HYBRID COMPUTER SOLUTION OF LINEAR STATE MODELS

Thesis for the Degree of Ph. D.

MICHIGAN STATE UNIVERSITY
WILLIAM C. ELLSWORTH

1969

This is to certify that the

thesis entitled

HYBRID COMPUTER
SOLUTION OF LINEAR STATE MODELS

presented by

William C. Ellsworth

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in E.E.

J. Sutherland Frame
Major professor

Date January 22, 1969

ABSTRACT

HYBRID COMPUTER SOLUTION OF LINEAR STATE MODELS

By

William C. Ellsworth

The solution of a set of differential equations of the form

$$\frac{d}{dt}X(t) = AX(t),$$

where $X(t)$ is an n-dimensional vector function of time and A is an $(n \times n)$ real square matrix, is desired in many engineering problems. By solving the system of equations on a hybrid computer, the digital computer can be assigned the arithmetic and automated analog set-up, and the analog computer can simultaneously integrate all the state variables, thus utilizing the best features of both machines. The procedure can be programmed to require not more than $(2n-1)$ potentiometers nor more than $2n$ operational amplifiers (n bipolar integrators) on the analog computer. This requirement can be met by tridiagonalizing the A matrix on the digital computer as well as having the digital computer scale the transformed system in time, amplitude and initial conditions before integrating the transformed system on the analog computer.

In this thesis, an improved method for tridiagonalizing an arbitrary real square matrix is developed based on Lanczos' method.[1] After transforming the given matrix A by unitary Householder transformations into a similar upper Hessenberg matrix A_H, the latter is transformed into a tridiagonal matrix $T = RA_HC$ by appropriate matrices R (with rows R_i) and $C = R^{-1}$ (with columns C_i). The vectors R_{i+1} and C_{i+1} are obtained recursively in the Lanczos algorithm starting with vectors R_1 and C_1 which are supposed to be arbitrary. But, if R_1 and C_1 lie in certain unknown subspaces, the algorithm breaks down. By choosing $C_1 = [1\ 0\ \cdots\ 0]^T$, and obtaining C_{i+1} directly from column (i+1) of the idempotent matrix $U - \sum_{j=1}^{i} C_jR_j$, the matrices R and C become unit upper triangular in the regular case, and the vector C_{i+1} is never indeterminate. Whenever the computation of R_{i+1} from the recurrence relation breaks down, this new method describes a procedure for continuing the algorithm.

The tridiagonalized system is scaled in time and amplitude by an automated procedure. Time scaling multiplies the tridiagonal matrix T by a constant β. Amplitude scaling transforms βT by a diagonal matrix into a final tridiagonal matrix in which at least (n-1) of the off-diagonal elements adjacent to the main diagonal are scaled to 0, 0.1, 1, 10, or 100 in magnitude. Thus, at most (2n-1) potentiometers are needed on the analog computer.

The tridiagonalization, as well as the setting of the potentiometers, can be done automatically by the digital computer. The digital computer can also sample the analog solution and transforms it back to produce the solution $X(t)$ for the given initial conditions $X(0)$.

---

[1] R. L. Causey and R. T. Gregory, "On Lanczos' Algorithm for Tridiagonalizing Matrices," Society of Industrial and Applied Mathematics Review, III, No. 4 (October, 1961), 322-328.

HYBRID COMPUTER SOLUTION OF LINEAR STATE MODELS

By

William C. Ellsworth

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering

1969

ACKNOWLEDGMENTS

The author acknowledges the advice and guidance furnished by Dr. J. S. Frame throughout the period of time devoted to this thesis. He also thanks Dr. J. B. Kreer for his assistance in selecting the thesis topic as well as his counsel on the analog computer aspects of this thesis. Furthermore, the author wishes to thank Dr. H. E. Koenig, Dr. R. C. Dubes, and Dr. Y. Tokad for their advice and guidance during the writing of this thesis. Finally, because of her devotion and loyalty, the author is indebted to his wife Barbara for her assistance and understanding.

TABLE OF CONTENTS

I INTRODUCTION

In many engineering problems, it is necessary to solve linear state models of the form

$$\begin{aligned} \frac{d}{dt}X(t) &= AX(t) \\ F(t) &= CX(t) \end{aligned} \tag{1.1}$$

where $X(t)$ is an n-dimensional vector function of time, A is an $(n \times n)$ constant matrix, $F(t)$ is a p-dimensional vector function of time, and C is a $(p \times n)$ constant matrix.

The homogeneous system of differential equations (1.1) includes the class of linear state models in which the drivers can be generated as the solution of the set of linear differential equations with constant coefficients shown below.

$$\begin{aligned} \frac{d}{dt}\underline{X}(t) &= \underline{AX}(t) + \underline{BE}(t) \\ F(t) &= \underline{CX}(t) + \underline{DE}(t) \end{aligned} \tag{1.2}$$

where $\underline{E}(t)$ is a particular solution of

$$\frac{d}{dt}\underline{E}(t) = \underline{GE}(t) \tag{1.3}$$

By combining (1.2) and (1.3), we obtain the system

$$\frac{d}{dt}X(t) = \frac{d}{dt}\begin{bmatrix} \underline{X}(t) \\ \underline{E}(t) \end{bmatrix} = \begin{bmatrix} \underline{A} & \underline{B} \\ 0 & \underline{G} \end{bmatrix}\begin{bmatrix} \underline{X}(t) \\ \underline{E}(t) \end{bmatrix} = AX(t)$$

$$F(t) = [\underline{C}\ \underline{D}]\begin{bmatrix} \underline{X}(t) \\ \underline{E}(t) \end{bmatrix} = CX(t) \qquad (1.4)$$

which has the form (1.1).

The object of this thesis is to present a method by which the hybrid computer may be used to solve equations of the form (1.1) in a manner which is numerically stable and optimal in the following sense:

a. It should use a minimum number of operational amplifiers.

b. It should use a minimum number of potentiometers.

To solve equations of the form (1.1) on the analog computer, one potentiometer is needed for every entry of A not 0, 1, 10, or 100. To simplify the computation and meet the criterion of optimality, the matrix A is transformed into βSAS^{-1}, a scalar multiple of a matrix similar to A, having at most (3n-2) non-zero entries of which as many as possible are 1, 10, or 100 in magnitude. However, for reasons of numerical stability, we wish to avoid methods requiring the solution of the characteristic equation, such as the transformation to Jordan form. The proposed method will require only rational operations and the extraction of square roots.

The change of variables

$$t = \beta\tau,\ Z(\tau) = SX(t) = SX(\beta\tau) \qquad (1.5)$$

transforms the differential equations in (1.1) into

$$\frac{d}{dt} Z(\tau) = \beta SAS^{-1} Z(\tau) \qquad (1.6)$$

After this transformation has been implemented on the digital computer, the resulting equations (1.6) are solved on the analog computer. The digital computer can then sample the continuous analog solution periodically, re-transform the solution back to the problem solution X(t), and print the entries of X(t) and F(t). The entire process, from reading the equations (1.1) to printing the solution, can be made fully automatic using the hybrid computer.

In the solution procedure described in this thesis, the best properties of both the digital and analog computers are utilized; the digital computer does the arithmetic, bookkeeping, and analog set-up, and the analog computer integrates all the transformed state variables simultaneously.

The means proposed for determining an optimal matrix, similar to A, without computing the eigenvalues is to transform the matrix A into a tridiagonal matrix T having at most (3n-2) non-zero entries on or adjacent to the main diagonal.

Chapter 2 presents five ways for tridiagonalizing a matrix that are described in the recent literature. Each of these methods has significant shortcomings which, for algorithmic computation, involve undue complexities and possible breakdowns.

One type of shortcoming is a numerical instability due to division by a small quantity. The other type is called a breakdown and results when the algorithm cannot be continued

without modifications which may or may not be possible.

Chapter 3 presents an original tridiagonalization algorithm based on Lanczos'[1] method, but which is better adapted to automation. The new algorithm produces a unit upper triangular transforming matrix in the regular case, starting with a computer-determined initial vector which the computer itself modifies in the irregular cases. Two theorems are proved which show that the algorithm can always be implemented, even if a breakdown occurs.

Once the tridiagonal matrix is obtained, it must be scaled both in time and amplitude. Furthermore, the initial conditions must also be scaled before the system (1.6) is ready for solution on the analog computer. As another original contribution in this thesis, Chapter 4 presents an automatic scaling procedure whereby as many as possible of the entries on the three diagonals are assigned the values 0, 1, 10, or 100 in magnitude.

II KNOWN TRIDIAGONAL TRANSFORMATION METHODS

Five recently published methods for transforming an $(n \times n)$ real but non-symmetric matrix into a similar tridiagonal matrix serve as a background for the theoretical development in this thesis. Following a description of the techniques and shortcomings of each of these methods, a new and improved method will be presented in Chapter 3.

2.1 Lanczos' Method[1]

Given the arbitrary, real, square matrix A of order n, Lanczos' method attempts to construct non-singular transformation matrices R and C such that

$$RAC = T \tag{2.1.1}$$

$$RC = U \tag{2.1.2}$$

where U is the unit matrix and T is tridiagonal of the form

$$T = \begin{bmatrix} t_1 & q_2 & 0 & \cdots & 0 & 0 \\ e_2 & t_2 & q_3 & \cdots & 0 & 0 \\ 0 & e_3 & t_3 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & t_{n-1} & q_n \\ 0 & 0 & 0 & \cdots & e_n & t_n \end{bmatrix} \tag{2.1.3}$$

Let R_i denote the i^{th} row of R and C_i the i^{th} column of C; then R_i and C_i are computed recursively starting with somewhat arbitrary vectors R_1 and C_1 satisfying the relation $R_1C_1 = 1$.

Equations (2.1.1) and (2.1.3) require that the entries in T have the following forms:

$$t_{ii} = t_i = R_iAC_i, \quad i = 1, \cdots, n \tag{2.1.4}$$

$$t_{i+1,i} = e_{i+1} = R_{i+1}AC_i, \quad i = 1, \cdots, n-1 \tag{2.1.5}$$

$$t_{i,i+1} = q_{i+1} = R_iAC_{i+1}, \quad i = 1, \cdots, n-1. \tag{2.1.6}$$

Using (2.1.2), it can be seen that (2.1.1) may be written in two other forms

$$RA = TR \tag{2.1.7}$$

$$AC = CT \tag{2.1.8}$$

It then follows that

$$q_{i+1}R_{i+1} = R_i(A - t_iU) - e_iR_{i-1}, \quad i = 1, \cdots, n-1 \tag{2.1.9}$$

$$e_{i+1}C_{i+1} = (A - t_iU)C_i - q_iC_{i-1}, \quad i = 1, \cdots, n-1, \tag{2.1.10}$$

where $q_1 = e_1 = 0$, $R_0 = 0$, and $C_0 = 0$.

As an intermediate step for subsequent discussion, let the vectors X and Y at stage i be defined as

$$X = R_i(A - t_iU) - e_iR_{i-1} \tag{2.1.11}$$

and

$$Y = (A - t_iU) - q_iC_{i-1} \tag{2.1.12}$$

At the i^{th} iteration, the quantities t_i, e_{i+1}, C_{i+1}, q_{i+1}, and R_{i+1} are to be determined in the order indicated. First, t_i is obtained from (2.1.4). If the column vector Y

in (2.1.12) is not zero, set

$$e_{i+1} = 1, \quad C_{i+1} = Y. \tag{2.1.13}$$

If, on the other hand $Y = 0$, set

$$e_{i+1} = 0 \tag{2.1.14}$$

and choose some vector orthogonal to $R_1, \cdots, R_i$ as C_{i+1}. If the row vector X in (2.1.11) is not zero, then q_{i+1} is computed using (2.1.6) and R_{i+1} is then determined from

$$R_{i+1} = \frac{X}{q_{i+1}} \quad . \tag{2.1.15}$$

However, if $X = 0$, set

$$q_{i+1} = 0 \tag{2.1.16}$$

and choose some vector orthogonal to $C_1, \cdots, C_i$ as R_{i+1}, normalized so that $R_{i+1}C_{i+1} = 1$. These choices of C_{i+1} and R_{i+1} are consistent with (2.1.6) and (2.1.5), since the orthogonality criteria require

$$\begin{aligned} R_{i+1}(AC_i) &= R_{i+1}(e_{i+1}C_{i+1} + t_iC_i + q_iC_{i-1}) \\ &= e_{i+1}(R_{i+1}C_{i+1}) + t_i(R_{i+1}C_i) + q_i(R_{i+1}C_{i-1}) \\ &= e_{i+1} + 0 + 0 \end{aligned} \tag{2.1.17}$$

$$\begin{aligned} (R_iA)C_{i+1} &= (q_{i+1}R_{i+1} + t_iR_i + e_iR_{i-1})C_{i+1} \\ &= q_{i+1}(R_{i+1}C_{i+1}) + t_i(R_iC_{i+1}) + e_i(R_{i-1}C_{i+1}) \\ &= q_{i+1} + 0 + 0. \end{aligned} \tag{2.1.18}$$

The vectors R_{i+1} and C_{i+1} can be expressed directly in terms of R_1, C_1, and A with the help of Lanczos' polynomials. Consider the characteristic matrix $[\lambda U - T]$. Householder[3]

calls the leading principal i^{th} order minors of $[\lambda U - T]$ Lanczos polynomials. Letting $P_{-1}(\lambda) = 0$ and $P_0(\lambda) = 1$, the expansion of the minor $P_i(\lambda)$ by cofactors of the last column yields the recurrence relation

$$P_i(\lambda) = P_{i-1}(\lambda)(\lambda - t_i) - P_{i-2}(\lambda)e_i q_i \qquad (2.1.19)$$

Assuming that $q_{i+1} \neq 0$ and $e_{i+1} \neq 0$ for $i = 2, \cdots, n$ and using (2.1.19), we see by induction that (2.1.9) and (2.1.10) may be written as

$$q_2 \cdots q_{i+1} R_{i+1} = R_1 P_i(A) \qquad (2.1.20)$$

$$e_2 \cdots e_{i+1} C_{i+1} = P_i(A) C_1 . \qquad (2.1.21)$$

If $P_i(\lambda) = \sum_{j=1}^{i} P_{ij} \lambda^{j-1}$, then

$$R_1 P_i(A) = \sum_{j=1}^{i} p_{ij} R_1 A^{j-1} = \sum_{j=1}^{i} p_{ij} F_j \qquad (2.1.22)$$

where $F_j = R_1 A^{j-1}$ is row j of the Krylov matrix F.

$$F = \begin{bmatrix} F_1 \\ F_2 \\ \cdot \\ \cdot \\ \cdot \\ F_n \end{bmatrix} = \begin{bmatrix} R_1 \\ R_1 A \\ \cdot \\ \cdot \\ \cdot \\ R_1 A^{n-1} \end{bmatrix} . \qquad (2.1.23)$$

Hence, from (2.1.20) and (2.1.22), the matrix R is related to the Krylov matrix F by the simple formula

$$R = L_T F, \qquad (2.1.24)$$

where L_T is a lower triangular matrix whose entries in row $(i+1)$ are the coefficients of λ^j in $P_i(\lambda)$ divided by the cumulative product $q_2 \cdots q_{i+1}$.

In Lanczos' method, the vectors R_1 and C_1 are chosen arbitrarily subject to the condition that $R_1C_1 = 1$. Some choices of R_1 and C_1 lead to a breakdown because one or both of the products in (2.1.20) and (2.1.21) vanish, but these choices cannot be predicted in advance because the coefficients in the polynomial $P_i(A)$ depend on R_1 and C_1.

2.2 Elimination Method

In the Elimination Method, described by Strachey and Francis[7], the given square matrix A of order n is first transformed to the lower Hessenberg form

$$H = \begin{bmatrix} h_{11} & H_{12} & 0 & \cdots & 0 \\ h_{21} & h_{22} & h_{23} & \cdots & 0 \\ h_{31} & h_{32} & h_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ h_{n1} & h_{n2} & h_{n3} & \cdots & h_{nn} \end{bmatrix} \qquad (2.2.1)$$

using pivotal condensation with row and column interchanges. This Hessenberg matrix H, similar to A, is then transformed into tridiagonal form, when possible, by a special case of Lanczos' Method.

Definition: An elimination transformation of the matrix A is an elementary similarity transformation EAE^{-1} where the elementary matrix E consists of the unit matrix plus one off-diagonal element e_{ij}, which is chosen so that EAE^{-1} has one more zero entry than A.

For a 3×3 matrix, the reduction to Hessenberg form is accomplished by one elementary transformation as follows, assuming $a_{12} \neq 0$.

$$EAE^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & e_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -e_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & 0 \\ a'_{21} & a'_{22} & a'_{23} \\ a_{31} & a_{32} & a'_{33} \end{bmatrix} = A'. \qquad (2.2.2)$$

The primed elements of A' are those that were altered by the transformation. Note that a zero is produced if $e_{23} = a_{13}/a_{12}$. However, if $a_{12} = 0$, a transposition of rows and columns must precede this step. By allowing row and column interchanges during the transformation to lower Hessenberg form, we can ensure that $|e_{ij}| \leq 1$ for optimum numerical stability.

In the elimination procedure that reduces the lower Hessenberg matrix to a tridiagonal matrix, the key step is to subtract $(h_{ij}/h_{j+1,j})$ times row $(j+1)$ from row $i (i > j+1)$ to produce a zero in the ij position. Row and column interchanges cannot be used at this stage since they would alter the form of the upper triangular portion. Hence some of the e_{ij}'s may have magnitudes greater than one. This may lead to numerical instability.

Additional difficulties arise if some of the sub-diagonal entries are zero. Consider an example in which H has the form

$$H = \begin{bmatrix} h_{11} & h_{12} & 0 & 0 \\ 0 & h_{22} & h_{23} & 0 \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} . \qquad (2.2.3)$$

Since $h_{21} = 0$, this method cannot be used to reduce h_{31} and h_{41} to zero.

An analysis of the transformation from Hessenberg to tridiagonal form by this method reveals that the transformation is identical with Lanczos' Method in which $R_1 = C_1^T = [1\ 0\ \cdots\ 0]$, and, consequently, R and C are both lower triangular.

2.3 Kublanovskaya's Method[4]

An earlier method of Hessenberg[2] is essentially equivalent to the first half of the elimination method in producing an upper Hessenberg matrix H similar to A. Given an n'th order square matrix A, find a lower triangular, non-singular matrix C such that

$$C^{-1}AC = H, \qquad (2.3.1)$$

where

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1n} \\ 1 & h_{22} & h_{23} & \cdots & h_{2n} \\ 0 & 1 & h_{33} & \cdots & h_{3n} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 & \cdots & h_{nn} \end{bmatrix} . \qquad (2.3.2)$$

The columns C_i of C and the entries h_{ij} of H are computed as follows. Let (2.3.1) be written as

$$AC = CH. \qquad (2.3.3)$$

By investigating column i on both sides of (2.3.3), it can be seen that

$$AC_i = \sum_{j=1}^{i} C_j h_{ji} + C_{i+1} \qquad (2.3.4)$$

or

$$C_{i+1} = (A - h_{ii}U)C_i - \sum_{j=1}^{i-1} C_j h_{ji} . \qquad (2.3.5)$$

In general, the first i entries of column i in (2.3.4) are used to find h_{ki} for $k = 1, \cdots, i$. The last (n-i) entries of (2.3.5) are used to find C_{i+1} since C is lower triangular. The process is then repeated with i replaced by i+1. The author uses $[1\ 0\ \cdots\ 0]^T$ for C_1. Consider the following example.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{22} & 0 \\ 0 & c_{32} & c_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{22} & 0 \\ 0 & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ 1 & h_{22} & h_{23} \\ 0 & 1 & h_{33} \end{bmatrix}$$

$i = 1: AC_1 = C_1 h_{11} + C_2$

$$\left.\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} = \begin{bmatrix} h_{11} \\ c_{22} \\ c_{32} \end{bmatrix}\right\} \begin{array}{l} \rightarrow h_{11} \\ \rightarrow C_2 \end{array}$$

$i = 2: AC_2 = C_1 h_{12} + C_2 h_{22} + C_3$

$$\left.\begin{bmatrix} a_{12}c_{22} + a_{13}c_{32} \\ a_{22}c_{22} + a_{23}c_{32} \\ a_{32}c_{22} + a_{33}c_{32} \end{bmatrix} = \begin{bmatrix} h_{12} \\ c_{22}h_{22} \\ c_{32}h_{22} + c_{33} \end{bmatrix}\right\} \begin{array}{l} \rightarrow h_{12} \\ \rightarrow h_{22} \text{ if } c_{22} \neq 0 \\ \rightarrow C_3 \end{array}$$

$i = 3: AC_3 = C_1 h_{13} + C_2 h_{23} + C_3 h_{33}$

$$\left.\begin{bmatrix} a_{13}c_{33} \\ a_{23}c_{33} \\ a_{33}c_{33} \end{bmatrix} = \begin{bmatrix} h_{13} \\ c_{22}h_{23} \\ c_{32}h_{23} + c_{33}h_{33} \end{bmatrix}\right\} \begin{array}{l} \rightarrow h_{13} \\ \rightarrow h_{23} \text{ if } c_{22} \neq 0 \\ \rightarrow h_{33} \text{ if } c_{33} \neq 0 \end{array}$$

In this method, C_3 is an arbitrary vector, which may lead to a breakdown whenever $c_{ii} = 0$.

Kublanovskaya describes a transformation to tridiagonal form that is equivalent to the Elimination Method and has the same difficulties with stability and breakdown. Given an upper Hessenberg matrix H of the form (2.3.1), find an upper triangular, non-singular matrix R such that

$$RHR^{-1} = T, \qquad (2.3.6)$$

where T is tridiagonal of the form

$$T = \begin{bmatrix} t_{11} & 1 & 0 & \cdots & 0 \\ t_{21} & t_{22} & 1 & \cdots & 0 \\ 0 & t_{32} & t_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & t_{nn} \end{bmatrix} \qquad (2.3.7)$$

The rows R_i of R and the entries t_{ii} and $t_{i+1,i}$ of T are computed recursively as follows. Let (2.3.6) be written as

$$RH = TR. \qquad (2.3.8)$$

Upon investigating row i on both sides of (2.3.8), we have

$$R_iH = t_{i,i-1}R_{i-1} + t_{ii}R_i + R_{i+1} \qquad (2.3.9)$$

or

$$R_{i+1} = R_i(H - t_{ii}U) - t_{i,i-1}R_{i-1}. \qquad (2.3.10)$$

In general, the $(i-1)^{st}$ and i^{th} entries of (2.3.9) are used to find $t_{i,i-1}$ and t_{ii} respectively. The last (n-i) entries of (2.3.10) are used to find R_{i+1}. The process is again repeated with i replaced by (i+1).

Once again, note that R_1 is an arbitrary vector, which may lead to a breakdown when $r_{ii} = 0$.

2.4 T Algorithm[8]

When possible, the T Algorithm transforms a square complex matrix to tridiagonal form by a sequence of similarity

transformations called quasi-rotation matrices. It does so by using the (i+1, i) entry and the (i, i+1) entry to annihilate the (j, i) and (i, j) entries at the same stage, where $j > (i+1)$.

Definition: A matrix R with entries r_{ij} is called a quasi-rotation matrix if and only if det $R = \pm 1$ and $r_{ij} = \delta_{ij}$ with the possible exceptions of r_{pp}, r_{pq}, r_{qp}, and r_{qq} for any p and q $(p \neq q)$.

The suitable quasi-rotation matrices used at the (i, j) stage are determined from ten categories listed in a table. The categories are listed according to whether the elements in the (i, j), (j, i), (i, i+1), and (i+1, i) positions are zero or non-zero. Therefore, a total of sixteen cases result.

The author admits that this algorithm might involve extensive programming because of the table look-up. He also gives a sufficiency theorem for tridiagonalization of a matrix by the T Algorithm.

For three of the sixteen possible cases, the algorithm can fail; however, the author presents a modified T Algorithm which yields a matrix that is almost lower triangular.

Application of the T Algorithm to certain matrices leads to numerical instability. When the algorithm does work, it leads to a unique transformation matrix.

2.5 La Budde's Method[5]

This method, like the others, seeks to find non-singular transformation matrices R and C such that for a square matrix A of order n,

$$RAC = T \tag{2.5.1}$$

and

$$RC = U \tag{2.5.2}$$

where T is tridiagonal. Furthermore, the matrices R and C are each the products of (n-2) matrices; i.e.,

$$\begin{aligned} R &= R^{(n-2)} \cdots R^{(1)} \\ C &= C^{(1)} \cdots C^{(n-2)} \end{aligned} \tag{2.5.3}$$

At stage j in this method, the matrix A has been transformed into a similar matrix $A^{(j)}$ of the form

$$A^{(j)} = \left[\begin{array}{ccc|c} & T & & 0 \\ & & & \\ & & & W^T \\ \hline 0 & V & & B \end{array}\right] \begin{array}{l} \} \text{ j rows} \\ \\ \\ \} \text{ n-j rows} \end{array} \tag{2.5.4}$$

where T is tridiagonal, V and W are (n-j) dimensional column vectors, and B is an (n-j) × (n-j) submatrix.

Starting with two column vectors X and Y of dimension (n-j), we construct the elementary matrices $R^{(j)}$ and $C^{(j)}$ where

$$R^{(j)} = \begin{bmatrix} U_j & 0 \\ 0 & U_{n-j}+aXY^T \end{bmatrix}, \quad C^{(j)} = \begin{bmatrix} U_j & 0 \\ 0 & U_{n-j}+bXY^T \end{bmatrix}. \qquad (2.5.5)$$

In order that $R^{(j)}C^{(j)} = U$, the vectors X and Y must be related to the scalars a and b by the equation

$$Y^TX = -\frac{(a+b)}{ab}. \qquad (2.5.6)$$

The matrix $A^{(j+1)} = R^{(j)}A^{(j)}C^{(j)}$ differs from $A^{(j)}$ by replacing V, W^T, and B by V', W'^T, and B' where

$$\begin{aligned} V' &= (U_{n-j} + aXY^T)V \\ W'^T &= W^T(U_{n-j} + bXY^T) \\ B' &= (U_{n-j} + aXY^T)B(U_{n-j} + bXY^T). \end{aligned} \qquad (2.5.7)$$

The scalars a and b and the vectors X and Y are chosen so that (2.5.6) is satisfied, all but the first entries of V' and W' are zero, and

$$p = W'^TV' = W^TV. \qquad (2.5.8)$$

The choice is not unique, but it must avoid values of a and b near zero.

Scalars c and d are defined by

$$\frac{1}{c} = W^TX, \quad \frac{1}{d} = Y^TV. \qquad (2.5.9)$$

From (2.5.7) and (2.5.9) we obtain

$$X = (V'-V)\frac{d}{a}, \quad Y = (W'-W)\frac{c}{b} \qquad (2.5.10)$$

and substituting (2.5.10) into (2.5.9), we have

$$\frac{1}{c} = (w_1 v_1' - p)\frac{d}{a}, \quad \frac{1}{d} = (w_1' v_1 - p)\frac{c}{b} . \qquad (2.5.11)$$

Then,

$$(\frac{a}{cd} + p)(\frac{b}{cd} + p) = (w_1 v_1')(w_1' v_1) = p a_{j+1,j} a_{j,j+1} . \qquad (2.5.12)$$

The solution for $\frac{1}{cd}$ is

$$\frac{1}{cd} = \frac{\{-p(a+b) \pm \sqrt{p^2(a+b)^2 + 4abp(a_{j,j+1}a_{j+1,j} - p)}\}}{2ab} \qquad (2.5.13)$$

In order that all but the first entries of W' and V' be zero, it is necessary that

$$x_k = -(\frac{d}{a})a_{kj}, \quad y_k = -(\frac{c}{b})a_{jk} \qquad (2.5.14)$$

for k = j+2, $\cdots$, n. Then x_{j+1} and y_{j+1} are expressed as

$$x_{j+1} = \left\{\frac{1}{cd} + \frac{p - a_{j,j+1}a_{j+1,j}}{a}\right\}\frac{d}{a_{j,j+1}}$$

$$y_{j+1} = \left\{\frac{1}{cd} + \frac{p - a_{j,j+1}a_{j+1,j}}{b}\right\}\frac{c}{a_{j+1,j}} \qquad (2.5.15)$$

In order to solve (2.5.14) and (2.5.15) for the x's and y's, it is necessary that the scalars a, b, c, and d be finite and non-zero. For some matrices, it can happen that the scalar product p is zero at some stage; then the algorithm breaks down because $\frac{1}{cd} = 0$. Otherwise, since either c or d is arbitrary in (2.5.13), there is no loss of generality if we let d = 1. To solve for c, the product ab as well as p must be non-zero. The signs of a and b can be chosen so that the discriminant in (2.5.13) is positive, and the

sign of the radical is chosen so that c is finite. The difficulty in choosing a and b is to avoid having p = 0 at the next stage. Wang and Gregory[9] point out that such a choice is not always possible.

Parlett[6] has shown that when A is an unreduced lower Hessenberg matrix, this method is identical with the Elimination Method.

Lanczos' Method appears to be the most general since the Elimination Method and Kublanovskaya's Method are special cases of the former. Furthermore, Lanczos' Method appears to be better than the T Algorithm or the method of LaBudde. The T Algorithm involves a table look-up with sixteen cases. LaBudde's Method involves rational operations and the solution of a quadratic equation at each stage, but with no assurance of avoiding a breakdown at the next stage.

Becasue of the generality of Lanczos' Method, a modification of his method is presented in Chapter III. This modified procedure is then used to tridiagonalize the matrix A.

III A NEW SCHEME FOR TRIDIAGONALIZING AN ARBITRARY REAL SQUARE MATRIX

In order to bring the new tridiagonalization procedure into perspective, it is necessary to examine the ways in which Lanczos' Method can break down. This examination, as well as an introduction to the new two-step procedure, is presented in Section 3.1. The first step of the procedure is discussed in Section 3.2. Section 3.3 describes the second step and includes a lemma and two theorems which show that it is always possible to transform an arbitrary real non-symmetric matrix into tridiagonal form without using the eigenvalues.

3.1 Analysis of the Problem

In order to overcome the difficulties associated with Lanczos' Method, it is necessary to examine the ways in which his method can break down. At any stage, three cases may occur.

The regular case, or Case 1, is the case in which the algorithm proceeds from one stage to the next without a breakdown.

There are four ways in which the algorithm can break down; these are grouped into two irregular cases, Case 2

and Case 3. The reason for this grouping will become apparent when we consider how to proceed with the algorithm following a breakdown.

A breakdown occurs when, at any stage of the algorithm, we have $e_{i+1}q_{i+1}R_{i+1}C_{i+1} = 0$. In terms of the vectors X and Y in equations (2.1.11) and (2.1.12), a breakdown occurs whenever $XY = 0$.

We will say that a Case 2 breakdown occurs when either $X = 0$, or $Y = 0$, or both $X = 0$ and $Y = 0$. A Case 3 breakdown occurs when $X \neq 0$, $Y \neq 0$, but $XY = 0$. To show that Case 3 can indeed occur, consider the following example.

Let A, R_1, and C_1 be as shown.

$$A = \begin{bmatrix} 1 & -1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_1 = [1 \quad 0 \quad 0 \quad 0], \quad C_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

For $i = 1$, we would compute $t_1 = R_1AC_1 = 1$.

Then

$$X = R_1(A - t_1U) = [0 \quad -1 \quad 1 \quad 0] \neq 0$$

and

$$Y = (A - t_1U)C_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \neq 0$$

but

$$XY = 0.$$

Let m be the degree of the minimal polynomial. A Case 2 breakdown will always occur at the $(m+1)^{st}$ stage. The breakdown, in which X = 0 and Y = 0, is unavoidable. We will show that a simple procedure can be used at the stage in which a Case 2 breakdown occurs to permit continuation with Case 1. Furthermore, a Case 2 breakdown in which $X \neq 0$ and Y = 0 will never occur, because the new algorithm does not generate the columns of C in the same manner as the rows of R. As will be pointed out in a lemma, the columns C_i of C are computed recrusively as functions of rows $R_1, \cdots, R_{i-1}$ of R and columns $C_1, \cdots, C_{i-1}$ of C.

As a first step in transforming the matrix A to tridiagonal form T, we will transform it to upper Hessenberg form A_H by using Householder[3] transformations to be described in Section 3.2 such that

$$A_H = HAH^* \qquad (3.1.1)$$

$$HH^* = U \text{ (H is unitary)}. \qquad (3.1.2)$$

The upper Hessenberg matrix A_H has the form

$$A_H = \begin{bmatrix} a_{h11} & a_{h12} & a_{h13} & \cdots & a_{h1n} \\ e_2 & a_{h22} & a_{h23} & \cdots & a_{h2n} \\ 0 & e_3 & a_{h33} & \cdots & a_{h3n} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & a_{hnn} \end{bmatrix} . \qquad (3.1.3)$$

The reason for this transformation is to introduce the desired zeros below the subdiagonal by stable computation.

A unitary transformation preserves the lengths of vectors and does not involve division by small quantities.

In the second step, a matrix R is computed such that $RA_HR^{-1} = RA_HC = T$ is tridiagonal, and has the same subdiagonal entries as A_H. Whenever A_H has a subdiagonal with nonvanishing entries, Theorem 1, in Section 3.3, shows how to find an initial vector R_1 such that there will be no breakdown in the algorithm. The occurrence of a subdiagonal zero entry in A_H invalidates the hypothesis of Theorem 1, but may or may not cause a breakdown.

When a Case 3 breakdown does occur, the algorithm necessitates returning to the last occurrence of a Case 2 breakdown (whereby the first stage is treated like a Case 2 breakdown). At this point, the algorithm shows, by Theorem 2 in Section 3.3, how to progress one stage past the point at which the Case 3 breakdown occurred.

The combined two steps of the procedure for tridiagonalizing an arbitrary real non-symmetric matrix A can be represented by the change of variables

$$Y(t) = \underline{S}X(t) \qquad (3.1.4)$$

where

$$\underline{S} = RH \qquad (3.1.5)$$

The original state model

$$\frac{d}{dt}X(t) = AX(t), \quad F(t) = CX(t) \qquad (3.1.6)$$

then becomes

$$\frac{d}{dt}Y(t) = TY(t), \quad F(t) = C'Y(t) \qquad (3.1.7)$$

where $T = \underline{S}A\underline{S}^{-1}$ and $C' = C\underline{S}^{-1}$.

3.2 Householder Transformation to Upper Hessenberg Form[3]

As mentioned in Section 3.1, our first aim is to find a unitary matrix H such that $HAH^* = A_H$ is in upper Hessenberg form. The matrix H is found as the product of (n-2) Householder matrices H_i; i.e.,

$$H = H_{n-2} \cdots H_1 . \tag{3.2.1}$$

Let $A = A^{(1)}$ be partitioned as

$$A^{(1)} = \begin{bmatrix} a_{11}^{(1)} & A_{12}^{(1)} \\ A_{21}^{(1)} & A_{22}^{(1)} \end{bmatrix} \tag{3.2.2}$$

where $A_{12}^{(1)}$ is an (n-1) dimensional row vector, $A_{21}^{(1)}$ is an (n-1) dimensional column vector, and $A_{22}^{(1)}$ is an (n-1) $\times$ (n-1) submatrix. Let α_1 be defined by

$$\alpha_1 = + \sqrt{A_{21}^{(1)*} A_{21}^{(1)}}, \tag{3.2.3}$$

and let $X^{(1)}$ be an n dimensional column vector defined by

$$X^{(1)} = \begin{bmatrix} 0 \\ - - - \\ x_2^{(1)} \\ x_3^{(1)} \\ \vdots \\ x_n^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ - - - \\ \\ A_{21}^{(1)} \\ \\ \end{bmatrix} - \begin{bmatrix} 0 \\ - - - \\ \pm\alpha_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.2.4}$$

where the sign of α_1 is chosen to maximize $|x_2^{(1)}|$. The first Householder matrix H_1 is defined by

$$H_1 = U - 2Z_1 \tag{3.2.5}$$

where Z_1 is the following Hermitian idempotent of rank one:

$$Z_1 = \frac{X^{(1)}X^{(1)*}}{X^{(1)*}X^{(1)}} \tag{3.2.6}$$

It is easily shown that

$$H_1^* = H_1, \quad H_1H_1^* = H_1^2 = U. \tag{3.2.7}$$

Transforming $A^{(1)}$ by this H_1, a partially transformed matrix $A^{(2)}$ is obtained:

$$A^{(2)} = H_1A^{(1)}H_1^* . \tag{3.2.8}$$

In general, for i = 1, $\cdots$, n-2, we have

$$A^{(i)} = \left[\begin{array}{c|c} A_{11}^{(i)} & A_{12}^{(i)} \\ \hline 0 \;\; A_{21}^{(i)} & A_{22}^{(i)} \end{array}\right] , \tag{3.2.9}$$

where $A_{11}^{(i)}$ is an i^{th} order upper Hessenberg matrix, $A_{12}^{(i)}$ is an (i) $\times$ (n-i) dimensional submatrix, $A_{21}^{(i)}$ is an (n-i) dimensional column vector, and $A_{22}^{(i)}$ is an (n-i) $\times$ (n-i) submatrix. Let

$$\alpha_i = + \sqrt{A_{21}^{(i)*}A_{21}^{(i)}} \tag{3.2.10}$$

and

$$X^{(i)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline x_{i+1}^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline \\ A_{21}^{(i)} \\ \\ \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline \pm\alpha_i \\ \vdots \\ 0 \end{bmatrix} , \qquad (3.2.11)$$

where the sign of α_i is chosen to maximize $|x_{i+1}^{(i)}|$. As before, the Householder matrix H_i is defined by

$$H_i = U - 2 \left(\frac{X^{(i)} X^{(i)*}}{X^{(i)*} X^{(i)}}\right) , \qquad (3.2.12)$$

and $A^{(i)}$ is transformed into

$$A^{(i+1)} = H_i A^{(i)} H_i^* . \qquad (3.2.13)$$

When $i = (n-2)$, the resulting matrix $A^{(n-1)}$ is in upper Hessenberg form; i.e.,

$$A_H = A^{(n-1)} = H_{n-2} A^{(n-2)} H_{n-2} = HAH^* . \qquad (3.2.14)$$

It is noteworthy at this time to point out that if A is symmetric, the well known method of Givens is precisely this Householder transformation applied to A, and the final matrix is tridiagonal.

3.3 A Modified Lanczos Transformation from Hessenberg to Tridiagonal Form

Given the upper Hessenberg matrix A, our goal is to describe an algorithm to construct a unit upper

triangular matrix R (in the event that a Case 3 occurs, R might not be unit upper triangular) that will transform A into a similar tridiagonal matrix T; i.e.,

$$RAR^{-1} = RAC = T \ . \tag{3.3.1}$$

In terms of the matrix G = RA, this becomes

$$TR = RA = G \ . \tag{3.3.2}$$

The matrices A, G, T, and R have the forms:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ e_2 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & e_3 & a_{33} & \cdots & a_{3n} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 \cdots & e_n & a_{nn} \end{bmatrix}, \quad G = \begin{bmatrix} g_{11} & g_{12} & g_{13} & \cdots & g_{1n} \\ e_2 & g_{22} & g_{23} & \cdots & g_{2n} \\ 0 & e_3 & g_{33} & \cdots & g_{3n} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 \cdots & e_n & g_{nn} \end{bmatrix} \tag{3.3.3}$$

$$T = \begin{bmatrix} t_1 & q_2 & 0 & \cdots & 0 \\ e_2 & t_2 & q_3 & \cdots & 0 \\ 0 & e_3 & t_3 & \cdots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 \cdots & e_n & t_n \end{bmatrix}, \quad R = \begin{bmatrix} 1 & r_{12} & r_{13} & \cdots & r_{1n} \\ 0 & 1 & r_{23} & \cdots & r_{2n} \\ 0 & 0 & 1 & \cdots & r_{3n} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Recurrence Formulas

We equate the i^{th} rows of TR and G in (3.3.2).

$$e_i R_{i-1} + t_i R_i + q_{i+1} R_{i+1} = G_i \ . \tag{3.3.4}$$

Comparing the i^{th}, $(i+1)^{st}$, and j^{th} $(j > i+1)$ entries, it can be seen that

$$i: \quad e_i r_{i-1,i} + t_i = g_{ii} \tag{3.3.5}$$

$$i+1: \quad e_i r_{i-1,i+1} + t_i r_{i,i+1} + q_{i+1} = g_{i,i+1} \tag{3.3.6}$$

$$j: \quad e_i r_{i-1,j} + t_i r_{ij} + q_{i+1} r_{i+1,j} = g_{ij}. \tag{3.3.7}$$

Since the subdiagonal entries of T are identical to those of A, then (3.3.5) and (3.3.6) may be used to compute the coefficients t_i and q_{i+1} in T as functions of R_{i-1}, R_i, and A; i.e.,

$$t_i = g_{ii} - e_i r_{i-1,i} = a_{ii} + e_{i+1} r_{i,i+1} - e_i r_{i-1,i} \tag{3.3.8}$$

$$q_{i+1} = g_{i,i+1} - e_i r_{i-1,i+1} - t_i r_{i,i+1} \tag{3.3.9}$$

If $q_{i+1} \neq 0$, then (3.3.4) may be used to compute the vector R_{i+1};

$$R_{i+1} = \frac{1}{q_{i+1}} (G_i - e_i R_{i-1} - t_i R_i). \tag{3.3.10}$$

The case where $q_{i+1} = 0$ will be treated later.

Krylov Factorization

Let F be the Krylov matrix whose rows are the iterates of R_1 under A.

$$F = \begin{bmatrix} R_1 \\ R_1 A \\ \cdot \\ \cdot \\ \cdot \\ R_1 A^{n-1} \end{bmatrix} \tag{3.3.11}$$

In the regular case, F has a factorization as the triple

product of a unit lower triangular matrix L, a diagonal matrix D, and a unit upper triangular matrix V; i.e.,

$$F = LDV \tag{3.3.12}$$

where L, D, and V have the forms:

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \ell_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & 1 \end{bmatrix}, \quad D = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix},$$

$$V = \begin{bmatrix} 1 & v_{12} & \cdots & v_{1n} \\ 0 & 1 & \cdots & v_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{3.3.13}$$

Let $f\binom{1,2,\cdots,i-1,i}{1,2,\cdots,i-1,j}$ be the minor of F formed from rows $1,2,\cdots,i-1,i$ and columns $1,2,\cdots,i-1,j$; further, let $f_i = f\binom{1,2,\cdots,i-1,i}{1,2,\cdots,i-1,i}$ be the i^{th} leading principal minor of F. It can then be shown that the entries in L and V are expressible as:

$$\ell_{ij} = \frac{f\binom{1,2,\cdots,j-1,i}{1,2,\cdots,j-1,j}}{f_j}, \quad v_{ij} = \frac{f\binom{1,2,\cdots,i-1,i}{1,2,\cdots,i-1,j}}{f_i}. \tag{3.3.14}$$

Since D and F have equal leading principal minors, it follows that

$$d_i = \frac{f_i}{f_{i-1}} \quad \text{where } f_0 = 1. \tag{3.3.15}$$

It will now be shown that the upper triangular matrix V in (3.3.12) is identical to the matrix R in (3.3.1). The vector R_{i+1} is a linear combination of the vectors R_1, R_1A, $\cdots$, R_1A^i, in which R_1A^i has a non-vanishing coefficient. Hence, the matrix R is a left multiple of F by a lower triangular matrix, say $(L'D')^{-1}$, where L' is a unit lower triangular matrix and D' is a diagonal matrix. Then,

$$F = L'D'R = LDV. \tag{3.3.16}$$

By rewriting (3.3.16) as

$$(L')^{-1}LD = D'RV^{-1}, \tag{3.3.17}$$

where the left side is lower triangular and the right side is upper triangular, it can be seen that both sides must be diagonal. Hence, $(L')^{-1}L - RV^{-1} = U$ and $D = D'$. Therefore, the factorization (3.3.12) is unique and $V = R$.

Note that $(L'D')^{-1} = (LD)^{-1}$ is the matrix L_T referred to in equation (2.1.24).

It can be shown that $d_i = q_2 q_3 \cdots q_i$, so the q_{i+1}'s are related to the entries in D and F by the formulas

$$q_{i+1} = \frac{d_{i+1}}{d_i} = \frac{f_{i-1}f_{i+1}}{f_i^2} \tag{3.3.18}$$

and

$$f_i = \prod_{j=2}^{i} d_j = \prod_{j=2}^{i} q_j^{i+1-j} \text{ where } d_1 = f_1 = f_0 = 1. \tag{3.3.19}$$

The t_i's are related to the subdiagonal entries in L by

$$t_1 = \ell_{21}, \quad t_i = \ell_{i+1,i} - \ell_{i,i-1} \text{ for } i = 2, \cdots, n-1, \tag{3.3.20}$$

and t_n must be computed using either (3.3.8) with $e_{n+1}r_{n,n+1} = 0$ or (2.1.4).

At stage (i+1), the computation of R_{i+1} requires that $q_{i+1} \neq 0$. This condition is met if and only if $f_{i+1} \neq 0$. Hence, the non-vanishing of the leading principal minors of F is necessary and sufficient for the continuation of the algorithm in the regular case.

We will now investigate the minors f_i as functions of R_1 and show, by means of two theorems, that it is always possible to transform a given Hessenberg matrix to tridiagonal form without involving the eigenvalues. Furthermore, the existence proofs are constructive. Although they may not be computationally optimal, they do include methods for determining the vector R_1 which can be written as

$$R_1 = [1 \; r_2 \; r_3 \; \cdots \; r_n]. \qquad (3.3.21)$$

Let the notation $f_{ij}(r_2, \cdots, \underline{r_k})$ imply that the entries f_{ij} of the Krylov matrix F depend linearly on $r_2, \cdots, r_k$ for $2 \leq k < (i+j)$, and if $(i+j) = (k+1) \leq (n+1)$, they involve r_k explicitly in a term $e_2e_3\cdots e_kr_k$. We emphasize this latter point by underlining r_k in the notation $f_{ij}(r_2, \cdots, \underline{r_k})$. Hence, the matrix F can be written in the following form:

$$F = \tag{3.3.22}$$

$$\begin{bmatrix} 1 & r_2 & r_3 & \cdots & r_n \\ f_{21}(\underline{r_2}) & f_{22}(r_2,\underline{r_3}) & f_{23}(r_2,r_3,\underline{r_4}) & \cdots & f_{2n}(r_2,\ldots,r_n) \\ f_{31}(r_2,\underline{r_3}) & f_{32}(r_2,r_3,\underline{r_4}) & f_{33}(r_2,\ldots,r_5) & \cdots & f_{3n}(r_2,\ldots,r_n) \\ \vdots & \vdots & \vdots & & \vdots \\ f_{n1}(r_2,\ldots,\underline{r_n}) & f_{n2}(r_2,\ldots,r_n) & f_{n3}(r_2,\ldots,r_n) & \cdots & f_{nn}(r_2,\ldots,r_n) \end{bmatrix}.$$

Theorem 1: Given the Hessenberg matrix A in the form of (3.1.3) in which none of the e_i's are zero (or close to zero), there exists an initial row vector R_1 as in (3.3.21) leading to a unit upper triangular matrix R that transforms the matrix A to tridiagonal form. More specifically, all f_i's can be made non-zero.

Proof: Form the Krylov matrix F as in (3.3.22). It can be seen that $f_1 = 1 \neq 0$. Using functional notation, it can also be seen that $f_2 = f_2(r_2,r_3)$. The method proceeds as follows. For f_2, assume temporarily that $r_3 = 0$, and select r_2 such that f_2 is not zero. Once selected, consider r_2 to be fixed from this point on; i.e., $f_{ij}(r_2,r_3,\ldots r_k)$ becomes $f_{ij}(r_3,\ldots r_k)$ in general. Following this line of thought, we see that $f_3 = f_3(r_3,r_4,r_5)$. Again assume that $r_4 = r_5 = 0$, and select r_3 such that $f_2(r_3) \neq 0$ and $f_3(r_3) \neq 0$. Note that at this stage there are four values of r_3 to be avoided since f_2 is linear in r_3, and f_3 is cubic in r_3. Continuing as before, we now assume that r_3 is also fixed and $f_{ij}(r_3,r_4,\ldots,r_k)$ becomes $f_{ij}(r_4,\ldots,r_k)$.

In general, at stage i we are finding values of r_i such that $f_j \neq 0$ for $j \leq i$ where $f_j = f_j(r_i \ldots)$. This is always possible since we are dealing with a matrix F of finite order; hence, there are only a finite number of values for r_i to be avoided at each stage, and since for f_i the term $e_2 \ldots e_i r_i$ appears in which the e_j's are non-zero. Therefore, all f_i's can be made non-zero.

Q.E.D.

It is important to point out that the e_i's may be completely arbitrary (including zero) and yet have the factorization work without breakdown. It is only necessary that one be able to define r_i so that the f_j's involving r_i are not zero for $j \leq i$. We shall examine the computational technique in order to motivate the second theorem.

Computational Technique

Assume that the algorithm for factoring the Krylov matrix F has computed the vectors $R_2, \ldots, R_k$, but breaks down at stage (k+1) because $f_{k+1} = 0$. The action taken at this stage depends on whether the breakdown occurs under Case 2 or Case 3.

Whether or not a breakdown occurs, column C_{k+1} is computed as a function of $R_1, \ldots, R_k$ and $C_1, \ldots, C_k$ according to the following lemma.

Lemma: Let R be a unit upper triangular matrix of order n. Then the column C_{k+1} of the matrix $C = R^{-1}$ can be computed recursively as a function of rows $R_1, \ldots, R_k$ of R and

columns $C_1,\ldots,C_k$ of C by the formula

$$C_{k+1} = (U - \sum_{i=1}^{k} C_i R_i)U_{k+1} \qquad (3.3.23)$$

where U_{k+1} is the $(k+1)^{st}$ column of the unit matrix U.

Proof: Since $C = R^{-1}$, then

$$U = RC = CR = \sum_{i=1}^{k} C_i R_i + \sum_{i=k+1}^{n} C_i R_i \ . \qquad (3.3.24)$$

The second summation in (3.3.24) defines an idempotent matrix W_{k+1} where

$$W_{k+1} = U - \sum_{i=1}^{k} C_i R_i = \sum_{i=k+1}^{n} C_i R_i \ . \qquad (3.3.25)$$

The last (n-i) entries of C_i are zero, and the first (i-1) entries of R_i are zero since R and C are both unit upper triangular. Hence, the matrix W_{k+1} has the following form in which C_{k+1} appears as column (k+1):

$$W_{k+1} = \begin{bmatrix} 0 & \ldots & 0 & c_{1,k+1} & * & \ldots & * \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \ldots & 0 & c_{k,k+1} & * & \ldots & * \\ 0 & \ldots & 0 & 1 & * & \ldots & * \\ 0 & \ldots & 0 & 0 & 1 & \ldots & * \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \ . \qquad (3.3.26)$$

The $(k+1)^{st}$ column of W_{k+1} is $W_{k+1}U_{k+1} = C_{k+1}$. Hence, (3.3.23) follows from (3.3.25).

Q.E.D.

The reason for introducing the above algorithm for computing the columns C_{k+1} of the matrix C will become

apparent when we next discuss the actual technique of applying the Krylov matrix factorization.

Consider the Krylov matrix F to be initially partitioned as follows

$$F = F_1 = \begin{bmatrix} d_1 & F_{R1} \\ F_{L1} & H_1 \end{bmatrix} , \qquad (3.3.27)$$

where F_{R1} is an (n-1) dimensional row vector, F_{L1} is an (n-1) dimensional column vector, and H_1 is an (n-1) $\times$ (n-1) submatrix. Since d_1 is not zero ($d_1 = 1$ since $r_{11} = 1$), then F_1 may be factored into the product of three matrices

$$F_1 = \begin{bmatrix} 1 & 0 \\ \frac{F_{L1}}{d_1} & U_{n-1} \end{bmatrix} \begin{bmatrix} d_1 & 0 \\ 0 & F_2 \end{bmatrix} \begin{bmatrix} 1 & F_{R1}/d_1 \\ 0 & U_{n-1} \end{bmatrix} = L_1 D_1 V_1 \qquad (3.3.28)$$

where

$$F_2 = \frac{d_1 H_1 - F_{L1} F_{R1}}{d_1} , \qquad (3.3.29)$$

and where $R_1 = [1 \; F_{R1}/d_1]$ is read from V_1.

Again, let F_2 be partitioned as

$$F_2 = \begin{bmatrix} d_2 & F_{R2} \\ F_{L2} & H_2 \end{bmatrix} . \qquad (3.3.30)$$

If $d_2 = q_2 \neq 0$, then F_2 may be factored as follows:

$$F_2 = \begin{bmatrix} 1 & 0 \\ \frac{F_{L2}}{d_2} & U_{n-2} \end{bmatrix} \begin{bmatrix} d_2 & 0 \\ 0 & F_3 \end{bmatrix} \begin{bmatrix} 1 & F_{R2}/d_2 \\ 0 & U_{n-2} \end{bmatrix} = L_2 D_2 V_2 \qquad (3.3.31)$$

where $R_2 = [0\ 1\ F_{R2}/d_2]$ from V_2 and where

$$F_3 = \frac{d_2 H_2 - F_{L2} F_{R2}}{d_2} \quad . \tag{3.3.32}$$

In general, at the $(k+1)^{st}$ stage, we have F_{k+1} partitioned as

$$F_{k+1} = \begin{bmatrix} d_{k+1} & F_{R,k+1} \\ F_{L,k+1} & H_{k+1} \end{bmatrix} \tag{3.3.33}$$

where we have assumed that $d_i \neq 0$ for $i = 1, \cdots, k$. Note that all the previous factorizations may be combined into a single factorization

$$F = LDV = \begin{bmatrix} 1 & 0 & & & \\ & 1 & & 0 & \\ \frac{F_{L1}}{d_1} & & & & \\ & \frac{F_{L2}}{d_2} & \cdots & 1 & 0 \\ & & & \frac{F_{Lk}}{d_k} & U_{n-k} \end{bmatrix} \begin{bmatrix} d_1 & & & & \\ & d_2 & & 0 & \\ & & \cdot & & \\ & & & d_k & 0 \\ 0 & & & 0 & F_{k+1} \end{bmatrix}$$

$$\begin{bmatrix} 1 & & F_{R1}/d_1 & & \\ 0 & 1 & & F_{R2}/d_2 & \\ & & \vdots & & \\ & & 1 & & F_{Rk}/d_k \\ & 0 & & 0 & U_{n-k} \end{bmatrix} \quad . \tag{3.3.34}$$

Now, assume that $d_{k+1} = 0$. This is the first indication of a breakdown. To determine whether it is a Case 2 or Case 3 breakdown, we must also examine the entries of $F_{R,k+1}$ in (3.3.33). The first row of F_{k+1} can be expressed in terms of the entries q_i of the matrix T and the row R_{k+1} of the matrix R as

$$[0\cdots 0 d_{k+1} F_{R,k+1}] = q_2 q_3 \cdots q_{k+1} R_{k+1} \ . \qquad (3.3.35)$$

The product $q_2 \cdots q_k$ is non-zero since we assumed $d_i \neq 0$ for $i = 1,\cdots,k$. Hence, $d_{k+1} = 0$ implies that $q_{k+1} r_{k+1,k+1} = 0$. In Case 2 where $F_{R,k+1}$ is the zero vector, we choose $q_{k+1} = 0$, and determine R_{k+1} as the sum of the $(k+1)^{st}$ row of the idempotent W_{k+1} and any desired linear combination of lower rows. This choice, like the initial choice of R_1, is not unique. We then continue with the factorization as before, with the exception that for $i > (k+1)$,

$$[0\cdots 0 \ d_i \ F_{Ri}] = q_{k+2} \cdots q_i R_i \qquad (3.3.36)$$

where $d_i = q_{k+2} \cdots q_i$.

It is noteworthy to point out at this time that if m is the degree of the minimal polynomial of A ($m < n$), then there will always be a breakdown at the $(m+1)^{st}$ stage. Since A is upper Hessenberg, the Krylov matrix $[C_1, AC_1, \cdots, A^{n-1}C_1]$ is upper triangular with m^{th} diagonal entry $e_2 e_3 \cdots e_m$. If $m < n$, then $(m+1)$ columns are dependent, and $e_{m+1} = 0$.

If however, $F_{R,k+1}$ has some non-zero entry, one cannot set $q_{k+1} = 0$. This causes a Case 3 breakdown in which $r_{k+1,k+1} = R_{k+1}C_{k+1} = 0$. It is the occurrence of a Case 3 breakdown that motivates the second theorem.

If a Case 3 breakdown occurs, we must revert to the last occurrence of a Case 2 breakdown, say at stage j (or to the start if no such breakdown occurred after j = 1), and attempt to select a new vector R_j with the usual constraint that $r_{ji} = 0$ for i = 1, $\cdots$, j-1 and $r_{jj} = 1$. Under some circumstances, it is not possible to find a vector R_j for the given Hessenberg matrix A such that $f_{k+1} \neq 0$, as the following example indicates.

Example: Let the matrix A be as follows, and let R_1 be represented as indicated in the first row of the Krylov matrix F.

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & x & y \\ 0 & 0 & 1+x+y \\ 0 & 0 & 1+x+y \end{bmatrix} . \quad (3.3.37)$$

Here it is obvious that $f_2 = 0$ for every choice of R_1.

Under these circumstances, when an R_j cannot be found such that $f_{k+1} \neq 0$, a series of similarity transformations are performed before the factorization is again used. These transformations consist of using the matrices R_a, H_R, and H_C to be discussed below. The final matrix R may no longer be unit upper triangular as will be shown later in equation (3.3.46).

Let R_a be a matrix formed from rows $R_1, \cdots, R_k$ of R and the last (n-k) rows of U. Furthermore, let the matrix W be defined by

$$W = R_a A R_a^{-1} \tag{3.3.38}$$

where

$$R_a = \begin{bmatrix} R_{11} & R_{12} \\ 0 & U \end{bmatrix}. \tag{3.3.39}$$

It can then be shown that W has the form

$$W = \left[\begin{array}{c|c} T_{11} & 0 \\ \hline 0 & B \\ \hline 0 & W_{22} \end{array}\right] = \begin{bmatrix} W_{11} & W_{12} \\ 0 & W_{22} \end{bmatrix}, \tag{3.3.40}$$

where T_{11} is tridiagonal, and where $B = [b_{k+2} \cdots b_n]$ has at least one non-zero entry, which appeared in a different form in $F_{R,k+1}$ in equation (3.3.33). Note that the reason $W_{21} = 0$ is that had there been an e_{k+1} in the upper right corner, then Theorem 1 indicates that there was an R_j which would not have led to this Case 3 breakdown; hence, the assumption that $W_{21} = 0$ is valid.

The matrix H_R is a Householder matrix of the form

$$H_R = \begin{bmatrix} U & 0 \\ 0 & \underline{H}_R \end{bmatrix}, \tag{3.3.41}$$

which is used to collapse all the non-zero entries of B into q and to force $b_i = 0$ for $(k+2) \leq i \leq n$; see equation

(3.3.43). Referring to Section 3.2, $\underline{H}_R$ is row-determined rather than being column-determined. Letting the matrix W' be defined by

$$W' = H_R W H_R^* , \qquad (3.3.42)$$

then W' has the form:

$$W' = \left[\begin{array}{c|cc} T_{11} & 0 & \\ \hline & q & 0 \\ \hline 0 & & W'_{22} \end{array}\right] = \begin{bmatrix} W'_{11} & W'_{12} \\ 0 & W'_{22} \end{bmatrix} . \qquad (3.3.43)$$

Finally, the transformation defined by

$$A' = H_C W' H_C^* , \qquad (3.3.44)$$

in which H_C is a column-determined Householder matrix, has the effect of transforming W'_{22} into upper Hessenberg form. Hence, the form of A' is

$$A' = \left[\begin{array}{c|cc} & 0 & 0 \\ T_{11} & & \\ & q & 0 \\ \hline 0 & t & \underline{B} \\ \hline & e & \\ 0 & \vdots & A'_{22} \\ & 0 & \end{array}\right] = \left[\begin{array}{c|c} T'_{11} & 0 \\ \hline & \underline{B} \\ \hline e & \\ \vdots & A'_{22} \\ 0 & \end{array}\right] , \qquad (3.3.45)$$

where the tridiagonal block T'_{11} is $(k+1) \times (k+1)$ and A'_{22} is in upper Hessenberg form.

The result of these three transformations is that the order of the leading tridiagonal block has been increased by one.

If $e = 0$ in (3.3.45), then A' has the form of W in (3.3.40), and the above process would be repeated if $\underline{B} \neq 0$. However, if $\underline{B} = 0$, then A' could be repartitioned to increase the size of the tridiagonal block by one.

If $e \neq 0$ in (3.3.45), then Theorem 1 shows that there is an initial row vector R_1 for which the Krylov factorization again applies. The following theorem has just been proved.

Theorem 2: When a Case 3 breakdown occurs at stage (k+1) of the factorization defined by (3.3.27) through (3.3.33), it is possible to progress to at least the $(k+2)^{nd}$ stage in three steps: (1) transform A into $R_a A R_a^{-1} = W$ where R_a is constructed from R as described above, (2) transform W into $H_R W H_R^* = W'$ by row-determined Householder transformations with product H_R, (3) transform W' into $H_C W' H_C^* = A'$ by column-determined Householder transformations with product H_C.

The above transformations can be abbreviated by letting

$$\underline{R} = H_C H_R R_a ; \tag{3.3.46}$$

hence,

$$A' = \underline{R}\, A\, \underline{R}^{-1} . \tag{3.3.47}$$

The procedure described above transforms an arbitrary real n by n matrix into tridiagonal form without requiring a knowledge of the eigenvalues. The operations are linear except for the square root extractions in the Householder transformations. The rare exceptional cases are fully accounted for.

IV SCALING

In order to solve a set of linear differential equations on the analog computer, it is necessary that the variables lie within the operating range of the computer. Time, amplitude, and initial condition scaling transform the problem variables to solution variables amenable to the analog computer. The procedure for time and amplitude scaling of the tridiagonal system of equations described in this chapter is unique in that for the first time, an explicit method for scaling is presented which does not rely on the trial and error method used in the past. This new procedure is facilitated by the system of equations being in tridiagonal form.

Let ε_1 and ε_2 $(0 < \varepsilon_1 < \varepsilon_2)$ be the low and high values of the operating range. ε_1 is determined by the tolerable drift of the operational amplifiers while ε_2 is determined by the capabilities of the read-out devices associated with the analog computer. As will be shown in Section 4.2, amplitude scaling does not affect the diagonal entries of T. Furthermore, all terms $\sqrt{|e_i q_i|}$ are unaffected by amplitude scaling; only the individual entries e_i and q_i are affected. Therefore, the time scale factor β is used to transform all

diagonal entries and all $\sqrt{|e_i q_i|}$'s to values that are less than or equal to ε_2 in magnitude, and amplitude scaling places the individual off-diagonal entries of the time scaled matrix (βT) within the operating range.

Finally, initial condition scaling ensures that, for stable systems of equations, the state variables will also lie within the operating range.

4.1 Time Scaling

In terms of the elements t_i, e_j, and q_j of the tri-diagonal matrix T, let the scalars b_1 and b_2 be defined by

$$\begin{aligned} b_1 &= \min(|t_i|, +\sqrt{|e_j q_j|}) \\ b_2 &= \max(|t_i|, +\sqrt{|e_j q_j|}) \end{aligned} \tag{4.1.1}$$

for $i = 1, \cdots, n$ and $j = 2, \cdots, n$. The effect of the time scale factor β is to place b_2 at the high end of the operating range; i.e., β is computed by the formula

$$\beta = \frac{\varepsilon_2}{b_2} \quad . \tag{4.1.2}$$

The resulting inequality then becomes

$$0 \leq \beta b_1 \leq \beta b_2 = \varepsilon_2 \tag{4.1.3}$$

where either $0 < \varepsilon_1 \leq \beta b_1$ or $0 \leq \beta b_1 < \varepsilon_1$ may exist on the low end of the operating range.

In terms of a change of variables, time scaling is applied by letting

$$t = \beta \tau \quad . \tag{4.1.4}$$

Then the system of equations

$$\frac{d}{dt}Y(t) = TY(t) \qquad (4.1.5)$$

becomes

$$\frac{d}{d\tau}Y(\tau) = (\beta T)Y(\tau). \qquad (4.1.6)$$

4.2 Amplitude Scaling

In order to place the off-diagonal entries of the tri-diagonal matrix (βT) within the operating range of the analog computer, amplitude scaling is applied next by using a non-singular diagonal scaling matrix K. The resulting similar amplitude and time scaled matrix T' is related to the time scaled matrix (βT) by

$$T' = K(\beta T)K^{-1}, \qquad (4.2.1)$$

where T' is of the form

$$T' = \begin{bmatrix} y_1 & z_2 & 0 & \dots & 0 \\ x_2 & y_2 & z_3 & \dots & 0 \\ 0 & x_3 & y_3 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & y_n \end{bmatrix} = \begin{bmatrix} \beta t_1 & \frac{k_1}{k_2}\beta q_2 & 0 & \dots & 0 \\ \frac{k_2}{k_1}\beta e_2 & \beta t_2 & \frac{k_2}{k_3}\beta q_3 & \dots & 0 \\ 0 & \frac{k_3}{k_2}\beta e_3 & \beta t_3 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & \beta t_n \end{bmatrix}. \qquad (4.2.2)$$

Note that the product of the opposite off-diagonal terms remains unchanged by amplitude scaling since

$$x_i z_i = \left(\frac{k_i}{k_{i-1}}\beta e_i\right)\left(\frac{k_{i-1}}{k_i}\beta q_i\right) = (\beta e_i)(\beta q_i). \qquad (4.2.3)$$

The k_i's are determined initially in the form of ratios (k_i/k_{i-1}).

Assume temporarily that all e_i's and q_i's are not zero; it is then always possible to make x_i and z_i equal in magnitude where

$$|x_i| = |z_i| = \beta \sqrt{|e_i q_i|} \leq \varepsilon_2 \qquad (4.2.4)$$

Although this is good for analog stability and accuracy, in general it necessitates a potentiometer for each x_i and z_i. A reduction in the number of potentiometers needed can be made if we force either $|x_i|$ or $|z_i|$ to have values of 1, 10, or 100. This is possible, as will now be shown, with the advantage that except when $\beta \sqrt{|e_i q_i|} < 0.1$, $|x_i|$ will differ from $|z_i|$ by less than one order of magnitude.

For each of the (n-1) pairs of entries, let the scalars c_1, c_2, and c_3 be defined by

$$c_1 = \beta \min (|e_i|, |q_i|) = \begin{cases} c_{1e} & \text{if } |e_i| < |q_i| \\ c_{1q} & \text{if } |e_i| > |q_i| \end{cases}$$

$$c_2 = \beta \max (|e_i|, |q_i|) = \begin{cases} c_{2q} & \text{if } |e_i| < |q_i| \\ c_{2e} & \text{if } |e_i| > |q_i| \end{cases} \qquad (4.2.5)$$

$$c_3 = \beta \sqrt{|e_i q_i|} \; .$$

It is now necessary to specify values for ε_1 and ε_2 which depend on the analog computer. For the Applied Dynamics AD-4, let $\varepsilon_1 = 0.1$, $\varepsilon_2 = 100$, and let α_j be defined by

$$\alpha_j = 10^{j-1} \varepsilon_1 \qquad (4.2.6)$$

for $j = 1, \ldots, 4$; i.e., $\alpha_1 = \varepsilon_1$ and $\alpha_4 = \varepsilon_2$.

The following table lists the scale factor ratios in the form (k_i/k_{i-1}) for c_{1e} and c_{1q} (c_{2q} and c_{2e}) depending on the value of c_3.

	k_i/k_{i-1}	
	if $c_1 = c_{1e}$	if $c_1 = c_{1q}$
$c_3 \leq \alpha_1$	$\beta\|q_i\|/\alpha_1$	$\alpha_1/\beta\|e_i\|$
$\alpha_1 < c_3 \leq \sqrt{\alpha_1\alpha_2}$	$\alpha_1/\beta\|e_i\|$	$\beta\|q_i\|/\alpha_1$
$\sqrt{\alpha_1\alpha_2} < c_3 \leq \alpha_2$	$\beta\|q_i\|/\alpha_2$	$\alpha_2/\beta\|e_i\|$
$\alpha_2 < c_3 \leq \sqrt{\alpha_2\alpha_3}$	$\alpha_2/\beta\|e_i\|$	$\beta\|q_i\|/\alpha_2$
$\sqrt{\alpha_2\alpha_3} < c_3 \leq \alpha_3$	$\beta\|q_i\|/\alpha_3$	$\alpha_3/\beta\|e_i\|$
$\alpha_3 < c_3 \leq \sqrt{\alpha_3\alpha_4}$	$\alpha_3/\beta\|e_i\|$	$\beta\|q_i\|/\alpha_3$
$\sqrt{\alpha_3\alpha_4} < c_3 \leq \alpha_4$	$\beta\|q_i\|/\alpha_4$	$\alpha_4/\beta\|e_i\|$

Although the foregoing scheme appears complicated, it is quite easy to program and ensures that (n-1) of the x_i's and z_i's will have magnitudes equal to α_j. Furthermore, each x_i and z_i equal to α_2, α_3, or α_4 in magnitude reduces by one the number of potentiometers needed.

If either e_i or q_i is zero, but not both are zero, then we will have $c_1 = 0$ and c_2 equal to β times the magnitude of the non-zero entry. The ratio (k_i/k_{i-1}) will be defined by

$$\frac{k_i}{k_{i-1}} = \begin{cases} \dfrac{\beta|q_i|}{\alpha_1} & \text{if } |q_i| \neq 0 \\ \dfrac{\alpha_1}{\beta|e_i|} & \text{if } |e_i| \neq 0 \ . \end{cases} \tag{4.2.7}$$

Lastly, if both e_i and q_i are zero, set

$$\frac{k_i}{k_{i-1}} = 1. \tag{4.2.8}$$

Now that we have (n-1) ratios of the form (k_i/k_{i-1}), we solve for the k_i's by letting $k_1 = 1$ and compute the remaining k_i's by

$$k_i = (k_i/k_{i-1})k_{i-1} \tag{4.2.9}$$

for $i = 2,\cdots,n$. Once the k_i's are known, we compute T' by (4.2.2).

Any parameter that is transformed to a value less than 0.1 in magnitude can be replaced by zero due to the limitations in accuracy of the analog computer. The reason for this is that the circuitry for this value would consist of a potentiometer, set for the value, coupled into a gain of one amplifier whose output would be too noisy to be significant or reliable. Consequently, there are some problems that cannot be solved on the analog computer.

Once again, in terms of a change of variables, let $Z(\tau)$ be defined by

$$Z(\tau) = KY(\tau). \tag{4.2.10}$$

The system of equations (4.1.6) then becomes

$$\frac{d}{d\tau}Z(\tau) = K(\beta T)K^{-1}Z(\tau), \tag{4.2.11}$$

or in terms of the original matrix A, we have

$$\frac{d}{d\tau}Z(\tau) = \beta SAS^{-1}Z(\tau) \qquad (4.2.12)$$

where

$$S = KRH. \qquad (4.2.13)$$

4.3 Initial Condition Scaling

Initial condition scaling ensures that the dynamic range of the state variables, as solved on the analog computer, lies within the operating range by imposing a relationship between one volt and one unit of solution. At this point, there appears to be no well defined method for scaling the initial conditions since the behavior of the state variables also depends on the system eigenvalues.

One possible method for scaling the initial conditions is to use an iterative approach which is dependent on the length of the time interval over which the solution is desired. Let the scalar c_4 be defined by

$$c_4 = \max |z_i(0)| \qquad (4.3.1)$$

for $i = 1, \cdots, n$. Starting with an initial value for the scale factor α as

$$\alpha = \frac{\varepsilon_2}{2c_4} \quad , \qquad (4.3.2)$$

attempt to solve the system of equations. If no saturation of the operational amplifiers occurs, then the initial choice is sufficient. If however, saturation does occur decrease α by a small amount and try again.

The actual set of equations to be solved by the analog computer then becomes

$$\frac{d}{d\tau}\,[\alpha Z(\tau)] = \beta\, SAS^{-1}\,[\alpha Z(\tau)], \qquad (4.3.3)$$

or under the change of variables $Z'(\tau) = \alpha Z(\tau)$, we have

$$\frac{d}{d\tau}\,Z'(\tau) = \beta\, SAS^{-1}\,Z'(\tau). \qquad (4.3.4)$$

V PROGRAM AND EXAMPLES

A discussion of the digital computer program is presented in Section 5.1 followed by three examples in Section 5.2. The material concerning the program is not intended to be comprehensive, but rather to show the program structure in general, as related to the tridiagonalization and the scaling. The examples are intended to show numerically how the program implements the theory of this thesis.

5.1 Program Flow

The logical flow of information in the digital computer program is indicated in the following two outlines where each level of indentation represents a decision or action taken as a result of the decision. These outlines show only the major operation involved and are not intended to be comprehensive. The first outline indicates the logic for the tridiagonalization, and the second outline indicates the logic for the scaling.

Furthermore, each similarity transformation is combined with S and S^{-1} when the transformations are generated; i.e., S is replaced by RS, and S^{-1} is replaced by $S^{-1}C$. These combinations are not indicated in the outlines.

Tridiagonalization

At this time it is important to point out that an exhaustive search for the proper initial row vector R_1 is not practical, in time and programming, unless necessary for an extremely rare occurrence of a specialized Case 3. There is at least one other method, although not foolproof, which has been demonstrated to be worthwhile for dealing with the Case 3 breakdown. The procedure for implementing this alternate method will now be discussed; this alternate method was used in the program that computed the examples in Section 5.2.

Let j be the stage of factorization in which the most recent Case 2 occurred. After exhausting the limited choices for R_j in attempting to progress past the stage (k+1) in which the Case 3 breakdown occurred, the following three transformations are performed:

$$W = R_a A R_a^{-1}, \tag{5.1.1}$$

$$W' = PWP^{-1}, \tag{5.1.2}$$

and

$$A' = H'W'(H')^*. \tag{5.1.3}$$

The matrix R_a is formed from the first (j-1) rows of R and the last (n-j+1) rows of U. It can be shown that W has the form

$$W = R_a \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} R_a^{-1} = \begin{bmatrix} T_{11} & 0 \\ A_{21} & W_{22} \end{bmatrix}, \tag{5.1.4}$$

where T_{11} is tridiagonal, and A_{22} is zero except for the entry e_j in the upper right corner.

Let ℓ be the column in $F_{R,k+1}$ of (3.3.33) in which the first non-zero entry occurred with the most recent R_j (now denoted $\underline{R}_j$). The matrix P is the unit matrix except for the entries $p_{\ell i}$, $i = k+1, \cdots, \ell-1$; these entries are -1's. The matrix W' then has the form:

$$W' = \begin{bmatrix} U & 0 \\ 0 & \underline{P} \end{bmatrix} WP^{-1} = \begin{bmatrix} T_{11} & 0 \\ A_{21} & W'_{22} \end{bmatrix} . \qquad (5.1.5)$$

The effect of this transformation is to attempt to make $f_{k+1} \neq 0$ when the Krylov factorization, using $\underline{R}_j$ to start with, again reaches the $(k+1)^{st}$ stage.

Since W'_{22} in (5.1.5) is not necessarily in upper Hessenberg form, then the Householder transformation H' in (5.1.3) is applied so that A'_{22} in (5.1.6) is in upper Hessenberg form.

$$A' = \begin{bmatrix} U & 0 \\ 0 & \underline{H}' \end{bmatrix} W'(H')^* = \begin{bmatrix} T_{11} & 0 \\ A_{21} & A'_{22} \end{bmatrix} \qquad (5.1.6)$$

The Krylov factorization is again attempted treating A'_{22} in (5.1.6) as A and $\underline{R}_j$ as R_1. The resulting R matrix is now in the form

$$R = \begin{bmatrix} U & 0 \\ 0 & \underline{R} \end{bmatrix} \qquad (5.1.7)$$

where the first row of $\underline{R}$ is $\underline{R}_j$.

For a given n and i, let $R_i' = [r_i \cdots r_n]$ be a row vector of length k = n-i+1 whose entries $r_i = 1$ and $r_j = 0$ or 1 for $i < j \leq n$. R_i' can be considered to be the binary representation of ℓ_i where $2^{k-1} \leq \ell_i \leq 2^k - 1$.

Example: If n = 5 and i = 2, the possible values of ℓ_i lie in the range $2^3 = 8 \leq \ell_i \leq 2^4 - 1 = 15$.

The digital program for the tridiagonalization uses the above values for R_i, when R_i is the result of a Case 2, starting at the low end of the range and ranging through all binary numbers to the high end of the range before it performs the partial transformation at a subsequent Case 3 stage. It is conceivable that one of these possible R_i's might not cause a Case 3 to occur, although a previous R_i did.

In the outline for the tridiagonalization, the following symbols (combinations of letters; i.e., THS is a matrix name) are used to denote matrices: A, H, H*, B, THS, R, W, C, F, RA, CA, WT, FT, and Z. With the exception of B, THS, W, WT, FT, and Z, the remaining symbols parallel the matrices referred to earlier in the thesis. B, THS, WT, FT, and Z are used for temporary matrix storage, and W is used to store the successive idempotents as they are computed. Since A is assumed to be real, H* is actually H^T. Furthermore, R_I is row I of R, F_{RI} is row I of the factored Krylov matrix F (see equation (3.3.33) with k+1 replaced with I), C_I is column I of C, and W is the idempotent.

As scalars, the symbols N, I, IT, and JT are used where N is the order of the matrix, I is a stage of factorization in which R_{I+1} is computed, IT + 1 is the most recent Case 2 row in the factorization, and JT (normally equal to zero) is set to a one when there is a Case 3 at a certain stage with all the limited choices for the prior Case 2 R_{IT+1}'s being depleted.

Outline

Given the matrix A and order N

Perform Householder transformation (HAH* = B)

Set THS = B (save Hessenberg matrix B)

(a) Compute R_1 from binary vectors

Compute C_1 and W_2

Compute Krylov matrix F

Set I = 0, IT = 0, JT = 0 (I is the current stage of factorization, IT + 1 is the most recent Case 2 stage, JT is changed to one when a Case 3 has occured with all possible choices of R_{IT+1} depleted).

Go to (c)

(b) Does I = N-1?

Yes

Go to (h)

No

(c) Set I = I+1

Factor Krylov matrix in place

Test cases

Case 1

(d) Set $R_{I+1} = F_{R,I+1}/d_{I+1}$

Compute C_{I+1} and W_{I+2}

Go to (b)

Case 2

Set RA = R, CA = C, WT = W, and FT = F (save present R, C, W, and F)

Set IT = I and JT = 0

(e) Compute R_{I+1} from binary vectors

Go to (d)

Case 3

Set I = IT

Is JT = 0?

Yes

Is IT = 0?

Yes

Are all binary vectors for R_1 used?

Yes

Set JT = 1

Go to (f)

No

Go to (a)

No

Are all binary vectors for R_{IT+1} used?

Yes

Set JT = 1

Compute (RA)(THS)(CA) = B

Go to (g)

No

Set R = RA, C = CA, W = WT, and F = FT

Go to (e)

No

(f) Set B = THS (restore upper Hessenberg or Partial tridiagonal matrix)

(g) Compute $PBP^{-1} = Z$

Perform Householder transformation $HZH^* = B$

Set THS = B (save partial tridiagonal matrix)

Compute Krylov matrix using most recent R_{IT+1}

Go to (d)

(h) Compute $R_b(THS)C_b = T$

Scaling

In the outline for scaling, T is used to denote the tridiagonal matrix. E, Q, C, and RK are used to denote vectors where E is the subdiagonal, Q is the superdiagonal, C is the geometric mean of the sub and super diagonals, and RK is initially the ratio of the amplitude scale factors and lastly the vector of amplitude scale factors. The scalars E1, E2, B2, BETA, and N correspond to ε_1, ε_2, b_2, β, and n in Chapter IV.

Outine

Given T, N, E1, E2

Set $E_I = |t_{I,I-1}|$, $Q_I = |t_{I-1,I}|$, and $C_I = \sqrt{E_I Q_I}$

for $I = 2, \cdots, N$

Find B2 = max ($|t_{I,I}|$, C_I)

Compute BETA = E2/B2 (time scale factor)

Compute (BETA)T = T (time scaled)

Compute $(BETA)E_I = E_I$, $(BETA)Q_I = Q_I$, and $(BETA)C_I = C_I$ for I = 2,···,N

Compute $(RK)_I = k_i/k_{i-1}$ for i = I = 2,···,N

Given $k_1 = 1$, compute k_i in $(RK)_I$ for i = I = 2,···,N

Compute $(RK)(T)(RK)^{-1}$ = T (time and amplitude scaled)

5.2 Examples

Three examples are given in this section to illustrate the tridiagonalization and the time and amplitude scaling procedures that were presented in Chapters III and IV.

Example 5.2.1 illustrates the Householder transformation which transforms an aribitrary real non-symmetric matrix A to upper Hessenberg form A_H where,

$$A_H = HAH^T. \tag{5.2.1}$$

Examples 5.2.2 and 5.2.3 illustrate the tridiagonalization and the scaling of an upper Hessenberg matrix A_H where

$$T' = \beta S A_H S^{-1}. \tag{5.2.2}$$

Example 5.2.2 illustrates Case 2 operations, and Example 5.2.3 illustrates Case 3 operations.

The IBM 1800 computed these examples using extended precision which is thirty-one bits of mantissa or roughly nine decimal places; however, printed answers are rounded to two or three decimal places. In these examples, 0_{-j} is

used to denote a number whose characteristic is 10^{-j} $(0 < j)$; i.e., small order of magnitude. Zeros are shown where an exact zero was produced by the computer. Also, the symbol (—>) is used meaning "is factored into". The Krylov matrix F is factored in place, where F = LDV is stored as (L - U) + D + (V - U).

$$F = LDV = \begin{bmatrix} & & & V-U \\ & & D & \\ & \iddots & & \\ L-U & & & \end{bmatrix} \qquad (5.2.3)$$

Example 5.2.1: This example illustrates the Householder transformation of a real non-symmetric matrix A to upper Hessenberg form A_H.

$$A = \begin{bmatrix} 4 & -1 & -1 & 3 \\ 1 & 3 & -2 & 2 \\ 2 & -2 & 2 & 5 \\ 2 & -1 & -4 & 0 \end{bmatrix}$$

$A_H = HAH^T$ (Householder transformation)

$$H = \begin{bmatrix} 1.000 & 0 & 0 & 0 \\ 0 & -0.333 & -0.667 & -0.667 \\ 0 & -0.133 & -0.667 & 0.733 \\ 0 & -0.933 & 0.333 & 0.133 \end{bmatrix},$$

$$A_H = \begin{bmatrix} 4.0 & -1.0 & 3.0 & 1.0 \\ -3.0 & 1.0 & -4.4 & -0.8 \\ 0 & 5.0 & 0 & -2.0 \\ 0 & 0_{-8} & -1.0 & 4.0 \end{bmatrix}$$

Example 5.2.2: This example illustrates the tridiagonalization procedure with a Case 1 followed by two Case 2 factorizations of the Krylov matrix F. It also illustrates the results of the time and amplitude scaling procedure.

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

(Krylov factorization)

$$F = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 3 & 4 \\ 1 & 5 & 12 & 16 \\ 1 & 14 & 39 & 52 \end{bmatrix} \xrightarrow{1} \left[\begin{array}{c|ccc} 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 3 & 4 \\ 1 & 4 & 12 & 16 \\ 1 & 13 & 39 & 52 \end{array}\right] \xrightarrow{2} \left[\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 1 & 1 & 3 & 4 \\ \hline 1 & 4 & 0 & 0 \\ 1 & 13 & 0 & 0 \end{array}\right],$$

$$\left[\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 1 & 1 & 3 & 4 \\ \hline 1 & 4 & 1 & 0 \\ 1 & 13 & 0 & 0 \end{array}\right] \xrightarrow{2} \left[\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 1 & 1 & 3 & 4 \\ 1 & 4 & 1 & 0 \\ \hline 1 & 13 & 0 & 0 \end{array}\right], \quad \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 3 & 4 \\ 1 & 4 & 1 & 0 \\ 1 & 13 & 0 & 1 \end{bmatrix}$$

$T = RA_HC$ (Tridiagonalization)

$$R = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 3 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

$\beta = 33.33$

$T' = K(\beta T)K^{-1}$ (Scaling)

$$K = \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 333.3 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1333.3 \end{bmatrix}, \quad T' = \begin{bmatrix} 33.3 & 0.1 & 0 & 0 \\ 0 & 100.0 & 0 & 0 \\ 0 & 0.1 & 0 & -0.1 \\ 0 & 0 & 0 & 100.0 \end{bmatrix}$$

$S = KR$ (Transforming matrix)

$$S = \begin{bmatrix} 1.0 & 1.0 & 0 & 0 \\ 0 & 333.3 & 10^4 & 1333.3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1333.3 \end{bmatrix}$$

Example 5.5.3: This example illustrates the tridiagonalization procedure with a Case 1 followed by a Case 3 factorization of the Krylov matrix F. Instead of changing the initial row vector R_1, this vector is kept so as to illustrate the use of the P matrix in (5.1.5) followed by a Householder transformation. A new Krylov matrix F' is computed using R_1 and is factored with three Case 1's occurring.

$$A_H = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

(Krylov factorization)

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 3 & 3 & 3 & 1 \\ 9 & 9 & 10 & 4 \end{bmatrix} \overrightarrow{1} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 3 & 3 & 3 & 1 \\ 9 & 9 & 10 & 4 \end{bmatrix} \overrightarrow{3} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 3 & 3 & 0 & 1 \\ 9 & 9 & 1 & 4 \end{bmatrix},$$

$$W' = PA_H P^{-1}$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix}, \quad W' = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 3 & 2 & 2 & 1 \\ 0 & 1 & 1 & 0 \\ -3 & -3 & -2 & 0 \end{bmatrix}$$

$A' = H'W'(H')^T$ (Householder transformation)

$$H' = \begin{bmatrix} 1.00 & 0 & 0 & 0 \\ 0 & -0.71 & 0 & 0.71 \\ 0 & 0.58 & -0.58 & 0.58 \\ 0 & 0.41 & 0.82 & 0.41 \end{bmatrix}, \quad A' = \begin{bmatrix} 1.00 & -0.71 & 0_{-9} & 1.22 \\ -4.24 & 2.00 & -0.82 & -4.04 \\ 0 & 1.22 & 0_{-9} & -0.71 \\ 0 & 0_{-9} & 0 & 1.00 \end{bmatrix}$$

(Krylov factorization)

$$F = \begin{bmatrix} 1.00 & 0 & 0 & 0 \\ 1.00 & -0.71 & 0_{-9} & 1.22 \\ 4.00 & -2.12 & 0.58 & 5.31 \\ 13.00 & -6.36 & 1.73 & 18.37 \end{bmatrix} \overrightarrow{1} \begin{bmatrix} 1.00 & 0 & 0 & 0 \\ 1.00 & -0.71 & 0_{-9} & -1.22 \\ 4.00 & -2.12 & 0.58 & 5.31 \\ 13.00 & -6.36 & 1.73 & 18.37 \end{bmatrix} \overrightarrow{1}$$

$$\begin{bmatrix} 1.00 & 0 & 0 & 0 \\ 1.00 & 1.00 & 0_{-9} & -1.73 \\ 4.00 & 3.00 & 0.58 & 1.63 \\ 13.00 & 9.00 & 1.73 & 7.35 \end{bmatrix} \xrightarrow[1]{} \begin{bmatrix} 1.00 & 0 & 0 & 0 \\ 1.00 & 1.00 & 0_{-9} & -1.73 \\ 4.00 & 3.00 & 1.00 & 2.83 \\ 13.00 & 9.00 & 3.00 & 2.45 \end{bmatrix}$$

$T = R_b A' R_b^{-1} = R_b A' C_b$ (Tridiagonalization)

$$R_b = \begin{bmatrix} 1.00 & 0 & 0 & 0 \\ 0 & 1.00 & 0_{-9} & -1.73 \\ 0 & 0 & 1.00 & 2.83 \\ 0 & 0 & 0 & 1.00 \end{bmatrix}, \quad T = \begin{bmatrix} 1.00 & -0.71 & 0_{-18} & 0 \\ -4.24 & 2.00 & 0.82 & 0_{-18} \\ 0 & 1.22 & 0_{-9} & 4.24 \\ 0 & 0_{-9} & 0_{-18} & 1.00 \end{bmatrix}$$

$\beta = 50.0$

$T' = K(\beta T)K^{-1}$ (Scaling)

$$K = \begin{bmatrix} 1.00 & 0 & 0 & 0 \\ 0 & 0.47 & 0 & 0 \\ 0 & 0 & 0.77 & 0 \\ 0 & 0 & 0 & 1630.00 \end{bmatrix}, \quad T' = \begin{bmatrix} 50.0 & 75.0 & 0_{-16} & 0 \\ -100.0 & 100.0 & 25.0 & 0_{-10} \\ 0 & 100.0 & 0_{-8} & 0.1 \\ 0 & 0_{-4} & 0_{-13} & 50.0 \end{bmatrix}$$

$S = KR_b H' P$ (Transforming matrix)

$$S = \begin{bmatrix} 1.00 & 0 & 0 & 0 \\ 0_{-9} & -0.67 & -0.67 & 0_{-9} \\ 1.33 & 0 & 0_{-9} & 1.33 \\ -666.67 & 0 & 666.67 & 666.67 \end{bmatrix}$$

These three examples have been presented in an attempt to illustrate the tridiagonalization and the time and amplitude scaling procedures of Chapters III and IV.

VI CONCLUSIONS

In this chapter, a summary of the original results is presented followed by an indication of some additional areas of study which have presented themselves through the development of this thesis.

In many engineering problems, the need to solve a system of simultaneous homogeneous linear differential equations with constant coefficients arises. This thesis showed how this system of equations,

$$\frac{d}{dt}X(t) = AX(t),$$

in which $X(t)$ is an n-dimensional vector function of time and A is an $(n \times n)$ real square matrix, can be solved on the hybrid computer using no more than $2n$ operational amplifiers (n bipolar integrators) nor more than $(2n-1)$ potentiometers, as opposed to the more conventional means of solution using n^2 potentiometers, on the analog computer.

By tridiagonalizing the matrix A, the former n^2 possible number of non-zero parameters is reduced to at most $(3n-2)$. Since one potentiometer is usually needed for each non-zero entry, by appropriately scaling the tridiagonal matrix, the number of potentiometers needed is further reduced to $(2n-1)$.

In case automatic patching is desired, usually accomplished using one reed relay for each non-zero entry in the coefficient matrix, a considerable savings in the number of relays needed is affected by tridiagonalizing the coefficient matrix.

A new procedure for tridiagonalizing an arbitrary real non-symmetric matrix A, without using the eigenvalues, was presented in Chapter III. After transforming the matrix A by unitary Householder transformations into a similar upper Hessenberg matrix A_H, the latter is transformed into a tridiagonal matrix $T = RA_HC$ by appropriate matrices R (with rows R_i) and $C = R^{-1}$ (with columns C_i). The rows R_i are computed recursively by factoring the Krylov matrix F whose rows are the interates of R_1 under A_H. The columns C_i are computed recursively from an idempotent matrix $W_i = U - \sum_{j=1}^{i-1} C_jR_j$ by $C_i = W_iU_i$ where U_i is the i^{th} column of the unit matrix.

In Section 3.3, Theorem 1 was proved which showed that the tridiagonalization is always possible without breaking down when A_H has all non-zero subdiagonal entries. If A_H has at least one zero subdiagonal entry, the hypothesis of the theorem is invalidated in which case the procedure may or may not break down. When the procedure does break down, it does so under either Case 2 or Case 3. Case 2 is handled by obtaining R_i from the idempotent W_{i-1}. For Case 3, Theorem 2, proved in Section 3.3, shows how to advance at least one additional stage.

A lemma was also proved in Section 3.3; this lemma showed a new method for inverting a unit upper triangular matrix.

For the first time, an explicit procedure for time and amplitude scaling was presented. This procedure, presented in Chapter IV, is facilitated by the system of equations being in tridiagonal form, and ensures that no more than (2n-1) potentiometers are needed on the analog computer.

This thesis will be concluded by indicating some areas in which additional investigation could be done. It would be useful to extend this solution procedure to encompass other forms of differential equations such as those that are non-linear. Also, the scaling of the initial conditions could perhaps be accomplished by means other than the method described in Section 4.3. For instance, the feasibility of using time-varying initial conditions over the desired interval of solution, in the form of a series of steps or as a continuous function, could be investigated.

REFERENCES

1. Causey, R. L., and Gregory, R. T. "On Lanczos' Algorithm for Tridiagonalizing Matrices," Society of Industrial and Applied Mathematics Review, III, No. 4 (October, 1961), 322-328.

2. Faddeyev, D. K., and Faddeyeva, V. N. Computational Methods of Linear Algebra. English translation. San Francisco: W.H. Freeman and Company, 1963, 241-247.

3. Frame, J. S. Matrix Theory and Linear Algebra. Unpublished book.

4. Kublanovskaya, V. N. "Reduction of an Arbitrary Matrix to the Tridiagonal Form," U.S.S.R. Computational Mathematics and Mathematical Physics, IV, No. 3(1964), 202-203.

5. LaBudde, C. D. "The Reduction of an Arbitrary Real Square Matrix to Tri-Diagonal Form Using Similarity Transformations," Mathematics of Computation, VII (1963), 433-437.

6. Parlett, Beresford. "A Note on LaBudde's Algorithm," Mathematics of Computation, XVIII (1964), 505-506.

7. Strachey, C., and Francis, J. G. F. "The Reduction of a Matrix to Codiagonal Form by Elimination," Computer Journal, IV(1961-1962), 168-176.

8. Waltman, William L., and Lambert, Robert J. "T-Algorithm for Tridiagonalization," Journal of Society of Industrial and Applied Mathematics, XIII, No. 4 (December, 1965), 1069-1078.

9. Wang, H. H., and Gregory, R. T. "On the Reduction of an Arbitrary Real Square Matrix to Tridiagonal Form," Mathematics of Computations, XVIII(1964), 501-505.

10. Wilkinson, J. H. "The Calculation of Eigenvectors by the Method of Lanczos," Computer Journal, I (1958), 148-152.

11. Wilkinson, J. H. "Instability of the Elimination Method of Reducing a Matrix to Tridiagonal Form," Computer Journal, V (1962-1963), 61-70.