JOURNAL OF COMPUTATIONAL PHYSICS 32, 377–402 (1979)

The Reduction of a General Complex Matrix to a Condensed Form by Bounded Single Element Transformations

DOUGLAS F. HAGER AND ROY G. GORDON

Department of Chemistry, Harvard University, Cambridge, Massachusetts 02138

Received October 27, 1978

An algorithm is proposed to reduce by elementary similarity transformations a general complex matrix to the most condensed form obtainable. As many off-diagonal elements of the transformed matrix are made to vanish as is possible, subject to the requirement that numerical stability be maintained. For any input matrix, all elements below the diagonal are transformed to zero. Using a two-part algorithm, elements above the diagonal are also eliminated when this can be done with numerical stability. The first part attempts to eliminate all elements above the diagonal; if this is possible, a diagonal transformed matrix is obtained. When this first step fails to zero all elements above the diagonal, the second part of the algorithm attempts to eliminate all elements which remain above the first superdiagonal elements; if this is possible, a matrix in Jordan canonical form is obtained. In more difficult cases some non-zero matrix elements above the superdiagonal remain because their elimination would have destroyed the numerical stability of the results. These condensed matrices are useful in simplifying the formation of various matrix functions. Because of the numerical stability of the algorithm one can be confident of the accuracy of these matrix functions.

I. INTRODUCTION

Computations in physics and chemistry often require the calculation of quantities which are most concisely expressed as functions of matrices. In many cases, these matrices depend on a scalar parameter, and the appropriate matrix function must be evaluated for a large number of values of the scalar parameter. The two examples of such calculations with which we have been concerned are

(a) computations of functions $\mathscr{I}(\omega)$ of the form $\mathscr{I}(\omega) = \mathbf{d}_1{}^T(\mathbf{A} + \mathbf{I}\omega)^{-1}\,\mathbf{d}_2$ (or, equivalently, the solution of the set of linear equations $(\mathbf{A} + \mathbf{I}\omega)\mathbf{x} = \mathbf{d}_2$) for several values of the parameter ω ($\mathbf{A}$ is a general complex matrix; $\mathbf{d}_1{}^T$ and $\mathbf{d}_2$ are row and column vectors, respectively; $\mathbf{x}$ is a column vector; $\mathbf{I}$ is the unit matrix; and ω is a scalar) [7–13, 17]; and

(b) functions $G(t)$ of the form $\mathbf{d}_1{}^T \exp(\mathbf{A}t)\,\mathbf{d}_2$ [1, 7, 12] for many values of the scalar parameter t.

Direct computation of these functions for large matrices and for many different parameter values can be prohibitively costly of computer time. Whenever a complete eigenanalysis of $\mathbf{A}$ is possible (a sufficient but not necessary condition for this is that

0021-9991/79/090377–26$02.00/0

the eigenvalues be distinct), $\mathbf{A}$ may be transformed to diagonal form, $\mathbf{D} = \mathbf{U}^{-1}\mathbf{A}\mathbf{U}$, where $\mathbf{U}$ is a matrix consisting of the eigenvectors of $\mathbf{A}$ and $\mathbf{U}^{-1}$ is its inverse. In such a case,

$$\mathscr{I}(\omega) = \mathbf{d}_1{}^T[\mathbf{U}(\mathbf{D} + \mathbf{I}\omega)\,\mathbf{U}^{-1}]^{-1}\,\mathbf{d}_2 \tag{I-1}$$

and

$$G(t) = \mathbf{d}_1{}^T \exp\{\mathbf{U}\mathbf{D}\mathbf{U}^{-1}t\}\,\mathbf{d}_2\,. \tag{I-2}$$

These relations may be rewritten as

$$\mathscr{I}(\omega) = (\mathbf{d}_1{}^T\mathbf{U})[\mathbf{D} + \mathbf{I}\omega]^{-1}(\mathbf{U}^{-1}\mathbf{d}_2) \tag{I-3}$$

and

$$G(t) = (\mathbf{d}_1{}^T\mathbf{U}) \exp\{\mathbf{D}t\}(\mathbf{U}^{-1}\mathbf{d}_2). \tag{I-4}$$

In these forms, $\mathscr{I}(\omega)$ and $G(t)$ may be trivially evaluated, with little cost of computer time, for a great many values of the scalar parameters ω or t. Several methods [13] have been proposed for the evaluation of the eigenvalues and eigenvectors of diagonalizable complex matrices. The most popular method involves the use of the *QR* algorithm [2, 3] to find the eigenvalues and the use of inverse iteration [19] to calculate the corresponding matrix of eigenvectors. This scheme has been adapted by Gordon and co-workers [7–13] to simplify the calculation of functions of the form of $\mathscr{I}(\omega)$ and $G(t)$.

Occasionally, however, physical problems arise in which the eigenvalues of the matrix $\mathbf{A}$ become degenerate or nearly degenerate. In such cases, the matrix $\mathbf{A}$ may still be similar to a diagonal matrix. Unfortunately, inverse iteration may be unable to calculate the complete set of eigenvectors needed to evaluate the functions $\mathscr{I}(\omega)$ or $G(t)$ [13]. Because of the relative effortlessness of the calculation of these functions when $\mathbf{A}$ has been diagonalized, it is beneficial computationally to develop an algorithm which can, in these cases, insure the determination of a complete set of eigenvectors.

The degeneracy or near degeneracy of sets of eigenvalues may also, in many cases, mean that the matrix $\mathbf{A}$ cannot be numerically diagonalized at all; the simplest form to which $\mathbf{A}$ may be transformed is then Jordan canonical form [15]. At least two algorithms [6, 16] have been developed to attack the problem of obtaining the Jordan canonical form ($\mathbf{J}$) similar to a general complex matrix and of obtaining the corresponding complete set of eigenvectors and generalized eigenvectors which make up $\mathbf{U}$ in the similarity relation $\mathbf{J} = \mathbf{U}^{-1}\mathbf{A}\mathbf{U}$. The problem of finding the unique Jordan canonical form similar to a given complex matrix is a difficult one [6]. The similarity transformation matrices $\mathbf{U}$ and $\mathbf{U}^{-1}$, not being unitary matrices, have no a priori bounds on their norms; the calculation of the Jordan canonical form may then, in some cases, be numerically unstable. (We will discuss the concept of "stability" in the context of our problem in the next section.) The attainment of the Jordan canonical form similar to the matrix $\mathbf{A}$ is, nonetheless, most desirable whenever numerical stability can be maintained because the calculation of $(\mathbf{J} + \mathbf{I}\omega)^{-1}$ or $\exp\{\mathbf{J}t\}$ is nearly as effortless as the corresponding calculation for diagonal forms. Unfortunately, serious doubt has been cast on the advisability of trying to determine the Jordan canonical form of a matrix for a general complex matrix [6].

Finally, in cases where Jordan canonical form cannot be obtained while maintaining numerical stability, it is still useful to transform the matrix to block upper triangular form **B**, where the blocks may be diagonal, be of Jordan form, or contain non-zero elements everywhere in the upper triangle of the blocks. Formation of $(\mathbf{B} + \mathbf{I}\omega)^{-1}$ or $\exp\{\mathbf{B}t\}$ then reduces to the problem of finding the inverse or the exponential of the individual upper triangular blocks. Those blocks which are diagonal or of Jordan form may still be trivially inverted or exponentiated for the several values of the scalar parameters ω and t. The general upper triangular blocks must still be inverted or exponentiated by classical methods [15] for each value of the scalar parameters. But in most cases of physical interest these blocks are few in number and always of considerably smaller order than the original matrix. Thus, reaching a condensed block upper triangular form will still cause a considerable saving in the computation time for matrix functions of the type we have discussed; for example, performing an order (N^3) process for five 10×10 blocks is 25 times faster than performing the process for the entire 50×50 matrix.

In the present paper, we propose an algorithm which reduces an arbitrary complex matrix to a "condensed" form, while insuring "numerical stability." The method aims, by unitary and elementary similarity transformations [19], to transform the matrix so as to eliminate as many off-diagonal elements as possible, subject to the restriction that the transformed matrix be similar to a matrix very nearly the same as the original matrix (this notion will be made precise in the next section).

The production of our condensed form is favorable not only because of the computational time saved, but also because we can bound the accumulation of errors introduced by the successive application of our elementary similarity transformations. In contrast, algorithms for reducing a general matrix to Jordan form do not, to our knowledge, give a precise bound on the errors introduced in the calculation of the Jordan canonical form and the corresponding set of eigenvectors and generalized eigenvectors.

Section II discusses the concept of numerical stability as applied to the algorithm we present here. Section III describes our algorithm for obtaining block upper triangular form. Section IV discusses the general problem of reducing the matrix to a more condensed form. Sections V and VI describe the algorithm we present for obtaining the more condensed form. Some numerical examples of the use of our algorithm are given in Section VII. Finally, the algorithm and its advantages are summarized in Section VIII.

II. The Concept of Stability in This Reduction Scheme[1]

As will be explained in Section III, we first perform Householder transformations to obtain a matrix in upper Hessenberg form and then use the QR algorithm to transform the matrix to upper triangular form. The bounding of errors for these two

[1] In this discussion, bold face capital letters will denote matrices and bold face lower case letters will denote vectors.

methods (based on unitary transformations) has been studied in detail [19], and the methods are known to be quite stable. Our analysis will begin with the propagation of errors due to the non-unitary similarity transformations which we describe in Sections III, VI and VII.

The general matrices $\mathbf{A}$ which enter the problems referenced in Section I will have elements which have been experimentally determined or theoretically calculated and will contain some sampling or measurement uncertainties. This suggests that our interest should not lie in obtaining a diagonal or Jordan canonical or block upper triangular form exactly similar to the matrix $\mathbf{A}$. Rather, we desire a condensed form which is similar to a matrix very nearly the same as $\mathbf{A}$, where very nearly means that the differences between $\mathbf{A}$ and the matrix similar to the condensed form are either less than or of the order of the uncertainties in the matrix $\mathbf{A}$ itself (or alternatively, for matrices not arising from such physical problems and whose elements are known exactly, of the magnitude of round-off error in whatever precision of arithmetic is desired for the calculation). Precisely, if the condensed form is denoted by $\mathbf{T}$, we desire the error matrix $\mathbf{K}$, defined by $\mathbf{A} - \mathbf{UTU}^{-1} = \mathbf{K}$, to be smaller than the errors implicit in the matrix $\mathbf{A}$ itself. We define a small error matrix by the requirement that $\|\mathbf{K}\|/\|\mathbf{A}\|$ is smaller than some desired value. ($\|\cdot\|$ denotes a matrix norm; in our calculation we use the norm $\|\mathbf{C}\|_\infty = \max_i \sum_{j=1}^{n} | C_{ij} |$.) With such limitations the calculated function

$$\begin{aligned}\mathscr{I}(\omega) &= (\mathbf{d}_1{}^T\mathbf{U})[\mathbf{T} + \mathbf{I}\omega]^{-1}(\mathbf{U}^{-1}\mathbf{d}_2)\\ &= \mathbf{d}_1{}^T[\mathbf{A}^1 + \mathbf{I}\omega]^{-1}\,\mathbf{d}_2\end{aligned}$$

will differ from the desired function $\mathscr{I}(\omega) = \mathbf{d}_1{}^T[\mathbf{A} + \mathbf{I}\omega]^{-1}\,\mathbf{d}_2$ only because of inherent uncertainties in our knowledge of the matrix $\mathbf{A}$ or because of round-off error of the magnitude found in the precision of arithmetic we use for the calculation. This property is what we will refer to as the numerical stability of our algorithm.

The individual transformations described in Sections IV, VI, and VII are elementary similarity transformations of the type (vii) described by Wilkinson [19] on page 45. We will follow his error analysis as presented on pages 124–126. Denoting the ith similarity matrix as $\bar{\mathbf{H}}_i$, the transformations will be described by

$$\bar{\mathbf{A}}_i = \mathbf{F}_i + \bar{\mathbf{H}}_i^{-1}\bar{\mathbf{A}}_{i-1}\bar{\mathbf{H}}_i$$

where $\mathbf{F}_i$ is the roundoff error introduced by performing the similarity transformation.

After s such transformations,

$$\begin{aligned}&\bar{\mathbf{A}}_s = \mathbf{F} + \mathbf{G}_1^{-1}\mathbf{A}_0\mathbf{G}_1\,, \quad \text{where}\\ &\mathbf{G}_i \equiv \bar{\mathbf{H}}_i \cdots \bar{\mathbf{H}}_s \quad \text{and}\\ &\mathbf{F} \equiv \mathbf{F}_s + \mathbf{G}_s^{-1}\mathbf{F}_{s-1}\mathbf{G}_s + \cdots + \mathbf{G}_2^{-1}\mathbf{F}_1\mathbf{G}_2\,.\end{aligned}$$

Defining $\mathbf{K}$ by

$$\mathbf{K} \equiv \mathbf{G}_1\mathbf{F}\mathbf{G}_1^{-1},$$

we may rewrite this as

$$\bar{\mathbf{A}}_s = \mathbf{G}_1^{-1}(\mathbf{K} + \mathbf{A}_0)\,\mathbf{G}_1\,, \qquad \text{where}$$

$$\mathbf{K} \equiv \mathbf{L}_s\mathbf{F}_s\mathbf{L}_s^{-1} + \mathbf{L}_{s-1}\mathbf{F}_{s-1}\mathbf{L}_{s-1}^{-1} + \cdots \mathbf{L}_1\mathbf{F}_1\mathbf{L}_1^{-1}$$

and

$$L_i \equiv \bar{\mathbf{H}}_1\bar{\mathbf{H}}_2 \cdots \bar{\mathbf{H}}_i\,.$$

We may calculate and bound the norm of **K** as

$$\begin{aligned} \| \mathbf{K} \| &= \left\| \sum_{i=1}^{s} \mathbf{L}_i\mathbf{F}_i\mathbf{L}_i^{-1} \right\| \\ &\leqslant \sum_{i=1}^{s} \| \mathbf{L}_i\mathbf{F}_i\mathbf{L}_i^{-1} \|. \\ &\leqslant \sum_{i=1}^{s} \| \mathbf{L}_i \| \, \| \mathbf{L}_i^{-1} \| \, \| \mathbf{F}_i \|. \end{aligned}$$

We require that at each step of our calculations

$$\| \mathbf{L}_i \| \, \| \mathbf{L}_i^{-1} \| \leqslant N_L^2.$$

Then

$$\| \mathbf{K} \| \leqslant N_L^2 \sum_{i=1}^{s} \| \mathbf{F}_i \|.$$

$\mathbf{F}_i$ is merely the matrix of round-off errors produced at step i. We assume that for computations on a machine with t binary bits for each number,

$$\| \mathbf{F}_i \| \leqslant f(i, n)\, 2^{-t} \, \| \bar{\mathbf{A}}_{i-1} \|$$

[19, p. 120 ff.], where $f(i, n)$ is related to the number of operations required at the ith step for an $n \times n$ matrix. $f(i, n)$ is some smooth function of the parameters i and n. If we require at each step of our calculation that

$$\| \bar{\mathbf{A}}_k \| \leqslant N_A \, \| \mathbf{A}_0 \|,$$

we find

$$\| \mathbf{K} \| \leqslant N_L^2 N_A \, \| \mathbf{A}_0 \| \, 2^{-t} \sum_{j=1}^{s} f(j, n).$$

$\sum_{j=1}^{s} f(j, n)$ will be related to the number of arithmetic operations required to obtain $\bar{\mathbf{A}}_s$. Each operation described in the following sections will require a maximum of

$O(n^2)$ elementary similarity transformations. Each transformation requires $O(n)$ multiplications, leading to the observation of Section VII that each operation requires a maximum of $O(n^3)$ multiplications. Wilkinson [19, p. 148 ff.] shows that for a sequence of $O(n^2)$ such similarity transformations, $\sum_{j=1}^{s} f(j, n)$ grows for large n as $\sum_{j=1}^{n} f(j, n) \cong Kn^{3/2}$, where K is a numerical constant of order unity or slightly greater. For truly random errors a factor of $n^{3/4}$ might well be more realistic than $n^{3/2}$ for large n [19, p. 138]. At any rate,

$$\frac{\| \mathbf{K} \|}{\| A_0 \|} \leqslant N_L^2 N_A\, 2^{-t} \sum_{j=1}^{s} f(j, n).$$

If our transformations were, instead of elementary similarities, *exactly*[2] unitary transformations

$$\frac{\| \mathbf{K} \|}{\| \mathbf{A}_0 \|} \lesssim 2^{-t} \sum_{j=1}^{s} f(j, n).$$

We will in the following examples use the rule that N_L and N_A should be chosen so that with double precision arithmetic (16 digits on the IBM 360/91) the bound on $\| \mathbf{K} \|$ is no worse than we would expect for exactly unitary transformations in single precision (6–7 digits). If our knowledge of the physical process described by $\mathbf{A}_0$ implies that the individual elements themselves or some conservation rule among elements of a given row or column are known to some different precision, the bounds would be tightened or loosened to give the proper level of precision. In choosing the bounds N_L and N_A we will be guided by our knowledge of the appropriate physical problem; they are not arbitrary, but must be determined for each new problem. In our test results and subsequent work, we require that at each step $\| L_i \| \| L_i^{-1} \| \leqslant 10^6$ and $\| \mathbf{A}_i \| \leqslant 10^3 \| \mathbf{A}_0 \|$; any transformations which would violate these restrictions will not be performed. Thus, in double precision,

$$\frac{\| \mathbf{K} \|}{\| \mathbf{A}_0 \|} \leqslant 10^{-7} \sum_{j=1}^{s} f(j, n);$$

the calculation will be at least as good as one with that precision expected for a computation employing exactly unitary transformations in single precision arithmetic.

Because the bounds on the norms of the transformation matrices or the transformed matrix itself may be approached by the acceptance of a single large elementary similarity transformation, an additional test may be applied to each transformation, limiting the norm of the transformation to a value less than that necessary to obey the bound applied to the accumulated trasnformation matrices. This test is not necessary to the stability of the algorithm we apply. But it is meant to prevent a single pathological transformation from increasing the bounds to a level such that transformations characterized by a smaller norm cannot be performed consistent

[2] An *exactly* unitary matrix $\mathbf{U}$ is one which satisfies the relation $\mathbf{UU}^H \equiv \mathbf{I}$ exactly. ($\mathbf{U}^H$ denotes the Hermitian conjugate of $\mathbf{U}$; $\mathbf{I}$ is the unit matrix.)

with maintaining numerical stability. In the test calculations we describe later we require both that each transformation have a norm less than $N_T \equiv (N_L)^{1/2} + 1$, and that the norms of the accumulated transformations and of the transformed matrix satisfy the inequalities previously proposed. This will insure the preservation of numerical stability while allowing no one pathological transformation to be accepted which technically preserves numerical stability but prevents further (and smaller) transformations which would lead to a more condensed final form of the matrix.

III. Preliminary Steps in the Reduction Scheme

The first three steps in our scheme to transform a general $n \times n$ complex matrix **A** to condensed form are identical to those used in well-known methods if diagonalizing a general complex matrix:

(i) Equilibration of the matrix using diagonal similarity transformations to reduce the norm of the matrix. The algorithm is due to Osborne [18].

(ii) Householder unitary similarity transformations [19, pp. 347–351] are used to eliminate the lowest $n - (\ell + 1)$ elements of column ℓ for $1 \leqslant \ell \leqslant n - 2$. The transformed matrix has only zero elements below the first subdiagonal (this is known as upper Hessenberg form) when this process terminates.

(iii) The matrix in upper Hessenberg form is brought into upper triangular form by QR transformations [19], which are also of the unitary similarity form.

The result of these calculations is an upper triangular matrix **T** and the transformation matrix $\mathbf{U}(\mathbf{T} = \mathbf{U}^{-1}\mathbf{A}\mathbf{U})$. The inverse of **U** is immediately available because it is a product of the transposes of the unitary transformations in stages (ii) and (iii), and the diagonal similarity transformation of stage (i). This inverse is required both for transforming the vectors discussed in Section I, and for calculating the error bounds of Section II. The eigenvalues of **A** are now known, but further transformations must in general be performed to find the left and right eigenvectors.

Beginning at the $(n - 1, n)$ element, one forms[3] $\mathbf{P}_{n-1,n} = \mathbf{I} + k\mathbf{e}_{n-1}\mathbf{e}_n{}^T$. One obtains trivially the relation $\mathbf{P}^{-1}_{n-1,n} = \mathbf{I} - k\mathbf{e}_{n-1}\mathbf{e}_n{}^T$. If the product $\mathbf{P}^{-1}_{n-1,n}\mathbf{T}\mathbf{P}_{n-1,n} = \mathbf{T}^1$ is formed, one finds that $\mathbf{T}^1$ differs from **T** only by elements in the nth column above the diagonal. In particular $T^{-1}_{n-1,n} = T_{n-1,n} - k(T_{n,n} - T_{n-1,n-1})$. If $T_{n,n} \neq T_{n-1,n-1}$, k may be chosen as $T_{n-1,n}/(T_{n,n} - T_{n-1,n-1})$ and $T^{-1}_{n-1,n}$ is set to zero. If $T_{n,n} = T_{n-1,n-1}$, no such transformation can be performed to zero element $(n - 1, n)$ of the matrix. One can continue this process on rows $n - 2, n - 3,\ldots, 1$ of the matrix to eliminate the upper triangular elements of **T**. Within each row ℓ, one begins with the $(\ell, \ell + 1)$ element, forms $P_{\ell,\ell+1}$ as above, transforms the matrix, then forms $P_{\ell,\ell+2}$, etc. Figure 1 shows the order of the eliminations. This order of transformation is determined by noting that the transformation $P_{i,j}$ affects elements in the ith row

[3] The vector $\mathbf{e}_i$ is a column vector with all zero elements except element i, which is unity. $\mathbf{e}_i{}^T$ denotes the transpose of $\mathbf{e}_i$ (a row vector).

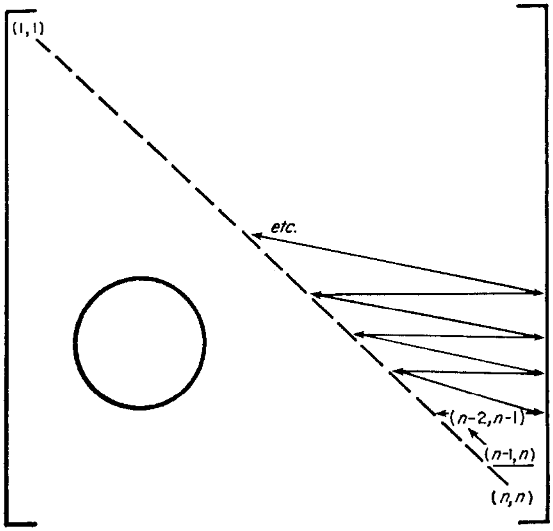

FIG. 1. Order of transformations to eliminate elements linking non-degenerate eigenvalues. The element at which the algorithm begins is underlined.

to the right of (i, j) and elements in the jth column above (i, j) as well as the element (i, j) itself.

The results of this process in exact arithmetic is to leave only those off-diagonal elements linking degenerate eigenvalues. By suitable permutations, eigenvalues linked by off-diagonal terms may be grouped such that the final matrix is a series of upper triangular blocks along the diagonal. These permutations are based on the presence of non-negligible off-diagonal elements linking a set of eigenvalues in a given block, not on the equality of the eigenvalues.

This procedure is straightforward and without hazard when applied using exact arithmetic. In cases where finite precision arithmetic is used and nearly degenerate eigenvalues are present, the decision of when $T_{jj} - T_{ii}$ may be taken as zero must be made with caution. In the three procedures leading to upper triangularization of the entire matrix, all transformations are either diagonal or unitary, insuring numerical stability. If T_{jj} and T_{ii} differ by a quantity small compared to the element T_{ij}, the transformation would introduce large (compared to unity) elements into the calculation of the transformation matrix, decreasing the numerical stability of the method. In addition, one may eliminate elements which link eigenvalues which would be equal if exact arithmetic were used, but differ because of round-off error in previous calculations.

In our algorithm, we decide whether to perform an elimination on the basis of the effect that the transformation may have on the numerical stability (in the sense of Section II) of the method. If the transformation would cause either (1) the product $\| \mathbf{U} \|_\infty \| \mathbf{U}^{-1} \|_\infty$ to exceed the bound set according to the rules of Section II of (2) the quotient $\| \mathbf{T} \|_\infty / \| \mathbf{T}_0 \|_\infty$ ($\mathbf{T}_0$ is the upper triangular matrix first obtained after application of the QR algorithm) to exceed its bound, the transformation is not performed and we move on to try to eliminate the next element in our sequence.

A test based solely on the accumulated affect of the transformations on $\| \mathbf{U} \|_\infty \| \mathbf{U}^{-1} \|_\infty$ and $\| \mathbf{T} \|_\infty / \| \mathbf{T}_0 \|_\infty$ will guarantee that the transformed matrix will be similar to a matrix nearly (in the sense of Section II) equal to the original matrix. However, a single pathologically large transformation may be accepted early in the transformation scheme which raises one or both of the products of norms nearly to its bound. This could prevent the acceptance of several smaller (in the norm sense) transformations encountered later in the calculational scheme. To prevent this, we use an ancillary test: if a single element transformation would be accepted based on the bounds for $\| \mathbf{U} \|_\infty \| \mathbf{U}^{-1} \|_\infty$ and $\| \mathbf{T} \|_\infty / \| \mathbf{T}_0 \|_\infty$, its own norm is compared to a preset tolerance. The choice of this preset tolerance is a bit arbitrary, but our experience has been that a tolerance which rejects transformations where $\| \mathbf{P}_{i,j} \|_\infty \geqslant (N_L)^{1/2} + 1$ prevents a single transformation from pathologically increasing the norms to values so near their bounds as to prevent further transformations from being accepted, while not rejecting a large number of single transformations which could have safely been accepted. Again, we emphasize that this ancillary test in no way alters the stability properties of the method, but is merely a stratagem for attempting to find as sparse a set of upper triangular blocks as possible while maintaining numerical stability. This test may effectively be removed from the algorithm by raising the preset tolerance to $N_L + 1$ or a larger value.

It should be emphasized that if we raise the bounds on $\| U \|_\infty \cdot \| \mathbf{U}^{-1} \|_\infty$ and $\| \mathbf{T} \|_\infty / \| \mathbf{T}_0 \|_\infty$, we will allow more transformations with larger elements $k = T_{ij}/(T_{ij} - T_{ii})$ to be performed. By changing these bounds, the form of the block triangular matrix (which is the result of this series of transformations) may be altered rather arbitrarily. Indeed, with finite precision arithmetic any matrix can be transformed into "diagonal" form by allowing the bounds to be sufficiently large. This illustrates one intrinsic difficulty in applying numerical methods to matrices with degenerate or nearly degenerate eigenvalues.

Because our interest lies in using this method as a tool to obtain numerical results rather than to study the space spanned by the eigenvectors and generalized eigenvectors of the matrix, we choose our tolerance so as to insure maintenance of numerical stability in the sense described in Section II. We may thus err in not transforming the matrix to its most reduced form at this or a later stage. This can slightly increase the difficulty in performing further computations with the matrix and its eigenvectors, but we are at least confident these later computations will not be affected by numerical instability introduced at this stage.

If an $n \times n$ matrix $\mathbf{A}$ possesses n linearly independent eignevectors and infinite precision arithmetic is performed, the matrix should be reduced to diagonal form by this procedure, and no further transformations are necessary; the eigenvalues and the right and left eigenvectors are known. For many such cases, the same result is obtained when finite precision calculations are used. For some matrices with a complete set of eigenvectors, the procedure described above is unable to achieve a full eigenanalysis because of the limitations of finite precision arithmetic. And in all cases where $\mathbf{A}$ is not diagonalizable by similarity transformations, a complete eigenanalysis cannot be obtained by the methods described in this section. In the following sections we propose and test an algorithm for eliminating as many off-diagonal elements as possible from matrices of these latter two "difficult" types.

IV. The Problem of Obtaining Jordan Form

The matrix in block upper triangular form may be reduced further. In theory at least, one can eliminate all the elements above the superdiagonal, leading to Jordan canonical form. In this section, we briefly outline the usual approach to the problem of this reduction so as to gain insight into the fundamental computational difficulties of the problem. Our computational methods are described in Sections V and VI. Although our methods are formally quite different from the approach described here, it can be shown that for the simple, stable problem consisting of no zero or "nearly" zero elements along the superdiagonal, both approaches lead to identical results.

To simplify the exposition, we will assume here and in Section V that the diagonal elements of each block are exactly equal, $T_{ii} = T_{jj}$, for $(i,j)\epsilon$ block. In Section VI, we will study the effects of the diagonal elements being only "nearly" equal as they are in any computation using finite precision arithmetic.

By transforming the original matrix to a matrix consisting of several independent upper triangular blocks, we have reduced our problem from treating an $n \times n$ matrix to that of treating separately several smaller $M_i \times M_i$ upper triangular matrices $\mathbf{V}_i$. Within each block we want to find a set of eigenvectors and generalized eigenvectors $\mathbf{q}_j$ such that $(\mathbf{V}_i - \lambda_i\mathbf{I})\,\mathbf{q}_j = \nu_j\mathbf{q}_{j-1}$, $j = 1, M_i$ where $\nu_j = 0$ or 1 depending on whether $\mathbf{q}_j$ is respectively an eigenvector or generalized eigenvector. If there is no zero along the superdiagonal of $\mathbf{V}_i$, this is a simple set of inhomogeneous linear equations which may be solved recursively for $j = 2, M_i$ with $\mathbf{q}_1 = (1\ 0\ 0 \cdots 0)^T$ and $\nu_j = 1$ $(j = 2, M_i)$.

If there exists one or more zeros along the superdiagonal, there is more than one eigenvector for the block (i.e., the block splits into more than one Jordan block). One must then solve the homogeneous set of equations $(\mathbf{V}_i - \lambda_i\mathbf{I})\,\mathbf{q}^{(0)} = 0$ to find a basis set for the eigenvectors. This is a formidable computational problem for which a reliable algorithm has only recently become available [4, 5]. Even with this algorithm (which requires one to fix an arbitrary criterion for determining what is a zero singular value[4]) one may have difficulty finding (or insuring that one has found) a complete basis set for the eigenvectors. Further, proper linear combinations of the $\mathbf{q}^{(0)}$ must be found to insure that the inhomogeneous system is consistent. This requires solution of an additional set of homogeneous linear equations [15, p. 338]. Finally, the equations must be solved for all the $\mathbf{q}_j$ and an inversion of the matrix $\mathbf{Q} = [\mathbf{q},\ldots, \mathbf{q}_i]$ performed to obtain the inverse transformation matrix.

[4] The singular values of a matrix $\mathbf{A}$ are the non-negative eigenvalues of the Hermitian matrix $\begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix}$ [15, p. 336].

As before, when finite precision arithmetic is used, one may find a "nearly" zero element along the superdiagonal. Such small nonzero elements ϵ_κ along the superdiagonal cause the solution of the inhomogeneous linear equations with a single basis eigenvector to become unstable (elements of order ϵ_κ^{-1} can enter the calculation). On the other hand, an additional basis eigenvector (i.e., another linearly independent solution of $(\mathbf{V}_i - \lambda_i \mathbf{I})\mathbf{q} = \mathbf{0}$) may be difficult to calculate. Finally, it must be remembered that a numerical inversion of the set of eigenvectors and generalized eigenvectors must yet be performed, and we have no guarantee that this process will not introduce large numbers and round-off error to the computations.

We do not intend to discuss these difficulties in great detail. The point we wish to stress is that an intrinsic difficulty in obtaining Jordan canonical form is present whenever a nearly zero element lies along the superdiagonal. (This is similar to the problems encountered in Section III when there is a zero or nearly zero difference between eigenvalues.) If exact arithmetic could be performed, the difficulty would merely be one of performing tedious calculations. But in finite precision calculations, one is faced always with the problem of deciding what is essentially zero and what is non-zero. In the approach we have briefly described in this section, it is not clear how to make this decision or what the effects of the decision may be. The new method described in Sections V and VI will emphasize looking at the effects (i.e., on the norms of the accumulated transformation matrices and of the transformed matrix itself) of this decision to guide us in obtaining a criterion for the decision. This new algorithm cannot, however, insure that Jordan canonical form—with the simplicity it affords subsequent calculations—will be obtained. Nonetheless, we will obtain a condensed form of the matrix which, as described in Section I, will still substantially reduce the time needed to calculate the matrix functions we have cited.

Golub and Wilkinson [6] have formulated an algorithm based on the approach described in this section which is an alternative to the methods we propose in the following. Because their algorithm does not allow one to set a tolerance level for the accumulation of round-off error in a calculation, we find the algorithm we propose more suited to the computational problems described here. On the other hand, their algorithm will give insight into the structure of the eigenspace of a given problem. For our physical and chemical computations, we are not wedded to the need for a complete eigenanalysis; rather we need to reduce the computations to a manageable size (and time) while insuring the retention of numerical stability. The algorithm we propose does this for the problems which we have encountered.

V. Algorithm to Obtain "Condensed" Form: Case of Equal Diagonal Elements in Each Block

In the algorithm proposed here, a series of single element transformations of a form similar to those described in Section III are used. The transformations are performed in order to annihilate as many elements as possible above the superdiagonal in each of the $M_i \times M_i$ blocks, $\mathbf{V}_i$, of the block upper triangular matrix $\mathbf{T}$, which results from the algorithm described in Section III. We begin the transformation scheme in block $\mathbf{V}_1$, then pass to block $\mathbf{V}_2$, next to block $\mathbf{V}_3$, etc.

The transformations are based on the ratio of the element to be removed (T_{ij}) to the superdiagonal element in row i ($T_{i,i+1}$) or to the superdiagonal element in column j ($T_{j-1,j}$). There is no intrinsic advantage to using a scheme based on one of these ratios to a scheme based on the other. We will, in the following, discuss the details of the two types of transformations. After this brief discourse, we will discuss the scheme we have devised to choose which type of transformation will be used to obtain a condensed form of each upper triangular block of the matrix.

Transformation of Type I

If T_{ij} is a non-zero element above the superdiagonal in one of the blocks, say $\mathbf{V}_\ell$, we define our transformation matrix as $\mathbf{P}_{i,j} = \mathbf{I} + k\mathbf{e}_{i+1}\mathbf{e}_j{}^T$; the inverse transformation matrix is trivially $\mathbf{P}_{i,j}^{-1} = \mathbf{I} - k\mathbf{e}_{i+1}\mathbf{e}_j{}^T$. In the similarity relation $\mathbf{T}^1 = \mathbf{P}_{i,j}^{-1}\mathbf{T}\mathbf{P}_{i,j}$, all the upper triangular blocks except $\mathbf{V}_\ell$ are unchanged. Within the block $\mathbf{V}_\ell$, elements of row $(i + 1)$ are transformed according to $T^1_{i+1,m} = T_{i+1,m} - kT_{j,m}$ for $j + 1 \leqslant m \leqslant \sum_{a=1}^{\ell} M_a$ and elements of column j are transformed as $T^1_{p,j} = T_{p,j} + kT_{p,i+1}$ for $(\sum_{a=1}^{\ell-1} M_a) + 1 \leqslant p \leqslant i$. $\sum_{a=1}^{\ell} M_a$ is the value of the column index which marks the right boundary of $\mathbf{V}_\ell$. $(\sum_{a=1}^{\ell-1} M_a) + 1$ marks the upper boundary of $\mathbf{V}_\ell$ and is defined to be 1 when $\ell = 1$. Because the diagonal elements of the block have been assumed in this section to be equal, $T^1_{i+1,j} = T_{i+1,j} - k(T_{jj} - T_{i+1,i+1}) \equiv T_{i+1,j}$. Thus, the transformation affects only the (i,j) element, elements in row $(i + 1)$ of the block with a column index $m > j$ and elements in column j of the block with row index $p < i$. No elements outside the block $\mathbf{V}_\ell$ are affected by the transformation (see Figure 2).

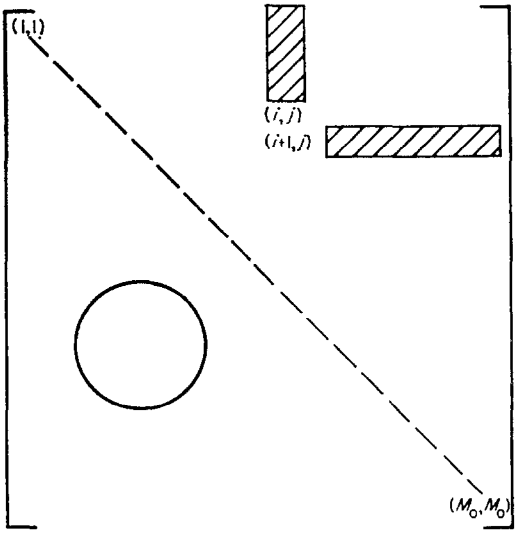

FIG. 2. Elements affected by elimination of element (i,j) during Stage 1.

If $T_{i,i+1} \neq 0$, $T_{i,j}$ may be eliminated by choosing $k = -T_{i,j}/T_{i,i+1}$. Again, the transformation will be rejected if it would cause the product $\| \mathbf{U} \|_\infty \| \mathbf{U}^{-1} \|_\infty$ or the quotient $\| \mathbf{T} \|_\infty / \| \mathbf{T}_0 \|_\infty$ to become greater than the bounds defined in Sections II and [illegible] (to prevent one pathologically large transformation from unduly increasing the two sets of norms) may also be applied to the transformation. Each of these tests is applied in precisely the same manner as described in Section III, and are meant to preserve numerical stability in the sense described in Section II.

As in Section III, the order of the transformations is important to insure that non-zero elements are not reintroduced to sections of the matrix already zeroed. This transformation must be performed in the order

(a) column 3, column 4,..., column M_ℓ of block $\mathbf{V}_\ell$;

(b) within column m, begin with the $(m - 2, m)$ element and move up the column to the element in the first row of block $\mathbf{V}_\ell$. (See Figure 3.)

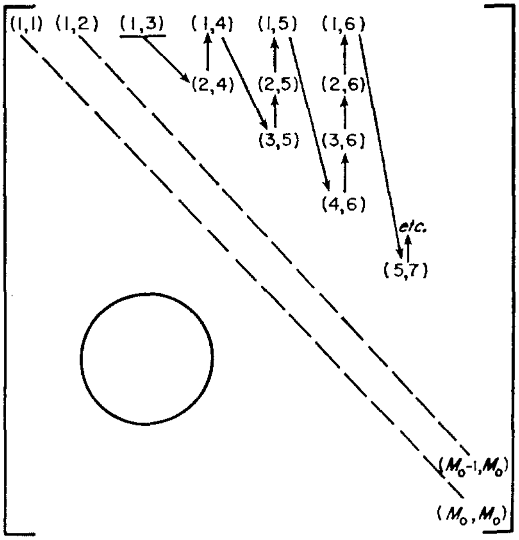

FIG. 3. Order of transformation (Stage I). The element at which the algorithm begins is underlined.

Transformation of Type II

If $T_{i,j}$ is again a non-zero element above the superdiagonal in block $\mathbf{V}_\ell$, the transformation matrix is defined as $\mathbf{P}_{i,j} = \mathbf{I} + k\mathbf{e}_i\mathbf{e}_{j-1}^T$; and the inverse transformation is simply $\mathbf{P}_{i,j}^{-1} = \mathbf{I} - k\mathbf{e}_i\mathbf{e}_{j-1}^T$. In the similarity relation $\mathbf{T}^1 = \mathbf{P}_{i,j}^{-1}\mathbf{T}\mathbf{P}_{i,j}$, all the upper triangular blocks except $\mathbf{V}_\ell$ are unchanged. Within the block $\mathbf{V}_\ell$, elements of the row i are transformed according to $T_{i,m}^1 = T_{i,m} - kT_{j-1,m}$, $j \leqslant m \leqslant \sum_{a=1}^{\ell} M_a$; and elements of column $(j - 1)$ are transformed as $T_{p,j-1}^1 = T_{p,j-1} + kT_{p,i}$, $(\sum_{a=1}^{\ell-1} M_a) + 1 \leqslant p \leqslant i - 1$. (The summations have the same definitions and meanings as in the

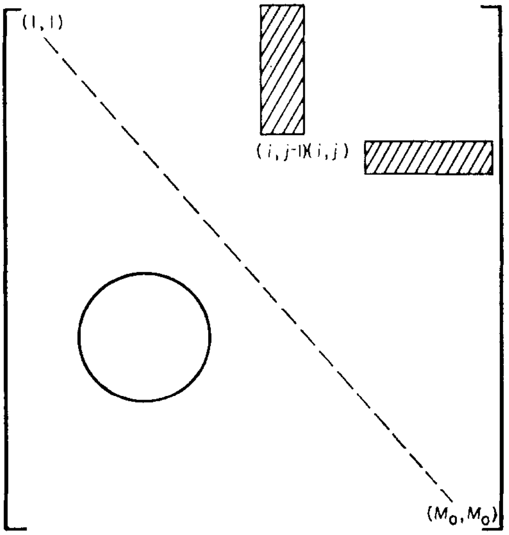

FIG. 4. Elements affected by elimination of element (i, j) during Stage II.

case of transformations of Type I.) Because the diagonal elements of the block have been assumed in this section to be equal, $T^1_{i,j-1} = T_{i,j-1} - k(T_{j-1,j-1} - T_{i,i}) \equiv T_{i,j-1}$. Thus, the transformation affects only the (i, j) element, elements in row i of the block with a column index $m > j$, and elements in column $(j - 1)$ of the block with row index $p < i$. No elements outside the block $\mathbf{V}_\ell$ are affected by the transformation (see Figure 4). If $T_{j-1,j} \neq 0$, element $T_{i,j}$ may be eliminated by choosing $k = T_{i,j}/T_{j-1,j}$. The transformation will be rejected if it fails any of the norm tests which were described for the case of transformations ofType I. So as not to reintroduce non-zero elements where the matrix has been zeroed, one must follow the order

(a) perform the transformation successively on rows $M_\ell - 2$, $M_\ell - 3, \ldots, 1$;

(b) within a row r begin with the $(r, r + 2)$ element and move to the right to element (r, M_ℓ). (See Figure 5.)

There is little to guide us in choosing which type of transformation to use in our reduction scheme. It is clear that, within any particular block $\mathbf{V}_i$, we must use the same type of transformation to attempt to remove all the elements above the super-diagonal. This constraint is the result of our need not to reintroduce non-zero elements in positions which have been previously zeroed. To illustrate this point, suppose that we began our calculations in block $\mathbf{V}_i$ using transformations of Type I. If we failed to remove element T_{mj} by this transformation, we could turn to a transformation of Type II to attempt to zero element T_{mj}. But such a transformation—even if it were acceptable on the basis of our norm tolerances—would be unacceptable because it would alter elements in column $(m - 1)$; these elements would have already been subject to transformations of the type I (see Figure 3) and we might well reintroduce non-zero elements to the column in positions which we had just zeroed.

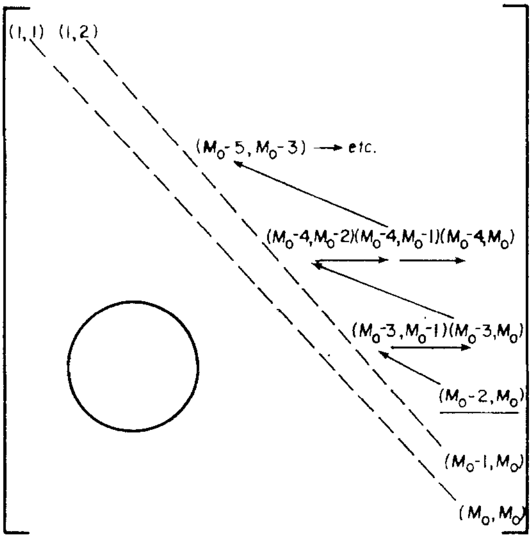

FIG. 5. Order of transformations (Stage II). The element at which the algorithm begins is underlined.

The upper triangle might be less sparse after removing element T_{mj} than before. A similar argument would hold if we had begun with transformations of Type II.

To follow a full sweep of the block $\mathbf{V}_i$ using one type of transformation with a full sweep using the second type of transformation might also leave a less sparse upper triangle than we would have obtained had we halted after the first set of transformations. We cannot show that there is no combination of the two types of transformations which would unambiguously lead to a more condensed form than either of the types applied by itself to the block. We hope that such a combination will be discovered by someone cleverer than the present authors. But at this point in the development of the algorithm, we must content ourselves to using only one type of transformation in the process of condensing any one upper triangular block.

The problem of choosing this one type of transformation for any given block $\mathbf{V}_i$ remains. As we remarked earlier, we have little real guidance in making this choice. Indeed, for the test problems we have studied, both types of transformations have led to nearly identical results. But the choice should not be made in an arbitrary manner. We will, for a given block $\mathbf{V}_i$, choose that type of transformation (Type I or Type II) which seems likely to remove the most elements above the superdiagonal—and, thus, lead to the most condensed form of the block.

Either transformation scheme will fail to remove an element when the ratio of the magnitude of the element to the magnitude of the corresponding superdiagonal element is large. For simplicity, we will divide such occurrences into two categories: (a) cases where the elements are much greater in magnitude than the mean of the absolute values of the superdiagonal elements; and (b) cases where the magnitude of the element is not abnormally large in comparison with the mean of the absolute values of superdiagonal elements, but where the magnitude of the corresponding superdiagonal element is small compared to this mean. Case (a) will cause either scheme to bypass elimination of the large element(s) above the superdiagonal. But while occurrences of this sort can occur in the reduction of a general complex matrix, physical and chemical considerations make them very unlikely for problems of the type described in the Introduction. The effect of occurrences which fall in category (b) can be different depending on the type of transformation chosen. As an example, study the upper triangular block

$$\begin{pmatrix} \lambda_1 & 1 & 10^{-1} & 10^{-2} \\ 0 & \lambda_1 & 1 & 10^{-1} \\ 0 & 0 & \lambda_1 & 0 \\ 0 & 0 & 0 & \lambda_1 \end{pmatrix}$$

Transformations of type I will bring this block into Jordan canonical form, while transformations of type II will succeed in removing only the (1, 3) element.

One goal of the scheme we choose is to minimize the product $\| \mathbf{P} \|_\infty \| \mathbf{P}^{-1} \|_\infty$ where $\mathbf{P}$ is the product of the ℓ transformations $\mathbf{Q}_i$ used to reduce the block. We recall that $\| \mathbf{P} \|_\infty \leqslant \prod_{i=1}^{\ell} \| \mathbf{Q}_i \| = \prod_{i=1}^{\ell} (1 + | k_i |)$ and $\| \mathbf{P}^{-1} \|_\infty \leqslant \prod_{i=1}^{\ell} \| \mathbf{Q}^{-1} \| = \prod_{i=1}^{\ell} (1 + | k_i |)$. We wish then to reduce these bounds by choosing the type of transformation (I or II) which reduces the factors $(1 + | k_i |)$. To use this desire to help choose which type of transformation to perform, we have, for each upper triangular block, followed the procedure:

(1) Find the mean of the absolute value of the superdiagonal elements.

(2) Pick out all superdiagonal elements which are an order of magnitude smaller than the mean calculated in (1). If there are none, pick out the superdiagonal element of smallest magnitude.

(3) Form the products $\prod_{i=1}^{m} (1 + | k_i' |)$ and $\prod_{j=1}^{m^1} (1 + | k_j'' |)$ for the m elements above the superdiagonal in the same row(s) as the superdiagonal elements chosen in (2) and the m^1 elements above the superdiagonal in the same column(s) as these superdiagonal elements. k' and k'' are chosen as the ratio of the element above the superdiagonal to the corresponding superdiagonal element. This provides us with a guess of the relative size (in the sense of a norm) of the contributions of the single element transformations of Type I to those of Type II for those cases which should be the most difficult for the reduction scheme.

(4) Choose transformations of Type I if $\prod_{i=1}^{m} (1 + | k_i' |) \leqslant \prod_{j=1}^{m^1} (1 + | k_j'' |)$. Otherwise, choose transformations of Type II.

This scheme is by no means foolproof or without deficiencies. In particular, k_i' (or k_i'') may be a very poor approximation to the true k_i used in the transformation. Nonetheless, by this procedure we have handled those manageable cases (those earlier classed as category (b)) which could cause difficulties in the reduction scheme in order to provide some guidance in choosing either transformations of type I or of type II for use in the reduction of a particular upper triangular block.

If no proposed transformation within a particular block was rejected because it would have violated the bounds on the magnitudes of the accumulated transformation matrix or of the norm of the transformed matrix (or, when this test is used, of the norm of the individual transformation matrices), all elements above the superdiagonal in block $\mathbf{V}_i$ have been eliminated; unnormalized Jordan form has been obtained.

If some elements have not been zeroed because one or more transformations were rejected, we are unable to obtain Jordan form by the algorithm proposed here. Failure of the algorithm to remove an element above the superdiagonal is symptomatic of a small (at least compared to the element which cannot be made zero) superdiagonal element. This is precisely the general difficulty in obtaining Jordan canonical form discussed in Section IV. Failure of our algorithm to zero elements above the superdiagonal is not symptomatic of a new problem introduced by our methods. But our criterion for the maintenance of numerical stability makes the treatment of this generic symptom more precise than in methods which do not have error bounds.

upper triangular blocks zeroes as many of the upper triangular elements in each degenerate block as is practical, consistent with the bounds on the norms of the accumulated transformations and of the transformation matrix itself. The result is not necessarily a matrix in (un-normalized) Jordan form[5] or any other canonical form; it may be a matrix consisting of diagonal blocks, Jordan blocks, and some blocks merely in upper triangular form. The matrix may be forced into a canonical form by relaxing the bounds, but only at the expense of allowing large elements to enter the transformation matrices (and possibly the transformed matrix itself) and introducing the concomitant danger of instability, due to roundoff error, to this and subsequent calculations. By this algorithm, we will, however, have reduced the problem of calculating matrix functions such as $\mathscr{I}(\omega)$ and $G(t)$ for many values of the scalar parameters to a more manageable problem without sacrificing numerical stability.

VI. The Effect of Having Only Nearly Equal Diagonal Elements of the Block

If the diagonal elements of the upper triangular block are not strictly equal (as is the general case with finite precision calculations), the transformations described in Section V are changed only in that

$$T_{i+1,j} = T_{i+1,j} - k(T_{jj} - T_{i+1,i+1})\text{: Type 1}$$

$$T_{i,j-1} = T_{i,j-1} - k(T_{j-1,j-1} - T_{i,i})\text{: Type 2}$$

rather than leaving these elements unaltered. For superdiagonal elements $(j-1, j)$

[5] "Unnormalized" Jordan form describes a matrix in block form with zeros linking different blocks and with non-zero elements only in the diagonal and superdiagonal rows of each block. The superdiagonal elements in each block do not necessarily have the value unity. Matrices in "unnormalized" Jordan form may be transformed to Jordan canonical form by a series of diagonal transformations. For calculational purposes, the two forms have nearly identical properties.

or $(i, i + 1)$ this is unimportant. For other elements, this corresponds to reintroducing a non-zero value to an element which should have previously been transformed to zero in the transformation scheme. In general, the element reintroduced will be small since the diagonal elements are nearly equal and $| k |$ is indirectly bounded by the bounds on $\| \mathbf{U} \|_\infty \| \mathbf{U}^{-1} \|_\infty$ and $\| \mathbf{T} \| / \| \mathbf{T}_0 \|_\infty$. The elements reintroduced will be of order $(T_{\ell,\ell} - T_{m,m})^2{}_{\max}/T_{\mathrm{s.d.}}$ in magnitude, where $(T_{\ell,\ell} - T_{m,m})_{\max}$ is the maximum difference between eigenvalues in the degenerate block and $T_{\mathrm{s.d.}}$ is the pivot superdiagonal element. k is proportional to the element being eliminated, so that on a second sweep of the transformation sequence to eliminate the non-zero values reintroduced in the first sweep, the method would reintroduce elements proportional to $(T_{\ell\ell} - T_{mm})^3_{\max}/T^2_{\mathrm{s.d.}}$. Subsequent sweeps would cause the elements reintroduced to be proportional to $(T_{\ell\ell} - T_{mm})^{a+1}_{\max}/T^a_{\mathrm{s.d.}}$, where a is the number of sweeps performed. Because $T_{\ell\ell} - T_{mm}$ is a small number, the elements reintroduced should eventually become smaller in magnitude than $\epsilon * \| T \|_\infty$ where ϵ is an arbitrary small positive number. The elements reintroduced may then be safely neglected. The iteration process will not converge if $T_{\mathrm{s.d.}} < | T_{\ell\ell} - T_{mm} |$. This corresponds to the situation of a small, "nearly" zero superdiagonal element—the occurrence which causes our method generally to fail to eliminate an element. Thus, failure of the iteration process to converge is not due to a new limitation of the algorithm, but is a symptom of the general difficulty encountered in this problem. The iteration sequence is halted when the number and magnitudes of the remaining upper diagonal elements no longer decrease.

VII. Test Results

Several tests were performed using the algorithm described in this paper. We remind the reader that the explicit aim of the algorithm is not a complete eigenanalysis of a matrix; rather, the algorithm attempts to produce a condensed form of the matrix which facilitates evaluations of matrix quantities which are functions of a scalar (such as those described in Section I) for many values of the scalar. Nonetheless, in the few physical chemical problems we have used as tests and in most applications to previously published test matrix systems, we have obtained a complete eigenanalysis. We present three tests of this type for a comparison of the accuracy of our methods to that of other algorithms intended for the eigenanalysis of general complex matrices. We present a fourth test result in which we do not obtain any particular canonical form with our algorithm. We hope, with this test, to show that the algorithm proposed here provides considerable computational advantage for problems of the type described in the Introduction.

The results quoted here were found using a double precision version (double precision = 16 digits on an IBM 360/91) of the program. The bounds discussed in Section 2 are $N_L = N_A = 10^3$. That is, we expect the results to have accuracies at least as good as that accuracy obtained in single precision (7 digits) computations employing exactly unitary similarities.

(1) In most previous applications of the QR algorithm, the eigenvectors were determined by inverse iteration. As noted earlier, we intend this algorithm to apply to types of degenerate and nearly degenerate problems where inverse iteration often does not provide a complete set of eigenvectors. Therefore, we have chosen to accumulate the transformation matrices during the calculation in order to find the complete set of eigenvectors. In order to show how well this scheme calculates the eigenvectors, we have tested the algorithm on a matrix which is diagonalizable. This does not test our main result, the algorithm of the previous two sections, but does isolate the question of the propriety of calculating the eigenvectors by this method. We use the matrix [20]

$$\begin{bmatrix} 1+3i & 2+i & 3+2i & 1+i \\ 3+4i & 1+2i & 2+i & 4+3i \\ 2+3i & 1+5i & 3+i & 5+2i \\ 1+2i & 3+i & 1+4i & 5+3i \end{bmatrix}$$

which has no degenerate eigenvalues and a complete set of four eigenvectors.

Calculated Eigenvalues

$$\begin{array}{r} 9.78365812\underline{5}2 + 9.3225142247\underline{0}i \\ -3.3710097852\underline{1} - 0.7704539869\underline{7}i \\ 2.2216823475\underline{3} + 1.8489933596\underline{7}i \\ 1.3656693099\underline{0} - 1.4010535974\underline{1}i \end{array}$$

Calculated Eigenvectors

$$\begin{bmatrix} 6.32337764\underline{7} \times 10^{-1} - 1.43280794\underline{5} \times 10^{-2}i \\ 8.73758591\underline{5} \times 10^{-1} + 8.10578078\underline{4} \times 10^{-3}i \\ 1.000000000 + 0.0000000000i \\ 9.43717588\underline{6} \times 10^{-1} + 3.79846348\underline{2}i \end{bmatrix}$$

$$\begin{bmatrix} -5.0609646\underline{2}0 \times 10^{-1} + 5.83451962\underline{7} \times 10^{-1}i \\ 1.000000000 + 0.000000000i \\ 5.18319483\underline{8} \times 10^{-1} - 7.14657072\underline{0} \times 10^{-1}i \\ -5.53484981\underline{9} \times 10^{-1} + 1.87563320\underline{3} \times 10^{-2}i \end{bmatrix}$$

$$\begin{bmatrix} -7.96620817\underline{4} \times 10^{-1} + 3.04980785\underline{8} \times 10^{-1}i \\ -1.7883439\underline{5}0 \times 10^{-1} + 4.297241478 \times 10^{-1}i \\ -2.52814311\underline{2}0 \times 10^{-1} + 3.8173404952 \times 10^{-2}i \\ 1.000000000 + 0.0000000000i \end{bmatrix}$$

$$\begin{bmatrix} -8.465987045 \times 10^{-4} + 7.302035513 \times 10^{-1}i \\ -8.7941701580 \times 10^{-2} - 3.879017996\underline{6}0 \times 10^{-1}i \\ 1.000000000 + 0.000000000i \\ -4.32089374\underline{6}5 \times 10^{-1} - 4.33426369\underline{4}1 \times 10^{-1}i \end{bmatrix}$$

The eigenvectors have been normalized such that the element of largest magnitude is set to unity. The first digit in disagreement with the values published in reference 20 are underlined.

(2) To test the algorithm of the previous section, the method proposed here was applied to a 6 × 6 matrix with two quadratic divisors and whose eigenvalues and set of generalized eigenvectors are known exactly:

$$(1 + 0.5i) \begin{bmatrix} 2 & -2 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -2 & 1 \\ -1 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & -3 & 0 & 0 & -1 \\ 0 & 0 & 3 & 0 & 4 & 5 \end{bmatrix}$$

Eigenvalues

Exact	Computed
$1 + \frac{1}{2}i$	$1.00000000002 + 0.5000000002i$
$1 + \frac{1}{2}i$	$0.99999999999997 + .49999999999998i$
$1 + \frac{1}{2}i$	$1.0000000000001 + .500000000001i$
$-2 - i$	$-1.99999999999992 - .99999999999999i$
$4 + 2i$	$4.000000000003 + 2.000000000001i$
$4 + 2i$	$3.99999999999998 + 1.999999999997i$

Exact Eigenvectors

$$\begin{array}{cccc} \lambda = 1 & \lambda = 1 & \lambda = -2 & \lambda = 4 \\ \begin{bmatrix} -1 \\ -1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ -1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 1 \end{bmatrix} \end{array}$$

Exact Generalized Eigenvectors

$$\begin{array}{cc} \lambda = 1 & \lambda = 4 \\ \begin{bmatrix} -1 \\ 1/2 \\ 0 \\ -1/2 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ -1/8 \\ 0 \\ 1/8 \\ 1 \end{bmatrix} \end{array}$$

The calculated eigenvectors were normalized such that the element of largest magnitude was set to unity. The imaginary parts of the calculated eigenvectors were all less than 10^{-13} in magnitude. The real parts of the calculated eigenvectors differed from the exact eigenvectors by less than 2 parts in 10^{13}. We also calculated the matrix $\mathbf{H} = \mathbf{UDU}^{-1}$, where $\mathbf{U}$ is the matrix of eigenvectors and $\mathbf{D}$ is the diagonalized form of the original matrix. If we denote the original matrix as $\mathbf{A}$, we find that $(\| \mathbf{A} \|_\infty - \| \mathbf{H} \|_\infty)/\| \mathbf{A} \|_\infty < 10^{-15}$. The diagonal form obtained is clearly similar to a matrix very nearly the same as our original matrix.

(3) Several 12×12 complex matrices A' were obtained by multiplying the matrix

$$A = \begin{bmatrix}
6 & 1 & & & & & & & & & & \\
& 6 & 1 & & & & & & & & & \\
& & 6 & 1 & & & & & & & \bigcirc & \\
& & & 6 & 1 & & & & & & & \\
& & & & 6 & 1 & & & & & & \\
& & & & & 6 & 0 & & & & & \\
& & & & & & 3 & 1 & & & & \\
& & & & & & & 3 & 1 & & & \\
& & & & & & & & 3 & 1 & & \\
& \bigcirc & & & & & & & & 3 & 1 & \\
& & & & & & & & & & 3 & 1 \\
& & & & & & & & & & & 3
\end{bmatrix}$$

by known $\mathbf{P}$ and $\mathbf{P}^{-1}$ such that $\mathbf{A}' = \mathbf{P}^{-1}\mathbf{AP}$. We then used a preliminary version of the algorithm of Golub and Wilkinson and the double precision version of our algorithm to reduce $\mathbf{A}'$ to Jordan form. The accuracies of the results were comparable. Our algorithm required approximately 55 % less core than the program of Golub and Wilkinson. In addition, 15–25 % less time was required. This last point may be misleading in that (a) the programs do not perform exactly the same functions and (b) our program performs all arithmetic in the real field while the program of Golub and Wilkinson uses the machine supplied routines to perform complex arithmetic. As a more precise indicator of the efficiency of our algorithm, we have counted the number of multiplications for each step in the limit of large N (the dimension of the matrix):

	Number of complex multiplications	
	For matrix alone	For matrix and eigenvectors
Hessenberg Form [19, p. 349]	3	[illegible]
QR algorithm[a] [19, p. 542]	$\sim 8\,N^3$	$\sim 16\,N^3$
Reduce to block upper triangular form	$\sim \frac{N^3}{3}$	$\sim N^3$
Reduce to Jordan or "nearly" Jordan form (per iteration)[b]	$\sum_\alpha \frac{M_\alpha{}^3}{3}$	$\sum_\alpha \left[\frac{M_\alpha + 2N}{3}\right] M_\alpha{}^2$

[a] Assuming two iterations per eigenvalue.

[b] M_α is the dimension of the α upper triangular block, $\Sigma_\alpha M_\alpha = N$.

The algorithm of Golub and Wilkinson may necessitate the calling of a singular decomposition routine, an order $\sum_\alpha M_\alpha{}^3$ ($\sum_\alpha M_\alpha = N$) process, up to $2N$ times. The result of both our computational experience and operation count cause us to believe that our algorithm requires less core and is faster than the more classical approach to the problem. Furthermore, we can bound the accumulated round-off effects of our single element transformations. The level of round-off error we will tolerate does, in fact, determine which single element transformations will be performed and which will not.

(4) As a final test,we consider the application of the algorithm of Sections III–VI to a set of $n \times n$ Frank matrices $\mathbf{F}^{(n)}$ ($n = 2, 3,\ldots, 12$) defined by

$$\begin{array}{lll} F_{ij}^{(n)} = n - j + 1 & & i = 1, n;\quad j = i, n \\ F_{j+1,j}^{(n)} = n - j & & j = 1, n - 1 \\ F_{ij}^{(n)} = 0 & & \text{otherwise} \end{array}$$

For rather small values of n (depending on the precision of the machine calculations), some of the eigenvalues and eigenvectors are very ill-conditioned [6]. However, the simple transformation

$$\begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & 0 & & \\ & & 1 & -1 & & \\ & & & \ddots & \ddots & \\ & 0 & & & \ddots & -1 \\ & & & & & 1 \end{bmatrix} (\mathbf{F}^{(n)} - \mathbf{I})$$

$$= \begin{bmatrix} (1-\lambda) & \lambda & & & \\ n-1 & (1-\lambda) & \lambda & & \\ & n-2 & (1-\lambda) & \ddots & \\ & & \ddots & \ddots & \lambda \\ & & & 1 & (1-\lambda) \end{bmatrix} = \mathbf{G}^{(s)}$$

enables one to determine the eigenvalues of $F^{(n)}$ from those of a "quasi-symmetric" tridiagonal matrix,

$$\mathbf{T}^{(s)} = \begin{bmatrix} 0 & 1 & & & & \\ s-1 & 0 & 1 & & & \\ & s-2 & 0 & 1 & & \\ & & s-3 & \ddots & \ddots & \\ & & & \ddots & \ddots & 1 \\ & & & & 1 & 0 \end{bmatrix}$$

The determination of these latter eigenvalues is a well-conditioned problem for all values of n [6]. Previous study [6] has shown that the smaller eigenvalues of $\mathbf{F}^{(n)}$ are ill-conditioned and that the QR algorithm will show substantial error in the determination of these eigenvalues. In the following we briefly describe our results for the most ill-conditioned problem we have studied, $\mathbf{F}^{(12)}$. Our algorithm utilizes the QR algorithm to transform the matrix in Hessenberg form to upper triangular form. Using double precision arithmetic we find the four smallest eigenvalues of $\mathbf{F}^{(12)}$ to be

$$\lambda_{12} = 0.0310\cdots, \quad \lambda_{11} = 0.0495\cdots, \quad \lambda_{10} = 0.0812\cdots, \quad \lambda_9 = 0.1436\cdots.$$

These values are in agreement with the results of reference 12. Slight perturbations in the upper triangular elements caused by round-off error in the application of Osborne's equilibration algorithm prior to use of the QR algorithm led to the computation of two pairs of complex conjugate eigenvalues

$$\lambda_{11,12} = -0.04678 \pm 0.08914i$$
$$\lambda_{9,10} = 0.15556 \pm 0.17213i.$$

Thus, our algorithm shows the same instabilities, when applied to $\mathbf{F}^{(12)}$, as described in reference 6.

Application of the single element transformations to zero the upper triangular elements introduces to the upper triangle some elements large compared to those along the diagonal. This is a reflection of the inability of the QR algorithm to accurately determine the upper triangular form unitarily similar to the original Frank matrix. The disposition of the large elements introduced to the upper triangle depends on the bounds on the norms of the matrix transformations and of the transformed matrix. Elements so large that the transformation is not performed will remain in the upper triangle. Smaller elements will be annihilated introducing additive factors of the order of the transformation element to the transformation matrices $\mathbf{U}$ and $\mathbf{U}^{-1}$. With the norm bounds given at the beginning of this section, elements of magnitude 5×10^2 were introduced to $\mathbf{U}^{-1}$. The final transformed matrix may be placed in diagonal form by increasing the size of the bounds on the norms; or more elements may be retained in the upper triangle by decreasing these bounds. Thus, we may obtain a very large number of final forms of the matrix, each of which is similar to a matrix $\mathbf{H}^{(n)}$ nearly equal to the original Frank matrix, $\mathbf{H}^{(n)} = \mathbf{F}^{(n)} + \mathbf{K}$, where $\|\mathbf{K}\|_\infty / \|\mathbf{F}^{(n)}\|_\infty$ is a very small number determined essentially by the values of N_L and N_A.

For $N_L = N_A = 10^3$, we find $\|\mathbf{K}\|_\infty / \|\mathbf{F}^{(12)}\|_\infty$ to be about 3×10^{-9}. The a priori upper bound of Section II would be $10^{-7} \sum_{j=1}^{s} f(j, 12)$. For $N_L = N_A = 10^2$, we find $\|\mathbf{K}\|_\infty / \|\mathbf{F}^{(12)}\|_\infty$ to be about 10^{-12}. In this case, the a priori upper bound would be $10^{-10} \sum_{j=1}^{s'} f(j, 12)$. The smaller value of $N_A = N_L$ for this second calculation implies that the relative error should be a factor of 10^3 smaller than for the first computation. The computed relative error is in fact 3×10^3 smaller for the second computation than for the first. If we assume that $n = 12$ is large enough that $\sum_{j=1}^{s} f(j, n)$ may be taken to be $Kn^{3/2}$, as in Section II, (K is a number of magnitude unity or slightly greater), the a priori bounds have the numerical values $4K \times 10^{-6}$ and $4K \times 10^{-9}$, respectively. The computed relative errors are a factor of approximately 10^3 smaller than this. Assuming a more random distribution of roundoff errors, $\sum_{j=1}^{s} f(j, n) \simeq K^* n^{3/4}$, the a priori bounds are respectively, $6K^* \times 10^{-7}$ and $6K^* \times 10^{-10}$, which are still between two and three orders of magnitude larger than the computed relative errors. Our a priori bounds to the error, at least for the Frank matrix problem, are very generous. This calculation shows explicitly our ability to control round-off error by varying N_L and N_A. The cost of decreasing the round-off error by more than three orders of magnitude was the retention of two extra elements in the upper triangle. The block structure was unaltered by this change.

We are unable to calculate accurately the eigenvalues and particularly the eigenvectors of the Frank matrix $\mathbf{F}^{(12)}$. Nonetheless, we have retained knowledge of the transformations used in reducing the matrix to a condensed form and we can use the results of the calculation to obtain functions such as $\mathscr{I}(\omega)$ for given vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ and many values of ω. We have performed such a calculation directly with $\mathbf{F}^{(12)}$ and from the condensed form of $\mathbf{F}^{(12)}$ obtained with $N_L = N_A = 10^2$ for 1000 values of ω (and a more or less arbitrary choice of $\mathbf{d}_1$ and $\mathbf{d}_2$). The calculation was approximately 12 times faster using the reduced form of $\mathbf{F}^{(12)}$; the results are accurate to 12 significant figures.

VIII. Conclusion

The algorithm presented here may be summarized as follows:

(1) Equilibrate the matrix by Osborne's algorithm;

(2) Reduce to upper Hessenberg form;

(3) Upper triangularize by the QR algorithm;

(4) Eliminate upper triangular elements by single element transformations based on the ratio $(i,j)/|(i,i) - (j,j)|$. Stop if diagonal form is obtained.

(5) Permute the matrix to block upper triangular form.

(6) Work on each block separately. Eliminate remaining elements above the superdiagonal by single element transformations based on the ratios $(i,j)/(i, i + 1)$ (Type I) or $(i,j)/(j - 1,j)$ (Type II). Iterate if non-zero values are reintroduced to previously zeroed positions because $(i, i) \neq (j,j)$.

Our emphasis on single element transformations is not necessary to the spirit of this approach; single row and single column schemes have also been investigated. But such single element transformations possess significant virtues:

(1) Simple programming;

(2) Both trivial calculation of the inverse transformation and a bound on the inverse transformation identical to the bound on the transformation itself.

(3) Easy perception of the cause of a large element in the transformation;

(4) A quite fast algorithm.

Further, we have found them to be completely equivalent to the whole row or whole column elimination schemes and, when no small superdiagonal elements are present, to the simple linear equations solution method of finding generalized eigenvectors discussed in Section IV.

This computational method is meant to reduce a matrix by stable transformations to a simpler form. It aims to obtain diagonal form or Jordan canonical form as the final product. We find, however, that attempts to force a matrix into such canonical forms sometimes require acceptance of large transformation elements which may be accompanied by the introduction of large round-off errors and numerical instability to the calculations. We have chosen to forego obtaining a canonical form when instabilities may result, instead choosing to perform subsequent calculations with a simplified, but noncanonical form of the matrix. Subsequent calculations may be made more difficult (but these cases have been few in our experience); however, their numerical accuracy will be guaranteed to any desired level.

Acknowledgments

The authors thank Professor Gene Golub for sending us a copy of his program for reducing a matrix to Jordan form. Our work was supported in part by the National Science Foundation, and D.F.H. acknowledges support as a National Science Foundation Predoctoral Fellow, 1971–74.

References

1. A. B. Elkowitz and R. E. Wyatt, *J. Chem. Phys.* **63** (1975), 702.
2. J. G. F. Francis, *Comput. J.* **4** (1961), 265.

3. J. G. F. Francis, *Comput. J.* 4 (1962), 332.
4. G. H. Golub and W. H. Kahan, *J. SIAM Numer. Anal. Ser. B* **2** (1965), 205.
5. G. H. Golub and C. Reinsch, *Numer. Math.* **14** (1970), 403.
6. G. H. Golub and J. H. Wilkinson, Stanford Computer Science Report Number 478, 1975.
7. R. G. Gordon, *J. Chem. Phys.* **46** (1967), 4399.
8. R. G. Gordon, *Advan. Magn. Resonance* **3** (1968), 1.
9. R. G. Gordon and T. Messenger, *in* "Electron Spin Relaxation in Liquids" (L. T. Muus and P. W. Atkins, Eds.), p. 341, Plenum, New York, 1972.
10. R. G. Gordon and W. B. Neilsen, *J. Chem. Phys.* **58** (1973), 4131.
11. R. G. Gordon and W. B. Neilsen, *J. Chem. Phys.* **58** (1973), 4149.
12. R. G. Gordon and R. Shafer, *J. Chem. Phys.* **58** (1973), 5422.
13. R. G. Gordon and J. I. Steinfeld, *in* "Proceedings, Israel Scientific Research Conference on Molecular Energy Transfer, En Boqeq, December, 1973."
14. R. S. Martin and J. H. Wilkinson, *Numer. Math.* **12** (1968), 349.
15. B. Noble, "Applied Linear Algebra," Prentice–Hall, Englewood Cliffs, N. J., 1969.
16. A. Ruhe, preprint, 1975.
17. A. E. Stillman and R. N. Schwartz, *J. Magn. Reson.* **22** (1976), 269.
18. E. E. Osborne, *J. Assoc. Comput. Mach.* 7 (1960), 338.
19. J. H. Wilkinson, "The Algebraic Eigenvalue Problem," Oxford Univ. Press (Clarendon), London/New York, 1965.