method for solving (11.1) is via the S.V.D. of B:

$$B = U\Sigma V^H, \tag{11.4}$$

where Σ has n_1 zero elements, assumed to be in the last n_1 diagonal positions. Hence we wish to solve

$$\Sigma(V^H x) \equiv \Sigma y = U^H([u_1 \mid u_2 \mid u_3 \mid v_1 \mid v_2 \mid w_1 \mid w_2]Z)\alpha \tag{11.5}$$

$$= ([U^H u_1 \mid U^H u_2 \mid U^H u_3 \mid U^H v_1 \mid U^H v_2 \mid U^H w_1 \mid U^H w_2]Z)\alpha. \tag{11.6}$$

Solutions are obtained via these values of α for which the right-hand side p has zero components in the last n_1 positions. In our example, $n_1 = 3$, and the equations become

$$\begin{bmatrix} \sigma_1 y_1 = p_1 \\ \sigma_2 y_2 = p_2 \\ \sigma_3 y_3 = p_3 \\ \sigma_4 y_4 = p_4 \\ 0 y_5 = 0 \\ 0 y_6 = 0 \\ 0 y_7 = 0 \end{bmatrix}. \tag{11.7}$$

Components y_5, y_6, y_7 are therefore arbitrary and in our algorithm are taken to be zero, since they merely result in including multiples of u_1, u_2, u_3 in the vector α derived from y.

We have still to discuss how we avoid duplicating the vectors we have previously produced. Suppose at the stage when we are determining the vectors v_1 and v_2 we have computed a Z (which might be called Z_1) such that

$$[U^H u_1 \mid U^H u_2 \mid U^H u_3]Z_1 = \left[\begin{array}{ccc} X & X & X \\ X & X & X \\ X & X & X \\ X & X & X \\ X & 0 & 0 \\ X & 0 & 0 \\ X & 0 & 0 \end{array}\right] = [p^{(1)} p^{(2)} p^{(3)}]. \tag{11.8}$$

Then v_1 and v_2 are obtained by solving $\Sigma y = p$, where p takes in turn each of the vectors $p^{(2)}$ and $p^{(3)}$, giving independent solutions. Now when we have to solve the set

$$\Sigma y = ([U^H u_1 \mid U^H u_2 \mid U^H u_3 \mid U^H v_1 \mid U^H v_2]Z)\alpha, \tag{11.9}$$

if we take $Z = \begin{bmatrix} Z_1 & \\ & I \end{bmatrix}[W_2]$, then (11.9) becomes

$$\Sigma y = ([p^{(1)} \mid p^{(2)} \mid p^{(3)} \mid U^H v_1 \mid U^H v_2][W_2])\alpha. \tag{11.10}$$

Columns two and three have already been dealt with and gave us v_1 and v_2. The new solutions are obtained by solving

$$\Sigma y = ([p^{(1)} \mid U^H v_1 \mid U^H v_2]Z_2)\alpha. \tag{11.11}$$

Notice that in this stage we are left with only three columns (i.e., n_1) just as in the previous stage. Again we determine a Z_2 which will make the trailing columns of the matrix in parentheses on the right of (11.10) of the requisite form, i.e., with zero in positions 5, 6, 7 (the last n_1 positions). The number of vectors of this form will decide how many vectors of grade 3 we obtain. The algorithm is now complete, and we observe that at each stage we are dealing with a system of the form

$$\Sigma y = (RZ)\alpha, \tag{11.12}$$

where R is always an $n \times n_1$ matrix, and we wish to determine Z so that RZ is of the requisite form, i.e., its trailing columns have zeros in the last n_1 positions. This is conveniently done via an S.V.D. composition. We write

$$R = \begin{matrix} \boxed{\begin{matrix} R_1 \\ R_2 \end{matrix}} \\ \underbrace{}_{n_1} \end{matrix} \begin{matrix} \\ \}n_1 \end{matrix}, \tag{11.13}$$

where R_2 is an $n_1 \times n_1$ matrix. If $R_2 = U_2\Sigma_2 V_2^H$, where Σ_2 has at the sth stage n_s zero diagonals, then taking $Z = V_2$,

$$RZ = \begin{bmatrix} R_1 V_2 \\ U_2\Sigma_2 \end{bmatrix}, \tag{11.14}$$

and the last n_s columns are of the required form with regard to the n_s zero elements in Σ_2.

The general algorithm may now be described by its typical stage at which we determine vectors of grade $s + 1$. We assume that by this time we have

$$\begin{aligned} &n_1 \text{ vectors of grade } 1: u_1^{(1)}, u_2^{(1)}, \cdots, u_{n_1}^{(1)}, \\ &n_2 \text{ vectors of grade } 2: u_1^{(2)}, u_2^{(2)}, \cdots, u_{n_2}^{(2)}, \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ &n_s \text{ vectors of grade } s: u_1^{(s)}, u_2^{(s)}, \cdots, u_{n_s}^{(s)}. \end{aligned} \tag{11.15}$$

We then assemble an $n \times n_1$ matrix $R^{(s+1)}$, the first $n_1 - n_s$ columns of which will be denoted by $P^{(s)}$, the origin of which will become obvious during the description of this next stage. The remaining n_s columns are $U^H u_1^{(s)}, \cdots, U^H u_{n_s}^{(s)}$. This matrix $R^{(s+1)}$ is partitioned in the form

$$R^{(s+1)} = \begin{matrix} \begin{bmatrix} R_1^{(s+1)} \\ R_2^{(s+1)} \end{bmatrix} \\ \underbrace{\phantom{R_2^{(s+1)}}}_{n_1} \end{matrix} \begin{matrix} \\ \}n_1 \end{matrix}. \tag{11.16}$$

If the S.V.D. of $R_2^{(s+1)}$ is $U^{(s+1)}\Sigma^{(s+1)}(V^{(s+1)})^H$, where the number of zero elements in $\Sigma^{(s+1)}$ is denoted by n_{s+1}, then

$$R^{(s+1)}V^{(s+1)} = \left[\begin{array}{c} R_1^{(s+1)}V^{(s+1)} \\ \hline U^{(s+1)}\Sigma^{(s+1)} \end{array}\right] = \left[\begin{array}{c|c} P_{11}^{(s+1)} & P_{12}^{(s+1)} \\ \hline \underbrace{P_{21}^{(s+1)}}_{n_1 - n_{s+1}} & \underbrace{0}_{n_{s+1}} \end{array}\right] \tag{11.17}$$

and the vectors $y_1^{(s+1)}, \cdots, y_{n_{s+1}}^{(s+1)}$ are obtained by solving $\Sigma y = p$, where p takes each of the last n_{s+1} columns of the matrix on the right in turn. The $u_i^{(s+1)}$ are then obtained by multiplying these vectors by V. The $n \times (n_1 - n_{s+1})$ matrix in the first $n - n_{s+1}$ columns of the matrix on the right of (11.17) is the matrix $P^{(s+1)}$ required in the next stage. The process terminates when $n_{s+1} = 0$.

In the first stage we solve

$$\Sigma y = 0 \tag{11.18}$$

and obtain the solutions $y = e_{n-n_1+1}, e_{n-n_1+2}, \cdots, e_n$, the last n_1 columns of the identity matrix and hence $u_1^{(1)}, \cdots, u_{n_1}^{(1)}$ are the last n_1 columns of V. The vectors of grade 1 are therefore orthogonal, but this is not true of any of the subsequent sets of vectors, though by taking $y_{n-n_1+1}, \cdots, y_n$ to be zero in each of the subsequent solutions of $\Sigma y = p$, one ensures that all vectors of grades higher than one are orthogonal to those of grade 1.

Observe that the successive S.V.D.'s are all performed on a matrix of order $n_1 \times n_1$. In the case when $n_1 = 1$ and there is only one block in the J.c.f. associated with the current eigenvalue, this will be a 1×1 matrix at each stage and the process comes to an end when the last element of $U^H u_1^{(s)}$ is nonzero!

12. Comments on algorithms for principal vectors. So far we have concentrated mainly on the formal aspects of the algorithms, though in using the S.V.D. we are tacitly recognizing numerical difficulties. The first problem is how to select our λ when forming $B = A - \lambda I$. In practice, the eigenvalues of A should have been found using some stable algorithm such as the QR algorithm. Although the computed λ_i may be arbitrarily bad, each should be exact for some matrix $A + E_i$, where E_i is such that $\|E_i\|_2/\|A\|_2$ is merely a modest multiple of the computer precision. Hence $B = A - \lambda_i I$ should have at least one negligible singular value relative to $\|A\|_2$, however "poor" λ_i may be in an absolute sense. However, if A really is defective, the computed λ_i are probably not the best values to use. If, for example, A has a well-defined J.c.f. (i.e., in the optimum quasi-J.c.f., the superdiagonal elements are of the order of magnitude of $\|A\|_2$) and there is just one block $J_r(\lambda_1)$ associated with λ_1, one will expect the computed λ_i to include a set of r values $\bar{\lambda}_1, \cdots, \bar{\lambda}_r$ which, though not particularly close to λ_1, will be such that their sum is very close to $r\lambda_1$. If one could *recognize* such a block, one should use the mean of those values $\bar{\lambda}$ and then work with $B = A - \bar{\lambda}I$. However, in practice, the situation will be much more obscure than this, and it is a difficult problem to decide which values of λ to use.

Whichever of the algorithms we use, we shall need at each stage when an S.V.D. is performed a satisfactory criterion for deciding which singular values may be regarded as "zero". The situation is most satisfactory in connection with

the deflation technique. At each stage, the matrix on which the S.V.D. is performed has been determined by a unitary similarity on $(A - \lambda I)$, and it is reasonable to use some tolerance $\varepsilon\|A\|_2$ throughout, when ε is "small" but appreciably larger than the machine precision.

In the powering algorithm, the rth matrix is of degree r in the elements of A, and the decision is much less satisfactory. A modification of the procedure has been developed which ameliorates this difficulty, but matrix powering would seem to have nothing to recommend it in comparison with the deflation algorithm.

The Golub–Wilkinson algorithm is far superior from the point of view of economy of computation; while the first S.V.D. is done on $A - \lambda I$, the others are all performed on a submatrix of a set of n_1 vectors. If the vectors $u_j^{(i)}$ are normalized at each stage, a negligible singular value would be one which is small compared with unity. If in the matrix Σ obtained from $B = A - \lambda I$ itself the smallest singular value to be regarded as nonzero is quite close to the tolerance, then in determining all subsequent solutions of equations of the form $\Sigma y = p$, the element y_{n-n_i} is obtained by dividing by this *almost negligible* σ_{n-n_i}. The vectors obtained with this process are not orthogonal, as they are with the other two and there does appear to be a danger that they may be almost linearly dependent with a consequent loss of digital information.

None of the three processes gives principal vectors satisfying the chain reaction typical of the columns of the X producing the J.c.f. Modified vectors satisfying the chain reaction can be determined from the computed vectors, but the volume of work is substantial and care is needed to avoid losing digital information. Some such loss is inevitably involved in going from the *orthogonal* sets given by the powering and deflation algorithms, since the vectors in the chains may be arbitrarily near to linear dependence. Indeed, one might well ask whether one *should* move from the orthogonal sets to sets satisfying the chain relations. The answer must depend on what the vectors are needed for, and here numerical analysts would welcome discussion with applied mathematicians, since this is clearly a subjective matter. Further experimentation is necessary before the algorithm can be fully assessed.

13. Poorly-defined J.c.f. As mentioned previously, there is a natural tendency to construct "difficult" examples for testing purposes by taking a J.c.f. and subjecting it to some simple similarity transformation. Such examples severely underestimate the difficulties associated with ill-conditioned matrices. The point is well illustrated by considering the Frank matrices F_n defined typically by

$$F_5 = \begin{bmatrix} 5 & 4 & 3 & 2 & 1 \\ 4 & 4 & 3 & 2 & 1 \\ & 3 & 3 & 2 & 1 \\ & & 2 & 2 & 1 \\ & & & 1 & 1 \end{bmatrix}. \tag{13.1}$$

Even for quite modest values of n, some of the eigenvalues and eigenvectors are very ill-conditioned, and yet one has a simple method of determining them by

observing that, for example,

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} (F_5 - \lambda I)$$

(13.2)

$$= \begin{bmatrix} 1-\lambda & \lambda & & & \\ 4 & 1-\lambda & \lambda & & \\ & 3 & 1-\lambda & \lambda & \\ & & 2 & 1-\lambda & \lambda \\ & & & 1 & 1-\lambda \end{bmatrix} = G_5.$$

This result is quite general and enables us to determine the eigenvalues of F_5, for example, from those of the quasi-symmetric tridiagonal matrix

(13.3)

$$T_5 = \begin{bmatrix} 0 & 1 & & & \\ 4 & 0 & 1 & & \\ & 3 & 0 & 1 & \\ & & 2 & 0 & 1 \\ & & & 1 & 0 \end{bmatrix}.$$

The determination of these latter eigenvalues is a well-conditioned problem for all values of n. We are able to remove the ill-condition in this way because the transformation can be performed exactly, i.e., without rounding error. The eigenvalues of F_n are very sensitive to perturbation in elements in the top right-hand corner, and by transforming to T_n and then working *explicitly with a tridiagonal matrix* one ensures that no rounding errors are effectively made in these elements! From this transformation it is easy to show that eigenvalues of F_n are such that $\lambda_r = 1/\lambda_{n-r+1}$. It is the smaller eigenvalues which are ill-conditioned.

To illustrate the nature of the ill-conditioning, we concentrate for the moment on F_{12} and discuss the problem from the point of view of computation on KDF9 which has a 39-digit binary mantissa, i.e., rather less than 12 decimal digits of accuracy.

By row transformations, we see that $\det(F_n) = 1$, and if $\tilde{F}_n$ is the matrix resulting from a perturbation ε in position $(1, n)$, we have $\det(\tilde{F}_n) = 1 \pm (n-1)!\varepsilon$. Since the determinant is the product of the eigenvalues, it is evident that changes of $\pm 1/(n-1)!$ in this element alter the product of the eigenvalues from the true value, 1, to 0 and 2, respectively. When $n = 100$, for example, this represents a change of approximately 10^{-158}. To obtain the eigenvalues correct to 10 decimals even with an extremely stable *general purpose* algorithm would require computation

with a mantissa of about 170 decimal digits. Yet the eigenvalues may be determined via T_{100} working to ten decimals only.

For $n = 12$, the situation is not yet too serious with 12-digit decimal computation, since $11! = 4 \times 10^7$. One can expect to obtain some correct digits even in the most ill-conditioned eigenvalues. The quantities s_i for the four smallest eigenvalues are

$$s_{12} = 5 \times 10^{-8}, \quad s_{11} = 3 \times 10^{-8}, \quad s_{10} = 4 \times 10^{-8}, \quad s_9 = 15 \times 10^{-8}, \tag{13.4}$$

the corresponding eigenvalues being

$$\begin{aligned} \lambda_{12} &= 0.0310 \cdots, \quad \lambda_{11} = 0.0495 \cdots, \\ \lambda_{10} &= 0.0812 \cdots, \quad \lambda_9 = 0.1436 \cdots, \end{aligned} \tag{13.5}$$

where we have given only the order of magnitude of the s_i. In fact the errors in the eigenvalues as computed on KDF9 using the very stable QR algorithm were 4×10^{-6}, 7×10^{-6}, 5×10^{-6} and 10^{-7}, respectively, and from the sensitivity considerations discussed in § 4, these results are seen to be extremely creditable.

From the discussion in that section, we also know that there is certainly a matrix having a double eigenvalue λ_{11} at a distance within $\|F_{12}\|s_{11}$, but in fact, F_{12} is much nearer to a defective matrix than this. Indeed, it is near to quite a number of different defective matrices. Let us consider first the possibilty of inducing defectiveness by a perturbation ε in the (1, 12)-element only. The modified characteristic equation is

$$\prod(\lambda_i - \lambda) - 11!\varepsilon = 0. \tag{13.6}$$

If we draw the graph $y = \prod(\lambda_i - \lambda)$, then the modified eigenvalues are at the values of λ for which $\prod(\lambda_i - \lambda) = 11!\varepsilon$. The situation is illustrated in Fig. 1.

Taking ε to be negative, we obtain a double root when the line $y = 11!\varepsilon$ is tangential to the curve, which first occurs at a point between λ_{11} and λ_{12}. If we take ε positive, a double root is obtained when the line is tangential to the curve at a point between λ_{10} and λ_{11}. It is surprisingly easy to compute these points quite accurately, provided $\prod(\lambda_i - \lambda)$ is computed from G_{12}, not from F_{12}! The value of ε is quite a lot smaller than $\|F_{12}\|s_{11}$, and on reflection, this is not surprising. In establishing that result, we attempted to induce a double eigenvalue at the value λ_i itself for which the s_i is small. It is to be expected that a smaller perturbation is needed to produce a double eigenvalue at some point "between" that λ_i and some other λ_j. As we have seen, there are always perturbations εB for which $\partial\lambda_i/\partial\varepsilon = \pm 1/s_i$. At least one of the other λ_j must be changing fast "to keep the trace correct", and we would expect to be able to make $\lambda_i(\varepsilon)$ and some $\lambda_j(\varepsilon)$ move towards each other. As they get nearer, we would expect s_i to get even smaller, and one feels intuitively that a perturbation nearer the order of magnitude $|\frac{1}{4}(\lambda_i - \lambda_j)s_i|$ is likely to give a double eigenvalue at a value of roughly $\frac{1}{2}(\lambda_i + \lambda_j)$. The quantity $|\frac{1}{4}(\lambda_i - \lambda_j)s_i|$ is likely to be much smaller than $\|A\|s_i$ since the relevant λ_j is likely to be at least "fairly close" to λ_i. This certainly proves to be true for F_{12}. In fact, a value $\varepsilon = -10^{-10}(3.95 \cdots)$ gives a double root between λ_{11} and λ_{12} at $\lambda = 0.038 \cdots$.

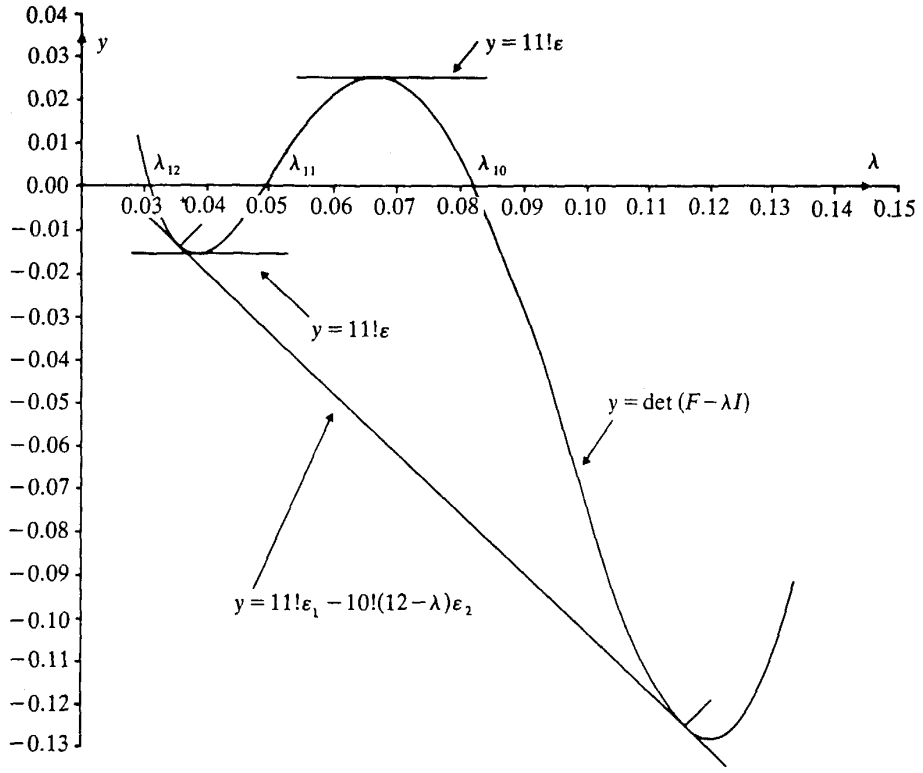

FIG. 1

If perturbations ε_1 and ε_2 are made in $F_{12}(1, 12)$ and $F_{12}(2, 12)$, respectively, then the characteristic equation becomes

$$\prod(\lambda_i - \lambda) - 11!\varepsilon_1 + 10!\varepsilon_2(12 - \lambda) = 0, \tag{13.7}$$

and the eigenvalues are at the intersection of the straight line $y = 11!\varepsilon_1 - 10!\varepsilon_2 \times (12 - \lambda)$ with the curve $y = \prod(\lambda_i - \lambda)$. By appropriate choices of ε_1 and ε_2, this line can be made tangential to the curve at two points, one between λ_{12} and λ_{11} and one between λ_{10} and λ_9. The values are in fact $\varepsilon_1 = -10^{-7}(6.24\cdots)$ and $\varepsilon_2 = -10^{-7}(3.9\cdots)$ and it gives coincident eigenvalues at $0.036\cdots$ and 0.116. Notice that if one attempts to solve these perturbed matrices by the QR algorithm on KDF9, the separation of the "paired" eigenvalues may, at first sight, seem disappointing. Two points should be emphasized. First, since the KDF9 has a mantissa with less than 12 decimal digits, the perturbations ε_i cannot be inserted with any great precision since they occur via entries $1 + \varepsilon_i$. Hence even if the ε_i are determined accurately, they cannot be included in $1 + \varepsilon_i$ without incurring an error of between 10^{-11} and 10^{-12}. Further, in solving the perturbed matrix $A + E$ on KDF9, the effect of rounding errors will imply that a computed λ_i is an eigenvalue of $A + E + E_i$ when $\|E_i\|_2/\|A\|_2$ is likely to be a modest multiple of 2^{-39} (i.e., $10^{-11.7}$). Since we are now extremely close to a defective matrix, s_i will be quite a lot smaller than the corresponding value for A itself. In fact,

with $\varepsilon_1 = -10^{-10}(3.9\cdots)$, the two close computed values of λ were $0.03758\cdots$ and $0.03968\cdots$, the mean of these being $0.03863\cdots$; this is very close to the minimum point of $\prod(\lambda - \lambda_i)$ between λ_{12} and λ_{11}. Again working with the perturbed version of G_{12}, it is possible not only to insert the perturbations accurately (since they now arise as $\varepsilon_1 - \varepsilon_2$ and ε_2 and not as $1 + \varepsilon_1$ and $1 + \varepsilon_2$), but also to compute the eigenvalues of the perturbed matrix accurately. Altogether, the Frank matrices provide good material for investigating ill-conditioned eigenvalues and eigenvectors. It is clear that by the time $n = 20$, F_n is very near to a large number of defective matrices having different sets of multiple eigenvalues and even elementary divisors of different degrees. It is natural to ask what information one should really extract and why.

Continuing with F_{12} and KDF9 (and we make no excuse for being so specific—the "difficulty" involved in dealing with a matrix is intimately associated with the precision of computation one is prepared to use; on a 40-decimal digit computer F_{12} could reasonably be regarded as well-conditioned!) the dilemma is particularly acute. The computed $\lambda_9, \lambda_{10}, \lambda_{11}, \lambda_{12}$ all have *some* accuracy, and it is debatable whether there is anything to be gained by pretending that they are equal or equal in pairs, etc. On the other hand, if one treats them as distinct and computes the corresponding eigenvectors, not only will these eigenvectors inevitably be inaccurately determined, but they will also be almost linearly dependent. Indeed, if we use the QR algorithm, they will be exact for some $A + E$ with $\|E\|_2/\|A\|_2$ of the order of 2^{-39}. The s_i for this matrix will be quite close to those of A itself, and the smallest of these is roughly 3×10^{-8}. How much information do we have at best about the space of dimension four spanned by the corresponding eigenvectors? From § 5 we see that these vectors are linearly dependent to within less than 3×10^{-8}. Certainly the fourth orthogonal direction is extremely poorly-determined. Indeed, all four vectors are "fairly" parallel, and in performing the Schmidt orthogonalization process, there will be a loss of figures at each stage.

Would it not be better to group these four eigenvalues together and to attempt to determine directly a set of four orthogonal vectors spanning the corresponding invariant subspace? One can certainly determine the subspace in this way much more accurately. Whether it is better or not depends on what one really wants. If accuracy is an overriding consideration, the "best" thing to do is to group all 12 eigenvalues together, and then any 12 orthogonal vectors specify the subspace exactly, $e_1, e_2, \cdots, e_{12}$ being an obvious choice! Here we have the ultimate absurdity of perfect accuracy in a set of vectors but no information.

A sensible compromise would seem to be the following. On a t-digit computer, we might aim to determine the smallest groupings of the eigenvalues for which one can claim that all the computed orthogonal bases "have t' correct digits". Obviously one must have $t' < t$, and if one insists on t' being too close to t, one runs the risk of being forced into large groups with a consequent loss of information. There is no need to get into abstruse discussions about the meaning to be attached to the angle between a computed set of s orthogonal vectors and an exact set of s vectors defining the subspace. Since we are unlikely to require less than 3 decimal digits (say), we would merely be arguing about the relative merits of θ, $\sin\theta$, $\tan\theta$, $2\sin(\theta/2)$, etc., when $\theta < 10^{-3}$, and clearly such matters are of no importance. The following is a perfectly adequate measure of the angle between the orthonormal

set $u_1, u_2, \cdots, u_s$ and the orthonormal set $v_1, v_2, \cdots, v_s$. We may write

$$u_i = \alpha_{i1}v_1 + \cdots + \alpha_{is}v_s + r_i, \qquad i = 1, \cdots, s, \tag{13.8}$$

and the r_i might reasonably be called the residual vectors. If the two bases spanned the same subspace, then $r_i = 0$. Therefore $\max \|r_i\|$ may be regarded as a measure of the errors in the u_i relative to the v_i. In fact, $\|r_i\|$ is the sine of the angle between u_i and the space spanned by the v_i.

14. Calculation of orthogonal bases of invariant subspaces. In classical similarity theory, unitary similarities play quite an important role, since when $X^H = X^{-1}$,

$$B = X^{-1}AX = X^HAX, \tag{14.1}$$

and hence matrices which are unitarily similar are also *conjunctive*. The fundamental result with respect to unitary similarities is that for any complex matrix A, there exists a unitary matrix X such that

$$X^HAX = T, \tag{14.2}$$

where T is upper triangular with the eigenvalues of A on its diagonal. This is known as the *Schur canonical form*. The ordering of the λ_i on the diagonal may be chosen arbitrarily.

Unitary transformations are of great significance for numerical analysts because a wide range of algorithms based on them are numerically stable. When A is real, it may in general have some complex eigenvalues, though these of course occur in conjugated pairs. It is convenient to remain in the real field whenever possible, and there is a single modification of Schur's result which states that when A is real, there is an *orthogonal* X (i.e., a real unitary X) such that

$$X^TAX = T, \tag{14.3}$$

where T is now almost triangular, except that corresponding to each complex conjugate pair of eigenvalues T has a 2×2 block on the diagonal having as its two eigenvalues this complex pair. This is usually known as the Wintner–Murnaghan canonical form [29].

It is precisely this form which is produced by the double Francis QR algorithm, perhaps the most widely used general-purpose algorithm for finding the eigensystem of a nonnormal real matrix. This algorithm works directly with a real-upper Hessenberg matrix but a general real matrix may be reduced to this form by a (real) orthogonal similarity. (For detailed discussions, see [28].) The combined reduction from general form to real almost triangular T is extremely stable, and it has been proved [25] that the computed matrix is such that

$$T = X^T(A + E)X, \tag{14.4}$$

where X is exactly orthogonal and $\|E\|/\|A\|$ is a modest multiple of the machine precision. Further, the computed $\overline{X}$ is very close to the exactly orthogonal X for which (14.4) is true and hence, in particular, has columns which are orthogonal almost to working accuracy. Since the computed T is exactly orthogonally similar to $A + E$ and the s_i are invariant with respect to orthogonal transformations,

the s_i of the matrix T give information that really is relevant. The left-hand and right-hand eigenvectors of T may be readily computed; the right-hand eigenvectors are required in any case, and the additional work needed to compute the left-hand eigenvectors of T is a negligible percentage of that for the complete reduction. Ignoring the 2×2 blocks for the moment, we determine the left-hand and right-hand vectors for the eigenvalue in position r on the diagonal by a triangular back substitution with matrices of order $n - r$ and r, respectively. The vectors are of the forms

$$(14.5) \qquad (0, \cdots, 0, y_r, y_{r+1}, \cdots, y_n) \quad \text{and} \quad (x_1, x_2, \cdots, x_r, 0, \cdots, 0),$$

and if these are normalized vectors, the corresponding $s = x_r y_r$. The complication caused by the 2×2 blocks is not substantial and is discussed in detail in [28, pp. 372, 374]. The computed s_i are invaluable in any case, since they give the sensitivities of the eigenvalues of T, i.e., of $(A + E)$.

Now let us consider the eigenvalues in the first s positions along the diagonal of T. We may write

$$(14.6) \qquad X^{-1}(A + E)X = \begin{bmatrix} T_{11} & T_{12} \\ \underbrace{0}_{s} & \underbrace{T_{22}}_{n-s} \end{bmatrix} \begin{matrix} \}s \\ \}n-s \end{matrix},$$

and hence

$$(14.7) \qquad (A + E)X_s = X_s T_{11},$$

where X_s consists of the first s columns of the orthogonal matrix X. Notice that this is true even if there are 2×2 blocks included in T_{11}, provided the first of a pair of conjugate eigenvalues is not in position s. These s orthogonal vectors therefore provide an orthogonal basis for the invariant subspace of $A + E$ corresponding to this group of s eigenvalues, and, as we have remarked, even the *computed* columns of X are accurately orthogonal. They do, of course, provide information only about the subspaces of $A + E$ rather than of A itself, but any loss of accuracy due to this perturbation is inherent in the problem and cannot be avoided without working to a higher precision (or exactly!) at least in some significant part of the computation. Although the individual eigenvectors corresponding to those s eigenvalues may be almost linearly dependent, the columns of X_s, being orthogonal, cannot have this shortcoming.

There is no *correspondingly simple* method for computing a set of orthogonal vectors giving the invariant subspace corresponding to a set of λ_i which are not in the leading position. However, given any collection of λ_i, it is possible to transform T into an upper triangular $\tilde{T}$ having these λ_i in the leading positions by means of an orthogonal similarity. Hence we have an orthogonal Y such that

$$(14.8) \qquad Y^T(T + F)Y = \tilde{T},$$

where F is the result of rounding errors, and since the process is stable, $\|F\|/\|T\|$ and hence $\|F\|/\|A\|$ is of the order of the machine precision. Hence, finally,

$$(14.9) \qquad Y^T X^T (A + E + G)YX = \tilde{T},$$

where $G = XFX^T$ and $\|G\|_2 = \|F\|_2$, and the first s columns of YX give an orthogonal basis of the subspace corresponding to the selected s eigenvalues.

The transformation from T to $\bar{T}$ was first described by Ruhe [14]. It is achieved by a sequence of orthogonal similarities, each of which is a plane rotation and is based on the following observation. If

$$T = \begin{bmatrix} p & q \\ 0 & r \end{bmatrix}, \tag{14.10}$$

then there is a plane rotation R such that

$$R^T T R = \begin{bmatrix} r & q \\ 0 & p \end{bmatrix}.$$

If this is true, then clearly

$$R^T \begin{bmatrix} 0 & q \\ 0 & r-p \end{bmatrix} R = \begin{bmatrix} r-p & q \\ 0 & 0 \end{bmatrix} \tag{14.11}$$

or

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 0 & q \\ 0 & r-p \end{bmatrix} \cdot \begin{bmatrix} r-p & q \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}. \tag{14.12}$$

For this to be true, $(r-p)\cos\theta - q\sin\theta = 0$, giving

$$\cos\theta = q/\alpha, \quad \sin\theta = (r-p)/\alpha, \quad \alpha = \sqrt{(r-p)^2 + q^2}, \tag{14.13}$$

and a simple verification shows that with this choice of R, the relation *is* true. Ruhe gave the analogous result in the complex case; in this, q becomes $\bar{q}$ in the transformed matrix. Using this algorithm, any eigenvalue may be brought into any required position along the diagonal by a sequence of plane rotations. When T is real but has 2×2 blocks corresponding to complex eigenvalues, an analogous result is true in which a complex pair is always kept together in the form of a real 2×2 block. One needs only two additional algorithms which serve to interchange the position of a single real diagonal element and a real 2×2 block and to interchange the positions of two 2×2 blocks. (N.B., the 2×2 blocks need not remain invariant; only their eigenvalues.) The relevant algorithms have been coded on KDF9 and are numerically stable.

There remains the problem of the grouping, and there does not yet appear to be a perfectly satisfactory method of deciding on this. It cannot be decided purely on the basis of the separation, since even multiple eigenvalues corresponding to elementary divisors of moderate degree will not in general lead to "close" eigenvalues in the computed set. Further, even when the exact λ_i and λ_j are by no means pathologically close, they may be so sensitive that small perturbations in A may make them so. A good working test is that a perturbation E *may* make them coincident if

$$\frac{\|E\|_2}{|s_i|} + \frac{\|E\|_2}{|s_j|} \geqq |\lambda_i - \lambda_j|, \tag{14.14}$$

though since $\|E\|/s_i$ is merely a first order perturbation, a smaller $\|E\|$ than this may well be adequate.

However, we are not merely concerned with whether the computed λ_i and λ_j could belong to a multiple root. If this is our criterion, then the groups will be much smaller than is advisable. A reasonably satisfactory rule is that if our aim is to have t' decimal digits correct in the subspace on a t-digit decimal computer, then λ_j should be coupled with λ_i when

$$|\lambda_i - \lambda_j| \max(|s_i|, |s_j|) \leqq 10^{(t'-t)}\|A\|_F, \tag{14.15}$$

where $\|A\|_F$ rather than $\|A\|_2$ is used since a practical criterion is required. This criterion has been applied on KDF9 and found to be quite sound. We may illustrate this in action by means of a simple example. Consider the matrix

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ \varepsilon & 0 & 1 \end{bmatrix}. \tag{14.16}$$

The eigenvalues are $1 + \varepsilon^{1/3}$, $1 + \omega\varepsilon^{1/3}$, $1 + \omega^2\varepsilon^{1/3}$, where ω is a complex cube root of unity. The separation is $\varepsilon^{1/3}\sqrt{3}$, and hence when ε is of the order of machine precision, the eigenvalues will not appear unduly close. But the left-hand eigenvector corresponding to $\varepsilon^{1/3}$ is $[\varepsilon^{2/3}, \varepsilon^{1/3}, 1]$, and the right-hand eigenvector is $[1, \varepsilon^{1/3}, \varepsilon^{2/3}]$, and hence the corresponding $s_1 = 3\varepsilon^{2/3}/(1 + \varepsilon^{2/3} + \varepsilon^{4/3})$ with similar results for the other eigenvalues. Hence $(\lambda_1 - \lambda_2)s_1 \doteq 3\sqrt{3}\varepsilon$, and this product fully exposes the danger. These two eigenvalues would be grouped together even if one were making the tolerance very lax.

One difficulty encountered in experimentation with algorithms for finding invariant subspaces is that of obtaining a correct orthogonal basis against which to test computed subspaces except in the case of rather artificially constructed matrices. In practice, we have found it useful to work with A itself and with $\tilde{A}$ such that $\tilde{a}_{ij} = a_{n+1-i,n+1-j}$, i.e., $\tilde{A}$ is the reflection of A in its center point. Eigenvectors, etc., of $\tilde{A}$ are merely those of A with components in the reverse order. If A and $\tilde{A}$ are solved by the same algorithm, then one can compare orthogonal bases obtained with the two matrices. At least one computed subspace has an error which is of the order of magnitude of the angle between the two computed subspaces. Where it has been possible to determine a correct basis by independent means, the error in each of the computed subspaces has proved to be of the same order of magnitude as the angle between them. One might expect this to be true generally unless there is some special reason for errors to be correlated in some way.

For matrices with well-defined J.c.f.'s, the orthogonal bases determined by an algorithm based on the above have been correct almost to working accuracy. Even if one takes t' almost equal to t, only the eigenvalues associated with multiple roots have been grouped together.

The results obtained with the Frank matrices are interesting. For $n = 16$, the 9 smallest computed eigenvalues and their true values are given in Table 1. Six of the computed values are complex and with imaginary parts which are quite comparable with the real parts. Only with λ_{10} do we begin to have any significant

accuracy, and λ_9 has four correct figures. The largest eigenvalues were given very accurately.

TABLE 1

Computed eigenvalues	True eigenvalues
$\lambda_{16} = -0.02710 + i\,(0.04506)$	$\lambda_{16} = 0.02176$
$\lambda_{15} = -0.02710 - i\,(0.04506)$	$\lambda_{15} = 0.03133$
$\lambda_{14} = 0.06121 + i\,(0.09907)$	$\lambda_{14} = 0.04517$
$\lambda_{13} = 0.06121 - i\,(0.09907)$	$\lambda_{13} = 0.06712$
$\lambda_{12} = 0.1882 + i\,(0.06248)$	$\lambda_{12} = 0.1051$
$\lambda_{11} = 0.1882 - i\,(0.06248)$	$\lambda_{11} = 0.1775$
$\lambda_{10} = 0.3342$	$\lambda_{10} = 0.3307$
$\lambda_9 = 0.6809$	$\lambda_9 = 0.6809$
$\lambda_8 = 1.469$	$\lambda_8 = 1.469$

Orthogonal bases were computed for subspaces of dimensions 2, 4, 6, 7, 8, 9 obtained by grouping the corresponding number of smallest eigenvalues together. (Notice we did not compute spaces of dimension 3, 5 since conjugate pairs were always kept together in order to be able to work in the real field.) The angles between the computed bases and the true subspace are given in Table 2. The subspaces of order 2 and 4 are scarcely of any significant accuracy, but that of order 6 is correct to about 3 decimals and that of order 7 to almost six decimals. Notice that this accuracy in the subspace is attained, although some of the λ_i are very poor. (It should be emphasized, though, that every computed λ_i is an eigenvalue of some $A + E$, with $\|E\|/\|A\|$ of the order of 2^{-39}.)

TABLE 2

Dimension	Angle between computed and true subspace
2	3.05×10^{-2}
4	1.73×10^{-2}
6	6.23×10^{-4}
7	1.74×10^{-6}
8	1.73×10^{-8}
9	2.67×10^{-10}

We may look at these results from an alternative point of view. If the matrix F_{16} is regarded as having relative errors of order 10^{-12} in its elements, then the invariant subspace corresponding to its two smallest elements is scarcely determined at all, while that corresponding to its smallest 7 eigenvalues for example is determined to about six decimals.

15. Inverse iteration and ill-conditioned eigensystems. Inverse iteration is one of the main tools used in practice for the calculation of eigenvectors from computed eigenvalues. The motivation for inverse iteration, due to Wielandt [23], springs from the observation that if A is a matrix with a complete set of eigenvectors x_i,

then an arbitrary vector may be expressed in the form

$$y = \sum_{i=1}^{n} \alpha_i x_i \tag{15.1}$$

and hence

$$(A - kI)^{-1} y = \sum_{i=1}^{n} \alpha_i x_i/(\lambda_i - k). \tag{15.2}$$

If $|\lambda_j - k| \ll |\lambda_i - k|$ $(i \neq j)$, the components of x_j will be very much larger than the coefficients of the remaining x_i, unless the vector y happens to be very deficient in x_j. If, in particular, k is a very accurate approximation to λ_j, the right-hand side of (15.2) may be written in the form

$$\left(\frac{1}{\lambda_j - k}\right)\left[\alpha_j x_j + \sum_{i \neq j} \alpha_i(\lambda_j - k)x_i/(\alpha_i - k)\right] \tag{15.3}$$

and the normalized form of this vector will be x_j to very high accuracy. However, for nonnormal matrices, a computed λ may not be particularly near to *any* eigenvalue, and it appears that one can no longer expect such a spectacular performance in one iteration.

Varah was the first to point out that this is not so. The simplest way to see this is to forget about the expansion of y and concentrate directly on the solution of $(A - \lambda I)z = y$, where $\|y\|_2 = 1$. We may write

$$(A - \lambda I)z/\|z\|_2 = y/\|z\|_2, \quad w = z/\|z\|_2, \quad y/\|z\|_2 = r \tag{15.4}$$

giving

$$(A - \lambda I)w = r, \quad \|w\|_2 = 1, \quad \|r\| = 1/\|z\|_2 = \varepsilon \quad \text{(say)}. \tag{15.5}$$

The first of equations (15.5) may be expressed in the form

$$(A - rw^H)w = \lambda w, \tag{15.6}$$

and hence λ and w are an exact eigenvalue of eigenvector of the matrix $A - rw^H$. Since $\|rw^H\|_2 = \|r\|_2 = \varepsilon$, it is evident that if $\|z\|_2$ is "large", λ and w are satisfactory since they are exact for a neighboring matrix.

Now if we start with a value of λ which is an exact eigenvalue of $A + E$, then however poor λ may otherwise be,

$$(A + E - \lambda I)q = 0 \quad \text{for some } \|q\|_2 = 1. \tag{15.7}$$

Hence $(A - \lambda I)q = -Eq$ and if one takes $y = -Eq/\|Eq\|_2$, the solution of $(A - \lambda I)z = y$ is $z = q/\|Eq\|_2$ and $\|z\| \geqq 1/\|E\|_2$. With this choice of y, then, we obtain a very large z in one iteration, and the corresponding $w = z/\|z\|_2$ is a satisfactory eigenvector corresponding to λ. Obviously, if we take as initial y an arbitrary unit vector, the probability of it being *very* deficient in the vector $-Eq/\|Eq\|_2$ is very small and hence inverse iteration will "work" in one iteration with almost any starting vector.

However, Varah also produced an argument which suggested that when λ is related to an ill-conditioned eigenvalue, there are severe disadvantages in

performing more than one step of inverse iteration, and a satisfactory analysis of the phenomenon was subsequently given by Wilkinson [27]. It is instructive to analyze this phenomenon in terms of the S.V.D. decomposition. We showed in § 5 that if λ_i is an ill-conditioned eigenvalue, the associated s_i is small and the matrix X of eigenvectors has a small singular value $\sigma_n < |s_i|$. If the S.V.D. of X is

$$X = U\Sigma V^H, \qquad XV = U\Sigma, \tag{15.8}$$

then

$$u_n = \frac{Xv_n}{\sigma_n} = \frac{1}{\sigma_n}[\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n], \quad \text{where } \alpha_i = v_{in}, \tag{15.9}$$

and hence the unit vector u_n expanded in terms of the x_i has very large coefficients. If we take an arbitrary vector y, it can be expressed in the form

$$y = \beta_1 u_1 + \cdots + \beta_n u_n, \tag{15.10}$$

where the β_i are distributed in a natural way. When it is transformed to its expansion in terms of the x_i, we have

$$y = \left[\frac{\alpha_1\beta_n}{\sigma_n} + \cdots\right]x_1 + \left[\frac{\alpha_2\beta_n}{\sigma_n} + \cdots\right]x_2 + \cdots + \left[\frac{\alpha_n\beta_n}{\sigma_n} + \cdots\right]x_n, \tag{15.11}$$

and in general all the coefficients of the x_i will be very large but will be in ratios which are independent of the β_i, provided β_n is not small. From (15.11),

$$\begin{aligned} z = (A - \lambda I)^{-1}y = &\left[\frac{\alpha_1\beta_n}{\sigma_n} + \cdots\right]\frac{x_1}{\lambda_1 - \lambda} + \left[\frac{\alpha_2\beta_n}{\sigma_n} + \cdots\right]\frac{x_2}{\lambda_2 - \lambda} + \cdots \\ &+ \left[\frac{\alpha_n\beta_n}{\sigma_n} + \cdots\right]\frac{x_n}{\lambda_n - \lambda}, \end{aligned} \tag{15.12}$$

and z will in general be a large vector for two reasons: first, because σ_n is small and second, because usually one of the $(\lambda_i - \lambda)$ will be moderately small (though not usually pathologically so). Now when z is normalized prior to doing the second iteration, the coefficients of the x_i in this normalized z will no longer be special in the way that they were in the first "arbitrary" y.

In fact, the normalized vector will be essentially

$$\begin{aligned} &\frac{\lambda_i - \lambda}{\lambda_1 - \lambda}[\alpha_1 + \cdots]x_1 + \frac{\lambda_i - \lambda}{\lambda_2 - \lambda}[\alpha_2 + \cdots]x_2 + \cdots + [\alpha_i + \cdots]x_i \\ &\qquad + \cdots + \frac{\lambda_i - \lambda}{\lambda_n - \lambda}[\alpha_n + \cdots]x_n, \end{aligned} \tag{15.13}$$

and the coefficients of the x_i will be of order unity. In the first vector these coefficients were all large but canceled out to give a vector of normal size. Consequently, in the second step of inverse iteration, the growth in size will come only from the comparative smallness of a $\lambda_i - \lambda$ and will not be reinforced by the smallness of σ_n. This will be true of all subsequent steps unless at the rth step all the quantities $((\lambda_i - \lambda)/(\lambda_j - \lambda))^r$ are almost equal, when the normalized value will have large

components of each of the x_i in the same ratios as in the first vector. In this case, every rth iteration will give a large growth and consequently a satisfactory vector. This situation will usually occur when A has an elementary divisor of degree r. Varah has effectively used this behavior of the iterates to give information on the structure of the J.c.f. of A [21].

The analysis may be carried out in an alternative way which is also instructive. We observe first that if λ_i is an exact eigenvalue of A, then $A - \lambda_i I$ is singular, and if

$$(A - \lambda_i I) = U\Sigma V^H, \tag{15.14}$$

then $\sigma_n = 0$. Consequently,

$$(A - \lambda_i I)v_n = 0, \qquad u_n^H(A - \lambda_i I) = 0, \tag{15.15}$$

and v_n and u_n are normalized right-hand and left-hand eigenvectors of A, with $u_n^H v_n = s_i$.

Now suppose λ_i is an exact eigenvalue of $A + E$; then $\sigma_n(A - \lambda_i I) \leqq \|E\|_2$. If we now write

$$A - \lambda_i I = U\Sigma V^H \quad \text{and} \quad (A - \lambda_i I)v_s = \sigma_s u_s, \tag{15.16}$$

then $\sigma_n \leqq \|E\|$. An arbitrary unit vector y may now be expanded in the form

$$y = \sum \alpha_j u_j, \quad \text{with } \|\alpha\|_2 = 1 \tag{15.17}$$

and

$$z = (A - \lambda I)^{-1}y = \sum \frac{\alpha_j v_j}{\sigma_j}. \tag{15.18}$$

The coefficient of v_n is α_n/σ_n, and

$$|\alpha_n/\sigma_n| \geqq \alpha_n/\|E\|. \tag{15.19}$$

Unless y is accidentally deficient in u_n, the full growth takes place in the first iteration. The normalized z is essentially of the form

$$v_n + \sum_1^{n-1} \gamma_i v_i, \tag{15.20}$$

where the γ_i are small. To see the effect of the second iteration, one requires an expansion in terms of the u_i rather than the v_i, and we now show that in this expansion the coefficient of u_n is small. Indeed, since $u_n^H v_n$ is roughly s_i from the previous argument, and all the γ_i are small, this is immediately obvious. The normalized z is therefore an unfortunate vehicle for inverse iteration since it is deficient in u_n.

16. Improvement of an invariant subspace. Suppose A has been reduced to upper triangular form T by a unitary similarity X with a group of associated λ_i in the s leading diagonal positions of T. We then have, for the computed X and T,

$$AX - XT = E. \tag{16.1}$$

The error analysis guarantees that E will be almost negligible to working accuracy.

Each element of the matrix E may be determined in practice by accumulating the whole of the inner product involved in double-precision before rounding. If $F = X^{-1}E$, then

$$X^{-1}AX = T + X^{-1}E = T + F, \tag{16.2}$$

and since the computed X is almost exactly orthogonal, one can compute F via X^TE. From an invariant subspace of $T + F$ one can improve the corresponding subspace of A itself. We partition $T + F$ in the form

$$\begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} + \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}, \tag{16.3}$$

where T_{11} contains the grouped eigenvalues. The relevant invariant subspace of T is spanned by the first s columns of I, and hence if we write

$$\left(\begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} + \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}\right)\begin{bmatrix} I \\ \overline{Y} \end{bmatrix} = \begin{bmatrix} I \\ \overline{Y} \end{bmatrix}[T_{11} + G_{11}], \tag{16.4}$$

$[I \quad Y^T]$ gives the improved subspace. From (16.4), neglecting second order quantities,

$$T_{11} + F_{11} + T_{12}Y = T_{11} + G_{11}, \quad T_{22}Y + F_{21} = YT_{11}, \tag{16.5}$$

and Y is the solution of

$$[T_{22}Y - YT_{11}] = -F_{21}. \tag{16.6}$$

The matrix Y may be determined column by column via the relations

$$T_{22}y_1 - t_{11}y_1 = -f_1, \qquad (T_{22} - t_{11}I)y_1 = -f_1, \tag{16.7}$$

$$T_{22}y_2 - t_{12}y_1 - t_{22}y_2 = f_2, \qquad (T_{22} - t_{22}I)y_2 = -f_2 + t_{12}y_1. \tag{16.8}$$

In general, the rth column of Y is the solution of a triangular system of equations with matrix $(T_{22} - t_{rr}I)$.

From Y one can determine G_{11} via (16.5).

If one includes the second order terms, then (16.6) becomes

$$[T_{22}Y - YT_{11}] = -F_{21} + [-F_{22}Y + Y(T_{12}Y + F_{11} + F_{12}Y)], \tag{16.9}$$

and after solving (16.6), an improved right-hand side is that in (16.9) in which the computed Y is used. In this way, Y may be repeatedly improved by iteration.

However, there is little point in this. The matrix F is not known exactly. There are errors made in computing E in the first place and further errors in computing $X^{-1}E$, and here no purpose is served in computing Y accurately. In (16.3) we have purposely refrained from writing

$$\begin{bmatrix} \overline{T}_{11} & \overline{T}_{12} \\ F_{21} & \overline{T}_{22} \end{bmatrix}, \tag{16.10}$$

where $\overline{T}_1 = T_{11} + F_{11}, \quad \overline{T}_{12} = T_{12} + F_{12}, \quad \overline{T}_{22} = T_{22} + F_{22},$

although this would have simplified the expressions. This is because it is necessary to keep the F matrix separate from the T matrix on the computer. The information

in F would promptly be lost if the addition were carried out. However, there are obviously slight advantages in replacing (16.6) by

$$(\bar{T}_{22}Y - Y_{11}\bar{T}_{11}) = -F_{21}. \tag{16.11}$$

The improved subspace X_1 is now $\tilde{X}_1 = X_1 + X_2 Y$. It is no longer quite orthogonal, but this is of no importance. If one wishes to continue with the refinement of the subspace, one should return to the computation of the residual via the relation

$$A[\tilde{X}_1 \mid X_2] - [\tilde{X}_1 \mid X_2][\tilde{T}] = \tilde{E}, \tag{16.12}$$

where

$$\tilde{T} = \begin{bmatrix} T_{11} + G_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}. \tag{16.13}$$

The new $\tilde{E}$ will not be smaller than E in general, but the next correction to the subspace will be. If $\tilde{X} = [\tilde{X}_1 \mid X_2]$, then

$$\tilde{X}^{-1}A\tilde{X} - \tilde{T} = (\tilde{X})^{-1}\tilde{E} = \tilde{F}, \tag{16.14}$$

and one can still use the approximation $(\tilde{X})^{-1} = X^T$. When computing the new correction Y, the equations corresponding to (16.6) will be

$$[T_{22}\tilde{Y} - \tilde{Y}\tilde{T}_{11}] = -\tilde{F}_{21}, \tag{16.15}$$

and $\tilde{T}_{11} = T_{11} + S_{11}$ is no longer upper triangular. However, we may use T_{11} in place of $\tilde{T}_{11}$ since YS_{11} will be of second order.

The process of iterative refinement is wholly analogous to that used with linear equations (see, e.g., [25, Chap. 4]). In general, we can continue until we have a basis which is correct to working accuracy. Indeed, at the time when the process terminates, the new X_1 will be obtained in terms of the old X_1 and the original X_2 by the relation

$$X_1 \text{ (new)} = X_1 \text{ (old)} + X_2 Y, \tag{16.16}$$

where $X_2 Y$ is small. The true sum on the right-hand side will be accurate to more than the working precision.

The final X_1 will not have orthogonal columns, but they will be almost orthogonal. If true orthogonality is required, no appreciable loss of accuracy will occur when the Schmidt orthogonalization process is performed.

Acknowledgments. The authors wish to thank Mrs. G. Peters of the National Physical Laboratory and Dr. Peter Businger of the Bell Telephone Laboratories for the computational work on invariant subspaces, and Dr. B. Heap and Mr. Clive Hall also of the National Physical Laboratory for the production of Fig. 1 on the computer KDF9.

REFERENCES

[1] C. DAVIS AND W. M. KAHAN, *The rotation of eigenvectors by a perturbation. III*, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.

[2] D. K. FADDEEV AND V. N. FADDEEVA, *Computational Methods of Linear Algebra*, W. H. Freeman, San Francisco and London, 1963.

[3] G. H. GOLUB AND W. M. KAHAN, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.
[4] G. H. GOLUB AND C. REINSCH, *Singular value decomposition and least squares solutions*, Numer. Math., 14 (1970), pp. 403–420.
[5] R. T. GREGORY, *Defective and derogatory matrices*, this Review, 2 (1960), pp. 134–139.
[6] A. S. HOUSEHOLDER, *The Theory of Matrices in Numerical Analysis*, Ginn (Blaisdell), Boston, 1964.
[7] W. M. KAHAN, *Conserving confluence curbs ill-condition*, Computer Sci. Tech. Rep. 6, Univ. of Calif., 1972.
[8] T. KATO, *Perturbation Theory for Linear Operators*, Springer-Verlag, Berlin, 1966.
[9] G. KREISSELMEIER, *On the determination of the Jordan form of a matrix*, IEEE Trans. Automatic Control, AC-18 (1973), pp. 686–687.
[10] V. N. KUBLANOVSKAYA, *On a method of solving the complete eigenvalue problem for a degenerate matrix*, Ž. Vyčisl. Mat. i Mat. Fiz., 6 (1966), pp. 611–620 = USSR Computational Math and Math. Phys., 6 (1968), no. 4, pp. 1–14.
[11] P. LANCASTER, *Theory of Matrices*, Academic Press, New York, 1969.
[12] B. N. PARLETT AND C. REINSCH, *Balancing a matrix for calculation of eigenvalues and eigenvectors*, Numer. Math., 13 (1969), pp. 293–304.
[13] G. PETERS AND J. H. WILKINSON, *Studies in Numerical Analysis*, B. K. P. Scaife, ed., Academic Press, New York and London, 1974.
[14] A. RUHE, *An algorithm for numerical determination of the structure of a general matrix*, BIT, 10 (1970), pp. 196–216.
[15] ———, *Perturbation bounds for means of eigenvalues and invariant subspaces*, Ibid., 10 (1970), pp. 343–354.
[16] ———, *Properties of a matrix with a very ill-conditioned eigenproblem*, Numer. Math., 15 (1970), pp. 57–60.
[17] G. W. STEWART, *On the sensitivity of the eigenvalue problem*, SIAM J. Numer. Anal., 9 (1972), pp. 669–686.
[18] ———, *Error bounds for invariant subspaces of closed operators*, Ibid., 8 (1971), pp. 796–808.
[19] ———, *Introduction to Matrix Computations*, Academic Press, New York and London, 1973.
[20] J. M. VARAH, *Rigorous machine bounds for the eigensystem of a general complex matrix*, Math. Comp., 22 (1968), pp. 793–801.
[21] ———, *Computing invariant subspaces of a general matrix when the eigensystem is poorly conditioned*, Ibid., 24 (1970), pp. 137–149.
[22] ———, *Invariant subspace perturbations for a non-normal matrix*, IFIP Conf., 1971.
[23] H. WIELANDT, *Bestimmung höherer eigenwerte durch gebrochene iteration*, Ber, B44/J/37 der Aerodynamischen Versuchanstalt Göttingen, 1944.
[24] J. H. WILKINSON, *Rounding Errors in Algebraic Processes*, Her Majesty's Stationery Office, Prentice-Hall, Englewood Cliffs, N.J., 1963.
[25] ———, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1968.
[26] ———, *Note on matrices with a very ill-conditioned eigenproblem*, Numer Math., 19 (1972), pp. 176–178.
[27] ———, *Inverse iteration in theory and in practice*, Istituto Nazionale di Alta Matematica Symposia Mathematica, 10 (1972).
[28] J. H. WILKINSON AND C. REINSCH, *Handbook for Automatic Computation, Vol. II: Linear Algebra*, Springer-Verlag, Berlin and New York, 1971.
[29] A. WINTNER AND F. D. MURNAGHAN, *A canonical form for real matrices under orthogonal transformations*, Proc. Nat. Acad. Sci. U.S.A., 17 (1931), pp. 417–420.