ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

ORNL/TM-11477

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0315425 3

An Improved Method for One-Way Dissection with Singular Diagonal Blocks

Jesse L. Barlow
Udaya B. Vemulapati

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.
NTIS price codes—Printed Copy: A12 Microfiche: A01.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-11477

Engineering Physics and Mathematics Division

Mathematical Sciences Section

An Improved Method for One-Way Dissection with Singular Diagonal Blocks

Jesse L. Barlow
and
Udaya B. Vemulapati
Department of Computer Science
The Pennsylvania State University
University Park, PA 16802

Date Published: June 1990

Research supported by the National Science Foundation under grant no. CCR-8700172, the Air Force Office of Scientific Research under grant no. AFOSR-88-0161, and the Office of Naval Research under grant no. N0014-80-0517. Research performed in part at Oak Ridge National Laboratory and supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy under contract DE-AC05-840R21400 with Martin Marietta Energy Systems, Inc. Research supported by the Office of Naval Research under grant no. N00024085-C-6041.

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-ACO5-840R21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0315425 3

CONTENTS

Abstract v

1. Introduction 1

2. Description of Algorithms 1

3. Error Analysis of the Revised Algorithms 8

4. Tests and Conclusions 10
   4.1 Stability Tests 10
   4.2 Hypercube Implementation 11

      4.2.1 Host Program 11
      4.2.2 Node Program 11
      4.2.3 Complexity of the Parallel Algorithm 12

5. Acknowledgement 13

ABSTRACT

We consider matrices arising out of the one-way dissection method for solving large sparse systems of linear equations. The systems that we consider are those that may have singular diagonal blocks. Such systems arise in certain fluid flow problems.

Gunzberger and Nicholaides [Lin. Alg. Appl. 64,pp.183-189,1985] proposed a method for resolving the singularity in the diagonal blocks. This method uses the Moore-Penrose pseudoinverse. We propose two improvements to the Gunzberger-Nicholaides procedure: (1) The substitution of a weighted pseudoinverse for the Moore-Penrose pseudoinverse; (2) A more elegant implementation of the back substitution procedure. A stability analysis of both our procedure and the Gunzberger-Nicholaides procedure is given. Both our analysis and empirical tests show that our method has better numerical stability properties than the Gunzberger-Nicholaides procedure. We also implement our procedure on Intel iPSC/1 Hypercube. Our improvement to the back substitution method makes the natural parallelism in the problem easier to exploit.

1 Introduction

The basic problem is to solve the $n \times n$ system of linear equations

$$Ax = s, \tag{1.1}$$

where A and s have the form

$$A = \begin{bmatrix} B_1 & 0 & 0 & \cdot & S_1 \\ 0 & B_2 & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & B_k & S_k \\ G_1^T & G_2^T & \cdot & G_k^T & F \end{bmatrix}; \quad s = (s_1, s_2, \ldots, s_k, s_{k+1})^T. \tag{1.2}$$

Here $B_i, i = 1, 2, \ldots, k$, are $m_i \times m_i$ matrices, F is a $p \times p$ matrix and G_i and S_i are $m_i \times p$ matrices, where $p + \sum_{i=1}^{k} m_i = n$. Each $s_i, i = 1, 2, \ldots, k$, is an m_i-vector and s_{k+1} is a p-vector. This is a matrix that arises out of the so-called one-way dissection ordering. Much of the discussion of one-way dissection in the literature has concerned symmetric, positive definite systems. This implies, that $B_i, i = 1, 2, \ldots, k$, and F are symmetric, positive definite, and $G_i = S_i, i = 1, 2, \ldots, k$. Instead we make the much weaker assumption that $rank(A) = n$, i.e. , that A is nonsingular. Thus we have that $rank(B_i) = l_i \leq m_i, i = 1, 2, \ldots, k$. Applications of such systems are given in [9].

Gunzberger and Nicholaides [8] suggested an algorithm based upon Gaussian elimination with singular pivots. It uses the Moore-Penrose inverses of the diagonal blocks $B_i, i = 1, 2, \ldots, k$. The Moore-Penrose pseudoinverse of a matrix B, denoted by B^+, is the unique matrix satisfying the four Moore-Penrose conditions

$$\begin{array}{ll} 1.\ BB^+B = B & 3.\ (BB^+)^T = BB^+ \\ 2.\ B^+BB^+ = B^+ & 4.\ (B^+B)^T = B^+B \end{array} \tag{1.3}$$

We will use the notation $B^{(i)}$,$B^{(i,j)}$, or $B^{(i,j,k)}$ to denote matrices satisfying conditions i,j, or k among those in (1.3). The procedure in [8] has a simple elimination procedure, but a complicated back substitution procedure.

In this paper, we suggest an alternative method for resolving the singularity in the diagonal blocks $B_i, i = 1, 2, \ldots, k$. This method is based upon the weighted pseudoinverse discussed in a fundamental paper by Elden [5]. We give evidence that this method is more stable. We also give a more elegant back substitution procedure, which makes the algorithm easier to implement on a message passing architecture. These algorithms are outlined in section two. An error analysis of our proposed algorithm is given in section three. Empirical tests verifying the stability properties of our algorithm are given in section four. We also give an implementation on Intel Hypercube(iPSC/1) in section four. The implementation is very straightforward.

2 Description of Algorithms

We first describe the elimination procedure of Gunzberger and Nicholaides [8] for solving (1.1). It makes use of the Moore-Penrose pseudoinverses of the diagonal blocks $B_i, i = 1, 2, \ldots, k$. The other elimination procedures in this section will take a similar form.

Algorithm 1 *Block Elimination using the Moore-Penrose Pseudoinverse [8]*

1. Compute

$$\tilde{F} = F - \sum_{i=1}^{k} G_i^T B_i^+ S_i \ ; \quad \tilde{s}_{k+1} = s_{k+1} - \sum_{i=1}^{k} G_i^T B_i^+ s_i$$

$$\tilde{G}_i^T = G_i^T (I - B_i^+ B_i) \quad \textit{projection of } G_i^T \textit{ onto orthogonal complement of } Range(B_i)$$

2. *For $i = 1, 2, \ldots, k$ find an $m_i \times (m_i - l_i)$ matrix X_i such that $B_i X_i = 0$. Note that $l_i = rank(B_i)$. Thus X_i is a basis for the null space of B_i. Algorithms for finding such a basis are given by Heath [10] and Pothen [12].*

We note that the terms $G_i^T B_i^+ S_i$, $G_i^T B_i^+ s_i$, $i = 1, 2, \ldots, k$, can be computed independently, as can the null space bases $X_i, i = 1, 2, \ldots, k$. The same is true for the $\tilde{G}_i^T, i = 1, 2, \ldots, k$, but we will see later that it is not necessary to compute these matrices at all.

The back substitution phase of the Gunzberger-Nicholaides procedure is somewhat complicated. Let $x = (x_1, x_2, \ldots, x_k, x_{k+1})^T$, where the block components $x_i, i = 1, 2, \ldots, k, k+1$, are of the form

$$x_i = y_i + z_i, \quad \text{where } y_i^T z_i = 0, i = 1, 2, \ldots, k, k+1, \tag{2.1}$$

and the vectors z_i satisfy

$$B_i z_i = 0, \quad i = 1, 2, \ldots, k \quad , \tag{2.2a}$$

$$\tilde{F} z_{k+1} = 0 \quad . \tag{2.2b}$$

Since A is nonsingular, $\tilde{F}$ is also nonsingular (cf. [8]). Thus

$$z_{k+1} = 0 \quad , \tag{2.3a}$$

$$x_{k+1} = y_{k+1} \quad . \tag{2.3b}$$

Then Algorithm 1 reduces (1.1) to the system

$$B_i y_i + S_i x_{k+1} = s_i, \quad i = 1, 2, \ldots, k, \tag{2.4a}$$

$$\sum_{i=1}^{k} \tilde{G}_i^T z_i + \tilde{F} x_{k+1} = \tilde{s}_{k+1}. \tag{2.4b}$$

From (2.2a), $\tilde{G}_i^T z_i = G_i^T z_i, i = 1, 2, \ldots, k$, thus we can replace (2.4b) with

$$\sum_{i=1}^{k} G_i^T z_i + \tilde{F} x_{k+1} = \tilde{s}_{k+1}. \tag{2.5}$$

Thus $\tilde{G}_i^T$ need never be explicitly computed. The system (2.4) can be written

$$My = \tilde{s} - Nz \tag{2.6}$$

where

$$M = \begin{bmatrix} B_1 & 0 & \cdot & 0 & S_1 \\ 0 & B_2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & B_k & S_k \\ 0 & \cdot & \cdot & 0 & \tilde{F} \end{bmatrix} \tag{2.7a}$$

$$N = \begin{bmatrix} 0 & \cdot & \cdot & \cdot & S_1 \\ 0 & \cdot & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & S_k \\ G_1^T & G_2^T & \cdot & G_k^T & 0 \end{bmatrix} \tag{2.7b}$$

$$\begin{aligned} \tilde{s} &= (s_1, s_2, \ldots, s_k, \tilde{s}_{k+1})^T; \\ y &= (y_1, \ldots, y_k, y_{k+1})^T; \\ z &= (z_1, \ldots, z_k, z_{k+1})^T \end{aligned} \tag{2.7c}$$

The consistency of (2.6) and the nonsingularity of $\tilde{F}$ follow from the nonsingularity of A. If we assume that z is known, and let

$$f = (f_1, f_2, \ldots, f_k, f_{k+1})^T = \tilde{s} - Nz \tag{2.8}$$

then a basic (non-unique) solution y is given by

$$y_{k+1} = x_{k+1} = \tilde{F}^{-1} f_{k+1} \tag{2.9a}$$

$$y_i = B_i^+ (f_i - S_i y_{k+1}) \tag{2.9b}$$

From [8], we have that y solves (2.6). Thus if we define the matrix Φ such that

$$y = \Phi f, \tag{2.10}$$

where Φ has the form

$$\Phi = \begin{bmatrix} B_1^+ & 0 & \cdot & \cdot & -B_1^+ S_1 \tilde{F}^{-1} \\ \cdot & B_2^+ & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & B_k^+ & -B_k^+ S_k \tilde{F}^{-1} \\ 0 & \cdot & \cdot & 0 & \tilde{F}^{-1} \end{bmatrix}, \tag{2.11}$$

we note the following fact about Φ. Its proof is obvious.

Lemma 2.1 $\Phi = M^{(1,2,4)}$. *That is, Φ is a (1,2,4)-pseudoinverse of M.*

If we combine (2.6) and (2.10) we have

$$(I - M\Phi)Nz = (I - M\Phi)\tilde{s}. \tag{2.12}$$

Let

$$X = diag(X_1, X_2, \ldots, X_k, 0),$$

where X_i are defined in Algorithm 1. Thus (2.12) becomes

$$Tw = g \tag{2.13}$$

where $T = (I - M\Phi)NX$; $z = Xw$;$g = (I - M\Phi)\tilde{s}$. Equation (2.13) is consistent but overdetermined (cf. [8]). It can be solved by an orthogonal factorization of T (in [8] ,the use of normal equations is advocated). Gunzberger and Nicholaides show that T must have full rank if A has full rank. We assume that the dimensions of the null spaces of $B_i, i = 1, 2, \ldots, k$, are much smaller than the dimensions of the blocks themselves. That is, $m_i - l_i \ll m_i$. Thus, the solution of (2.13) should be very fast compared with the rest of the algorithm. We state the procedure as Algorithm 2.

Algorithm 2 *Back Substitution Procedure [8]*

1. *Explicitly form $T = (I - M\Phi)NX; g = (I - M\Phi)\tilde{s}$.*
2. *Solve $Tw = g$ by orthogonal factorization (or normal equations).*
3. *Let $z = Xw$ and solve*
$$y = \Phi(\tilde{s} - Nz)$$
4. *The solution $x = y + z$.*

We propose two changes in Algorithms 1 and 2. The first is a simplification of the back substitution procedure. This simplification uses computations arising directly out of the elimination procedure. To describe that, we give a more specific version of Algorithm 1 which includes the method for computing $B_i^+, i = 1, 2, \ldots, k$. The method is slightly different from that given in [8], but uses the method for computing B_i^+ given in Golub and Van Loan[7, pp.162-167].

Algorithm 3 *Implementation of Block Elimination using the Moore-Penrose pseudoinverse*

1. *For $i = 1, 2, \ldots, k$ perform steps 2-7.*

2. *Factor B_i into*

$$B_i = Q_i \begin{bmatrix} U_i^{[1]} & U_i^{[2]} \\ 0 & 0 \end{bmatrix} P_i^T$$

where Q_i is orthogonal, $U_i^{[1]}$ is an $l_i \times l_i$ upper triangular, $U_i^{[2]}$ is an $l_i \times (m_i - l_i)$ matrix, and P_i is a permutation matrix. This factorization and the determination of rank l_i can be done by orthogonal decomposition with column pivoting (cf. [11, Chapter 10]) or some other method(cf. [2,4,6]).

3. *Compute*

$$\begin{bmatrix} S_i^{[1]} & s_i^{[1]} \\ S_i^{[2]} & s_i^{[2]} \end{bmatrix} = Q_i^T (S_i, s_i)$$

where $S_i^{[1]}$ is $l_i \times p$ and $S_i^{[2]}$ is $(m_i - l_i) \times p$.

4. *Solve for $\hat{S}_i$ and $\hat{s}_i$*

$$U_i^{[1]}(\hat{S}_i, \hat{s}_i) = (S_i^{[1]}, s^{[1]})$$

5. *Compute*

$$X_i = \begin{bmatrix} -[U_i^{[1]}]^{-1} U_i^{[2]} \\ I_{m_i - l_i} \end{bmatrix} \tag{2.14}$$

X_i is a common choice for the null basis matrix of B_i (cf.[10,12]).

6. *Factor*

$$X_i = Z_i \begin{bmatrix} W_i \\ 0 \end{bmatrix}$$

where Z_i is orthogonal and W_i is upper triangular and compute

$$(V_i, v_i) = Z_i \begin{bmatrix} 0 & 0 \\ 0 & I_{p-m_i+l_i} \end{bmatrix} Z_i^T (\hat{S}_i, \hat{s}_i)$$

The items (V_i, v_i) are projections of $(\hat{S}_i, \hat{s}_i)$ onto the space orthogonal to the null space of B_i, thus providing $B_i^+(S_i, s_i)$.

7. *Compute*

$$(R_i, r_i) = -G_i^T (V_i, v_i)$$

8. *Compute*

$$\tilde{F} = F + \sum_{i=1}^{k} R_i; \ \tilde{s}_{k+1} = s_{k+1} + \sum_{i=1}^{k} r_i$$

Algorithm 3 requires

$$2m_i l_i (m_i - l_i) + \tfrac{2}{3} l_i^3 + 2m_i l_i (p+1) + l_i^2 (m_i - l_i) + \\ (m_i - l_i)^2 (m_i - \tfrac{1}{3}(m_i - l_i)) + 4(m_i - l_i) m_i (p+1) + p(p+1) l_i + O(m^2)$$

flops for each $i = 1, 2, \ldots, k$. Let $m = \max_{1 \le i \le k} m_i$. If $p \le m$ and $|m_i - l_i| \le c = O(1)$ this simplifies to

$$2m_i l_i(p+1) + \frac{2}{3} l_i^3 + p(p+1) l_i + O(cm^2)$$

for each $i = 1, 2, \ldots, k$. We assume here that all of blocks in (1.2) are dense.

If we consider equation (2.4a) and apply the reduction from Algorithm 3, we have

$$U_i y_i + S_i^{[1]} x_{k+1} = s_i^{[1]} \tag{2.15a}$$

$$S_i^{[2]} x_{k+1} = s_i^{[2]}. \tag{2.15b}$$

where $U_i = (U_i^{[1]}, U_i^{[2]})$. Since equation (2.15) is just an orthogonal reduction of some rows from $Ax = s$, it follows that it is underdetermined but consistent. Using the null basis (2.14) for B_i and by letting

$$\hat{G}_i = G_i^T X_i$$

equation (2.5) becomes

$$\sum_{i=1}^{k} \hat{G}_i w_i + \tilde{F} x_{k+1} = \tilde{s}_{k+1} \tag{2.16}$$

where $z_i = X_i w_i$. Thus if we let $S^{[2]} = (S_i^{[2]}, \ldots, S_k^{[2]})^T$, $s^{[2]} = (s_i^{[2]}, \ldots, s_k^{[2]})^T$, and $\hat{G} = (\hat{G}_1, \ldots, \hat{G}_k)$, then x_{k+1} and $w = (w_1, w_2, \ldots, w_k)^T$ solve the linear system

$$\begin{bmatrix} \hat{G} & \tilde{F} \\ 0 & S^{[2]} \end{bmatrix} \begin{bmatrix} w \\ x_{k+1} \end{bmatrix} = \begin{bmatrix} \tilde{s}_{k+1} \\ s^{[2]} \end{bmatrix}. \tag{2.17}$$

The nonsingularity of A guarantees that (2.15) is a nonsingular system of linear equations. For problems arising in practice, its dimension will be small compared to the dimension of A. It can be solved by Gaussian elimination with partial pivoting or orthogonal decomposition. Such a reduction is much simpler than the back substitution procedure in Algorithm 2. The values of y_i and x_i, $i = 1, 2, \ldots, k$ can be recovered from (2.9a) and the step

$$x_i = y_i + X_i w_i. \tag{2.18}$$

The computation (2.9a) can be simplified into

$$y_i = [U]_i^+ (s_i^{[1]} - S_i^{[1]} x_{k+1}) \tag{2.19}$$

thereby avoiding the reuse of the orthogonal factor Q_i. We now formally state this procedure as Algorithm 4. This algorithm is a method for solving (2.5) and is simply a particular implementation of Algorithm 2.

Algorithm 4 *Improved Back Substitution Procedure*

1. *Solve the linear system in equation (2.17) for x_{k+1} and $w = (w_1, w_2, \ldots, w_k)^T$ using orthogonal factorization by Householder transformations.*

2. *For $i = 1, 2, \ldots, k$, do steps 3–6*

3. *Compute*

$$f_i^{[1]} = s_i^{[1]} - S_i^{[1]} x_{k+1}. \tag{2.20}$$

4. Let

$$g_i = -\begin{bmatrix} I_{l_i} \\ 0 \end{bmatrix} [U_i^{[1]}]^{-1} f_i^{[1]}, \tag{2.21}$$

$$\hat{g}_i = (I_{m_i - l_i}, 0) Z_i^T g_i.$$

5. Solve

$$\begin{bmatrix} U_i^{[1]} & U_i^{[2]} \\ 0 & W_i \end{bmatrix} y_i = \begin{bmatrix} f_i^{[1]} \\ \hat{g}_i \end{bmatrix}.$$

Here $S_i^{[1]}$, $s_i^{[1]}$, $G_i^{[1]}$, $U_i^{[1]}$, W_i, and Z_i are from Algorithm 3.

6. Compute

$$x_i = y_i + X_i w_i,$$

where X_i is in Algorithm 3.

The back substitution procedure requires

$$\frac{2}{3}[p + \sum_{i=1}^{k}(m_i - l_i)]^3 + \sum_{i=1}^{k}[3l_i(m_i - l_i) + \frac{1}{2}m_i^2 + \frac{1}{2}l_i^2] + O(m)$$

flops. If $\max_{1 \le i \le k} |m_i - l_i| = c = O(1)$ then this reduces to

$$\frac{2}{3}[p + kc]^3 + \frac{1}{2}\sum_{i=1}^{k}[l_i^2 + m_i^2] + O(cm)$$

flops.

The second modification to Algorithms 1 and 2 is to replace B_i^+ with $B_i^{(1,3)}, i = 1, 2, \ldots, k$, .i.e. any matrix $B_i^{(1,3)}$ satisfying Penrose conditions 1 and 3. For the elimination algorithm, this is equivalent to solving (cf.[5])

$$\min_{(V_i, v_i)} \|B_i(V_i, v_i) - (S_i, s_i)\|_F$$

and then computing

$$\tilde{F} = F - \sum_{i=1}^{k} G_i^T V_i, \tag{2.22a}$$

$$\tilde{s}_{k+1} = s_{k+1} - \sum_{i=1}^{k} G_i^T v_i, \tag{2.22b}$$

$$\tilde{G}_i^T = G_i^T (I - B_i^{(1,3)} B_i). \tag{2.22c}$$

It is essential that all of the columns of

$$(H_i, h_i) = (S_i, s_i) - B_i(V_i, v_i) \tag{2.23}$$

be vectors in the space orthogonal to the columns of B_i. It is guaranteed by the use of $B_i^{(1,3)}$. This allows us to set up equation (2.17) by orthogonal factorization of B_i by column pivoting or some other method to detect rank (e.g. [2,4,6]). When we substitute $B_i^{(1,3)}$ for B_i^+, we lose the property that

$y_i^T z_i = 0$, but this property is not necessary for the algorithm to work. Again since $\tilde{G}_i^T z_i = G_i^T z_i$, it is not necessary to do the computation (2.22c).

The matrix $B_i^{(1,3)}$ is not unique unless B_i has full rank. In our modified algorithm, we can choose $B_i^{(1,3)}$ so as to minimize $||G_i^T B_i^{(1,3)} S_i||_F$ and $||G_i^T B_i^{(1,3)} s_i||_2$. As will be shown in the next section this leads to a new algorithm with better numerical stability properties. Elden [5] showed that the (1,3) pseudoinverse with this property is the weighted pseudoinverse defined below.

Definition 2.1 *The G-weighted pseudoinverse of B is defined by*

$$B_G^+ = (I - (G^T P)^+ G^T) B^+$$

where

$$P = I - B^+ B.$$

In [5], it is shown that the matrix B_G^+ is the (1,3)-inverse such that

$$||G^T B_G^+ S||_F \leq ||G^T B^{(1,3)} S||_F \tag{2.24}$$

for all (1,3)-inverses of B and matrices S. The G-weighted pseudoinverses $[B_G]_i^+$ need not and should not be explicitly computed. Instead we compute the quantities

$$R_i = -G_i^T [B_G]_i^+ S_i \quad i = 1, 2, \ldots, k, \tag{2.25a}$$

$$r_i = -G_i^T [B_G]_i^+ s_i \quad i = 1, 2, \ldots, k, \tag{2.25b}$$

and then compute

$$\tilde{F} = F + \sum_{i=1}^{k} R_i; \quad \tilde{s}_{k+1} = s_{k+1} + \sum_{i=1}^{k} r_i. \tag{2.26}$$

The quantities (R_i, r_i) are simply the residuals of the least squares problem

$$\min_{(V_i, v_i) \in T_{B_i}} ||G_i^T (V_i, v_i)||_F, \tag{2.27}$$

where T_{B_i} is the set of minimizers of

$$\min_{(V_i, v_i) \in \mathbb{R}^{m_i \times (p+1)}} ||B_i (V_i, v_i) - (S_i, s_i)||_F.$$

The computation of (V_i, v_i) is not necessary. The residuals (R_i, r_i) can be computed directly. The problem (2.27) has an unique solution if $rank \begin{bmatrix} B_i \\ G_i^T \end{bmatrix} = m_i, i = 1, 2, \ldots, k$. This is a direct consequence of nonsingularity of A. We now give a more detailed description of this procedure. Steps 1-4 are the Björck-Golub(cf.[3]) direct elimination procedure for solving (2.27).

Algorithm 5 *Block Elimination Scheme Using the weighted Pseudoinverse*

1. *For $i = 1, 2, \ldots, k$, do steps 2-5*
2. *Same as Steps 2-3 of Algorithm 3.*
3. *Let $G_i^T = (G_i^{[1]}, G_i^{[2]})$ where $G_i^{[1]}$ is a $p \times l_i$ matrix and $G_i^{[2]}$ is a $p \times (m_i - l_i)$ matrix. Compute $\hat{G}_i = G_i^{[2]} - G_i^{[1]} [U_i^{[1]}]^{-1} U_i^{[2]}$ and $(\hat{S}_i, \hat{s}_i) = -G_i^{[1]} [U_i^{[1]}]^{-1} (S_i, s_i)$. (Note that $\hat{G}_i = G_i^T X_i$).*

4. Factor

$$\hat{G}_i = Z_i \begin{bmatrix} W_i \\ 0 \end{bmatrix},$$

where Z_i is orthogonal and W_i is upper triangular. Then compute

$$(R_i, r_i) = Z_i \begin{bmatrix} 0 & 0 \\ 0 & I_{p-n_i+l_i} \end{bmatrix} Z_i^T (\hat{S}_i, \hat{s}_i).$$

5. Compute

$$\tilde{F} = F + \sum_{i=1}^{k} R_i; \quad \tilde{s}_{k+1} = s_{k+1} + \sum_{i=1}^{k} r_i.$$

With the change that

$$g_i = -G_i^{[1]}[U_i^{[1]}]^{-1} f_i^{[1]} \tag{2.28}$$

in (2.21), the back substitution procedure in Algorithm 4 can be used directly after Algorithm 5. This adds an additional $l_i p$ flops for each $i = 1, 2, \ldots, k$. Except for differences in terms of $O(m^2)$, the operation count for Algorithm 5 is identical to that of Algorithm 3. We note however, one difference that the matrix

$$\tilde{\Phi} = \begin{bmatrix} B_1^{(1,3)} & 0 & \cdot & \cdot & -B_1^{(1,3)} S_1 \tilde{F}^{-1} \\ \cdot & B_2^{(1,3)} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & B_k^{(1,3)} & -B_k^{(1,3)} S_k \tilde{F}^{-1} \\ 0 & \cdot & \cdot & 0 & \tilde{F}^{-1} \end{bmatrix}$$

is only a (1)-pseudoinverse of M. This can be verified easily. However this is enough to assure that $y = \tilde{\Phi} f$ satisfies (2.10) . Hence we can use the back substitution procedure in Algorithm 4.

The stability properties of these direct elimination procedures can be shown using well known properties of methods for solving constrained least squares problems and systems of linear equations (cf. [7,13]). These properties are given in the next section.

3 Error Analysis of the Revised Algorithms

We now use backward error analysis (cf. [13]) to bound the errors in the computational versions of the algorithms in section one. The general form of the reductions are

$$\tilde{F} = F - \sum_{i=1}^{k} G_i^T B_i^{(1,3)} S_i; \quad \tilde{s}_{k+1} = s_{k+1} - \sum_{i=1}^{k} G_i^T B_i^{(1,3)} s_i \tag{3.1}$$

where $B_i^{(1,3)}$ is a (1,3) pseudoinverse of B_i. First we need the following lemma from [1] on the Björck-Golub direct elimination procedure as applied in steps 1-4 of Algorithm 3.

Lemma 3.1 *Let steps 2-5 of Algorithm 5 be implemented using Householder transformations in floating point arithmetic with machine unit μ. Then the factorization of each of the blocks $\begin{bmatrix} B_i \\ G_i^T \end{bmatrix}$, $i = 1, 2, \ldots, k$, satisfy*

$$\begin{bmatrix} B \\ G^T \end{bmatrix} = Y \begin{bmatrix} U_1 & U_2 \\ 0 & 0 \\ 0 & W \\ 0 & 0 \end{bmatrix} P^T + \begin{bmatrix} \delta B \\ \delta G^T \end{bmatrix} \tag{3.2a}$$

$$(S, s) = Y \begin{bmatrix} \hat{S}^{[1]} & \hat{s}^{[1]} \\ \hat{S}^{[2]} & \hat{s}^{[2]} \end{bmatrix} + (\delta S, \delta s) \tag{3.2b}$$

where $Y = Q_1 L Q_2$, Q_1 and Q_2 are orthogonal, and L is unit lower triangular. The backward errors δB, δG, δS, and δs satisfy

$$\|\delta B\|_F \leq \phi_B \|B\|_F \, \mu + O(\mu^2), \tag{3.3a}$$

$$\|\delta G\|_F \leq \phi_G \, \tau_G \left\| \begin{pmatrix} B \\ G^T \end{pmatrix} \right\|_F \mu + O(\mu^2), \tag{3.3b}$$

$$\|(\delta S, \delta s)\|_F \leq \phi_S \, \tau_S \|(S, s)\|_F \, \mu + O(\mu^2), \tag{3.3c}$$

where ϕ_B, ϕ_G, and ϕ_S are modestly sized polynomials in the dimension of B, G, and S, and

$$\tau_G = \max\{\|[U^{[1]}]^{-1} U^{[2]}\|_F \, , \max_{1 \leq q \leq l} 1 + \|[U^{[1]}_{(q)}]^{-1} U^{[2]}_{(q)}\|_F\}, \tag{3.4}$$

$$\tau_S = \|G^{[1]}[U^{[1]}]^{-1}\|_2, \tag{3.5}$$

where $U_{(q)} = (U^{[1]}_{(q)}, U^{[2]}_{(q)})$ is the first q rows of U, $U^{[1]}_{(q)}$ is a $q \times q$ nonsingular matrix, and $U^{[2]}_{(q)}$ is a $q \times (n-q)$ matrix, $q = 1, 2, \ldots, l$.

A backward error analysis of the reduction stage of Algorithm 5 can be obtained from this theorem by substituting I for G^T.

Let $(\hat{R}_i, \hat{r}_i), i = 1, 2, \ldots, k$,$\hat{F}$, and $\hat{s}_{k+1}$ be the computed values of (R_i, r_i), $\tilde{F}$, and $\tilde{s}_{k+1}$ from Algorithm 3 or 5. Then we have the computational equations

$$(\hat{R}_i, \hat{r}_i) = (G_i + \delta G_i)^T (B_i + \delta B)^{(1,3)} (S_i + \delta S_i, s_i + \delta s_i),$$

$$\tilde{F} = F + \delta F + \sum_{i=1}^{k} \hat{R}_i = fl\left(F + \sum_{i=1}^{k} \hat{R}_i\right),$$

$$\tilde{s}_{k+1} = s_{k+1} + \delta s_{k+1} + \sum_{i=1}^{k} \hat{r}_i = fl\left(s_{k+1} + \sum_{i=1}^{k} \hat{r}_i\right),$$

where δG_i, δB_i, $(\delta S_i, \delta s_i)$ are errors that can be bounded by Lemma 3.1 and $fl(\cdot)$ denotes the floating point computation of the contents. The errors δF and δs_{k+1} are just the errors in the floating point sums. Thus from standard bounds on errors in sums we have

$$\|\delta F\|_F \leq \phi_F \, \mu \max\{\|F\|_F, \max_{1 \leq i \leq k} \|R_i\|_F\} + O(\mu^2),$$

$$\|\delta s_{k+1}\|_2 \leq \phi_S \, \mu \max\{\|s_{k+1}\|_2, \max_{1 \leq i \leq k} \|r_i\|_2\} + O(\mu^2),$$

where ϕ_F and ϕ_S are modestly sized polynomials in the dimensions of A. From the definition of R_i and r_i we have

$$R_i = G_i^T B_i^{(1,3)} S_i; \quad r_i = G_i^T B_i^{(1,3)} s_i, \quad i = 1, 2, \ldots, k, \quad ;$$

thus

$$\|R_i\|_F \leq \|G_i^T B_i^{(1,3)}\|_2 \|S_i\|_F, \tag{3.6}$$

$$\|r_i\|_2 \leq \|G_i^T B_i^{(1,3)}\|_2 \|s_i\|_2. \tag{3.7}$$

Using the fact that

$$\begin{bmatrix} [U_i^{[1]}]^{-1} \\ 0 \end{bmatrix} = U_i^{(1,2,3)}$$

is a (1,2,3) pseudoinverse of U_i and (2.24) we have that

$$||G_i^T[B_G^+]_i||_2 \leq ||[G_i^{[1]}][U_i^{[1]}]^{-1}||_2. \tag{3.8}$$

This gives us the following bound for Algorithm 5. Note that if the Moore-Penrose pseudoinverse B_i^+ is substituted for $[B_G^+]_i$, the inequality (3.8) may be false since the Moore-Penrose pseudoinverse does not satisfy (2.24) . The following theorem summarizes our results.

Theorem 3.1 *Let Algorithm 3 or 5 be implemented using Householder transformations in floating point arithmetic with machine unit μ. Let the backward substitution phase be done using Algorithm 4. Then the computed solution $\bar{x}$ satisfies*

$$(A + \delta A)\bar{x} = s + \delta s$$

where

$$||\delta A||_F \leq \phi_A \, \tau_A ||A||_F \, \mu + O(\mu^2)$$

$$||\delta s||_2 \leq \phi_s \, \tau_A ||s||_2 \, \mu + O(\mu^2)$$

$$\tau_A = \max\{\max_{1\leq i\leq k} ||G_i^{[1]}[U_i^{[1]}]^{-1}||_2, \max_{1\leq i\leq k} ||G_i B_i^{(1,3)}||_2\}$$

and ϕ_A and ϕ_s are modestly sized polynomials in the dimension of A.

We now give a corollary that gives stronger stability results for Algorithm 5. It is a straightforward consequence of Theorem 3.1 and equation (3.8).

Corollary 3.1 *Let Algorithm 5 be implemented using Householder transformations in floating point arithmetic with machine unit μ. Then δA and δs in Theorem 3.1 satisfy*

$$||\delta A||_F \leq \phi_A \, \tau_G ||A||_F \, \mu + O(\mu^2),$$

$$||\delta s||_2 \leq \phi_s \, \tau_G ||s||_2 \, \mu + O(\mu^2),$$

where

$$\tau_G = \max_{1\leq i\leq k} ||G_i^{[1]}[U_i^{[1]}]^{-1}||_2,$$

and ϕ_A and ϕ_S are modestly sized polynomials in the dimension of A.

The bound τ_G arises out of of the Björck-Golub procedure. The factors $||G_i^T[B_G^+]_i||_2, i = 1, 2, \ldots, k$, arise out of the condition of each of the problems of the form (2.27). We note that the bound in Corollary 3.1 is smaller than that in Theorem 3.1. The Moore-Penrose inverse does not satisfy the inequality (3.8) and we know of no error bound as good as that in Corollary 3.1 for Algorithm 3.

Thus the error bounds obtained by this analysis are better for Algorithm 5 than for Algorithm 3. In the next section, we give numerical tests which seem to indicate that Algorithm 5 will give more reliable answers.

4 Tests and Conclusions

4.1 Stability Tests

We implemented Algorithms 3 and 5 in FORTRAN single precision on the SUN3 with the back substitution procedure in Algorithm 4. The two algorithms differ only in their computation of $B_i^{(1,3)}, i = 1, 2, \ldots, k$.

The matrix A is generated randomly. Rank one singularities are introduced into each diagonal block by replacing the last row of each such block by the sum of its other rows. Then the right hand side is formed by making the known solution vector $(1, 1, \ldots, 1)^T$. We then calculated the relative error in the solution. The results are shown in Table 1. Here the experiments clearly suggest that Algorithm 5 has better numerical stability properties than Algorithm 3. Thus we see that the use of the weighted pseudo-inverse rather than the Moore-Penrose pseudoinverse gives us a better method of resolving the singularity in the diagonal blocks.

4.2 Hypercube Implementation

To simplify the implementation on a Hypercube, it is assumed that each diagonal block B_i and F are of equal size i.e. $m_i = p, i = 1, 2, \ldots, k$, and that $p = k + 1$, i.e. the size of each diagonal block is also equal to the number of diagonal blocks. It then follows that $p^2 = n$. The number of processors in the Hypercube is denoted by P (numbered from 1 to P). It is further assumed that the number of diagonal blocks $k + 1$ is at least as large as the number of processors (P).

The blocks $B_i, i = 1, 2, \ldots, k$ are equally distributed among the first $P - 1$ processors, along with the corresponding S_i and G_i matrices. And the matrix F is processed by the node P. A brief description of the algorithm emphasizing the flow of data between the processors follows.

4.2.1 Host Program

generate matrix A and the vector s
compute the number of blocks that each node numbered from 1 to $P - 1$ gets
for $i := 1$ *to* $P - 1$
  send appropriate blocks of B, S, G *and* s to node i
send F to node P
wait for the solution parts to arrive from all the nodes

4.2.2 Node Program

if it is not the last node (P) then
  receive the matrix blocks B, G, S and s
  diagonalize each B_i and solve the LSE problem as described in Algorithm 3
  send the matrices $\hat{G}_i$ and $S_i^{[2]}$ along with $s_i^{[2]}, R_i$ and r_i to node P (cf. Algorithm 5)
  wait for x_{k+1} and w_i vectors to arrive from node P
  complete the solution process to get x_i
  send x_i's to the host

n	k+1	Estimated Condition No.	Error:Alg.3	Error:Alg.5
2	2	2.0E02	0	0
4	2	1.0E01	9.0E-6	6.0E-7
10	2	4.0E01	3.0E-6	2.0E-6
10	3	1.0E02	4.0E-6	2.0E-6
20	4	3.0E02	9.0E-6	3.0E-6
40	5	8.0E02	2.0E-4	4.0E-5
60	6	9.0E02	8.0E-5	7.0E-6
80	8	2.0E03	1.0E-4	2.0E-5
100	10	2.0E04	2.0E-3	5.0E-5

Table 1: Error in Algorithms 3 and 5 for Random Matrices

size of each block(p)	no. of processors(P)	time in seconds
8	8	1.22
8	4	1.46
8	2	2.64
16	16	5.36
16	8	7.86
16	4	11.46
32	32	34.12
32	16	48.62
32	8	71.94

Table 2: Timings results on Intel Hypercube

else
receive F from the host
receive the matrices $\hat{G}_i$ and $S_i^{[2]}, s_i^{[2]}, R_i$ and r_i sent by all other nodes
solve the system (2.17)
broadcast x_{k+1} and appropriate blocks of w_i to all the other $P-1$ nodes
send x_{k+1} to host

The above Algorithm was implemented in FORTRAN on an Intel hypercube(iPSC/1) at the ACRF facility at Argonne National Laboratory and the Table 2 shows the timings results from these experiments. The matrix in each case was an $p^2 \times p^2$ matrix. For a fixed value of p, the problem was run on cubes of different dimensions to determine the speed-up. *The time shown is elapsed time in seconds from the moment the host starts sending the data to the nodes till the final solution is returned to the host.*

It appears from the results that by increasing the number of processors by a factor of 2, one would get a speed-up by a factor of 1.43. The main reason is that the back substitution process has a bottleneck — the other nodes must remain idle while node P determines x_{k+1} and w.

4.2.3 Complexity of the Parallel Algorithm

It is assumed that the time required to transmit a message of N words from one node to another is $(\alpha + \beta N)d$ where α is the start-up time for the message and β is the time required to send one word after the initial message is set-up and d is the distance between the nodes.

The only communication required in the parallel algorithm described above is the transmission of $\hat{G}_i$, $S_i^{[2]}$, $s_i^{[2]}$, R_i and r_i to node P and vectors x, w from node P to nodes 1 to $P-1$. Since the size of R_i is much larger than other matrices and since the maximum distance between any two nodes on the Hypercube is $\log P$, it is easily seen that the upper bound on the communication complexity of the algorithm is $O([P\alpha + \beta(p^2 + Pn)] \log P)$.

The computational complexity is easier to bound because all the computational work except the solution of (2.17) is done in parallel and hence it is divided equally among $P-1$ processors. However the matrix in the system (2.17) is of the order $p + \sum_{i=1}^{n}(m_i - l_i)$ and hence only $\frac{2}{3}(p + \sum_{i=1}^{n}(m_i - l_i))^3$ are not done in parallel.

5 Acknowledgment

The authors would like to thank the Argonne National Laboratory for the usage of the Intel Hypercube at the ACRF facility. We would also like to acknowledge the help of John Gilbert and some anonymous referees.

References

[1] J. L. Barlow and S. L. Handy. The direct solution of weighted and equality constrained least squares problems. *SIAM Journal on Scientific and Statistical Computing*, 9(4):704–16, July 1988.

[2] Christan H. Bischof. Incremental condition estimation. *SIAM Journal on Matrix Analysis and Applications*, 11(2):312–322, April 1990.

[3] Å. Björck and Gene H. Golub. Iterative refinement of linear least squares solutions by Householder transformations. *BIT*, 7:322–337, 1967.

[4] T. F. Chan. Rank revealing QR factorizations. *Linear Algebra and Its Applications*, 88/89:67–82, 1987.

[5] L. Elden. Perturbation theory for the least squares problem with equality constraints. *SIAM Journal on Numerical Analysis*, 17:338–350, 1980.

[6] L. V. Foster. Rank and null space calculations using matrix decomposition without column pivoting. *Linear Algebra and Its Applications*, 74:47–72, 1986.

[7] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins Press, Baltimore, 1983.

[8] M. Gunzberger and R. Nicholaides. Elimination with noninvertible pivots. *Linear Algebra and Its Applications*, 64:183–189, 1985.

[9] M. Gunzberger and R. Nicholaides. On substructuring algorithms and solution techniques for the numerical approximation of partial differential equations. *Applied Numerical Mathematics*, 64:243–256, 1986.

[10] M. T. Heath. Some extensions of an algorithm for sparse linear least squares problems. *SIAM Journal on Scientific and Statistical Computing*, 3:223–237, 1982.

[11] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice Hall, Englewood Cliff, N.J., 1974.

[12] Alex Pothen. *Sparse Null Bases and Marriage Theorems*. PhD thesis, Cornell University, Ithaca,NY, 1984.

[13] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, London, 1965.

INTERNAL DISTRIBUTION

1. B. R. Appleton
2. J. J. Dongarra
3. J. B. Drake
4. G. A. Geist
5-6. R. F. Harbison
7. M. T. Heath
8. M. R. Leuze
9-13. F. C. Maienschein
14. E. G. Ng
15. C. E. Oliver
16. G. Ostrouchov
17-21. S. A. Raby
22. C. H. Romine
23. R. C. Ward
24. P. H. Worley
25. Central Research Library
26. ORNL Patent Office
27. K-25 Plant Library
28. Y-12 Technical Library/ Document Reference Station
29. Laboratory Records - RC
30-31. Laboratory Records Department

EXTERNAL DISTRIBUTION

32. Dr. Donald M. Austin, Office of Scientific Computing, Office of Energy Research, ER-7, Germantown Building, U.S. Department of Energy, Washington, DC 20545

33. Dr. Robert G. Babb, Department of Computer Science and Engineering, Oregon Graduate Center, 19600 N.W. Walker Road, Beaverton, OR 97006

34. Lawrence J. Baker, Exxon Production Research Company, P.O. Box 2189, Houston, TX 77252-2189

35-39. Dr. Jesse L. Barlow, Department of Computer Science, Pennsylvania State University, University Park, PA 16802

40. Dr. Edward H. Barsis, Computer Science and Mathematics, P.O. Box Box 5800, Sandia National Laboratory, Albuquerque, NM 87185

41. Dr. Chris Bischof, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

42. Prof. Ake Bjorck, Department of Mathematics, Linkoping University, Linkoping 58183, Sweden

43. Dr. James C. Browne, Department of Computer Sciences, University of Texas, Austin, TX 78712

44. Dr. Bill L. Buzbee, Scientific Computing Division, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307

45. Dr. Donald A. Calahan, Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109

46. Dr. Tony Chan, Department of Mathematics, University of California, Los Angeles, 405 Hilgard Avenue, Los Angeles, CA 90024

47. Dr. Jagdish Chandra, Army Research Office, P.O. Box 12211, Research Office, P.O. Box 12211, Research Triangle Park, NC 27709

48. Dr. Eleanor Chu, Department of Computer Science, University of Waterloo, Ontario, Canada N2L 3G1

49. Prof. Tom Coleman, Department of Computer Science, Cornell University, Ithaca, NY 14853

50. Dr. Paul Concus, Mathematics and Computing, Lawrence Berkeley Laboratory, Berkeley, CA 94720

51. Prof. Andy Conn, Department of Combinatorics, and Optimization, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

52. Dr. Jane K. Cullum, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598

53. Dr. George Cybenko, Computer Science Department, University of Illinois, Urbana, IL 61801

54. Dr. George J. Davis, Department of Mathematics, Georgia State University, Atlanta, GA 30303

55. Prof. John J. Doming, Department of Nuclear Engineering and Physics, Thornton Hall, McCormick Road, University of Virginia, Charlottesville, VA 22901

56. Dr. Iain Duff, CSS Division, Harwell Laboratory, Didcot, Oxon OX11 ORA, England

57. Prof. Pat Eberlein, Department of Computer Science, SUNY/Buffalo, Buffalo, NY 14260

58. Dr. Stanley Eisenstat, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520

59. Dr. Lars Elden, Department of Mathematics, Linkoping University, 58183 Linkoping, Sweden

60. Dr. Howard C. Elman, Computer Science Department, University of Maryland, College Park, MD 20742

61. Dr. Albert M. Erisman, Boeing Computer Services, 565 Andover Park West, Tukwila, WA 98188

62. Dr. Peter Fenyes, General Motors Research Laboratory, Department 15, GM Technical Center, Warren, MI 48090

63. Prof. David Fisher, Department of Mathematics, Harvey Mudd College, Claremont, CA 91711

64. Dr. Geoffrey C. Fox, Booth Computing Center 158-79, California Institute of Technology, Pasadena, CA 91125

65. Dr. Paul O. Frederickson, Computing Division, Los Alamos National Laboratory, Los Alamos, NM 87545

66. Dr. Fred N. Fritsch, L-300, Mathematics and Statistics Division, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550

67. Dr. Robert E. Funderlic, Department of Computer Science, North Carolina State University, Raleigh, NC 27650

68. Dr. Dennis B. Gannon, Computer Science Department, Indiana University, Bloomington, IN 47405

69. Dr. David M. Gay, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974

70. Dr. C. William Gear, Computer Science Department, University of Illinois, Urbana, IL 61801

71. Dr. W. Morven Gentleman, Division of Electrical Engineering, National Research Council, Building M-50, Room 344, Montreal Road, Ottawa, Ontario, Canada K1A 0R8

72. Dr. Alan George, Vice President, Academic and Provost, Needles Hall, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

73. Dr. John Gilbert, Xerox Palo Alto Research Center, 3333 Coyote Hill Road Palo Alto, CA 94304

74. Dr. Jacob D. Goldstein, The Analytic Sciences Corporation, 55 Walkers Brook Drive, Reading, MA 01867

75. Prof. Gene H. Golub, Department of Computer Science, Stanford University, Stanford, CA 94305

76. Dr. Joseph F. Grcar, Division 8331, Sandia National Laboratories, Livermore, CA 94550

77. Dr. Per Christian Hansen, Copenhagen University Observatory, @Ø@ster Voldgade 3, DK-1350 Copenhagen K, Denmark

78. Prof. Robert M. Haralick, Boeing Clairmont Egtvedt Prof., Department of Electrical Engineering, Director, Intelligent Systems Laboratory, University of Washington, 402 Electrical Engineering, Building, FT-10, Seattle, Washington 98195

79. Dr. Don E. Heller, Physics and Computer Science Department, Shell Development Co., P.O. Box 481, Houston, TX 77001

80. Dr. F. J. Helton, GA Technologies, P.O. Box 81608, San Diego, CA 92188

81. Dr. Charles J. Holland, Air Force Office of Scientific Research, Building 410, Bolling Air Force Base, Washington, DC 20332

82. Dr. Robert E. Huddleston, Computation Department, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550

83. Dr. Ilse Ipsen, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520

84. Ms. Elizabeth Jessup, Department of Computer Science, Yale University, P.O. Box 2158, Yale Station, New Haven, CT 06520

85. Prof. Barry Joe, Department of Computer Science, University of Alberta, Edmonton, Alberta, Canada T6G 2H1

86. Dr. Harry Jordan, Department of Electrical and Computer Engineering, University of Colorado, Boulder, CO 80309

87. Dr. Bo Kagstrom, Institute of Information Processing, University of Umea, 5-901 87 Umea, Sweden

88. Dr. Hans Kaper, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

89. Dr. Linda Kaufman, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974

90. Dr. Robert J. Kee, Applied Mathematics Division 8331, Sandia National Laboratories, Livermore, CA 94550

91. Dr. Kenneth Kennedy, Department of Computer Science, Rice University, P.O. Box 1892, Houston, TX 77001

92. Ms. Virginia Klema, Statistics Center, E40-131, MIT, Cambridge, MA 02139

93. Dr. Richard Lau, Office of Naval Research, 1030 E.Green Street, Pasadena, CA 91101

94. Dr. Alan J. Laub, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106

95. Dr. Robert L. Launer, Army Research Office, P.O. Box 12211, Research Triangle Park, NC 27709

96. Dr. Charles Lawson, MS 301-490, Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109

97. Prof. Peter D. Lax, Director, Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012

98. Dr. James E. Leiss, 13013 Chestnut Oak Drive, Gaithersburg, MD 20878

99. Dr. John G. Lewis, Boeing Computer Services, P.O. Box 24346, M/S 7L-21, Seattle, WA 98124-0346

100. Dr. Heather M. Liddell, Director, Center for Parallel Computing, Department of Computer Science and Statistics, Queen Mary College, University of London, Mile End Road, London E1 4NS, England

101. Dr. Joseph Liu, Department of Computer Science, York University, 4700 Keele Street, Downsview, Ontario, Canada M3J 1P3

102. Dr. Franklin Luk, Electrical Engineering Department, Cornell University, Ithaca, NY 14853

103. James G. Malone, General Motors Research Laboratories, Warren, MI 48090-9055

104. Dr. Thomas A. Manteuffel, Computing Division, Los Alamos National Laboratory, Los Alamos, NM 87545

105. Dr. Bernard McDonald, National Science Foundation, 1800 G Street, NW, Washington, DC 20550

106. Dr. James McGraw, Lawrence Livermore National Laboratory, L-306, P.O. Box 808, Livermore, CA 94550

107. Dr. Paul C. Messina, California Institute of Technology, Mail Code 158-79, Pasadena, CA 91125

108. Dr. Cleve Moler, Ardent Computers, 550 Del Ray Avenue, Sunnyvale, CA 94086

109. Prof. Neville Moray, Department of Mechanical and Industrial Engineering, University of Illinois, 1206 West Green Street, Urbana, IL 61801

110. Dr. Brent Morris, National Security Agency, Ft. George G. Meade, MD 20755

111. Dr. Dianne P. O'Leary, Computer Science Department, University of Maryland, College Park, MD 20742

112. Dr. James M. Ortega, Department of Applied Mathematics, University of Virginia, Charlottesville, VA 22903

113. Prof. Chris Paige, Department of Computer Science, McGill University, 805 Sherbrooke Street W., Montreal, Quebec, Canada H3A 2K6

114. Dr. John F. Palmer, NCUBE Corporation, 915 E. LaVieve Lane, Tempe, AZ 85284

115. Prof. Roy P. Pargas, Department of Computer Science, Clemson University, Clemson, SC 29634-1906

116. Prof. Beresford N. Parlett, Department of Mathematics, University of California, Berkeley, CA 94720

117. Prof. Merrell Patrick, Department of Computer Science, Duke University, Durham, NC 27706

118. Dr. Robert J. Plemmons, Departments of Mathematics and Computer Science, North Carolina State University, Raleigh, NC 27650

119. Dr. Alex Pothen, Department of Computer Science, Pennsylvania State University, University Park, PA 16802

120. Dr. John K. Reid, CSS Division, Building 8.9, AERE Harwell, Didcot, Oxon, England OX11 0RA

121. Dr. John R. Rice, Computer Science Department, Purdue University, West Lafayette, IN 47907

122. Dr. Garry Rodrigue, Numerical Mathematics Group, Lawrence Livermore Laboratory, Livermore, CA 94550

123. Dr. Donald J. Rose, Department of Computer Science, Duke University, Durham, NC 27706

124. Dr. Ahmed H. Sameh, Computer Science Department, University of Illinois, Urbana, IL 61801

125. Dr. Michael Saunders, Systems Optimization Laboratory, Operations Research Department, Stanford University, Stanford, CA 94305

126. Dr. Robert Schreiber, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180

127. Dr. Martin H. Schultz, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520

128. Dr. David S. Scott, Intel Scientific Computers, 15201 N.W.Greenbrier Parkway, Beaverton, OR 97006

129. Dr. Lawrence F. Shampine, Mathematics Department, Southern Methodist University, Dallas, TX 75275

130. Dr. Kermit Sigmon, Department of Mathematics, University of Florida, Gainesville, FL 32611

131. Dr. Danny C. Sorensen, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

132. Prof. G. W. Stewart, Computer Science Department, University of Maryland, College Park, MD 20742

133. Dr. Kosmo D. Tatalias, Atlantic Aerospace Electronics Corporation, 6404 Ivy Lane, Suite 300, Breenbelt, MD 20770-1406

134. Prof. Charles Van Loan, Department of Computer Science, Cornell University, Ithaca, NY 14853

135. Dr. Robert G. Voigt, ICASE, MS 132-C, NASA Langley Research Center, Hampton, VA 23665

136. Prof. Mary F. Wheeler, Mathematics Department, University of Houston, 4800 Calhoun, Houston, TX 77204-3476

137. Dr. Andrew B. White, Computing Division, Los Alamos National Laboratory, Los Alamos, NM 87545

138. Dr. Arthur Wouk, Army Research Office, P.O. Box 12211, Research Triangle Park, NC 27709

139. Dr. Margaret Wright, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974

140. Dr. A. Yeremin, Department of Numerical Mathematics of the USSR Academy of Sciences, Gorki Street 11, Moscow, 103905, USSR

141. Office of Assistant Manager for Energy Research and Development, U.S. Department of Energy, Oak Ridge Operations Office, P.O. Box 2001 Oak Ridge, TN 37831-8600

142-151. Office of Scientific & Technical Information, P.O. Box 62, Oak Ridge, TN 37831