187
12-15-81
(2)

(2)

I-400

ORNL/CSD--88

DE82 002253

UCC-ND

MASTER

**NUCLEAR
DIVISION**

**UNION
CARBIDE**

**Least Squares Adjustment of
Large-Scale Geodetic Networks
by Orthogonal Decomposition**

J. A. George
G. H. Golub
M. T. Heath
R. J. Plemmons

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

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.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A03; 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.

DISCLAIMER

This book 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/CSD-88
Distribution Category UC-32

LEAST SQUARES ADJUSTMENT OF LARGE-SCALE GEODETIC
NETWORKS BY ORTHOGONAL DECOMPOSITION*

J. A. George¹
G. H. Golub²
M. T. Heath
R. J. Plemmons³

¹University of Waterloo

²Stanford University

³North Carolina State University

Sponsor: D. A. Gardiner
Originator: M. T. Heath

Date Published - November 1981

*This report was prepared for written submission to the Proceedings of the Symposium on Geodetic Networks and Computations held in Munich, Germany, August 30 - September 4, 1981. Research sponsored by the National Geodetic Survey of the National Ocean Survey, NOAA, U.S. Department of Commerce, under Interagency Agreement No. 40-1108-80 and by the Applied Mathematical Sciences Research Program, Office of Energy Research, U.S. Department of Energy under contract No. W-7405-eng-26 with the Union Carbide Corporation.

COMPUTER SCIENCES DIVISION

at

Oak Ridge National Laboratory

Post Office Box Y

Oak Ridge, Tennessee 37830

Union Carbide Corporation, Nuclear Division
operating the
Oak Ridge Gaseous Diffusion Plant • Paducah Gaseous Diffusion Plant
Oak Ridge Y-12 Plant • Oak Ridge National Laboratory
under Contract No. W-7405-eng-26
for the
Department of Energy

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

LIST OF ILLUSTRATIONS

<u>Figure No.</u>		<u>Page</u>
2.1	One level of dissection and block structure induced in A.....	7
2.2	Two levels of dissection and block structure induced in A.....	7
2.3	Structure of the matrix A with all separators numbered last and only the structure of the leading diagonal blocks exploited.....	7
3.1	Computational flow of Algorithm 1.....	11
4.1	Pictorial description of the first step of the $ P $ -step reduction of A to upper trapezoidal form...	14
4.2	Depiction of the second step of the block-reduction of A to upper trapezoidal form.....	15
4.3	Block structure of A_i	16
FIGURE 5.1		18

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

CONTENTS

	<u>Page</u>
ILLUSTRATIONS.....	iii
ABSTRACT.....	1
1. Introduction.....	2
2. Helmert Blocking, Dissection and Partitioning.....	5
3. An Efficient Scheme Using Givens Rotations.....	8
4. Decomposition of A Using Auxiliary Storage.....	12
5. Some Extensions to Algorithm 1.....	17
5.1 Rank Deficient Problems.....	18
5.2 Updating and Constraints.....	21
6. Conclusion	23
REFERENCES	24

LEAST SQUARES ADJUSTMENT OF LARGE-SCALE GEODETIC
NETWORKS BY ORTHOGONAL DECOMPOSITION

J. A. George
G. H. Golub
M. T. Heath
R. J. Plemmons

ABSTRACT

This article reviews some recent developments in the solution of large sparse least squares problems typical of those arising in geodetic adjustment problems. The new methods are distinguished by their use of orthogonal transformations which tend to improve numerical accuracy over the conventional approach based on the use of the normal equations. The adaptation of these new schemes to allow for the use of auxiliary storage and their extension to rank deficient problems are also described.

1. Introduction

A geodetic position network is a mathematical model consisting of mesh points (representing control points or geodetic stations) having unknown positions on a reference surface. These mesh points are joined by lines representing one or more observations involving the two stations at the ends of the line. The observations may be distances between stations, or angles between two lines emanating from a station. Equations relating the unknown coordinates of the stations and the observed distance and/or angle observations lead to an overdetermined system of nonlinear equations. These measurements are then "adjusted" by solving the system in the least squares sense. The equations normally involve only a few unknowns, typically only four or five. From obvious physical considerations, the topology of the network tends to display only "local connectedness"; measurements normally involve stations which are physically adjacent.

These geodetic network adjustment problems lead (after linearization) to a large sparse overdetermined system of m linear equations in n unknowns

$$Ax \approx b, \quad (1.1)$$

where the observation matrix A has full column rank. The conventional approach to the solution of the associated least squares (adjustment) problem is via the Cholesky factorization of the system of normal equations

$$A^T Ax = A^T b. \quad (1.2)$$

Setting $B = A^T A$ and $\bar{b} = A^T b$, one computes x by factoring B as

$$B = R^T R, \quad (1.3)$$

with R upper triangular, and then solving the two triangular systems

$$R^T y = \bar{b}$$

and $Rx = y.$ (1.4)

It is well known among numerical analysts that the normal equations approach to solving a least squares problem may suffer numerical difficulties, especially when the matrix is ill conditioned or the residual is large, and over the past fifteen years the methods of choice in terms of accuracy and reliability are based on orthogonal reduction techniques [11,16]. Their essential feature is that A itself is reduced to R , avoiding the possible numerical problems, such as cancellation and ill-conditioning, associated with explicit computation of B . A sequence of orthogonal matrices $Q_1, Q_2, \dots, Q_s$ is determined which, when applied to A , reduces it to upper trapezoidal form:

$$Q_s \cdots Q_2 Q_1 A = \begin{pmatrix} R \\ 0 \end{pmatrix}, \quad (1.5)$$

with R upper triangular.

Defining $Q^T = Q_s \cdots Q_2 Q_1$, we have $A = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$, and defining

$$Q^T b = \begin{pmatrix} c \\ d \end{pmatrix}, \quad (1.6)$$

where c is an n -vector, the least squares solution x to (1.1) is obtained by solving $Rx = c$. Note that the R in (1.3) and (1.5) is the same, apart from possible differences in the signs of the rows. To see this it is only necessary to observe that

$$A^T A = (R^T \ 0) Q^T Q \begin{pmatrix} R \\ 0 \end{pmatrix} = R^T R,$$

along with the observation that the Cholesky factor of a positive definite symmetric matrix is unique.

Orthogonal decomposition methods have not been commonly used in practice for large sparse problems because of the widely held belief that they caused unacceptably high fill-in. It was also unclear how to design ordering strategies so as to minimize or significantly reduce such fill-in and how to design data structures that would efficiently accomodate the fill-in. One of the main purposes of this article is to review some recent work which addresses these issues.

An outline of the paper is as follows. In section 2 we review some of the observations made by Golub and Plemmons [12] regarding Helmert blocking (nested dissection) and the corresponding structure such blocking induces in the design matrix A . We also review their proposal for computing an orthogonal factorization of the matrix, exploiting the induced block structure. In order to exploit sparsity at a finer level, section 3 describes a scheme for solving general sparse least squares problems which was proposed and implemented by George and Heath [5]. The scheme is also based on an orthogonal

decomposition, but uses no more storage than the conventional normal equation approach. Combining this basic algorithm with the exploitation of the block structure resulting from dissection leads to an effective scheme which utilizes auxiliary storage for very large problems [7], as reviewed in section 4. Finally, section 5 discusses some extensions of the method of section 3 to rank-deficient problems, and to problems which cannot be handled efficiently using the original method proposed by George and Heath, such as those having a few dense rows or equality constraints [14].

2. Helmert Blocking, Dissection and Partitioning

The size of most adjustment problems dictates that they must be broken down into subproblems which are connected in a well defined way, and which can be solved sequentially. The method of Helmert blocking has been used for many years in geodesy, and is called "substructuring" or "dissection" in the engineering and numerical analysis literature. We briefly review the basic idea, since it is exploited in the auxiliary storage scheme of section 4.

Following Golub and Plommons [12], suppose we have a geodetic position network on a geographical region as depicted in Figure 2.1. Suppose one picks a latitude such that approximately one half of all stations lie on each side of this latitude. This forms two blocks of interior stations and one block of separator or junction stations. Now order the stations so that those in the interior region A_1 appear

first, those in A_2 appear second, and those in the junction region B appear last. Now order the equations so that those involving stations in A_1 come first, those in A_2 come second, and those involving only stations in B come last. Thus, we have induced a block upper triangular structure in the observation matrix A , as shown in Figure 2.1 (for simplicity additional structure in the rightmost block column is not shown). Of course we can recursively apply this dissection idea, choosing separator (junction) stations sets in A_1 and A_2 , and thereby inducing a partitioning having exactly the same form, but on a smaller scale, in the leading diagonal blocks in the matrix of Figure 2.1. This process is depicted in Figure 2.2. It can be repeatedly applied to obtain a block upper triangular form having more (and smaller) diagonal blocks.

Golub and Plemmons [12] propose an orthogonalization algorithm which exploits some of the block structure in the matrix A that is induced by a multi-level Helmert blocking. Specifically, they assume that all the separator blocks are numbered last, leaving 2^{ℓ} leading diagonal blocks when the dissection has ℓ levels. Their algorithm exploits this leading diagonal block structure, but does not exploit the structure in the remaining blocks. For our example in Figure 2.2, the matrix structure they exploit is shown in Figure 2.3. Other methods for producing a similar block structure have been proposed by Weil and Kettler [18] and by Plemmons and Hume [15]. With all of these methods, however, exploitation of block structure alone is

FIGURE 2.1
One level of dissection and block structure induced in A .

FIGURE 2.2
Two levels of dissection and block structure induced in A .

FIGURE 2.3
Structure of the matrix A with all separators numbered last and only the structure of the leading diagonal blocks exploited.

inadequate for handling very large problems. An algorithm which exploits sparsity at a finer level is the subject of the next section.

3. An Efficient Scheme Using Givens Rotations

In this section we review the basic ideas of a method proposed by George and Heath [5] for efficiently solving sparse linear least squares problems. Mathematically, it involves the determination of a sequence of rotation matrices which reduces A to upper trapezoidal form, as in (1.5). The reduction is done row by row, and for purposes of a clear exposition, should be viewed as beginning with an empty matrix R_0 , and then successively rotating each row of the m rows of A into R_0 , obtaining the sequence $R_1, R_2, \dots, R_m$.

The novelty of the method is its use of well developed sparse matrix techniques to find a good column ordering for A which yields a sparse R , and the utilization of a very efficient data structure for R . Note that R is independent of the row ordering of A ; this can be seen by observing that for any m by m permutation matrix P , $A^T P^T P A = A^T A = R^T R$.

With these observations, we present the basic steps of the method upon which we then elaborate.

Algorithm 1

1. Determine the structure (not the numerical values) of $B = A^T A$.
2. Find an ordering for B (column ordering for A) which has a sparse Cholesky factor R .
3. Symbolically factorize the reordered B , generating a row-oriented data structure for R .

4. Compute R and c by processing the rows of $[A, b]$ one by one using Givens rotations.
5. Solve $Rx = c$, then permute the components of x back into the original column ordering of A .

Steps 1 through 3 exploit well-developed techniques for solving sparse positive definite systems [8]. In step 1, the structure of $A^T A$ is determined, which of course need not involve any floating point operations and consequent rounding errors. Then an ordering algorithm is applied to the output of step 1, providing a symmetric reordering of $A^T A$, which corresponds to a column permutation of A . Next an algorithm is applied to B which determines the structure of its Cholesky factor R , and sets up a data structure in which to store it.

These first three steps correspond exactly to what would be done in solving (1.2) using current sparse matrix techniques. The next step would then be to place the numerical values of B in the data structure for R and compute R in place using Cholesky's method. However, to avoid the explicit computation of B , the method diverges from the normal equations approach at this point. Steps 1-3 will have provided a good column ordering for A , so that R will be sparse, along with an efficient data structure which exploits that sparsity. Step 4 involves the application of Givens rotations to the rows of A one at a time, gradually building up R . These rotations are applied simultaneously to b , and when all rows of A have been processed ("rotated into R "), step 5 is executed to obtain x .

The computational flow of the algorithm we have just described is depicted in Figure 3.1. The details of step 4 of the algorithm are of some interest and will be needed for reference later. Let a^T be a given row of A to be processed next, with the components of a^T suitably permuted to reflect the new column ordering for A . Let j be the subscript of the first nonzero component of a^T . If row j of the data structure is still vacant, as will likely be the case early in the row by row processing, then a^T may simply be placed into row j of the data structure. If, on the other hand, row j of the data structure is already occupied by previously stored numerical values, then row j may be used to annihilate the first nonzero of a^T with a Givens rotation. This process is repeated until either an unoccupied row is found in which to place the working row or all its nonzeros have been annihilated.

The advantages of the method are as follows:

- (1) Unlike the normal equations approach, the potential instability due to the explicit computation of B is avoided.
- (2) Storage requirements are essentially the same as that for the normal equations approach. Any sparsity exploitation available to the normal equations approach may also be used with this approach.
- (3) The use of row operations facilitates updating operations [14] and the use of auxiliary storage [7].
- (4) The numerical computation is performed using a fixed (static) data structure; dynamic storage allocation to accommodate fill-in (transient or otherwise) is not necessary. The importance of this property is difficult to overemphasize. It allows the use of a very efficient data structure for R and efficient numerical computation.

(5) Since the order in which the rows of A are processed in step 4 does not affect the ultimate sparsity of R , this order can be exploited, if desired, for other purposes such as reducing the total arithmetic operation count, facilitating use of auxiliary storage, or enhancing stability for problems with widely varying weights. Row orderings which improve efficiency for certain problems and column orderings have been proposed by George and Heath [5], George and Ng [9], and Plemmons and Hume [15].

The matrix B , and hence R , may be quite full if A has one or more fairly full rows. In order to avoid this severe fill-in, such rows may be skipped in the initial processing and then be incorporated into the solution later by means of the updating procedure which is described in detail in section 5.

FIGURE 3.1

Computational flow of Algorithm 1.

4. Decomposition of A Using Auxiliary Storage

In this section we review the basic ideas developed by George, Heath, and Plemmons [7] to compute an orthogonal decomposition of A using auxiliary storage. The scheme exploits the block upper triangular form that can be induced in A through Helmert blocking or dissection, as discussed in section 2. It also uses the algorithm and implementation described in section 3 as its basic core computational module.

The first task is to determine a column ordering for A that would induce the block upper triangular form, as described in section 2. In [7] a so-called "incomplete nested dissection" algorithm is applied to the symmetric graph of $A^T A$, automatically producing a partitioning and ordering of the columns of A which induce a block upper triangular form. For convenience, we assume A has been correspondingly permuted, and we denote the column partitioning by

$$P = \{Y_1, Y_2, \dots, Y_p\}$$

where Y_i corresponds to the column numbers of block i. For details about how this reordering algorithm works, the reader is referred to George, Heath, and Plemmons [7] and the references cited there.

After P is determined, the actual numerical reduction of A proceeds. For illustration, we take $|P| = 3$, with A having the form shown in Figure 4.1. The computation consists of p major steps; the i-th step results in the generation of $|Y_i|$ rows of the upper triangular factor R of A. The computation involved in the first step

is depicted in Figure 4.1, and all the other major steps are similar, generating the sequence of successively smaller matrices $A_1, A_2, A_3, \dots, A_p$. At the first step, which transforms $A = A_1$ to A_2 , the matrix $[A_{11} | \tilde{A}_1]$ is reduced to upper triangular form using Algorithm 1 described in section 3. The matrix $\tilde{A}_1$ consists of those columns of A_{12} and A_{13} which are nonnull. (Thus, $\tilde{A}_1$ is simply a "compressed" version of A_{12} and A_{13} .) The first $|Y_1|$ rows of the resulting upper triangular matrix are the first $|Y_1|$ rows of R , except that the last $|Y_1|$ columns are "compressed," bearing the same relationship to R as $\tilde{A}_1$ does to $[A_{12} | A_{13}]$. The last $|Y_1|$ rows of the upper triangular matrix are "expanded" and put on the top of the second row-block of A , yielding A_2 . The next step of the computation is depicted in Figure 4.2, and the final step has no special features.

FIGURE 4.1

Pictorial description of the first step of the $|P|$ -step reduction of A to upper trapezoidal form.

FIGURE 4.2
Depiction of the second step of the block-reduction of A to upper trapezoidal form.

Thus in the general case, after $i-1$ steps of the reduction process, the matrix A_i will have the form shown in Figure 4.3.

$$A_i = \begin{pmatrix} \tilde{A}_{i-1,i} & \tilde{A}_{i-1,i+1} & \dots & \dots & \tilde{A}_{i-1,p} \\ A_{ii} & A_{i,i+1} & & & A_{i,p} \\ \vdots & \vdots & & & \vdots \\ & & \ddots & & \\ & & & \ddots & \\ & & & & A_{pp} \end{pmatrix}$$

FIGURE 4.3
Block structure of A_i .

Note that since the rows of A_i are stored on auxiliary storage at all times, the only storage needed at the i -th stage of the computation is that required for the presumably sparse upper triangular matrix R_i , where

$$R_i = \begin{array}{|c|} \hline R_{ii} \\ \hline \tilde{R}_{ii} \\ \hline \end{array}$$

Another important practical observation is that in Algorithm 1 of section 3, and in this one, there is no restriction on the order that the rows of any of the A_i are stored so that any beneficial row ordering may be used.

For more details about the actual implementation, such as management of auxiliary files, and results of numerical experiments showing its performance, the reader should consult [7]. As an example, we report that a geodetic network involving 17469 equations and 4554 unknowns can be solved on an IBM 3033 in a total time (including all I/O time etc.) of 107 seconds, with a total storage requirement of 30,141 words. Refinements to the implementation can reduce execution time substantially.

5. Some Extensions to Algorithm 1

We now describe briefly some algorithms which extend the basic method described in Section 3 to handle rank deficient problems,

linear equality constraints, and the updating of solutions. Our objective is to provide only a summary of the important ideas; details can be found in Heath [14].

5.1 Rank Deficient Problems

If the matrix A does not have linearly independent columns then there is no longer a unique solution to the least squares problem (1.1). In this case a common practice is to compute the least squares solution having minimum Euclidean norm, which is unique. The orthogonal factorization (1.5) still exists in the rank deficient case and can be stably computed by a sequence of orthogonal transformations. The triangular factor R , however, is no longer nonsingular so that it cannot be used directly to solve the least squares problem.

If Algorithm 1 is applied to a matrix A of rank $k < n$ using exact arithmetic, the result is an n by n triangular factor of rank k which has $n - k$ zeros on its diagonal. Thus, R might be thought of as schematically depicted in Fig. 5.1(a).

FIGURE 5.1

However, recalling the details of step 4 of Algorithm 1, the working row is always placed into the data structure so that its diagonal

entry is nonzero, and diagonal entries can only grow as a result of further processing with Givens rotations. Thus, if a diagonal entry of R is zero, then that row must have been untouched by the Givens processing, and therefore all its entries must be zero, as depicted in Fig. 5.1(b). Now a matrix of the latter form can be permuted into the form

$$Q^T A P = \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} \quad (5.1)$$

by row interchanges (Fig. 5.1(c)) and column interchanges (Fig. 5.1(d)). As we shall see below, however, access to R and S in their original locations is all that is required for our purposes, so that such row and column interchanges need not actually be performed. For simplicity we use the notation of (5.1) although in practice this form need not be obtained explicitly.

As usual, the situation is more complicated in finite precision arithmetic. One might expect numerical rank deficiency in A to be revealed by the presence of $n - k$ small diagonal elements in the triangular factor, and this is usually, though not always, the case. It is no longer true, however, that a diagonal entry being small implies that the remainder of the row is also negligible. Any row whose diagonal element falls below some tolerance may be considered to have resulted from premature termination of normal processing. Processing is therefore resumed on such a row, including the corresponding right hand side component, until the row is annihilated

in the usual manner, and a row of zeros is left in its original location in the data structure.

After all such additional row processing is complete, the result is a triangular factor of the form of Fig. 5.1(b) and a corresponding estimate for the rank k . Although it is undoubtedly somewhat less robust in determining the rank than algorithms using explicit column pivoting during the factorization process [16], this approach seems to work very well in practice.

Having obtained the form (5.1) it can be shown [14] that the minimum norm least squares solution to (1.1) is given by the minimum residual vector for the least squares problem

$$\begin{bmatrix} K \\ -I \end{bmatrix} z \approx \begin{bmatrix} y \\ 0 \end{bmatrix}, \quad (5.2)$$

where the k by $n - k$ matrix K and k -vector c are the respective solutions to the triangular systems $RK = S$ and $Ry = c$, with c consisting of the first k components of $Q^T b$. Observe that these systems are easily solved using the sparse triangular factor R already computed and that the presumably small system (5.2) can be solved by any method, such as Householder transformations, suitable for small dense problems. These steps for computing the minimum-norm solution are summarized below.

Algorithm 2

1. Perform steps 1 through 4 of Algorithm 1, carrying out additional Givens processing, if necessary, to annihilate any rows having diagonal elements falling below given tolerance. The result is a permuted form of (5.1).

2. Solve $Ry = c$.

3. Solve $RK = S$.

4. Compute the orthogonal factorization $U \begin{bmatrix} K \\ -I \end{bmatrix} = \begin{bmatrix} L^T \\ 0 \end{bmatrix}$, with
 U orthogonal and L lower triangular.

$$5. \begin{bmatrix} s \\ t \end{bmatrix} = U \begin{bmatrix} y \\ 0 \end{bmatrix}.$$

$$6. x = U^T \begin{bmatrix} 0 \\ t \end{bmatrix}.$$

5.2 Updating and Constraints

If an otherwise sparse matrix A has a few rows with a relatively large number of nonzeros, the resulting fill in $A^T A$ (and hence in its Cholesky factor) would severely limit the effectiveness of Algorithm 1. It is therefore desirable to withhold such rows from the initial structure determination and Givens processing, then update the resulting solution to incorporate the effect of the omitted rows. It may also be the case, due to theoretically known relationships or to a few unusually accurate measurements, that it is desirable that some of the equations in a linear system be satisfied exactly while the remaining equations are satisfied only in the least squares sense. If such constraints are numerous and sparse, they may be handled by appropriate weighting. Any dense constraints, however, would also lead to excessive fill in the triangular factor. Thus, a useful extension of Algorithm 1 would handle constrained least squares problems of the form

$$\begin{bmatrix} A \\ E \end{bmatrix} x \approx \begin{bmatrix} b \\ f \end{bmatrix} \quad \text{subject to } Gx = h, \quad (5.3)$$

where A is m by n , E is p by n , G is q by n ; vectors b , f , and h are dimensioned accordingly; and p and q are relatively small compared to n .

In [14] the following algorithm is derived for solving (5.3).

Algorithm 3

1. Perform steps 1 through 4 of Algorithm 1 on $[A, b]$.
2. Solve $Ry = c$.
3. Solve $R^T J = E^T$ and $R^T K = G^T$.
4. Compute the orthogonal factorization $U \begin{bmatrix} J & K \\ I & 0 \end{bmatrix} = \begin{bmatrix} L^T \\ 0 \end{bmatrix}$,
with U orthogonal and L lower triangular.
5. $r = \begin{bmatrix} f - Ey \\ h - Gy \end{bmatrix}$
6. Solve $Ls = r$.
7. $\begin{bmatrix} u \\ v \end{bmatrix} = U^T \begin{bmatrix} s \\ 0 \end{bmatrix}$.
8. Solve $Rz = u$.
9. $x = y + z$.

6. Conclusion

We have summarized the recent work of the authors which provides a stable and effective alternative, based on orthogonal factorization using Givens rotations, to the conventional normal equations approach to sparse least squares problems arising in geodetic adjustment computations. The method requires no more storage than the normal equations, and is amenable to the convenient use of auxiliary storage as well as other extensions, such as handling rank deficiency, updating and constraints. Further generalizations of Algorithm 1 along the lines of section 5 have been proposed by Bjorck [2].

Other alternatives to the normal equations which have been proposed include the method of Pöters and Wilkinson [17] as adapted to the sparse case by Bjorck and Duff [3], the augmented matrix or sparse tableau method suggested for the sparse case by Hachtel [13] and numerous iterative methods. Surveys and comparisons of all of these methods are to be found in [1], [4], [6], and [10].

We hope that these new developments will be of benefit to the geodetic community, especially for poorly conditioned problems in which the numerical stability of the conventional normal equations approach is questionable.

REFERENCES

1. Å. Björck, Methods for sparse linear least squares problems, in Sparse Matrix Computations (J. R. Bunch and D. J. Rose, Eds.), Academic Press, 1976, pp. 177-199.
2. Å. Björck, A general updating algorithm for constrained linear least squares problems, Tech. Report LiTH-MAT-R-81-18, Linköping University, Sweden, July 1981.
3. Å. Björck and I. S. Duff, A direct method for solving sparse linear least squares problems, Linear Algebra Appl. 34 (1980), pp. 43-67.
4. I. S. Duff and J. K. Reid, A comparison of some methods for the solution of sparse overdetermined systems of linear equations, J. Inst. Maths. Appl. 17 (1976), pp. 267-280.
5. A. George and M. T. Heath, Solution of sparse linear least squares problems using Givens rotations, Linear Algebra Appl. 34 (1980), pp. 69-83.
6. A. George, M. T. Heath, and E. Ng, A comparison of some methods for solving sparse linear least squares problems, Tech. Report ORNL/CSD-87, Union Carbide Corporation, Oak Ridge, Tennessee, October 1981. (submitted to SIAM J. Sci. Statist. Comput.)
7. A. George, M. T. Heath, and R. J. Plemmons, Solution of large-scale sparse least squares problems using auxiliary storage, Tech. Report ORNL/CSD-63, Union Carbide Corporation, Oak Ridge, Tennessee, August 1980. (to appear in SIAM J. Sci. Statist. Comput., December 1981).
8. A. George and J. W. H. Liu, Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall, 1981.
9. A. George and E. Ng, On row and column orderings for sparse least squares problems, Tech. Report CS-81-09, University of Waterloo, Ontario, Canada, March 1981. (submitted to SIAM J. Numer. Anal.)
10. P. E. Gill and W. Murray, The orthogonal factorization of a large sparse matrix, in Sparse Matrix Computations (J. R. Bunch and D. J. Rose, Eds.) Academic Press, 1976, pp. 201-212.
11. G. H. Golub, Numerical methods for solving linear least squares problems, Numer. Math. 7 (1965), pp. 206-216.

12. G. H. Golub and R. J. Plemmons, Large-scale geodetic least-squares adjustment by dissection and orthogonal decomposition, *Linear Algebra Appl.* 34 (1980), pp. 3-27.
13. G. D. Hachtel, Extended application of the sparse tableau approach - finite elements and least squares, Tech. Report, Elec. Sci. Engr. Dept. and Comput. Sci. Dept., UCLA, 1974.
14. M. T. Heath, Some extensions of an algorithm for sparse linear least squares problems, Tech. Report ORNL/CSD-80, Union Carbide Corporation, Oak Ridge, Tennessee, May 1981. (submitted to *SIAM J. Sci. Statist. Comput.*).
15. D. Hume, Ordering sparse least squares problems, MS Thesis, University of Tennessee, Knoxville, Tennessee, August 1981.
16. C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, 1974.
17. G. Peters and J. H. Wilkinson, The least squares problem and pseudo-inverses, *Comput. J.* 13 (1970), pp. 309-316.
18. R. L. Weil and P. C. Kettler, Rearranging matrices to block-angular form for decomposition (and other) algorithms, *Management Sci.* 18 (1971), pp. 98-108.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

INTERNAL DISTRIBUTION

1. Central Research Library	16. D. A. Gardiner
2. Patent Office	17. K. E. Gipson/
Biometrics Library
3. Y-12 Technical Library
Document Reference Section	18. L. J. Gray
4. Laboratory Records Department - RC	19-23. M. T. Heath
5-7. Laboratory Records Department	24. S. A. McGuire
8. K. O. Bowman	25. J. E. Pierce
9. J. A. Carpenter	26. C. A. Serbin
10. H. P. Carter/CSD X-10 Library	27. R. C. Ward
11. R. L. Cox	28. G. W. Westley
12. D. J. Downing	29. D. G. Wilson
13. J. B. Drake	30. T. Wright
14. R. E. Funderlic	31. C. B. Yount
15. P. W. Gaffney	

EXTERNAL DISTRIBUTION

32. Prof. V. Ashkenazi, Dept. of Civil Engineering, The University of Nottingham, Nottingham, England NG7 2RD
33. Dr. Donald M. Austin, Division of Engineering, Mathematical, & Geosciences, Office of Basic Energy Sciences, Germantown Building, Department of Energy, Washington, DC 20545
34. Prof. Åke Björck, Dept. of Mathematics, Linköping University, Linköping 58183, Sweden
35. Dr. Paul Boggs, Mathematics Division, U.S. Army Research Office, Box 12211, Research Triangle Park, NC 27009
36. Prof. James Bunch, UCSD, C-012, APM Building, La Jolla, CA 92093
37. Dr. T. D. Butler, T-3, Hydrodynamics, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87545
38. Dr. Bill L. Buzbee, C-3, Applications Support & Research, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87545
39. Dr. L. Lynn Cleland, Engineering Research Division, Lawrence Livermore Laboratory, P.O. Box 808, Livermore, CA 94550

40. Dr. James S. Coleman, Division of Engineering, Mathematical and Geosciences, Office of Basic Energy Sciences, Department of Energy, ER-17, MC G-256, Germantown, Washington, DC 20545
41. Dr. James Coronas, Ames Laboratory, Iowa State University, Ames, IA 50011
42. Dr. Iain S. Duff, CSS Division, Building 8.9, AERE Harwell, Didcot, Oxon, England
43. Prof. George J. Davis, Dept. of Mathematics, Georgia State University, Atlanta, GA 30303
44. Dr. Marvin D. Erickson, Computer Technology, Systems Dept., Pacific Northwest Laboratory, P.O. Box 999, Richland, WA 99352
45. Dr. Albert M. Erisman, Boeing Computer Services, 565 Andover Park West, Tukwila, WA 98188
46. Dr. Kirby W. Fong, L-560, Lawrence Livermore Laboratory, P.O. Box 5509, Livermore, CA 94550
47. Dr. Fred N. Fritsch, L-300, Mathematics and Statistics Division, Lawrence Livermore Laboratory, P.O. Box 808, Livermore, CA 94550
48. Dr. David M. Gay, MIT-CREMS, Rm. E38-278, 292 Main Street, Cambridge, MA 02142
- 49-53. Dr. J. Alan George, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
54. Prof. Max Goldstein, Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012
- 55-59. Prof. Gene H. Golub, Department of Computer Science, Stanford University, Stanford, CA 94305
60. Dr. Richard J. Hanson, Numerical Mathematics Division 5122, Sandia Laboratories, Albuquerque, NM 87115
61. Dr. Richard L. Hooper, Statistics, Systems Department, Pacific Northwest Laboratory, P.O. Box 999, Richland, WA 99352
62. Dr. Robert E. Huddleston, Applied Mathematics Division 8332, Sandia Laboratories, Livermore, CA 94550
63. Dr. Robert J. Kee, Applied Mathematics Division 8331, Sandia Laboratories, Livermore, CA 94550
64. Prof. Peter D. Lax, Director, Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012

65. Dr. Charles Lawson, Jet Propulsion Laboratory, Pasadena, CA 91103
66. Ms. Judith A. Mahaffey, Statistics, Systems Department, Pacific Northwest Laboratory, P.O. Box 999, Richland, WA 99352
67. Dr. Thomas A. Manteuffel, Numerical Mathematics Division 5642, Sandia Laboratories, Albuquerque, NM 87185
68. Prof. Peter Meissl, Institute for Mathematical and Numerical Geodesy, Technical University of Graz, Technikerstra 4, A-8010, Graz, Austria
69. Dr. Paul C. Messina, Applied Mathematics Division, Argonne National Laboratory, Argonne, IL 60439
70. Dr. George Michael, Computation Department, Lawrence Livermore Laboratory, P.O. Box 808, Livermore, CA 94550
71. Prof. Cleve Moler, Department of Computer Science, University of New Mexico, Albuquerque, NM 87131
72. Esmond Ng, Department of Computer Science, University of Waterloo, Waterloo, Canada N2L 3G1
73. Dr. Basil Nichols, T-7, Mathematical Modeling and Analysis, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87545
74. Prof. Chris Paige, Computer Science Department, McGill University, 805 Sherbrooke Street W., Montreal, Quebec, Canada H3A 2K6
75. Dr. Ronald Peierls, Applied Mathematics Department, Brookhaven National Laboratory, Upton, NY 11973
- 76-80. Prof. Robert J. Plemmons, Departments of Mathematics and Computer Science, North Carolina State University, Raleigh, NC 27650
81. Dr. James C. T. Pool, Mathematical and Information Sciences Division, Office of Naval Research, Department of Navy, Arlington, VA 22217
82. Dr. William G. Poole, Boeing Computer Services, 565 Andover Park West, Tukwila, WA 98188
83. Mr. Allen Pope, OAC12, NOAA/NOS/NGS, 11400 Rockville Pike, Rockville, MD 20852
84. Dr. Carl Quong, Computer Science and Applied Mathematics Department, Lawrence Berkeley Laboratory, Berkeley, CA 94720
85. Dr. John K. Reid, CSS Division, Building 8.9, AERE Harwell, Didcot, Oxon, England

86. Dr. Donald J. Rose, Bell Laboratories, Murray Hill, NJ 07974
87. Dr. Milton E. Rose, Director, ICASE, M/S 132C, NASA Langley Research Center, Hampton, VA 28665
88. Dr. Bert W. Rust, Scientific Computing Division, Technology Building, National Bureau of Standards, Washington, DC 20234
89. Dr. Michael Saunders, Systems Optimization Laboratory, Operations Research Department, Stanford University, Stanford, CA 94305
90. Prof. Martin H. Schultz, Department of Computer Science, Yale University, 10 Hillhouse Avenue, P.O. Box 2158 Yale Station, New Haven, CT 06520
91. Dr. Lawrence F. Shampine, Numerical Mathematics Division 5642, Sandia Laboratories, P.O. Box 5800, Albuquerque, NM 87115
92. Dr. John N. Shoosmith, M/S 125, NASA Langley Research Center, Hampton, VA 23366
93. Prof. G. W. Stewart, Computer Science Department, University of Maryland, College Park, MD 20742
94. Prof. Charles Van Loan, Department of Computer Science, Cornell University, Ithaca, NY 14853
95. Dr. Ray A. Waller, S-1, Statistics, Los Alamos Scientific Laboratory P.O. Box 1663, Los Alamos, NM 87545
96. Dr. James H. Wilkinson, Division of Numerical Analysis and Computer Science, National Physical Laboratory, Teddington, Middlesex TW11 0LW, England
97. Dr. Ralph A. Willoughby, Mathematical Sciences Department, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598
98. Dr. Margaret Wright, Systems Optimization Laboratory, Operations Research Department, Stanford University, Stanford, CA 94305
99. Mr. Robert Ziegler, Defense Mapping Agency, DMAHTC-GSGC, 6500 Brookes Lane, Washington, DC 20315
100. Office of Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37830
- 101-
286. Given Distribution as shown in TID-4500 under Mathematics and Computers Category