MATHEMATICS OF COMPUTATION, VOLUME 28, NUMBER 128
OCTOBER 1974, PAGES 1067–1087

On the Modification of LDL^T Factorizations

By R. Fletcher and M. J. D. Powell

Abstract. A number of methods are compared for modifying an LDL^T factorization of a positive definite matrix A when a matrix of rank one is added to A. Both the efficiency and error propagation characteristics are discussed, and it is shown that a suitable method must be chosen with care. An error analysis of some of the algorithms is given which has novel features. A worked example which also illustrates error growth is presented. Extensions to the methods are desribed which enable them to be used advantageously in representing and updating positive semidefinite matrices.

1. Introduction. Assume that an $n \times n$ positive definite symmetric matrix A is given in the factorized form

$$A = LDL^T = \sum_{i=1}^{n} \mathbf{l}_i d_i \mathbf{l}_i^T \tag{1.1}$$

where $L = [\mathbf{l}_1, \mathbf{l}_2, \cdots, \mathbf{l}_n]$ is a lower triangular matrix with unit diagonal elements, and D is a diagonal matrix with diagonal elements $d_i > 0$. This paper is concerned with the problem of computing the factors of a modified matrix

$$\widetilde{A} = A + \sigma \mathbf{z}\mathbf{z}^T \tag{1.2}$$

in an efficient manner, where $\widetilde{A}$ is known from other considerations to be positive definite. Thus it is necessary to compute a unit lower triangular matrix $\widetilde{L}$ and a diagonal matrix $\widetilde{D}$ with $\widetilde{d}_i > 0$ such that

$$\widetilde{A} = \widetilde{L}\widetilde{D}\widetilde{L}^T = \sum_{i=1}^{n} \widetilde{\mathbf{l}}_i \widetilde{d}_i \widetilde{\mathbf{l}}_i^T. \tag{1.3}$$

The problem has a well-known application to quasi-Newton methods for unconstrained optimization (see Gill and Murray [5], for example). In Section 2 of this paper, it is shown very simply how to write down an algorithm for a solution of this problem, and this algorithm turns out to be a specialization of one given by Bennett [1]. He considers modifying the factors of a general matrix $A = LDU$ when a matrix XCY^T of rank k $(k \leqslant n)$ is added to A, and the specialization is that for which $U = L^T$ and $k = 1$. Gentleman [3] suggests an algorithm for computing the

Received August 24, 1973.

AMS (MOS) subject classifications (1970). Primary 65F30; Secondary 15A24.

Key words and phrases. Matrix updating, error analysis, updating a triangular factorization, quasi-Newton method, positive definite matrix, rank one matrix, rank one correction.

triangular factors of the normal matrix in a linear least squares calculation. Building up this normal matrix, an observation at a time, is essentially repetition of (1.2), so Gentleman's method also turns out to be applicable to this problem, and has some features in common with the Bennett method. Gill and Murray [5] suggest two other methods which are apparently different and more expensive than the Bennett and Gentleman methods. However, it turns out that one of these methods is very closely related to the Bennett method. The relationships between these various methods are brought out in Section 2, where we also introduce a new method of similar type, but with important differences of detail.

It is important to know which of these various algorithms, if any, is suitable for use in a general purpose subroutine. In Section 3, we consider this problem, looking for an algorithm of general applicability, which guarantees $\tilde{A}$ to be positive definite, which has good error growth properties, and which is otherwise efficient from time and storage considerations. We show that the problem of maintaining positive definiteness in $\tilde{A}$, when round-off errors are present, must be tackled carefully, and we give an example which shows that with some strategies, arbitrarily large error growth can take place. We also consider growth of error in general, and an error analysis of some of the methods is given in Section 5. Again, not all algorithms are stable in this respect. However, it is possible to construct algorithms which are satisfactory in all respects, and we find that a variation of our new method is preferable. A worked example is presented in Section 6, which also illustrates how errors can grow if care is not taken.

The detailed error analysis in Section 5 possesses novel features. A posteriori bounds are given on each matrix element, rather than being expressed in terms of norms of the overall matrix. Because of this, it has been possible to show that the bounds are independent of any bad scaling which may be present in A. A similar analysis enables bounds for the least squares calculation to be determined. In this case, A has the form $A = B^TB$ where B is an $m \times n$ matrix $(m > n)$. Fletcher and Powell produced the draft of a report early in 1973, in which this extension was carried out, and these bounds were superior to those given by Gentleman [3]. However, for the least squares problem it is more appropriate to relate errors to the matrix B, so as to avoid any ill-conditioning which might be implied in forming the matrix A. This has been done in a more recent report due to Gentleman [2].

In Section 4, it is shown how positive semidefinite matrices, such as projection matrices, may be represented. This problem has an application to methods for optimization, subject to linear constraints on the variables. Advantages of the LDL^T representation, as against explicit storage of A, are described.

2. Methods for Updating LDL^T.

A simple way exists of determining the factorization of $\tilde{A}$. It follows from the observation that $d_1\mathbf{l}_1\mathbf{l}_1^T + \sigma\mathbf{z}\mathbf{z}^T$ (where $l_{11} = 1$) can be written as $\lambda\mathbf{x}\mathbf{x}^T + \mu\mathbf{y}\mathbf{y}^T$, where $\mathbf{x}$ and $\mathbf{y}$ are linear combinations of $\mathbf{l}_1$

and $\mathbf{z}$ which satisfy $x_1 = 1$ and $y_1 = 0$. The last condition is fulfilled directly using $\mathbf{y} = \mathbf{z} - z_1\mathbf{l}_1$ and $\mathbf{x}$ can therefore be written generally as $\mathbf{x} = \mathbf{l}_1 + \beta\mathbf{y}$. It follows on equating terms that $\lambda = d_1 + \sigma z_1^2$, $\beta = \sigma z_1/\lambda$ and $\mu = \sigma - \lambda\beta^2$. Because $y_1 = 0$, $\lambda\mathbf{x}\mathbf{x}^T$ can be identified as $\tilde{d}_1\tilde{\mathbf{l}}_1\tilde{\mathbf{l}}_1^T$. Thus, the first column of the modified factorization has been found, and the problem reduced to that of updating $\Sigma_{i=2}^n d_i\mathbf{l}_i\mathbf{l}_i^T$ by the rank one matrix $\mu\mathbf{y}\mathbf{y}^T$. Both these matrices have a zero first row and column, so the problem has been reduced to one in $n - 1$ variables. Repetition of this process enables the complete factorization of $\tilde{A}$ to be determined.

If $\sigma_i\mathbf{z}^{(i)}\mathbf{z}^{(i)T}$ denotes the remaining rank one matrix when $\Sigma_{j=1}^{i-1}\tilde{d}_j\tilde{\mathbf{l}}_j\tilde{\mathbf{l}}_j^T$ has been determined (defining $\sigma_1 = \sigma$, $\mathbf{z}^{(1)} = \mathbf{z}$ and noting that $z_j^{(i)} = 0$ for all $j < i$), then the basic step of the algorithm can be expressed by

$$\tilde{\mathbf{l}}_i\tilde{d}_i\tilde{\mathbf{l}}_i^T + \sigma_{i+1}\,\mathbf{z}^{(i+1)}\,\mathbf{z}^{(i+1)T} = \mathbf{l}_i\,d_i\,\mathbf{l}_i^T + \sigma_i\,\mathbf{z}^{(i)}\,\mathbf{z}^{(i)T}. \tag{2.1}$$

This basic step involves the operations

$$v_i = z_i^{(i)}, \tag{2.2a}$$

$$\tilde{d}_i = d_i + \sigma_i v_i^2 \tag{2.2b}$$

terminate if $i = n$,

$$\beta_i = \sigma_i v_i/\tilde{d}_i, \tag{2.2c}$$

$$\mathbf{z}^{(i+1)} = \mathbf{z}^{(i)} - v_i\mathbf{l}_i, \tag{2.2d}$$

$$\tilde{\mathbf{l}}_i = \mathbf{l}_i + \beta_i\,\mathbf{z}^{(i+1)}, \tag{2.2e}$$

$$\sigma_{i+1} = \sigma_i - \tilde{d}_i\beta_i^2, \tag{2.2f}$$

and the complete algorithm requires this sequence to be repeated for $i = 1, 2, \cdots, n$. This process is the specialization to (1.2) of the algorithm given by Bennett [1] for updating the factors of a general matrix on adding a matrix of low rank, except that Bennett writes $q_i = p_i = v_i\sigma_i$ and uses the equation $\sigma_{i+1} = \sigma_i - p_iq_i/\tilde{d}_i$ in place of the equivalent equation (2.2f). In the form (2.2), the complete updating process requires $n^2 + O(n)$ multiplications and no extra storage, assuming that L and D are overwritten by $\tilde{L}$ and $\tilde{D}$, and that $\mathbf{z}$ is overwritten by the successive $\mathbf{z}^{(i)}$.

A similar process has been described recently by Gentleman [3] in the slightly different context of factorizing the normal matrix of a linear least squares calculation. In this case, $\sigma > 0$, but the method may also be applied to the problem (1.2) for any value of σ for which the calculated $\tilde{A}$ is positive definite. It follows from (2.2f), (2.2c) and (2.2b) that

$$\sigma_{i+1} = \sigma_i d_i/\tilde{d}_i \tag{2.3}$$

and this equation is used by Gentleman in place of (2.2f). It also follows from (2.2e) and (2.2d), after some manipulation using (2.2c) and (2.2b), that

$$\tilde{\mathbf{l}}_i = \mathbf{l}_i d_i/\tilde{d}_i + \beta_i \mathbf{z}^{(i)}. \tag{2.4}$$

This equation is preferred in place of (2.2e) in some circumstances by Gentleman. It has the disadvantage that the calculation of $\tilde{\mathbf{l}}_i$ requires an extra $i - 1$ multiplications as against (2.2e). On the other hand, there are also features in favour of (2.4) which will be further discussed in Section 3.

Another equivalent method, which to our knowledge has not previously been reported, is one in which the Bennett/Gentleman approach is implemented in terms of $t_i = \sigma_i^{-1}$ rather than σ_i. We shall call this method the *simple t-method*; it consists of recurring for $i = 1, 2, \cdots, n$ the following sequence of operations

$$v_i = z_i^{(i)}, \tag{2.5a}$$

$$t_{i+1} = t_i + v_i^2/d_i, \tag{2.5b}$$

$$\tilde{d}_i = d_i\, t_{i+1}/t_i \tag{2.5c}$$

$$\text{terminate if } i = n,$$

$$\beta_i = (v_i/d_i)/t_{i+1}, \tag{2.5d}$$

$$\mathbf{z}^{(i+1)} = \mathbf{z}^{(i)} - v_i \mathbf{l}_i, \tag{2.5e}$$

$$\tilde{\mathbf{l}}_i = \mathbf{l}_i + \beta_i \mathbf{z}^{(i+1)}, \tag{2.5f}$$

where $t_1 = 1/\sigma$ and $\mathbf{z}^{(1)} = \mathbf{z}$. Equation (2.5c) is a rearrangement of (2.3) which is used here to compute $\tilde{d}_i$ rather than σ_{i+1}. Equation (2.5b) is obtained by rearranging (2.2b) and (2.5c). Equation (2.5d) for calculating β_i is preferred to (2.2c) because it saves a multiplication and also gives slightly better error propagation. An error analysis of the simple t-method and variations thereof is given in Section 5, and a worked example in Section 6.

Two methods, which are at first sight different, are proposed by Gill and Murray [5]. In their *Method* B, Eq. (1.2) is written

$$\tilde{A} = L(D + \sigma\mathbf{v}\mathbf{v}^T)L^T \tag{2.6}$$

where $\mathbf{v}$ is determined by back substitution in the equations

$$L\mathbf{v} = \mathbf{z}. \tag{2.7}$$

Gill and Murray point out that the factorization of $D + \sigma\mathbf{v}\mathbf{v}^T$ can be written

$$D + \sigma\mathbf{v}\mathbf{v}^T = \bar{L}\tilde{D}\bar{L}^T \tag{2.8}$$

where $\tilde{D}$ is the diagonal matrix in (1.3), and $\bar{L}$ is the unit lower triangular matrix whose elements below the diagonal have the values

(2.9) $$\bar{l}_{ij} = v_i\beta_j, \quad j < i.$$

The quantities $\tilde{D}$ and $\beta_1, \beta_2, \cdots, \beta_{n-1}$ can be determined from D, σ and $\mathbf{v}$ in $O(n)$ multiplications. Gill and Murray also show that the special form of $\bar{L}$ can be exploited to calculate the product $\tilde{L} = L\bar{L}$ in $n^2 + O(n)$ multiplications. Thus, the modified factorization (1.3) can be determined by first calculating $\mathbf{v}$ from (2.7), then forming $\tilde{D}$ and $\beta_1, \cdots, \beta_{n-1}$, and finally calculating $\tilde{L} = L\bar{L}$. This method requires a total of $3n^2/2 + O(n)$ multiplications.

In fact, Gill and Murray's Method B and the Bennett method (2.2) are closely related. In particular, the quantities v_i and β_i occurring in (2.2) are the same as the v_i and β_i occurring in (2.9). Furthermore, the back substitution process (2.7) can be written

(2.10) $$\left.\begin{aligned} v_i &= z_i^{(i)} \\ \mathbf{z}^{(i+1)} &= \mathbf{z}^{(i)} - v_i\mathbf{l}_i \end{aligned}\right\}, \quad i = 1, 2, \cdots, n,$$

which are just the equations (2.2a) and (2.2d). These equations occur naturally in Gill and Murray's recurrence relation for calculating $\tilde{L} = L\bar{L}$. The Bennett method (2.2) achieves its efficiency relative to the Gill and Murray method by exploiting the equivalence between these two processes, and running the back substitution in parallel with the computation of $\tilde{D}$ and $\tilde{L}$.

Gill and Murray also suggest an alternative *Method* A for computing $\tilde{D}$ and β. Because this method is much more complicated than that used in Method B, it will not be described in detail. However, the philosophy behind the method is important and we shall consider this further in Section 3. Details of all the methods described in the present section (except the simple t-method) can also be found in Gill et al. [4] with some additional less efficient methods for updating the LDL^T factorization.

3. Choosing a Method.

In this section, a number of important considerations which govern the choice of method will be discussed. The aim is to give a method which is applicable either when $\sigma > 0$ or $\sigma < 0$, which is stable, which guarantees the positive definiteness of $\tilde{A}$, and which is most efficient from time and storage considerations, subject to these conditions.

The first problem which we shall discuss is that of maintaining positive definiteness of $\tilde{A}$. It is often the case that theoretical considerations indicate that $\tilde{A}$ must be positive definite, apart from the effects of round-off error. This is always so when $\sigma > 0$, but is also often the case when $\sigma < 0$. It is imperative that this property be maintained in the presence of round-off error. Now, the Bennett algorithm (2.2) can give rise to negative $\tilde{d}_i$ (and hence indefinite $\tilde{A}$), either in Eq. (2.2b) when $\sigma < 0$, or due to a change of sign in σ_{i+1} in (2.2f) when $\sigma > 0$. The Gentleman algorithm avoids the latter possibility through the use of (2.3) for calculating σ_{i+1}, but the former still applies. The simple t-method can lose positive definiteness when $\sigma < 0$ due to t_{i+1} becoming positive in (2.5b).

We shall now consider how the simple t-method might be modified to deal with the case where $\sigma < 0$ but t_{i+1} becomes positive, although $\tilde{A}$ is known to be positive definite from theoretical considerations. Similar comments will apply to the Bennett and Gentleman methods.

We note that (2.3) and the definition of $t_i = \sigma_i^{-1}$ give

$$\frac{\sigma_1}{\sigma_{i+1}} = \frac{t_{i+1}}{t_1} = \prod_{j=1}^{i} \tilde{d}_j \Big/ \prod_{j=1}^{i} d_j$$

and hence

$$\frac{\sigma_1}{\sigma_{n+1}} = \frac{t_{n+1}}{t_1} = \frac{\det \tilde{D}}{\det D} = \frac{\det \tilde{A}}{\det A}.$$

These equations not only give insight into the role played by the σ_i or t_i, but indicate particularly the need to keep the signs of the t_i or σ_i constant, in the presence of the round-off error.

One obvious way of keeping $\tilde{A}$ positive definite with the simple t-method is to check whether t_{i+1} is negative after using (2.5b), and if not, to replace it by some suitable negative value, for instance t_i. However, usually by the time a nonpositive t_{i+1} is found, a substantial part of the factorization of $\tilde{A}$ will already have been calculated; the following simple example shows that catastrophic error growth may have taken place.

Let A be diagonal so that $L = I$ and $D = \text{diag}(10^{-6}/(1 - 10^{-6}), 1/(1 - 10^{-6}), 1)$. Let $\sigma = -1$ and $\mathbf{z} = \mathbf{v} = (.001, .001, .001)$. Although $\tilde{A}$ for this example is not positive definite, a perturbation to σ such that $\sigma = -1 + 10^{-6}$ is enough to make $\tilde{A}$ positive definite. If 10^{-6} is regarded as the relative machine precision, then this example can be regarded as one for which $\tilde{A}$ is positive definite to machine precision, and is therefore one for which the algorithm ought to work. In fact, in exact arithmetic, the algorithm gives the values $t_1 = -1$, $t_2 = -10^{-6}$, $t_3 = -10^{-12}$, $t_4 = 10^{-6} - 10^{-12}$. Because $t_4 > 0$, it is replaced by a negative value and this value does not affect what follows. Consider the calculation of $\tilde{\mathbf{l}}_2$. The value of β_2 is $(v_2/d_2)/t_3 \sim -10^9$ and hence $\tilde{l}_{32} \sim -10^6$. Because $\tilde{d}_2 \sim 10^{-6}, \tilde{\mathbf{l}}_2 \tilde{d}_2 \tilde{\mathbf{l}}_2^T$ gives a contribution to $\tilde{A}_{33}$ of $\sim +10^6$. Yet, the value of $\tilde{A}_{33}$ in the perturbed problem with $\sigma = -1 + 10^{-6}$ is $\tilde{A}_{33} \sim 1$. Thus substantial error growth takes place, before the loss of positive definiteness is recognized.

It is desirable to find a technique for overcoming this deficiency which introduces a time penalty only when the value of t_{i+1} indicates that positive definiteness has been lost, because these occasions are rare. For instance, one approach which avoids using substantial extra storage is to store the β_i and the t_i, and if a $t_{i+1} > 0$ is encountered, then the original problem can be recovered by running all the recurrences in (2.5) backward for values $i, i-1, \cdots, 1$. However, the example of the last paragraph shows that large numbers can arise in $\tilde{\mathbf{l}}_j$, $j < i$ when $t_{i+1} > 0$, which tends to give large errors in the recomputed $\mathbf{l}_j$. Therefore, we have not considered this approach any further.

It seems therefore that two alternatives remain. One is to accept the inefficiency of always calculating the vector $\mathbf{v}$ initially so that the sign of t_{n+1} can be checked before overwriting the original matrices L and D. If $t_{n+1} > 0$, then the problem must be modified, a subject to which we will return shortly. This is the philosophy behind Gill and Murray's Method A. The other alternative is always to make available $\sim \frac{1}{2}n^2$ extra storage locations, which almost doubles the storage space required for the numbers of the calculation. Various approaches are then possible, the most simple of which is to store the original problem so that it can be recovered exactly if a $t_{i+1} > 0$ is encountered. However, we have decided to accept the time penalty of calculating the vector $\mathbf{v}$ initially, rather than the storage penalty. One reason for this decision is that in many applications (including those to quasi-Newton methods for minimization), the vector $\mathbf{v}$ is already available at the start of the updating calculation. This can arise in at least two ways. It may be that the vector $A^{-1}\mathbf{z}$ has already been calculated, whence $\mathbf{v}$ is available because the first step in calculating $A^{-1}\mathbf{z}$ is to back substitute in the equations $L\mathbf{v} = \mathbf{z}$. Alternatively, $\mathbf{z}$ may have been calculated as $\mathbf{z} = A\mathbf{u}$ where $\mathbf{u}$ is some arbitrary vector, in which case the intermediate quantity $DL^T\mathbf{u}$ is the required vector $\mathbf{v}$.

Therefore, we now assume that the vector $\mathbf{v}$ is available at the start of the algorithm when $\sigma < 0$, and we consider what is the best way to modify the problem if a $t_{i+1} > 0$ is obtained (or a $\tilde{d}_i < 0$ in the Bennett/Gentleman method). We shall also consider whether the simple t-method or the Bennett/Gentleman method is more suitable in these circumstances. An important fact to realize is that when a $t_{n+1} > 0$ is calculated, then, because of the assumption that theoretically $\tilde{A}$ is positive definite, the correct value of t_{n+1} should be negative but of a magnitude comparable to the round-off errors in the calculation. A satisfactory procedure is to recalculate the t_i using

(3.1) $$t_{n+1} = \epsilon/\sigma,$$

(3.2) $$t_i = t_{i+1} - v_i^2/d_i, \quad i = n, n-1, \cdots, 1,$$

where ϵ is the relative machine precision. Thus, σ is replaced by the new value of t_1^{-1} which gives a problem that is close to the original problem by our assumptions on $\tilde{A}$. These new values of t_i are then used in the recursion (2.5). The error analysis for the quantities t_i calculated by (3.2) is almost the same as that for (2.5b); a similar result to that of Section 5 applies to the modified problem.

A similar approach is possible with the Gentleman method. If a $\tilde{d}_i < 0$ is detected, then, although it is not possible to recur Eqs. (2.2b) and (2.3) in the opposite direction, it is possible to use (3.1) and (3.2) as before, additionally calculating $\tilde{d}_i$ from $\tilde{d}_i = d_i\, t_{i+1}/t_i$. However, there are a number of minor disadvantages to this. One is that the Bennett/Gentleman approach overwrites the d_i by $\tilde{d}_i$. Thus, an extra vector of storage must be made available in case the t_i and $\tilde{d}_i$ have to be recalculated. Another is that to calculate $\tilde{d}_n$ by the Gentleman approach requires about $5n$ arithmetic operations, whereas to calculate t_{n+1} by the t-method requires only $3n$. Thus, not only are there $2n$ wasted operations, but the necessity for fewer operations implies that there is less likelihood of $t_{n+1} > 0$ being obtained rather than $\tilde{d}_n < 0$ (assuming that the values obtained in exact arithmetic would have had the correct sign). In fact, the error analysis in Section 5 indicates that, even in cases when the calculation does not break down on this account, and whether or not $\sigma < 0$, the Bennett/Gentleman method has a term which increases as $5n$ in the error bound, whereas the t-method has a corresponding term which increases only as $3n$. (Compare bounds (5.28), (5.29) and (5.30) in Section 5.) This term ultimately dominates for large n, and so it provides a further reason in favour of the t-method. These considerations have led us to prefer the t-method and we shall not consider the Bennett/Gentleman method for calculating $\tilde{D}$ and the σ_i any further. The arguments which have led us to this decision are partly those which influenced Gill and Murray [4] in preferring their Method A. However, Method A is less efficient than the t-method and does not seem to have any corresponding advantages.

Finally, we have to decide whether to choose Eq. (2.5f) or Eq. (2.4) to calculate $\tilde{\mathbf{l}}_i$. As we pointed out above, (2.5f) is cheaper, but the following remark favours Eq. (2.4). Let (2.5f) be written

$$\tilde{\mathbf{l}}_i = \mathbf{l}_i + \beta_i \mathbf{z}^{(i+1)} = (1 - v_i\beta_i)\,\mathbf{l}_i + \beta_i \mathbf{z}^{(i)} \tag{3.3}$$

by virtue of (2.5e). Then, if $1 - v_i\beta_i = d_i/\tilde{d}_i$ is very small, as may happen when $\sigma > 0$, then Eq. (3.3) shows that to use (2.5f) might cause cancellation between the $\mathbf{l}_i$ term and the $v_i\beta_i\mathbf{l}_i$ term implicit in $\beta_i\mathbf{z}^{(i+1)}$. An error analysis for the linear least squares calculation using (2.5f) has been given by Gentleman [3] and shows a factor $\sqrt{\tilde{d}_i/d_i}$ in the error bound. The bounds given in Section 5 for the simple t-method also contain the factor $\sqrt{\tilde{d}_i/d_i}$. The worked example of Section 6 shows how error growth of this magnitude is achieved in practice. If (2.4) is used in place of (2.5f), the factor $\sqrt{d_i/\tilde{d}_i}$ appears instead. Thus, the most obvious approach to minimize the total error bound is to use (2.5f) when $\sigma < 0$ and (2.4) when $\sigma > 0$. This method requires $3n^2/2 + O(n)$ multiplications overall when $\sigma > 0$.

A more attractive alternative is available, however, which reduces the total number of multiplications required when $\sigma > 0$. It is to use (2.5f), unless $\sqrt{\tilde{d}_i/d_i}$ is rather large ($\sqrt{\tilde{d}_i/d_i} > K$, say). Gentleman suggests this idea and uses the value $K = 10$. Ideally, K should be as large as possible, subject to the contribution from $\sqrt{\tilde{d}_i/d_i}$ to the error not becoming dominant. In our error analysis, the coefficients

13, 4 and 6 $\sqrt{\tilde{d}_i/d_i}$ appear in (5.16) and, in the total error bound, the terms $17 + 12\sqrt{\tilde{d}_i/d_i}$ occur. This suggests that $K = 10$ is rather on the large side. Therefore, we use Eq. (2.5f) instead of Eq. (2.4) only when $\sqrt{\tilde{d}_i/d_i} \leqslant 2$. We call the resulting method the *composite t-method.* We monitored the value of $\tilde{d}_i/d_i$ on a large number of applications of the composite t-method and find that (2.4) is used on fewer than $1/n$ of the total number of occasions. Hence, to all intents and purposes, the composite t-method can be regarded as one which takes $n^2 + O(n)$ multiplications when $\sigma > 0$, which is the advantage claimed for the simple t-method.

In practice, Eq. (2.4) is applied in the form

$$\tilde{\mathbf{l}}_i = \gamma_i \mathbf{l}_i + \beta_i \mathbf{z}^{(i)} \tag{3.4}$$

where $\gamma_i = d_i/\tilde{d}_i = t_i/t_{i+1}$. We prefer not to use the value of γ_i in the calculation of $\tilde{d}_i$ (see Eq. (2.5c)), for, if we did, then *every* iteration would require an additional division to apply (2.5c) rather than a multiplication. Nor do we use the value of α_i to calculate the term $\mathbf{l}_i d_i/\tilde{d}_i$ in (2.4) by $\mathbf{l}_i/\alpha_i$, because this again requires $n - i$ divisions on those occasions when (2.4) is used. Instead, we use multiplications in both calculations, for the price of wasting one operation on those few occasions when the ratio $\tilde{d}_i/d_i$ is so large that Eq. (3.4) has to be used.

In fact, the bounds given in Section 5 for the simple t-method sometimes exaggerate the dangers of using (2.5f) all the time, and indeed it has been observed in practice that methods using (2.5f) do not necessarily give error growth when $\tilde{d}_i/d_i$ is large (W. Murray—private communication). Reference to Eq. (3.3) indicates that even when $d_i/\tilde{d}_i$ is small, serious error growth can occur only if, in addition, an element l_{ji} is large for some $j > i$. The error analysis in Section 5 is extended to show this effect, which is also illustrated by the worked example of Section 6. However the error analysis does not lead to a more convenient computational scheme, because the test which it suggests would be expensive to calculate.

To summarize what we think to be the best method, we outline the steps in the composite t-method, including the modifications to ensure that $\tilde{A}$ remains positive definite. The inefficiency introduced by having the same code to cater for both $\sigma < 0$ and $\sigma > 0$ is negligible.

The *composite t-method*: terminate if $\sigma = 0$: Let $t_1 = \sigma^{-1}$ and $\mathbf{z}^{(1)} = \mathbf{z}$: then

(i) if $\sigma > 0$ go to (v),
(ii) if $\mathbf{v}$ is not available, solve $L\mathbf{v} = \mathbf{z}$ for $\mathbf{v}$,
(iii) compute $t_{i+1} = t_i + v_i^2/d_i$, $i = 1, 2, \cdots, n$,
(iv) if any $t_{i+1} \geqslant 0$, recompute the t_i from (3.1) and (3.2),
(v) repeat for $i = 1, 2, \cdots, n$ the following sequence

```
$v_i = z_i^{(i)}$,
if $\sigma > 0$ then $t_{i+1} = t_i + v_i^2/d_i$,
$\alpha_i = t_{i+1}/t_i$,
$\tilde{d}_i = d_i\alpha_i$,
terminate if $i = n$,
$\beta_i = (v_i/d_i)/t_{i+1}$,
$\mathbf{z}^{(i+1)} = \mathbf{z}^{(i)} - v_i\mathbf{l}_i$,
if $\alpha_i > 4$ then $\gamma_i = t_i/t_{i+1}$ and $\tilde{\mathbf{l}}_i = \gamma_i\mathbf{l}_i + \beta_i\mathbf{z}^{(i)}$,
        else $\tilde{\mathbf{l}}_i = \mathbf{l}_i + \beta_i\mathbf{z}^{(i+1)}$.
```

A FORTRAN subroutine, code MC11A, has been written for this process and is available through the Harwell Subroutine Library. It uses only $\sim \frac{1}{2}n^2$ storage locations, L and D being stored in a one dimensional form. Additional entry points are available, which could easily be separated off as subroutines, to calculate $A\mathbf{u}$ and $A^{-1}\mathbf{u}$ when L and D are represented in this special form. Facility is provided for making the vector $\mathbf{v}$ available to the updating routine. Entry points are also given to factorize an arbitrary positive definite matrix, and to multiply out the factors either to give A or A^{-1}.

4. Semidefinite Matrices.

It is also possible to update positive semidefinite matrices using some of the methods described in Sections 2 and 3, and, in particular, the t-methods are suitable. Indeed, semidefiniteness gives some advantages that are discussed later in this section. However, special action may have to be taken when any $d_i = 0$ or $\tilde{d}_i = 0$. There are three cases.

Firstly, the case when the rank of A is expected to remain unchanged is considered. In this case, $\mathbf{z}$ is in the column space of A and so can be written

$$\mathbf{z} = A\mathbf{u} = LDL^T\mathbf{u}. \tag{4.1}$$

If we define $N \equiv \{i\colon d_i = 0\}$ and denote $DL^T\mathbf{u}$ by $\mathbf{w}$, then $\mathbf{w}_i = 0$ $\forall i \in N$. But $\mathbf{v}$ is defined by $\mathbf{z} = L\mathbf{v}$ so $\mathbf{v} = \mathbf{w}$. Hence, $v_i = 0$ $\forall i \in N$. Thus, if $i \in N$, then, by virtue of (2.2b) and (2.1), $\tilde{d}_i = d_i = 0$, and so $\mathbf{z}^{(i+1)} = \mathbf{z}^{(i)}$ and $\tilde{\mathbf{l}}_i$ is arbitrary. Thus, the only action which needs to be taken is to set $v_i = 0$ and $t_{i+1} = t_i$. If a value of $t_{i+1} \geqslant 0$ is calculated when $\sigma < 0$, then the recalculation of the t_i based on (3.1) and (3.2) can be carried out as before, except that we set $t_i = t_{i+1}$ if $i \in N$.

In the second case, the rank of A is expected to decrease by one. In this situation, $\sigma < 0$ and it is known that $t_{n+1} = 0$. A typical instance would be the operation

$$\tilde{A} = A - A\mathbf{u}\mathbf{u}^T A/\mathbf{u}^T A\mathbf{u}$$

for arbitrary $\mathbf{u}$. If the simple t-method (2.5) were applied to such problems, it is unlikely that a t_{n+1} would be obtained which was exactly zero. In these circumstances, it is appropriate to set $t_{n+1} = 0$ and to use the recurıence (3.2) to calculate the t_i, setting $v_i = 0$, $t_i = t_{i+1}$, $i \in N$. The ability to use the code written for (3.2) is very convenient. Note that if $N \supseteq \{p, p+1, \cdots, n\}$ then $v_p = v_{p+1} = \cdots = v_n = 0$ and hence $t_p = t_{p+1} = \cdots = t_n = t_{n+1} = 0$. It is therefore important to avoid calculating t_{p+1}/t_p in such circumstances. In fact, when the value of i is such that $t_{i+1} = 0$, then, after setting $\tilde{d}_i = 0$, the whole process can be terminated.

In the third case, the rank of A is expected to increase by one. In this case, σ is always positive and the increase in rank occurs when the composite t-method finds $i \in N$ and $v_i \neq 0$. Then, from (2.2b), $\tilde{d}_i = \sigma_i v_i^2 = v_i^2/t_i$ and $\tilde{\mathbf{l}}_i = \mathbf{z}^{(i)}/v_i$ by virtue of (2.4). The iteration is terminated after setting $\tilde{\mathbf{l}}_j = \mathbf{l}_j$ and $\tilde{d}_j = d_j$ for all $j > i$. If $v_i = 0$, however, the action appropriate to the first case is taken, and the iteration is continued. If $v_i = 0$ for all $i \in N$, then the rank of A does not increase by one as expected.

One advantage to be gained by representing positive semidefinite matrices in this way is that multiplication by A is quicker because only those terms $\mathbf{l}_i d_i \mathbf{l}_i^T$, $i \notin N$ need be considered. A more important advantage is that the factorized form of A has the correct rank imposed upon it, whereas if A were stored explicitly, rounding errors would make the rank of A indeterminate. A similar advantage is that when updating A knowing that the rank of A should remain the same, then the operation of setting $v_i = 0$ for $i \in N$ annihilates some of the rounding errors incurred in forming $\mathbf{z}$ in (4.1) which would otherwise cause $\mathbf{z}$ not to lie in the column space of A.

These modifications for singular matrices are relevant to numerical methods such as gradient projection (Rosen [7]) and Goldfarb's method [6] for minimizing functions of many variables subject to linear constraints. One of the practical difficulties with Goldfarb's method is caused by loss of accuracy due to errors in a supposedly positive semidefinite matrix, and it may be worth investigating whether representing matrices in the way described here is preferable. Facilities for dealing with semidefinite matrices as described in this section are incorporated into the subroutine MC11A.

5. An Error Analysis.

A computer error analysis of the updating processes of the t-methods will now be given. The ith basic step of the methods, corresponding exactly to Eq. (2.1), is

$$\mathbf{l}^* d^* \mathbf{l}^{*T} + \mathbf{z}^* \mathbf{z}^{*T}/t^* = \mathbf{l} d \mathbf{l}^T + \mathbf{z}\mathbf{z}^T/t \tag{5.1}$$

where asterisks denote quantities which must be evaluated on the ith step. Computed quantities will be denoted by bars, and the first step will be to bound the error $E^{(i)}$ in carrying out (5.1), defined by

$$\bar{\mathbf{l}}^{*}\bar{d}^{*}\bar{\mathbf{l}}^{*T}+\bar{\mathbf{z}}^{*}\bar{\mathbf{z}}^{*T}/\bar{t}^{*}=\mathbf{l}d\mathbf{l}^{T}+\mathbf{z}\mathbf{z}^{T}/t+E^{(i)}. \tag{5.2}$$

The $\bar{\mathbf{z}}^{*}$ and $\bar{t}^{*}$ from one basic step are the exact data $\mathbf{z}$ and t for the next basic step. It therefore follows that the complete computed factorization is an exact factorization of a perturbation of the original problem given by

$$\tilde{A}=\tilde{L}\tilde{D}\tilde{L}^{T}=LDL^{T}+\sigma\mathbf{z}\mathbf{z}^{T}+E \tag{5.3}$$

where $E=\Sigma_{i}E^{(i)}$. Bounds on E will be derived. Note that the matrix $\tilde{A}$ is now a *computed* matrix, instead of the exact matrix of (1.2).

The equation $a=(b*c)(1+e)$ where $|e|\leqslant\epsilon$, see Wilkinson [8], will form the basis of the analysis, where $*$ represents any of $+,-,x,/$, where a is the result of computing and rounding $b*c$ in floating-point arithmetic, and where ϵ is the relative machine precision (2^{-p} in many cases).

In the simple t-method (see Eqs. (2.5)) the following computations are made

$$\begin{aligned}
&\bar{t}^{*}=(1+e_{1})(t+(1+e_{2})(1+e_{3})v^{2}/d),\\
&\bar{d}^{*}=(1+e_{4})(1+e_{5})d\bar{t}^{*}/t,\\
&\bar{\beta}=(1+e_{2})(1+e_{6})(v/d)/\bar{t}^{*}\\
&\left.\begin{aligned}&\bar{z}_{j}^{*}=(1+e_{1j})(z_{j}-(1+e_{2j})vl_{j})\\ &\bar{l}_{j}^{*}=(1+e_{3j})(l_{j}+(1+e_{4j})\bar{\beta}\bar{z}_{j}^{*})\end{aligned}\right\},\quad j>i,
\end{aligned} \tag{5.4}$$

where $e_{1},e_{2},\cdots$ and $e_{1j},e_{2j},\cdots$ each represent separate errors bounded by ϵ. The analysis is simplified if the intermediate hatted quantities

$$\begin{aligned}
\hat{v}&=v,\\
\hat{t}&=t,\\
\hat{d}&=d/((1+e_{2})(1+e_{3})),\\
\hat{z}_{j}&=z_{j},\\
\hat{l}_{j}&=(1+e_{2j})l_{j},
\end{aligned} \tag{5.5}$$

are introduced. Also if these quantities are substituted into the right-hand sides of Eqs. (2.5), then we can define $\hat{t}^{*},\hat{d}^{*},\hat{\beta},\hat{\mathbf{z}}^{*}$ and $\hat{\mathbf{l}}^{*}$ as being the resulting left-hand sides without error. Then it follows from (5.2) that $E^{(i)}$ is the matrix

$$E^{(i)}=(\bar{\mathbf{l}}^{*}\bar{d}^{*}\bar{\mathbf{l}}^{*T}-\hat{\mathbf{l}}^{*}\hat{d}^{*}\hat{\mathbf{l}}^{*T})+(\bar{\mathbf{z}}^{*}\bar{\mathbf{z}}^{*T}/\bar{t}^{*}-\hat{\mathbf{z}}^{*}\hat{\mathbf{z}}^{*T}/\hat{t}^{*})+(\hat{\mathbf{l}}\hat{d}\hat{\mathbf{l}}^{T}-\mathbf{l}d\mathbf{l}^{T}). \tag{5.6}$$

Now, (5.5) gives the equation

$$\hat{l}_j\hat{d}\hat{l}_k - l_j d l_k = [(1 + e_{2j})(1 + e_{2k})(1 + e_2)^{-1}(1 + e_3)^{-1} - 1]l_j d l_k$$
$$= (e_{2j} + e_{2k} - e_2 - e_3 + O(\epsilon^2))l_j d l_k, \quad j > i,\ k > i, \tag{5.7}$$

where the $O(\epsilon^2)$ indicates a term that is bounded by a small multiple of ϵ^2. It also follows from (5.5) that

$$\bar{t}^* = \hat{t}^*(1 + e_1),$$
$$\bar{z}_j^* = \hat{z}_j^*(1 + e_{1j}), \quad j > i, \tag{5.8}$$

and hence that

$$\bar{z}_j^*\bar{z}_k^*/\bar{t}^* - \hat{z}_j^*\hat{z}_k^*/\hat{t}^* = (-e_1 + e_{1j} + e_{1k} + O(\epsilon^2))\bar{z}_j^*\bar{z}_k^*/\bar{t}^*. \tag{5.9}$$

The definitions of $\hat{\beta}$, $\hat{d}$ and $\bar{\beta}$ imply

$$\hat{\beta} = \bar{\beta}(1 + e_1)(1 + e_3)/(1 + e_6) \tag{5.10}$$

so that from the definitions of $\hat{l}_j^*$ and $\bar{l}_j^*$, and from the second part of (5.8), there follows

$$\hat{l}_j^* - \bar{l}_j^* = [e_{2j} - e_{3j}]\, l_j$$
$$+ [(1 + e_1)(1 + e_3)(1 + e_6)^{-1}(1 + e_{1j})^{-1} - (1 + e_{3j})(1 + e_{4j})]\,\bar{\beta}\bar{z}_j^*. \tag{5.11}$$

Eliminating $\bar{\beta}\bar{z}_j^*$ using the last equation of (5.4) then gives the result

$$\hat{l}_j^* = \bar{l}_j^*(1 + e_1 + e_3 - e_6 - e_{1j} - e_{3j} - e_{4j} + O(\epsilon^2))$$
$$+ l_j(-e_1 - e_3 + e_6 + e_{1j} + e_{2j} + e_{4j} + O(\epsilon^2)). \tag{5.12}$$

Furthermore, by definition of $\hat{d}^*$ and the first part of (5.8),

$$\bar{d}^* = (1 + e_1)(1 + e_2)(1 + e_3)(1 + e_4)(1 + e_5)\hat{d}^* \tag{5.13}$$

so subsituting from (5.12) and (5.13), it follows that

$$\bar{l}_j^*\bar{d}^*\bar{l}_k^* - \hat{l}_j^*\hat{d}^*\hat{l}_k^*$$
$$= \bar{l}_j^*\bar{d}^*\bar{l}_k^*(-e_1 + e_2 - e_3 + e_4 + e_5 + 2e_6$$
$$+ e_{1j} + e_{3j} + e_{4j} + e_{1k} + e_{3k} + e_{4k} + O(\epsilon^2))$$
$$+ l_j\bar{d}^*\bar{l}_k^*(e_1 + e_3 - e_6 - e_{1j} - e_{2j} - e_{4j} + O(\epsilon^2))$$
$$+ \bar{l}_j^*\bar{d}^*l_k(e_1 + e_3 - e_6 - e_{1k} - e_{2k} - e_{4k} + O(\epsilon^2))$$
$$+ l_j\bar{d}^*l_k(O(\epsilon^2)), \quad j > i,\ k > i. \tag{5.14}$$

Then (5.6), (5.7), (5.9) and (5.14) give an expression for $E_{jk}^{(i)}$, $j > i$, $k > i$, namely

$$
\begin{aligned}
E_{jk}^{(i)} = {}& \bar{l}_j^* \bar{d}^* \bar{l}_k^* (-e_1 + e_2 - e_3 + e_4 + e_5 + 2e_6 \\
&\qquad + e_{1j} + e_{3j} + e_{4j} + e_{1k} + e_{3k} + e_{4k} + O(\epsilon^2)) \\
&+ l_j \bar{d}^* \bar{l}_k^* (e_1 + e_3 - e_6 - e_{1j} - e_{2j} - e_{4j} + O(\epsilon^2)) \\
&+ \bar{l}_j^* \bar{d}^* l_k (e_1 + e_3 - e_6 - e_{1k} - e_{2k} - e_{4k} + O(\epsilon^2)) \\
&+ l_j d l_k (-e_2 - e_3 + e_{2j} + e_{2k} + O(\epsilon^2) + O(\epsilon^2)\bar{d}^*/d) \\
&+ \bar{z}_j^* \bar{z}_k^* / \bar{t}^* (-e_1 + e_{1j} + e_{1k} + O(\epsilon^2)).
\end{aligned} \tag{5.15}
$$

More simple expressions obtain for $E_{ij}^{(i)} = E_{ji}^{(i)}$, $j > i$, and for $E_{ii}^{(i)}$, which can be covered by allowing (5.16) to apply for all $j \geqslant i$, $k \geqslant i$. Of course, $E_{jk}^{(i)} = 0$ if either $j < i$ or $k < i$.

We shall now proceed to obtain a bound on $E_{jk}^{(i)}$. Because of the way in which the $O(\epsilon^2)$ terms occur, it is clear that, except perhaps for the $O(\epsilon^2)\bar{d}^*/d$ term, these terms will only make small relative perturbations to the coefficients in the final result; with this in mind, such terms will henceforth be neglected. Then (5.15) yields

$$
\begin{aligned}
|E_{jk}^{(i)}| \leqslant \epsilon \{ & 3|\bar{z}_j^* \bar{z}_k^* / \bar{t}^*| + 13 a_j a_k + (4 + O(\epsilon)\bar{d}^*/d) b_j b_k \\
& + 6\sqrt{\bar{d}^*/d}(a_j b_k + b_j a_k)\}
\end{aligned} \tag{5.16}
$$

where $a_j = \bar{d}^{*1/2} |\bar{l}_j^*|$ and $b_j = d^{1/2} |l_j|$. Now a_j and b_j are just elements in the Choleski (LL^T) factors of $\bar{A}$ and A, respectively, so (5.16) indicates correctly that unacceptably large errors may occur only when $\sqrt{\bar{d}^*/d}$ is large, which happens only when $\sigma > 0$.

A similar analysis can be performed when Eq. (3.4) is used to calculate $\bar{\mathbf{l}}^*$ instead of Eq. (2.5f). In this case, we replace the last line of expression (5.4) by the equations

$$
\begin{aligned}
&\bar{\gamma} = (1 + e_7) t / \bar{t}^*, \\
&\bar{l}_j^* = (1 + e_{3j})((1 + e_{4j})\bar{\gamma} l_j + (1 + e_{5j})\beta z_j), \qquad j > i.
\end{aligned} \tag{5.17}
$$

The formula analogous to (5.12) is now

$$
\begin{aligned}
\hat{l}_j^* = {}& \bar{l}_j^* (1 + e_1 + e_3 - e_6 - e_{3j} - e_{5j} + O(\epsilon^2)) \\
& + (-e_3 + e_6 - e_7 + e_{2j} - e_{4j} + e_{5j} + O(\epsilon^2)) l_j d / \bar{d}^*
\end{aligned} \tag{5.18}
$$

and this leads to the following expression for $E_{jk}^{(i)}$, $j > i$, $k > i$,

$$(5.19)\quad \begin{aligned} E_{jk}^{(i)} = {} & \bar{l}_j^*\bar{d}^*\bar{l}_k^*(-e_1 + e_2 - e_3 + e_4 + e_5 + 2e_6 \\ & \qquad + e_{3j} + e_{5j} + e_{3k} + e_{5k} + O(\epsilon^2)) \\ & + l_j d\bar{l}_k^*(e_3 - e_6 + e_7 - e_{2j} + e_{4j} - e_{5j} + O(\epsilon^2)) \\ & + \bar{l}_j^* d l_k(e_3 - e_6 + e_7 - e_{2k} + e_{4k} - e_{5k} + O(\epsilon^2)) \\ & + l_j d l_k(-e_2 - e_3 + e_{2j} + e_{2k} + O(\epsilon^2) + O(\epsilon^2)d/\bar{d}^*) \\ & + \bar{z}_j^*\bar{z}_k^*/\bar{t}^*(-e_1 + e_{1j} + e_{1k} + O(\epsilon^2)), \end{aligned}$$

and hence to the bound

$$(5.20)\quad \begin{aligned} |E_{jk}^{(i)}| \leqslant \epsilon \{ & 3|\bar{z}_j^*\bar{z}_k^*/\bar{t}^*| + 11a_ja_k + (4 + O(\epsilon)d/\bar{d}^*)b_jb_k \\ & + 6\sqrt{d/\bar{d}^*}(a_jb_k + b_ja_k)\}. \end{aligned}$$

This bound, in contrast to (5.16), shows that unacceptably large errors may occur only when $\sqrt{d/\bar{d}^*}$ is large, which happens only when $\sigma < 0$. It follows from (5.16) and (5.20) that very satisfactory error bounds hold for composite methods. For instance, if (2.5f) is used if $\sigma < 0$ and (3.4) if $\sigma > 0$, then it follows that the remaining $O(\epsilon^2)$ terms can be neglected, and

$$(5.21)\quad |E_{jk}^{(i)}| \leqslant \epsilon\{3|\bar{z}_j^*\bar{z}_k^*/\bar{t}^*| + 13a_ja_k + 4b_jb_k + 6(a_jb_k + b_ja_k)\}$$

Moreover, for the composite t-method defined at the end of Section 3, where (2.5f) is used if and only if $\bar{d}^*/d \leqslant 4$, we have the bound

$$(5.22)\quad |E_{jk}^{(i)}| \leqslant \epsilon\{3|\bar{z}_j^*\bar{z}_k^*/\bar{t}^*| + 13a_ja_k + 4b_jb_k + 12(a_jb_k + b_ja_k)\}.$$

To make further progress, we apply the Cauchy-Schwarz inequality to (5.22) to obtain

$$(5.23)\quad |E_{jk}^{(i)}| \leqslant \epsilon\{3|\bar{z}_j^*||\bar{z}_k^*|/|t^*| + (25\bar{d}^*\bar{l}_j^{*2} + 16dl_j^2)^{1/2}(25\bar{d}^*\bar{l}_k^{*2} + 16dl_k^2)^{1/2}\}.$$

This inequality can be used to obtain bounds for the total error E (see Eq. (5.3)) for the composite t-method. The case $\sigma > 0$ is considered first.

The computed matrix $\widetilde{A}$ in (5.3) satisfies the equation

$$(5.24)\quad \widetilde{A} = \sum_{p=1}^{i-1}\widetilde{\mathbf{l}}_p\widetilde{d}_p\widetilde{\mathbf{l}}_p^T + \sum_{p=i}^{n}\mathbf{l}_pd_p\mathbf{l}_p^T + \mathbf{z}^{(i)}\mathbf{z}^{(i)T}/t_i + \sum_{p=i}^{n}E^{(p)},$$

for $i = 1, 2, \cdots, n$, where $\widetilde{\mathbf{l}}_p$ is the vector $\bar{\mathbf{l}}^*$ of (5.2) calculated on the pth stage. The diagonal elements of this matrix equation give the inequality

$$(5.25)\quad |z_j^{(i)}|/t_i^{1/2} \leqslant (\widetilde{A}_{jj} + H_{jj})^{1/2}$$

where $H = \Sigma_i |E^{(i)}|$. Moreover, by the definition of $\tilde{A}$ and A, the inequalities

$$\sum_{p=1}^{i} \tilde{d}_p \tilde{l}_{jp}^2 \leqslant \tilde{A}_{jj}, \qquad \sum_{p=1}^{i} d_p l_{jp}^2 \leqslant A_{jj} \leqslant \tilde{A}_{jj} \tag{5.26}$$

also hold. Now the definition of H, expression (5.23), the Cauchy-Schwarz inequality, and expressions (5.25) and (5.26) give the bounds

$$\begin{aligned} H_{jk} &= \sum_{i=1}^{j} |E_{jk}^{(i)}| \qquad (j \leqslant k) \\ &\leqslant \epsilon \left\{ \sum_{i=1}^{j} 3|z_j^{(i+1)}|\,|z_k^{(i+1)}|/t_{i+1} \right. \\ &\quad \left. + \left(25 \sum_{i=1}^{j} \tilde{d}_i \tilde{l}_{ji}^2 + 16 \sum_{i=1}^{j} d_i l_{ji}^2\right)^{1/2} \left(25 \sum_{i=1}^{j} \tilde{d}_k \tilde{l}_{ki}^2 + 16 \sum_{i=1}^{j} d_k l_{ki}^2\right)^{1/2} \right\} \qquad (5.27) \\ &\leqslant \epsilon \left\{ 3j\sqrt{(\tilde{A}_{jj} + H_{jj})(\tilde{A}_{kk} + H_{kk})} + 41\sqrt{\tilde{A}_{jj}\tilde{A}_{kk}} \right\}. \end{aligned}$$

By letting $j = k$, it follows that $H_{jj} \leqslant \tilde{A}_{jj}\epsilon(41 + 3j)/(1 - 3j\epsilon)$, so we may neglect the H_{jj} and H_{kk} terms in (5.27) because they are second order. Thus, because $|E_{jk}| \leqslant H_{jk}$, we obtain the inequality

$$|E_{jk}| \leqslant \epsilon(3j + 41)\sqrt{\tilde{A}_{jj}\tilde{A}_{kk}}, \tag{5.28}$$

where $j \leqslant k$, which is our main bound on the total error of the calculation. Of course, it is possible to simplify $\sqrt{\tilde{A}_{jj}\tilde{A}_{kk}}$ to $\max_i \tilde{A}_{ii}$ and hence to $\|\tilde{A}\|$, but (5.28) is sharper. Also, it provides a bound that, in terms of relative errors, is invariant under scaling of rows and columns in A. Note that the dependence of (5.28) on the dimension of the problem is quite modest. A similar analysis for the case $\sigma < 0$, when inequality (5.21) holds, yields the result

$$|E_{jk}| \leqslant \epsilon(3j + 29)\sqrt{A_{jj}A_{kk}}. \tag{5.29}$$

A similar type of error analysis can be carried out when using the Gentleman-Bennett formulae. For a composite strategy that uses (2.5f) when $\bar{d}^*/d \leqslant 4$ and (3.4) otherwise, the bounds

$$\begin{aligned} |E_{jk}| &\leqslant \epsilon(5j + 39)\sqrt{\tilde{A}_{jj}\tilde{A}_{kk}}, \qquad \sigma > 0, \\ |E_{jk}| &\leqslant \epsilon(5j + 27)\sqrt{A_{jj}A_{kk}}, \qquad \sigma \leqslant 0, \end{aligned} \tag{5.30}$$

are obtained, where again $j \leqslant k$.

Although the results (5.28) and (5.29) indicate the satisfactory nature of the composite t-method, (5.16) may exaggerate the dangers of using the simple formula (2.5f) when $\sigma > 0$. It has been observed in practice that $\bar{d}^*/d$ can become very large without any growth in error being apparent. This can be explained using the above error analysis but rearranging the bounds in a different way. For instance, if all the off-diagonal elements of L satisfy the inequality

(5.31) $$|l_{ji}| \leqslant \sqrt{\bar{A}_{jj}/(j\bar{d}_i)} \quad \forall i, j, j > i,$$

then the last term in (5.16) can be written

(5.32) $$6(|l_j|\bar{d}^*|\bar{l}_k^*| + |\bar{l}_j^*|\bar{d}^*|l_k|) \leqslant 6(|\bar{d}^{*1/2}\bar{l}_k^*|\sqrt{\bar{A}_{jj}/j} + |\bar{d}^{*1/2}\bar{l}_j^*|\sqrt{\bar{A}_{kk}/k}).$$

If the same analysis is followed, it is readily seen that a bound like (5.28) applies, but with 29 replacing 41, even when $\bar{d}_i/d_i$ is large (provided that the term $O(\epsilon)\bar{d}_i/d_i$ can be neglected). An alternative way of presenting the result is to define

(5.33) $$\mu = \max_i \mu_i = \max_i \min\left(\sqrt{\bar{d}_i/d_i}, \max_{j>i} |l_{ji}|\sqrt{j\bar{d}_i/\bar{A}_{jj}}\right)$$

in which μ_i is small either if $\sqrt{\bar{d}_i/d_i}$ is small or if the subdiagonal elements l_{ji}, $j > i$, are all small. Using the simple notation introduced at the start of Section 5 for the ith basic step, if $\max_{j>i}|l_j|\sqrt{j\bar{d}^*/\bar{A}_{jj}} < \sqrt{\bar{d}^*/d}$ then it follows that $|\bar{d}^{*1/2}l_j| \leqslant \mu_i\sqrt{\bar{A}_{jj}/j}$ for all $j > i$. If, on the other hand, $\sqrt{\bar{d}^*/d}$ is smaller, then $|\bar{d}^{*1/2}l_j| \leqslant \mu_i|d^{1/2}l_j|$. Thus, from (5.16), by the same argument as before, one of the following inequalities must be true,

$$E_{jk}^{(i)} \leqslant \epsilon\{3|\bar{z}_j^*||\bar{z}_k^*|/|\bar{t}^*| + ((13 + 6\mu_i)\bar{d}^*\bar{l}_j^{*2} + 4dl_j^2 + 6\mu_i\bar{A}_{jj}/j)^{1/2}$$
$$((13 + 6\mu_i)\bar{d}^*\bar{l}_k^{*2} + 4dl_k^2 + 6\mu_i\bar{A}_{kk}/k)^{1/2}\}$$

or

$$E_{jk}^{(i)} \leqslant \epsilon\{3|\bar{z}_j^*||\bar{z}_k^*|/|\bar{t}^*| + ((13 + 6\mu_i)\bar{d}^*\bar{l}_j^{*2} + (4 + 6\mu_i)dl_j^2)^{1/2}$$
$$((13 + 6\mu_i)\bar{d}^*\bar{l}_k^{*2} + (4 + 6\mu_i)dl_k^2)^{1/2}\}$$

and this implies that

$$E_{jk}^{(i)} \leqslant \epsilon\{3|\bar{z}_j^*||\bar{z}_k^*|/|\bar{t}^*| + ((13 + 6\mu_i)\bar{d}^*\bar{l}_j^{*2} + (4 + 6\mu_i)dl_j^2 + 6\mu_i\bar{A}_{jj}/j)^{1/2}$$
$$((13 + 6\mu_i)\bar{d}^*\bar{l}_k^{*2} + (4 + 6\mu_i)dl_k^2 + 6\mu_i\bar{A}_{kk}/k)^{1/2}\}.$$

Following the same argument as above now leads to the overall result

(5.34) $$|E_{jk}| \leqslant \epsilon(3j + 17 + 18\mu)\sqrt{\bar{A}_{jj}\bar{A}_{kk}}, \quad j \leqslant k, \ \sigma > 0.$$

Therefore, if μ is small (either because the $\sqrt{\tilde{d}^*/d}$ are all small, or because the $|l_j|$ are small when $\sqrt{\tilde{d}^*/d}$ is large), then an acceptable bound on the error is obtained. Even if μ is not small, the error it causes does not grow with n. Nonetheless, because μ might be large, it is necessary to use some sort of composite process. Unfortunately, the expense of calculating $\max_{j>i}|l_{ji}|\sqrt{j\tilde{d}_i/\tilde{A}_{jj}}$ precludes its use in an algorithm. However, it does explain practical experience with the simple formula (2.5f) more fully than the bounds based solely on $\sqrt{\tilde{d}^*/d}$.

6. A Worked Example. An example of the modification process is given to illustrate the working of the simple and composite t-methods. The example is also chosen to show the effect of the errors which can arise when using formula (2.5f) in the simple t-method when $\tilde{d}_i/d_i$ is large. To show this error growth, it is necessary, by virtue of the latter part of Section 5, that the example have the property that both $\tilde{d}_i/d_i$ is large and also that some l_{ji}, $j > i$, is large. An appropriate example is constructed in the following way. The exact rational factorization of the 4×4 segment of the Hilbert matrix $(H_{ij} = 1/(i + j - 1))$ is given by

$$(6.1) \qquad L = \begin{matrix} 1 & & & \\ 1/2 & 1 & & \\ 1/3 & 1 & 1 & \\ 1/4 & 9/10 & 3/2 & 1 \end{matrix} \qquad D = \begin{matrix} 1 & & & \\ & 1/12 & & \\ & & 1/180 & \\ & & & 1/2800 \end{matrix}$$

In this representation, any errors are avoided which arise from the ill-conditioning normally associated with a Hilbert matrix. The matrix A is chosen by scaling H by multiplying both its second column and then second row by 10^{-2}, giving

$$(6.2) \qquad A = \begin{matrix} 1 & 1/2 \cdot 10^{-2} & 1/3 & 1/4 \\ 1/2 \cdot 10^{-2} & 1/3 \cdot 10^{-4} & 1/4 \cdot 10^{-2} & 1/5 \cdot 10^{-2} \\ 1/3 & 1/4 \cdot 10^{-2} & 1/5 & 1/6 \\ 1/4 & 1/5 \cdot 10^{-2} & 1/6 & 1/7 \end{matrix}$$

with factors

$$(6.3) \qquad L = \begin{matrix} 1 & & & \\ 1/200 & 1 & & \\ 1/3 & 100 & 1 & \\ 1/4 & 90 & 3/2 & 1 \end{matrix} \qquad D = \begin{matrix} 1 & & & \\ & 1/12 \cdot 10^{-4} & & \\ & & 1/180 & \\ & & & 1/2800 \end{matrix}$$

This factorization has the property that l_{32} and l_{42} are large relative to 1, and that d_2 is small. To create a matrix $\tilde{A}$ for which $\tilde{d}_2 \sim 1$ so that $\tilde{d}_2/d_2$ is large,

TABLE 1. *A Worked Example Showing Error Growth*

$L = 1$				$D = 1$				$\sigma = 1$	$z = 1$
.005	1				$.8333_{10}-5$				1
.3333	100	1				$.5556_{10}-2$			1
.25	90	1.5	1				$.3571_{10}-3$		1

i	1	2	3	4
v_i	1	.9950	−98.83	59.40
v_i/d_i	1	$1194_{10}2$	$-1779_{10}1$	$1663_{10}2$
t_{i+1}	2	$1188_{10}2$	$1877_{10}3$	$1176_{10}4$
t_{i+1}/t_i	2	$5940_{10}1$	15.80	6.265
$\tilde{d}_i$	2	.4950	.08778	$.2237_{10}-2$
β_i	.5	1.005	$-.9478_{10}-2$	
$z^{(i+1)}$	0			
	.9950			
	.6667	−98.83		
	.7500	−88.80	59.40	

$\tilde{1}_i$ (simple t-method, using (2.5f))

1		
.5025	1	
.6667	.6800	1
.6250	.7600	.9370

$\tilde{1}_i$ (composite t-method, using (3.4))

1		
.5025	1	
.6667	.6717	1
.6250	.7553	.9365

equivalent $\tilde{A}$ (simple t-method), lower triangle only

2			
1.0050	1.0000		
1.3334	1.0066	1.2056	
1.2500	1.0043	1.1714	1.1465

equivalent $\tilde{A}$ (composite t-method), lower triangle only

2			
1.0050	1.0000		
1.3334	1.0025	1.2001	
1.2500	1.0020	1.1667	1.1429

1 is added to every element of A, so destroying the bad scaling. This corresponds to a rank one modification of A for which

$$\sigma = 1, \quad \mathbf{z} = \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \end{matrix} \tag{6.4}$$

To show the error growth, the factors in (6.3) are represented to *four* significant decimal digits, and computations are carried out in four digit decimal floating-point arithmetic. The results obtained using the simple and composite t-methods are set out in detail in Table 1. On examining these results, it will be seen that the methods do not differ in the numbers obtained for $\tilde{\mathbf{l}}_1$ because, at this stage, the bad scaling which has been introduced has no effect. However, for the second column of $\tilde{L}$, $\sqrt{\tilde{d}_2/d_2} = \sqrt{t_3/t_2} = \sqrt{59400} \sim 250$. Thus, from the error analysis, we might expect to lose two significant figures in $\tilde{\mathbf{l}}_2$ when using the simple t-method, and, indeed, comparison of $\tilde{\mathbf{l}}_2$ produced by the two methods shows that this is what happens. If the computation of $\tilde{l}_{32}$ and $\tilde{l}_{42}$ is carried out separately by (2.5f) and by (3.4), then the loss of significance in using the former is easily seen. The matrix $\tilde{A}$ corresponding to the calculated factors $\tilde{L}$ and $\tilde{D}$ has also been calculated (exactly) and it can be seen that the matrix $\tilde{A}$ given by the composite t-method agrees to within 1 in the fifth significant figure with the matrix obtained by adding 1's to the elements of (6.2). The matrix $\tilde{A}$ obtained from the simple t-method however shows errors in the last two figures.

Mathematics Department
University of Dundee
Dundee, Scotland

Theoretical Physics Division
Atomic Energy Research Establishment
Harwell, England

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

2. W. M. GENTLEMAN, *Error Analysis of QR Decompositions by Givens Transformations*. (Author's Manuscript, 1973.)

3. W. M. GENTLEMAN, "Least squares computations by Givens transformations without square roots," *J. Inst. Math. Appl.*, v. 12, 1973, pp. 329–336.

4. P. E. GILL, G. H. GOLUB, W. MURRAY & M. A. SAUNDERS, *Methods for Modifying Matrix Factorizations*, NPL Report NAC 29, 1972.

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

6. D. GOLDFARB, "Extension of Davidon's variable metric method to maximization under linear inequality and equality constraints," *SIAM J. Appl. Math.*, v. 17, 1969, pp. 739–764. MR **44** #7978.

→ J. B. ROSEN, "The gradient projection method for nonlinear programming. I. Linear constraints," *J. Soc. Indust. Appl. Math.*, v. 8, 1960, pp. 181–217. MR **22** #3601.

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