CS 67

COMPUTATIONAL CONSIDERATIONS REGARDING THE CALCULATION OF CHEBYSHEV SOLUTIONS FOR OVERDETERMINED LINEAR EQUATION SYSTEMS BY THE EXCHANGE METHOD

BY

RICHARD H. BARTELS
GENE H. GOLUB

TECHNICAL REPORT NO. CS 67
JUNE 2, 1967

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

COMPUTATIONAL CONSIDERATIONS REGARDING THE CALCULATION OF CHEBYSHEV SOLUTIONS FOR OVERDETERMINED LINEAR EQUATION SYSTEMS BY THE EXCHANGE METHOD*

By

Richard H. Bartels

Gene H. Golub

*This project was supported in part by NSF and ONR.

Abstract

An implementation, using Gaussian LU decomposition with row interchanges, of Stiefel's exchange algorithm for determining a Chebyshev solution to an overdetermined system of linear equations is presented. The implementation is computationally more stable than those usually given in the literature. A generalization of Stiefel's algorithm is developed which permits the occasional exchange of two equations simultaneously. Finally, some experimental comparisons are offered.

Table of Contents

1. Introduction

The problem of finding a vector $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_n)$ which solves an overdetermined system of equations

$$r_i(x) \equiv \sum_{j=1}^{n} a_{ij} x_j - d_i = 0 \qquad (i=1,\ldots, m; \text{ where } m > n)$$

in the sense that

$$\max_{1 \leq i \leq m} |r_i(\hat{x})| \leq \max_{1 \leq i \leq m} |r_i(x)|$$

for any $x \in E^n$ is treated by Stiefel in [1]. Such an $\hat{x}$ is called a Chebyshev or minimax solution to the system.

Given an overdetermined system of linear equations $Ax = d$ whose matrix of coefficients satisfies the Haar condition (each $n \times n$ submatrix is nonsingular), Stiefel presents in [1] an algorithm called the exchange method for finding a Chebyshev solution. In a later paper, [2], the exchange method is shown to be equivalent to the simplex method applied to a suitable linear programming problem.

In this regard, Stiefel suggests the use of techniques drawn from the simplex method for the implementation of his algorithm. These techniques are characterized by their use of Jordan elimination, for the most part without row or column interchanges to pick the most advantageous pivots, for solving linear equation systems which arise during the computation. These methods are fast but computationally unstable. In this paper we propose a computational scheme based upon the more stable method of Gaussian LU-decomposition using row interchanges. Attention is paid to the peculiarities of the exchange method to make computation as fast as possible.

Afterwards a generalization of Stiefel's algorithm is presented which permits the occasional exchange of two equations at once.

Finally some experimental comparisons of selection rules for use with the exchange method are tabulated.

2. Background Theory

There is a full treatment of the theory and the exchange method in Chapter 2 of [9]. (The exchange method is called the ascent algorithm in this work.) We therefore confine ourselves in this section and the next to a statement of pertinent results, omitting proofs.

According to corollary 7.4.7., page 410, of [4], any overdetermined system of linear equations has a Chebyshev solution. The following lemma and theorem serve to characterize these solutions.

Lemma: Let $B = [b_{ij}]$ be a $p \times q$ matrix with rows $B_1, \ldots, B_p$.

There is a vector $y = (y_1, \ldots, y_q)$ such that

$$\sum_{j=1}^{q} b_{ij} y_j < 0 \qquad \text{for all } i=1, \ldots, p$$

if and only if $0 \neq \sum_{i=1}^{p} \alpha_i B_i$ for all nontrivial choices of of nonnegative scalars $\alpha_1, \ldots, \alpha_p$.

This lemma is a special case of corollary 6, page 115, of [5].

Let $Ax = d$ be an overdetermined system of m linear equations in n unknowns. For any vector $x = (x_1, \ldots, x_n)$, denote the residuals

$$\sum_{j=1}^{n} a_{ij}x_j - d_i \qquad (i=1,\ldots, m) \text{ by } r_i(x) .$$

Let A_i be the i^{th} row of the matrix A.

Given any fixed vector, $v = (v_1, \ldots, v_n)$, we may assume with no loss of generality that the equations have been ordered and numbered so that

$$\max_{1 \leq i \leq m} |r_i(v)| = |r_1(v)| = \ldots = |r_k(v)| > |r_{k+1}(v)| \geq \cdots \geq |r_m(v)| ,$$

where $1 \leq k \leq m$.

Theorem: There is a vector z for which

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

if and only if $0 \neq \sum_{i=1}^{k} w_i \operatorname{sgn}(r_i(v))A_i$ for all nontrivial choices of nonnegative scalars $w_1, \ldots, w_k$.

For the purposes of the exchange method we restrict our attention henceforth to overdetermined systems of m linear equations in n unknowns, $Ax = d$, for which rank$(A) = n$.

To begin, suppose that $m = n+1$. There is no loss of generality in assuming that the equations have been ordered so that the first n rows, $A_1, \ldots, A_n$, of A are linearly independent. Thus, scalars $\lambda_1, \ldots, \lambda_{n+1}$ can be found with $\lambda_{n+1} \neq 0$ such that

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

Denoting $\mathrm{sgn}(\lambda_i)$ by s_i, set $\epsilon = - \dfrac{\sum_{i=1}^{n+1} \lambda_i d_i}{\sum_{j=1}^{n+1} |\lambda_j|}$, and solve the system

$$\begin{bmatrix} A_1 \\ \vdots \\ A_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} + \epsilon \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix} \ .$$

So $r_i(x) = \sum_{j=1}^{n} a_{ij} x_j - d_i = s_i \epsilon$ for $i = 1, \ldots, n$.

Furthermore, $r_{n+1}(x) = s_{n+1}\epsilon$, as can easily be shown.

Therefore, $\mathrm{sgn}(r_i(x)) = s_i \ \mathrm{sgn}(\epsilon)$ for all i .

But $$0 = \sum_{i=1}^{n+1} \lambda_i A_i = \sum_{i=1}^{n+1} |\lambda_i| s_i A_i \ .$$

And so $0 = \mathrm{sgn}(\epsilon)\ 0 = \sum_{i=1}^{n+1} |\lambda_i| s_i \ \mathrm{sgn}(\epsilon) A_i = \sum_{i=1}^{n+1} |\lambda_i| \mathrm{sgn}(r_i(x)) A_i$.

Hence, by the preceding theorem, $x = (x_1, \ldots, x_n)$ is a Chebyshev solution for the given system. (For an alternate discussion of $(n+1) \times n$ systems see [6].)

Returning to the general case $(m \geq n+1)$, suppose for some set of $n+1$ rows of A the first n of which are linearly independent (with complete generality, the first $n+1$ rows of A) we construct the

Chebyshev solution x as above and find that, under correct ordering of equations $n+2$ through m,

$$|r_1(x)| = \cdots = |r_{n+1}(x)| \geq |r_{n+2}(x)| \geq \cdots \geq |r_m(x)| \;.$$

Then x is a Chebyshev solution to the full given system.

We further note that the value

$$|\epsilon| = \max_{1\leq i\leq n+1} |r_i(x)| = \max_i |r_i(x)|$$

must be greater than the value $\inf\limits_{y\epsilon E^n} \max\limits_{1\leq j\leq n+1} |r_{k_j}(y)|$ for any other collection of $n+1$ rows $A_{k_1}, \ldots, A_{k_{n+1}}$ from the matrix, since

$$\inf_y \max_{1\leq j\leq n+1} |r_{k_j}(y)| \leq \max_{1\leq j\leq n+1} |r_{k_j}(x)| \quad (x \text{ as above})$$

$$\leq \max_{1\leq i\leq m} |r_i(x)| = |\epsilon| \;.$$

Following the convention put forth in [1], any subsystem

$$\begin{bmatrix} A_{i_1} \\ \vdots \\ A_{i_{n+1}} \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} d_{i_1} \\ \vdots \\ d_{i_{n+1}} \end{bmatrix}$$

of the given system with rank $\left(\begin{bmatrix} A_{i_1} \\ \vdots \\ A_{i_{n+1}} \end{bmatrix}\right) = n$ will be called a <u>reference subsystem</u>, and the rows $A_{i_1}, \ldots, A_{i_{n+1}}$ will be called a <u>reference set</u>. If $x = (x_1, \ldots, x_n)$ is a Chebyshev solution to a reference subsystem, the value

$$|\epsilon| = |r_{i_1}(x)| = \ldots = |r_{i_{n+1}}(x)| = \inf_{y \epsilon E^n} \max_{1 \leq j \leq n+1} |r_{i_j}(y)|$$

will be called the <u>reference deviation</u> for the reference subsystem. It is uniquely determined by the reference subsystem.

3. <u>The Exchange Method</u>

Stiefel's algorithm consists of starting with a reference subsystem and modifying it one equation at a time so as to increase the reference deviation by each change. Each modification proceeds as follows:

We may assume that $A_1, \ldots, A_{n+1}$ is a reference set. Let $x = (x_1, \ldots, x_n)$ be a Chebyshev solution to the corresponding reference subsystem computed as above. So we have $\epsilon, \lambda_1, \ldots, \lambda_{n+1}$ which satisfy

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

$$\text{b)} \quad \epsilon = - \frac{\sum_{i=1}^{n+1} \lambda_i d_i}{\sum_{j=1}^{n+1} |\lambda_j|}$$

c) $r_i(x) = s_i\epsilon$ for $i = 1,\ldots, n+1$,

where $s_i = \mathrm{sgn}(\lambda_i)$.

If x is not a Chebyshev solution to the full given system, then by the discussion in the previous section, there is an $\alpha\epsilon\{n+2,\ldots, m\}$ for which $|r_\alpha(x)| > |\epsilon|$. Let $\rho_1,\ldots, \rho_{n+1}$ be scalars for which

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

In order to proceed, we impose

<u>Condition 1:</u> $\lambda_i \neq 0$ for all $i = 1,\ldots, n+1$.

If this holds, let $\beta\epsilon\{1,\ldots, n+1\}$ be such that

$$\frac{\sigma_\alpha s\rho_\beta}{\lambda_\beta} = \max_{1\leq i\leq n+1} \frac{\sigma_\alpha s\rho_i}{\lambda_i},$$

where $\sigma_\alpha = \mathrm{sgn}(r_\alpha(x))$, and $s = \mathrm{sgn}(\epsilon)$.

Now impose

<u>Condition 2:</u> $A_1,\ldots, A_{\beta-1}, A_{\beta+1},\ldots, A_{n+1}, A_\alpha$ are a reference set.

We form a Chebyshev solution $x' = (x_1',\ldots, x_n')$ to the reference subsystem

$$\begin{bmatrix} A_1 \\ \vdots \\ A_{\beta-1} \\ A_{\beta+1} \\ \vdots \\ A_{n+1} \\ A_\alpha \end{bmatrix} \begin{bmatrix} z_1 \\ \cdot \\ \cdot \\ \cdot \\ z_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_{\beta-1} \\ d_{\beta+1} \\ \vdots \\ d_{n+1} \\ d_\alpha \end{bmatrix}$$ in the usual fashion,

producing $\epsilon', \lambda_1', \ldots, \lambda_{\beta-1}', \lambda_{\beta+1}', \ldots, \lambda_{n+1}', \lambda_\alpha'$ such that

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

$$\text{b}')\quad \epsilon' = - \frac{\displaystyle\sum_{\substack{i=1\\ i\neq\beta}}^{n+1} \lambda_i' d_i + \lambda_\alpha' d_\alpha}{|\lambda_\alpha'| + \displaystyle\sum_{\substack{j=1\\ j\neq\beta}}^{n+1} |\lambda_j'|}$$

$$\text{c}')\quad r_i(x') = s_i'\epsilon' \quad \text{for} \quad i=1,\ldots, \beta-1, \beta+1,\ldots, n+1, \alpha$$

where $s_i' = \operatorname{sgn}(\lambda_i')$.

We further have

$$\lambda_i' = \sigma_\alpha \lambda_\alpha' \lambda_i \left[\frac{\rho_\beta \sigma_\alpha}{\lambda\beta} - \frac{\rho_i \sigma_\alpha}{\lambda_i} \right] \quad (i=1,\ldots, n+1;\ i\neq\beta) .$$

(Note that, by the choice of β, the product of the term in brackets with $s = \operatorname{sgn}(\epsilon)$ is nonnegative.)

Furthermore,

$$\text{if} \quad K = |\lambda_\alpha'| + \sum_{\substack{k=1\\ k\neq\beta}}^{n+1} |\lambda_k'| \quad \text{and} \quad c = \frac{|\lambda_\alpha'|}{K} ,$$

it can readily be shown that

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

It is important to note that, if condition 1 is satisfied by the second reference set (i.e., $\lambda_i' \neq 0$ for $i=1,\ldots, \beta-1, \beta+1,\ldots, n+1, \alpha$), then $c > 0$. Therefore $|\epsilon'| > |\epsilon|$, since $|r_\alpha(x)| > |\epsilon|$. The strictness of the inequality $|\epsilon'| > |\epsilon|$ implies, by a simple contradiction argument, that if an initial reference set is chosen and subsequently modified as above by exchanging successive non-reference set rows of the matrix A for rows in the reference set, and if conditions 1 and 2 hold at each exchange, the process must converge upon a Chebyshev solution for the full system.

4. Jordan Elimination

An excellent example of an implementation of the exchange method which uses Jordan elimination is given on page 50 of [9].

Briefly, given indices $\{i_1,\ldots, i_{n+1}\} \subseteq \{1,\ldots, m\}$, numbers $\lambda_1,\ldots, \lambda_{n+1}$ are found so that

$$\sum_{k=1}^{n+1} \lambda_k = 1$$

and

$$\sum_{k=1}^{n+1} \lambda_k A_k = 0 .$$

Setting $s_k = \operatorname{sgn}(\lambda_k)$ for $k=1,\ldots, n+1$, the matrix

$$C = \left[\begin{array}{c|c|c} A^T_{i_1} & \cdots & A^T_{i_{n+1}} \\ \hline s_1 & \cdots & s_{n+1} \end{array}\right]^{-1}$$

is formed using a sequence of n+1 pivot operations (Jordan elimination steps).

Each exchange step, then, involves forming

$$[x_1, \ldots, x_n, \epsilon] = [d_{i_1}, \ldots, d_{i_{n+1}}]C,$$

computing

$$r_j = \sum_{k=1}^{n+1} a_{jk} x_k - d_j \quad \text{for all} \quad j \neq i_1, \ldots, i_{n+1},$$

selecting α so that $|r_\alpha| = \max$, and forming

$$[\rho_1, \ldots, \rho_{n+1}] = [a_{\alpha,1}, \ldots, a_{\alpha,n}, \operatorname{sgn}(r_\alpha)]C^T .$$

The last column of C has the form

$$\begin{bmatrix} \lambda_1/G \\ \lambda_2/G \\ \vdots \\ \lambda_{n+1}/G \end{bmatrix},$$

where

$$G = \sum_{k=1}^{n+1} |\lambda_k| .$$

Hence, β is selected as an index for which

$$\operatorname{sgn}(r_\alpha)\ \operatorname{sgn}(\epsilon)\ \rho_\beta / C_{\beta,n+1} = \max .$$

An appropriate pivot operation on C ends the exchange step. The λ_k can be found in

$$\frac{n^3}{3} + 2n^2 + \frac{8n}{3} + 1$$

operations (counting only multiplications and divisions), and the initial computation of C requires an additional $n^3 + 3n^2 + n$ operations.

In each exchange step the quantities

$$x_1, \ldots, x_{n+1}, \epsilon, \rho_1, \ldots, \rho_{n+1}$$

require $2n^2 + 4n + 2$ operations to compute, and the updating of C demands an additional $n^2 + 2n + 1$ operations. Hence, k exchanges may be carried out with

$$\frac{4n^3}{3} + (3k + 5)n^2 + (6k + \frac{11}{3})n + 3k + 1$$

operations.

While row and column interchanges can be permitted during the initial sequence of Jordan elimination steps which forms C, so that pivot elements of largest possible magnitude can be selected, no pivot choice is possible during the subsequent updatings of C . For simple examples of the danger implicit in this fact see [10,11]. The danger is studied at greater depth in [3,7,8].

5. LU Decomposition

Starting from any reference subsystem of the given overdetermined system, the exchange method produces a new reference subsystem at the cost of solving three nonsingular sets of $n+1$ linear equations:

$$P\lambda = r_1$$

$$P^T x = r_2$$

$$P\rho = r_3 \; .$$

The vector r_1 is given, but r_2 depends upon λ and r_3 depends upon x. If three such systems of equations were given in isolation, the general method of solution would consist of making an accurate LU decomposition of P using Gaussian elimination and backsolving six triangular systems of linear equations. This can be done with

$$\frac{n^3}{3} + 4n^2 + O(n)$$

operations. With Stiefel's algorithm, however, this price need not be paid at every exchange. The matrix P', derived from P by one exchange, differs from P only in its β^{th} column. If column interchanges are not permitted in computing LU decompositions, then the decomposition, $L'U'$, of P' is identical in certain portions to the decomposition, LU, of P, affording a saving of work. Furthermore, pivotal selection using row interchanges can be allowed. While an example of a matrix is given in [7] for which this strategy is poor, it is the strategy commonly used and is almost always stable in practice (e.g., see comments to this effect in [3] and [8]). In any event it is superior to the strategy of making no pivot selection.

The work done in carrying out k exchange steps, involving columns $\beta_1, \ldots, \beta_k$ of P, can be cut to

$$(k + 1)(\frac{n^3}{3} + 4n^2 + 7n) - \frac{n + 1}{3}(4k + 1) + \frac{7k + 10}{3}$$

$$+ \frac{2}{3} \sum_{j=1}^{k} \beta_j^3 - (n + 3) \sum_{j=1}^{k} \beta_j^2 + (2n + \frac{13}{3}) \sum_{j=1}^{k} \beta_j$$

operations.

For example, if $\beta_1 = \cdots = \beta_k = \frac{n}{2}$, this becomes

$$(k + 2)\frac{n^3}{6} + (\frac{17k}{4} + 4)n^2 + O(n),$$

roughly half the work that would be required if no advantage were taken of the similarities between P and P'.

6. Detailed Outline of an LU Implementation

1. Select $n+1$ indices $\{i_1, \ldots, i_{n+1}\} \subseteq \{1, \ldots, m\}$ so that the matrix

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

is nonsingular. If this cannot be done, terminate with an appropriate indication. The user may then check whether the system $Ax = d$ can be satisfied exactly.

2. Perform the Gaussian reduction of P^T into the product of a unit lower triangular matrix L and an upper triangular matrix U. All information about L and U can be stored in the space initially occupied by P^T plus one vector (for interchange information). In each column the element of largest magnitude on or below the diagonal is to be used as the pivot. If the LU decomposition of a matrix differing from P^T only in the β^{th} column is available, one can save computation by using the first $\beta-1$ columns and (as pointed out by W. Kahan of Toronto) the upper-right-hand $(\beta-1) \times (n-\beta)$ submatrix of this decomposition as the corresponding segments of the decomposition of P^T. If $\text{rank}(P^T) < n+1$, terminate.

3. Solve

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

This requires the forward-solution of

$$Lv = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}$$

followed by the back-solution of $U\lambda = v$. (Permutations due to the row interchanges of step (2) are ignored in the remainder of the outline). If $v_1, \ldots, v_n$ are available from a forward-solution involving an L whose first β-1 columns are identical with those of the matrix L being used here, only $v_\beta, \ldots, v_n$ need be computed. If any λ_i is zero, terminate.

4. Set

$$\epsilon = 1 / \sum_{i=1}^{n+1} |\lambda_i| \quad .$$

If ϵ is less than any value of ϵ previously computed for the current data, go to step (9).

5. Solve

$$Px = \epsilon \begin{bmatrix} \mathrm{sgn}(\lambda_1) \\ \vdots \\ \mathrm{sgn}(\lambda_{n+1}) \end{bmatrix} \quad .$$

x_{n+1} will turn out to be -1 .

6. Compute

$$r_j(x) = \sum_{k=1}^{n} a_{jk} x_k - d_j$$

for each

$$j \notin \{i_1, \ldots, i_{n+1}\} .$$

Let α be an index for which $|r_\alpha(x)|$ is maximal. If $|r_\alpha(x)| \leq \epsilon$, then $(x_1, \ldots, x_n)$ is a candidate as the Chebyshev solution of $Ax = d$; go to step (10).

7. Solve $P_\mu^T = A_\alpha^T$ (α^{th} column of A^T) .

8. Find $\beta \epsilon \{1, \ldots, n+1\}$ so that

$$\frac{\mu_\beta}{\lambda_\beta} \operatorname{sgn}[r_\alpha(x)]$$

is maximal. Replace the set of indices $\{i_1, \ldots, i_{n+1}\}$ by

$$\{i_1, \ldots, i_{\beta-1}, \alpha, i_{\beta+1}, \ldots, i_{n+1}\} .$$

Replace the β^{th} column of P^T by A_α^T . Go to step (2).

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

10. Iteratively refine the solution to the system

$$Px = \epsilon \begin{bmatrix} \operatorname{sgn}(\lambda_1) \\ \vdots \\ \operatorname{sgn}(\lambda_{n+1}) \end{bmatrix}$$

according to the scheme given on page 121 of [3]. (The convergence of this refinement process is established in [12]).

Check the residuals $r_j(x)$ for

$$j \notin \{i_1, \ldots, i_{n+1}\} .$$

If

$$|r_\alpha(x)| = \max_j |r_j(x)| \leq \epsilon,$$

then give $[x_1, \ldots, x_{n+1}]^T$ as the Chebyshev solution. If this residual check is not successful, but the refinement process has been carried out before and the last refined value of ϵ is greater than the current refined value of ϵ, return the last refined values of $x_1, \ldots, x_n$ as a doubtful solution. Otherwise return to step (7).

7. Remarks on the Outline

We have ignored scaling strategies in programming our implementation.

Step (10) serves to improve the final values of $\epsilon, x_1, \ldots, x_n$. It is usually performed only once. It is not uncommon to produce values for $\epsilon, x_1, \ldots, x_n$ which are correct substantially to full machine precision; i.e., compare runs A and D in the appendix. The decisions made in step (10), after the refinement, have been included as an attempt to supply the Chebyshév solution for the reference subsystem having the largest reference deviation in those infrequent cases where the test

$$\max_{j\notin\{i_1,\dots,\ i_{n+1}\}} |r_j(x)| \leq \epsilon$$

consistently fails to be satisfied.

Note that the LU decomposition of P^T is used to solve the system of equations $Px = \epsilon\ \mathrm{sgn}(\lambda)$ (step 5). In [3] it is shown that the computed solution to $Ax = b$ via LU decomposition is the exact solution to $(A + K)x = b$, where a bound on $\|K\|_\infty$ can be placed. It is easily shown that the computed solution to $A^Ty = d$ via the LU decomposition of A is the exact solution to $(A + H)y = d$, where the same bound pertains to $\|H\|_\infty$.

8. Algol 60 Description

```
procedure Chebyshev (A,d,h,m,n,refset,epz,insufficientrank,zerolambda);
value m,n; integer m,n; real array A,d,h;
integer array refset; real epz; label insufficientrank, zerolambda;
begin
real procedure ipr (ii,ℓℓ,uu,aa,bb,cc);
value ℓℓ,uu,cc; real aa,bb,cc; integer ii,ℓℓ,uu;
begin comment single-precision inner-product routine;
real sum;
   sum := cc;
   for ii := ℓℓ step 1 until uu do sum := sum + aa×bb;
   ipr := sum;
```

Erratum: Insert in §8 before the Algol 60 procedure of Computer Science Report No. 67, Stanford University.

The parameters to procedure Chebyshev are:

identifier	type	comments
m	integer	Number of equations.
n	integer	Number of unknowns.
A	real array	Matrix of coefficients. Array bounds - [0:m-1, 0:n-1].
d	real array	Right-hand-side vector. Array bounds - [0:m-1].
h	real array	Solution vector. Array bounds - [0:n-1].
refset	integer array	Final reference equation numbers. Array bounds - [0:n].
epz	real	Final reference deviation.
zerolambda	label	Exit for condition 1 failure.
insufficientrank	label	Exit for condition 2 failure, or in case rank(A) < n .

The parameters m, n, A, and d are not changed by Chebyshev. We direct the user's attention to the identifier eta appearing in the procedure and to the comment explaining its value and purpose.

```
end ipr;
real procedure ip2 (ii,ℓℓ,uu,aa,bb,cc);
comment ip2 is a version of ipr which accumulates the products aa×bb in
        a double-precision sum, whose final value, rounded to single-
        precision, is taken as the value of ip2.;
procedure trisolv (fis,fid,fie,sis,sie,fi,si,soℓ,rhs,mat,piv,vip);
value fis,fid,fie; integer fis,fid,fie,sis,sie,fi,si;
real soℓ,rhs,mat,piv; real procedure vip;
begin real tl,t2;
comment trisolv solves a triangular system of linear equations.  The
        off-diagonal part of the system's matrix is given by mat, the
        diagonal part by piv, and the right hand side of the system by
        rhs.  The solution is developed in soℓ.  By appropriately
        setting the first five parameters, either an upper or a lower
        triangular system can be treated.  Column-by-column Gauss
        decomposition of a matrix can be compactly expressed using
        trisolv.  vip is a vector inner-product routine.;
   for fi := fis step fid until fie do
   begin tl := -vip (si,sis,sie,soℓ,mat,-rhs); t2 := piv;
         si := fi; soℓ := if t2 = 1 then tl else tl/t2;
   end;
end trisolv;
Boolean finished; switch decompbranch := return,itr;
switch failures := insufficientrank,zerolambda;
integer ml,nl,npl,i,j,k,ℓ,b,aℓ,al,ℓst,ℓ0,ℓ1,ℓ01,cnt;
real lasteps,preveps,ref,s,t,cps,eta,cnorm,snorm;
real array P[o:n,o:n],lam,rv,sv,x,w,xr[o:n];
integer array r[o:n],ix[o:m-1];
comment The subsystem of  n+1  equations currently being investigated
        is listed in ix[o],..., ix[n] .  The other equations are listed
        in the remainder of  ix .  r contains row indices.  Row inter-
        changes during the Gauss decomposition of  P  are carried out
        by permuting the elements of  r ;
procedure resid (vip); real procedure vip;
begin
comment resid computes those components of the residual vector  Ax-d
        associated with the equations not in the reference subsystem.
        The sign, magnitude, and associated equation number of the
        largest component are saved.  vip is a vector inner-product
        routine.;
   ref := -1;
   for j := npl step 1 until ml do
   begin
      i := ix[j];
      t := vip (k,0,nl,x[k],A[i,k],-d[i]);
      if abs (t) > ref then begin ref := abs (t);
                                  aℓ := j; s := sign (t);
                            end;
   end;
end resid;
   ml := m-1; nl := n-1; npl := n+1;
   lasteps := 0; preveps := -1;
   for i := 0 step 1 until n do r[i] := ix[i] := i;
   for i := npl step 1 until ml do ix[i] := i;
comment The initial reference subsystem is chosen by making a copy of
        the transpose of  A  bordered with  d  and carrying out a
        Gaussian reduction upon it with row and column interchanges
        used to select the largest possible pivot at each stage.;
   begin
   real array TAB[o:n,o:ml];
      for j := 0 step 1 until ml do
      begin
         TAB[n,j] := d[j];
         for i := 0 step 1 until nl do TAB[i,j] := A[j,i];
      end;
      for i := 0 step 1 until n do
      begin
         t := 0;
         for j := i step 1 until n do
         begin
            k := r[j];
            for ℓ := i step 1 until ml do
            begin
               ref := TAB[k,ix[ℓ]];
               if abs (ref) > t then
               begin s := ref; t := abs (ref); aℓ := j; b := ℓ; end;
            end;
         end;
         if t = 0 then begin j := 1; go to singular; end;
         k := r[aℓ]; r[aℓ] := r[i]; ℓst := r[i] := k;
         k := ix[b]; ix[b] := ix[i]; al := ix[i] := k;
         for j := i+1 step 1 until ml do
         begin
            ℓ := ix[j];
            ref := TAB[ℓst,ℓ]/s;
            for k := i+1 step 1 until n do
            begin
               aℓ := r[k];
               TAB[aℓ,ℓ] := TAB[aℓ,ℓ] - TAB[aℓ,al] x ref;
            end;
         end;
      end;
   end;
   b := 0; al := 1;
comment The following segment of the program performs a column-by-column
        Gaussian reduction of the matrix associated with the reference
        equations, forming an upper and a lower triangular matrix into
        the array  P .  (Each diagonal element of the lower triangular
        matrix is one.)  Interchanges of rows take place so that the
        largest pivot in each column is employed.  It is assumed that
        b-1  columns have already been decomposed.  If the matrix is
        not of full rank, the exit insufficientrank is taken, and it
        is left up to the user to determine if the given overdetermined
        system can be solved exactly.;
```

```
body:
   ℓ0 := b; ℓ1 := b+1; ℓ01 := b-1;
   for i := b step 1 until n do
   begin
      ℓ := ix[ℓ0];
      trisolv (if i=b then 0 else b,1,ℓ01,0,j-1,j,k,P[ℓ0,r[k]],
         if r[j]=n then d[ℓ] else A[ℓ,r[j]],P[k,r[j]],1,ipr);
      trisolv (ℓ0,1,n,0,ℓ01,j,k,P[ℓ0,r[k]],
         if r[j]=n then d[ℓ] else A[ℓ,r[j]],P[k,r[j]],1,ipr);
      ref := 0;
      for j := ℓ0 step 1 until n do
      begin
         t := P[ℓ0,r[j]];
         if ref < abs (t) then begin ref := abs (t); s := t; k := j; end;
      end;
      if ref = 0 then begin j := 1; go to singular; end;
      if ℓ0 = n then go to decompbranch[al];
      j := r[k]; r[k] := r[ℓ0]; r[ℓ0] := j;
      for j := ℓ1 step 1 until n do P[ℓ0,r[j]] := P[ℓ0,r[j]]/s;
      ℓ01 := ℓ0; ℓ0 := ℓ1; ℓ1 := ℓ1+1;
   end;
singular:
   for i := 0 step 1 until n do refset[i] := ix[i];
   go to failures[j];
return:
comment Solve for the lambdas.;
```

```
    trisolv (b,1,n,0,j-1,j,k,sv[k], if r[j]=n then -1 else 0;
                 P[k,r[j]],1,ipr);
    trisolv (n,-1,0,j+1,n,j,k,lam[k],sv[j],P[k,r[j]],P[j,r[j]],ipr);
comment Compute epsilon for the reference subsystem of equations.;
    t := 0;
    for i := 0 step 1 until n do t := t+abs(lam[i]);
    eps := 1/t;
comment Each new value of eps must be greater than the previous one.
        If this is not so, the solution may have been "overshot".;
    if eps < lasteps then go to ed;
    lasteps := eps;
comment Solve for the vector  x,  the Chebyshev solution of the reference
        subsystem of equations.;
    for i := 0 step 1 until n do xr[i] := sign(lam[i]) × eps;
    trisolv (0,1,n,0,i-1,i,j,w[j],xr[i],P[i,r[j]],P[i,r[i]],ipr);
    trisolv (n,-1,0,i+1,n,i,j,x[r[j]],w[i],P[i,r[j]],1,ipr);
comment x[n]  should be -1 .  It can be used to purify eps and the other
        components of  x .;
    ref := -x[n];
    for i := 0 step 1 until n1 do x[i] := x[i]/ref;
    eps := eps/ref;
comment For each index  ix[n+1],..., ix[m-1]  compute the residual
        A[ix[j],o] × x[o] + ... + A[ix[j],n-1] × x[n-1] - d[ix[j]] .
        If the largest of these in magnitude is not greater than eps,
        go to itr to refine the vector  x,  for it may be the Chebyshev
        solution of the full system.;
```

```
   resid (ipr);
   if ref ≤ eps then go to itr;
ovr:
   k := ix[aℓ];
comment The following linear-system solution is computed in order to
        determine which equation is to be dropped from the reference
        set of equations.;
   trisolv (0,1,n,0,i-1,i,j,w[j], if r[i]= n then d[k]
            else A[k,r[i]],P[j,r[i]],1,ipr);
   trisolv (n,-1,0,i+1,n,i,j,w[j],w[i],P[j,r[i]],P[i,r[i]],ipr);
comment s is the sign of the residual with greatest magnitude.  Find
        the largest of the ratios  w[k]/lam[k] × s .  If any component
        of lam is zero, the exit zerolambda is taken.;
   ref := lam[n]; b := n;
   if ref = 0 then begin j := 2; go to singular; end;
   ref := w[n]/ref × s
   for j := 0 step 1 until n1 do
   begin
      t := lam[j];
      if t=0 then begin j := 2; go to singular; end;
      t := w[j]/t × s;
      if t > ref then begin b := j; ref := t; end;
   end;
comment Form a new reference subsystem by exchanging the  ix[aℓ]-th
        and  ix[b]-th  equations.;
   ix[aℓ] := ix[b];  ix[b] := k;  al := 1;  go to body;
ed:
```

```
comment Restore the previous reference subsystem.;
   eps := lasteps; al := 2;
   j := ix[aℓ]; ix[aℓ] := ix[b]; ix[b] := j; go to body;
itr:
   lasteps := 0; cnt := 0;
comment Iteratively refine the vector x;
ilp:
   cnt := cnt + 1; if cnt > 10 then go to insufficientrank;
   cnorm := snorm := 0;
   for t := 0 step 1 until n do
   begin
      k := ix[i];
      t := abs (x[i]);
      if snorm < t then snorm := t;
      rv[i] := -ip2 (j,0,n,x[j], if j=n then d[k] else A[k,j], -xr[i]);
   end;
   trisolv (0,1,n,0,i-1,i,j,rv[j],rv[i],P[i,r[j]],P[i,r[i]],ip2);
   trisolv (n,-1,0,i+1,n,i,j,w[r[j]],rv[i],P[i,r[j]],1,ip2);
   for i := 0 step 1 until n do
   begin
      s := w[i];
      x[i] := x[i] + s;
      s := abs (s);
      if cnorm < s then cnorm := s;
   end;
   if cnorm/snorm > eta then go to ilp;
```

```
comment eta is to be preset with a small positive multiple of the largest
        positive single-precision machine number  ω  having the property
        that  1+ω = 1-ω = 1  in single-precision arithmetic.  The small
        multiple will depend upon the peculiarities of the machine's
        rounding process and will have to be empirically determined.;
   ref := -x[n]
   for i := 0 step 1 until n1 do x[i] := x[i]/ref;
   eps := eps/ref;
comment Determine whether a Chebyshev solution has been found.  If any
        residual is greater in magnitude than eps while eps is smaller
        than a value produced from an earlier refinement, give up, print
        a warning, and return the best  x  computed thus far.;
   resid (ip2);
   if ref ≤ eps then finished := true
   else if eps > preveps then finished := false
   else begin comment Print out "DOUBTFUL SOLUTION";
      go to skip; end;
   preveps := eps; refset[n] := ix[n];
   for i := 0 step 1 until n1 do
   begin
      refset[i] := ix[i];
      h[i] := x[i];
   end;
   if ¬ finished then go to ovr;
skip:
   epz := preveps;
end  Chebyshev;
```

9. Sample Runs

The output reproduced in the appendix was produced by four programs implementing the exchange method. At each exchange step the reference set, value of ϵ, values for the x_i, and the non-reference residuals were listed followed by the equations to be switched in the next exchange. Upon termination, a count of exchanges and solution refinements (where applicable) was printed along with the computation time required (print time excluded). The computed Chebyshev solution for the full system was then printed followed by the final reference set and a list of all residuals.

A common data system, $Ax = d$, was given to the four programs. The matrix A consisted of the 17×9 Hilbert matrix segment

$$a_{i,j} = \frac{1}{i+j+1} \quad (i=0,\dots, 16;\ j=0,\dots, 8) .$$

The right-hand vector d had components

$$d_i = i \quad (i=0,\dots, 16) .$$

Output A was produced by a version of the program given in section 8 using double-precision arithmetic.

Output B was produced by a program using the techniques outlined in section 4. This program, however, based its computation on the matrix

$$B = \left[\begin{array}{|c|c|c|} \hline A^T_{i_1} & \cdots & A^T_{i_{n+1}} \\ \hline -d_{i_1} & \cdots & -d_{i_{n+1}} \\ \hline \end{array}\right]^{-1}$$

rather than on the matrix C . This permits the initial

$$\frac{n^3}{3} + O(n^2)$$

operations for the calculation of the λ_i to be saved, for the last column of B satisfies

$$\begin{cases} \sum_{k=1}^{n+1} b_{k,n+1} A_{i_k} = 0 \\ - \sum_{k=1}^{n+1} b_{k,n+1} d_{i_k} = 1 \end{cases} .$$

Now, however,

$$\varepsilon = 1/ \sum_{k=1}^{n+1} |b_{k,n+1}|$$

must be computed separately at each exchange. Note that, on the sample data, this program has failed to recognize the terminal reference set, giving the wrong answer.

The suggestion has been made that the exchange method be implemented using Jordan elimination techniques, but that a section of code be provided to clean up the solution once it has been attained. Output C was produced by such a program. Clean-ups were carried out in double-precision. Since this program, just as program B, failed to recognize the final reference set at the first encounter, the clean-up section was called upon twice for the given data set - once to put the program back on the right track, and once for the final solution refinement. By good fortune the final reference set was recognized the second time around.

Output D was produced by a B5500 Burroughs Extended Algol version of the procedure given in section 8.

10. Double-Exchange Algorithm

Instead of introducing one vector into the reference set, we consider the problem of introducing two vectors simultaneously. (What follows can easily be generalized to the problem of introducing several vectors simultaneously.)

Without loss of generality, we assume that $A_1, \ldots, A_{n+1}$ form a reference set. Let $\lambda_1, \ldots, \lambda_{n+1}$ be such that

$$\sum_{k=1}^{n+1} \lambda_k A_k = 0$$

under the normalization

$$\sum_{k=1}^{n+1} \lambda_k d_k = -1 \quad .$$

Then

$$\epsilon = 1/ \sum_{k=1}^{n+1} |\lambda_k| > 0,$$

and if x is the Chebyshev solution for this reference subsystem, $\operatorname{sgn}(\lambda_k) = \operatorname{sgn}(r_k(x))$ for $k=1,\ldots, n+1$.

For ease of notation we write

$$B_i = \operatorname{sgn}(r_i(x))A_i \quad \text{for all} \quad i$$

$$\tau_k = \operatorname{sgn}(r_k(x))\lambda_k = |\lambda_k| \quad \text{for} \quad k=1,\dots, n+1 \quad .$$

Thus

$$\sum_{k=1}^{n+1} \tau_k B_k = 0 \quad \text{and} \quad \epsilon = 1/\sum_{k=1}^{n+1} \tau_k \quad .$$

We assume that

$$|r_{\alpha_1}(x)| \geq |r_{\alpha_2}(x)| > \epsilon$$

for some $\alpha_1,\alpha_2 > n+1$. Since $B_1,\dots, B_{n+1}$ have rank n, there exist

$$\mu_1^{(1)},\dots, \mu_{n+1}^{(1)} \quad \text{and} \quad \mu_1^{(2)},\dots, \mu_{n+1}^{(2)}$$

so that

$$B_{\alpha_j} = \sum_{k=1}^{n+1} \mu_k^{(j)} B_k \quad \text{for} \quad j=1, 2 \quad .$$

The $\mu_k^{(j)}$ will be unique if we also demand that

$$d_{\alpha_j} = \sum_{k=1}^{n+1} d_k \mu_k^{(j)} \quad \text{for} \quad j=1, 2 \quad .$$

We wish to find rows A_{β_1}, A_{β_2} $(\beta_1, \beta_2 \in \{1, \ldots, n+1\})$ to exchange with $A_{\alpha_1}, A_{\alpha_2}$ in order to form a reference set with a greater reference deviation ϵ'. Associated with this will be a reference subsystem Chebyshev solution x'. Demanding suitable agreement between the signs of $r_k(x)$ and $r_k(x')$, we may use the characterization theorem of section 2 to determine β_1 and β_2. Viz., we ask for numbers γ_1 and γ_2 such that

$$\gamma_1 B_{\alpha_1} + \gamma_2 B_{\alpha_2} + \sum_{i=1}^{n+1} (\tau_i - \gamma_1 \mu_i^{(1)} - \gamma_2 \mu_i^{(2)}) B_i = 0$$

with

$$\tau'_{\alpha_j} = \gamma_j \geq 0 \quad \text{for} \quad j=1, 2$$

$$\tau'_i = (\tau_i - \gamma_1 \mu_i^{(1)} - \gamma_2 \mu_i^{(2)}) \geq 0$$

$$\text{for} \quad i=1, \ldots, n+1$$

and for two indices β_1, β_2

$$\tau'_{\beta_1} = \tau'_{\beta_2} = 0 \quad .$$

The normalizations of the $\mu_k^{(j)}$ have been chosen so that

$$\epsilon' = 1 \Big/ \left[\left(\sum_{i=1}^{n+1} \tau'_i \right) + \tau'_{\alpha_1} + \tau'_{\alpha_2} \right] \quad .$$

We wish to choose γ_1, γ_2 under the above constraints so as to maximize ϵ'. This is equivalent to determining the minimum of

$$\left(\sum_{i=1}^{n+1} \tau_i'\right) + \tau_{\alpha_1}' + \tau_{\alpha_2}'$$

$$= \left(\sum_{i=1}^{n+1} \tau_i\right) + \gamma_1\left(1 - \sum_{k=1}^{n+1} \mu_k^{(1)}\right) + \gamma_2\left(1 - \sum_{j=1}^{n+1} \mu_j^{(2)}\right)$$

Since

$$\sum_{k=1}^{n+1} \tau_k$$

is fixed, and (as can easily be shown)

$$e_j \equiv \sum_{k=1}^{n+1} \mu_k^{(j)} - 1 > 0 \quad (j=1,\ 2),$$

we wish to determine $\gamma_1, \gamma_2 \geq 0$ so that

$$\gamma_1 e_1 + \gamma_2 e_2$$

is maximized subject to

$$\gamma_1 \mu_k^{(1)} + \gamma_2 \mu_k^{(2)} \leq \tau_k \quad \text{for } k=1,\ldots, n+1 .$$

This is a standard linear programming problem. Note that the single-exchange algorithm can be expressed as the above problem with the additional constraint

$$\gamma_2 = 0 \ .$$

Thus the ε' of the double-exchange can be no less than the ε' given by the single-exchange of section 3. Note further that conditions 1 and 2 of section 3 do not appear in the development of the double-exchange.

Computation can be simplified by considering the dual to the above linear programming problem. We introduce the surplus variables z_{n+2}, z_{n+3} and minimize

$$\sum_{i=1}^{n+1} \tau \, z_i$$

subject to

$$z_i \geq 0 \quad \text{for all} \quad i,$$

$$\sum_{k=1}^{n+1} \mu_k^{(1)} z_r - z_{n+2} = e_1,$$

and

$$\sum_{k=1}^{n+1} \mu_k^{(2)} z_k - z_{n+3} = e_2 \quad .$$

If either surplus variable is nonzero in the solution, then B_{α_1} and B_{α_2} cannot simultaneously be introduced into the reference set. The correct single-exchange, however, is then readily obtainable from the dual problem solution.

In section 16 are presented some timing results from a program implementing this algorithm. Comparing these results with those from the single-exchange implementations of sections 13-15, we see that the extra effort involved is not paid for by a net reduction in time. Also we have observed that in practice rather less than half of the exchange steps carried out permit the simultaneous switching of two reference equations.

11. Computational Comparisons of Variations for the Exchange

In the procedure given in section 8, the non-reference equation chosen to enter the reference system at each exchange was the α^{th}, whose residual satisfied

$$\text{(a)} \quad |r_\alpha(x)| = \max_{i \notin \left\{\substack{\text{reference set} \\ \text{indices}}\right\}} |r_i(x)| \ .$$

According to the theory, however, the exchange method will converge so long as the reference deviation after each exchange exceeds the reference deviation before. And for this to be true, it is sufficient only that α satisfy $|r_\alpha(x)| > |\varepsilon|$ (conditions 1 and 2 given section 3 being assumed always to hold).

Alternate versions of the procedure presented in section 8 were prepared for Stanford's B5500 wherein the few statements determining α according to (a) were changed for statements implementing other selection rules. The unaltered procedure and the alternates, together with an implementation of the double-exchange method described in section 10, were run on random systems of equations of several sizes. Averages of times required and number of exchanges made are given. Note that the procedure of section 8 gave the most favorable times.

12. The Data

Data for the comparison runs was generated by a procedure written in Burroughs Extended Algol. The procedure produced a matrix

$$A = [a_{ij}]_{\substack{i=o,\dots,\ m-1 \\ j=o,\dots,\ n-1}}$$

and a vector

$$d = [d_i] \quad i=o,\dots,\ m-1$$

each of whose elements had the form $\xi \times \eta$, where ξ was a pseudo-random variable distributed approximately uniformly in the interval $[o, +1]$, as computed by the mixed congruential method

$$\begin{cases} \xi_o = o \\ \xi_{n+1} = (2^{11} - 3)\xi_n + 211527139 \mod 2^{27} \\ \qquad \text{for } n \geq 1 \ , \end{cases}$$

and η was chosen pseudo-randomly from among the numbers

$$+1, +8^{-1}, +8^{-2}, +8^{-3}, -1, -8^{-1}, -8^{-2}, -8^{-3} .$$

Every decision rule was applied to ten system, each of m equations in n unknowns, where

$$(m,n) \epsilon \{(10,4),(20,4),(30,4),(40,4),(20,9),(30,9),(40,9),(30,19)\} .$$

13. Selection of the Equation with Largest Residual Magnitude

The procedure given in section 8 produced the following statistics ($\mu \equiv$ mean; $\sigma \equiv$ standard deviation):

Time Required (Seconds)

m \ n	4	9	19
10	μ=0.677 σ=0.110		
20	μ=1.079 σ=0.142	μ=4.043 σ=0.850	
30	μ=1.246 σ=0.236	μ=5.947 σ=1.170	μ=28.620 σ=6.802
40	μ=1.558 σ=0.266	μ=7.265 σ=1.740	

Number of Exchanges

m \ n	4	9	19
10	$\mu=3.40$		
	$\sigma=1.56$		
20	$\mu=5.90$	$\mu=9.10$	
	$\sigma=1.81$	$\sigma=3.05$	
30	$\mu=5.90$	$\mu=13.40$	$\mu=16.80$
	$\sigma=2.21$	$\sigma=3.64$	$\sigma=5.21$
40	$\mu=6.70$	$\mu=14.60$	
	$\sigma=2.10$	$\sigma=5.16$	

14. Selection of the First Suitable Equation Found

The first variant program examined each non-reference equation in turn until one was found whose residual magnitude exceeded the reference deviation. That equation was selected for introduction into the reference system. Statistics for this variant follow.

Time Required (seconds)

m \ n	4	9	19
10	$\mu=0.813$ $\sigma=0.261$		
20	$\mu=1.420$ $\sigma=0.638$	$\mu=7.361$ $\sigma=2.788$	
30	$\mu=2.122$ $\sigma=0.974$	$\mu=11.303$ $\sigma=2.699$	$\mu=63.785$ $\sigma=30.424$
40	$\mu=2.181$ $\sigma=0.606$	$\mu=14.252$ $\sigma=5.458$	

Number of Exchanges

m \ n	4	9	19
10	$\mu=5.60$ $\sigma=3.98$		
20	$\mu=13.10$ $\sigma=9.69$	$\mu=22.60$ $\sigma=10.83$	
30	$\mu=20.50$ $\sigma=13.32$	$\mu=36.40$ $\sigma=10.43$	$\mu=48.60$ $\sigma=26.04$
40	$\mu=21.80$ $\sigma=8.83$	$\mu=47.60$ $\sigma=21.30$	

15. Selection So As to Give the Greatest Reference Deviation Increase

Given any non-reference row A_α for which $|r_\alpha(x)| > |\varepsilon|$, solve

$$P_\mu^T = A_\alpha^T \quad .$$

Then, if β is such that

$$\frac{\operatorname{sgn}(\varepsilon)\ \operatorname{sgn}(r_\alpha(x))\mu_\beta}{\lambda_\beta}$$

is maximal, A_α must replace A_{i_β} in the reference set. The new λ's can be computed as follows:

$$\begin{cases} \lambda_\beta' = \lambda_\beta/\mu_\beta \\ \lambda_i' = \lambda_i - \dfrac{\mu_i}{\mu_\beta}\lambda_\beta \quad (i \neq \beta) \end{cases}$$

Then

$$|\varepsilon'| = \frac{|\lambda_\beta'|}{K}\,|r_\alpha(x)| + (1 - \frac{|\lambda_\beta'|}{K})\,|\varepsilon| \ ,$$

where

$$K = \sum_{i=1}^{n+1} |\lambda_i'| \quad .$$

Using these results, a variant of the procedure given in section 8 was prepared in which the non-reference equation selected to enter the reference system at each exchange was that one which would give the greatest value to $|\varepsilon'|$.

Time Required (seconds)

m \ n	4	9	19
10	μ=0.821 σ=0.187		
20	μ=1.315 σ=0.327	μ=5.900 σ=1.886	
30	μ=1.528 σ=0.313	μ=9.798 σ=2.423	μ=42.481 σ=7.921
40	μ=2.134 σ=0.465	μ=14.685 σ=3.825	

Number of Exchanges

m \ n	4	9	19

10	μ=3.50
	σ=1.36		

20	μ=4.60
	σ=1.74
		μ=8.10
		σ=2.84	

30	μ=3.90
	σ=1.04
		μ=10.40
		σ=2.20
			μ=16.20
			σ=3.03

40	μ=5.10
	σ=1.70
		μ=13.40
		σ=3.23	

16. Double-Exchange Algorithm

Time Required (seconds)

m \ n	4	9	19

10	μ=0.900
	σ=0.147		

20	μ=1.258
	σ=0.215
		μ=4.557
		σ=0.836	

30	μ=1.442
	σ=0.271
		μ=6.487
		σ=0.951
			μ=36.650
			σ=8.179

40	μ=1.912
	σ=0.677
		μ=9.413
		σ=1.507	

Number of Exchange Cycles*

m \ n	4	9	19
10	$\mu=2.40$ $\sigma=1.02$		
20	$\mu=3.90$ $\sigma=1.38$	$\mu=5.60$ $\sigma=1.80$	
30	$\mu=3.60$ $\sigma=1.36$	$\mu=8.10$ $\sigma=1.70$	$\mu=14.7$ $\sigma=4.67$
40	$\mu=5.10$ $\sigma=3.24$	$\mu=12.4$ $\sigma=2.36$	

*(An exchange cycle consisted of the simultaneous switching of two equations where possible. Otherwise it consisted of a standard single-exchange.)

References

[1] Stiefel, Edward L.: Über diskrete und lineare Tschebyscheff-Approximation. Numerische Mathematik 1, 1-28 (1959).

[2] Stiefel, Eduard L.: Note on Jordan elimination, linear programming and Tchebycheff approximation. Numerische Mathematik 2, 1-17 (1960).

[3] Wilkinson, J. H.: Rounding Errors in Algebraic Processes. Prentice-Hall (Series in Automatic Computation), 1963.

[4] Davis, Philip J.: Interpolation and Approximation. Blaisdell, 1963.

[5] Fan, Ky: On Systems of Linear Inequalities. pp. 99-156 in Linear Inequalities and Related Systems, ed. by H. W. Kuhn and A. W. Tucker, Annals of Mathematics Studies No. 38, Princeton, 1956.

[6] Moursund, David: Chebyshev Solution of n+1 Linear Equations in n Unknowns. ACM Journal, v. 12, no. 3 (July, 1965), pp. 383-387.

[7] Wilkinson, J. H.: Error Analysis of Direct Methods of Matrix Inversion. ACM Journal, v. 8, pp. 281-330.

[8] Wilkinson, J. H.: Rounding Errors in Algebraic Processes,Proceedings of the International Conference on Information Processing, UNESCO 1959.

[9] Cheney, E. W.: Introduction to Approximation Theory. McGraw-Hill (International Series in Pure and Applied Mathematics), 1966.

[10] Moler, C. B.: Accurate Solution of Linear Algebraic Systems, A survey presented at the Spring Joint Computer Conference, April, 1967.

[11] Forsythe, G. E.: Today's Computational Methods of Linear Algebra, SIAM Review, to appear.

[12] Moler, C. B.: Iterative Refinement in Floating Point, J. Assoc. Comput. Mach., v. 14 (1967), to appear.

APPENDIX

A

```
***** HILBERT DATA *****

 17 EQUATIONS IN  9 UNKNOWNS

EXCHANGE ALGORITHM IN DOUBLE-PRECISION


REFERENCE SET:
    0    2   11    1    5   16    3    8    4    9
  1.65566,11074,00287,19989,63@ -3 = EPS
  3.97047,60096,43108,06759,27@  3 = X[  0]
 -2.70355,98109,22439,41451,33@  5 = X[  1]
  4.55974,74511,68592,81444,73@  6 = X[  2]
 -3.26294,80229,02832,31393,35@  7 = X[  3]
  1.20427,47950,72163,13981,29@  8 = X[  4]
 -2.48030,00464,27790,08801,45@  8 = X[  5]
  2.87782,84818,63886,24654,61@  8 = X[  6]
 -1.75788,53710,63890,56612,90@  8 = X[  7]
  4.39490,84805,13472,43622,04@  7 = X[  8]
 -2.57761,12267,27838,37828,09@ -3 = RESIDUAL[ 10]
  1.07571,35978,74505,82246,03@ -3 = RESIDUAL[  6]
  1.00245,84291,63795,29485,83@ -2 = RESIDUAL[ 12]
  1.95676,92968,81866,18250,25@ -2 = RESIDUAL[ 13]
  2.48292,91781,18352,39794,14@ -2 = RESIDUAL[ 14]
  1.98447,14020,05852,36957,66@ -2 = RESIDUAL[ 15]
  3.46169,10322,94211,82277,29@ -3 = RESIDUAL[  7]
EXCHANGE EQUATION  11 WITH EQUATION  14

REFERENCE SET:
    0    2   14    1    5   16    3    8    4    9
  3.29205,02104,33056,37236,26@ -3 = EPS
  5.40554,73215,77523,85646,17@  3 = X[  0]
 -3.59146,43813,84385,18429,92@  5 = X[  1]
  5.93749,90812,11901,92471,06@  6 = X[  2]
 -4.17859,65144,44931,34071,22@  7 = X[  3]
  1.52049,75097,12577,52825,27@  8 = X[  4]
 -3.09350,90379,71150,66375,69@  8 = X[  5]
  3.55123,22422,42775,98985,69@  8 = X[  6]
 -2.14895,91603,99356,68537,18@  8 = X[  7]
  5.32811,40323,12399,58580,08@  7 = X[  8]
 -1.04956,72051,91559,43804,88@ -2 = RESIDUAL[ 10]
 -2.27328,65222,76318,77735,80@ -4 = RESIDUAL[  6]
 -1.08864,60005,98963,99208,98@ -2 = RESIDUAL[ 12]
 -3.91191,80148,65456,42774,21@ -3 = RESIDUAL[ 13]
 -1.34090,31468,42742,36611,82@ -2 = RESIDUAL[ 11]
  5.51335,12173,87940,65038,29@ -3 = RESIDUAL[ 15]
  4.64657,18263,68794,95518,65@ -3 = RESIDUAL[  7]
EXCHANGE EQUATION   9 WITH EQUATION  11

REFERENCE SET:
    0    2   14    1    5   16    3    8    4   11
  5.30006,47585,98979,14408,70@ -3 = EPS
  6.27879,92051,09165,06026,78@  3 = X[  0]
```

```
-4.09612,36199,45232,02513,46@  5 = X[  1]
 6.67733,07627,41189,48351,34@  6 = X[  2]
-4.64706,76438,19307,65657,34@  7 = X[  3]
 1.67565,40534,98819,77968,69@  8 = X[  4]
-3.38355,28056,64260,96258,72@  8 = X[  5]
 3.85958,25436,98984,77619,29@  8 = X[  6]
-2.32292,40280,66317,48340,80@  8 = X[  7]
 5.73258,79224,06205,97422,59@  7 = X[  8]
-2.61547,83860,89942,20876,97@ -3 = RESIDUAL[ 10]
-4.40861,93560,47416,28225,15@ -3 = RESIDUAL[  6]
-4.10282,78390,15665,83395,49@ -3 = RESIDUAL[ 12]
 6.41483,09155,73710,43478,49@ -4 = RESIDUAL[ 13]
 2.48599,96575,45978,62499,50@ -3 = RESIDUAL[  9]
 5.20435,98784,03175,26575,97@ -3 = RESIDUAL[ 15]
 2.53687,03218,72219,94661,28@ -3 = RESIDUAL[  7]

TERMINATION

NUMBER OF EXCHANGES MADE WAS   2
TIME IN SECONDS =     5.48

SOLUTION VECTOR:
 6.27879,92051,09165,06026,78@  3 = X[  0]
-4.09612,36199,45232,02513,46@  5 = X[  1]
 6.67733,07627,41189,48351,34@  6 = X[  2]
-4.64706,76438,19307,65657,34@  7 = X[  3]
 1.67565,40534,98819,77968,69@  8 = X[  4]
-3.38355,28056,64260,96258,72@  8 = X[  5]
 3.85958,25436,98984,77619,29@  8 = X[  6]
-2.32292,40280,66317,48340,80@  8 = X[  7]
 5.73258,79224,06205,97422,59@  7 = X[  8]
```

```
REFERENCE SET:
   0    2  14    1    5  16    3    8    4  11

RESIDUALS:
 5.30006,47585,99232,20639,30@ -3 = RESIDUAL[  0]
-5.30006,47585,98899,13948,56@ -3 = RESIDUAL[  1]
 5.30006,47585,99232,20639,30@ -3 = RESIDUAL[  2]
-5.30006,47585,98677,09488,07@ -3 = RESIDUAL[  3]
 5.30006,47585,99232,20639,30@ -3 = RESIDUAL[  4]
-5.30006,47585,99176,69524,17@ -3 = RESIDUAL[  5]
-4.40861,93560,47416,28225,15@ -3 = RESIDUAL[  6]
 2.53687,03218,72219,94661,28@ -3 = RESIDUAL[  7]
 5.30006,47585,98954,65063,68@ -3 = RESIDUAL[  8]
 2.48599,96575,45978,62499,50@ -3 = RESIDUAL[  9]
-2.61547,83860,89942,20876,97@ -3 = RESIDUAL[ 10]
-5.30006,47585,99232,20639,30@ -3 = RESIDUAL[ 11]
-4.10282,78390,15665,83395,49@ -3 = RESIDUAL[ 12]
 6.41483,09155,73710,43478,49@ -4 = RESIDUAL[ 13]
 5.30006,47585,98899,13948,56@ -3 = RESIDUAL[ 14]
 5.20435,98784,03175,26575,97@ -3 = RESIDUAL[ 15]
-5.30006,47585,99287,71754,42@ -3 = RESIDUAL[ 16]
```

B

```
***** HILBERT DATA *****

 17 EQUATIONS IN   9 UNKNOWNS

TABLEAU-JORDAN ALGORITHM

COMPUTATION:


REFERENCE SET:
    0     2    11     1     5    16     3     8     4     9
EPS =  1,65552306833@-03
X[  0] =   3,96754579750@+03
X[  1] = ¯2,70186705478@+05
X[  2] =   4,55729602267@+06
X[  3] = ¯3.26142253588@+07
X[  4] =   1,20377941299@+08
X[  5] = ¯2,47939286842@+08
X[  6] =   2,87688384861@+08
X[  7] = ¯1,75736324827@+08
X[  8] =   4,39371863702@+07
RESIDUAL[ 10] = ¯2,59399414063@-03
RESIDUAL[  6] =  1.12915039063@-03
RESIDUAL[ 12] =  1.00135803223@-02
RESIDUAL[ 13] =  1,95732116699@-02
RESIDUAL[ 14] =  2.48756408691@-02
RESIDUAL[ 15] =  1.98707580566@-02
RESIDUAL[  7] =  3.50952148438@-03
EXCHANGE EQUATION  14 WITH EQUATION  11


REFERENCE SET:
    0     2    14     1     5    16     3     8     4     9
EPS =  3,28310147078@-03
X[  0] =   5,39410272212@+03
X[  1] = ¯3,58453469098@+05
X[  2] =   5,92696646630@+06
X[  3] = ¯4,17172676402@+07
X[  4] =   1.51816446535@+08
X[  5] = ¯3,08905188982@+08
X[  6] =   3,54640223455@+08
X[  7] = ¯2.14618733457@+08
X[  8] =   5,32156997712@+07
RESIDUAL[ 10] = ¯1.04675292969@-02
RESIDUAL[  6] =  2.13623046875@-04
RESIDUAL[ 12] = ¯1.06506347656@-02
RESIDUAL[ 13] = ¯3.57055664063@-03
RESIDUAL[ 11] = ¯1.31225585938@-02
RESIDUAL[ 15] =  5.70678710940@-03
RESIDUAL[  7] =  4.85229492188@-03
EXCHANGE EQUATION  11 WITH EQUATION   9


REFERENCE SET:
    0     2    14     1     5    16     3     8     4    11
```

```
EPS =  5.27490255522@-03
X[  0] =  6.25957197210@+03
X[  1] = -4.08466807872@+05
X[  2] =  6.66013000470@+06
X[  3] = -4.63595775586@+07
X[  4] =  1.67191156159@+08
X[  5] = -3.37645239754@+08
X[  6] =  3.85193433377@+08
X[  7] = -2.31855808203@+08
X[  8] =  5.72232867450@+07
RESIDUAL[ 10] = -2.47192382813@-03
RESIDUAL[  6] = -4.02832031250@-03
RESIDUAL[ 12] = -4.15039062500@-03
RESIDUAL[ 13] =  8.54492187500@-04
RESIDUAL[  9] =  2.71606445313@-03
RESIDUAL[ 15] =  5.40161132813@-03
RESIDUAL[  7] =  2.86865234375@-03
EXCHANGE EQUATION  15 WITH EQUATION  14


REFERENCE SET:
     0     2    15     1     5    16     3     8     4    11
EPS =  5.28695616646@-03
X[  0] =  6.26683727230@+03
X[  1] = -4.08903084965@+05
X[  2] =  6.66673904500@+06
X[  3] = -4.64026386640@+07
X[  4] =  1.67337406497@+08
X[  5] = -3.37924821718@+08
X[  6] =  3.85496685423@+08
X[  7] = -2.32030023004@+08
X[  8] =  5.72644634710@+07
RESIDUAL[ 10] = -2.71606445313@-03
RESIDUAL[  6] = -4.42504882813@-03
RESIDUAL[ 12] = -4.08935546875@-03
RESIDUAL[ 13] =  4.57763671875@-04
RESIDUAL[  9] =  2.38037109375@-03
RESIDUAL[ 14] =  5.24902343750@-03
RESIDUAL[  7] =  2.80761718750@-03

TERMINATION

NUMBER OF EXCHANGES WAS    3
```

```
TIME IN SECONDS =     2.18

SOLUTION VECTOR:
X[  0] =  6.26683727230@+03
X[  1] = -4.08903084965@+05
X[  2] =  6.66673904500@+06
X[  3] = -4.64026386640@+07
X[  4] =  1.67337406497@+08
X[  5] = -3.37924821718@+08
X[  6] =  3.85496685423@+08
X[  7] = -2.32030023004@+08
X[  8] =  5.72644634710@+07
```

```
REFERENCE SET:
   0    2  15    1    5   16    3    8    4   11

RESIDUALS:
R[  0] =  5.03054932963@-03
R[  1] = ¯5.42299644908@-03
R[  2] =  5.21381948085@-03
R[  3] = ¯5.37152852515@-03
R[  4] =  5.21255879778@-03
R[  5] = ¯5.29412850653@-03
R[  6] = ¯4.35175706136@-03
R[  7] =  2.57771749335@-03
R[  8] =  5.29555141585@-03
R[  9] =  2.44170932121@-03
R[ 10] = ¯2.67697239932@-03
R[ 11] = ¯5.35566792433@-03
R[ 12] = ¯4.13586084108@-03
R[ 13] =  6.37302589960@-04
R[ 14] =  5.32223943878@-03
R[ 15] =  5.24279958825@-03
R[ 16] = ¯5.26047150140@-03
```

C

```
***** HILBERT DATA *****

 17 EQUATIONS IN    9 UNKNOWNS

TABLEAU-JORDAN ALGORITHM WITH CLEAN-UPS

COMPUTATION:


REFERENCE SET:
     0     2    11     1     5    16     3     8     4     9
EPS =   1.65552306833@-03
X[  0] =   3.96754579750@+03
X[  1] =  -2.70186705478@+05
X[  2] =   4.55729602267@+06
X[  3] =  -3.26142253588@+07
X[  4] =   1.20377941299@+08
X[  5] =  -2.47939286842@+08
X[  6] =   2.87688384861@+08
X[  7] =  -1.75736324827@+08
X[  8] =   4.39371863702@+07
RESIDUAL[ 10] =  -2.59399414063@-03
RESIDUAL[  6] =   1.12915039063@-03
RESIDUAL[ 12] =   1.00135803223@-02
RESIDUAL[ 13] =   1.95732116699@-02
RESIDUAL[ 14] =   2.48756408691@-02
RESIDUAL[ 15] =   1.98707580566@-02
RESIDUAL[  7] =   3.50952148438@-03
EXCHANGE EQUATION  14 WITH EQUATION  11


REFERENCE SET:
     0     2    14     1     5    16     3     8     4     9
EPS =   3.28310147078@-03
X[  0] =   5.39410272212@+03
X[  1] =  -3.58453469098@+05
X[  2] =   5.92696646630@+06
X[  3] =  -4.17172676402@+07
X[  4] =   1.51816446535@+08
X[  5] =  -3.08905188982@+08
X[  6] =   3.54640223455@+08
X[  7] =  -2.14618733457@+08
X[  8] =   5.32156997712@+07
RESIDUAL[ 10] =  -1.04675292969@-02
RESIDUAL[  6] =   2.13623046875@-04
RESIDUAL[ 12] =  -1.06506347656@-02
RESIDUAL[ 13] =  -3.57055664063@-03
RESIDUAL[ 11] =  -1.31225585938@-02
RESIDUAL[ 15] =   5.70678710940@-03
RESIDUAL[  7] =   4.85229492188@-03
EXCHANGE EQUATION  11 WITH EQUATION   9


REFERENCE SET:
     0     2    14     1     5    16     3     8     4    11
```

```
EPS =  5.27490255522@-03
X[  0] =  6.25957197210@+03
X[  1] = -4.08466807872@+05
X[  2] =  6.66013000470@+06
X[  3] = -4.63595775586@+07
X[  4] =  1.67191156159@+08
X[  5] = -3.37645239754@+08
X[  6] =  3.85193433377@+08
X[  7] = -2.31855808203@+08
X[  8] =  5.72232867450@+07
RESIDUAL[ 10] = -2.47192382813@-03
RESIDUAL[  6] = -4.02832031250@-03
RESIDUAL[ 12] = -4.15039062500@-03
RESIDUAL[ 13] =  8.54492187500@-04
RESIDUAL[  9] =  2.71606445313@-03
RESIDUAL[ 15] =  5.40161132813@-03
RESIDUAL[  7] =  2.86865234375@-03
EXCHANGE EQUATION  15 WITH EQUATION  14


REFERENCE SET:
    0     2    15     1     5    16     3     8     4    11
EPS =  5.28695616646@-03
X[  0] =  6.26683727230@+03
X[  1] = -4.08903084965@+05
X[  2] =  6.66673904500@+06
X[  3] = -4.64026386640@+07
X[  4] =  1.67337406497@+08
X[  5] = -3.37924821718@+08
X[  6] =  3.85496685423@+08
X[  7] = -2.32030023004@+08
X[  8] =  5.72644634710@+07
RESIDUAL[ 10] = -2.71606445313@-03
RESIDUAL[  6] = -4.42504882813@-03
RESIDUAL[ 12] = -4.08935546875@-03
RESIDUAL[ 13] =  4.57763671875@-04
RESIDUAL[  9] =  2.38037109375@-03
RESIDUAL[ 14] =  5.24902343750@-03
RESIDUAL[  7] =  2.80761718750@-03

DOUBLE-PRECISION IMPROVEMENT

REFERENCE SET:
   0    2   15    1    5   16    3    8    4   11
 5.27844,60693,62039,81163,15@ -3 = EPS
 6.26576,18467,80947,90481,13@  3 = X[  0]
-4.08829,48947,36984,00166,41@  5 = X[  1]
 6.66547,14708,78323,27085,97@  6 = X[  2]
-4.63934,08832,78590,99856,31@  7 = X[  3]
 1.67302,98250,58417,63321,99@  8 = X[  4]
-3.37853,62341,23599,23544,09@  8 = X[  5]
 3.85414.13358,25437,64006,68@  8 = X[  6]
-2.31979,81579,50857.74466,80@  8 = X[  7]
 5.72519,98266,81440,61070,73@  7 = X[  8]
-2.62825,04150,30675,13559,69@ -3 = RESIDUAL[ 10]
```

```
-4.37417,88682,24776,19546,13@ -3 = RESIDUAL[  6]
-4.04298,63798,48512,16842,23@ -3 = RESIDUAL[ 12]
 7.29110,39988,26042,87569,16@ -4 = RESIDUAL[ 13]
 2.45568,04766,19148,67774,15@ -3 = RESIDUAL[  9]
 5.39490,96191,16761,14581,83@ -3 = RESIDUAL[ 14]
 2.54498,84901,42729,98658,72@ -3 = RESIDUAL[  7]
EXCHANGING EQUATION  15 WITH EQUATION  14

IMPROVEMENT COMPLETE - SOLUTION NOT YET ATTAINED


REFERENCE SET:
   16     0     2     8     1     4    11     3    14     5
EPS =  5.33509677215@-03
X[  0] =  6.27151768010@+03
X[  1] = -4.09134927009@+05
X[  2] =  6.66967430290@+06
X[  3] = -4.64188111470@+07
X[  4] =  1.67384270496@+08
X[  5] = -3.38001744455@+08
X[  6] =  3.85568624224@+08
X[  7] = -2.32065738301@+08
X[  8] =  5.72717628140@+07
RESIDUAL[ 10] = -2.99072265625@-03
RESIDUAL[  6] = -4.51660156250@-03
RESIDUAL[ 12] = -4.42504882813@-03
RESIDUAL[ 13] =  4.27246093750@-04
RESIDUAL[  9] =  2.19726562500@-03
RESIDUAL[ 15] =  5.03540039063@-03
RESIDUAL[  7] =  2.19726562500@-03

DOUBLE-PRECISION IMPROVEMENT

REFERENCE SET:
  16    0    2    8    1    4   11    3   14    5
 5.30006,47585,99124,11354,72@ -3 = EPS
 6.27879,92051,09148,09028,68@  3 = X[  0]
-4.09612,36199,45217,87480,48@  5 = X[  1]
 6.67733,07627,41163,91494,08@  6 = X[  2]
-4.64706,76438,19289,03505,14@  7 = X[  3]
 1.67565,40534,98812,94935,71@  8 = X[  4]
-3.38355,28056,64247,14797,90@  8 = X[  5]
 3.85958,25436,98969,13231,21@  8 = X[  6]
-2.32292,40280,66308,18546,28@  8 = X[  7]
 5.73258,79224,06183,38542,49@  7 = X[  8]
-2.61547,83860,90108,74222,34@ -3 = RESIDUAL[ 10]
-4.40861,93560,46417,08152,93@ -3 = RESIDUAL[  6]
-4.10282,78390,15554,81165,24@ -3 = RESIDUAL[ 12]
 6.41483,09155,76485,99234,65@ -4 = RESIDUAL[ 13]
 2.48599,96575,45923,11384,38@ -3 = RESIDUAL[  9]
 5.20435,98784,03230,77691,09@ -3 = RESIDUAL[ 15]
 2.53687,03218,72775,05812,51@ -3 = RESIDUAL[  7]

TERMINATION

NUMBER OF EXCHANGES WAS   3
NUMBER OF SOLUTION REFINEMENTS WAS   2
```

```
TIME IN SECONDS =     7.22

SOLUTION VECTOR:
X[   0] =  6.27879920510@+03
X[   1] = -4.09612361995@+05
X[   2] =  6.67733076280@+06
X[   3] = -4.64706764382@+07
X[   4] =  1.67565405350@+08
X[   5] = -3.38355280567@+08
X[   6] =  3.85958254369@+08
X[   7] = -2.32292402806@+08
X[   8] =  5.73258792240@+07
```

```
REFERENCE SET:
  16    0    2    8    1    4   11    3   14    5

RESIDUALS:
R[   0] =  5.14068166714@-03
R[   1] = ¯5.43694106409@-03
R[   2] =  5.18006117448@-03
R[   3] = ¯5.40693307699@-03
R[   4] =  5.20372052472@-03
R[   5] = ¯5.38778342253@-03
R[   6] = ¯4.48913723469@-03
R[   7] =  2.46245600966@-03
R[   8] =  5.23089072730@-03
R[   9] =  2.42137425212@-03
R[  10] = ¯2.67611798788@-03
R[  11] = ¯5.35718275265@-03
R[  12] = ¯4.15681161849@-03
R[  13] =  5.90306844980@-04
R[  14] =  5.25141799559@-03
R[  15] =  5.15800396980@-03
R[  16] = ¯5.34433614822@-03
```

D

```
***** HILBERT DATA *****

 17 EQUATIONS IN   9 UNKNOWNS

GOLUB-BARTELS PROCEDURE

COMPUTATION:

REFERENCE SET:
   0    2   11    1    5   16    3    8    4    9
EPS =   1.63874096800@-03
X[   0] =   3.95476513672@+03
X[   1] =  -2.69427188477@+05
X[   2] =   4.54586958984@+06
X[   3] =  -3.25401344453@+07
X[   4] =   1.20127178823@+08
X[   5] =  -2.47461156623@+08
X[   6] =   2.87170803023@+08
X[   7] =  -1.75439439202@+08
X[   8] =   4.38671014765@+07
RESIDUAL[  10] =  -2.65502929688@-03
RESIDUAL[   6] =   1.12915039063@-03
RESIDUAL[  12] =   1.00746154785@-02
RESIDUAL[  13] =   1.96876525879@-02
RESIDUAL[  14] =   2.49633789063@-02
RESIDUAL[  15] =   1.99508666992@-02
RESIDUAL[   7] =   3.63159179688@-03
EXCHANGING EQUATION  11 WITH EQUATION  14

REFERENCE SET:
   0    2   14    1    5   16    3    8    4    9
EPS =   3.27942767739@-03
X[   0] =   5.39426611328@+03
X[   1] =  -3.58494248535@+05
X[   2] =   5.92794839060@+06
X[   3] =  -4.17255830410@+07
X[   4] =   1.51850098514@+08
X[   5] =  -3.08978286177@+08
X[   6] =   3.54727695046@+08
X[   7] =  -2.14673084690@+08
X[   8] =   5.32293995821@+07
RESIDUAL[  10] =  -1.04675292969@-02
RESIDUAL[   6] =  -6.10351562500@-05
RESIDUAL[  12] =  -1.08947753906@-02
RESIDUAL[  13] =  -3.87573242188@-03
RESIDUAL[  11] =  -1.32446289063@-02
RESIDUAL[  15] =   5.70678710940@-03
RESIDUAL[   7] =   4.94384765625@-03
EXCHANGING EQUATION   9 WITH EQUATION  11

REFERENCE SET:
   0    2   14    1    5   16    3    8    4   11
EPS =   5.30913757220@-03
X[   0] =   6.27637304690@+03
X[   1] =  -4.09471829589@+05
```

```
X[   2] =   6.67528130660@+06
X[   3] =  -4.64577978574@+07
X[   4] =   1.67523101313@+08
X[   5] =  -3.38276837059@+08
X[   6] =   3.85875499209@+08
X[   7] =  -2.32246048897@+08
X[   8] =   5.73151735720@+07
RESIDUAL[ 10] =  -2.59399414063@-03
RESIDUAL[  6] =  -4.39453125000@-03
RESIDUAL[ 12] =  -4.15039062500@-03
RESIDUAL[ 13] =   6.40869140630@-04
RESIDUAL[  9] =   2.53295898438@-03
RESIDUAL[ 15] =   5.03540039063@-03
RESIDUAL[  7] =   2.28881835938@-03

ITERATIVE IMPROVEMENT

REFINED VALUES:
EPS =   5.30006475859@-03
X[   0] =   6.27879920510@+03
X[   1] =  -4.09612361995@+05
X[   2] =   6.67733076280@+06
X[   3] =  -4.64706764382@+07
X[   4] =   1.67565405350@+08
X[   5] =  -3.38355280567@+08
X[   6] =   3.85958254369@+08
X[   7] =  -2.32292402806@+08
X[   8] =   5.73258792240@+07
RESIDUAL[ 10] =  -2.67611798788@-03
RESIDUAL[  6] =  -4.48913723469@-03
RESIDUAL[ 12] =  -4.15681161849@-03
RESIDUAL[ 13] =   5.90306844980@-04
RESIDUAL[  9] =   2.42137425212@-03
RESIDUAL[ 15] =   5.15800396980@-03
RESIDUAL[  7] =   2.46245600966@-03

TERMINATION

NUMBER OF EXCHANGES MADE WAS    2
NUMBER OF SOLUTION REFINEMENTS WAS   1
```

```
TIME IN SECONDS =     3.10

SOLUTION VECTOR:
X[  0] =  6.27879920510@+03
X[  1] = -4.09612361995@+05
X[  2] =  6.67733076280@+06
X[  3] = -4.64706764382@+07
X[  4] =  1.67565405350@+08
X[  5] = -3.38355280567@+08
X[  6] =  3.85958254369@+08
X[  7] = -2.32292402806@+08
X[  8] =  5.73258792240@+07
```

```
REFERENCE SET:
    0    2  14     1     5   16     3     8     4   11

RESIDUALS:
R[   0] =   5.14068166714@-03
R[   1] =  -5.43694106409@-03
R[   2] =   5.18006117448@-03
R[   3] =  -5.40693307699@-03
R[   4] =   5.20372052472@-03
R[   5] =  -5.38778342253@-03
R[   6] =  -4.48913723469@-03
R[   7] =   2.46245600966@-03
R[   8] =   5.23089072730@-03
R[   9] =   2.42137425212@-03
R[  10] =  -2.67611798788@-03
R[  11] =  -5.35718275265@-03
R[  12] =  -4.15681161849@-03
R[  13] =   5.90306844980@-04
R[  14] =   5.25141799559@-03
R[  15] =   5.15800396980@-03
R[  16] =  -5.34433614822@-03
```