MATHEMATICS OF COMPUTATION, VOLUME 28, NUMBER 126, APRIL 1974, PAGES 505–535

Methods for Modifying Matrix Factorizations

P. E. Gill, G. H. Golub, W. Murray and M. A. Saunders*

Abstract. In recent years, several algorithms have appeared for modifying the factors of a matrix following a rank-one change. These methods have always been given in the context of specific applications and this has probably inhibited their use over a wider field. In this report, several methods are described for modifying Cholesky factors. Some of these have been published previously while others appear for the first time. In addition, a new algorithm is presented for modifying the complete orthogonal factorization of a general matrix, from which the conventional QR factors are obtained as a special case. A uniform notation has been used and emphasis has been placed on illustrating the similarity between different methods.

1. Introduction. Consider the system of equations

$$Ax = b$$

where A is an $n \times n$ matrix and b is an n-vector. It is well known that x should be computed by means of some factorization of A, rather than by direct computation of A^{-1}. The same is true when A is an $m \times n$ matrix and the minimal least squares solution is required; in this case, it is usually neither advisable nor necessary to compute the pseudo-inverse of A explicitly (see Peters and Wilkinson [13]).

Once x has been computed, it is often necessary to solve a modified system

$$\bar{A}\bar{x} = \bar{b}.$$

Clearly, we should be able to modify the factorization of A to obtain factors for $\bar{A}$, from which $\bar{x}$ may be computed as before. In this paper, we consider one particular type of modification, in which $\bar{A}$ has the form

$$\bar{A} = A + \alpha yz^T$$

where α is a scalar and y and z are vectors of the appropriate dimensions. The matrix αyz^T is a matrix of rank one, and the problem is usually described as that of *updating* the factors of A following a *rank-one modification.*

There are at least three matters for consideration in computing modified factors:

(a) The modification should be performed in as few operations as possible. This is especially true for large systems when there is a need for continual updating.

(b) The numerical procedure should be stable. Many of the procedures for modifying matrix inverses or pseudo-inverses that have been recommended in the literature are numerically unstable.

(c) If the original matrix is sparse, it is desirable to preserve its sparsity as much as possible. The factors of a matrix are far more likely to be sparse than its inverse.

Received February 26, 1973.

AMS (MOS) subject classifications (1970). Primary 65F30; Secondary 15A06, 15A39, 90C05, 90C20, 90C30.

* The work of G. H. Golub and M. A. Saunders was in part supported by the AEC and the NSF.

Modification methods have been used extensively in numerical optimization, statistics and control theory. In this paper, we describe some methods that have appeared recently, and we also propose some new methods. We are concerned mainly with algebraic details and shall not consider sparsity hereafter. The reader is referred to the references marked with an asterisk for details about particular applications.

1.1. *Notation.* The elements of a matrix A and a vector x will be denoted by a_{ij} and x_j respectively. We will use A^T to denote the transpose of A, and $||x||_2$ to represent the 2-norm of x, i.e., $||x||_2 = (x^Tx)^{1/2}$. The symbols Q, R, L and D are reserved for matrices which are respectively orthogonal, upper triangular, unit lower triangular and diagonal. In particular, we will write $D = \text{diag}(d_1, d_2, \cdots, d_n)$. The jth column of the identity matrix I will be written as e_j and e will denote the vector $[1, 1, \cdots, 1]^T$.

2. Preliminary Results. Most of the methods given in this paper are based in some way upon the properties of orthogonal matrices. In the following, we discuss some important properties of these matrices with the intention of using the material in later sections.

2.1. *Givens and Householder Matrices.* The most common application of orthogonal matrices in numerical analysis is the reduction of a given n-vector z to a multiple of a column of the identity matrix, e.g., find an $n \times n$ orthogonal matrix P such that

$$Pz = \pm\rho e_1 \tag{1}$$

or

$$Pz = \pm\rho e_n. \tag{2}$$

This can be done by using either a sequence of plane rotation (Givens) matrices or a single elementary hermitian (Householder) matrix. In order to simplify the notation we will define the former as

$$\begin{bmatrix} c & s \\ s & -c \end{bmatrix} \tag{3}$$

and call this a *Givens matrix* rather than a plane rotation since it corresponds to a rotation followed by a reflection about an axis.

This matrix has the same favorable numerical properties as the usual plane rotation matrix (see Wilkinson [16, pp. 131–152]), but it is symmetric. The choice of c and s to perform the reduction

$$\begin{bmatrix} c & s \\ s & -c \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} +\rho \\ 0 \end{bmatrix}$$

is given by

$$\begin{aligned} &\rho^2 = z_1^2 + z_2^2, \\ &\rho = \text{sign}\,(z_1)(\rho^2)^{1/2} \quad \text{and} \\ &c = z_1/\rho, \qquad s = z_2/\rho. \end{aligned} \tag{4}$$

Note that $0 \leqq c \leqq 1$. In order to perform the reduction (1) or (2), we must embed the matrix (3) in the n-dimensional identity matrix. We shall use P_j^i to denote the matrix which, when applied to the vector $[z_1, z_2, \cdots, z_n]^T$, reduces z_j to zero by forming a linear combination of this element with z_i. If $i < j$, then

$$P_j^i z = \begin{array}{c} \begin{array}{ccccccccccc} & & & i & & & & j & & & \end{array} \\ \left[\begin{array}{ccccccccccc} 1 & & & & & & & & & & \\ & \ddots & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & c & & & & s & & & \\ & & & & 1 & & & & & & \\ & & & & & \ddots & & & & & \\ & & & & & & 1 & & & & \\ & & & s & & & & -c & & & \\ & & & & & & & & 1 & & \\ & & & & & & & & & \ddots & \\ & & & & & & & & & & 1 \end{array}\right] \end{array} \begin{bmatrix} z_1 \\ \vdots \\ z_i \\ \vdots \\ z_j \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} z_1 \\ \vdots \\ \bar{z}_i \\ \vdots \\ 0 \\ \vdots \\ z_n \end{bmatrix}.$$

Alternatively, if $i > j$, the (i, i)th and (j, j)th elements of P_j^i are $-c$ and $+c$, respectively. There are several sequences of Givens matrices which will perform the reduction (1) or (2); for example, if we want to reduce z to e_1, we can use

$$(5) \qquad P_2^1 P_3^2 \cdots P_{n-1}^{n-2} P_n^{n-1} z \quad \text{or} \quad P_2^1 P_3^1 \cdots P_{n-1}^1 P_n^1 z.$$

To perform the same reduction in one step, using a single Householder matrix, we have

$$P = I + \tau^{-1} u u^T,$$

where

$$(6) \qquad \begin{aligned} u &= z + \rho e_1, \\ \tau &= -\rho u_1 \quad \text{and} \\ \rho &= \operatorname{sign}(z_1) \|z\|_2. \end{aligned}$$

This time, P is such that $Pz = -\rho e_1$.

In the 2-dimensional case, we can show that the Householder matrix is of the form

$$P = \begin{bmatrix} -c & -s \\ -s & c \end{bmatrix} = -\begin{bmatrix} c & s \\ s & -c \end{bmatrix}$$

where c, s are the quantities defined earlier for the Givens matrix. Hence, when embedded in n dimensions, the 2×2 Householder and 2×2 Givens transformations are analytically the same, apart from a change of sign. (Although these matrices are $n \times n$, we shall often refer to them as "2×2" orthogonal matrices.)

There are several applications where 2-dimensional transformations are used. The amount of computation needed to multiply a $2 \times n$ matrix A by a 2×2 Householder matrix computed using Eqs. (6) is $4n + O(1)$ multiplications and $3n + O(1)$ additions. If this computation is arranged as suggested by Martin, Peters and Wilkinson [11] and the relevant matrix is written as

$$I + \begin{bmatrix} -u_1/\rho \\ -u_2/\rho \end{bmatrix} [1 \;\; u_2/u_1],$$

then the multiplication can be performed in $3n + O(1)$ multiplications and $3n + O(1)$ additions. Straightforward multiplication of A by a Givens matrix requires $4n + O(1)$ multiplications and $2n + O(1)$ additions. Again, the work can be reduced to $3n + O(1)$ multiplications and $3n + O(1)$ additions, as follows.

Let the Givens matrix be defined as in (4). Define the quantity

$$\mu = z_2/(z_1 + \rho), \qquad |\mu| \leqq 1.$$

Since $s = z_2/\rho$, we can write s as $s = \mu(c + 1)$. Similarly, we have $c = 1 - \mu s$. A typical product can be written in the form

$$\begin{bmatrix} c & s \\ s & -c \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} c & s \\ \mu(c+1) & \mu s - 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{7}$$
$$= \begin{bmatrix} y_1 c + y_2 s \\ y_1 \mu(c+1) + y_2(\mu s - 1) \end{bmatrix}$$

which will be denoted by

$$= \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix}.$$

Consequently, in order to perform the multiplication (7), we form

$$\bar{y}_1 = c y_1 + s y_2 \quad \text{and} \quad \bar{y}_2 = \mu(y_1 + \bar{y}_1) - y_2.$$

Note that this scheme is preferable only if the time taken to compute a multiplication is more than the time taken to compute an addition. Also, it may be advisable with both algorithms to modify the computation of ρ to avoid underflow difficulties.

In the following work, we will consider only 2×2 Givens matrices, although the results apply equally well to 2×2 Householder matrices since, as noted earlier, the two are essentially the same.

2.2. *Products of Givens Matrices.* The following results will help define some new notation and present properties of certain products of orthogonal matrices.

LEMMA I. *Let $P_{j+1}{}^{j}$ be a Givens matrix defined as in* (4). *Then the product*

$$P_n^{n-1} P_{n-1}^{n-2} \cdots P_2^1$$

is of the form

$$H_L(p, \beta, \gamma) = \begin{bmatrix} p_1\beta_1 & \gamma_1 & & & & \\ p_2\beta_1 & p_2\beta_2 & \gamma_2 & & & \\ p_3\beta_1 & p_3\beta_2 & p_3\beta_3 & \ddots & & \\ \vdots & \vdots & \vdots & & \gamma_{n-2} & \\ p_{n-1}\beta_1 & p_{n-1}\beta_2 & p_{n-1}\beta_3 & \cdots & p_{n-1}\beta_{n-1} & \gamma_{n-1} \\ p_n\beta_1 & p_n\beta_2 & p_n\beta_3 & \cdots & p_n\beta_{n-1} & p_n\beta_n \end{bmatrix}$$

where the quantities p_i, β_i and γ_i are defined by either of the following recurrence relations:

Forward Recurrence.

1. *Set* $p_1 = c_1/\pi$, $\beta_1 = \pi$, $\eta_1 = s_1/\pi$, $\gamma_1 = s_1$, *where* π *is an arbitrary nonzero scalar.*
2. *For* $j = 2, 3, \cdots, n-1$, *set* $p_j = c_j\eta_{j-1}$, $\gamma_j = s_j$, $\beta_j = -c_{j-1}/\eta_{j-1}$, $\eta_j = s_j\eta_{j-1}$.
3. *Set* $p_n = \eta_{n-1}$, $\beta_n = -c_{n-1}/p_n$.

Backward Recurrence.

1. *Set* $p_n = \pi$, $\beta_n = -c_{n-1}/\pi$, $\eta_{n-1} = s_{n-1}/\pi$, $\gamma_{n-1} = s_{n-1}$, *where* π *is an arbitrary nonzero scalar.*
2. *For* $j = n-1, n-2, \cdots, 3, 2$, *set* $p_j = c_j/\eta_j$, $\gamma_{j-1} = s_{j-1}$, $\beta_j = -c_{j-1}\eta_j$, $\eta_{j-1} = s_{j-1}\eta_j$.
3. *Set* $p_1 = c_1/\beta_1$, $\beta_1 = \eta_1$.

Proof. We will prove the lemma in the forward recurrence case; the remaining case can be proved in a similar way. Assume that the product $P_{k+1}{}^k P_k{}^{k-1} \cdots P_4{}^3 P_3{}^2 P_2{}^1$ ($k < n-1$) is given by

$$\begin{bmatrix} p_1\beta_1 & \gamma_1 & & & & & & & \\ p_2\beta_1 & p_2\beta_2 & & \ddots & & & & & \\ \vdots & \vdots & & & & & & & \\ p_k\beta_1 & p_k\beta_2 & \cdots & p_k\beta_k & \gamma_k & & & & \\ \eta_k\beta_1 & \eta_k\beta_2 & \cdots & \eta_k\beta_k & -c_k & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix} \tag{8}$$

This is true for $k = 1$ by definition. The next product $P_{k+2}{}^{k+1} P_{k+1}{}^k P_k{}^{k-1} \cdots P_3{}^2 P_2{}^1$ is given by

$$\begin{pmatrix} p_1\beta_1 & \gamma_1 & & & & & & \\ p_2\beta_1 & p_2\beta_2 & \ddots & & & & & \\ \vdots & \vdots & \ddots & & & & & \\ p_k\beta_1 & p_k\beta_2 & \cdots & p_k\beta_k & \gamma_k & & & \\ c_{k+1}\eta_k\beta_1 & c_{k+1}\eta_k\beta_2 & \cdots & c_{k+1}\eta_k\beta_k & -c_{k+1}c_k & s_{k+1} & & \\ s_{k+1}\eta_k\beta_1 & s_{k+1}\eta_k\beta_2 & \cdots & s_{k+1}\eta_k\beta_k & -s_{k+1}c_k & -c_{k+1} & & \\ & & & & & & 1 & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{pmatrix}.$$

If we define $p_{k+1} = c_{k+1}\eta_k$, $\gamma_{k+1} = s_{k+1}$, $\beta_{k+1} = -c_k/\eta_k$, $\eta_{k+1} = s_{k+1}\eta_k$, then the product $P_{k+2}{}^{k+1} \cdots P_2{}^1$ is of a similar form to (8). Continuing in this way, and finally setting $p_n = \eta_{n-1}$ and $\beta_n = -c_{n-1}/p_n$ gives the required result.

For later convenience, we shall use the notation

$$(H_L(p, \beta, \gamma))^T = H_U(\beta, p, \gamma).$$

The matrices $H_U(\beta, p, \gamma)$ and $H_L(p, \beta, \gamma)$ are defined as *special* upper- and lower-Hessenberg matrices respectively. In the same way, we define a *special* upper-triangular matrix $R(\beta, p, \gamma)$ as having the form

$$R(\beta, p, \gamma) = \begin{pmatrix} \gamma_1 & \beta_1 p_2 & \beta_1 p_3 & \cdots & \beta_1 p_n \\ & \gamma_2 & \beta_2 p_3 & \cdots & \beta_2 p_n \\ & & \gamma_3 & \cdots & \beta_3 p_n \\ & & & \ddots & \vdots \\ & & & \gamma_{n-1} & \beta_{n-1} p_n \\ & & & & \gamma_n \end{pmatrix}.$$

The particular recurrence relation used to form $H_L(p, \beta, \gamma)$ will depend upon the order in which the Givens matrices are generated. For example, if $P_n{}^{n-1}$ is formed first, then the backward recurrence relation can be used.

We have only considered a particular sequence of Givens matrices. Similar formulae can be derived to compute the lower-Hessenberg matrix associated with the sequence

$$P_{n-1}^n P_{n-2}^{n-1} \cdots P_2^3 P_1^2.$$

We state the following lemma without proof.

LEMMA II. *Let* $D = \operatorname{diag}(d_1, d_2, \cdots, d_n)$, $\Gamma_1 = \operatorname{diag}(\gamma_1, \gamma_2, \cdots, \gamma_{n-1}, 1)$, $\Gamma_2 = \operatorname{diag}(1, \gamma_1, \gamma_2, \cdots, \gamma_{n-1})$ *and* $e = (1, 1, \cdots, 1, 1)^T$. *Then*

1. $DH_L(p, \beta, \gamma) = H_L(\bar{p}, \bar{\beta}, \bar{\gamma})D$ *where* $\bar{\beta}_i = \beta_i/d_i$, $\bar{p}_i = d_i p_i$, $i = 1, 2, \cdots, n$, $\bar{\gamma}_i = d_i\gamma_i/d_{i+1}$, $i = 1, 2, \cdots, n-1$.
2. $R(\beta, p, \gamma)D = DR(\bar{\beta}, \bar{p}, \gamma)$ *where* $\bar{\beta}_i = \beta_i/d_i$, $\bar{p}_i = d_i p_i$, $i = 1, 2, \cdots, n$.

3. $R(\beta, p, \gamma) = DR(\bar{\beta}, p, e)$ *where* $\bar{\beta}_i = \beta_i/\gamma_i$, $i = 1, 2, \cdots, n-1$, $d_i = \gamma_i$, $i = 1, 2, \cdots, n$.

4. $H_L(p, \beta, \gamma) = \Gamma_1 H_L(\bar{p}, \beta, e) = H_L(p, \bar{\beta}, e)\Gamma_2$ *where* $\bar{p}_i = p_i/\gamma_i$ $(i < n)$, $\bar{p}_n = p_n$, *and* $\bar{\beta}_i = \beta_i/\gamma_i$ $(i > 1)$, $\bar{\beta}_1 = \beta_1$.

5. *If* $H_L(\bar{p}, \bar{\beta}, \gamma) = H_L(p, \beta, \gamma)$ *then* $\bar{\gamma}_i = \gamma_i$ *and* $\bar{p}_i = \alpha p_i$, $\bar{\beta}_i = \beta_i/\alpha$ *for* $i = 1, 2, \cdots, n$, *where* α *is some constant.*

The next three lemmas show how the product of special matrices with various general matrices may be computed efficiently.

LEMMA III. *Let B be an $m \times n$ matrix and $H_L(p, \beta, \gamma)$ an $n \times n$ special lower-Hessenberg matrix. The product $\bar{B} = BH$ can be formed using either of the following recurrence relations*:

Forward Recurrence.

1. $w^{(1)} = Bp$, $\bar{b}_{i1} = \beta_1 w_i^{(1)}$, $i = 1, 2, \cdots, m$;

2. $$\left.\begin{aligned} w_i^{(j)} &= w_i^{(j-1)} - p_{j-1} b_{i,j-1} \\ \bar{b}_{ij} &= \gamma_{j-1} b_{i,j-1} + \beta_j w_i^{(j)} \end{aligned}\right\}, \qquad \begin{aligned} i &= 1, 2, \cdots, m, \\ j &= 2, 3, \cdots, n. \end{aligned}$$

Backward Recurrence.

1. $w_i^{(n)} = p_n b_{in}$, $i = 1, 2, \cdots, m$;

2. $$\left.\begin{aligned} \bar{b}_{ij} &= \gamma_{j-1} b_{i,j-1} + \beta_j w_i^{(j)} \\ w_i^{(j-1)} &= p_{j-1} b_{i,j-1} + w_i^{(j)} \end{aligned}\right\}, \qquad \begin{aligned} i &= 1, 2, \cdots, m, \\ j &= n, n-1, \cdots, 2; \end{aligned}$$

3. $$\bar{b}_{i1} = \beta_1 w_i^{(1)}, \qquad i = 1, 2, \cdots, m.$$

Proof. We will give a proof for the forward recurrence case. The backward recurrence case can be shown in a similar way. The first column of $\bar{B}$ is given by

$$\bar{b}_{i1} = \beta_1 \sum_{j=1}^{n} b_{ij} p_j, \qquad i = 1, 2, \cdots, m.$$

If we define

$$w^{(1)} = Bp,$$

or

$$w_i^{(1)} = \sum_{j=1}^{n} b_{ij} p_j, \qquad i = 1, 2, \cdots, m, \tag{9}$$

then

$$\bar{b}_{i1} = \beta_1 w_i^{(1)}, \qquad i = 1, 2, \cdots, m.$$

Forming the second column, we have

$$\bar{b}_{i2} = \gamma_1 b_{i2} + \beta_1 \sum_{j=2}^{n} b_{ij} p_j, \qquad i = 1, 2, \cdots, m. \tag{10}$$

From Eq. (9), we have

$$w_i^{(1)} - b_{i1} p_1 = \sum_{j=2}^{n} b_{ij} p_j, \qquad i = 1, 2, \cdots, m,$$

and, if this vector is defined as $w^{(2)}$, then (10) becomes

$$\bar{b}_{i2} = \gamma_1 b_{i1} + \beta_2 w_i^{(2)}, \qquad i = 1, 2, \cdots, m.$$

The other columns of $\bar{B}$ are formed in exactly the same way.

The backward recurrence is more efficient, unless the product Bp is known *a priori.* It is also more convenient if $\bar{B}$ occupies the same storage as B.

The forward and backward recurrence relations require approximately 75% of the work necessary to form the same product by successively multiplying B by each of the individual Givens matrices. Since $H_L(p, \beta, \gamma)$ is an orthogonal matrix, there exists a vector v such that $H_L(p, \beta, \gamma)v = \alpha e_1$, and we can regard $H_L(p, \beta, \gamma)$ as the matrix which reduces v to αe_1. An equivalent reduction can be obtained by multiplying v by a single Householder matrix. If we have a product of the form

$$H_L(p^{(1)}, \beta^{(1)}, \gamma^{(1)}) \cdots H_L(p^{(r)}, \beta^{(r)}, \gamma^{(r)})B,$$

the computational effort involved in applying Lemma III is less than that using a similar product of the equivalent Householder matrices. This is because for D, a certain diagonal matrix, the product can be written as

$$DH_L(\bar{p}^{(1)}, \bar{\beta}^{(1)}, e) \cdots H_L(\bar{p}^{(r)}, \bar{\beta}^{(r)}, e)B$$

using Lemma II, parts 1 and 4.

LEMMA IV. *Let R be an upper-triangular matrix and $H_U(\beta, p, \gamma)$ a special upper-Hessenberg matrix. The product $\bar{H} = H_U(\beta, p, \gamma)R$ is an upper-Hessenberg matrix which can be determined using either of the following recurrence relations:*

Forward Recurrence.

1. *Set* $w^{(1)} = R^T p$,

$$\bar{h}_{1j} = \beta_1 w_j^{(1)}, \qquad j = 1, 2, \cdots, n.$$

2. *For* $i = 2, 3, \cdots, n$, *set*

$$\begin{aligned} \bar{h}_{i,i-1} &= \gamma_{i-1} r_{i-1,i-1}, \\ \left.\begin{aligned} w_j^{(i)} &= w_j^{(i-1)} - p_{i-1} r_{i-1,j} \\ \bar{h}_{i,j} &= \gamma_{i-1} r_{i-1,j} + \beta_i w_j^{(i)} \end{aligned}\right\}, &\qquad j = i, i+1, \cdots, n. \end{aligned}$$

Backward Recurrence.

1. $w_n{}^{(n)} = p_n r_{nn}$.
2. *For* $i = n, n-1, \cdots, 3, 2$, *set*

$$\bar{h}_{i,i-1} = \gamma_{i-1} r_{i-1,i-1}, \qquad w_{i-1}^{(i-1)} = p_{i-1} r_{i-1,i-1},$$

$$\left.\begin{aligned} \bar{h}_{ij} &= \gamma_{i-1} r_{i-1,j} + \beta_i w_j^{(i)} \\ w_j^{(i-1)} &= p_{i-1} r_{i-1,j} + w_j^{(i)} \end{aligned}\right\}, \qquad j = i, i+1, \cdots, n.$$

3. $$\bar{h}_{1j} = \beta_1 w_j{}^{(1)}, \qquad j = 1, 2, \cdots, n.$$

Proof. This lemma is proved in a similar way to Lemma III.

LEMMA V. *Let R be upper-triangular and $R(\beta, p, \gamma)$ a special upper-triangular matrix. The product $\bar{R} = R(\beta, p, \gamma)R$ can be found using either of the following recur-*

rence relations:

Forward Recurrence.

1. *Set* $w^{(1)} = R^T p$.
2. *For* $i = 1, 2, \cdots, n$, *set*

$$\bar{r}_{ii} = \gamma_i r_{ii},$$

$$\left.\begin{aligned} w_j^{(i+1)} &= w_j^{(i)} - p_i r_{ij} \\ \bar{r}_{ij} &= \gamma_i r_{ij} + \beta_i w_j^{(i+1)} \end{aligned}\right\}, \qquad j = i+1, i+2, \cdots, n.$$

Backward Recurrence.

1. *For* $i = n, n-1, \cdots, 1$, *set*

$$w_i^{(i)} = p_i r_{ii}, \qquad \bar{r}_{ii} = \gamma_i r_{ii},$$

$$\left.\begin{aligned} \bar{r}_{ij} &= \gamma_i r_{ij} + \beta_i w_j^{(i+1)} \\ w_j^{(i)} &= w_j^{(i+1)} + p_i r_{ij} \end{aligned}\right\}, \qquad j = i+1, i+2, \cdots, n.$$

The forward recurrence relation can be formulated in the following alternative manner:

1. Set $w^{(1)} = R^T p$.
2. For $i = 1, 2, \cdots, n$, set

$$\bar{r}_{ii} = \gamma_i r_{ii},$$

$$\left.\begin{aligned} w_j^{(i+1)} &= w_j^{(i)} - p_i r_{ij} \\ \bar{r}_{ij} &= (\gamma_i - \beta_i p_i) r_{ij} + \beta_i w_j^{(i)} \end{aligned}\right\}, \qquad j = i+1, \cdots, n.$$

This formulation requires an additional $n^2/2$ multiplications. It has been shown by Gentleman [4] that the use of the more efficient relationship can lead to numerical instabilities in certain applications.

If the products of n 2×2 Givens matrices are accumulated into a single *special* matrix, it has been demonstrated in Lemmas I–V how certain savings can be made in subsequent computations. The nature of the forward and backward recurrence relations are such that, when a value of s_j is very small, underflow could occur in the subsequent computation of η_j. This will result in a division by zero during the computation of the next β_j. It will be shown in the following section how this difficulty can be avoided by judicious choice of the scalar π.

In certain applications, the vector v which is such that

$$H_U(\beta, p, \gamma) v = \|v\|_2 \, e_1$$

is known. Since $H_U(\beta, p, \gamma)$ is orthogonal, we have, from its definition, that

$$H_U(\beta, p, \gamma)^T H_U(\beta, p, \gamma) v = \|v\|_2 \, H_U(\beta, p, \gamma)^T e_1 = \|v\|_2 \, H_L(p, \beta, \gamma) e_1$$

which gives $v = \beta_1 \|v\|_2 \, p$, and the vector v is parallel to the vector p. If the value of π is chosen as $\pi = c_1/v_1$, then the vector p is *equal* to v. If ρ_j denotes the quantity defined at (4), this gives the modified algorithm:

Backward Recurrence.

1. Set $\beta_n = -c_{n-1}/v_n$, $\gamma_{n-1} = s_{n-1}$.
2. For $j = n-1, \cdots, 3, 2$, set $\beta_j = -c_{j-1}/\rho_j$, $\gamma_{j-1} = s_{j-1}$.
3. Set $\beta_1 = c_1/v_1 = 1/\rho_1$.

In the cases where v_i is not known *a priori*, π can be set at 2^{-t}, where the computation is carried out on a machine with a t-digit binary mantissa. Since the value of η_i is such that

$$\eta_i = s_i s_{i-1} \cdots s_1/\pi$$

during forward recurrence, and

$$\eta_i = s_i s_{i+1} \cdots s_{n-1}/\pi$$

during backward recurrence, this choice of π is such that η_i is unlikely to underflow.

If even this strategy is insufficient, the product of the Givens matrices can be split into subproducts. For example, if at the kth product, η_k is intolerably small, we can form the subproduct:

$$\left[\begin{array}{c|c} I & 0 \\ \hline 0 & H_L(p', \beta', \gamma') \end{array}\right] P_{k+1}^k \left[\begin{array}{c|c} H_L(p'', \beta'', \gamma'') & 0 \\ \hline 0 & I \end{array}\right]$$

where $H_L(p', \beta', \gamma')$ and $H_L(p'', \beta'', \gamma'')$ are smaller special matrices of dimension $(n - k) \times (n - k)$ and $k \times k$, respectively. Clearly, a product of separate Givens matrices can be viewed as being a product of special matrices in which a "split" has occurred at every step. Each element in the subproduct is an individual Givens matrix.

3. Modification of the Cholesky Factor. In this section, we consider the case where a symmetric positive definite matrix A is modified by a symmetric matrix of rank one, i.e., we have

$$\bar{A} = A + \alpha z z^T.$$

Assuming that the Cholesky factors of A are known, viz. $A = LDL^T$, we wish to determine the factors

$$\bar{A} = \bar{L}\bar{D}\bar{L}^T.$$

It is necessary to make the assumption that A and $\bar{A}$ are positive definite since otherwise the algorithms for determining the modified factors are numerically unstable, even if the factorization of $\bar{A}$ exists. Several alternative algorithms will be presented and comments made upon their relative merits. Any of these general methods can be applied when A is of the form $A = B^T B$ and rows or columns of the matrix B are being added or deleted. In this case, it may be better to use specialized methods which modify the orthogonal factorization of B:

$$QB = \left[\begin{array}{c} R \\ \hline 0 \end{array}\right].$$

The reader is referred to Section 5 for further details. The methods in this section are all based upon the fundamental equality

$$\bar{A} = A + \alpha z z^T = L(D + \alpha p p^T)L^T,$$

where $Lp = z$, and p is obtained from z by a forward substitution. If we form the factorization

$$D + \alpha pp^T = \tilde{L}\tilde{D}\tilde{L}^T, \tag{11}$$

the required modified Cholesky factors are of the form

$$\bar{A} = L\tilde{L}\tilde{D}\tilde{L}^T L^T$$

giving

$$\bar{L} = L\tilde{L} \quad \text{and} \quad \bar{D} = \tilde{D},$$

since the product of two lower-triangular matrices is a lower-triangular matrix. The manner in which the factorization (11) is performed will characterize a particular method.

Method C1. *Using Classical Cholesky Factorization.* The Cholesky factorization of $D + \alpha pp^T$ can be formed directly. We will use this method to prove inductively that $\tilde{L}$ is special.

Assume at the jth stage of the computation that

$$\tilde{l}_{rs} = p_r\beta_s, \qquad r = j, j+1, \cdots, n; s = 1, 2, \cdots, j-1, \tag{12}$$

and that all these elements have been determined. Explicitly forming the jth column of $\tilde{L}\tilde{D}\tilde{L}^T$ gives the following equations for $\tilde{d}_j$ and $\tilde{l}_{rj}$, $r = j+1, \cdots, n$:

$$\sum_{i=1}^{j-1} \tilde{d}_i \tilde{l}_{ji}^2 + \tilde{d}_j = d_j + \alpha p_j^2 \tag{13}$$

and

$$\sum_{i=1}^{j-1} \tilde{d}_i \tilde{l}_{ji} \tilde{l}_{ri} + \tilde{d}_j \tilde{l}_{rj} = \alpha p_j p_r, \qquad r = j+1, \cdots, n. \tag{14}$$

Using Eq. (12) with (13) and (14) gives

$$p_j^2 \sum_{i=1}^{j-1} \beta_i^2 \tilde{d}_i + \tilde{d}_j = d_j + \alpha p_j^2$$

and

$$p_j p_r \sum_{i=1}^{j-1} \tilde{d}_i \beta_i^2 + \tilde{d}_j \tilde{l}_{rj} = \alpha p_j p_r, \qquad r = j+1, \cdots, n.$$

From the last equation, we have

$$\tilde{l}_{rj} = \frac{p_j}{\tilde{d}_j}\left[\alpha - \sum_{i=1}^{j-1} \tilde{d}_i \beta_i^2\right] p_r, \qquad r = j+1, \cdots, n,$$

and defining

$$\beta_j = \frac{p_j}{\tilde{d}_j}\left[\alpha - \sum_{i=1}^{j-1} \tilde{d}_i \beta_i^2\right]$$

gives $\tilde{l}_{rj} = p_r\beta_j$. Hence, the subdiagonal elements of the jth column of $\tilde{L}$ are multiples of the corresponding elements of the vector p.

Now forming the first column of $\tilde{L}\tilde{D}\tilde{L}^T$, we obtain the equations

$$\tilde{d}_1 = d_1 + \alpha p_1^2,$$

$$\tilde{d}_1 \tilde{l}_{r1} = \alpha p_1 p_r, \qquad r = 2, \cdots, n,$$

which shows that the subdiagonal elements of the first column of $\tilde{L}$ are multiples of the corresponding elements of p. Consequently, we have proved by induction that $\tilde{L}$ is special.

This result implies that we need only compute the values of $\tilde{d}_j$, β_j, $j = 1, \cdots, n$, in order to obtain the factorization of $D + \alpha pp^T$. In practice, we define the auxiliary quantity

$$\alpha_j = \alpha - \sum_{i=1}^{j-1} \tilde{d}_i \beta_i^2.$$

The recurrence relations for α_j, $\tilde{d}_j$ and β_j then become

$$\alpha_1 = \alpha$$

$$\left.\begin{aligned} \tilde{d}_j &= d_j + \alpha_j p_j^2 \\ \beta_j &= \alpha_j p_j / \tilde{d}_j \\ \alpha_{j+1} &= \alpha_j d_j / \tilde{d}_j \end{aligned}\right\}, \qquad j = 1, 2, \cdots, n.$$

The product $\bar{L} = L\tilde{L}$ can be computed in terms of the β_j by forward recurrence using Lemma V. Note that L and $\tilde{L}$ are both *unit* lower-triangular matrices and that this results in some simplification of the algorithm. The vector $w^{(1)}$ needed to initialize the recurrence relations is known since $w^{(1)} = Lp = z$. Also, each of the vectors $w^{(j)}$ ($j = 1, 2, \cdots, n$) can be obtained during the jth stage of the initial back substitution $Lp = z$, since

$$w_r^{(j)} = \sum_{i=j}^{n} l_{ri} p_i = z_r - \sum_{i=1}^{j-1} l_{ri} p_i, \qquad r = j, j+1, \cdots, n.$$

The final recurrence relations for modifying L and D are as follows:

Algorithm C1.

1. Define $\alpha_1 = \alpha$, $w^{(1)} = z$.
2. For $j = 1, 2, \cdots, n$, compute

$$p_j = w_j^{(j)},$$

$$\bar{d}_j = d_j + \alpha_j p_j^2,$$

$$\beta_j = p_j \alpha_j / \bar{d}_j,$$

$$\alpha_{j+1} = d_j \alpha_j / \bar{d}_j,$$

$$\left.\begin{aligned} w_r^{(j+1)} &= w_r^{(j)} - p_j l_{rj} \\ \bar{l}_{rj} &= l_{rj} + \beta_j w_r^{(j+1)} \end{aligned}\right\}, \qquad r = j+1, \cdots, n.$$

Using the expression for $w_r^{(j+1)}$, we can rearrange the equation for $\bar{l}_{rj}$ in the form

$$\begin{aligned} \bar{l}_{rj} &= l_{rj} + \beta_j (w_r^{(j)} - p_j l_{rj}) \\ &= (1 - \beta_j p_j) l_{rj} + \beta_j w_r^{(j)} \\ &= (d_j / \bar{d}_j) l_{rj} + \beta_j w_r^{(j)}, \end{aligned}$$

which is the form of the algorithm given by Gill and Murray [5]. However, this increases the number of multiplications by 50%.

One of the earliest papers devoted to modifying matrix factorizations is that by Bennett [2], in which LDU factors are updated following a rank m modification:

$$\bar{L}\bar{D}\bar{U} = LDU + XCY^T,$$

where X, Y are $n \times m$ and C is $m \times m$. It should be noted that

(i) the algorithm given by Bennett is numerically stable only when $L = U^T$, $X = Y$ and both D and $\bar{D}$ are positive definite, and

(ii) Algorithm C1 is almost identical to the special case of Bennett's algorithm when $m = 1$, $C = \alpha$ and $X = Y = z$.

The number of operations necessary to compute the modified factorization using Algorithm C1 is $n^2 + O(n)$ multiplications and $n^2 + O(n)$ additions.

If the matrix A is sufficiently positive definite, that is, its smallest eigenvalue is sufficiently large relative to some norm of $\bar{A}$, then Algorithm C1 is numerically stable. However, if $\alpha < 0$ and $\bar{A}$ is near to singularity, it is possible that rounding error could cause the diagonal elements $\bar{d}_i$ to become zero or arbitrarily small. In such cases, it is also possible that the $\bar{d}_i$ could change sign, even when the modification may be known from theoretical analysis to give a positive definite factorization. It may then be advantageous to use one of the following methods, because with these the resulting matrix will be positive definite *regardless of any numerical errors made.*

Method C2. *Using Householder Matrices.* In this method, the factorization (11) is performed using Householder matrices. To do this, we must write

$$\bar{A} = LD^{1/2}(I + \alpha vv^T)D^{1/2}L^T,$$

where v is the solution of the equations $LD^{1/2}v = z$. The matrix $I + \alpha vv^T$ can be factorized into the form

$$I + \alpha vv^T = (I + \sigma vv^T)(I + \sigma vv^T) \tag{15}$$

by choosing $\sigma = \alpha/(1 + (1 + \alpha v^T v)^{1/2})$.

The expression under the root sign is a positive multiple of the determinant of $\bar{A}$. If $\bar{A}$ is positive definite σ will be real.

We now perform the Householder reduction of $I + \sigma vv^T$ to lower-triangular form

$$\hat{L} = (I + \sigma vv^T)P_1P_2 \cdots P_{n-1}.$$

We will only consider application of the first Householder matrix P_1. The effect of the remainder can easily be deduced.

Let $P_1 = I + uu^T/\tau$ and partition v in the form $v^T = [v_1 \vdots w^T]$. The (1, 1) element of $I + \sigma vv^T$ is then $\theta = 1 + \sigma v_1^2$ and P_1 must reduce the vector $[\theta \vdots \sigma v_1 w^T]$ to a multiple of e_1^T. Using the relations of Section 2, we define

$$\rho^2 = \theta^2 + \sigma^2 v_1^2 w^T w,$$

$$u_1 = \theta + \rho \quad \text{and}$$

$$\tau = -\rho u_1.$$

(Note that we have taken $\rho = +(\rho^2)^{1/2}$, because we know that $\theta > 0$.) Now u has the form

$$u^T = [u_1 \ \vdots \ \sigma v_1 w^T],$$

i.e., the vector of elements $u_2, \cdots, u_n$ is a multiple of the vector w.

The result of applying the first Householder transformation can therefore be written as

$$(I + \sigma v v^T)\left(I + \frac{1}{\tau} u u^T\right) = \left[\begin{array}{c|c} -\rho & 0 \\ \hline \delta w & I + \bar{\sigma} w w^T \end{array}\right]$$

for suitable values of the scalars δ and $\bar{\sigma}$ which will be determined as follows. The first column is given by

$$\left[\begin{array}{c} -\gamma \\ \hline \delta w \end{array}\right] = (I + \sigma v v^T)\left[e_1 + \frac{1}{\tau} u_1 u\right]$$

$$= \left[\begin{array}{c|c} 1 + \sigma v_1^2 & \sigma v_1 w^T \\ \hline \sigma v_1 w & I + \sigma w w^T \end{array}\right]\left[\begin{array}{c} 1 + \frac{1}{\tau} u_1^2 \\ \hline \frac{1}{\tau} u_1 \sigma v_1 w \end{array}\right]$$

which implies that

$$\delta w = \left(1 + \frac{1}{\tau} u_1^2\right)\sigma v_1 w + \frac{1}{\tau} u_1 \sigma v_1 (1 + \sigma w^T w) w$$

so

$$\delta = \left(1 + \frac{1}{\tau} u_1^2\right)\sigma v_1 + \frac{1}{\tau} u_1 \sigma v_1 (1 + \sigma w^T w).$$

A small amount of algebraic manipulation gives

$$\delta = -\sigma \frac{v_1}{\rho}(2 + \sigma v^T v).$$

Similarly, for the scalar $\bar{\sigma}$, we have

$$I + \bar{\sigma} w w^T = [\sigma v_1 w \mid I + \sigma w w^T]\left[\begin{array}{c} \frac{1}{\tau} u_1 \sigma v_1 w^T \\ \hline I + \frac{1}{\tau} \sigma^2 v_1^2 w w^T \end{array}\right]$$

giving

$$\bar{\sigma} = \frac{1}{\tau} u_1 \sigma^2 v_1^2 + \sigma + \frac{1}{\tau} \sigma^2 v_1^2 + \frac{1}{\tau} \sigma^3 v_1^2 w^T w$$

which can be shown to be equal to

$$\bar{\sigma} = -\frac{1}{\tau}\sigma(1 + \rho) = \frac{\sigma(1 + \rho)}{\rho(\theta + \rho)}.$$

The $(n-1) \times (n-1)$ submatrix $I + \bar{\sigma} w w^T$ has the same structure as $I + \sigma v v^T$ and a Householder matrix can be applied in exactly the same fashion. It can be shown that

$$1 + \bar{\sigma} w^T w = \frac{1}{\rho}(1 + \sigma v^T v)$$

and so the sign choice in the definition of each of the Householder matrices remains the same.

For notational convenience, we will write ρ_j, θ_j, δ_j, and σ_{j+1} for the quantities ρ, θ, δ, and $\bar{\sigma}$ at the jth step of the reduction, and use ρ, δ for the vectors (ρ_j), (δ_j).

The full reduction is now

$$(I + \sigma v v^T) P_1 P_2 \cdots P_{n-1} = R(\delta, v, -\rho)^T$$

which gives

$$\bar{A} = L D^{1/2} R(\delta, v, -\rho)^T R(\delta, v, -\rho) D^{1/2} L^T.$$

From Lemma II, we have

$$R(\delta, v, -\rho) D^{1/2} = R(\delta, D^{1/2} v, \rho) = D^{1/2} R(D^{-1/2}\delta, p, \rho) = D^{1/2} \Gamma R(\beta, p, e),$$

where

$$\begin{aligned} \Gamma &= \operatorname{diag}(\rho_j), \\ \left.\begin{aligned} p_j &= d_j^{1/2} v_j \\ \beta_j &= -\delta_j / (d_j^{1/2} \rho_j) \end{aligned}\right\}&, \qquad j = 1, \cdots, n. \end{aligned}$$

(Note that p is the solution of $Lp = z$, as before.)

Following our convention for unit-triangular matrices, we define

$$L(p, \beta, e) = R(\beta, p, e)^T.$$

The net result is that

$$\bar{L} = L L(p, \beta, e) \quad \text{and} \quad \bar{D} = \Gamma D \Gamma,$$

which must be analytically equivalent to the factors obtained by Algorithm C1. What we have done is find alternative expressions for β_j and $\bar{d}_j$, the most important being $\bar{d}_j = \rho_j{}^2 d_j$. Since $\rho_j{}^2$ is computed as a sum of squares, this expression guarantees that the computed $\bar{d}_j$ can never become negative. In Algorithm C1, the corresponding relation is $\bar{d}_j = d_j + \alpha_j p_j{}^2$ where $\operatorname{sign}(\alpha_j) = \operatorname{sign}(\alpha)$. If $\alpha < 0$ and $\bar{L}\bar{D}\bar{L}^T$ is nearly singular, it is possible that rounding errors could give $\bar{d}_j \ll 0$. In such cases, Algorithm C2 is to be preferred.

The analytical equivalence of the two algorithms can be seen through the relation between α_j and σ_j. For example, Eq. (15) implies that

$$\alpha_1 = \sigma_1(2 + \sigma_1 v^T v)$$

and if this is substituted into $\bar{d}_1 = d_1 + \alpha_1 p_1{}^2$ we get

$$\bar{d}_1 = \rho_1^2 d_1,$$

which agrees with $\bar{D} = \Gamma D \Gamma$. In general, if we define

$$\alpha_j = \sigma_j\left(2 + \sigma_j \sum_{i=j}^{n} v_i^2\right),$$

the expression for δ_j simplifies, giving

$$\beta_j = -\frac{\delta_j}{d_j^{1/2}\rho_j} = \frac{\alpha_j v_j}{d_j^{1/2}\rho_j} = \frac{\alpha_j p_j}{d_j \rho_j^2} = \frac{\alpha_j p_j}{\bar{d}_j}$$

which is the expression obtained for β_j in Algorithm C1. In practice, we retain this form for Algorithm C2. The method for computing $\bar{L}$ from L and $L(p, \beta, e)$ is also the same as before. The iteration can be summarized as follows.

Algorithm C2.

1. Solve $Lp = z$.
2. Define

$$\left.\begin{aligned} w_j^{(1)} &= z_j \\ s_j &= \sum_{i=j}^{n} p_i^2/d_i \equiv \sum_{i=j}^{n} q_i \end{aligned}\right\}, \qquad j = 1, 2, \cdots, n,$$

$$\alpha_1 = \alpha,$$

$$\sigma_1 = \alpha/[1 + (1 + \alpha s_1)^{1/2}].$$

3. For $j = 1, 2, \cdots, n$, compute

(a) $\quad q_j = p_j^2/d_j,$

(b) $\quad \theta_j = 1 + \sigma_j q_j,$

(c) $\quad s_{j+1} = s_j - q_j,$

(d) $\quad \rho_j^2 = \theta_j^2 + \sigma_j^2 q_j s_{j+1},$

(e) $\quad \bar{d}_j = \rho_j^2 d_j,$

(f) $\quad \beta_j = \alpha_j p_j/\bar{d}_j,$

(g) $\quad \alpha_{j+1} = \alpha_j/\rho_j^2,$

(h) $\quad \sigma_{j+1} = \sigma_j(1 + \rho_j)/[\rho_j(\theta_j + \rho_j)],$

(i) $$\left.\begin{aligned} w_r^{(j+1)} &= w_r^{(j)} - p_j l_{rj} \\ \bar{l}_{rj} &= l_{rj} + \beta_j w_r^{(j+1)} \end{aligned}\right\}, \qquad r = j + 1, j + 2, \cdots, n.$$

Note that the initial back substitution takes place separately from the computation of $L(p, \beta, e)$, because of the need to compute the vector p before computing s_1. This adds $n^2/2 + O(n)$ multiplications to the method but ensures that the algorithm will always yield a positive definite factorization even under extreme circumstances and allows $\bar{L}$ to be computed by either the forward or backward recurrence relations given in Lemma V. The method requires $3n^2/2 + O(n)$ multiplications and $n + 1$ square roots.

Method C3. *Using Givens Matrices* I. One of the most obvious methods of modifying the Cholesky factors of A in the particular case when $\alpha > 0$ is as follows.

Consider the reduction of the matrix $[\alpha^{1/2}z \ \vdots \ R^T]$ to lower-triangular form, i.e.,

$$[\alpha^{1/2}z \ \vdots \ R^T]P = [\bar{R}^T \ \vdots \ 0],$$

where P is a sequence of Givens matrices of the form $P = P_2{}^1 P_3{}^2 \cdots P_{n+1}{}^n$. We have

$$[\bar{R}^T \ \vdots \ 0]\begin{bmatrix} \bar{R} \\ \hline 0 \end{bmatrix} = \bar{R}^T\bar{R} = [\alpha^{1/2}z \ \vdots \ R^T]PP^T\begin{bmatrix} \alpha^{1/2}z^T \\ \hline R \end{bmatrix} = R^TR + \alpha zz^T.$$

Consequently, $\bar{R}^T$ is the required factor.

This algorithm can be generalized when $\alpha < 0$. The rank-one modification will be written as

$$\bar{R}^T\bar{R} = R^TR - \alpha zz^T, \qquad \alpha > 0,$$

for convenience. The vector p is computed such that

$$R^Tp = z,$$

and we set

$$\delta_n^2 = (1 - \alpha p^Tp)/\alpha.$$

We now form the matrix

$$\left[\begin{array}{c|c} p & R \\ \hline \delta_n & 0 \end{array}\right]$$

and premultiply by an orthogonal matrix P of the form $P = P_1{}^{n+1} \cdots P_{n-1}{}^{n+1}P_n{}^{n+1}$ such that the vector p is reduced to zero. This gives

$$P\left[\begin{array}{c|c} p & R \\ \hline \delta_n & 0 \end{array}\right] = \left[\begin{array}{c|c} 0 & \bar{R} \\ \hline \delta_0 & r^T \end{array}\right]$$

in which case the following relations must hold

(16) $$p^Tp + \delta_n^2 = \delta_0^2,$$

(17) $$R^Tp = \delta_0 r,$$

(18) $$R^TR = \bar{R}^T\bar{R} + rr^T.$$

Equation (16) implies that $\delta_0{}^2 = 1/\alpha$, Eq. (17) implies that $r = z/\delta_0 = \alpha^{1/2}z$, and, finally, (18) gives $R^TR = \bar{R}^T\bar{R} + \alpha zz^T$, as required. This method requires $5n^2/2 + O(n)$ multiplications and $n + 1$ square roots.

Method C4. *Using Givens Matrices* II. For this method, we shall be modifying the factorization

$$\bar{R}^T\bar{R} = R^TR + \alpha zz^T.$$

From this equation we have

(19) $$\bar{A} = R^T(I + \alpha pp^T)R,$$

where $R^Tp = z$. We can write $\bar{A}$ in the form

(20) $$\bar{A} = R^TP^TP(I + \alpha pp^T)P^TPR,$$

where P is an orthogonal matrix. The matrix P is chosen as a product of Givens matrices such that

$$Pp = P_2^1 P_3^2 \cdots P_{n-1}^{n-2} P_n^{n-1} p = \rho e_1, \tag{21}$$

where $|\rho| = \|p\|_2$. Eq. (19) can be written as

$$\bar{A} = R^T P^T (I + \alpha \rho^2 e_1 e_1^T) PR.$$

As each Givens matrix $P_{j+1}{}^{j}$ is formed, it is multiplied into the upper-triangular matrix R. This has the effect of filling in the subdiagonal elements of R to give an upper-Hessenberg matrix H. We have

$$H = PR, \qquad \bar{A} = H^T J^T J H,$$

where J is an identity matrix except for the (1, 1) element which has the value $(1 + \alpha p^T p)^{1/2}$. If $\bar{A}$ is positive definite, the square root will be real. The formation of the product JH modifies the first row of H to give $\bar{H}$ which is still upper Hessenberg.

A second sequence of Givens matrices is now chosen to reduce $\bar{H}$ to upper-triangular form, i.e.,

$$\bar{P}\bar{H} = P_n^{n-1} P_{n-1}^{n-2} \cdots P_3^2 P_2^1 \bar{H} = \bar{R}.$$

Then

$$\bar{A} = \bar{H}^T \bar{H} = \bar{H}^T \bar{P}^T \bar{P} \bar{H} = \bar{R}^T \bar{R}$$

as required. This algorithm requires $9n^2/2 + O(n)$ multiplications and $2n - 1$ square roots.

Method C5. *Using Givens Matrices* III. If we write Eq. (19) as in Method C2, viz.

$$\bar{A} = R^T (I + \sigma p p^T)(I + \sigma p p^T) R,$$

where $\sigma = \alpha/(1 + (1 + \alpha p^T p)^{1/2})$. If P is the matrix defined in (21), we can write

$$\bar{A} = R^T (I + \sigma p p^T) P^T P (I + \sigma p p^T) R = R^T H^T H R, \tag{22}$$

where $H = P(I + \sigma p p^T) = P + \sigma \rho e_1 p^T$. According to Lemma I, P is a special upper-Hessenberg matrix of the form $P = H_U(\beta, \bar{p}, \gamma)$ for some vectors $\bar{p}$, β and γ. Now the first row of P is a multiple of $\bar{p}^T$ by definition, and, furthermore, $Pp = \rho e_1$ implies that $p = \rho P^T e_1$, so the first row of P is also a multiple of p. From Lemma II, it follows that by choosing $\bar{p}_n = p_n$ when forming P as a special matrix, we can ensure that $P = H_U(\beta, p, \gamma)$ for some β and γ.

Assuming this choice of $\bar{p}_n$ is made, we have

$$H = H_U(\beta, p, \gamma) + \sigma \rho e_1 p^T = H_U(\bar{\beta}, p, \gamma)$$

where $\bar{\beta}$ differs from β only in the first element, i.e. $\bar{\beta} = \beta + \sigma \rho e_1$. Now H can be reduced to upper-triangular form $\tilde{R}$ by a second sequence of Givens matrices $\bar{P}$:

$$\bar{P}H = P_n^{n-1} P_{n-1}^{n-2} \cdots P_3^2 P_2^1 H = \tilde{R}.$$

It can be readily shown that $\tilde{R}$ is of the form

$$\tilde{R} = R(\tilde{\beta}, p, \tilde{\gamma})$$

where the vectors $\tilde{\beta}$ and $\tilde{\gamma}$ are given by the following recurrence relations:

1. $\eta_1 = \bar{\beta}_1$;
2. $$\left.\begin{aligned} \tilde{\beta}_i &= c_i\eta_i + s_i\bar{\beta}_i \\ \tilde{\gamma}_i &= c_i\eta_i p_i + s_i\gamma_i \\ \eta_{i+1} &= s_i\eta_i - c_i\bar{\beta}_i \end{aligned}\right\}, \qquad i = 1, 2, \cdots, n-1;$$
3. $\tilde{\gamma}_n = \eta_n$.

The quantities c_i and s_i are the elements of the Givens matrices in $\bar{P}$. They reduce the subdiagonal elements γ_i of H to zero at each stage, and are defined in the usual way. The final product $\bar{R} = \tilde{R}R$ can be computed using Lemma V.

This algorithm requires $2n^2 + O(n)$ multiplications and $2n - 1$ square roots. The work has been reduced, relative to Method C4, by accumulating both sequences of Givens matrices into the special matrix $\tilde{R}$ and modifying R just once, rather than twice.

4. Modification of the Complete Orthogonal Factorization. If A is an $m \times n$ matrix of rank t, $m \geqq n$, $t \leqq n$, the complete orthogonal factorization of A is

$$QAZ = \left[\begin{array}{c|c} R & 0 \\ \hline 0 & 0 \end{array}\right] \tag{23}$$

where Q is an $m \times m$ orthogonal matrix, Z an $n \times n$ orthogonal matrix and R a $t \times t$ upper-triangular matrix (see Faddeev et al. [3], Hanson and Lawson [10]).

The pseudo-inverse of A is given by

$$A^+ = Z\left[\begin{array}{c|c} R^{-1} & 0 \\ \hline 0 & 0 \end{array}\right]Q.$$

In order to obtain the pseudo-inverse of $\bar{A} = A + yz^T$, where y and z are m and n vectors respectively, we consider modifying the complete orthogonal factorization of A. (With no loss of generality we have omitted the scalar α.)

From Eq. (23), we have

$$Q\bar{A}Z = \left[\begin{array}{c|c} R & 0 \\ \hline 0 & 0 \end{array}\right] + pq^T$$

where $p = Qy$ and $q = Z^Tz$. If the vectors p and q are partitioned as follows:

$$p = \begin{bmatrix} u \\ \hline \bar{u} \end{bmatrix}\begin{matrix} \}t \\ \}m-t \end{matrix}, \qquad q = \begin{bmatrix} w \\ \hline \bar{w} \end{bmatrix}\begin{matrix} \}t \\ \}n-t \end{matrix},$$

we can choose Q_{I} and Z_{I} to be either single Householder matrices or products of Givens matrices such that

$$Q_{\mathrm{I}}\bar{u} = \alpha e_1 \quad \text{and} \quad \bar{w}^T Z_{\mathrm{I}} = \beta e_1^T,$$

where α and β are scalars such that $|\alpha| = ||\bar{u}||_2$ and $|\beta| = ||\bar{w}||_2$. Note that application of these matrices leaves the matrix R unchanged. For convenience, we will now work with the $(t + 1) \times (t + 1)$ matrix S_{I} which is defined as

$$S_{\mathrm{I}} = \left[\begin{array}{c|c} R & 0 \\ \hline 0 & 0 \end{array}\right] + \left[\begin{array}{c} u \\ \hline \alpha \end{array}\right] [w^T \ \vdots \ \beta].$$

We next perform two major steps which will be called *sweeps*.

First Sweep. Choose an orthogonal matrix Q_{II} such that

$$Q_{\mathrm{II}} \left[\begin{array}{c} u \\ \hline \alpha \end{array}\right] = P_2^1 P_3^2 \cdots P_t^{t-1} P_{t+1}^t \left[\begin{array}{c} u \\ \hline \alpha \end{array}\right] = \gamma_1 e_1$$

where $\gamma_1^2 = \|u\|_2^2 + \alpha^2$. If S_{I} is multiplied on the left by Q_{II} and the resulting product defined as S_{II}, we have

$$S_{\mathrm{II}} = Q_{\mathrm{II}} S_{\mathrm{I}} = \left[\begin{array}{c|c} r_{\mathrm{II}}^T & 0 \\ \hline R_{\mathrm{II}} & 0 \end{array}\right] + \gamma_1 e_1 [w^T \ \vdots \ \beta] = \left[\begin{array}{c|c} \bar{r}_{\mathrm{II}}^T & \gamma_1 \beta \\ \hline R_{\mathrm{II}} & 0 \end{array}\right],$$

where R_{II} is an upper-triangular matrix. The t diagonal elements of R_{II} are filled in one at a time by the application of each 2×2 orthogonal matrix. We have defined

$$\bar{r}_{\mathrm{II}}^T = r_{\mathrm{II}}^T + \gamma_1 w^T.$$

Second Sweep. We now construct an orthogonal matrix Q_{III} which, when applied to S_{II} from the left, reduces S_{II} to upper-triangular form. If this triangular matrix is defined as S_{III}, we have

$$S_{\mathrm{III}} = Q_{\mathrm{III}} \left[\begin{array}{c|c} \bar{r}_{\mathrm{II}}^T & \gamma_1 \beta \\ \hline R_{\mathrm{II}} & 0 \end{array}\right] = \left[\begin{array}{c|c} R_{\mathrm{III}} & s_{\mathrm{III}} \\ \hline 0 & \delta_{\mathrm{III}} \end{array}\right],$$

where Q_{III} is of the form

$$Q_{\mathrm{III}} = P_{t+1}^t \cdots P_3^2 P_2^1.$$

The matrix S_{III} may or may not be the upper-triangular matrix required, depending upon $\rho(\bar{A})$, the rank of $\bar{A}$. The different cases that can arise are summarized in the following table:

$\alpha \backslash \beta$	$= 0$	$\neq 0$
$= 0$	$\rho(\bar{A}) = t$ or $t - 1$	$\rho(\bar{A}) = t$
$\neq 0$	$\rho(\bar{A}) = t$	$\rho(\bar{A}) = t + 1$

Case I. $\alpha \neq 0$, $\beta \neq 0$. In this case, S_{III} has full rank and

$$\left[\begin{array}{c|c} R_{\mathrm{III}} & s_{\mathrm{III}} \\ \hline 0 & \delta_{\mathrm{III}} \end{array}\right] = \bar{R}.$$

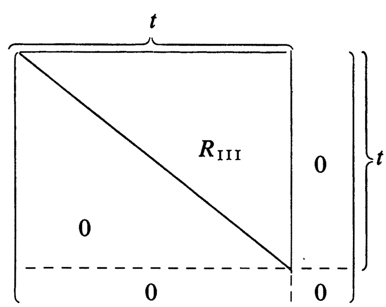
The final orthogonal matrix $\bar{Q}$ is given by

$$\bar{Q} = \underbrace{\begin{bmatrix} Q_{\mathrm{III}} & 0 \\ 0 & I \end{bmatrix}}_{t+1} \underbrace{\begin{bmatrix} Q_{\mathrm{II}} & 0 \\ 0 & I \end{bmatrix}}_{t+1} \underbrace{\begin{bmatrix} I & 0 \\ 0 & Q_{\mathrm{I}} \end{bmatrix}}_{t} Q \tag{24}$$

and

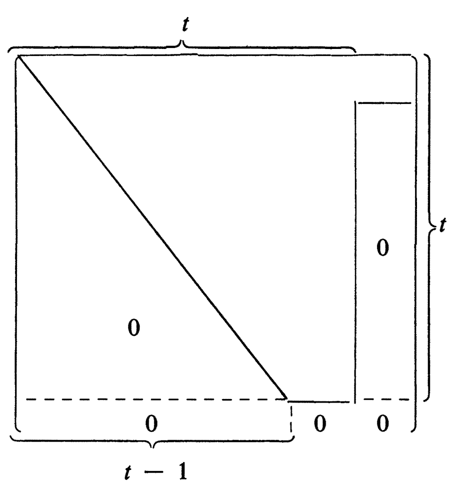
$$\bar{Z} = Z \underbrace{\begin{bmatrix} I & 0 \\ 0 & Z_{\mathrm{I}} \end{bmatrix}}_{t}.$$

Case II. $\alpha \neq 0$, $\beta = 0$. If the first and second sweeps are followed carefully, it can be seen that S_{III} is of the form

$S_{\mathrm{III}} =$ 

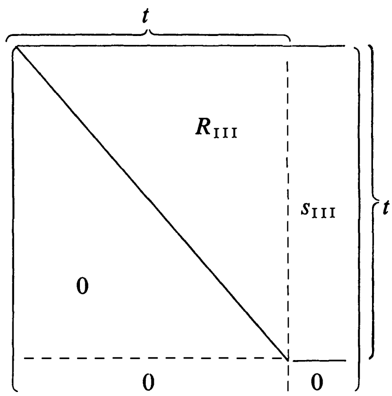
i.e., $s_{\mathrm{III}} = 0$ and $\delta_{\mathrm{III}} = 0$. As in Case I, S_{III} is in the required form and we define the modified factors accordingly.

Case III. $\alpha = 0$, $\beta \neq 0$. The first orthogonal transformation of the first sweep is an identity, and the matrix S_{II} has the form

$S_{\mathrm{II}} =$ 

Application of the second sweep (Q_{III}) gives the matrix S_{III} in the form

$S_{III} =$



i.e., $\delta_{III} = 0$.

An orthogonal matrix Z_{II} is now applied on the right to reduce s_{III} to zero, thus

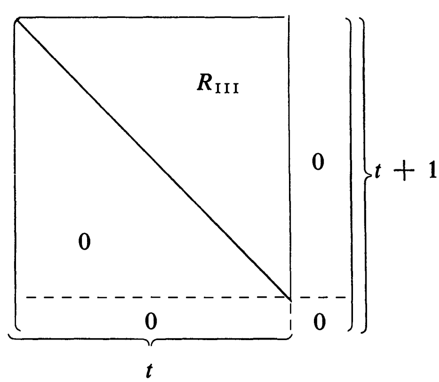
$$S_{III}Z_{II} = S_{III}P^{t}_{t+1}P^{t-1}_{t+1}P^{t-2}_{t+1} \cdots P^{1}_{t+1} = \left[\begin{array}{c|c} \bar{R} & 0 \\ \hline 0 & 0 \end{array}\right].$$

The modified factors are $\bar{Q}$ as defined in (22), and

$$\bar{Z} = Z\left[\begin{array}{c|c} I & \\ \hline & Z_{I} \end{array}\right]\left[\begin{array}{c|c} Z_{II} & \\ \hline & I \end{array}\right].$$

Case IV. $\alpha = 0$, $\beta = 0$, $\rho(\bar{A}) = t$. The matrix S_{III} has the following form:

$S_{III} =$



If the diagonal elements of R_{III} are all nonzero, then $\text{rank}(\bar{A}) = \text{rank}(R_{III}) = t$ and the factors are completely determined. Otherwise, exactly one of the diagonal elements of R_{III} may be zero, since the rank of $\bar{A}$ can drop to $t - 1$. In this case, two more partial sweeps must be made to reduce R_{III} to strictly upper-triangular form, as follows.

Case V. $\alpha = 0$, $\beta = 0$, $\rho(\bar{A}) = t - 1$. Suppose that the kth diagonal of R_{III} is zero. The matrix can be partitioned in the form

$$R_{\mathrm{III}} = \begin{array}{cccc} k-1 & & t-k & \\ \left[\begin{array}{c|c|c} R_{\mathrm{IV}} & s_{\mathrm{IV}} & W \\ \hline & & r_{\mathrm{IV}}^T \\ \hline & & R_{\mathrm{V}} \end{array}\right] & \begin{array}{l} \}\, k-1 \\ \\ \}\, t-k \end{array} \end{array}$$

where R_{IV}, R_{V} are upper-triangular with dimensions $(k-1) \times (k-1)$ and $(t-k) \times (t-k)$, respectively. An orthogonal transformation Q_{IV} is now applied on the left to reduce the submatrix

$$\begin{bmatrix} r_{\mathrm{IV}}^T \\ \hdashline R_{\mathrm{V}} \end{bmatrix}$$

to upper-triangular form in exactly the same way as the first sweep. Similarly, a transformation Z_{II} is applied (independently) from the right to reduce s_{IV} to zero in the submatrix $[R_{\mathrm{IV}}\ s_{\mathrm{IV}}]$. Thus

$$Q_{\mathrm{IV}} R_{\mathrm{III}} Z_{\mathrm{II}} = \left(\begin{array}{c:c:c} \bar{R}_{\mathrm{IV}} & 0 & W \\ \hdashline \multicolumn{2}{c:}{0} & \bar{R}_{\mathrm{V}} \\ \cdashline{3-3} & & 0 \end{array}\right)$$

where $Q_{\mathrm{IV}} = P_t{}^k P_{t-1}{}^k \cdots P_{k+2}{}^k P_{k+1}{}^k$ and $Z_{\mathrm{II}} = P_k{}^{k-1} P_k{}^{k-2} \cdots P_k{}^2 P_k{}^1$.

Finally, a permutation matrix Z_{III} is applied to move the column of zeros to the right:

$$\left(\begin{array}{c:c:c} \bar{R}_{\mathrm{IV}} & 0 & W \\ \hdashline \multicolumn{2}{c:}{0} & \bar{R}_{\mathrm{V}} \\ \cdashline{3-3} & & 0 \end{array}\right) Z_{\mathrm{III}} = \left(\begin{array}{cc:c} \bar{R}_{\mathrm{IV}} & W & \\ & \bar{R}_{\mathrm{V}} & 0 \\ \hdashline \multicolumn{2}{c:}{0} & 0 \end{array}\right) = \left(\begin{array}{c:c} \bar{R} & 0 \\ \hdashline 0 & 0 \end{array}\right).$$

The modified factors are

$$\bar{Q} = \left[\begin{array}{c|c} Q_{\mathrm{IV}} & \\ \hline & I \end{array}\right] \left[\begin{array}{c|c} Q_{\mathrm{III}} & \\ \hline & I \end{array}\right] \left[\begin{array}{c|c} Q_{\mathrm{II}} & \\ \hline & I \end{array}\right] \left[\begin{array}{c|c} I & \\ \hline & Q_{\mathrm{I}} \end{array}\right] Q$$

and

$$\bar{Z} = Z \left[\begin{array}{c|c} I & \\ \hline & Z_{\mathrm{I}} \end{array}\right] \left[\begin{array}{c|c} Z_{\mathrm{II}} & \\ \hline & I \end{array}\right] \left[\begin{array}{c|c} Z_{\mathrm{III}} & \\ \hline & I \end{array}\right].$$

The number of operations necessary to compute the modified factors are summarized in the following table:

Description	Order of multiplications
Compute p, q	$m^2 + n^2$
Determine α, β	$4m(m - t) + 4n(n - t)$
First sweep	$2t^2 + 4mt$
Second sweep	$2t^2 + 4mt$
Additional computation for case III	$2t^2 + 4nt$
*Additional computation for case V	$\frac{4}{3} t^2 + 2t(n + m)$

* It has been assumed that if $W(k)$ is the amount of work when the kth diagonal element of R_{III} is zero, then the expected work is $(1/t) \sum_{k=1}^{t} W(k)$.

The maximum amount of computation necessary, which is of the order of $6\frac{2}{3}t^2 + 5(m^2 + n^2) + 2t(3m - n)$ multiplications, will occur when Case V applies. In the special case, when $\bar{A}$ and A are both of full column rank, then Z is the identity matrix and the amount of computation is of the order of $5m^2 + 4n^2 + 4mn$ multiplications. This reduces to $13n^2$ when $m = n$.

4.1. *Use of Special Matrices.* The number of operations can be decreased if some of the properties of special matrices outlined in Section 2 are utilized. Each Givens matrix must be multiplied into a Q matrix, Z matrix or upper-triangular matrix, depending upon the current stage of the algorithm. These multiplications can be performed by accumulating the product of each set of Givens matrices into the associated special matrix. Each Q_{I}, Z_{I}, Q_{II}, Z_{II}, $\cdots$, etc. will be either a special matrix or a permutation matrix. The orthogonal matrices Q_{I}, Z_{I}, $\cdots$, etc. will be formed, using Lemma I and Lemma II, as products of the form $\Delta_{\mathrm{I}}\tilde{Q}_{\mathrm{I}}$, $\nabla_{\mathrm{I}}\tilde{Z}_{\mathrm{I}}$, $\Delta_{\mathrm{II}}\tilde{Q}_{\mathrm{II}}$, $\nabla_{\mathrm{II}}\tilde{Z}_{\mathrm{II}}$, $\cdots$, etc. where Δ_{I}, ∇_{I}, Δ_{II}, ∇_{II}, $\cdots$, etc. are diagonal matrices and $\tilde{Q}_{\mathrm{I}}$, $\tilde{Z}_{\mathrm{I}}$, $\cdots$, etc. are special upper- (lower-) Hessenberg matrices with unit sub- (super-) diagonals. In addition, we assume that we modify the factorization

$$QAZ = \left[\begin{array}{c|c} DL^T & 0 \\ \hline 0 & 0 \end{array}\right].$$

At the initial stage, DL^T is unaffected by the pre- and post-multiplication with $\Delta_{\mathrm{I}}\tilde{Q}_{\mathrm{I}}$ and $\tilde{Z}_{\mathrm{I}}\nabla_{\mathrm{I}}$. The products

$$\left[\begin{array}{c|c} I & 0 \\ \hline 0 & \Delta_{\mathrm{I}}\tilde{Q}_{\mathrm{I}} \end{array}\right] Q, \qquad Z \left[\begin{array}{c|c} I & 0 \\ \hline 0 & \tilde{Z}_{\mathrm{I}}\nabla_{\mathrm{I}} \end{array}\right]$$

can be formed using Lemma III, the diagonal matrices being kept separate from the orthogonal products.

During the first sweep, we require the product

$$Q_{\mathrm{II}}\left[\begin{array}{c|c} R & 0 \\ \hline 0 & 0 \end{array}\right].$$

If this matrix is written in the form

$$\Delta_{\mathrm{II}}\tilde{Q}_{\mathrm{II}}\left[\begin{array}{c|c} DL^T & 0 \\ \hline 0 & 0 \end{array}\right],$$

it can be evaluated by bringing the diagonal matrix D to the left of $\tilde{Q}_{\mathrm{II}}$ by suitably altering the special matrix $\tilde{Q}_{\mathrm{II}}$ to $\tilde{Q}_{\mathrm{II}}'$ as in Lemma II. The remaining product involving $\tilde{Q}_{\mathrm{II}}'$ and L^T can be formed using Lemma III with backward recurrence. The multiplication of $\tilde{Q}_{\mathrm{II}}'$ by the current orthogonal matrix is performed similarly to that involving $\tilde{Q}_{\mathrm{I}}$ except that again the diagonal Δ_{I} must be brought through by altering $\tilde{Q}_{\mathrm{II}}$ to $\tilde{Q}_{\mathrm{II}}''$ (say).

If the remainder of the computation is carried out using the same techniques as those just described, the number of multiplications can be summarized as follows:

Description	Order of multiplications
Compute p, q	$m^2 + n^2$
Determine α, β	$2m(m - t) + 2n(n - t)$
First sweep	$t^2 + 2mt$
Second sweep	$2t^2 + 2mt$
Additional computation for case III	$2t^2 + 2nt$
Additional computation for case V	$\frac{4}{3}t^2 + t(n + m)$

The maximum amount of computation necessary is now of the order of $4\frac{1}{3}t^2 + 3(m^2 + n^2) + t(3m - n)$ multiplications, and this reduces to $3(m^2 + n^2) + 2mn$ multiplications in the full rank case. When $n = m = t$ the algorithm requires $8n^2 + O(n)$ operations.

5. Special Rank-One Modifications. We now consider some special cases of the complete orthogonal factorization which occur frequently, namely adding and deleting rows and columns from A. These cases deserve special attention because the modifications can be done in approximately half as many operations as in the general case. Since, in most applications, A is of full column rank, we will deal specifically with this case and modify the factorization

$$QA = \left[\begin{array}{c} R \\ \hline 0 \end{array}\right]$$

where A is $m \times n$, $m \geqq n$.

5.1. *Adding and Deleting Rows of A.* We first consider adding a row a^T to A. Assuming, without loss of generality, that this row is added in the $(m + 1)$th position, we have

$$\left[\begin{array}{c|c} Q & 0 \\ \hline 0 & 1 \end{array}\right]\left[\begin{array}{c} A \\ \hline a^T \end{array}\right] = \left[\begin{array}{c} R \\ \hline 0 \\ \hline a^T \end{array}\right] \equiv T.$$

Elementary orthogonal transformations are now applied from the left to reduce a^T to zero while maintaining the triangularity of R. This is done by defining the sequence

$$T^{(1)} = T, \quad T^{(j+1)} = P_{m+1}^{j} T^{(j)}, \qquad j = 1, 2, \cdots, n,$$

where $P_{m+1}{}^{j}$ reduces the $(m + 1, j)$th element of $T^{(j)}$ to zero. Note in particular the effect on the column e_{m+1} which has been added to Q. The first n elements are filled in one by one, thereby forming the last column of $\bar{Q}$:

$$P_{m+1}^{n} P_{m+1}^{n-1} \cdots P_{m+1}^{1} \left[\begin{array}{c|c} Q & e_{m+1} \\ \hline 0 & \end{array}\right] = \bar{Q} = [\bar{Q}_m \,\vdots\, \bar{q}_{m+1}] \quad \text{say}.$$

Elements $n + 1, n + 2, \cdots, m$ of $\bar{q}_{m+1}$ remain zero.

To *remove* a row from A, we now simply reverse the above process. This time, we have

$$QA = \left[\begin{array}{c|c} Q_m & q_{m+1} \end{array}\right]\left[\begin{array}{c} \bar{A} \\ \hline a^T \end{array}\right] = \left[\begin{array}{c} R \\ \hline 0 \\ \hline 0 \end{array}\right] \begin{array}{l} \}n \\ \}m - n \\ \}1 \end{array}$$

giving $Q_m\bar{A} + q_{m+1}a^T = QA$. Transformations $P_m{}^{m+1}, P_{m-1}{}^{m+1}, \cdots, P_1{}^{m+1}$ are chosen such that

$$Pq_{m+1} \equiv P_1^{m+1} \cdots P_{m-1}^{m+1} P_m^{m+1} q_{m+1} = e_{m+1}.$$

The last n transformations each introduce a nonzero into the bottom row of

$$\left[\begin{array}{c} R \\ \hline 0 \\ \hline 0 \end{array}\right]$$

(from right to left), giving

$$PQA = \left[\begin{array}{c} \bar{R} \\ \hline 0 \\ \hline r^T \end{array}\right].$$

Looking at the effect on the various partitions of Q, we have

$$PQ = \left[\begin{array}{c|c} \bar{Q} & 0 \\ \hline u^T & 1 \end{array}\right]$$

and, since PQ is orthogonal, it follows immediately that $u = 0$. Thus

$$PQ\left[\begin{array}{c}\bar{A}\\ \hline a^T\end{array}\right] = \left[\begin{array}{c|c}\bar{Q} & 0\\ \hline 0 & 1\end{array}\right]\left[\begin{array}{c}\bar{A}\\ \hline a^T\end{array}\right] = \left[\begin{array}{c}\bar{R}\\ \hline 0\\ \hline r^T\end{array}\right]$$

so that $r = a$, and also

$$\bar{Q}\bar{A} = \left[\begin{array}{c}\bar{R}\\ \hline 0\end{array}\right]$$

as required.

Often, it is necessary to modify R without the help of Q. In this case, we really want $\bar{R}$ such that

$$\bar{R}^T\bar{R} = R^TR \pm aa^T,$$

so, clearly, the methods of Section 3 would be applicable. Alternatively, we can continue to use elementary orthogonal transformations as just described. Adding a row to A is simple because Q was not required in any case. To delete a row, we first solve $R^Tp = a$ and compute $\delta^2 = 1 - ||p||^2$. The vector

$$\left[\begin{array}{c}p\\ \hline 0\\ \hline \delta\end{array}\right]\begin{array}{l}\}n\\ \}m-n\\ \}1\end{array} \tag{25}$$

now plays exactly the same role as q_{m+1} above. Dropping the unnecessary zeros in the center of this vector, we have

$$P_1^{n+1} \cdots P_{n-1}^{n+1}P_n^{n+1}\left[\begin{array}{c|c}p & R\\ \hline \delta & 0\end{array}\right] = \left[\begin{array}{c|c}0 & \bar{R}\\ \hline 1 & r^T\end{array}\right]$$

where as usual, the sequence $\{P_i^{n+1}\}$ has the effect of reducing p in (23) to zero and introducing the vector r^T beneath $\bar{R}$. Since the P_i^{n+1} are orthogonal, it follows that

$$\left[\begin{array}{c|c}0 & 1\\ \hline \bar{R}^T & r\end{array}\right]\left[\begin{array}{c|c}0 & \bar{R}\\ \hline 1 & r^T\end{array}\right] = \left[\begin{array}{c|c}p^T & \delta\\ \hline R^T & 0\end{array}\right]\left[\begin{array}{c|c}p & R\\ \hline \delta & 0\end{array}\right]$$

or

$$\left[\begin{array}{c|c}1 & r^T\\ \hline r & \bar{R}^T\bar{R} + rr^T\end{array}\right] = \left[\begin{array}{c|c}||p||^2 + \delta^2 & p^TR\\ \hline R^Tp & R^TR\end{array}\right]$$

so that $r = R^Tp = a$, and $\bar{R}^T\bar{R} = R^TR - aa^T$ as required.

5.2. *Adding and Deleting Columns of A.* Suppose a column is added to the matrix A, giving

$$\bar{A} = [A \; \vdots \; a].$$

Since

$$QA = \left[\begin{array}{c} R \\ \hline 0 \end{array}\right],$$

we have

$$QĀ = \left[\begin{array}{c|c} R & u \\ \hline 0 & v \end{array}\right], \tag{26}$$

where $[u^T \vdots v^T] = a^T Q^T$ and u and v are n and $m - n$ vectors, respectively. If an orthogonal matrix P is constructed such that

$$P\left[\begin{array}{c} u \\ \hline v \end{array}\right] = \left[\begin{array}{c} u \\ \hline \gamma \\ \hline 0 \end{array}\right],$$

where $\gamma = \pm ||v||_2$, then, premultiplying (24) by P leaves the upper-triangular matrix R unchanged and the new factors of $\bar{A}$ are

$$\bar{R} = \left[\begin{array}{c|c} R & u \\ \hline 0 & \gamma \end{array}\right] \quad \text{and} \quad \bar{Q} = PQ.$$

This method represents just a columnwise recursive definition of the QR factorization of A.

When Q is not stored or is unavailable, the vector u can be found by solving the system

$$R^T u = A^T a.$$

The scalar γ is then given by the relation

$$\gamma^2 = ||a||_2^2 - ||u||_2^2.$$

Rounding errors could cause this method to fail, however, if the new column a is nearly dependent on the columns of A. In fact, if R is built up by a sequence of these modifications, in which the columns of A are added one by one, the process is exactly that of computing the product $B = A^T A$ and finding the Cholesky factorization $B = R^T R$. It is well known that this is numerically less satisfactory than computing R using orthogonal matrices. In some applications, the sth column of $\bar{Q}$ is available even when $\bar{Q}$ is not and, consequently, γ can be computed more accurately from the relationship $\gamma = a^T \bar{q}_s$, where $\bar{q}_s$ is the sth column of $\bar{Q}$.

Some improvement in accuracy can also be obtained on machines which have the facility for performing the double-length accumulation of inner-products. In this case, the ith element of u is set to

$$u_i = \frac{1}{r_{ii}}\left\{\sum_{j=1}^{n} a_{ij}a_j - \sum_{j=1}^{i-j} u_j r_{ij}\right\},$$

where the two inner-products are formed as a single sum. Despite these improvements, this is still numerically less satisfactory than the previous method where Q was available.

A further possibility of improving the method arises when one column is being deleted and another is being added. A new column replacing the deleted column is equivalent to a rank-two change in A^TA and can be performed by any one of the methods given in Section 3. Even this is still not ideal, since the computation of the rank-one vectors require the matrix vector product $A^T(a - \bar{a})$, where a is the column being added and $\bar{a}$ is the column being deleted.

Finally, we describe how to modify the factors when a column is deleted from A. It will be assumed that $\bar{A}$ is obtained from A by deleting the sth column, which as usual will be denoted by a. Deleting the sth column of R gives

$$Q\bar{A} = \left(\begin{array}{c|c} R_1 & T_1 \\ \hline 0 & T_2 \\ \hline 0 & 0 \end{array}\right) \begin{array}{l} \}s - 1 \\ \}n - s + 1 \\ \}m - n \end{array}$$

where R_1 is an $(s - 1) \times (s - 1)$ upper-triangular matrix, T_1 is an $(s - 1) \times (n - s)$ rectangular matrix and T_2 is an $(n - s + 1) \times (n - s)$ upper-Hessenberg matrix. For example, with $n = 5$, $s = 3$ and $m = 7$, we have

$$\left(\begin{array}{c|c} R_1 & T_1 \\ \hline 0 & T_2 \\ \hline 0 & 0 \end{array}\right) = \left(\begin{array}{cc|cc} x & x & x & x \\ 0 & x & x & x \\ \hline 0 & 0 & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right).$$

Let partition T_2 be of the form

$$T_2 = \left(\begin{array}{c} r^T \\ \hline R_2 \end{array}\right) \begin{array}{l} \}1 \\ \}n - s \end{array}.$$

We now choose an orthogonal matrix P which reduces T_2 to upper-triangular form, using one of the methods described earlier. Thus

$$PT_2 = \left(\begin{array}{c} \bar{R}_2 \\ \hline 0 \end{array}\right) \begin{array}{l} \}n - s \\ \}1 \end{array}$$

where P is of the form $P = P_{n-s+1}{}^{n-s} \cdots P_3{}^2 P_2{}^1$. The modified triangular factor for $\bar{A}$ is

$$\bar{R} = \left[\begin{array}{c|c} R_1 & T_1 \\ \hline 0 & \bar{R}_2 \\ \hline 0 & 0 \end{array}\right] \begin{array}{l} \}s - 1 \\ \}n - s \\ \}m - n + 1 \end{array}$$

If Q is to be updated also, the appropriate rows must be modified; thus

$$Q = \left[\begin{array}{c} Q_1 \\ \hline Q_2 \\ \hline Q_3 \end{array}\right] \begin{array}{l} \}s - 1 \\ \}n - s + 1, \\ \}m - n \end{array} \qquad \bar{Q} = \left[\begin{array}{c} Q_1 \\ \hline PQ_2 \\ \hline Q_3 \end{array}\right].$$

It is sometimes profitable to regard this computation from a different point of view. The partitions of T_2 satisfy the relation $\bar{R}_2{}^T\bar{R}_2 = R_2{}^TR_2 + rr^T$, and this is analogous to the equation $\bar{R}^T\bar{R} = R^TR + aa^T$ which holds when we add a row a^T to A. We conclude that *deleting a column* may be accomplished by essentially the same techniques as used for *adding a row*.

6. Conclusions. In this report, we have presented a comprehensive set of methods which can be used to modify nearly all the factorizations most frequently used in numerical linear algebra. It has not been our purpose to recommend a particular method where more than one exist. Although the amount of computation required for each is given, this will not be the only consideration since the relative efficiencies of the algorithms may alter when applied to particular problems. An example of this is when the Cholesky factors of a positive definite matrix are stored in product form. In this case, the choice of algorithm is restricted to those that form the special matrices explicitly. The relative efficiencies of Methods C1 and C2 are consequently altered.

Division of Numerical Analysis and Computing
National Physical Laboratory
Teddington, England

Computer Science Department
Stanford University
Stanford, California 94305

Division of Numerical Analysis and Computing
National Physical Laboratory
Teddington, England

Applied Mathematics Division
Division of Scientific and Industrial Research
Wellington, New Zealand

* 1. R. H. Bartels, G. H. Golub & M. A. Saunders, "Numerical techniques in mathematical programming," *Nonlinear Programming* (Proc. Sympos., Univ. of Wisconsin, Madison, Wis., 1970), Academic Press, New York, 1970, pp. 123–176. MR **42** #7277.

2. J. M. Bennett, "Triangular factors of modified matrices," *Numer. Math.*, v. 7, 1965, pp. 217–221. MR **31** #1766.

3. D. K. Faddeev, V. N. Kublanovskaya & V. N. Faddeeva, "Sur les systèmes linéaires algébriques de matrices rectangulaires et mal-conditionnées," *Programmation en Mathé-*

matiques Numériques (Actes Colloq. Internat. C.N.R.S. No. 165, Besançon, 1966), Éditions Centre Nat. Recherche Sci., Paris, 1968, pp. 161–170. MR **37** #6017.

* 4. W. M. GENTLEMAN, *Least Squares Computations by Givens Transformations Without Square Roots*, Department of Applied Analysis and Computer Science Report CSRR-2062, University of Waterloo, Waterloo, Ontario, 1972.

* 5. P. E. GILL & W. MURRAY, *A Numerically Stable Form of the Simplex Algorithm*, National Physical Laboratory, DNAM Report Number 87, Teddington, England, 1970.

* P. E. GILL & W. MURRAY, "Quasi-Newton methods for unconstrained optimization," *J. Inst. Math. Appl.*, v. 9, 1972, pp. 91–108. MR **45** #9456.

* 7. P. E. GILL & W. MURRAY, *Two Methods For the Solution of Linearly Constrained and Unconstrained Optimization Problems*, National Physical Laboratory, DNAC Report Number 25, Teddington, England, 1972.

* 8. G. H. GOLUB & M. A. SAUNDERS, "Linear least squares and quadratic programming," *Integer and Nonlinear Programming*, North-Holland, Amsterdam, 1970.

* 9. G. H. GOLUB & P. H. STYAN, *Numerical Computations for Univariate Linear Models*, Computer Science Department Report Number CS-236-71, Stanford University, Stanford, California, 1971.

10. R. J. HANSON & C. L. LAWSON, "Extensions and applications of the Householder algorithm for solving linear least squares problems," *Math. Comp.*, v. 23, 1969, pp. 787–812. MR **41** #2905.

11. R. S. MARTIN, G. PETERS & J. H. WILKINSON, "The *QR* algorithm for real Hessenberg matrices," *Handbook for Automatic Computation*, Vol. 2. Edited by J. H. Wilkinson and C. Reinsch, Springer-Verlag, Berlin and New York, 1971, pp. 359–371.

* 12. W. MURRAY, *An Algorithm for Indefinite Quadratic Programming*, National Physical Laboratory, DNAC Report Number 1, Teddington, England, 1971.

13. G. PETERS & J. H. WILKINSON, "The least squares problem and pseudo-inverses," *Comput. J.*, v. 13, 1970, pp. 309–316.

* 14. M. A. SAUNDERS, *Large-Scale Linear Programming Using the Cholesky Factorization*, Computer Science Department Report Number CS-72-252, Stanford University, Stanford, California, 1972.

* 15. M. A. SAUNDERS, *Product Form of the Cholesky Factorization for Large-Scale Linear Programming*, Computer Science Department Report Number CS-72-301, Stanford University, Stanford, Calif., 1972.

16. J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965. MR **32** #1894.