \\ 26 \end{bmatrix},$$

then by the Meicler algorithm

$$x = (1.35676,\ 2.35676,\ 0.3567,\ 1.6108)^T \quad ,$$

whereas by the Golub-Bartells algorithm

$$x = (1.356753,\ 2.356752,\ 0.356756,\ 1.61807)^T \quad .$$

In Golub-Bartells algorithm an iterative refinement on the solution is made in Step 9 so that the refined solution is accurate to all digits in single precision.

It is also of interest to compare the CPU time for the two procedures as summarized in Table I.

TABLE I

COMPARISON OF CPU TIME FOR GOLUB-BARTELLS AND MEICLER ALGORITHMS

Procedure	CPU Time Ex. 1	CPU Time Ex. 2
Golub-Bartells	13.9 sec.	14.24 sec.
Meicler	16.5 sec.	15.35 sec.

As indicated in Table I, the Golub-Bartells algorithm is faster than the Meicler algorithm, as expected from the previously indicated operational count.

Considering computational stability, the Meicler algorithm calculates the inverse of an n by n matrix D at

the initial stage and modifies it at subsequent exchange steps. While row and column interchanges can be made at the initial step which forms D^{-1}, no pivot choice can be made at subsequent updatings of D^{-1}. The dangers in this process are studied in detail by Wilkinson [4].

LIST OF REFERENCES

LIST OF REFERENCES

1. Bartells, R. H., and Golub, G. "Stable Numerical Methods for obtaining the Chebyshev solution of an overdetermined system of equations." Communications of the ACM. Vol. II, Number 6, June 1968, pp. 401-406.

2. Meicler, Marcel. "A Steepest Ascent Method for the Chebyshev Problem." Mathematics of Computation, 23, 1969, pp. 813-817.

3. Cheney, E. W. Introduction to Approximation Theory. (International series in Pure and Applied Mathematics) McGraw-Hill, New York, 1966.

4. Wilkinson, J. H. Rounding Errors in Algebraic Process. (Series in Automatic Computation) Prentice-Hall, Englewood Cliffs, N.J., 1963.

5. Wilkinson, J. H. "Rounding Errors in Algebraic Process." Proc. Int. Conf. on Inform. Process. UNESCO, 1959.

6. Rice, J. R. "Chebyshev approximation in a Compact Metric space." Bull. Amer. Math. Soc., 68(1962), pp. 405-410.

7. Forsythe, G. E., and Moler, C. B. Computer Solution of Linear Algebraic Systems. (Series in Automatic Computation), Prentice-Hall, Englewood Cliffs, N.J., 1967.

8. Isaacson, E., and Keller, H. B. Analysis of Numerical Methods, John Wiley and Sons, Inc., New York, 1966.

9. Zukhovitskiy, S. E., and Avdeyeva, L. I. Linear and Connex Programming (Translated from the Russian by Scripta Technica Inc.), W. B. Saunders Company, Philadelphia, 1966.

APPENDIXES

```
FORTRAN IV G LEVEL  21                    MAIN

                C
                C--         THIS PROCEDURE COMPUTES A SOLUTION IN THE
                C--         CHEBYSHEV SENSE TO AN OVERDETERMINED SYSTEM OF
                C--         EQUATIONS AX=B USING AN EXCHANGE ALGORITHM DUE
                C--         TO BARTELLS AND GOLUB.  THE PARAMETERS IN THIS
                C--         PROCEDURES ARE ------
                C--         IDENTIFIER      TYPE           COMMENTS
                C
                C--            M          INTEGER      NUMBER OF EQUATIONS
                C--            N          INTEGAR      NUMBER OF UNKNOWNS
                C--            A          REAL ARRAY   MATRIX OF COEFFICIENTS
                C--            B          REAL ARRAY   RIGHT HAND SIDE VECTOR
                C--            X          REAL ARRAY   SOLUTION VECTOR
                C
                C--         THE INITIAL REFERENCE SUBSYSTEM IS MADE BY MAKING
                C--         A COPY OF THE TRANSPOSE OF A BORDERED WITH B AND
                C--         CARRYING OUT A GAUSSIAN REDUCTION UPON IT WITH
                C--         ROW INTERCHANGES USED TO SELECT THE MAXIMUM PIVOT
                C--         AT EACH STAGE.
                C
0001                        COMMON A(10,10),B(10)
0002                        DIMENSION P(10,10),PT(10,10),Y(10),X(10),XY(10),
                          1 ISIGMA(10),RES(10),IPIVOT(10),XLAMDA(10)
0003                        DIMENSION EPSILN(200),UL(10,10),BB(10)
0004                        DIMENSION IOLD(10),INEW(10)
0005                        READ (5,1) N
0006                        READ (5,1) M
0007                        READ (5,2) ((A(I,J),J=1,N),I=1,M)
0008                        WRITE (6,610)
0009                        WRITE (6,611)
0010                        WRITE (6,612)
0011                        WRITE (6,2) ((A(I,J),J=1,N),I=1,M)
0012                        WRITE (6,613)
0013                        READ (5,2) (B(I),I=1,M)
0014                        WRITE (6,2) (B(I),I=1,M)
0015                        WRITE (6,676) M,N
0016                        EPZ=1.OE-6
0017                        NN=N+1
0018                        DC 343 I=1,N
0019              343       BB(I)=0.0
0020                        BB(NN)=-1.0
0021                        IBIG=1
0022                        DO 101 I=1,NN
```

```
0023                    IOLD(I)=I
0024          101       INEW(I)=I
0025          100       WRITE (6,611)
0026                    DO 102 I=1,NN
0027          102       IOLD(I)=INEW(I)
0028                    DO 3 I=1,NN
0029                    DO 4 J=1,N
0030                    P(I,J)=A(K,J)
0031          4         CONTINUE
0032          3         CONTINUE
0033                    WRITE (6,222) IBIG
0034                    DO 6 I = 1,NN
0035                    P(I,NN)=B(I)
0036          6         CONTINUE
0037                    DO 44 I=1,NN
0038                    DO 45 J=1,NN
0039          45        PT(I,J)=P(J,I)
0040          44        CONTINUE
                        C
                        C
                        C--       THE FOLLOWING SEGMENT OF THE PROGRAM PERFORMS A
                        C--       GAUSSIAN REDUCTION OF THE MATRIX ASSOCIATED WITH
                        C--       THE REFERENCE EQUATIONS, FORMING AN UPPER TRIAN-
                        C--       GULAR MATRIX U AND A LOWER TRIANGULAR MATRIX L.
                        C--       EACH DIAGONAL ELEMENT OF L IS ONE.  IF THE MATRIX
                        C--       IS NOT OF FULL RANK, THE EXIT IS TAKEN AND IT IS
                        C--       LEFT UP TO THE USER TO DETERMINE IF THE GIVEN
                        C--       OVERDETERMINED SYSTEM OF EQUATIONS CAN BE SOLVED
                        C--       EXACTLY.
                        C
                        C
                        C--       SOLVE FOR THE LAMDAS
                        C
0041                    CALL DECOMP (PT,UL,IPIVOT,NN)
0042                    CALL SOLUTN (UL,XLAMDA,IPIVOT,BB,NN)
0043                    WRITE (6,614)
0044                    WRITE (6,210) (XLAMDA(I),I=1,NN)
0045                    DO 212 KJ=1,NN
0046                    IF (XLAMDA(KJ)) 212,213,212
0047          213       WRITE (6,214)
0048                    GO TO 991
0049          212       CONTINUE
                        C
                        C--       COMPUTE THE MINIMAX REFERENCE DEVIATION FOR THE
                        C--       SUBSYSTEM EACH VALUE OF THE REFERENCE DEVIATION
                        C--       MUST BE GREATER THAN THE PREVIOUS ONE.  IF THIS
                        C--       IS NOT SO THE SOLUTION MAY HAVE BEEN "OVERSHOT".
                        C
```

```
0050                    EPSILN(IBIG)=0.0
0051                    DO 8 I=1,NN
0052          8         EPSILN(IBIG)=EPSILN(IBIG)+ABS(XLAMDA(I))
0053                    EPSILN(IBIG)=1.0/EPSILN(IBIG)
0054                    WRITE (6,630) EPSILN(IBIG)
0055                    IF (IBIG.EQ.1) EPSMIN=EPSILN(1)
0056                    CALL TEST (EPSILN,EPSMIN,IBIG,IFLAG)
0057                    CALL SIGN (XLAMDA,ISIGMA,NN)
0058                    DO 10 I=1,NN
0059          10        Y(I)=EPSILN(IBIG)*ISIGMA(I)
0060                    IF (IFLAG) 999,998,999
                C
                C--     SOLVE FOR X , THE CHEBYSHEV SOLUTION OF THE
                C--     REFERENCE SUBSYSTEM OF EQUATIONS.
                C
0061          998       DO 555 I=1,NN
0062                    DO 556 J=1,NN
0063          556       P(I,J)=UL(J,I)
0064          555       CONTINUE
0065                    NN=N+1
0066                    CALL SOLV (P,X,Y,IPIVOT,NN)
0067                    WRITE (6,617)
0068                    WRITE (6,109( (X(I),I=1,N)
                C
                C--     FOR EACH INDEX N+1 THROUGH M COMPUTE THE RESIDUAL.
                C--     IF THE LARGEST OF THESE IN MAGNITUDE IS NOT
                C--     GREATER THAN THE CURRENT REFERENCE DEVIATION,
                C--     THEN REFINE THE VECTOR X . IT MAY BE THE CHEBY-
                C--     SHEV SOLUTION OF THE FULL SYSTEM.
                C
0069                    N2=NN+1
0070                    CALL RESD (RES,X,N2,M,N)
0071                    WRITE (6,615)
0072                    WRITE (6,109) (RES(I),I=N2,M)
                C
                C--     DETERMINE THE ROW TO BE REMOVED FROM THE REFERENCE
                C--     SET.  CALL THAT ROW ALPHA.
                C
0073                    CALL RMAX (RES,LALPHA,EPSNPR,N2,M,N)
0074                    WRITE (6,888) LALPHA
0075                    IF (EPSNPR.LE.EPSILN(IBIG)) GO TO 99
0076          111       DO 14 I=1,N
0077          14        Y(I)=A(LALPHA,I)
0078                    Y(NN)=B(LALPHA)
```

```
            C
            C--         THE FOLLOWING LINEAR SYSTEM OF EQUATIONS IS
            C--         SOLVED IN ORDER TO FIND OUT WHICH EQUATION IS TO
            C--         BE DROPPED FROM THE REFERENCE SET.
            C
0079                    CALL SOLUTN (UL,XY,IPIVOT,Y,NN)
0080                    IF (RES(LALPHA)) 17,15,16
0081        15          ISIGN=0
0082                    GO TO 18
0083        16          ISIGN=1
0084                    GO TO 18
0085        17          ISIGN=-1
0086        18          DO 19 I=1,NN
0087                    XY(I)=XY(I)/XLAMDA(I)
0088        19          XY(I)=ISIGN*XY(I)
            C
            C--         DETERMINE THE ROW TO ENTER THE REFERENCE SET,
            C--         SAY ROW BETA.  FORM A NEW REFERENCE SUBSYSTEM
            C--         BY INTERCHANGING ROWS ALPHA AND BETA.
            C
0089                    CALL CRAZY (XY,LBETA,TOP,1,NN,N)
0090                    WRITE (6,889) LBETA
0091                    DO 20 I=1,N
0092                    Y(I)=A(LALPHA,I)
0093                    A(LALPHA,I)=A(LBETA,I)
0094                    A(LBETA,I)=Y(I)
0095        20          CONTINUE
0096                    TEMP=B(LBETA)
0097                    B(LBETA)=B(LALPHA)
0098                    B(LALPHA)=TEMP
0099                    IT=LALPHA
0100                    INEW(LBETA)=IT
0101                    IBIG=IBIG+1
0102                    GO TO 100
0103        999         DO 105 JJ=1,NN
0104                    DO 106 J=1,N
0105                    II=IOLD(JJ)
0106                    P(JJ,J)=A(II,J)
0107        106         CONTINUE
0108        105         CONTINUE
0109                    DO 107 IK=1,NN
0110                    ID=IOLD(IK)
0111        107         P(IK,NN)=B(ID)
            C
            C--         ITERATIVELY REFINE X
            C
0112                    CALL IMPRUV (P,UL,X,Y,NN)
0113                    WRITE (6,109( (X(I),I=1,N)
```

```
FORTRAN IV G LEVEL 21                    MAIN

0114                      CALL RESD (RES,X,N2,M,N)
0115                      BIG=ABS(RES(LALPHA))
0116                      CALL RMAX (RES,LAL,BIGR,N2,M,N)
0117                      DIFFR=ABS(BIG-BIGR)
0118                      IF (DIFFR.LE.EPZ.AND.BIG.LE.EPSILN(IBIG)) GO TO
                         1 99
0119                      GO TO 111
0120          99          WRITE (6,616)
0121                      WRITE (6,109) (X(I),I=1,N)
0122          109         FORMAT (4F15.9)
0123                      WRITE (6,611)
0124          1           FORMAT(I10)
0125          2           FORMAT (4F10.5)
0126          610         FORMAT (//,25X,'SAMPLE OUTPUT FOR GOLUB-BARTELLS
                         1 ALGORITHM',/)
0127          611         FORMAT (/,120(1H*),/)
0128          612         FORMAT (/,5X,'THE MATRIX OF COEFFICIENTS',//)
0129          613         FORMAT (/,5X,'THE RIGHT HAND SIDE VECTOR',//)
0130          222         FORMAT (/,5X,'ITERATION STEP',I10,//)
0131          614         FORMAT (/,5X,'THE LAMDA VECTOR FOR THIS STEP
                         1 IS GIVEN BELOW',/)
0132          210         FORMAT (5F15.5)
0133          214         FORMAT (/,5X,'MATRIX IS SINGULAR',//)
0134          615         FORMAT (/,5X,'RESIDUAL FOR ROWS N+2 THROUGH M
                         1 FOLLOWS',/)
0135          888         FORMAT (//,5X,'ROW TO ENTER THE REFERENCE SET IS
                         1 ROW',I5,/)
0136          630         FORMAT (//,5X,'THE CHEBYSHEV RESIDUAL FOR THIS
                         1 REFERENCE SUBSYSTEM =',F15.5,/)
0137          889         FORMAT (//,5X,'ROW TO LEAVE THE REFERENCE SET IS
                         1 ROW',I5,/)
0138          616         FORMAT (//,5X,'CHEBYSHEV SOLUTION FOR THIS PROBLEM
                         1 IS GIVEN BELOW',//)
0139          617         FORMAT (/,5X,'CHEBYSHEV SOLUTION FOR THIS REFERENCE
                         1 SUBSYSTEM FOLLOWS',/)
0140          676         FORMAT (/,5X,'THIS IS A SYSTEM OF',I5,3X,'
                         1 EQUATIONS IN',I5,4X,'UNKNOWNS',//)
0141          991         STOP
0142                      END
```

```
FORTRAN IV G LEVEL 21                    SIGN

0001                    SUBROUTINE SIGN (SUMA,IB,N)
0002                    DIMENSION IB(10), SUMA(10)
0003                    DO 500 I=1,N
0004                    IF (SUMA(I)) 501,502,503
0005          501       IB(I)=-1
0006                    GO TO 500
0007          503       IB(I)=1
0008                    GO TO 500
0009          502       IB(I)=0
0010          500       CONTINUE
0011                    RETURN
0012                    END
```

```
FORTRAN IV G LEVEL 21                    RESD

0001                    SUBROUTINE RESD (RES,X,IFIRST,ILAST,N)
0002                    COMMON A(10,10),B(10)
0003                    DIMENSION X(10),RES(10)
0004                    DO 400 J=IFIRST,ILAST
0005                    SUM=0.0
0006                    DO 401 K=1,N
0007          401       SUM=SUM+A(J,K)*X(K)
0008                    RES(J)=SUM-B(J)
0009          400       CONTINUE
0010                    RETURN
0011                    END
```

```
FORTRAN IV G LEVEL 21                    RMAX

0001                    SUBROUTINE RMAX (R,LALPHA,BIG,I1,I2,N)
0002                    COMMON A(10,10),B(10)
0003                    DIMENSION R(10)
0004                    LASPHA=I1
0005                    BIG-ABS(R(I1))
0006                    DO 450 I=I1,I2
0007                    IF (BIG-ABS(R(I))) 451,450,450
0008          451       BIG-ABS(R(I))
0009                    LALPHA=I
0010          450       CONTINUE
0011                    RETURN
0012                    END
```

```
FORTRAN IV G LEVEL 21                    WHY

0001                    SUBROUTINE WHY (IWHY)
0002                    GO TO (641,642,643) , IWHY
0003          641       WRITE (6,644)
0004                    GO TO 689
0005          642       WRITE (6,645)
0006                    GO TO 689
0007          643       WRITE (6,646)
0008          644       FORMAT (/,' ','MATRIX WITH ZERO ROW',/)
0009          645       FORMAT (/,' ','SINGULAR MATRIX',/)
0010          646       FORMAT (/,' ','NO CONVERGENCE IN IMPRUV.',/)
0011          689       RETURN
0012                    END


FORTRAN IV G LEVEL 21                    DECOMP

0001                    SUBROUTINE DECOMP (PROD,UL,IPIVOT,N)
              C
              C--       THIS PROCEDURE PERFORMS GAUSSIAN ELIMINATION OF
              C--       A , REDUCING A TO THE PRODUCT OF AN UPPER TRIAN-
              C--       GULAR MATRIX U AND A LOWER TRIANGULAR MATRIX L
              C--       WHERE THE DIAGONAL ELEMENTS OF L ARE UNITY.
              C--       PARTIAL PIVOTING IS USED SO THAT THE PIVOT OF
              C--       LARGEST VALUE IS USED.  ARRAY IPIVOT STORES THE
              C--       EXCHANGE INFORMATION WHICH SHOULD BE AVAILABLE
              C--       WHEN THE EQUATIONS ARE SOLVED.
              C
0002                    COMMON A(10,10),B(10)
0003                    DIMENSION IPIVOT(10),UL(10,10),D(10)
0004                    DIMENSION PROD(10,10)
0005                    DO 500 I=1,N
0006                    IPIVOT(I)=I
0007                    ROWMAX=0.0
0008                    DO 501 J=1,N
0009                    UL(I,J)=PROD(I,J)
0010                    IF (ROWMAX-ABS(UL(I,J))) 502,501,501
0011          502       ROWMAX=ABS(UL(I,J))
0012          501       CONTINUE
0013                    IF (ROWMAX) 503,504,503
0014          504       CALL WHY(1)
0015                    D(I) = 0.0
0016                    GO TO 500
0017          503       D(I)=1.0/ROWMAX
0018          500       CONTINUE
0019                    N1=N-1
0020                    DO 505 IJ=1,N1
```

```
FORTRAN IV G LEVEL 21                    DECOMP

0021                    BIG=0.0
0022                    DO 506 I=IJ,N
0023                    IP=IPIVOT(I)
0024                    EBBY=ABS(UL(IP,IJ))*D(IP)
0025                    IF (EBBY-BIG) 506,506,507
0026          507       BIG=EBBY
0027                    ID=I
0028          506       CONTINUE
0029                    IF (BIG) 508,509,508
0030          509       CALL WHY(2)
0031                    GO TO 505
0032          508       IF (ID-IJ) 510,511,510
0033          510       J=IPIVOT(IJ)
0034                    IPIVOT(IJ)=IPIVOT(ID)
0035                    IPIVOT(ID)=J
0036          511       II=IPIVOT(IJ)
0037                    PIVOT=UL(II,IJ)
0038                    IJ1=IJ+1
0039                    DO 512 I=IJ1,N
0040                    JV=IPIVOT(I)
0041                    TEMP=-UL(JV,IJ)/PIVOT
0042                    UL(JV,IJ)=-TEMP
0043                    DO 512 J=IJ1,N
0044                    UL(JV,J)=UL(JV,J)+TEMP*UL(II,J)
0045          512       CONTINUE
0046          505       CONTINUE
0047                    II=IPIVOT(N)
0048                    IF (UL(II,N)) 599,598,599
0049          598       CALL WHY (2)
0050          599       RETURN
0051                    END


FORTRAN IV G LEVEL 21                    SOLUTN

0001                    SUBROUTINE SOLUTN (UL,X,IPIVOT,Y,N)
                C
                C--     THIS PROCEDURE SOLVES THE SYSTEM LUX=B, FIRST BY
                C--     SOLVING UX=Y BY FORWARD SUBSTITUTION AND THEN
                C--     SOLVING UY=B BY BACK SUBSTITUTION.
                C
0002                    COMMON A(10,10),B(10)
0003                    DIMENSION Y(10)
0004                    DIMENSION X(10),UL(10,10),IPIVOT(10)
0005                    N1=N+1
0006                    II=IPIVOT(1)
```

```
FORTRAN IV G LEVEL 21                    SOLUTN

0007                    X(1)=Y(II)
0008                    DO 601 I=2,N
0009                    II=IPIVOT(I)
0010                    II=I-1
0011                    SUM=0.0
0012                    DO 602 J=1,I1
0013          602       SUM=SUM+UL(II,J)*X(J)
0014          601       X(I)=Y(II)-SUM
0015                    II=IPIVOT(N)
0016                    X(N)=X(N)/UL(II,N)
0017                    DO 603 IB=2,N
0018                    I=N1-IB
0019                    II=IPIVOT(I)
0020                    I1 = I+1
0021                    SUM=0.0
0022                    DO 604 J=I1,N
0023          604       SUM=SUM+UL(II,J)*X(J)
0024          603       X(I)=(X(I)-SUM)/UL(II,I)
0025                    RETURN
0026                    END
```

```
FORTRAN IV G LEVEL 21                    IMPRUV

0001                    SUBROUTINE IMPRUV (PP,UL,X,EP,N)
                C
                C--     THIS PROGRAM REFINES ITERATIVELY THE SOLUTION X
                C--     ACCORDING TO THE SCHEME GIVEN BY FORSYTHE AND
                C--     MOLER.
                C--     FOR THE FIRST SOLUTION X COMPUTE THE DOUBLE
                C--     PRECISION RESIDUAL R1=B-AX1.  THEN SOLVE AZ1=R1.
                C--     IF WE KNOW Z1 EXACTLY THEN X2=X1+Z1 WOULD SOLVE
                C--     THE SYSTEM AX=B EXACTLY.  IF WE KNOW Z1 TO A FEW
                C--     DIGITS, FORMING R2=B-AX2 AND COMPUTE Z2 AND X3.
                C--     THE SINGLE PRECISION VECTORS X1,X2,X3----------
                C--     FORM A SEQUENCE WHICH CONVERGES TO A CONSTANT
                C--     VECTOR WHICH IS A SOLUTION TO AX=B ACCURATE TO
                C--     FULL SINGLE PRECISION.
                C
0002                    COMMON A(10,10),B(10)
0003                    DIMENSION EP(10),PP(10,10)
0004                    DIMENSION UL(10,10),X(10),DD(10),R(10)
0005                    DIMENSION IPIVOT(10)
0006                    REAL*8 SUM
0007                    EPZ=1.0E-8
0008                    ITMAX=100
```

```
FORTRAN IV G LEVEL 21                    IMPRUV

0009                    XNORM=0.0
0010                    DO 610 I=1,N
0011          610       XNORM=AMAX1(XNORM,ABS(X(I)))
0012                    IF (XNORM) 612,611,612
0013          611       EM=-ALOG10(EPZ)
0014                    GO TO 699
0015          612       DO 613 ITER=1,ITMAX
0016                    DO 614 I=1,N
0017                    SUM=0.0
0018                    DO 615 J=1,N
0019          615       SUM=SUM+PP(I,J)*X(J)
0020                    SUM=EP(I)-SUM
0021          614       R(I)=SUM
0022                    CALL DECOMP (PP,UL,IPIVOT,N)
0023                    CALL SOLUTN (UL,DD,IPIVOT,R,N)
0024                    DNORM=0.0
0025                    DO 616 I=1,N
0026                    TEMP=X(I)
0027                    X(I)=X(I)+DD(I)
0028                    DNORM=AMAX1(DNORM,ABS(X(I)-TEMP))
0029          616       CONTINUE
0030                    IF (ITER-1) 618,617,618
0031          617       EM=-ALOG10(AMAX1(DNORM,EPZ))
0032          618       IF (DNORM-EPZ*XNORM) 699,699,613
0033          613       CONTINUE
0034                    CALL WHY(3)
0035          699       RETURN
0036                    END


FORTRAN IV G LEVEL 21                    TEST

0001                    SUBROUTINE TEST (EPS,EPSMIN,LAST,IFLAG)
0002                    DIMENSION EPS(200)
0003                    IFLAG=0
0004                    DO 900 I=1,LAST
0005                    IF (EPS(I).LT.EPSMIN) GO TO 950
0006          900       CONTINUE
0007                    GO TO 951
0008          950       IFLAG=1
0009                    EPSMIN=EPS(I)
0010          951       RETURN
0011                    END
```

```
FORTRAN IV G LEVEL 21                    CRAZY

0001                    SUBROUTINE CRAZY (Z,LAL,PAL,IF,IL,N)
0002                    COMMON A(10,10),B(10)
0003                    DIMENSION Z(10)
0004                    LAL=IF
0005                    PAL=Z(IF)
0006                    DO 550 I=IF,IL
0007                    IF (PAL-Z(I)) 551,550,550
0008          551       PAL=Z(I)
0009                    LAL=I
0010          550       CONTINUE
0011                    RETURN
0012                    END
```

```
FORTRAN IV G LEVEL 21                    SOLV

0001                    SUBROUTINE SOLV (UL,X,Y,IPIVOT,N)
              C
              C--       THIS PROCEDURE SOLVES A SYSTEM OF EQUATIONS OF
              C--       THE FORM LUX=Y WHERE U IS UNIT UPPER TRIANGULAR
              C--       AND L IS LOWER TRIANGULAR.  FIRST SOLVE LB=Y BY
              C--       FORWARD SUBSTITUTION AND THEN SOLVE UX=B BY BACK
              C--       SUBSTITUTION.  THIS PROCEDURE IS A SLIGHT MODI-
              C--       FICATION OF THE PROCEDURE "SOLUTN" USED ELSEWHERE
              C--       IN THIS PROGRAM ON RETURN X CONTAINS THE SOLUTION
              C--       VECTOR.
              C
0002                    COMMON A(10,10),B(10)
0003                    DIMENSION UL(10,10),PL(10,10)
0004                    DIMENSION X(10),Y(10),IPIVOT(10),C(10)
0005                    N1=N+1
0006                    DO 200 IBIG=1,N
0007                    IP=IPIVOT(IBIG)
0008                    DO 100 I=1,N
0009          100       PL(I,IBIG)=UL(I,IP)
0010          200       CONTINUE
0011                    IP=IPIVOT(1)
0012                    C(1)=Y(1)/PL(1,1)
0013                    DO 2 I=2,N
0014                    I1=I-1
0015                    SUM=0.0
0016                    DO 1 J=1,I1
0017          1         SUM=SUM+PL(I,J)*C(J)
0018          2         C(I)=(Y(I)-SUM)/PL(I,I)
```

```
                C
                C--         FORWARD SUBSTITUTION IS NOW COMPLETE.  START
                C--         BACK SOLUTION
                C
0019                        IP=IPIVOT(N)
0020                        X(IP)=C(N)
0021                        DO 4 IK=2,N
                C
                C--         INDEX I GOES BACKWARDS, N-1,----------1.
                C
0022                        I=N1-IK
0023                        IP=IPIVOT(I)
0024                        I1=I+1
0025                        SUM=0.0
0026                        DO 3 J=I1,N
0027                        IPJ=IPIVOT(J)
0028            3           SUM=SUM+PL(I,J)*X(IPJ)
0029            4           X(IP)=C(I)-SUM
0030                        RETURN
0031                        END
```

```
FORTRAN IV G LEVEL 21                    MAIN

              C
              C--         THIS ROUTINE SOLVES AN OVER DETERMINED SYSTEM OF
              C--         EQUATIONS OF THE FORM AX=B, USING AN EXCHANGE
              C--         ALGORITHM DUE TO MEICLER.  M IS THE NUMBER OF
              C--         EQUATIONS BEING SOLVED AND N THE NUMBER OF
              C--         UNKNOWNS.
              C
0001                      COMMON A(10,10),D(10,10),DINV(10,10),B(10)
0002                      DIMENSION OMEGA(10),ISIGMA(10),RESD(10),EPSILN(10)
0003                      DIMENSION P(10),X(10),DLQ(10)
0004                      DIMENSION WJ(10)
0005                      LOGICAL ISING
0006                      READ (5,500) N
0007                      READ (5,500) M
0008                      READ (5,501) ((A(I,J),J=1,N),I=1,M)
0009                      WRITE (6,610)
0010                      WRITE (6,611)
0011                      WRITE (6,612)
0012                      WRITE (6,501) ((A(I,J),J=1,N),I=1,M)
0013                      READ (5,501) (B(I),I=1,M)
0014                      WRITE (6,613)
0015                      WRITE (6,501) (B(I),I=1,M)
0016                      WRITE (6,676) M,N
0017                      WRITE (6,611)
              C
              C--         STEP 1.  THE INITIAL REFERENCE SET IS FORMED BY
              C--         ROWS 1 THROUGH N+1.  CALCULATE THE REFERENCE
              C--         DEVIATION FOR EACH ROW N+1 THROUGH M.  THE SUB-
              C--         PROGRAM 'GETEPS' RETURNS THE REFERENCE DEVIATION
              C--         FOR EACH ROW.
              C
0018                      MN=M-N
0019                      EPZ=1.0E-5
0020                      DO 111 I=1,N
0021                      DO 112 J=1,N
0022                      DINV(I,J)=A(I,J)
0023          112         D(I,J)=A(I,J)
0024          111         CONTINUE
0025                      CALL INVERT (DINV,N,10,ISING)
0026                      WRITE (6,501) ((DINV(I,J),J=1,N),I=1,N)
0027                      N1=N+1
0028                      DO 9999 IL=1,100
```

```
0029                    WRITE (6,611)
0030                    WRITE (6,614) IL
0031                    CALL GETEPS (OMEGA,N1,M,EPSILN,N)
                C
                C--     STEP 2.  CALCULATE THE MAXIMUM REFERENCE
                C--     DEVIATION.  THIS GIVES THE CHEBYSHEV AMPLITUDE
                C--     FOR THE CURRENT ITERATION STEP.
                C
0032                    CALL AMAX (EPSILN,EPSNPR,LRMAX,N1,M)
0033                    WRITE (6,678) EPSNPR
                C
                C--     STEP 3.  CALCULATE THE RESIDUAL FOR EACH ROW IN
                C--     THE NONREFERENCE SET.  SUBPROGRAM 'RESDLQ',ON
                C--     CALL,RETURNS THE LEAST SQUARES RESIDUAL FOR THE
                C--     CURRENT REFERENCE SET.
                C
0034                    N1=N+1
0035                    CALL RESDLQ (DLQ,LRMAX,N)
0036                    CALL SIGN (DLQ,ISIGMA,N1)
0037                    DO 7 II=N1,M
0038                    DO 8 K=1,N
0039                    P(K)=0.0
0040                    DO 9 KK=1,N
0041            9       P(K)=P(K)+DINV(K,KK)*B(KK)
0042            8       CONTINUE
0043                    S1=0.0
0044                    DO 10 I=1,N
0045            10      S1=S1+A(II,I)*P(I)
                C----   CALCULATE SIGN VECTOR OF OMEGA FOR EACH OF THE
                        ROWS.
0046                    CALL OMEGN (OMEGA,II,N)
0047                    S2=0.0
0048                    DO 13 KJ=1,N
0049            13      S2=S2+OMEGA(KJ)*ISIGMA(KJ)
0050                    S2=S2*EPSNPR
0051                    RESD(II)=S1+S2-B(II)
0052                    WRITE (6,456) II,RESD(II)
0053            7       CONTINUE
                C
                C--     STEP 4.  CALCULATE THE MAXIMUM RESIDUAL AND THE
                C--     ROW FOR WHICH IT OCCURS.  CALL THAT ROW ALPHA.
                C--     ROW ALPHA WILL ENTER THE REFERENCE SET.
                C
0054                    BIG=ABS(RESD(N1))
0055                    DO 14 I=N1,M
0056                    IF (BIG-ABS(RESD(I))) 15,15,14
0057            15      BIG=ABS(RESD(I))
```

```
0058                    LALPHA=I
0059          14        CONTINUE
              C
              C--       IF MAXIMUM RESIDUAL IS EQUAL OR LESS THAN THE
              C--       CURRENT CHEBYSHEV AMPLITUDE WE HAVE REACHED THE
              C--       SOLUTION.  OUTPUT THAT SOLUTION.
              C
0060                    IF (ABS(EPSNPR-BIG).LE.EPZ) GO TO 99
              C
              C--       STEP 5.  FIND THE ROW WHICH SHOULD LEAVE THE
              C--       CURRENT REFERENCE SET.
              C--       TWO CASES OCCUR DEPENDING ON WHETHER ALPHA IS IN
              C--       FACT N+1 OR NOT.  IN THE FORMER CASE, SUBPROGRAM
              C--       'TRIAL' IS CALLED WHICH RETURNS THE REFERENCE
              C--       DEVIATIONS WHICH MAY BE EXPECTED IF ROW N+1
              C--       REPLACES EACH OF THE ROWS 1 THROUGH N.  IN THE
              C--       LATER CASE, SUBPROGRAM 'TRIAL1' IS CALLED WHICH
              C--       RETURNS THE REFERENCE DEVIATION WHICH WE CAN
              C--       ANTICIPATE IF ROW ALPHA REPLACES EACH OF THE
              C--       ROWS 1 THROUGH N.
              C
0061                    WRITE (6,557) LALPHA
0062                    DO 115 IBIG=1,N
0063                    LCH=IBIG
0064                    N2=N+2
0065                    N1=N+1
0066                    IF (LALPHA.EQ.N1) GO TO 123
0067                    CALL TRIAL1 (E1,LALPHA,IBIG,WJ,N)
0068                    GO TO 124
0069          123       CALL TRIAL (E1,LALPHA,N2,IBIG,N)
0070          124       EPSILN(IBIG)=E1
0071                    WRITE (6,558) IBIG,E1
0072          115       CONTINUE
              C
              C--       STEP 6.  FIND MAXIMUM OF REFERENCE DEVIATIONS
              C--       CALCULATED IN STEP 5 AND THE ROW FOR WHICH IT
              C--       OCCURRED.  CALL THAT ROW BETA.  ROW BETA SHOULD
              C--       NOW BE REPLACED BY ROW ALPHA IN THE REFERENCE
              C--       SET.
              C
0073                    BIG=EPSILN(1)
0074                    DO 19 I=1,N
0075                    IF (BIG-EPSILN(I)) 20,20,19
0076          20        BIG=EPSILN(I)
0077                    LPIVOT=I
0078          19        CONTINUE
0079                    CALL INVERS (LPIVOT,LALPHA,IFLAG,N)
0080                    IF (IFLAG.EQ.0) GO TO 991
```

```
            C---      ROW LPIVOT OF A IS TO REPLACE ROW LALPHA.
            C         INTERCHANGE ROWS LALPHA AND LPIVOT.
0081                  WRITE (6,616) LPIVOT
0082                  DO 220 II=1,N
0083        220       P(II)=A(LPIVOT,II)
0084                  DO 21 JJ=1,N
0085        21        A(LPIVOT,JJ)=A(LALPHA,JJ)
0086                  DO 22 KK=1,N
0087        22        A(LALPHA,KK)=P(KK)
0088                  PK=B(LPIVOT)
0089                  B(LPIVOT)=B(LALPHA)
0090                  B(LALPHA)=PK
            C
            C--       STEP 8.  ONE ITERATION IS NOW COMPLETE.  START
            C--       ALL OVER AGAIN.
            C
0091        9999      CONTINUE
0092                  WRITE (6,9998)
0093        9998      FORMAT (/,' ',' SOLUTION COULD NOT BE FOUND',//)
0094        99        CALL OUTPUT (EPSILN,ISIGMA,X,EPSNPR,N)
0095                  WRITE (6,611)
0096        456       FORMAT (/,5X,'RESIDUAL FOR ROW',15,'IS',F15.6,/)
0097        500       FORMAT (I10)
0098        501       FORMAT (4F10.5)
0099        610       FORMAT (//,25X,'SAMPLE OUTPUT FOR THE MEICLER
                      ALGORITHM',//)
0100        611       FORMAT (/,120(1H*),/)
0101        612       FORMAT (/,5X,'THE MATRIX OF COEFFICIENTS',/)
0102        613       FORMAT (/,5X,'THE RIGHT HAND SIDE VECTOR',/)
0103        614       FORMAT (/,5X,'ITERATION STEP',15,/)
0104        678       FORMAT (/,5X,'CHEBYSHEV REFERENCE DEVIATION FOR
                    1 THE CURRENT STEP IS',F15.6,/)
0105        557       FORMAT (/,5X,'ROW WITH MAXIMUM RESIDUAL IS',I5,/)
0106        558       FORMAT (/,5X,'REFERENCE DEVIATION IF ROW',15,
                    1 'LEAVES THE REFERENCE SET IS',F10.5,/)
0107        616       FORMAT (/,5X,'ROW TO LEAVE THE REFERENCE SET IS',
                    1 I5,/)
0108        676       FORMAT (/,5X,'THIS IS A SYSTEM OF',I5,3X,
                    1 'EQUATIONS IN',I5,4X,'UNKNOWNS',//)
0109        991       STOP
0110                  END
```

```
FORTRAN IV G LEVEL 21                    SIGN

0001                    SUBROUTINE SIGN (SUMA,IB,N)
              C
              C--       CALCULATE THE FUNCTION 'SIGN',
              C--       WHICH IS DEFINED AS SGN(X)=1, OR 0,OR -1 ACCORD-
              C--       ING AS X IS POSITIVE, ZERO, OR NEGATIVE RESPEC-
              C--       TIVELY.
              C
0002                    DIMENSION SUMA(10),IB(10)
0003                    DO 400 I=1,N
0004                    IF (SUMA(I)) 401,402,403
0005          402       IB(I)=0
0006                    GO TO 400
0007          401       IB(I)=-1
0008                    GO TO 400
0009          403       IB(I)=1
0010          400       CONTINUE
0011                    RETURN
0012                    END


FORTRAN IV G LEVEL 21                    OMEGN

0001                    SUBROUTINE OMEGN (OMEGA,IROW,N)
              C
              C--       CALCULATE W=V*DINVERSE.
              C
0002                    COMMON A(10,10),D(10,10),DINV(10,10),B(10)
0003                    DIMENSION OMEGA(10),XK(10)
0004                    N1=N+1
0005                    DO 605 KK=1,N
0006          605       XK(KK)=A(IROW,KK)
0007                    DO 601 I=1,N
0008                    OMEGA(I)=0.0
0009                    DO 602 J=1,N
0010          602       OMEGA(I)=OMEGA(I)+XK(J)*DINV(J,I)
0011          601       CONTINUE
0012                    RETURN
0013                    END
```

```
FORTRAN IV G LEVEL 21                    GETEPS

0001                    SUBROUTINE GETEPS (OMEGA,I1,ILAST,EPSILN,N)
              C
              C--       THIS PROCEDURE COMPUTES THE CHEBYSHEV REFERENCE
              C--       DEVIATION.
              C
0002                    COMMON A(10,10),D(10,10),DINV(10,10),B(10)
0003                    DIMENSION OMEGA(10),EPSILN(10)
0004                    DO 701 IBIG=I1,ILAST
0005                    CALL OMEGN (OMEGA,IBIG,N)
0006                    SUM=0.0
0007                    DO 702 I=1,N
0008          702       SUM=SUM+ABS(OMEGA(I))
0009                    X=0.0
0010                    DO 703 I=1,N
0011          703       X=X+OMEGA(I)*B(I)
0012                    XNEW=ABS(B(IBIG)-X)
0013                    EPSILN(IBIG)=XNEW/(1.0+SUM)
0014          701       CONTINUE
0015                    RETURN
0016                    END
```

```
FORTRAN IV G LEVEL 21                    OUTPUT

0001                    SUBROUTINE OUTPUT (EPSILN,ISIGMA,X,EPSNPR,N)
              C
              C--       OUTPUT THE CHEBYSHEV SOLUTION FOR THE PROBLEM
              C--       CURRENTLY UNDER INVESTIGATION.
              C
0002                    COMMON A(10,10),D(10,10),DINV(10,10),B(10)
0003                    DIMENSION EPSILN(10),X(10),ISGIMA(10),OMEGA(10)
0004                    N1=N+1
0005                    DO 800 I=1,N
0006          800       B(I)=B(I)+EPSNPR*ISIGMA(I)
0007                    DO 802 I=1,N
0008                    X(I)=0.0
0009                    DO 803 J=1,N
0010          803       X(I)=X(I)+DINV(I,J)*B(J)
0011          802       CONTINUE
0012                    WRITE (6,852)
0013                    WRITE (6,850) (X(I),I=1,N)
0014          850       FORMAT (4F10.5)
0015          852       FORMAT (/,5X,'CHEBYSHEV SOLUTION FOR THIS PROBLEM
                     1  IS GIVEN BELOW',/)
0016                    RETURN
0017                    END
```

```
FORTRAN IV G LEVEL 21                    AMAX

0001                    SUBROUTINE AMAX (X,TOP,LM,I1,ILAST)
0002                    DIMENSION X(10)
0003                    TOP=X(I1)
0004                    LM=I1
0005                    DO 444 I=I1,ILAST
0006                    IF (TOP-X(I)) 445,445,444
0007          445       TOP=X(I)
0008                    LM=I
0009          444       CONTINUE
0010                    RETURN
0011                    END
```

```
FORTRAN IV G LEVEL 21                    TRIAL

0001                    SUBROUTINE TRIAL (ETR,LAL,MAXE,II,N)
              C
              C--       COMPUTE THE FERENCE DEVIATION THAT CAN BE EXPECTED
              C--       IF ROW N+1 OF THE CURRENT REFERENCE SET ENTERS
              C--       THE REFERENCE SUBSYSTEM.
              C
0002                    COMMON A(10,10),D(10,10),DINV(10,10),B(10)
0003                    DIMENSION PR(10),BB(10,10),W(10)
0004                    DO 700 I=1,N
0005                    DO 701 J=1,N
0006                    IF (I.EQ.II) GO TO 702
0007                    BB(I,J)=A(I,J)
0008                    PR(I)=B(I)
0009                    GO TO 701
0010          702       BB(I,J)=A(LAL,J)
0011                    PR(I)=B(LAL)
0012          701       CONTINUE
0013          700       CONTINUE
0014                    CALL INVERT (BB,N,10,ISING)
0015                    DO 703 I=1,N
0016                    W(I)=0.0
0017                    DO 704 J=1,N
0018          704       W(I)=W(I)+A(MAXE,J)*BB(J,K)
0019          703       CONTINUE
0020                    SUM=0.0
0021                    DO 706 K=1,N
0022          706       SUM=SUM+W(K)*PR(K)
0023                    ETR=ABS(B(MAXE)-SUM)
0024                    DR=0.0
0025                    DO 705 KJ=1,N
0026          705       DR=DR+ABS(W(KJ))
```

```
FORTRAN IV G LEVEL 21                    TRIAL

0027                    DR=DR+1.0
0028                    ETR=ETR/DR
0029                    RETURN
0030                    END


FORTRAN IV G LEVEL 21                    RESDLQ

0001                    SUBROUTINE RESDLQ (DLQ,LRMAX,N)
                C
                C--     COMPUTE THE LEAST SQUARES RESIDUAL
                C
0002                    COMMON A(10,10),D(10,10),DINV(10,10),B(10)
0003                    DIMENSION P(10),DLQ(10)
0004                    NN=N+1
0005                    CALL OMEGN (P,LRMAX,N)
0006                    SUM=0.0
0007                    DO 600 I=1,N
0008            600     SUM=SUM+P(I)*P(I)
0009                    ALPHA=1.0/(1.0+SUM)
0010                    TEMP=0.0
0011                    DO 601 I=1,N
0012            601     TEMP=TEMP+P(I)*B(I)
0013                    TOP+B(LRMAX)-TEMP
0014                    FACT=ALPHA*TOP
0015                    DLQ(NN)=-1.0*FACT
0016                    DO 604 I=1,N
0017            604     DLQ(I)=FACT*P(I)
0018                    RETURN
0019                    END


FORTRAN IV G LEVEL 21                    TRIAL1

0001                    SUBROUTINE TRIAL1 (EPS,LAL,JAL,W,N)
                C
                C--     THIS PROCEDURE COMPUTES THE MINIMAX REFERENCE
                C--     DEVIATION IF ANY ROW N+2 THROUGH M ENTERS THE
                C--     REFERENCE SUBSYSTEM AT ANY ITERATION STEP.
                C
0002                    COMMON A(10,10),D(10,10),DINV(10,10),B(10)
0003                    DIMENSION P(10),W(10)
0004                    NN=N+1
0005                    CALL OMEGN (W,NN,N)
0006                    XLAMDA=0.0
```

FORTRAN IV G LEVEL 21 TRIAL 1

```
0007                    DO 700 I=1,N
0008          700       XLAMDA=XLAMDA+A(LAL,K)*DINV(I,JAL)
0009                    WJ=W(JAL)/XLAMDA
0010                    BETA=ABS(WJ)
0011                    SUM=0.0
0012                    DO 701 I=1,N
0013                    IF (I.EQ.JAL) GO TO 701
0014                    TEMP=0.0
0015                    DO 702 J=1,N
0016          702       TEMP=TEMP+A(LAL,J)*DINV(J,I)
0017                    SUM=SUM+ABS(W(I)-BETA*TEMP)
0018          701       CONTINUE
0019                    GAMA=BETA*SUM
0020                    SUM=0.0
0021                    DO 703 I=1,N
0022                    SUM=SUM+W(I)*B(I)
0023          703       CONTINUE
0024                    VN=B(NN)-SUM
0025                    CALL OMEGN (W,LAL,N)
0026                    PN=0.0
0027                    DO 704 I=1,N
0028          704       PN=PN+W(I)*B(I)
0029                    PN=B(LAL)-PN
0030                    TOP=ABS(VN-BETA*PN)
0031                    EPS=TOP/(1.0+GAMA)
0032                    RETURN
0033                    END
```

FORTRAN IV G LEVEL 21 INVERS

```
0001                    SUBROUTINE INVERS (LBETA,LV,IFLAG,N)
                C
                C--     THIS PROCEDURE COMPUTES THE INVERSE OF A MATRIX
                C--     D IF A ROW, SAY, P, REPLACES ONE ROW OF D.  THE
                C--     INVERSE OF D AFTER THIS REPLACEMENT IS MADE IS
                C--     CALCULATED FROM THE INVERSE OF D, USING A FORMULA
                C--     DUE TO E. W. CHENY.  (SEE FOR EXAMPLE CHENY
                C--     'APPROXIMATION THEORY', PAGE 49.
                C
0002                    COMMON A(10,10),D(10,10),DINV(10,10),B(10)
0003                    DIMENSION V(10)
0004                    XL=0.0
0005                    IFLAG=1
0006                    DO 250 I=1,N
0007          250       XL=XL+A(LV,I)*DINV(I,LBETA)
```

FORTRAN IV G LEVEL 21 INVERS

```
0008                    IF (XL) 252,251,252
0009          251       IFLAG=0
0010                    GO TO 269
0011          252       DO 253 I=1,N
0012          253       DINV(I,LBETA)=DINV(I,LBETA)/XL
0013                    DO 254 J=1,N
0014                    IF (J.EQ.LBETA) GO TO 254
0015                    SUM=0.0
0016                    DO 255 I=1,N
0017                    SUM=SUM+A(LV,I)*DINV(I,J)
0018          255       CONTINUE
0019                    V(J)=SUM
0020          254       CONTINUE
0021                    DO 257 K=1,N
0022                    IF (K.EQ.LBETA) GO TO 257
0023                    DO 256 KJ=1,N
0024                    DINV(KJ,K)=DINV(KJ,K)=V(K)*DINV(KJ,LBETA)
0025          256       CONTINUE
0026          257       CONTINUE
0027          269       RETURN
0028                    END
```

APPENDIX C

SAMPLE OUTPUT FOR GOLUB-BARTELLS ALGORITHM

```
**************************************************************************

 THE MATRIX OF COEFFICIENTS


2.00000   1.00000
3.00000   1.00000
1.00000   2.00000
1.00000   1.00000
1.00000  -1.00000

  THE RIGHT HAND SIDE VECTOR


6.90000   7.20000
7.00000   3.00000
1.00000

  THIS IS A SYSTEM OF    5    EQUATIONS IN    2    UNKNOWNS

**************************************************************************

  ITERATION STEP          1

  THE LAMDA VECTOR FOR THIS STEP IS GIVEN BELOW

    -0.84746         0.50847         0.16949

  THE CHEBYSHEV RESIDUAL FOR THIS REFERENCE         SUBSYSTEM =   0.65556

  CHEBYSHEV SOLUTION FOR THIS REFERENCE SUBSYSTEM   FOLLOWS

 1.611110687     3.022222519

  RESIDUAL FOR ROWS N+2 THROUGH M FOLLOWS

 1.633333206   -2.411111832
```

ROW TO ENTER THE REFERENCE SET IS ROW 5

ROW TO LEAVE THE REFERENCE SET IS ROW 3

****

ITERATION STEP 2

THE LAMDA VECTOR FOR THIS STEP IS GIVEN BELOW

-0.57143 0.42857 -0.14286

THE CHEBYSHEV RESIDUAL FOR THIS REFERENCE SUBSYSTEM = 0.87500

CHEBYSHEV SOLUTION FOR THIS REFERENCE SUBSYSTEM FOLLOWS

2.050000191 1.924998283

RESIDUAL FOR ROWS N+2 THROUGH M FOLLOWS

0.974998474 -1.100003242

ROW TO ENTER THE REFERENCE SET IS ROW 5

ROW TO LEAVE THE REFERENCE SET IS ROW 1

****

ITERATION STEP 3

THE LAMDA VECTOR FOR THIS STEP IS GIVEN BELOW

-0.35088 0.26316 -0.43860

THE CHEBYSHEV RESIDUAL FOR THIS REFERENCE SUBSYSTEM = 0.95000

CHEBYSHEV SOLUTION FOR THIS REFERENCE SUBSYSTEM FOLLOWS

2.049999237 1.999999046

```
 RESIDUAL FOR ROWS N+2 THROUGH M FOLLOWS
1.049998283    -0.800002098

 ROW TO ENTER THE REFERENCE SET IS ROW      4

 ROW TO LEAVE THE REFERENCE SET IS ROW     2

**************************************************************************

 ITERATION STEP          9

 THE LAMDA VECTOR FOR THIS STEP IS GIVEN BELOW

   -0.33333         0.50000        -0.16667

 THE CHEBYSHEV RESIDUAL FOR THIS REFERENCE          SUBSYSTEM =  1.00000

 CHEBYSHEV SOLUTION FOR THIS REFERENCE SUBSYSTEM      FOLLOWS
2.000000000     2.000000000
 RESIDUAL FOR ROWS N+2 THROUGH M FOLLOWS
0.800000191    -0.899999619

 ROW TO ENTER THE REFERENCE SET IS ROW      5

 CHEBYSHEV SOLUTION FOR THIS PROBLEM IS GIVEN       BELOW
2.000000000     2.000000000

**************************************************************************
```

SAMPLE OUTPUT FOR THE MEICLER ALGORITHM

```
**************************************************************************

  THE MATRIX OF COEFFICIENTS

2.00000   1.00000
3.00000   1.00000
1.00000   2.00000
1.00000   1.00000
1.00000  -1.00000

  THE RIGHT HAND SIDE VECTOR

6.90000   7.20000
7.00000   3.00000
1.00000

  THIS IS A SYSTEM OF    5    EQUATIONS IN    2    UNKNOWNS

**************************************************************************

-1.00000   1.00000
 3.00000  -2.00000

**************************************************************************

  ITERATION STEP    1

  CHEBYSHEV REFERENCE DEVIATION FOR THE CURRENT       STEP IS   0.900000

  RESIDUAL FOR ROW    3IS      -1.300014

  RESIDUAL FOR ROW    4IS       0.899993

  RESIDUAL FOR ROW    5IS      -0.699995

  ROW WITH MAXIMUM RESIDUAL IS    3
```

```
 REFERENCE DEVIATION IF ROW     1LEAVES THE   REFERENCE SET IS  0.77500

 REFERENCE DEVIATION IF ROW     2LEAVES THE   REFERENCE SET IS  0.98000

 ROW TO LEAVE THE REFERENCE SET IS     2

**************************************************************************

 ITERATION STEP    2

 CHEBYSHEV REFERENCE DEVIATION FOR THE CURRENT     STEP IS      0.980000

 RESIDUAL FOR ROW    3IS       0.659998

 RESIDUAL FOR ROW    4IS       0.980000

 RESIDUAL FOR ROW    5IS      -1.100004

 ROW WITH MAXIMUM RESIDUAL IS    5

 REFERENCE DEVIATION IF ROW     1LEAVES THE   REFERENCE SET IS  0.95000

 REFERENCE DEVIATION IF ROW     2LEAVES THE   REFERENCE SET IS  0.87500

 ROW TO LEAVE THE REFERENCE SET IS     1

**************************************************************************

 ITERATION STEP    3

 CHEBYSHEV REFERENCE DEVIATION FOR THE CURRENT     STEP IS      1.000001

 RESIDUAL FOR ROW    3IS       0.800000
```

RESIDUAL FOR ROW 4IS 1.000000

RESIDUAL FOR ROW 5IS 00.899999

CHEBYSHEV SOLUTION FOR THIS PROBLEM IS GIVEN BELOW

2.00000 2.00000

****

SAMPLE OUTPUT FOR GOLUB-BARTELLS ALGORITHM

****

THE MATRIX OF COEFFICIENTS

1.00000	0.0	0.0	0.0
0.0	1.00000	0.0	0.0
0.0	0.0	1.00000	0.0
0.0	0.0	0.0	1.00000
1.00000	1.00000	1.00000	2.00000
1.00000	-1.00000	2.00000	4.00000
3.00000	1.00000	7.00000	11.00000

THE RIGHT HAND SIDE VECTOR

2.00000 3.00000 1.00000 1.00000
7.50000 6.80000 26.00000

THIS IS A SYSTEM OF 7 EQUATIONS IN 4 UNKNOWNS

****

ITERATION STEP 1

THE LAMDA VECTOR FOR THIS STEP IS GIVEN BELOW

-2.00000 -2.00000 -2.00000 -4.00000 2.00000

THE CHEBYSHEV RESIDUAL FOR THIS REFERENCE SUBSYSTEM = 0.08333

CHEBYSHEV SOLUTION FOR THIS REFERENCE SUBSYSTEM FOLLOWS

1.916665077 2.916665077 0.916666448 0.916666448

RESIDUAL FOR ROWS N+2 THROUGH M FOLLOWS

-2.300002098 -0.833358765

```
  ROW TO ENTER THE REFERENCE SET IS ROW     6

  ROW TO LEAVE THE REFERENCE SET IS ROW     3

**************************************************************************

  ITERATION STEP          2

  THE LAMDA VECTOR FOR THIS STEP IS GIVEN BELOW

    -0.35714        -1.07143        -0.35714         0.00000         0.71429

  THE CHEBYSHEV RESIDUAL FOR THIS REFERENCE          SUBSYSTEM =  0.40000

  CHEBYSHEV SOLUTION FOR THIS REFERENCE SUBSYSTEM    FOLLOWS

 1.599997520     2.599996567     0.899997711     1.399999619

  RESIDUAL FOR ROWS N+2 THROUGH M FOLLOWS

-0.100002289     3.099960327

  ROW TO ENTER THE REFERENCE SET IS ROW     7

  ROW TO LEAVE THE REFERENCE SET IS ROW     1

**************************************************************************

  ITERATION STEP          3

  THE LAMDA VECTOR FOR THIS STEP IS GIVEN BELOW

     0.11494        -0.68966        -0.45977         0.34483         0.11494

  THE CHEBYSHEV RESIDUAL FOR THIS REFERENCE          SUBSYSTEM =  0.58000
```

CHEBYSHEV SOLUTION FOR THIS REFERENCE SUBSYSTEM FOLLOWS

2.680000305 2.419987679 -0.179996014 1.579997063

RESIDUAL FOR ROWS N+2 THROUGH M FOLLOWS

-1.179995537 0.680000305

ROW TO ENTER THE REFERENCE SET IS ROW 6

ROW TO LEAVE THE REFERENCE SET IS ROW 5

****

ITERATION STEP 4

THE LAMDA VECTOR FOR THIS STEP IS GIVEN BELOW

0.15625 -0.62500 -0.46875 0.15625 -0.15625

THE CHEBYSHEV RESIDUAL FOR THIS REFERENCE SUBSYSTEM = 0.64000

CHEBYSHEV SOLUTION FOR THIS REFERENCE SUBSYSTEM FOLLOWS

1.240003586 2.359996796 0.360001087 1.639998436

RESIDUAL FOR ROWS N+2 THROUGH M FOLLOWS

-0.260002136 -0.759996414

ROW TO ENTER THE REFERENCE SET IS ROW 7

ROW TO LEAVE THE REFERENCE SET IS ROW 4

****

****

ITERATION STEP 5

THE LAMDA VECTOR FOR THIS STEP IS GIVEN BELOW

0.16807 -0.63025 -0.46219 -0.04202 -0.25210

THE CHEBYSHEV RESIDUAL FOR THIS REFERENCE SUBSYSTEM = 0.64324

CHEBYSHEV SOLUTION FOR THIS REFERENCE SUBSYSTEM FOLLOWS

1.356753349 2.356742859 0.356760025 1.610808372

RESIDUAL FOR ROWS N+2 THROUGH M FOLLOWS

-0.208127022 0.610808372

ROW TO ENTER THE REFERENCE SET IS ROW 7

CHEBYSHEV SOLUTION FOR THIS PROBLEM IS GIVEN BELOW

1.356753349 2.356742859 0.356760025 1.610808372

****

SAMPLE OUTPUT FOR THE MEICLER ALGORITHM

```
**************************************************************************

  THE MATRIX OF COEFFICIENTS

1.00000    0.0        0.0        0.0
0.0        1.00000    0.0        0.0
0.0        0.0        1.00000    0.0
0.0        0.0        0.0        1.00000
1.00000    1.00000    1.00000    2.00000
1.00000   -1.00000    2.00000    4.00000
3.00000    1.00000    7.00000   11.00000

  THE RIGHT HAND SIDE VECTOR

2.00000    3.00000    1.00000    1.00000
7.50000    6.80000   26.00000

  THIS IS A SYSTEM OF     7     EQUATIONS IN     4     UNKNOWNS

**************************************************************************

1.00000    0.0        0.0        0.0
0.0        1.00000    0.0        0.0
0.0        0.0        1.00000    0.0
0.0        0.0        0.0        1.00000

**************************************************************************

  ITERATION STEP      1

  CHEBYSHEV REFERENCE DEVIATION FOR THE CURRENT      STEP IS      0.200000

  RESIDUAL FOR ROW     5IS        1.099999

  RESIDUAL FOR ROW     6IS       -0.200001

  RESIDUAL FOR ROW     7IS        4.999985

  ROW WITH MAXIMUM RESIDUAL IS     7
```

```
REFERENCE DEVIATION IF ROW     1LEAVES THE   REFERENCE SET IS  0.03333

REFERENCE DEVIATION IF ROW     2LEAVES THE   REFERENCE SET IS  0.02632

REFERENCE DEVIATION IF ROW     3LEAVES THE   REFERENCE SET IS  0.11905

REFERENCE DEVIATION IF ROW     4LEAVES THE   REFERENCE SET IS  0.11667

ROW TO LEAVE THE REFERENCE SET IS     3

**************************************************************************

ITERATION STEP     2

CHEBYSHEV REFERENCE DEVIATION FOR THE CURRENT       STEP IS     0.584000

RESIDUAL FOR ROW    5IS       0.523999

RESIDUAL FOR ROW    6IS      -0.584002

RESIDUAL FOR ROW    7IS      -1.143998

ROW WITH MAXIMUM RESIDUAL IS    7

REFERENCE DEVIATION IF ROW     1LEAVES THE   REFERENCE SET IS  0.09200

REFERENCE DEVIATION IF ROW     2LEAVES THE   REFERENCE SET IS  0.05862

REFERENCE DIVIATION IF ROW     3LEAVES THE   REFERENCE SET IS  0.08333

REFERENCE DIVIATION IF ROW     4LEAVES THE   REFERENCE SET IS  0.13377

ROW TO LEAVE THE REFERENCE SET IS     4

**************************************************************************
```

****

ITERATION STEP 3

CHEBYSHEV REFERENCE DEVIATION FOR THE CURRENT STEP IS 0.643244

RESIDUAL FOR ROW 5IS -0.208112

RESIDUAL FOR ROW 6IS -0.643249

RESIDUAL FOR ROW 7IS 0.610809

CHEBYSHEV SOLUTION FOR THIS PROBLEM IS GIVEN BELOW

1.35676 2.35676 0.35676 1.61081

****

VITA

Eayo Ouseph was born in Kerala, India, on June 14, 1945. He completed high school and attended University of Kerala, receiving the B.Sc. degree in Physics and Mathematics in March 1966, with the highest honors (first in the University). He entered the University of Illinois at Urbana in September 1969 and received an M.S. degree in Physics in January 1971. He was a graduate student at the University of Tennessee, Knoxville, from September 1972 to March 1974 and received an M.S. in Computer Science.