Adaption and Optimization of the Ibrahim Time Domain (ITD) Modal Parameter Identification Algorithm for Mini-Computer Use

1984

Mary T. Cogeos
University of Central Florida

Find similar works at: https://stars.library.ucf.edu/rtd

University of Central Florida Libraries http://library.ucf.edu

Part of the Engineering Commons

STARS Citation

https://stars.library.ucf.edu/rtd/4700

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
ADAPTATION AND OPTIMIZATION OF THE IBRAHIM TIME DOMAIN (ITD) MODAL PARAMETER IDENTIFICATION ALGORITHM FOR MINI-COMPUTER USE

BY

MARY THERESE COGEOS
B.S.E., University of Central Florida, 1982

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Graduate Studies Program of the College of Engineering University of Central Florida Orlando, Florida

Spring Term 1984
ABSTRACT

Conventional modal analysis is performed using FFT Analyzers with modal analysis software developed by one of many test/analysis software vendors. The analysis is performed in the frequency domain using frequency response functions. A recent advancement in modal analysis involves analyzing the measured free response of a structure in the time domain. The Ibrahim Time Domain (ITD) Algorithm is such a method which has been used and verified at government environmental test facilities. The ITD is an accurate numeric solution. Minimal operator interaction is required for proper performance of the algorithm. The results of the ITD Algorithm are far superior to those of conventional frequency response methods when the test structure has closely spaced or heavily damped modes of vibration. Currently, the ITD exists only on mainframe computers. The purpose of this research is to adapt the numeric routine to a mini-computer (DEC PDP-11/34). In this way, the ITD Algorithm could be incorporated in any environmental test facility with access to a mini-computer.
to Paul with my love
I would first like to thank my husband, Paul, for his love and his constant encouragement throughout this research.

I also thank my professors at the University of Central Florida, especially Dr. David Jenkins and Dr. Sayed Metwalli, two wonderful graduate advisors. To Sharon Darling, thank you for your patience and perfection in typing this paper.

Finally, I thank my colleagues at Martin Marietta Orlando Aerospace for their assistance, guidance and support. In particular, I extend my sincerest gratitude to Alex Pettit, Carl Sutphin, Roger Graves, Christine Blackshear and Michael Terebo. I ask that you all continue to support me as I strive to make the ITD Algorithm an integral part of future modal analysis performed in the Structural and Environmental Test Laboratory.
# TABLE OF CONTENTS

LIST OF ABBREVIATIONS. .................................................. vi
LIST OF SYMBOLS. ......................................................... vii
LIST OF FIGURES. .......................................................... x
LIST OF TABLES ............................................................. xi

Chapter
I. INTRODUCTION ............................................................ 1
II. THEORY OF THE ITD. ..................................................... 7
III. USER-SELECTABLE ALGORITHM CONSTANTS. ......................... 19
IV. DETERMINING THE "SYSTEM MATRIX", A .............................. 24
    Filling Response Matrices $\Phi$ and $\Phi^T$ ......................... 24
    Calculating $(\Phi\Phi^T)$ and $(\Phi^T\Phi)$. ......................... 31
    Choleski Decomposition of $X$. ...................................... 33
    Forward and Back Substitution. .................................... 36
V. DETERMINING THE EIGENVALUES OF THE "SYSTEM MATRIX", A. 38
VI. DETERMINING THE MODAL PARAMETERS ............................... 45
VII. RESULTS AND FUTURE RESEARCH. .................................... 52

Appendices
A. CHOLESKI DECOMPOSITION ............................................ 63
B. GAUSSIAN ELIMINATION FORWARD AND BACK SUBSTITUTION .......... 66
    Forward Substitution .............................................. 66
    Back Substitution ............................................... 69
C. PROOF THAT ELEMENTS ON THE DIAGONAL OF AN UPPER
    TRIANGULAR MATRIX ARE THE EIGENVALUES. ...................... 74
D. REDUCTION TO UPPER HESSENBURG. ................................... 76
TABLE OF CONTENTS (Continued)

E. QR ALGORITHM .................................................. 84
F. DETERMINING EIGENVECTORS THROUGH BACK SUBSTITUTION . . 107
   Case 1: All Real Eigenvalues and Eigenvectors . . . . . . . . 107
   Case 2: Both Real and Complex Eigenvalues and
           Eigenvectors ........................................... 110
   Case 3: All Complex Eigenvalues and Eigenvectors . . . . . 114

REFERENCES ...................................................... 116
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FRF</td>
<td>Frequency Response Function</td>
</tr>
<tr>
<td>IMAC</td>
<td>International Modal Analysis Conference</td>
</tr>
<tr>
<td>ITD</td>
<td>Ibrahim Time Domain</td>
</tr>
<tr>
<td>MCF</td>
<td>Modal Confidence Factor</td>
</tr>
<tr>
<td>MDOF</td>
<td>Multi Degree of Freedom</td>
</tr>
<tr>
<td>OAMCF</td>
<td>Overall Modal Confidence Factor</td>
</tr>
<tr>
<td>SDOF</td>
<td>Single Degree of Freedom</td>
</tr>
<tr>
<td>SF</td>
<td>Sample Frequency</td>
</tr>
<tr>
<td>SIFRF</td>
<td>Single Input Frequency Response Function</td>
</tr>
<tr>
<td>Matrices</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>A</td>
<td>Type of &quot;System Matrix&quot;</td>
</tr>
<tr>
<td>D</td>
<td>Diagonal matrix used in Choleski decomposition</td>
</tr>
<tr>
<td>H</td>
<td>Upper Hessenburg matrix</td>
</tr>
<tr>
<td>I</td>
<td>Permutation matrix used in the Hessenburg reduction</td>
</tr>
<tr>
<td>L</td>
<td>Lower triangular matrix with a unit diagonal used in the Choleski decomposition</td>
</tr>
<tr>
<td>~L</td>
<td>Lower triangular matrix which is the product of ( L \cdot D^2 ) used in the Choleski decomposition</td>
</tr>
<tr>
<td>N</td>
<td>Similarity transformation matrix</td>
</tr>
<tr>
<td>P_r</td>
<td>Elementary Hermitian matrix used in the QR Algorithm</td>
</tr>
<tr>
<td>T</td>
<td>Pseudo upper triangular matrix which may contain 2 x 2 sub-matrices on the diagonal</td>
</tr>
<tr>
<td>U</td>
<td>Upper triangular matrix used in the LU decomposition</td>
</tr>
<tr>
<td>X</td>
<td>Square response matrix which is the product of ( \Phi ) and ( \Phi^T )</td>
</tr>
<tr>
<td>X~</td>
<td>Square response matrix which is the product of ( \hat{\Phi} ) and ( \hat{\Phi}^T )</td>
</tr>
<tr>
<td>Y</td>
<td>Intermediate matrix used in the Gaussian elimination forward and back substitution</td>
</tr>
<tr>
<td>[\gamma]</td>
<td>Diagonal matrix of the eigenvalues ( e^{\lambda_k \Delta t} )</td>
</tr>
<tr>
<td>[\lambda]</td>
<td>Diagonal matrix of the eigenvalues ( e^{\lambda_k \Delta t^1} )</td>
</tr>
<tr>
<td>\Lambda</td>
<td>Non-diagonal (fully populated) matrix of complex exponentials</td>
</tr>
</tbody>
</table>
LIST OF SYMBOLS (Continued)

Matrices (continued)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Psi$</td>
<td>Eigenvector matrix of $A$</td>
</tr>
<tr>
<td>$\hat{\Psi}$</td>
<td>Eigenvector matrix of $H$ or $T$</td>
</tr>
<tr>
<td>$\hat{\Psi}$</td>
<td>Product of the eigenvector matrix $\Psi$ and the eigenvalue matrix $[\gamma]$</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>Response matrix</td>
</tr>
<tr>
<td>$\hat{\Phi}$</td>
<td>Response matrix delayed some $\Delta t_1$</td>
</tr>
</tbody>
</table>

Vectors

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>${p + iq}$</td>
<td>Complex eigenvector notation</td>
</tr>
<tr>
<td>${n(t)}$</td>
<td>Vector used to model the noise portion of the response measurements</td>
</tr>
<tr>
<td>${x(t)}$</td>
<td>Response vector</td>
</tr>
<tr>
<td>${\dot{x}(t)}$</td>
<td>Response vector delayed some time $\Delta t_1$</td>
</tr>
<tr>
<td>${\psi}$</td>
<td>Eigenvector of $A$</td>
</tr>
<tr>
<td>${\bar{\psi}}$</td>
<td>Eigenvector of $H$ or $T$</td>
</tr>
<tr>
<td>${\hat{\psi}}$</td>
<td>Eigenvector of $A$ multiplied by the scalar $\gamma_k$</td>
</tr>
</tbody>
</table>

Scalars and Constants

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(a + ib)$</td>
<td>Complex eigenvalue notation</td>
</tr>
<tr>
<td>$(\alpha + i\beta)$</td>
<td>Complex root of the characteristic equation</td>
</tr>
<tr>
<td>$e^{\lambda_k \Delta t}$</td>
<td>Complex exponential of $e$</td>
</tr>
<tr>
<td>$e^{\lambda_k \Delta t_1}$</td>
<td>Eigenvalue of the &quot;System Matrix&quot;, $A$</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>May be used to denote the root of the characteristic equation (Chapter II: $\lambda = \alpha + i\beta$), or the eigenvalue of $A$ (Chapters V and VI: $\lambda = a + ib$)</td>
</tr>
</tbody>
</table>
### Scalars and Constants (Continued)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi_{ij}$</td>
<td>Eigenvector element delayed some time $\Delta t_3$ from the element $\psi_{ij}$</td>
</tr>
<tr>
<td>$\zeta$</td>
<td>Damping factor (percent of critical damping)</td>
</tr>
<tr>
<td>$\omega_d$</td>
<td>Damped natural frequency</td>
</tr>
<tr>
<td>$\omega_n$</td>
<td>Natural frequency</td>
</tr>
<tr>
<td>$m$</td>
<td>Number of computational modes used in the mathematical model</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of structural modes</td>
</tr>
<tr>
<td>$p$</td>
<td>Number of actual measurements</td>
</tr>
<tr>
<td>$s$</td>
<td>Number of time samples used in the response matrices, $\phi$ and $\tilde{\phi}$</td>
</tr>
<tr>
<td>$N_i$</td>
<td>User selectable parameters ($i = 1, 2, \text{ or } 3$) representing the number of consecutive time samples used in each $\Delta t_i$</td>
</tr>
<tr>
<td>$\Delta t_i$</td>
<td>Time shifts ($i = 1, 2, \text{ or } 3$) used in filling the response matrices $\phi$ and $\tilde{\phi}$</td>
</tr>
<tr>
<td>NCOL</td>
<td>User selectable parameter denoting the number of columns ($s$) in the response matrices $\phi$ and $\tilde{\phi}$</td>
</tr>
<tr>
<td>NDOF</td>
<td>User selectable parameter denoting half the number of rows ($m$) in the response matrices $\phi$ and $\tilde{\phi}$</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1. Incorporating $\Delta t_1$, $\Delta t_2$ and $\Delta t_3$ to fill response matrices  27
2. Transpose of the response matrix, $\Phi$  30
D-1. Reduction to upper Hessenburg  79
D-2. Hessenburg reduction of 4th order $A$  83
# LIST OF TABLES

1. Results of the MINI-ITD for Case I. .......................... 54
2. Results of the MINI-ITD for Case II ......................... 55
3. Results of the MINI-ITD for Case III ....................... 56
4. Modal Parameter Identification for a 2 DOF System with Varying Degrees of Noise. .......................... 59
Modal analysis can be defined as a technique for representing the dynamic response of an elastic structure in terms of its modes of vibration. Mathematically speaking, the modes of vibration are the solutions of the eigenvalue problem derived from the differential equations of motion of the structure. Experimentally speaking, they are the parameters of the linear dynamic model of the structure. The parameters are the modal frequency (natural frequency), modal damping (factor of percent critical damping), and mode shape (Richardson 1984). The natural frequency and damping are contained in the pole or complex eigenvalue. The mode shape is referred to as the eigenvector.

A brief summary of present testing methods is given in Dr. David Brown's Keynote Address at the First International Modal Analysis Conference (IMAC). In this summary, he states that greater than ninety-five percent of conventional modal analysis testing is performed using Single-Input Frequency Response Function (SIFRF) (Brown 1982).

In the SIFRF method, and several others listed by Dr. Brown, modal analysis is performed using FFT Analyzers with modal analysis software developed by one of many test/analysis software vendors. All analysis is performed in the frequency domain. The Fourier transform of a response measurement is divided by the Fourier transform of
the driving signal to produce a Frequency Response Function (FRF). The FRF contains the information necessary to describe the dynamic response of location B due to an excitation force at point A. An entire set of FRF's is typically acquired with a single excitation location and many response locations. An alternate method is one response location and roving excitation (hammer impact test). Performing the test in either manner, with a single driving force input and varying response locations or with a single response and varying driving force locations, defines a single row or column of the FRF matrix. This is sufficient to determine all the modal parameters.

The modal parameters are estimated using one of many curve fitting routines. The most frequently used are listed below (Richardson 1984):

1. Quadrature (Peak) Picking
2. Single Degree of Freedom (SDOF) Circle Fitting
3. Complex Division
4. Complex Exponential
5. Rational Fraction Polynomial
6. Poly-Reference

The curve fits are approximations and normally require the assistance of a skilled engineer for proper performance. Although the solution is accurate when the modes of the structure are lightly damped and well spaced, significant errors may result when this is not the case. Complex symmetric structures often have closely spaced body bending
and torsional modes and the above curve fitting routines generally fail when trying to estimate these modal parameters.

SDOF curve fitting routines assume that the overall response near a resonance is dominated by a single, lightly damped mode. However, when modal density is high, modal interference greatly affects the parameter estimation. MDOF (multi-degree of freedom) curve fitting routines are designed to estimate modal parameters of an entire FRF. They give consistent and accurate results on data that is accurately analyzed using SDOF curve fits. However, when the system has heavily damped and/or closely spaced modes, the results of each MDOF parameter estimations may differ significantly from one another.

A recent advancement in modal analysis involves analyzing the test data in the time domain. The structure may or may not be driven. The Ibrahim Time Domain Algorithm employs the simpler undriven case. The ITD Algorithm determines the modal parameters from measured free response functions analyzed in the time domain. Minimal operator interaction is required for the proper performance of the algorithm.

In contrast to the above parameter estimation techniques, the ITD Algorithm is an exact numeric solution. No assumptions are made about modal spacing or damping.

In the early developmental stages of the ITD, the acceleration, velocity and displacement were required to formulate the mathematical model. Assuming the acceleration was measured, numerical integration was required to obtain the velocity and displacement time histories (Ibrahim and Mikulcik 1973 and 1976).
A more favorable approach was published in 1977 in which only one of the three response time histories was required. The mathematical model could be formulated using acceleration, velocity or displacement free response functions. It is this approach that is called the Ibrahim Time Domain Algorithm (Ibrahim and Mikulcik 1977).

Any of the three response forms may be obtained from:

1. free decays measured after random excitation to a structure is cut off
2. unit impulse response functions formed by inverse Fourier transformation of frequency response functions (Ibrahim and Mikulcik 1977)

Because free decay measurements may be used, the need to measure and process input forces is eliminated. In the cases of multi-shaker and acoustic excitation, this is a significant advantage.

The current ITD "identification process is a 'blind' technique requiring a minimal amount of operator input to compute parameters for many modes from a set of free response functions" (Ibrahim 1981b). Often, twenty or more structural modes have been identified in a single computer run.

The accuracy of this technique was demonstrated using computer simulated data with known results (Ibrahim 1981b). A range of modal parameters, user-selectable algorithm constants and noise levels were tested. The ITD gave consistently accurate results when using significantly oversized math models. The system is modeled with the number of computational degrees of freedom many times the number of structural modes desired.
A large modal survey (142 measurements) was performed on the Long Duration Exposure Facility (LDEF) Space Shuttle Payload (Ibrahim and Pappa 1981a). Excellent results again were obtained using the oversized math model.

Additional experimental and computer simulated tests and the ITD results may be found in additional references (Carne, Martinez and Ibrahim 1983; Pappa 1983; Horta and Hanks 1983; Ibrahim 1983).

The ITD has been established as a feasible alternative, or supplement, to other modal analysis techniques. Test and analysis facilities would greatly benefit if this technique could be routinely employed when modal coupling or damping prohibit conventional FRF modal analysis. The ITD currently exists on mainframe computers which contain standard numerical libraries to perform eigenanalysis. This poses a problem since few test facilities have access to such large computational ability. Access to mini-computers, however, is typically much less limited.

The purpose of this research is to optimize and adapt the ITD Algorithm for use on mini-computers. Significant optimization has been achieved by accommodating items peculiar to the ITD theory that cannot be included in general eigensolution routines. The methods of optimization include:

1. choosing specialized numeric routines to replace general purpose routines
2. eliminating unnecessary steps in numeric routines
3. reducing the number of iterations through acceleration techniques
4. use of common and one-dimensional array storage

In the following two chapters, the ITD Algorithm is discussed in detail. The first chapter covers the theory behind the algorithm. The second chapter covers the user-selectable parameters and methods used to insure accuracy when experimental noise is present. The remainder of the paper is devoted to optimizing the ITD numerical process of identifying modal parameters: modal frequency, damping factor and mode shape.
The governing equation of motion for a free, single degree of freedom system is

$$ M\ddot{x} + Cx + Kx = 0. \quad (II-1) $$

A solution to this second order, homogeneous, differential equation is

$$ x(t) = \psi e^{\lambda t}, \quad (II-2) $$

where $\lambda$ = root of the characteristic equation,

$$ m\lambda^2 + c\lambda + k = 0 $$

$\lambda$ = eigenvalue of the system

$\psi$ = scale factor.

Assuming the structure to be underdamped, $\lambda$ and $\psi$ occur in complex conjugate pairs.

The free response of a linear, infinite-degree of freedom systems can be approximated as the sum of $m$ complex normal modes of vibration. At time $t$, the structural response may be written as
\[ \{x(t)\} = \sum_{k=1}^{2m} \{\psi_k^*\} e^{\lambda_k t}. \quad \text{(II-3)} \]

where

\[ \{\psi_k\} = \text{k'th normal mode of vibration in a non-normalized form} \]

\[ \lambda_k = \text{k'th root of the characteristic equation.} \]

The characteristic equation has 2m roots, or m complex conjugate pairs of roots. The complex root may be written as

\[ \lambda_k = (\alpha + i\beta)_k. \quad \text{(II-4)} \]

The relationship between the complex eigenvalue, \((a + ib)\), and the root of the characteristic equation, \(\lambda\), is

\[ a + ib = e^{\lambda t} \]

\[ = e^{(\alpha + i\beta)t} \]

\[ = e^{\alpha t} (\cos \beta t + i \sin \beta t). \quad \text{(II-5a)} \]

\[ = e^{\alpha t} \]

\[ = e^{\alpha t} (\cos \beta t + i \sin \beta t). \quad \text{(II-5b)} \]

\[ = e^{\alpha t} (\cos \beta t + i \sin \beta t). \quad \text{(II-5c)} \]

Using equation (II-3), the response of the i'th location at time \(t_j\) is written as

\[ x_i(t_j) = \sum_{k=1}^{2m} \psi_{ik} e^{\lambda_k t_j}. \quad \text{(II-6)} \]

Equation (II-6) can be written in matrix form for 2m response locations at s instances of time. First, however, the following matrices must be defined.

Response Matrix, \(\Phi\)

\[ \Phi = \{\{x(t_1)\}, \{x(t_2)\}, \ldots, \{x(t_s)\}\} \]
\[ \phi = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,s} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,s} \\ \vdots & \vdots & \ddots & \vdots \\ x_{2m,1} & x_{2m,2} & \cdots & x_{2m,s} \end{bmatrix} \]  

(II-7)

Matrix of Eigenvectors, \( \psi \)

\[ \psi = \begin{bmatrix} \psi_1 \{ \psi_2 \} \cdots \{ \psi_{2m} \} \end{bmatrix} \]

\[ \psi = \begin{bmatrix} \psi_{1,1} & \psi_{1,2} & \cdots & \psi_{1,2m} \\ \psi_{2,1} & \psi_{2,2} & \cdots & \psi_{2,2m} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{2m,1} & \psi_{2m,2} & \cdots & \psi_{2m,2m} \end{bmatrix} \]  

(II-8)

Matrix of Eigenvalues, \( \Lambda \)

\[ \Lambda = \begin{bmatrix} \lambda_1 t_1 & \lambda_1 t_2 & \cdots & \lambda_1 t_s \\ \lambda_2 t_1 & \lambda_2 t_2 & \cdots & \lambda_2 t_s \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{2m} t_1 & \lambda_{2m} t_2 & \cdots & \lambda_{2m} t_s \end{bmatrix} \]  

(II-9)

The structural response can now be written in matrix form for \( 2m \) response locations at \( s \) instants of time.
\[ \phi = \psi \Lambda \] (II-10)

\[
\begin{bmatrix}
  x_{1,1} & x_{1,2} & \cdots & x_{1,s} \\
  x_{2,1} & x_{2,2} & \cdots & x_{2,s} \\
  \vdots & \vdots & \ddots & \vdots \\
  x_{2m,1} & x_{2m,2} & \cdots & x_{2m,s}
\end{bmatrix}
\]

\[
\begin{bmatrix}
  \psi_{1,1} & \psi_{1,2} & \cdots & \psi_{1,s} \\
  \psi_{2,1} & \psi_{2,2} & \cdots & \psi_{2,s} \\
  \vdots & \vdots & \ddots & \vdots \\
  \psi_{2m,1} & \psi_{2m,2} & \cdots & \psi_{2m,s}
\end{bmatrix}
\begin{bmatrix}
  \lambda_{1t1} & \lambda_{1t2} & \cdots & \lambda_{1ts} \\
  \lambda_{2t1} & \lambda_{2t2} & \cdots & \lambda_{2ts} \\
  \vdots & \vdots & \ddots & \vdots \\
  \lambda_{2mt1} & \lambda_{2mt2} & \cdots & \lambda_{2mts}
\end{bmatrix}
\]

The response matrix element \( x_{i,j} \) (location \( i \) at time \( t_j \)) is now quickly found by multiplying row \( i \) of \( \psi \) by column \( j \) of \( \Lambda \).

\[ x_{i,j} = (\text{Row } i \text{ of } \psi) \times (\text{Column } j \text{ of } \Lambda) \] (II-11)

\[ = \sum_{k=1}^{2m} \psi_{i,k} \lambda_{k} t_j \]

This equation is identical to equation (II-6).

Now, rewrite equation (II-3) with the measurement time \( t_j \) shifted a \( \Delta t \), so that

\[ \{ x(t_j + \Delta t) \} = \sum_{k=1}^{2m} \psi_{k} e^{\lambda_{k} (t_j + \Delta t)} \] (II-12a)
The new response vector \( \{x(t_j + \Delta t)\} \) will be called \( \{\hat{x}(t_j)\} \). The notation "^*" denotes a time shift of \( \Delta t \) from \( t_j \). \( \Delta t \) remains constant for every \( t_j \). Hence, the second exponential factor in equation (II-12b), \( e^{\lambda_k \Delta t} \), is a constant value for each \( \lambda_k \).

The eigenvector, \( \{\psi_k\} \), for any \( t_j \), may be modified to include the factor \( e^{\lambda_k \Delta t} \). Let \( \gamma_k = e^{\lambda_k \Delta t} \), such that

\[
\{\psi_k\} e^{\lambda_k t_j} e^{\lambda_k \Delta t} = \{\psi_k\} e^{\lambda_k t_j} \gamma_k
\]

\[
\{\psi_k\} e^{\lambda_k t_j} e^{\lambda_k \Delta t} = (\gamma_k \{\psi_k\}) e^{\lambda_k t_j}
\]

\[
= \{\hat{\psi}_k\} e^{\lambda_k t_j},
\]

(II-13)

where:

\[
\{\hat{\psi}_k\} = \gamma_k \{\psi_k\} = e^{\lambda_k \Delta t} \{\psi_k\}.
\]

(II-14)

The notation "^*" again denotes that \( \{\hat{\psi}_k\} \) differs from \( \{\psi_k\} \) by some function of \( \Delta t \).

Combine equations (II-12b) and (II-13) to get:

\[
\{x(t_j)\} = \sum_{k=1}^{2m} \{\hat{\psi}_k\} e^{\lambda_k t_j}.
\]

(II-15)

Equation (II-15) may be put in matrix notation for \( 2m \) response locations and \( s \) instances of time. Define matrices for \( \{x(t_j)\} \) and \( \{\hat{\psi}_k\} \).
in a similar manner to $\phi$ and $\psi$.

Response Matrix, $\hat{\phi}$

$$\hat{\phi} = \{x(t_1 + \Delta t), x(t_2 + \Delta t), \ldots, x(t_s + \Delta t)\}$$

$$= \{\hat{x}(t_1), \hat{x}(t_2), \ldots, \hat{x}(t_s)\}$$

$$= [\begin{array}{cccc}
\hat{x}_{1,1} & \hat{x}_{1,2} & \cdots & \hat{x}_{1,s} \\
\hat{x}_{2,1} & \hat{x}_{2,2} & \cdots & \hat{x}_{2,s} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{x}_{2m,1} & \hat{x}_{2m,2} & \cdots & \hat{x}_{2m,s}
\end{array}]$$

Matrix of eigenvectors, $\hat{\psi}$

$$\hat{\psi} = \{\hat{\psi}_1, \hat{\psi}_2, \ldots, \hat{\psi}_{2m}\}$$

$$= [\begin{array}{cccc}
\hat{\psi}_{1,1} & \hat{\psi}_{1,2} & \hat{\psi}_{1,2m} \\
\hat{\psi}_{2,1} & \hat{\psi}_{2,2} & \hat{\psi}_{2,2m} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{\psi}_{2m,1} & \hat{\psi}_{2m,2} & \hat{\psi}_{2m,2m}
\end{array}]$$

The matrices $\hat{\phi}$, $\hat{\psi}$ and $\Lambda$ are combined using equation (II-15) to give

$$\hat{\phi} = \hat{\psi}\Lambda.$$
The matrix $\hat{\Psi}$ can be expanded as the product of two matrices as follows:

$$\hat{\Psi} = [\{\hat{\psi}_1\}{\hat{\psi}_2} \ldots {\hat{\psi}_{2m}}]$$

$$= [y_1{\psi}_1 y_2{\psi}_2 \ldots y_{2m}{\psi}_{2m}]$$

$$= [{\psi}_1{\psi}_2 \ldots {\psi}_{2m}] \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{2m} \end{bmatrix}$$

$$\hat{\Psi} = \Psi[\gamma]$$

The matrix $[\gamma]$ is a diagonal matrix of the constants $\gamma_k = e^{\lambda_k \Delta t}$.

Combine equations (II-18) and (II-19) to get:

$$\hat{\phi} = \Psi[\gamma] \Lambda.$$
In summary, the structural response of a multi-degree of freedom system, for 2m response locations at time \( t \), is

\[
\{x(t)\} = \sum_{k=1}^{2m} \{\psi_k\} e^{\lambda_k t}.
\]

(II-3)

At time \( t + \Delta t \), the response of the same structure is

\[
\{x(t + \Delta t)\} = \sum_{k=1}^{2m} \{\psi_k\} e^{\lambda_k(t + \Delta t)}.
\]

(II-12a)

These equations may be written in matrix form. The response equation for \( s \) instants of time, \( t_1 \) to \( t_s \), is

\[
\phi = \Psi \Lambda.
\]

(II-10)

Displace each time instant by \( \Delta t \), \( (t_1 + \Delta t) \) through \( (t_s + \Delta t) \), to get the second response equation.

\[
\hat{\phi} = \hat{\Psi} \Lambda = \Psi \{\gamma\} \Lambda
\]

(II-18, 20)

The response matrices, \( \phi \) and \( \hat{\phi} \), may be obtained experimentally. The free decay acceleration, velocity or displacement values at 2m measurement locations and time \( t_j \) correspond to column \( j \) of \( \phi \). At time \( t_j + \Delta t \), the 2m data values correspond to column \( j \) of \( \hat{\phi} \).

The eigenvalue matrix, \( \Lambda \), is common to both equations (II-10) and (II-20). Solving equation (II-10) for \( \Lambda \) and substituting it
into equation (II-20) yields
\[ \hat{\phi} = \phi [\gamma_{-}] \phi^{-1} \phi \]  
(II-21)
Post multiplying this by \( \phi^{-1} \) yields
\[ \hat{\phi} \phi^{-1} = \phi [\gamma_{-}] \phi^{-1} \]  
(II-22)
Next, postmultiplying by \( \hat{\phi} \) yields
\[ (\hat{\phi} \phi^{-1}) \phi = \phi [\gamma_{-}] \]  
(II-23)
Equation (II-23) has the form of an eigenvalue problem where
\[ A = \hat{\phi} \phi^{-1} \]  
(II-24)
and \( [\gamma_{-}] = [\gamma_{-}] \).

Hence, the matrix product \( \hat{\phi} \phi^{-1} \) is a type of "system matrix", although not the true system matrix of the mechanical system. The eigenvalues of the "system matrix" of equation (II-24) are the complex exponential function \( \gamma_{k} = e^{\lambda k \Delta t} \). This "system matrix" is not unique. Rather, it is dependent on how the response matrices are filled.

The above derivation shows that the "system matrix" can be obtained from the response matrices \( \phi \) and \( \hat{\phi} \). Since it is not feasible to compute the inverse of \( \phi \), equation (II-24) is rewritten as
\[ A \phi = \hat{\phi} \]  
(II-25)
The above equation shows that a system matrix, \( A \), can be obtained from the two experimentally determined response matrices, \( \phi \) and \( \hat{\phi} \).
The transpose of this equation could be used to solve for $A$. However, it is simpler to pre-multiply the tranposed equation by $\phi$, thus making all the matrices square.

\[(\phi\phi^T)A^T = (\phi\phi^T)\]  

\[
[(2m \times s)(s \times 2m)](2m \times 2m) = [(2m \times s)(s \times 2m)]
\]

Solving equation (II-26) for $A^T$ requires solving $2m$ sets of $2m$ linear simultaneous equations. The steps in this solution include a Choleski Decomposition of $(\phi\phi^T)$, followed by Gaussian elimination forward and back substitution.

The system matrix, $A$, is now used to find the eigenvalues and their corresponding right hand eigenvectors. This procedure is covered in Chapters III and IV.

Assuming the structure to be underdamped, the eigenvalues and eigenvectors will occur in complex conjugate pairs. The modal parameters are determined from the complex conjugate eigenvalues, $a \pm ib$, and eigenvectors $\{p \pm iq\}$. The modal parameters to be determined are the natural frequencies, $\omega_n$, damped natural frequencies, $\omega_d$, damping coefficients, $\zeta$, and mode shapes. The first three parameters are determined from the eigenvalues, or more specifically, from the roots of the characteristic equation, $\lambda$. Recall equation (II-5a and 5b) which stated the following relation
Using fundamental vibration theory for an underdamped system, the root of the characteristic equation may be rewritten in terms of the modal parameters.

\[ a + ib = e^{\lambda \Delta t} \]  
\[ = e^{(\alpha + i\beta)\Delta t} \]  
\[ = e^{\alpha \Delta t} e^{i\beta \Delta t} \]

The eigenvalue, \( a + ib \), is now written in terms of the modal parameters: \( \omega_n \), \( \omega_d \) and \( \zeta \).

\[ \alpha = \zeta \omega_n \]  
\[ \beta = \omega_n \sqrt{1 - \zeta^2} = \omega_d \]

Since the eigenvalue, \( a + ib \), is known, equation (II-29) is solved for the modal parameters.

\[ a + ib = e^{(\zeta \omega_n + i\omega_d)\Delta t} \]  
\[ = e^{\zeta \omega_n \Delta t} e^{i\omega_d \Delta t} \]

Equate the real and imaginary parts on the right sides of equations (II-27c) and (II-30) to get

\[ \alpha = \frac{1}{\Delta t} \ln (a^2 + b^2)^{\frac{1}{2}} \]
\[ \alpha = \frac{1}{\Delta t} \ln (a^2 + b^2) \]  

(II-31)

\[ \beta = \frac{1}{\Delta t} \tan^{-1} \left( \frac{b}{a} \right) \]  

(II-32)

From equations (II-28a and b), the modal parameters are

\[ \omega_d = \beta \]  

(II-33a)

\[ \omega_n = (\alpha^2 + \beta^2)^{1/2} \]  

(II-33b)

\[ \zeta = \alpha / \omega_n \]  

(II-33c)

The last modal parameter to be determined is the mode shape. Recall how the "system matrix" was defined:

\[ A\{\psi_k\} = e^{\lambda_k \Delta t} \{\psi_k\} \ . \]

The k'th eigenvalue, \( e^{\lambda_k \Delta t} \), is known, so the above equation can be rewritten as

\[ [A - e^{\lambda_k \Delta t} I] \{\psi_k\} = 0. \]  

(II-34)

In a later section, we will show how \( \{\psi_k\} \) is determined using equation (II-34) and a process of back substitution.

All of the modal parameters have now been determined.
CHAPTER III
USER-SELECTABLE ALGORITHM CONSTANTS

Since all experimental data has some degree of noise, accommodations must be made to maintain accuracy in the solution. A standard method is to use an oversized math model of the system. If the system is expected to have $n$ structural modes within a given frequency range, it is modeled with $m$ modes, where $m >> n$. The dimension $m$ represents the number of computational modes, while $n$ represents the number of true structural modes. The remaining $(m - n)$ modes will be called "noise modes".

The ITD employs the oversized math model strategy and others to reduce the effects of experimental noise. Before any of these strategies are discussed, three user-selectable parameters will briefly be defined. These are three time shifts which are used to fill the response matrices, $\phi$ and $\hat{\phi}$. The first time shift, $\Delta t_1$, has already been discussed. It is the time shift between the two response matrices. The second time shift, $\Delta t_2$, is used to generate "pseudo measurements". The third time shift, $\Delta t_3$, is used so that Modal Confidence Factors, $MCF$, can be calculated for each mode. The $MCF$ indicates whether the identified mode is a structural mode or a computational noise mode. When specifying these time shifts, the actual parameters which will be specified are $N_1$, $N_2$ and $N_3$. They represent the number of consecutive time samples corresponding to each shift.
The time shift, \( \Delta t_i \), is therefore the number of time samples, \( N_i \), divided by sample frequency.

One additional parameter which the user must define is \( s \). This represents the number of columns, or time samples, in the response matrices \( \Phi \) and \( \hat{\Phi} \).

Assume the number of free response functions actually measured is \( p \). There are \( n \) structural modes, where \( n \) may be greater or less than \( p \). The \( p \) response functions are not noise-free. Hence, the system is modeled with \( m \) computational modes, where \( m \gg p \) and \( n \), creating an oversized math model. A total of \( 2m \) free response functions are required in the oversized math model; there are \( 2m \) rows in the response matrices \( \Phi \) and \( \hat{\Phi} \). Hence, \((2m - 2p)\) pseudo measurements must be created. As the percentage of noise increases, the number of pseudo measurements must increase.

The response of the system can be expressed as

\[
\{x(t)\} = \sum_{k=1}^{2n} \{\psi_k\} e^{\lambda_k t} + \{n(t)\},
\]

where \( \{n(t)\} \) is the noise portion of the data. When the math model allows for \( m \) modes, the noise is also modeled.

\[
\{x(t)\} = \sum_{k=1}^{2n} \{\psi_k\} e^{\lambda_k t} + \sum_{k=2n+1}^{2m} \{n_k\} e^{\lambda_k t}
\]

The noise is modeled as \( 2m - 2n \) complex exponential functions (Pappa and Ibrahim 1981a). As this number increases, it allows for a higher
order fit of the noise in the free response functions. In other words, the additional \( m - n \) modes provide an exit for the noise, thus increasing the accuracy of the structural modes determined.

The free response functions actually measured are used to fill the first \( p \) rows of \( \phi \). The remaining \( m - p \) rows in the top half of \( \phi \) are filled with pseudo measurements generated with the shift \( \Delta t_2 \). The shift \( \Delta t_2 \) is used to generate the second \( p \) rows, \( 2\Delta t_2 \) in the third \( p \) rows, etc., until the first \( m \) rows of \( \phi \) are filled.

At this point, some method of distinguishing between structural modes and noise modes must be determined. The eigenvector \( \{ \psi_j \} \), with corresponding eigenvalue \( \lambda_j \Delta t_1 \), is extracted as a mode of vibration. The displacement of location \( i \) corresponds to element \( \psi_{ij} \). The displacement of location \( i \) at some time \( \Delta t_3 \) later may be denoted as \( \psi_{ij}^* \). If \( \{ \psi_j \} \) is a structural mode, the displacement \( \psi_{ij}^* \) can be predicted from \( \psi_{ij} \) by the relation

\[
\psi_{ij}^* = \psi_{ij} e^{\lambda_j \Delta t_3}.
\] (III-3)

In general, if \( \{ \psi_j \} \) is a structural mode, the relation above should be satisfied for all locations:

\[
\{ \psi_j \}^* = \{ \psi_j \} e^{\lambda_j \Delta t_3}.
\] (III-4)

This relation can be used to establish a degree of correlation between any two mode shapes. The \( i \)'th elements of any two computed modes will exhibit the following equality:
The complex Modal Confidence Factor gives a degree of correlation between $\psi_{ij}$ and $\psi_{ij}^*$. For perfect correlation, the MCF will have a unit magnitude and zero phase. The MCF for any calculated mode is therefore

$$\text{MCF} = \frac{\psi_{ij}^*}{\psi_{ij} e^{\lambda_j \Delta t_3}}$$  \hspace{1cm} (III-6a)

or

$$\text{MCF} = \frac{\psi_{ij}}{\psi_{ij}^* e^{\lambda_j \Delta t_3}}.$$  \hspace{1cm} (III-6b)

Equation (III-6a or 6b) is used, depending on which results in a magnitude less than or equal to unity.

Recall that only the first $m$ rows of the $(2m \times s)\phi$ matrix have been filled. The first $p$ rows are measured free response functions, while $(m - p)$ rows are pseudo measurements. The relation in equation (III-4) is applied in filling the bottom $m$ rows. Each row $i$ in the top half of $\phi$ is delayed $\Delta t_3$ and placed in row $i + m$. In this way, the first $m$ rows of $\phi$ will be related to the second $m$ rows by $e^{\lambda_j \Delta t_3}$. This process can be thought of as two simultaneous solutions using the same free response functions. The first solution is obtained from the top half of $\phi$, where $t = t_1$ to $t_s$. The second solution is
obtained from the bottom half of $\diamond$, where $t = t_1 + \Delta t_3$ to $t_s + \Delta t_3$. The Modal Confidence Factors are obtained by comparing the two eigenvectors; $\{\psi_j\}$ is the first $m$ elements of the eigenvector, while $\{\psi_j\}^*$ is the second $m$ elements. In this manner, equations (III-6a and 6b) are rewritten for the elements of the eigenvector $\{\psi_j\}$ of length $2m$, as

$$MCF = \frac{\psi_i e^{\lambda_j \Delta t_3}}{\psi_{i+m,j}}$$  \hspace{1cm} (III-7a)

or

$$MCF = \frac{\psi_{i+m,j}}{\psi_i e^{\lambda_j \Delta t_3}}$$  \hspace{1cm} (III-7b)

to ensure a magnitude less than or equal to unity.

Each mode will have $m$ MCF's. These may be combined into an Overall Modal Confidence Factor, OAMCF. The OAMCF is the percentage of $m$ elements whose magnitude is at least 95% and has a phase within $10^\circ$ of $0^\circ$. An OAMCF of 90 to 100% would clearly indicate a structural mode. As this percentage decreases, computational noise modes become evident. Caution should be exercised when using the OAMCF since low values may be the result of local modes, a mode with many nodal points, or bad measurement data (Ibrahim 1983b).
CHAPTER IV
DETERMINING THE "SYSTEM MATRIX", A

The purpose of this chapter is to explain how the "system matrix", A, is determined. The equation being solved is:

\[(\phi\phi^T)A^T = (\hat{\phi}\hat{\phi}^T).\]  \hspace{1cm} (IV-1)

This equation represents 2m sets of 2m linear simultaneous equations.

The chapter is broken down into four sections. Each section corresponds to one major step in determining A. These steps are:

1. Filling the response matrices, \(\phi\) and \(\hat{\phi}\), according to the user-selectable parameters.
2. Calculating the matrix products \((\phi\phi^T)\) and \((\hat{\phi}\hat{\phi}^T)\) to obtain square \((2m \times 2m)\) matrices.
3. Choleski Decomposition of \((\phi\phi^T)\).
4. Gaussian elimination forward and back substitution to determine \(A^T\).

Filling Response Matrices \(\phi\) and \(\phi^T\)

The response matrices, \(\phi\) and \(\hat{\phi}\), have the dimensions \((2m \times s)\).

In this chapter, and those that follow, \(m\) will be called NDOF, representing the number of computational degrees of freedom. The second parameter, \(s\), will be called NCOL, representing the number of columns. The dimensions of \(\phi\) and \(\hat{\phi}\) are now \((2NDOF \times NCOL)\). In filling
the response matrices, the time shift parameters must also be chosen: \( N_1, N_2 \) and \( N_3 \). \( N_1 \) corresponds to the time shift between the two response matrices, \( \Delta t_1 \). \( N_2 \) corresponds to \( \Delta t_2 \) and is used to generate the "pseudo measurements". \( N_3 \) corresponds to \( \Delta t_3 \) and is used so that MCF's may be calculated.

The number of actual measurements is represented by the symbol "p". Assuming the free response functions are not noise-free, \( p \) must be less than DOF. The number of pseudo measurements to be generated are DOF less \( p \). These pseudo measurements, generated with \( \Delta t_2 \), fill the remaining top and bottom halves of \( \phi \) and \( \hat{\phi} \).

An example follows which shows how the matrices are filled once the necessary parameters are chosen. This example was obtained from a paper authored by Ibrahim and Pappa (1981a). There are three actual free response functions with a hundred time samples each:

\[
\begin{align*}
A: & \quad a_1 \quad a_2 \quad a_3 \quad \ldots \quad a_{100} \\
B: & \quad b_1 \quad b_2 \quad b_3 \quad \ldots \quad b_{100} \\
C: & \quad c_1 \quad c_2 \quad c_3 \quad \ldots \quad c_{100}
\end{align*}
\]

The desired computational degrees of freedom, \( \text{NDOF} \), are seven. There must be four pseudo measurements added to fill the upper and lower halves of \( \phi \) and \( \hat{\phi} \). The selected parameters are listed below.

\[
\begin{align*}
p & = 3 & N_1 & = 3 \\
\text{NDOF} & = 7 & N_2 & = 8 \\
\text{NCOL} & = 30 & N_3 & = 4
\end{align*}
\]
The time shifts, $\Delta t_i$, are obtained by dividing the number of consecutive time samples, $N_i$, by the sample frequency, $SF$.

$$\Delta t_1 = \frac{N_1}{SF}$$
$$\Delta t_2 = \frac{N_2}{SF}$$
$$\Delta t_3 = \frac{N_3}{SF}$$

With the parameters chosen, the response matrices are now filled as shown in Figure 1.

Figure 1 shows how the time shifts are used to fill matrices $\Phi$ and $\Phi$. The pseudo measurements, rows four through seven and eleven through fourteen, are generated using shift $N = 8$. The elements of row four $(p + 1)$ and seven $(2p + 1)$ are obtained from measurement $A$ by shifting $a_{i+8}$ and $a_{i+16}$, respectively. The bottom half of the matrices, rows eight through fourteen, are generated using shift $N = 4$. The elements of rows eight $(N_{DOF} + 1)$ and nine $(N_{DOF} + 2)$ are obtained from measurements $A$ and $B$ by shifting $a_{i+4}$ and $b_{i+4}$, respectively. Combining the two shifts, $(N_2 + N_3)$ row eleven $(N_{DOF} + p)$ is obtained from $A$ by shifting $a_{i+12}$. Finally, $\Phi$ is obtained from $\Phi$ using shift $N_1 = 3$. The elements of column $j$ are obtained by shifting either $a_{i+3}$, $b_{i+3}$ or $c_{i+3}$, depending on the row. The matrix products $(\Phi \Phi^T)$ and $(\Phi \Phi^T)$ must be performed in the next step of solving for the system matrix, $A$. Taking into account how these products can efficiently be performed will greatly affect how $\Phi$ and $\Phi$ are stored.
Fig. 1. Incorporating $\Delta t_1$, $\Delta t_2$ and $\Delta t_3$ to fill response matrices (Ibrahim and Pappa 1981a).
Before discussing how $\phi$ and $\hat{\phi}$ are stored, however, the following matrix notation must be defined. This notation will be used in the remainder of the paper. The $i$'th row of a matrix is denoted by a single subscript. The $j$'th column of a matrix is denoted by a single superscript. The element $(i, j)$, rather, the element of row $i$ and column $j$, is denoted by a double subscript. Using the matrix $\phi$, examples of these follows.

$\phi_i$ - row $i$ of $\phi$

$\phi^j$ - column of $\phi$

$\phi_{ij}$ - element $(i, j)$ of $\phi$

Now, refer back to the matrix products $(\phi \phi^T)$ and $(\hat{\phi} \hat{\phi}^T)$. Element $(i, j)$ of the matrix product $(\phi \phi^T)$ is obtained by multiplying row $i$ of $\phi$ by column $j$ of $\phi^T$.

$$(\phi \phi^T)_{ij} = \phi_i \ast \phi^T_j \quad (IV-2)$$

The element $(i, j)$ of the matrix product $(\hat{\phi} \hat{\phi}^T)$ is obtained in a similar manner, such that

$$(\hat{\phi} \hat{\phi}^T)_{ij} = \hat{\phi}_i \ast \hat{\phi}^T_j \quad (IV-3)$$

Noting that the $i$'th row of $\phi$ is the same as the $i$'th column of $\phi^T$, equations (IV-2) and (IV-3) may be rewritten as

$$(\phi \phi^T)_{ij} = \phi_i \ast \phi^T_j \quad (IV-4)$$
Using equation (IV-4) and (IV-5), it appears preferable to store the transpose of the response matrices, $\phi^T$ and $\hat{\phi}^T$, with dimensions (NCOL, 2NDOF). Refer back to Figure 1, which shows how $\phi$ and $\hat{\phi}$ are stored, according to the chosen parameters. The transpose of $\phi$, as shown in Figure 1, is given in Figure 2.

The only operations performed with $\phi^T$ and $\hat{\phi}^T$ are the matrix products $(\phi\phi^T)$ and $(\hat{\phi}\hat{\phi}^T)$. Hence, $\phi^T$ and $\hat{\phi}^T$ are only accessed an entire column at a time, as shown in equations (IV-4) and (IV-5). With this being the case, $\phi^T$ and $\hat{\phi}^T$ can very efficiently be used if stored by columns in a one-dimensional array.

In summary, the response matrices, $\phi$ and $\hat{\phi}$, contain the free response functions. The system matrix, $A$, is obtained from the matrix products $(\phi\phi^T)$ and $(\hat{\phi}\hat{\phi}^T)$. The response matrices are filled according to the user-selectable parameters:

- $p$ - number of actual free response functions
- NDOF - number of computational degrees of freedom
- NCOL - number of columns, or time samples, in $\phi$ and $\hat{\phi}$
- $N_1$ - corresponds to $\Delta t_1$, the time shift between $\phi$ and $\hat{\phi}$
- $N_2$ - corresponds to $\Delta t_2$, which is used to generate pseudo measurements
- $N_3$ - corresponds to $\Delta t_3$, which is used to calculate MCF's

The transpose of the response matrices, $\phi^T$ and $\hat{\phi}^T$, are stored by columns in a one-dimensional array as shown in equations (IV-6a/b).
\[ \phi^T = \begin{bmatrix} a_1 & b_1 & c_1 & a_9 & b_9 & c_9 & a_{17} \\ a_2 & b_2 & c_2 & a_{10} & b_{10} & c_{10} & a_{18} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{30} & b_{30} & c_{30} & a_{38} & b_{38} & c_{38} & a_{46} \end{bmatrix} \begin{bmatrix} a_5 & b_5 & c_5 & a_{13} & b_{13} & c_{13} & a_{21} \\ a_6 & b_6 & c_6 & a_{14} & b_{14} & c_{14} & b_{21} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{34} & b_{34} & c_{34} & a_{42} & b_{42} & c_{42} & a_{50} \end{bmatrix} \]

Fig. 2. Transpose of response matrix, \( \phi \)
Calculating ($\Phi^T\Phi$) and ($\Phi^T\hat{\Phi}$)

In FORTRAN and other programming languages, array indices increment, by default, in order of the leftmost index first. As a direct consequence of this, a two-dimensional array will, by default, be stored by columns. Hence, all operations are most efficiently performed by columns. This minimizes the time spent searching for array elements.

The free response function matrices, $\Phi$ and $\hat{\Phi}$, are used to obtain the "system matrix", $A$. This was shown in equation (IV-1). The matrix products, ($\Phi^T\Phi$) and ($\hat{\Phi}^T\hat{\Phi}$), are used since square matrices are necessary in the following decomposition and Gaussian elimination steps. The resulting square ($2\text{NDOF} \times 2\text{NDOF}$) matrices will be called $\overline{X}$ and $\hat{\overline{X}}$, where:

$$\overline{X} = \Phi^T\Phi \quad \text{(IV-7)}$$

$$\hat{\overline{X}} = \hat{\Phi}^T\hat{\Phi} \quad \text{(IV-8)}$$

Equation (IV-1) may be rewritten using equations (IV-7) and (IV-8) as
In a similar manner, equations (IV-4) and (IV-5) may be rewritten for element \((i, j)\) as

\[
\begin{align*}
X_{ij} &= \hat{\phi}_i \cdot \hat{\phi}_j \quad \text{(IV-10)} \\
\hat{X}_{ij} &= \hat{\phi}_i \cdot \hat{\phi}_j. \quad \text{(IV-11)}
\end{align*}
\]

The procedure of obtaining \(X\) and \(\hat{X}\) from \(\phi\) and \(\hat{\phi}\) can be optimized in many ways. The following five paragraphs discuss techniques used in this procedure.

1. The transpose of \(\phi\) and \(\hat{\phi}\) are stored so that the rows of both matrices become columns. Multiplying row \(i\) of \(\phi\) by column \(j\) of \(\phi^T\) (or \(\phi^\top\)) becomes the multiplication of column \(i\) of \(\phi^T\) by column \(j\) of \(\phi^T\) (or \(\phi^\top\)). All multiplications performed are column operations.

2. Matrices \(\phi^T\) and \(\hat{\phi}^T\) are stored as one-dimensional arrays. They are stored by columns (column 1, column 2, ..., column \(2\text{NDOF}\)) as was shown in equations (IV-6a) and (IV-6b).

3. To obtain column \(j\) of \(X\), columns one through \(2\text{NDOF}\) (\(i = 1\) to \(2\text{NDOF}\)) of \(\phi^T\) are multiplied by column \(j\) of \(\phi^T\). To account for the one-dimensional store, column \(j\) of \(\phi^T\) is stored in a separate work vector of length NCOL. This work vector is multiplied by the entire one-dimensional array \(\phi^T\), NCOL elements at a time. The result is column \(j\) of \(X\) with \(2\text{NDOF}\) elements. All matrix operations have been performed by columns.
4. To obtain column $j$ of $\hat{X}$, columns one through $2\text{NDOF}$ ($i = 1$ to $2\text{NDOF}$) of $\phi^T$ are multiplied by column $j$ of $\hat{\phi}^T$. Column $j$ of $\hat{\phi}^T$ is stored in a work vector of length NCOL. This work vector is multiplied by the entire one-dimensional array $\phi^T$, NCOL elements at a time. The result is column $j$ of $\hat{X}$ with $2\text{NDOF}$ elements.

5. The result of multiplying a matrix by its own transpose is a symmetric positive definite matrix. The matrix $X$ is, therefore, a symmetric positive definite matrix. The implications of this are significant. A result of $X$ being symmetric is that only the lower or upper triangle need be computed. This reduces the number of row ($i$) * column ($j$) operations to almost half of those necessary for a non-symmetric matrix; $\frac{1}{2}(m^2 + m)$ multiplications are required for a symmetric matrix, versus $m^2$ for a non-symmetric matrix. In this procedure, the lower triangular elements of $X$ are calculated.

**Choleski Decomposition of $X$**

With the square data matrices, $X$ and $\hat{X}$, determined, the next step in determining $A$ is the decomposition of $X$. Recall equation (IV-9) which stated:

$$X A^T = \hat{X} \quad \text{(IV-9)}$$

Equation (IV-9) is composed of $N$ sets of $N$ linear simultaneous equations, where $N = 2\text{NDOF}$. Matrix $X$ is the matrix of coefficients for each set of simultaneous equations.
The standard procedure to solve equation (IV-9) for $A^T$ is to decompose $X$ into an upper and lower triangular matrix.

$$X = LU,$$  \hspace{1cm} \text{(IV-12)}

where

$L =$ lower triangular matrix

$U =$ upper triangular matrix

Now, equation (IV-9) may be rewritten as

$$LUA^T = \hat{X}.$$  \hspace{1cm} \text{(IV-13)}

The following equations are then used to determine $A^T$.

$$LY = \hat{X} \rightarrow Y.$$  \hspace{1cm} \text{(IV-14)}

$$UA^T = Y \rightarrow A^T.$$  \hspace{1cm} \text{(IV-15)}

This procedure is known as the LU Decomposition. With $L$ being a lower triangular matrix, the intermediate matrix, $Y$, is found through forward substitution. Then, $A^T$ is found from the upper triangular matrix, $U$, and the intermediate matrix, $Y$, using back substitution. This procedure is non-iterative, stable and very straightforward since $L$ and $U$ are triangular matrices. The above procedure is equivalent to Gaussian elimination. The forward substitution step in which $Y$ is found is equivalent to the right-hand reduction stage of Gaussian elimination. The back substitution step in which $A^T$ is found is equivalent to the back substitution process of Gaussian
elimination. Typically, some form of partial pivoting is included in the decomposition of $X$ to increase stability (Jennings 1977).

It was shown in the previous section that the matrix of coefficients in our application, $X$, is a symmetric positive definite matrix. The most important result of $X$ being symmetric positive definite is that $A^T$ can be determined using a Choleski Decomposition, rather than the more general LU Decomposition.

A symmetric coefficient matrix can be decomposed into the following:

$$X = LDL^T,$$  \hspace{1cm} (IV-16)

where $L$ is a lower triangular matrix with a unit diagonal, and $D$ is a diagonal matrix. For a symmetric positive definite matrix, the diagonal matrix, $D$, will be positive definite and can be written as

$$D = D^{\frac{1}{2}} \ast D^{\frac{1}{2}}.$$  \hspace{1cm} (IV-17)

Since all $d_{ii}$ are positive, all $\sqrt{d_{ii}}$ are real. The final decomposition of the coefficient matrix becomes

$$X = LD^{\frac{1}{2}} \ast D^{\frac{1}{2}} L^T,$$  \hspace{1cm} (IV-18)

where:

$$L = LD^{\frac{1}{2}}.$$  \hspace{1cm} (IV-19)

The final decomposition equation becomes
The matrix $L$ is a real, lower triangular matrix. The symmetric positive definite matrix $\bar{X}$ can be decomposed into a lower triangular matrix. Recalling that $\bar{X}$ was stored in a lower triangular matrix, the storage space can be fully optimized by performing the entire decomposition and storing the results in the same lower triangular matrix. The Choleski Decomposition requires approximately half the number of computations for a LU Decomposition (Jennings 1977).

A further savings in necessary computation is achieved from the fact that pivoting of any sort in a Choleski Decomposition is not needed. If partial pivoting were employed, the symmetry of the reduction would be lost. This would forfeit the savings in computation and storage space. Furthermore, partial pivoting is not needed because the remaining coefficient matrix to be decomposed is symmetric positive definite throughout the reduction. The diagonal elements are positive and the product of any pair of diagonal elements is greater than the corresponding pair of twin off diagonal elements: $a_{jj} > a_{ij}^2$. The leading diagonal is, thus, said to be strong throughout the reduction. Numerical details of the Choleski Decomposition are given in Appendix A.

**Forward and Back Substitution**

With the decomposition $X$ complete, the "system matrix", $A$, is then solved for. The final decomposition equation as stated in the previous section is

$$LL^TA^T = \bar{X} \quad (IV-20)$$
The solution of this equation for $A^T$ can be divided into two distinct stages:

Stage I: Forward Substitution which yields the intermediate matrix, $Y$

$$\tilde{L}Y = \tilde{X} \rightarrow Y$$  \hspace{1cm} (IV-21)

Stage II: Back Substitution which yields the transposed system matrix, $A^T$

$$\tilde{L}^T \tilde{A} = Y \rightarrow A^T$$  \hspace{1cm} (IV-22)

Both of these stages are very straightforward since $\tilde{L}$ and $\tilde{L}^T$ are lower and upper triangular matrices, respectively.

The numerical details and an example of Stages I and II are given in Appendix B.

The result of the previous sections is the transposed "system matrix". By simply transposing this result, the true "system matrix", $A$, is determined.
CHAPTER V

DETERMINING EIGENVALUES OF THE "SYSTEM MATRIX", A

The eigenvalues and eigenvectors are determined from the "system matrix", A. The algebraic eigenvalue problem is

$$A\{\psi\} = \lambda\{\psi\},$$  \hspace{1cm} (V-1)

where

$$\lambda = \text{eigenvalue of } A$$

$$\{\psi\} = \text{eigenvector corresponding to } \lambda.$$

There are $2\text{NDOF}$ eigenvalues which occur in complex conjugate pairs if the system is underdamped. The algebraic eigenvalue problem can be expressed in matrix form for $2\text{NDOF}$ eigenvalues and eigenvectors as

$$A\Psi = \Psi[-\lambda].$$  \hspace{1cm} (V-2)

The matrix of eigenvalues, $[-\lambda]$, is a diagonal matrix. The eigenvectors of A are the columns of $\Psi$.

If A is reduced to an upper triangular matrix, the eigenvalues are those elements on the diagonal. A proof of this is given in Appendix C. Note that if eigenvalues occur as complex conjugate pairs, they are obtained from the $(2 \times 2)$ sub-matrices on the diagonal.
There are several methods by which $A$ can be reduced to triangular form. The selection of any method is based on how the matrix to be reduced is classified. In our application, $A$ is classified as a general, fully populated matrix. The most efficient method to reduce this type of matrix is reduction to nearly triangular form using a non-iterative technique. An iterative technique is then used to complete the reduction and assure stability.

The nearly triangular form referred to above is the upper Hessenberg matrix. It is an upper triangular matrix with one non-zero sub-diagonal row.

The "system matrix" is reduced to upper Hessenberg form using similarity transformations which preserve the eigenvalues of the original matrix. A similarity transformation can be expressed as

$$ A^{(2)} = N^{-1} A^{(1)} N. $$

Matrices $A_1$ and $A_2$ are diagonally similar. The transformation matrix, $N$, is a non-singular matrix of the same order as $A$. The type of transformation used is an elementary stabilized transformation. This method requires fewer multiplications than other methods:
- orthogonal Givens transformations require $10 \frac{n^3}{3}$ multiplications

- orthogonal Householder transformations require $5 \frac{n^3}{3}$ multiplications

- elementary stabilized transformations require $5 \frac{n^3}{6}$ multiplications (Jennings 1977)

For a matrix of order $n$, $(n - 2)$ transformations are required to reduce the fully populated matrix to upper Hessenburg.

The reduction proceeds one column at a time. The $k$'th transformation,

$$A^{(k+1)} = N_k^{-1} A^{(k)} N_k,$$  \hspace{1cm} (V-5)

reduces the elements $k + 2$ through $n$ of column $k$: $a_{k+2,k} = a_{k+3,k} = \ldots a_{n,k} = 0$. The elementary stabilized matrix $N$ is shown below.

$$N_k = \begin{bmatrix}
1 & & & \\
1 & 0 & & \\
& \ddots & \ddots & \\
& & 1 & \\
& & a_{k+2,k+1} & \\
& & a_{k+3,k+1} & \\
0 & \ddots & & \\
& & & 1
\end{bmatrix}$$  \hspace{1cm} (V-6)
The term "elementary" specifies that the diagonal terms are of unit value. The term "stabilized" specifies that all transformation values, \( n_{ij} \), are equal to or less than unity. Partial pivoting is used to ensure that element \( a_{k+1,k} \) is not less than the elements below it:

\[
a_{k+1,k} \geq a_{k+2,k}, a_{k+3,k}, \ldots, a_{n,k}.
\]

With this satisfied, all the transformation values will be stabilized using equation (V-7).

Recalling equation (V-5), the \((k + 1)\)'th transformation can be written as

\[
A^{(k+2)} = N_{k+1}^{-1} A^{(k+1)} N_{k+1}
\]

\[
= N_{k+1}^{-1} N_{k}^{-1} A^{(k)} N_{k} N_{k+1}.
\]  

The final Hessenburg matrix is determined from \((n - 2)\) transformations.

\[
H = A^{(n-2)} = N_{n-2}^{-1} N_{2}^{-1} N_{1}^{-1} A^{(1)} N_{1} N_{2} \ldots N_{n-2}
\]

\[
= N_{1}^{-1} A^{(1)} N_{1}.
\]
where

\[ N = N_1 N_2 \ldots N_{n-2}. \]

\[
N = \begin{bmatrix}
1 \\
0 & 1 \\
0 & n_{3,2} & 1 \\
0 & n_{4,2} & n_{4,3} & 1 \\
\vdots & \vdots & \vdots & \vdots & \ddots \\
0 & n_{n,2} & n_{n,3} & \ldots & 1
\end{bmatrix}
\]  

(V-10)

The storage space for this procedure can be optimized by storing the transformation matrix, \( N \), in \( H \). The transformation element \( n_{i,k+1} \) replaces the element that it reduces to zero, \( a_{i,k} \). An example of the storage scheme is given below for a system of order 6.

\[
H = \begin{bmatrix}
h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} & h_{1,5} & h_{1,6} \\
h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} & h_{2,5} & h_{2,6} \\
& h_{3,2} & h_{3,3} & h_{3,4} & h_{3,5} & h_{3,6} \\
& & h_{4,2} & h_{4,3} & h_{4,4} & h_{4,5} & h_{4,6} \\
& & & h_{5,3} & h_{5,4} & h_{5,5} & h_{5,6} \\
& & & & h_{6,3} & h_{6,4} & h_{6,5} & h_{6,6} \\
\end{bmatrix}
\]  

(V-11)

Additional details and an example of the elementary stabilized transformations are given in Appendix C.
Special attention must be paid to how the eigenvectors of $A$ are affected by the transformation of $A$ to an upper Hessenburg matrix.

If $\{\overline{\psi}_j\}$ is an eigenvector of $H$,

$$H\{\overline{\psi}_j\} = \lambda_j \{\overline{\psi}_j\}. \quad (V-12)$$

Knowing that

$$H = N^{-1} A^{(1)} N, \quad (V-13)$$

equation (IV-12) can be rewritten as

$$N^{-1} A^{(1)} N \{\overline{\psi}_j\} = \lambda_j \{\overline{\psi}_j\}$$

$$A^{(1)} N \{\overline{\psi}_j\} = \lambda_j N \{\overline{\psi}_j\}. \quad (V-14)$$

Recall from equation (V-1) that

$$A^{(1)} \{\psi_j\} = \lambda_j \{\psi_j\}, \quad (V-15)$$

where $\{\psi_j\}$ is a right hand eigenvector of $A^{(1)}$. Comparing equations (V-14) and (V-15) shows that

$$N\{\overline{\psi}_j\} = \{\psi_j\}. \quad (V-16)$$

The eigenvectors of the original "system matrix", $A^{(1)}$, are $\{\psi_j\}$. These can be recovered from the eigenvectors of the upper Hessenburg matrix, $\{\overline{\psi}_j\}$, by pre-multiplication of the total transformation matrix, $N$ (Jennings 1977).
With the "system matrix" reduced to an upper Hessenburg matrix, an iterative procedure is now employed to reduce it to an upper triangular matrix. The eigenvalues are the identically those elements on the diagonal. This iterative procedure is the QR Algorithm. For complex systems, the Double QR Algorithm is recommended. When the eigenvalues occur as complex conjugate pairs, a true upper triangular matrix cannot be obtained without complex arithmetic. Rather, the system is allowed to converge to 2 x 2 sub-matrices on the diagonal. The characteristic equation of each 2 x 2 sub-matrix is then solved for the eigenvalues.

Because of the complexity of the QR Algorithm, it is not covered here in the main text of this paper. However, for those who wish to understand how it is used in this procedure, please refer to Appendix E where the QR and Double QR Algorithm are covered in detail.
CHAPTER VI
DETERMINING THE MODAL PARAMETERS

The eigenvalues of the "system matrix" are used to determine the modal frequencies and damping factors. The structure was modeled with 2NDOF computational degrees of freedom. Assuming the structure to be underdamped, there are NDOF complex conjugate pairs of eigenvalues. These correspond to NDOF modal frequency and damping factor pairs. In Chapter II, equations II-27 through II-33, the relationship between the complex eigenvalue, \( a + ib \), and the modal frequency, \( \omega_n \), and damping, \( \zeta \), was explained. For convenience, these are summarized below.

\[
(a + ib) = e^{(\alpha + i\beta)\Delta t_1}
\]

where:

\[
\alpha = \frac{1}{2\Delta t_1} \left[ \ln \left( a^2 + b^2 \right) \right]
\]

\[
\beta = \frac{1}{\Delta t_1} \left[ \tan^{-1} \left( \frac{b}{a} \right) \right].
\]

\((\alpha + i\beta)\) is the root of the characteristic equation.

Recall \( \Delta t_1 \) to be the time shift between the two response matrices, \( \Phi \) and \( \hat{\Phi} \). It is, therefore, the \( \Delta t \) in equation (II-22) that defined the eigenvalues of the "system matrix", \( A \); the \( k \)'th eigenvalue of \( A \),
\( e^{(\alpha + i\beta)k\Delta t} \), becomes \( e^{(\alpha + i\beta)k\Delta t_1} \). The modal parameters are determined from the root of the characteristic equation.

\[
\omega_d = \beta \\
\omega_n = (\alpha^2 + \beta^2)^{1/2} \\
\zeta = \alpha/\omega_n
\]

Because of the circular nature of the exponential function, the total expression for the damped natural frequency is:

\[
\omega_d = \beta = 1/\Delta t_1 [\tan^{-1}(b/a) + (0, \pi, 2\pi, ...)].
\]

The point at which the identified frequencies will fold (where \( \pi, 2\pi, \) etc. must be added) is called the "folding frequency", \( f_\pi \) (hertz) (Ibrahim and Pappa 1981a). Since there is no way of knowing the required amount of phase correction (\( \pi, 2\pi, \) ...), only the frequencies between 0 and \( f_\pi \) can be accurately identified. Hence, \( f_\pi \) must be the maximum frequency, \( f_{\text{max}} \), used in selecting the sample rate. The value of \( f_\pi \), and consequently \( f_{\text{max}} \), is simply \( 1/(2\Delta t_1) \). Knowing the desired maximum frequency \( f_{\text{max}} \), \( \Delta t_1 \) is chosen with the single restriction that:

\[
1/\Delta t_1 > 2 f_{\text{max}}.
\]
The recommended sample rate is four or five times $f_{\text{max}}$. This will allow some flexibility in selecting $\Delta t_1$.

The final modal parameter to be determined is the mode shape. Define $T$ as the pseudo upper triangular matrix that contains the eigenvalues on the diagonal. This is the final matrix generated when all the QR Algorithm iterations are complete. The term pseudo is used since the complex eigenvalues are contained in $2 \times 2$ sub-matrices on the diagonal. The eigenvector of $T$, corresponding to the $k$'th eigenvalue, is determined using a simple back substitution process.

$$T \{ \psi_k \} = \lambda_k \{ \psi_k \} \quad \text{(VI-6)}$$

If $\lambda_k$ is real, the above equation is used to determine the real eigenvector $\{ \psi_k \}$. However, if the eigenvalue is one of a complex conjugate pair, $\lambda_k$ and $\lambda_{k-1}$, a pair of equations similar to equation (VI-6) are used to determine the eigenvectors $\{ \psi_k \}$ and $\{ \psi_{k-1} \}$.

$$T \{ p_k + iq_k \} = (a + ib)_k \{ p_k + iq_k \} \quad \text{(VI-7a)}$$

$$T \{ p_k - iq_k \} = (a - ib)_k \{ p_k - iq_k \} \quad \text{(VI-7b)}$$

where

$$\{ \psi_k \} = \{ p_k + iq_k \}$$

$$\{ \psi_{k-1} \} = \{ p_k - iq_k \}.$$
Since $\{\psi_k\}$ and $\{\overline{\psi}_{k-1}\}$ are complex conjugates, only one need be calculated and stored, say $\{\psi_k\}$. The real portion could be stored in $\{\psi_{k-1}\}$ and the imaginary portion in $\{\overline{\psi}_k\}$.

\[ \{p_k\} + \{\overline{\psi}_{k-1}\} \quad (VI-8a) \]
\[ \{q_k\} + \{\psi_k\} \quad (VI-8b) \]

Combining equations (VI-7a), (VI-8a) and (VI-8b) yields the final complex equation to be solved.

\[ T \{\psi_{k-1} + \overline{\psi}_k\} = (a + ib)_k \{\psi_{k-1} + \overline{\psi}_k\} \quad (VI-9) \]

The back substitution procedure used to solve equations (VI-6) and (VI-9) for real and complex eigenvectors, respectively, is given in Appendix F.

The eigenvector matrix $\Psi$ corresponds to the pseudo upper triangular matrix, $T$. However, mode shapes are the eigenvectors of the original "system matrix", $A$. The pseudo upper triangular matrix was derived from the "system matrix" through a series of similarity transformations. In Appendix C, it was stated that two similar matrices will have the same eigenvalues, but different eigenvectors. However, the eigenvectors are related through the total transformation matrix. Since matrices $A$ and $T$ are similar, their eigenvalues are the same. For the $k$'th eigenvalue,
All the similarity transformations generated in reducing the "system matrix" to a pseudo upper triangular matrix are combined into a single transformation matrix, \( N \). The system mode shape, \( \{ \psi_k \} \), is then obtained by pre-multiplying \( \{ \bar{\psi}_k \} \) by the total transformation matrix, \( N \).

\[
\{ \psi_k \} = N \{ \bar{\psi}_k \}
\]  \( \text{(VI-12)} \)

To determine the total transformation matrix, \( N \), recall all the similarity transforms and permutations used in reducing \( A \).

1. Elementary stabilized transformations, \( N_i \), used in the reduction to upper Hessenburg.
2. Permutation matrices, \( I_i \), used in the reduction to upper Hessenburg.
3. Elementary Hermitian matrices, \( P_i \), used in each iteration of the QR Algorithm.

The transformations of (1) and (2) may be summarized as

\[
H = (N_{n-2}^{-1} I_{n-2}^{-1} \ldots N_2^{-1} I_2^{-1} N_1^{-1} I_1^{-1}) A(I_1 N_1 I_2 N_2 \ldots I_{n-2} N_{n-2})
\]

\[
= N_H^{-1} A N_H
\]

\( \text{(VI-13)} \) \( \text{(VI-14)} \)
The transformations of (3), assuming \( k \) iterations, may be summarized as

\[
T = (P)_k \cdots (P)_2 (P)_1 H (P)_1^T (P)_2^T \cdots (P)_k^T ,
\]  

(VI-15)

where

\[
(P)_1 = P_{n-2} \ldots P_2 P_1 \quad \text{(1st iteration)}
\]

\[
(P)_2 = P_{n-2} \ldots P_2 P_1 \quad \text{(2nd iteration)}
\]

\[
\vdots
\]

\[
(P)_k = P_2 P_1 \quad \text{(k'th iteration on the truncated matrix)}.
\]

Combining equations (VI-14) and (VI-15), the total transformation matrix becomes

\[
T = (P)_k \cdots (P)_2 (P)_1 N_{H}^{-1} A N_{H} (P)_1^T (P)_2^T \cdots (P)_k^T
\]  

(VI-16)

\[
= N_{H}^{-1} A N ,
\]  

(VI-17)

where

\[
N = N_{H} (P)_1^T (P)_2^T \cdots (P)_k^T
\]

\[
= I_{1} N_{1} I_{2} N_{2} \cdots I_{n-2} N_{n-2} (P)_1^T (P)_2^T \cdots (P)_k^T
\]  

(VI-18)
The mode shape matrix, \( \Psi \), can now be determined by pre-multiplying \( \overline{\Psi} \) by the total transformation matrix, \( N \).

\[
\Psi = N \overline{\Psi}
\]  
(VI-19)

All the modal parameters have now been determined.
CHAPTER VII
RESULTS AND FUTURE RESEARCH

The optimized ITD code was implemented on a DEC PDP-11/34 mini-computer. The code, labeled MINI-ITD, was tested using synthesized free response data of a beam. The beam was modeled with clamped-clamped boundary conditions. Free response functions were generated for a 3 DOF system using the damped cosine normal mode summation equation (Ibrahim 1981a).

\[
\{x(t_j)\} = \sum_{k=1}^{3} A_k \{\psi_k\} e^{-\sigma_k t_j} \cos(\omega_{d,k} t_j) \quad (VII-1)
\]

where

\[
\{x(t_j)\} = \text{calculated response vector at time } t
\]
\[
A_k = \text{amplitude coefficient of the } k\text{'th mode}
\]
\[
\{\psi_k\} = \text{normalized characteristic function describing the deflection of the } k\text{'th mode}
\]
\[
\sigma_k = \text{real portion of the } k\text{'th root of the third order characteristic equation}
\]
\[
(\omega_{d,k}) = \text{imaginary portion of the } k\text{'th root of the third order characteristic equation}
\]

Free response functions for the beam, of unit length, were generated at locations

\[x_1 = 0.24, \quad x_2 = 0.48, \quad x_3 = 0.72.\]
Normalized characteristic functions and amplitude coefficients for the above equation were obtained from Thomson (1981).

\[ A_1 = 1.58271 \]
\[ \{ \psi_1 \} = \{ 0.51468, 1.00000, 0.63528 \}_T \quad (VII-2) \]

\[ A_2 = 1.50485 \]
\[ \{ \psi_2 \} = \{ 0.93700, 0.15118, -1.00000 \}_T \quad (VII-3) \]

\[ A_3 = 1.50782 \]
\[ \{ \psi_3 \} = \{ 1.00000, -0.95945, 0.77442 \}_T \quad (VII-4) \]

The natural frequencies, \( \omega_n \), and damping factor, \( \zeta \), were arbitrarily chosen. The corresponding damped natural frequency, \( \omega_d \), was calculated using equation (II-28b).

Initially, the MINI-ITD was tested with noise free data. Three cases were demonstrated:

- low modal density and low damping
- low modal density and high damping
- highly coupled modes.

The parameters chosen and the results obtained are shown in Tables 1, 2 and 3 for the three cases, respectively. In all three cases, the user-selectable algorithm constants were as follows:

- NMEAS = 3
- NDOF = 3
- NCOL = 30
- N1 = 3
- N2 = 8
- N3 = 4.
### TABLE 1

RESULTS OF THE MINI-ITD FOR CASE I*

<table>
<thead>
<tr>
<th>CASE I: 3 DOF System with Low Modal Density and Low Damping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural Frequency (Hz)</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Mode 1</td>
</tr>
<tr>
<td>Mode 2</td>
</tr>
<tr>
<td>Mode 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MCF: Magnitude (%)</th>
<th>MCF: Phase (Degrees)</th>
<th>OAMCF</th>
<th>Identification Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1</td>
<td>90.14</td>
<td>2.50</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>94.88</td>
<td>1.88</td>
<td></td>
</tr>
<tr>
<td></td>
<td>93.62</td>
<td>1.75</td>
<td></td>
</tr>
<tr>
<td>Mode 2</td>
<td>100.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>100.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Mode 3</td>
<td>100.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>100.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100.00</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>

* User-selectable algorithm constants used: NMEAS = 3, NDOF = 3, NCOL = 30, N1 = 3, N2 = 8, and N3 = 4.
### TABLE 2

RESULTS OF THE MINI-ITD FOR CASE II*

<table>
<thead>
<tr>
<th>Natural Frequency (Hz)</th>
<th>Damping (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[Input]</td>
</tr>
<tr>
<td>Mode 1</td>
<td>25.0</td>
</tr>
<tr>
<td>Mode 2</td>
<td>50.0</td>
</tr>
<tr>
<td>Mode 3</td>
<td>75.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MCF: Magnitude (%)</th>
<th>MCF: Phase (Degrees)</th>
<th>OAMCF</th>
<th>Identification Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1</td>
<td>91.44</td>
<td>2.12</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>95.46</td>
<td>1.52</td>
<td></td>
</tr>
<tr>
<td></td>
<td>94.46</td>
<td>1.46</td>
<td></td>
</tr>
<tr>
<td>Mode 2</td>
<td>87.56</td>
<td>3.76</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>67.96</td>
<td>10.53</td>
<td></td>
</tr>
<tr>
<td></td>
<td>92.67</td>
<td>2.08</td>
<td></td>
</tr>
<tr>
<td>Mode 3</td>
<td>100.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>100.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100.00</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>

* User-selectable algorithm constants used: NMEAS = 3, NDOF = 3, NCOL = 30, N1 = 3, N2 = 8, and N3 = 4.
TABLE 3

RESULTS OF THE MINI-ITD FOR CASE III*

CASE III: 3 DOF System with Highly Coupled Modes

<table>
<thead>
<tr>
<th>Natural Frequency (Hz)</th>
<th>Damping (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Input</td>
</tr>
<tr>
<td>Mode 1</td>
<td>25.0</td>
</tr>
<tr>
<td>Mode 2</td>
<td>25.5</td>
</tr>
<tr>
<td>Mode 3</td>
<td>26.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MCF: Magnitude (%)</th>
<th>MCF: Phase (Degrees)</th>
<th>OAMCF</th>
<th>Identification Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1</td>
<td>99.97</td>
<td>0.02</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>99.99</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>99.99</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Mode 2</td>
<td>99.95</td>
<td>0.02</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>99.90</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>99.92</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td>Mode 3</td>
<td>99.63</td>
<td>0.40</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>99.58</td>
<td>0.39</td>
<td></td>
</tr>
<tr>
<td></td>
<td>99.49</td>
<td>0.16</td>
<td></td>
</tr>
</tbody>
</table>

In cases I and II, the error in the modal parameters identified was negligible. In case III, with highly coupled modes, the error in the natural frequency identification was less than 0.25%, and the damping factor error was less than 5%. The damping factor identification is very sensitive to error. This becomes quite evident in the following test cases in which noise is added to the synthesized free response functions.

The mode shape extraction for all real modes has been verified. The extraction of the mode shape corresponding to the complex mode identified first has also been verified. This is shown in Tables 1 and 2, where mode 3 has an OAMCF of 1.00. In case III, mode 1 is identified first. The excellent results in MCF's and the OAMCF of 1.00 verify this mode shape to be correct.

The MINI-ITD was also tested using data to which varying degrees of noise were added. The noise was added as a percent of the overall rms amplitude of each free response function.

\[ \tilde{x}_i(t_j) = x_1(t_j) + \text{Noise}_j = x_1(t_j) + [(\%)(\bar{x}_i)(2r-r)] \]  \hspace{1cm} (VII-5)

where

\[
\begin{align*}
\% & = \text{percentage of noise} \\
\bar{x}_i & = \text{overall rms amplitude of the } i \text{'th free response function} \\
r & = \text{random number between 0 and 1.}
\end{align*}
\]

The factor \((2r-r)_j\) is used to give a random number between -1 and 1; thus, assuring the noise added had a zero mean.
A 2 DOF system was used in testing the MINI-ITD with data to which noise was added. Equation (VII-1) was used, with \( k = 2 \), to generate two free response functions without noise. Beam locations \( x_1 = 0.24 \) and \( x_2 = 0.48 \) were used with the amplitude coefficients and normalized characteristic function of equations (VII-2) and (VII-3). Noise was then added to the free response function using equation (VII-5). Five cases were tested: 0%, 1%, 2%, 5% and 10% noise. The results of these cases are shown in Table 4. In all cases, the error in identifying the natural frequencies is less than 1.00%. The damping factor identification was not as successful. Greater than 2% noise in the data made the extraction of the damping coefficients unreliable. Increasing the size of the math model will improve these results significantly. The noisy data was modeled with five computational DOF: \( N\text{MEAS} = 2 \) and \( N\text{DOF} = 5 \). Three pseudo measurements were generated to fill the response matrices. By increasing the number of computational DOF, the noise will be modeled more correctly and consequently, so will the true structural response. Unfortunately, no data is available yet to show this since the current file structure of the MINI-ITD limits the possible size of the math model; the number of computational DOF cannot be greater than five when using \( \text{NCOL} \) greater than thirty.

The extraction of the mode shapes for the 50 Hz mode are correct. This is verified by the OAMCF of 1.00. The phase portion of the MCF is more sensitive to noise than the magnitude portion.
TABLE 4

MODAL PARAMETER IDENTIFICATION FOR A 2 DOF SYSTEM WITH VARYING DEGREES OF NOISE*

<table>
<thead>
<tr>
<th>Noise (%)</th>
<th>Frequency (Hz)</th>
<th>Damping (%)</th>
<th>MCF: Mag (%)</th>
<th>MCF: Phase (Degrees)</th>
<th>OAMCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>25.00</td>
<td>5.00</td>
<td>83.4</td>
<td>0.00</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>(0.00)**</td>
<td>(0.00)</td>
<td>95.1</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>25.02</td>
<td>5.15</td>
<td>99.3</td>
<td>0.55</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>(0.00)</td>
<td>(3.00)</td>
<td>99.8</td>
<td>0.39</td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>25.02</td>
<td>5.53</td>
<td>94.0</td>
<td>1.00</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>(0.00)</td>
<td>(3.00)</td>
<td>99.8</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>5.0</td>
<td>24.94</td>
<td>7.92</td>
<td>93.1</td>
<td>1.39</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>(0.24)</td>
<td>(58.4)</td>
<td>99.6</td>
<td>2.58</td>
<td></td>
</tr>
<tr>
<td>10.0</td>
<td>24.61</td>
<td>15.32</td>
<td>92.7</td>
<td>5.47</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>(1.56)</td>
<td>(206.)</td>
<td>98.4</td>
<td>5.98</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Noise (%)</th>
<th>Frequency (Hz)</th>
<th>Damping (%)</th>
<th>MCF: Mag (%)</th>
<th>MCF: Phase (Degrees)</th>
<th>OAMCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>50.00</td>
<td>5.00</td>
<td>100.0</td>
<td>0.00</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>(0.00)</td>
<td>(0.00)</td>
<td>100.0</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>49.89</td>
<td>5.08</td>
<td>99.8</td>
<td>0.47</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>(0.22)</td>
<td>(1.60)</td>
<td>98.3</td>
<td>2.65</td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>49.79</td>
<td>5.46</td>
<td>99.2</td>
<td>0.87</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>(0.42)</td>
<td>(9.20)</td>
<td>97.3</td>
<td>4.80</td>
<td></td>
</tr>
<tr>
<td>5.0</td>
<td>49.51</td>
<td>8.46</td>
<td>95.4</td>
<td>1.74</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>(0.98)</td>
<td>(69.2)</td>
<td>96.0</td>
<td>9.34</td>
<td></td>
</tr>
<tr>
<td>10.0</td>
<td>49.52</td>
<td>19.39</td>
<td>83.5</td>
<td>2.50</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>(0.96)</td>
<td>(288.)</td>
<td>94.5</td>
<td>14.00</td>
<td></td>
</tr>
</tbody>
</table>

* User-selectable algorithm constants used: NMEAS = 2, NDOF = 5, NCOL = 30, N1 = 3, N2 = 8 and N3 = 4.

** The numbers in parentheses denote the percent error between the output value and the input value (the value used in synthesizing the free response functions).
Not until 10% noise is added to the data are the MCF acceptance limits exceeded:

magnitude $> 95.0\%$ and phase $< 10$. 

Immediate work on the MINI-ITD will include:
- determining all mode shapes correctly
- extracting the damping factor in a way less sensitive to noise (Ibrahim 1981a).

Future work on the MINI-ITD will be concentrated on optimizing the file structure and data storage. In this way, the allowable computational DOF will be significantly increased. This is a necessary requirement to make the damping factor extraction sufficiently accurate.

Once all modal parameters are identified with sufficient accuracy, the MINI-ITD results will be compared with those of conventional modal analysis software packages. Two particular cases will be tested:
- low modal density with high damping
- highly coupled modes.

The results will have significant implications since the modal parameters of these cases typically cannot be identified using conventional FRF curve fitting routines.

Assuming the MINI-ITD performs favorably with synthesized data, the next step in its verification will be to perform a modal analysis on a test structure. At this point, extensive work will
be required in generating the necessary file structure to make the MINI-ITD compatible with existing data acquisition systems. Once this is accomplished, the MINI-ITD will be operational!

The MINI-ITD has tremendous potential as a method to perform modal analysis. As the author, I stress that it is not meant as a replacement for existing frequency response methods, nor is it a cure-all. It is, however, another tool expected to give answers which are now very difficult to obtain: the natural frequencies, damping factors and mode shapes for highly damped and highly coupled modes. Dr. Samir Ibrahim, of Old Dominion University, Virginia, developed the ITD Algorithm which is an accurate numerical method. The ITD, when used on main-frame computers, can characterize the dynamic response of very large structures (greater than 30 DOF). However, many environmental test labs seldom require such extensive analysis capabilities. In view of this, I wanted to make the ITD available for those instants, when a mini-computer could satisfy the analysis needs. Through the MINI-ITD, I hope to make the ITD Algorithm an option in performing modal analysis in any environmental test lab.
A Choleski Decomposition for a real symmetric matrix, $X$, is defined as

$$X = LDL^T,$$  \hspace{1cm} (A-1)

where

$L = \text{lower triangular matrix with unit diagonal}$

$D = \text{diagonal matrix (Jennings 1977)}.$

This is a symmetric decomposition. If $X$ is also positive definite, the elements of $D$ are also positive definite. To prove this, let \{y\} be any vector, with \{x\} related to it by

$$L^T\{x\} = \{y\}. \hspace{1cm} (A-2)$$

Since $L^T$ is non-singular (unit diagonal), \{x\} must exist, and

$$\{y\}^T D \{y\} = \{x\}^T L D L^T \{x\}, \hspace{1cm} (A-3)$$

$$= \{x\}^T X \{x\}.$$ 

Since $X$ is positive definite,

$$\{x\}^T X \{x\} > 0.$$
Therefore, D must also be positive definite. The diagonal elements of D are its eigenvalues; \(d_{ii} > 0\). A direct result of this is that \(\sqrt{d_{ii}}\) is real and D may be conveniently expressed as

\[
D = D^{1/2} \cdot D^{1/2}.
\]  

Substituting this into equation (A-1) yields

\[
\begin{align*}
\bar{X} &= L \cdot D^{1/2} \cdot D^{1/2} \cdot L^T \\
\bar{X} &= \tilde{L} \cdot \tilde{L}^T,
\end{align*}
\]  

where

\[
\tilde{L} = LD^{1/2}.
\]  

Hence, the symmetric positive definite matrix, \(\bar{X}\), is decomposed into a single lower triangular matrix and its transpose. For a matrix of order \(n = 4\),

\[
\bar{X} = \begin{bmatrix}
1_{11} & 0 & 1_{11} & 1_{11} \\
1_{21} & 1_{22} & 1_{22} & 1_{22} \\
1_{31} & 1_{32} & 1_{33} & 1_{33} \\
1_{41} & 1_{42} & 1_{43} & 1_{44}
\end{bmatrix} \begin{bmatrix}
1_{11} & 1_{22} & 1_{33} & 1_{44} \\
1_{22} & 1_{33} & 1_{44} \\
1_{33} & 1_{44} \\
0 & 1_{44}
\end{bmatrix}. 
\]  

General formulas:

\[
\begin{align*}
1_{ii} &= \left(\frac{\bar{X}_{ii}}{1_{ii}} - \sum_{k=1}^{i-1} 1_{ik}^2 \right)^{1/2},
\end{align*}
\]
\[ l_{ij} = (x_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk})/l_{jj}, \quad (j < i) \quad (A-9) \]

Progression of Solution:

1. The original matrix, \( X \), is a symmetric positive definite matrix of order \( n \). Only the lower triangular elements of \( X \) are stored.

2. The solution proceeds by rows from row \( i = 1 \) to \( n \).

3. The elements in row \( i \) are solved for in the order of column \( j = 1 \) to \((i-1)\). The final element determined in row \( i \) is element \( l_{ii} \).

4. As element \( l_{ij} \) or \( l_{ii} \) is determined, it replaces the original \( x_{ij} \) or \( x_{ii} \), respectively, in the matrix store.

5. The resulting matrix, \( L \), is a lower triangular matrix of order \( n \).

6. To demonstrate, the elements of the fourth row of \( L \) as shown in equation (A-7) are calculated below.

\[
\begin{align*}
l_{41} &= \frac{x_{41}}{l_{11}} \\
l_{42} &= \frac{x_{42} - l_{41} l_{22}}{l_{22}} \\
l_{43} &= \frac{x_{43} - l_{41} l_{31} - l_{42} l_{32}}{l_{33}} \\
l_{44} &= \left(\frac{x_{44} - l_{41}^2 - l_{42}^2 - l_{43}^2}{l_{33}}\right)^{\frac{1}{2}}
\end{align*}
\]
APPENDIX B

GAUSSIAN ELIMINATION FORWARD AND BACK SUBSTITUTION

Beginning with the set of simultaneous linear equations \( X \Lambda^T = \hat{X} \), it is necessary to solve for the transposed system matrix. The matrix \( \hat{X} \) is decomposed into \( \tilde{L} \tilde{L}^T \) using a Choleski decomposition. \( L \) is a lower triangular matrix. The equation to be solved is now

\[
\tilde{L} \tilde{L}^T \Lambda^T = \hat{X}. \tag{B-1}
\]

The solution procedure is broken into two stages:

(I) Gaussian Elimination Forward Substitution yielding an intermediate matrix \( Y \).

\[
\tilde{L} \tilde{Y} = \hat{X} \quad \rightarrow \quad Y \tag{B-2}
\]

(II) Gaussian Elimination Back Substitution yielding the transposed system matrix.

\[
\tilde{L}^T \Lambda^T = Y \quad \rightarrow \quad \Lambda^T \tag{B-3}
\]

Forward Substitution

Because \( \tilde{L} \) is a triangular matrix, the forward substitution is easily performed. This stage is equivalent to the right-hand reduction stage of Gaussian elimination. For this reason, it is usually referred to as Gaussian elimination forward substitution. The three matrices involved are square and of order \( n \).
Below is an example using a fourth order set of equations.

\[ \tilde{Y} = \tilde{X} \]

\[
\begin{bmatrix}
l_{11} & l_{12} & 1 & 13 & 1 & 14 \\
l_{21} & l_{22} & 1 & 23 & 1 & 24 \\
l_{31} & l_{32} & l_{33} & 1 & 34 \\
l_{41} & l_{42} & l_{43} & l_{44} & 1
\end{bmatrix}
\begin{bmatrix}
y_{11} & y_{12} & y_{13} & y_{14} \\
y_{21} & y_{22} & y_{23} & y_{24} \\
y_{31} & y_{32} & y_{33} & y_{34} \\
y_{41} & y_{42} & y_{43} & y_{44}
\end{bmatrix}
= 
\begin{bmatrix}
x_{11} & x_{12} & x_{13} & x_{14} \\
x_{21} & x_{22} & x_{23} & x_{24} \\
x_{31} & x_{32} & x_{33} & x_{34} \\
x_{41} & x_{42} & x_{43} & x_{44}
\end{bmatrix}
\]

(B-4)

The matrix \( \tilde{Y} \) is determined one row at a time beginning with the first row and ending with the \( n \)'th row.

Row 1: \( y_{1j} = \frac{\hat{x}_{1j}}{l_{1j}} \)

\[
l_{11} y_{11} = \hat{x}_{11} + y_{11} = \frac{\hat{x}_{11}}{l_{11}}
\]

\[
l_{11} y_{12} = \hat{x}_{12} + y_{12} = \frac{\hat{x}_{12}}{l_{11}}
\]

\[
l_{11} y_{13} = \hat{x}_{13} + y_{13} = \frac{\hat{x}_{13}}{l_{11}}
\]

\[
l_{11} y_{14} = \hat{x}_{14} + y_{14} = \frac{\hat{x}_{14}}{l_{11}}
\]

Row 2: \( y_{2j} = \frac{(x_{2j} - l_{21} y_{1j})}{l_{22}} \)

\[
l_{21} y_{11} + l_{22} y_{21} = \hat{x}_{21} + y_{21} = \frac{(\hat{x}_{21} - l_{21} y_{11})}{l_{22}}
\]

\[
y_{22} = \frac{(\hat{x}_{22} - l_{21} y_{12})}{l_{22}}
\]
\[ y_{23} = \left( \hat{x}_{23} - 1_{21} y_{13} \right) / 1_{22} \]
\[ y_{24} = \left( \hat{x}_{24} - 1_{21} y_{14} \right) / 1_{22} \]

Row 3:
\[ y_{3j} = \left[ x_{3j} - \left( 1_{31} y_{1j} + 1_{32} y_{2j} \right) \right] / 1_{33} \]
\[ y_{31} = \left[ x_{31} - \left( 1_{31} y_{11} + 1_{32} y_{21} \right) \right] / 1_{33} \]
\[ y_{32} = \left[ x_{32} - \left( 1_{31} y_{12} + 1_{32} y_{22} \right) \right] / 1_{33} \]
\[ y_{33} = \left[ x_{33} - \left( 1_{31} y_{13} + 1_{32} y_{23} \right) \right] / 1_{33} \]
\[ y_{34} = \left[ x_{34} - \left( 1_{31} y_{14} + 1_{32} y_{24} \right) \right] / 1_{33} \]

Row 4:
\[ y_{4j} = \left[ x_{4j} - \left( 1_{41} y_{1j} + 1_{42} y_{2j} + 1_{43} y_{3j} \right) \right] / 1_{44} \]
\[ y_{41} = \left[ x_{41} - \left( 1_{41} y_{11} + 1_{42} y_{21} + 1_{43} y_{31} \right) \right] / 1_{44} \]
\[ y_{42} = \left[ x_{42} - \left( 1_{41} y_{12} + 1_{42} y_{22} + 1_{43} y_{32} \right) \right] / 1_{44} \]
\[ y_{43} = \left[ x_{43} - \left( 1_{41} y_{13} + 1_{42} y_{23} + 1_{43} y_{33} \right) \right] / 1_{44} \]
\[ y_{44} = \left[ x_{44} - \left( 1_{41} y_{14} + 1_{42} y_{24} + 1_{43} y_{34} \right) \right] / 1_{44} \]

The above equations can be summarized as
\[ y_{ij} = \hat{x}_{ij} - \sum_{k=1}^{i-1} 1_{jk} y_{kj} / 1_{ii} . \] (B-5)

This equation may be programmed in the following way, where \( Y \) is determined one row at a time.
DO 10 I = 1,N
    DO 20 J = 1,N
        SUM = 0.0
        DO 30 K = 1, I - 1
            SUM = SUM + L(I,K) * Y(K,J)
        30 CONTINUE
        Y(I,J) = (X(I,J) - SUM) / L(I,I)
    20 CONTINUE
10 CONTINUE

This procedure can be optimized by storing the intermediate matrix Y in \( \hat{X} \). Once \( y_{ij} \) is determined, \( x_{ij} \) is not used again. However, \( y_{ij} \) is used repeatedly in calculating the remainder of Y. Therefore, \( y_{ij} \) is stored in \( \hat{x}_{ij} \) and the additional \( n^2 \) storage spaces for storing a separate Y is avoided.

**Back Substitution**

This stage is equivalent to the back substitution process of Gaussian elimination. For this reason, it is usually referred to as Gaussian elimination back substitution. The equation \( \hat{L}^T \hat{A}^T = \hat{Y} \) is easily solved since \( \hat{L}^T \) is an upper triangular matrix. Recall that the intermediate matrix Y is actually stored in \( \hat{X} \).

The previous fourth order set of equations is used here again to demonstrate the second stage.
\[ \mathbf{L}^T \mathbf{A}^T = \mathbf{Y} = \mathbf{X} \]

\[
\begin{bmatrix}
1_{11} & 1_{21} & 1_{31} & 1_{41} \\
1_{22} & 1_{32} & 1_{42} \\
1_{33} & 1_{43} \\
1_{44}
\end{bmatrix}
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} \\
a_{21} & a_{22} & a_{23} & a_{24} \\
a_{31} & a_{32} & a_{33} & a_{34} \\
a_{41} & a_{42} & a_{43} & a_{44}
\end{bmatrix}
= 
\begin{bmatrix}
y_{11} & y_{12} & y_{13} & y_{14} \\
y_{21} & y_{22} & y_{23} & y_{24} \\
y_{31} & y_{32} & y_{33} & y_{34} \\
y_{41} & y_{42} & y_{43} & y_{44}
\end{bmatrix}
\]

Note that the indices of \( \mathbf{L} \) are reversed to indicate the \( \mathbf{L}^T \) is being used. The solution proceeds in the reverse order of the forward substitution. The \( n \)'th row of \( \mathbf{A}^T \) is determined first and the first row is found last.

Row 4: \( a_{4j} = y_{4j}/1_{44} = \hat{x}_{4j}/1_{44} \)

\[
a_{41} = y_{41}/1_{44} = \hat{x}_{41}/1_{44}
\]

\[
a_{42} = \hat{x}_{42}/1_{44}
\]

\[
a_{43} = \hat{x}_{43}/1_{44}
\]

\[
a_{44} = \hat{x}_{44}/1_{44}
\]

Row 3: \( a_{3j} = (x_{3j} - 1_{43} a_{4j})/1_{33} \)

\[
a_{31} = (x_{31} - 1_{43} a_{41})/1_{33}
\]

\[
a_{32} = (x_{32} - 1_{43} a_{42})/1_{33}
\]
\[ a_{33} = \frac{x_{33} - a_{43}}{1_{33}} \]
\[ a_{34} = \frac{x_{34} - a_{44}}{1_{33}} \]

Row 2:  
\[ a_{2j} = \frac{x_{2j} - (1_{32} a_{3j} + 1_{42} a_{4j})}{1_{22}} \]
\[ a_{21} = \frac{x_{21} - (1_{32} a_{31} + 1_{42} a_{41})}{1_{22}} \]
\[ a_{22} = \frac{x_{22} - (1_{32} a_{32} + 1_{42} a_{42})}{1_{22}} \]
\[ a_{23} = \frac{x_{23} - (1_{32} a_{33} + 1_{42} a_{43})}{1_{22}} \]
\[ a_{24} = \frac{x_{24} - (1_{32} a_{34} + 1_{42} a_{44})}{1_{22}} \]

Row 1:  
\[ a_{1j} = \frac{x_{1j} - (1_{21} a_{2j} + 1_{31} a_{3j} + 1_{41} a_{4j})}{1_{11}} \]
\[ a_{11} = \frac{x_{11} - (1_{21} a_{21} + 1_{31} a_{31} + 1_{41} a_{41})}{1_{11}} \]
\[ a_{12} = \frac{x_{12} - (1_{21} a_{22} + 1_{31} a_{32} + 1_{41} a_{42})}{1_{11}} \]
\[ a_{13} = \frac{x_{13} - (1_{21} a_{23} + 1_{31} a_{33} + 1_{41} a_{43})}{1_{11}} \]
\[ a_{14} = \frac{x_{14} - (1_{21} a_{24} + 1_{31} a_{34} + 1_{41} a_{44})}{1_{11}} \]

The above equations can be summarized as:
\[ a_{ij} = \frac{x_{ij} - \sum_{k=1+1}^{n} 1_{k1} a_{kj}}{1_{ii}} \]
and programmed in the following manner:

```
DO 10 I = N, 1, -1
  DO 20 K = I, N
    WORK(K) = L(K,I)
  DO 30 J = 1, N
    SUM = SUM + 0.00D + 00
  DO 40 K = I + 1, N
30       SUM = SUM + WORK(K) * A(J,K)
40       A(J,I) = (X(I,J) - SUM)/WORK(I)
10      CONTINUE
```

The result is the "system matrix", A. The above procedure is optimized in a manner similar to the forward substitution. The i'th column of L (the i'th row of $L^T$) is stored in a work vector of length n. Only elements from i to n need be stored. As the elements in row i of $A^T$ are calculated, they may overwrite the i'th column of L. Element $a_{ij}$ of $A^T$ is stored in location(j,i). This is the same as $a_{ij}$ of the true "system matrix", A. The transpose of the "system matrix" is automatically performed via the storage scheme. Recalling that the matrix L was stored in $X$, A is also stored in $X$.

In summary, the Choleski decomposition forward substitution and back substitution are all performed in the two matrices $X$ and
\( \hat{X} \). The matrices are both square matrices of order \( n \). The Choleski decomposition is performed, and the results stored in the lower triangle of \( \hat{X} \). The matrices \( \hat{X} (\hat{L}) \) and \( \hat{X} \) are used to perform the forward substitution. The result of this stage is the intermediate matrix \( Y \) which is stored in \( \hat{X} \). The back substitution is performed with the matrices \( \hat{X} (\hat{L}^T) \) and \( \hat{X} (Y) \). The resulting "system matrix" is stored in \( \hat{X} \), overwriting the elements of \( \hat{L} \).
APPENDIX C

PROOF THAT ELEMENTS ON THE DIAGONAL OF AN UPPER TRIANGULAR MATRIX ARE THE EIGENVALUES

The determinant of an upper triangular matrix is the product of the diagonal elements:

$$\det A = |A| = \prod_{i=1}^{n} a_{ii} \quad \text{(C-1)}$$

The determinant of a diagonal matrix is the product of the diagonal elements:

$$\det \lambda = \prod_{i=1}^{n} \lambda_i \quad \text{(C-2)}$$

Recall the algebraic eigenvalue equation for the i'th eigenvalue, \( \lambda \), and its corresponding eigenvector \( \{ \psi \} \) to be

$$A \{ \psi \} = \lambda \{ \psi \} \quad \text{(C-3)}$$

or

$$[A - \lambda I] \{ \psi \} = 0 \quad \text{(C-4)}$$

If \( A \) is upper triangular, the determinant of \([A - \lambda I]\) is

$$|A - \lambda I| = (a_{11} - \lambda)(a_{22} - \lambda) \ldots (a_{nn} - \lambda) \quad \text{(C-5)}$$
For any non-trivial solution to the algebraic eigenvalue problem, the determinant above must be zero.

\[(a_{11} - \lambda)(a_{22} - \lambda) \ldots (a_{nn} - \lambda) = 0\]  \hspace{1cm} (C-6)

The characteristic equation for \(n\) eigenvalues is

\[(\lambda - \lambda_1)(\lambda - \lambda_2) \ldots (\lambda - \lambda_n) = 0\]  \hspace{1cm} (C-7)

By comparing equations (C-6) and (C-7), it becomes obvious that \(a_{ii}\) must equal \(\lambda_i\). Therefore, if \(A\) is reduced to an upper triangular matrix, the diagonal elements are the eigenvalues of \(A\) (Wilkinson 1965).

It should be noted that if the eigenvalues are complex conjugate pairs, the matrix \(A\) cannot be fully reduced to upper triangular without using complex arithmetic. The eigenvalues in this case are obtained from \(2 \times 2\) sub-matrices on the diagonal. If left in this format, \(A\) will remain real, and complex arithmetic is avoided.
APPENDIX D
REDUCTION TO UPPER HESSENBURG

Two matrices, \( A \) and \( B \), are similar if and only if:

\[
A = N^{-1} B N
\]

where \( N \) is a non-singular matrix. The reduction of a matrix can be performed using similarity transformations.

\[
A^{(k+1)} = N_k^{-1} A^{(k)} N_k
\]

If \( N_k \) is non-singular, the matrices \( A^{(k)} \) and \( A^{(k+1)} \) are similar. \( N_k \) is the transformation matrix. Similarity transformations do not alter the eigenvalues of a matrix, but do alter its eigenvectors.

The eigenvectors of the original matrix, \( A^{(k)} \), are obtained by pre-multiplying the eigenvectors of \( A^{(k+1)} \) by \( N_k \).

In our application, similarity transformations are used to reduce the "system matrix," \( A \), to an upper Hessenburg matrix. The "system matrix" is a fully populated, general matrix. The Hessenburg matrix is an upper triangular matrix with one sub-diagonal row (Jennings 1977).
The reduction of the "system matrix," $A$, to upper Hessenburg is performed using elementary stabilized transformations. Defining a matrix as elementary denotes the diagonal terms to be unity and nearly all others to be zero. The elementary transformation matrix for the $k$'th reduction step, $N_k$, is defined as

$$
n_{ii} = 1$$

$$n_{i,k+1} = 0$$

for $i < k+1$

$$\neq 0$$

for $i > k+1$

$$n_{ij} = 0$$

for $i \neq j$, $j \neq k$.

In short, matrix $N_k$, of order $n$, has a unit diagonal, non-zero terms below the diagonal in column $(k+1)$, and all remaining terms equal to zero (Jennings 1977).

$$N_k = \begin{bmatrix}
k & k+1 & k+2 \\
1 & & \\
& 1 & \\
& & 1 \\
& & n_{k+2,k+1} \\
& & n_{k+3,k+1} \\
& & \vdots \\
n_{n,k+1} & & 1
\end{bmatrix}$$

The term stabilized denotes the transformation values, $n_{i,k+1}$, to be equal to or less than unity in magnitude. This assures stability in the $k$'th reduction step.
The reduction matrix, \( N_k \), reduces the necessary elements in column \( k \) of the 'system matrix,' \( A \); elements \( a_{k+2,k} \) to \( a_{n,k} \) are reduced to zero. Figure 3 gives an example of the similarity transformation reduction to upper Hessenburg. As shown in Figure 3, the first similarity transformation reduces column one of \( A^{(1)} \); elements \( a_{31} \) and \( a_{41} \) are reduced to zero. The second similarity transformation reduces column \( A^{(2)} \); element \( a_{42} \) is reduced to zero.

The required value of \( n_{i,k+1}^{(k+1)} \) can be determined by reviewing how \( a_{i,k}^{(k+1)} \) is obtained. Referring to Figure 3, in the first similarity transformation,

\[
\begin{align*}
a_{32}^{(2)} &= (-n_{32})a_{21} + a_{31} = 0 \quad \Rightarrow \quad n_{32} = a_{31}/a_{21} \\
a_{42}^{(2)} &= (-n_{42})a_{21} + a_{41} = 0 \quad \Rightarrow \quad n_{42} = a_{41}/a_{21}
\end{align*}
\]

and, in general

\[
\begin{align*}
n_{i,k+1}^{(k+1)} &= a_{i,k}/a_{k+1,k} \ .
\end{align*}
\]

Applying this to the second similarity transformation, the equation is confirmed.

\[
\begin{align*}
a_{42}^{(3)} &= (-n_{42})a_{32} + a_{32} \\
&= -(a_{42}/a_{32})a_{32} + a_{42} = 0
\end{align*}
\]
The fourth order matrix, $A$, is reduced to Upper Hessenburg.

First Similarity Transformation: $A^{(2)} = N_1^{-1} A^{(1)} N_1$

$$A^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -n_{32} & 1 & 0 \\ 0 & -n_{42} & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & n_{32} & 1 & 0 \\ 0 & n_{42} & 0 & 1 \end{bmatrix}$$

$$A^{(2)} = \begin{bmatrix} a_{11} & -a_{12} & a_{13} & a_{14} \\ a_{21} & -a_{22} & -a_{23} & a_{24} \\ 0 & a_{32} & -a_{33} & -a_{34} \\ 0 & a_{42} & a_{43} & -a_{44} \end{bmatrix}$$

modified by $N_1$

modified by $N_1^{-1}$

Second Similarity Transformation: $A^{(3)} = N_2^{-1} A^{(2)} N_2$

$$A^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -n_{43} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & n_{43} & 1 \end{bmatrix}$$

Fig. D-1. Reduction to Upper Hessenburg.
\[ A^{(3)} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{14} \\ a_{12} & a_{22} & \cdots & a_{24} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{42} & a_{44} \end{bmatrix} \]

Total Transformation: \[ A^{(3)} = N_2^{-1} N_1^{-1} A^{(1)} N_1 N_2 \]

\[ = N^{-1} A^{(1)} N \]

Fig. D-1. (Continued)
With two similarity transformations, the original "system matrix," A\(^{(1)}\), is reduced to an upper Hessenburg matrix, A\(^{(n-1)}\) or A\(^{(3)}\).

Recall that the similarity transformations to be used in this reduction were also defined as stabilized. A stabilized transformation has all transformation elements equal to or less than unit magnitude.

\[ n_{i,k+1} \leq 1 \]

To achieve this stabilizing affect, partial pivoting is employed.

On determining N\(_k\), each \( n_{i,k+1} \) is divided by \( a_{k+1,k} \):

\[ n_{i,k+1} = \frac{a_{i,k}}{a_{k+1,k}} \quad \text{(D-5)} \]

The magnitude of \( n_{i,k+1} \) will be less than or equal to one if

\[ |a_{k+1,k}| \geq |a_{i,k}|. \]

If element \( a_{k+1,k} \) has the greatest absolute value of all sub-diagonal elements in column \( k \), N\(_k\) will be a stabilized elementary transformation matrix.

Location (k+1,k) is called the pivot location for the k'th transformation. To perform the partial pivot, the sub-diagonal elements of column \( k \) are searched for the location of the element of largest magnitude. If this element already exists in the pivot location (k+1,k), no pivoting is necessary. However, if the elements exist in location (j,k), rows (k+1) and (j) are exchanged, as
well as columns \((k+1)\) and \((j)\). It is necessary to exchange rows and columns to preserve the original eigenvalues. This completes the partial pivoting of the \(k'\)th transformation. Now the transformation matrix, \(N_k'\), can be calculated, and the \(k'\)th transformation performed. A record of each row and column permutation must be kept so that the elements of the final eigenvectors can be placed in their proper order. A permutation vector of length \(n\) is used to record the exchanges. When row and column \((j)\) are exchanged with row and column \((k+1)\), elements in locations \((j)\) and \((k+1)\) of the permutation vector are also exchanged. In this way, a record is kept of every row and column permutation performed (Jennings 1977).

A numerical example of this procedure is given in Figure 4 for a matrix of order 4. Note how the transformation elements, \(n_{i,k+1}'\), are stored in the location of the element it reduces to zero, \(a_{i,k}\). This is a significant optimization of storage space. The permutation vector, called IPERM \((n)\) is used to record the row and column permutations.
\[ A^{(1)} = \begin{bmatrix} 1 & 4 & 8 & 6 \\ 5 & 2 & -1 & 3 \\ 8 & 7 & 6 & 3 \\ 2 & -10 & 5 & 4 \end{bmatrix} \quad \text{IPERM}^{(1)} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \]

\[ A^{(2)} = \begin{bmatrix} 1.0000 & 12.0000 & 4.0000 & 6.0000 \\ 8.0000 & 11.1250 & 7.0000 & 3.0000 \\ 0.6250 & -5.9531 & -2.3750 & 1.1250 \\ 0.2500 & -3.0313 & -11.7500 & 3.2500 \end{bmatrix} \quad \text{IPERM}^{(2)} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \]

\[ A^{(3)} = \begin{bmatrix} 1.0000 & 12.0000 & 7.0551 & 6.0000 \\ 8.0000 & 11.1250 & 8.5276 & 3.0000 \\ 0.6250 & -5.9531 & -1.8022 & 1.1250 \\ 0.2500 & 0.5092 & -9.1775 & 2.6772 \end{bmatrix} \quad \text{IPERM}^{(2)} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 4 \end{bmatrix} \]

\[ N = N_1 N_2 = \begin{bmatrix} 1.0000 \\ 0.0000 & 1.0000 \\ 0.0000 & 0.6250 & 1.0000 \\ 0.0000 & 0.2500 & 0.5092 & 1.0000 \end{bmatrix} \]

Fig. D-2. Hessenburg reduction of 4th order A.
APPENDIX E

QR ALGORITHM*

The QR Algorithm is a procedure which reduces an upper Hessenberg matrix to upper triangular form. The eigenvalues are then identically those elements on the diagonal. The QR Algorithm is simply stated as

\[ A_s = Q_s R_s \]  \hspace{1cm} (E-1)

\[ A_{s+1} = R_s Q_s . \]  \hspace{1cm} (E-2)

The matrix \( A_s \), where \( s \) denotes the \( s \)'th iteration, is decomposed into a real orthogonal matrix, \( Q_s \), and an upper triangular matrix, \( R_s \). Since \( Q_s \) is orthogonal, equations (E-1) and (E-2) may be rewritten as

\[ Q_s^T A_s = R_s \]  \hspace{1cm} (E-3)

\[ A_{s+1} = Q_s^T A_s Q_s . \]  \hspace{1cm} (E-4)

\( Q_s^T \) is the orthogonal matrix which reduces \( A_s \) to triangular form.

The matrix \( Q_s^T \) is determined in factorized form. It is the product of plane rotations, \( R_{ij} \), or elementary Hermitians, \( P_r \), using Givens or Householder triangularization, respectively. The \( s \)'th iteration

---

* The material in this Appendix was obtained from Wilkinson (1965).
is completed by post multiplying the upper triangular matrix, $R_s$, by $Q_s$; post-multiply $R_s$ by the transpose of the orthogonal factors that compose $Q_s^T$. The QR Algorithm, using equations (E-3) and (E-4), produces a nearly orthogonal matrix $A_s^T$. In turn, $A_{s+1}$ will be close to an exact orthogonal transformation of $A_s$. An exact orthogonal transformation of $A_s$ cannot be assured using equations (E-1) and (E-2).

The QR Algorithm is recommended for Hessenburg matrices rather than full general matrices. This is due to the amount of computations involved in each iteration. A general matrix is reduced to upper Hessenburg using stable similarity transformations. This is a non-iterative procedure. The upper Hessenburg matrix is then reduced to upper triangular form using the QR Algorithm. The QR Algorithm is preferred in this final reduction phase because of its numerical stability and speed in convergence. There are several methods in which the convergence to diagonal eigenvalues is accelerated: shifts of origin, single or double small sub-diagonal elements, and deflation. These methods will be discussed after the general QR procedure has been explained. The $s$'th iteration determines $A_{s+1}$ from $A_s$. The $(n-1)$ sub-diagonal elements tend to zero as the number of iterations increases. A proof of the convergence of $A_s$ to triangular form, with the eigenvalues of the diagonal, is given in Wilkinson (1965). We must note at this point that $A_s$ will tend to a true upper triangular matrix only if all the eigenvalues
of $A_s$ are real. In our application, however, we expect the eigenvalues to occur in complex conjugate pairs. $A_s$ will converge to $(2 \times 2)$ sub-matrices on the diagonal. The roots of these $(2 \times 2)$ sub-matrices are the complex conjugate eigenvalues. The sub-diagonal element $a_{r+1,r}$ of the $(2 \times 2)$ diagonal sub-matrix is not zero.

$$
\begin{bmatrix}
\ddots & a_{r,r} & a_{r,r+1} & a_{ij} \\
& a_{r+1,r} & a_{r+1,r+1} & \\
& & a_{r+1,r} & a_{r+1,r+1} \\
& & & 0 & \ddots
\end{bmatrix}
$$

(E-5)

The eigenvalues of this $(2 \times 2)$ sub-matrix, $\lambda_r$ and $\lambda_{r+1}$, are obtained from the roots of its characteristic equation. In this appendix, the eigenvalues will be called $\lambda$, not $e^\lambda$ as before.

The QR Algorithm, as stated in equations (E-3) and (E-4), does not include any techniques to accelerate convergence. Recall $A_s$ is in the form of upper Hessenburg. If any sub-diagonal element, $a_{r+1,r}'$, is zero to working accuracy, the transformation of $A_s$ to $A_{s+1}$ is not unique. However, a sub-diagonal "zero" can be used to accelerate the entire procedure of determining the eigenvalues of $A$. If a sub-diagonal element is zero to working accuracy, $a_{r+1,r} = \varepsilon$, the matrix $A_s$ can be partitioned as shown below:
\( A_s = \begin{bmatrix} X & X & X & X & X & X \\ X & X & X & X & X & X \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X & X & X & X & X & X \\ X & X & X & X & X & X \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix}, \quad \text{(E-6)} \)

where \( n = 6 \) and \( r = 2 \).

Once partitioned, matrices \( B \) and \( D \) are solved for the eigenvalues of \( A_s \). If \( B \) or \( D \) are of order two, their eigenvalues are the roots of the second order characteristic equation. If \( B \) or \( D \) are of order one, the isolated element is an eigenvalue itself. If \( B \) or \( D \) are of greater order than two, they are Hessenburg matrices, but of smaller order than \( A_s \). The QR Algorithm now continues with matrix \( D \) of order \( (n - r) \). Once \( D \) has been reduced to upper triangular form, and its eigenvalues determined, the same is performed on matrix \( B \) of order \( (r) \). This acceleration technique significantly reduces the number of computations. A single QR iteration, using Givens method, requires \( 4n^2 \) multiplications. One QR iteration of the entire matrix of equation (E-6) would require 144 multiplications. Using the partitioned matrix, one QR iteration of \( D \) requires 64 multiplications. Matrix \( B \) is of order two, so its characteristic equation may be solved directly for its eigenvalues. The advantage of a single small sub-diagonal element is obvious in one iteration.
A similar procedure to that just described exists for the case when there are two consecutive small sub-diagonal elements, $\varepsilon_1$ and $\varepsilon_2$. This acceleration method will not be explained until the Double QR Algorithm for complex conjugate eigenvalues is covered.

The most important method which accelerates convergence is incorporating "shifts of origin". Using Hessenburg matrices, rather than full general matrices, still does not give a sufficient rate of convergence. In general, a sub-diagonal element $a_{ij}$, for $i > j$, will converge at the rate of $(\lambda_i/\lambda_j)^s$. A sub-diagonal element of a Hessenburg matrix, $a_{r+1,r}$, will converge at the rate of $(\lambda_{r+1}/\lambda_r)^s$.

The purpose of the "shift of origin" is to adjust the convergence ratio for the element $a_{n,n-1}$. The element $a_{n,n-1}$ is reduced to zero as quickly as possible. With $a_{n,n-1}$ reduced to zero, the matrix is partitioned as in the single small sub-diagonal acceleration method. The result is eigenvalue $\lambda_n$ in location $a_{n,n}$. When $\lambda_n$ is determined, the matrix $A_{s+1}$ can be truncated by row and column $n$. The QR Algorithm continues on $A_{s+1}$ of order $(n - 1)$. When complex eigenvalues are involved, element $a_{n-1,n-2}$ is reduced to zero. The bottom right (2 x 2) sub-matrix is solved for the eigenvalues $\lambda_{n-1}$ and $\bar{\lambda}_n$. The last two rows and columns of $A_{s+1}$ are truncated. The QR Algorithm continues on $A_{s+1}$ of order $(n - 2)$.

Consider the matrix $(A_s - pI)$ which has eigenvalues $(\lambda_i - p)$. The element $a_{n,n-1}$ will tend to zero at the rate of $[((\lambda_n - p)/\
(\lambda_{n-1} - p))]^s$. Now, if $p$ is very close to $\lambda_n$, this ratio will rapidly converge to zero. If $p$ equals $\lambda_n$, $a_{n,n-1}$ will be zero in one iteration.
To incorporate the shift of origin, called $k_s$, the $s$'th iteration is performed as follows:

\[
A_s - k_s I = Q_s R_s \quad (E-7)
\]

\[
A_{s+1} = R_s Q_s + k_s I. \quad (E-8)
\]

Because $k_s$ is added back in equation (E-8), this modification is described as the QR Transformation with "shifts of origin and restoring". The matrix $A_{s+1}$ is similar to $A_s$. From equation (E-7),

\[
Q_s^T (A_s - k_s I) = R_s. \quad (E-9)
\]

Using equations (E-9) and (E-8), the similarity is demonstrated.

\[
A_{s+1} = Q_s^T (A_s - k_s I) Q_s + k_s I
\]

\[
= Q_s^T A_s Q_s - k_s Q_s^T I Q_s + k_s I
\]

\[
= Q_s^T A_s Q_s - k_s I + k_s I
\]

\[
= Q_s^T A_s Q_s \quad (E-10)
\]

Continuing with this procedure, it can be shown that all matrices are similar to $A_1$.

\[
A_{s+1} = Q_s^T Q_{s-1}^T \cdots Q_2^T Q_1^T A_1 Q_1 Q_2 \cdots Q_{s-1} Q_s \quad (E-11)
\]

For $s$ iterations,

\[
A_s = Q_s^T \cdots Q_2^T Q_1^T A_1 Q_1 Q_2 \cdots Q_{s-1}. \quad (E-12)
\]
For later use, equation (E-12) can also be written as

\[ Q_1 Q_2 \ldots Q_{s-2} \ Q_{s-1} \ A_s = A_1 \ Q_1 \ Q_2 \ldots Q_{s-2} \ Q_{s-1} \cdot \quad (E-13) \]

Subtracting \( k_s I \) from both sides of equation (E-12) yields

\[ (A_s - k_s I) = Q_s^T \ldots Q_2^T \ Q_1^T \ (A_1 - k_s I) \ Q_1 \ Q_2 \ldots Q_{s-1} \quad (E-14) \]

or, when written in a format similar to equation (E-13),

\[ Q_1 \ Q_2 \ldots Q_{s-2} \ Q_{s-1} \ (A_s - k_s I) = (A_1 - k_s I) \ Q_1 \ Q_2 \ldots Q_{s-2} \ Q_{s-1} \cdot \quad (E-15) \]

The following derivation uses equations (E-7) and (E-15) to determine the QR decomposition for \( s \) iterations.

\[ \begin{align*}
Q_1 \ Q_2 \ldots Q_{s-1} \ (Q_s R_s) R_{s-1} \ldots R_2 \ R_1 \\
= Q_1 \ Q_2 \ldots Q_{s-1} \ (A_s - k_s I) R_{s-1} \ldots R_2 \ R_1 \\
= (A_1 - k_s I) \ Q_1 \ Q_2 \ldots (Q_{s-1} R_{s-1}) \ldots R_2 \ R_1 \\
= (A_1 - k_s I) \ Q_1 \ Q_2 \ldots Q_{s-2} \ (A_{s-1} - k_{s-1} I) R_{s-2} \ldots R_2 \ R_1 \\
= (A_1 - k_s I)(A_1 - k_{s-1} I) \ Q_1 \ Q_2 \ldots (Q_{s-2} R_{s-2}) \ldots R_2 \ R_1 \\
\vdots \\
= (A_1 - k_s I)(A_1 - k_{s-1} I) \ldots (A_1 - k_2 I) \ Q_1 \ R_1 \\
= (A_1 - k_s I)(A_1 - k_{s-1} I) \ldots (A_1 - k_2 I)(A_1 - k_1 I)
\end{align*} \]
The result is

\[ Q_1 Q_2 \cdots Q_{s-1} Q_s R_s R_{s-1} \cdots R_2 R_1 \]

\[ = \prod_{i=1}^{s} (A_1 - k_i I). \quad (E-16) \]

To simplify equation (E-16), \( R_s \) and \( Q_s \) are defined as

\[ Q_s = Q_1 Q_2 \cdots Q_{s-1} Q_s \quad (E-17a) \]

\[ R_s = R_s R_{s-1} \cdots R_2 R_1. \quad (E-17b) \]

The above derivation can now be summarized as

\[ Q_s R_s = \prod_{i=1}^{s} (A_1 - k_i I). \quad (E-18) \]

The order of factors on the right of equation (E-18) is unimportant. What is important, however, is that the product \( Q_s R_s \) was shown to be the decomposition of the product of \( s \) factors \((A_1 - k_i I)\). The orthogonal matrix \( Q_s \) gives the triangular decomposition, \( R_s \), of \( \prod_{i=1}^{s} (A_1 - k_i I) \).

The QR Algorithm using shifts of origin with restoring, can be modified to "non-restoring shifts". This process is defined by

\[ (A_s - z_s I) = Q_s R_s \quad (E-19) \]
In this case,

\[ A_{s+1} = R_s Q_s. \]  

(E-20)

\[
A_{s+1} = R_s Q_s \\
= Q_s^T (A_s - z_s I) Q_s \\
= Q_s^T A_s Q_s - z_s I \\
= Q_s^T (Q_{s-1}^T A_{s-1} Q_{s-1} - z_{s-1} I) Q_s - z_s I \\
= Q_s^T Q_{s-1}^T A_{s-1} Q_{s-1} Q_s - (z_{s-1} + z_s) I \\
\vdots \\
A_{s+1} = Q_s^T Q_{s-1}^T \cdots Q_1^T A_1 Q_1 \cdots Q_{s-1} Q_s - (\sum_{i=1}^{s} z_i) I \\
A_{s+1} = Q_s^T Q_{s-1}^T \cdots Q_1^T (A_1 - (\sum_{i=1}^{s} z_i I)) Q_1 \cdots Q_{s-1} Q_s \\
\]  

(E-21)

(E-22)

The eigenvalues of \( A_{s+1} \) differ from \( A_1 \) by the sum of the shifts, \( z_i \).

The shift of origin, \( k_s \), may be chosen in different ways, depending on whether \( A_1 \) is real or complex, and on whether its eigenvalues are all real or both real and complex. In this application, \( A_1 \) will always be real.
When $A_1$ is known to have real eigenvalues, $a_{n,n-1}$ will tend to zero and $a_{n,n}$ to $\lambda_n$. For this reason, $k_s$ is taken to be $a_{n,n}^s$ as soon as $a_{n,n-1}^s$ is small, or $a_{n,n}$ begins to converge to $\lambda_n$. When $a_{n,n-1}^s$ is of order $\varepsilon$, and the shift $a_{n,n}$ is used, $a_{n,n-1}^{s+1}$ is of order $\varepsilon^2$ (Wilkinson 1965). Therefore, once $a_{n,n-1}$ is small, it converges to zero rapidly if the shift $k_s = a_{n,n}$ is used.

A method which gives faster convergence uses the real eigenvalues of the bottom right corner ($2 \times 2$) matrix. The real eigenvalues, say $p_s$ and $q_s$, are determined from the characteristic equation of the ($2 \times 2$) matrix. The shift, $k_s$, is taken to be $|p_s - a_{n,n}^s|$ or $|q_s - a_{n,n}^s|$, according to which is greater. This method may be used at every stage, or employed when the shift value shows signs of convergence. An acceptable criteria is to apply the shifts as soon as

$$|k_s / k_{s-1} - 1| < \frac{1}{2}.$$ 

When $A_1$ is suspected of having complex conjugate eigenvalues, $\lambda$ and $\bar{\lambda}$, a similar acceleration procedure to that just described may be used. The complex conjugate eigenvalues are determined from the bottom right corner ($2 \times 2$) matrix. If two QR iterations are performed using $\lambda$ and $\bar{\lambda}$, the resulting matrix is real. However, the intermediate matrix and all the arithmetic involved is complex. In 1961, Francis developed a procedure called the Dougle QR Algorithm, in which all complex arithmetic is avoided.
To introduce the Double QR Algorithm, consider the s'th stage of the transformation. The eigenvalue of the bottom right corner (2 x 2) matrix are used as shifts $k_1$ and $k_2$. This pair may be real or complex conjugates. Two QR iterations are performed with shifts $k_1$ and $k_2$, respectively. Equations (E-7) and (E-8) are used in these steps, such that

$$A_s - k_s I = Q_s R_s$$
$$R_s Q_s + k_s I = A_{s+1}$$

(E-23a,b)

$$A_{s+1} - k_{s+1} I = Q_{s+1} R_{s+1}$$
$$R_{s+1} Q_{s+1} + k_{s+1} I = A_{s+2}$$

(E-24a,b)

Matrices $A_{s+1}$ and $A_{s+2}$ are similar to $A_s$ as shown below. Using equations (E-23a and b),

$$R_s = Q_s^T (A_s - k_s I)$$

$$Q_s^T (A_s - k_s I) Q_s + k_s I = A_{s+1}$$

$$Q_s^T A_s Q_s = A_{s+1}$$

(E-25)

Using equations (E-24a and b) in a similar manner,

$$Q_{s+1}^T A_{s+1} Q_{s+1} = A_{s+2}$$

(E-26)

Equations (E-25) and (E-26) are combined, to get

$$Q_{s+1}^T Q_s^T A_s Q_s Q_{s+1} = A_{s+2}$$
Equations (E-27a) and (E-25) show that $A_{s+2}$ and $A_{s+1}$ are similar to $A_s$, respectively. Next, the decomposition of $(A_s - k_{s+1} I)(A_s - k_s I)$ must be determined.

\[(A_{s+1} - k_{s+1} I)(A_s - k_s I)\]

\[= (A_s - k_{s+1} I) Q_s R_s\]

from equation (E-23a). Subtract $k_{s+1} I$ from equation (E-25) to get

\[Q_s^T (A_s - k_{s+1} I) Q_s = A_{s+1} - k_{s+1} I\]

\[(A_s - k_{s+1} I) Q_s = Q_s (A_{s+1} - k_{s+1} I)\]  \hspace{1cm} (E-28)

Substituting equation (E-28) into the first expression yields

\[(A_s - k_{s+1} I)(A_s - k_s I) = Q_s (A_{s+1} - k_{s+1} I) R_s\]  \hspace{1cm} (E-29)

\[= Q_s Q_{s+1} R_{s+1} R_s\]

Define $Q$ and $R$ as

\[Q = Q_s Q_{s+1}\]  \hspace{1cm} (E-30a)

\[R = R_{s+1} R_s\]  \hspace{1cm} (E-30b)
Substituting equations (E-30a) and (E-30b) into equation (E-29) yields the orthogonal decomposition, QR.

\[ QR = (A_s - k_{s+1} I)(A_s - k_s I) \]  
(E-31)

If \( k_s \) and \( k_{s+1} \) are not exact eigenvalues, the decomposition is unique provided that \( R \) has positive diagonal elements. The right side of equation (E-31) is real. Therefore, \( Q \) is also real and is obtained from the triangular decomposition of \((A_s - k_{s+1} I)^* (A_s - k_s I)\).

Substitute equation (E-30a) into equation (E-27c) to get

\[ A_s Q = Q A_{s+2} \]  
(E-32)

Since \( A_s \) and \( A_{s+2} \) are similar, and \( Q \) is orthogonal, the matrix \( A_{s+2} \) is a Hessenburg matrix. The matrix \( Q \) can be determined without any complex arithmetic, as will be shown. Rearrange equation (E-31) such that

\[ Q^T (A_s - k_{s+1} I)(A_s - k_s I) = R \]  
(E-33)

\( Q^T \) is the orthogonal matrix that reduces \((A_s - k_{s+1} I)(A_s - k_s I)\) to the upper triangular matrix \( R \).

In the general case, \( Q^T \) is determined as the product of \( \frac{1}{2} n(n-1) \) plane rotations, \( R_{ij} \), for \( i < j \).

\[ Q^T = R_{n-1,n} \cdots (R_{2,n} R_{2,n-1} \cdots R_{2,3}) \ast (R_{1,n} R_{1,n-1} \cdots R_{1,3} R_{1,2}) \]  
(E-34)
The first row of $Q^T$ is the first row of the matrix product
\[ R_{1,n} R_{1,n-1} \cdots R_{1,3} R_{1,2}. \]
Multiplication of other plane rotation matrices, $R_{1,j}$, does not change this row. The matrix product $R_{1,n} * R_{1,n-1} \cdots R_{1,3} R_{1,2}$ is determined from the first column of $(A_s - k_s I)(A_s - k_{s+1} I)$. The first row of $Q^T$ can be determined from the first column of $(A_s - k_s I)(A_s - k_{s+1} I)$. For simplicity, let $s$ equal one. The first column of $(A_1 - k_1 I)(A_1 - k_2 I)$ contains only three elements which we will call $x_1$, $y_1$ and $z_1$.

\[
x_1 = (a_{11} - k_1 I) (a_{11} - k_2 I) + a_{12} a_{21}
\]
\[
= a_{11}^2 - (k_1 + k_2) a_{11} + k_1 k_2 + a_{12} a_{21}
\]
(E-35a)

\[
y_1 = a_{21} (a_{11} - k_2) + (a_{22} - k_1) a_{21}
\]
\[
= a_{21} (a_{11} + a_{22} - k_1 - k_2)
\]
(E-35b)

\[
z_1 = a_{32} a_{21}
\]
(E-35c)

Whether $k_1$ and $k_2$ are real or a complex conjugate pair, the elements $x_1$, $y_1$, and $z_1$ are all real. The matrices $R_{1,2}$ and $R_{1,3}$ are determined from $x_1$, $y_1$, and $z_1$. Consequently, the matrices $R_{1,2}$ and $R_{1,3}$ and the matrix product $R_{1,3} R_{1,2}$ are real. The remaining plane rotations for the first column of $(A_1 - k_1 I)(A_1 - k_2 I)$ are the identity matrix.
\[ R_{1,4} = R_{1,5} = \ldots = R_{1,n} = I \]

Above, we stated that the first row of \( Q^T \) equals the first row of the matrix product \( R_{1,n} R_{1,n-1} \ldots R_{1,3} R_{1,2} \). With \( R_{1,4} \ldots R_{1,n} \) equaling the identity matrix, the first row of \( Q^T \) is simply equal to the first row of \( R_{1,3} R_{1,2} \).

With \( s \) equal to one, equation (E-32) is rewritten, such that
\[
Q^T A_1 Q = A_3. \tag{E-36}
\]
where \( A_1 \) and \( A_3 \) are of upper Hessenburg form. A real matrix \( C_1 \) is defined as
\[
C_1 = R_{1,3} R_{1,2} A_1 R_{1,2}^T R_{1,3}^T. \tag{E-37}
\]

Given an orthogonal matrix \( S \) with its first column equal to \( \{e_1\} \), where
\[
\{e_1\} = \{1, 0, 0, \ldots 0\}^T,
\]
it can be shown for a real matrix \( C \), that
\[
S_1^T C_1 S_1 = B. \tag{E-38}
\]
Here, \( B \) is an upper Hessenburg matrix. Inserting equation (E-37) into (E-38) yields
\[
(S_1^T R_{1,3} R_{1,2}) A_1 (R_{1,2}^T R_{1,3}^T S_1) = B. \tag{E-39}
\]
Defining \( (S_1^T R_{1,3} R_{1,2}) \) as \( Q^T \),
\[
Q^T A_1 Q = B. \tag{E-40}
\]
The first row of $S_1^T$ is $\{e_1\}^T$. Consequently, the first row of $Q^T$ is the first row of $R_{1,3} R_{1,2}$. Previously we established that the first row of $Q^T$ equals the first row of $R_{1,3} R_{1,2}$. We can now conclude that the first row of $Q^T$ equals the first row of $Q^T$. With this established, a comparison of equations (E-36) and (E-40) shows that $B$ must be $A_3$. The matrix $A_3$ can, therefore, be obtained with no complex arithmetic using two steps of QR and shifts $k_1$ and $k_2$. This double shift technique is called the Double QR Algorithm.

Up to this point, the QR Algorithm has been explained using Givens method with plane rotation matrices, $R_{ij}$. For the Double QR with shifts, however, elementary Hermitian transformation matrices are much more efficient. For this reason, the details of the Double QR Algorithm will be discussed using Householder's method which uses elementary Hermitian matrices, $P_r$. The elementary Hermitian matrix, $P_r$, will replace the plane rotation matrix, $R_{ij}$, of Givens method.

We define the elementary Hermitian $P_r$ as

$$P_r = I - 2 \{\omega_r\}^T \{\omega_r\}^T,$$

(E-41)

where

$$\{\omega_r\} = \{0, 0, \ldots, 0, x, x, \ldots x\}.$$  

(E-42)

The elements in locations one through $(r - 1)$ are zero. Initially, $P_1$ is calculated using $x_1$, $y_1$, and $z_1$. Referring to the definition
of \{\omega_r\} in equation (E-42), only the first three elements of \{\omega_1\} will be non-zero. Further discussion on how \(P_1\) is calculated will be delayed until after the determination of \(P_r\) in general has been discussed.

For the present, assume \(P_1\) to be known. The intermediate matrix, \(C_1\), is determined by

\[
C_1 = P_1 A_1 P_1^T. \tag{E-43}
\]

For a system of order \(n = 8\), the matrix \(C_1\) has the form

\[
C_1 = \begin{bmatrix}
X & X & X & X & X & X & X & X \\
X & X & X & X & X & X & X & X \\
X & X & X & X & X & X & X & X \\
X & X & X & X & X & X & X & X \\
X & X & X & X & X & X & X & X \\
X & X & X & X & X & X & X & X \\
X & X & X & X & X & X & X & X \\
X & X & X & X & X & X & X & X
\end{bmatrix}. \tag{E-44}
\]

Continuing with Householder's method, the \(s\)'th iteration is completed by reducing \(C_1\) to upper Hessenberg. After the \(r\)'th stage of this iteration is complete,

\[
C_r = P_r \cdots P_2 P_1 A_1 P_1^T P_2^T \cdots P_r^T. \tag{E-45}
\]

For \(r = 3\), \(C_r\) has the form
The additional non-zero elements in the r'th stage are \((r, r+2), (r, r+3)\) and \((r+1, r+3)\). Pre-multiplying the \(P_{r+1}\) and post-multiplying by \(P_{r+1}^T\) will reduce elements \((r, r+2)\) and \((r, r+3)\) to zero. The result will be the matrix \(C_{r+1}\) which now has the non-zero elements \((r+1, r+3)\), \((r+1, r+4)\) and \((r+2, r+4)\). In general, pre- and post-multiplying by \(P_r\) and \(P_r^T\), respectively, reduces the element \((r, r+2)\) and \((r, r+3)\) of \(C_r\) to zero. There are a total of \((n-2)\) stages in each iteration. The first iteration can be written as

\[
P_{n-2} \cdots P_r \cdots P_2 P_1 A_1 P_1^T P_2^T \cdots P_r^T \cdots P_{n-2}^T = B. \tag{E-47}
\]

The matrix \(B\) is an upper Hessenburg matrix. \(P_1\) has the same first row as the orthogonal matrix, \(Q\), that triangularizes \((A_1 - k_1 I)^* (A_1 - k_2 I)\). Recall equation (E-33) for the iteration \(s = 1\).

\[
Q^T (A_1 - k_1 I)(A_1 - k_2 I) = R
\]
R was defined as upper triangular. In equation (E-47), the matrix product \( P_{n-2} \ldots P_r \ldots P_2 P_1 \) has the same first row as \( P_1 \). Therefore, the matrix product \( P_{n-2} \ldots P_r \ldots P_2 P_1 \) has the same first row as \( Q^T \), and \( B \) must equal \( A_3 \). It has now been shown that the Double QR Algorithm, with shifts \( k_1 \) and \( k_2 \), can be performed by Householders method using elementary Hermitian matrices. As with Givens method, no complex arithmetic is involved.

The elementary Hermitian matrix, \( P_r \), is determined in the following manner. To begin, equations (E-35a, b, and c) are modified for the \( r \)'th stage, such that

\[
x_r = a_{r,r}^2 - (k_1 + k_2) a_{r,r} + k_1 k_2 + a_{r,r+1} a_{r+1,r} \tag{E-48a}
\]

\[
y_r = a_{r+1,r} (a_{r,r} + a_{r+1,r+1} - k_1 - k_2) \tag{E-48b}
\]

\[
z_r = a_{r+2,r+1} a_{r+1,r} \tag{E-48c}
\]

Referring to equations (E-41) and (E-42), \( \{w_r\} \) is normalized and renamed as the vector \( \{p_r\} \).

\[
\{p_r\} = \{0, 0, \ldots, 0, 1, u_r v_r, 0, \ldots, 0\}^T \tag{E-49}
\]

The three non-zero elements of \( \{p_r\} \) are in location \( (r) \), \( (r + 1) \) and \( (r + 2) \). Next, define

\[
u_r = y_r/(x_r + s_r) \tag{E-50a}
\]

\[
v_r = z_r/(x_r + s_r) \tag{E-50b}
\]
The sign in equations (E-50a and b) is chosen according to

$$s_r^2 = x_r^2 + y_r^2 + z_r^2$$  \hspace{1cm} (E-50c)

$$||p_r||_2^2 = (1 + u_r^2 + v_r^2)$$ \hspace{1cm} (E-50d)

Finally, the elementary Hermitian matrix, \( P_r \), is defined as

$$P_r = I - 2\{p_r\}{p_r}^T/||p_r||_2^2.$$ \hspace{1cm} (E-51)

To simplify computations, let

$$\alpha = 2/||p_r||_2^2 = 2/(1 + u_r^2 + v_r^2),$$

which gives

$$P_r = I - \{p_r\} (\alpha \{p_r\}^T).$$ \hspace{1cm} (E-52)

As a final simplification, define a truncated transformation vector, \( \{\tilde{p}_r\} \), in which the leading zeros are ignored. The vector contains only the elements from \( r \) to \( n \) and has a length of \( (n - r + 1) \).

To illustrate this and the matrix \( P_r \), let \( n = 6 \) and \( r = 3 \).

\[
\{p_r\} = \{0 \ 0 \ 1 \ u_r \ v_r \ 0\}^T \hspace{1cm} \text{(E-53a)}
\]

\[
\{\tilde{p}_r\} = \{1 \ u_r \ v_r \ 0\}^T \hspace{1cm} \text{(E-53b)}
\]
The elementary Hermitian matrix is partitioned and only the lower right matrix need be determined, \([I - \{\mathbf{p}_r\} (\alpha \{\mathbf{\tilde{p}}_r\}^T)]\).

It will be shown how \(P_1\) can be determined using the same method in which \(P_r\) is determined. The matrix \(A_1\), here shown of order \(n = 6\), is modified by adding one column at the front. The elements contained in this column are \(x_1\), \(y_1\) and \(z_1\).

\[
C_o = \begin{bmatrix}
x_1 & X & X & X & X & X & X \\
y_1 & X & X & X & X & X & X \\
z_1 & X & X & X & X & X & X \\
o & X & X & X & X & X & X \\
o & X & X & X & X & X & X \\
o & X & X & X \\
o & X & X \\
o & X \\
o & X \\
o & X & X \\
o & X & X & X & X & X & X \\
\end{bmatrix} = \{\mathbf{p}_1\}^T A_1 \tag{E-55}
\]

\[
P_r = \begin{bmatrix}
1 & 0 \\
0 & 1 \\
& & & & & & \\
\end{bmatrix} - \begin{bmatrix}
1 - \alpha & -\alpha u_r & -\alpha v_r & 0 & 0 & 0 \\
-\alpha u_r & 1 - \alpha u_r^2 & -\alpha u_r v_r & 0 & 0 & 0 \\
-\alpha v_r & -\alpha u_r v_r & 1 - \alpha v_r^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\tag{E-54}
\]
This augmented matrix, called $C_0$, is used to derive $C_1$ in the same manner as $C_r$ is derived from $C_{r-1}$. The steps to determine $P_r$ are listed below with the case $r = 1$ given in parentheses.

Before pre-multiplying by $P_r$ ($P_1$), the current $C_{r-1}$ ($C_0$) has the Hessenburg form with three additional non-zero elements:

$$C_{r-1}^{(0)} (C_{1,0}) \rightarrow x_r (x_1)$$

$$C_{r+1,r-1}^{(0)} (C_{2,0}) \rightarrow y_r (y_1)$$

$$C_{r+2,r-1}^{(0)} (C_{3,0}) \rightarrow z_r (z_1).$$

The transformation vector, $\{p_r\} (\{p_1\})$, is determined using equations (E-50a) through (E-50e). In this case, $\{p_1\}$ is the same as $\{p_1\}$ since there are no leading zeros to be truncated. The elementary Hermitian matrix $P_r$ ($P_1$) is now determined using equation (E-51) or (E-52).

At this point, the final acceleration method, being two consecutive small sub-diagonal elements, can be explained. Given $A_s$ with the two consecutive elements $a_{r+1,r}$ and $a_{r+2,r+1}$ of order $s$, the matrix can be partitioned in a similar manner to equation (E-6).

$$A_s = \begin{bmatrix}
X & X & X & X & X & X \\
X & X & X & X & X & X \\
X & X & X & X & X & X \\
X & X & X & X & X & X \\
X & X & X & X & X & X \\
X & X & X & X & X & X \\
X & X & X & X & X & X \\
X & X & X & X & X & X
\end{bmatrix} = \begin{bmatrix}
X & Y \\
E & Z
\end{bmatrix}$$

(E-56)
Elements \( a_{r+1,r} \) and \( a_{r+2,r+1} \) are denoted by \( \varepsilon_1 \) and \( \varepsilon_2 \).

The \( s' \)th iteration begins with the transformation matrix \( P_{r+1} \), rather than \( P_1 \), using \( x_{r+1}, y_{r+1} \) and \( z_{r+1} \) in equations (E-48a, b and c).

Two references recommended for further computational and theoretical details are Wilkinson (1965) and Wilkinson and Reinsch (1971).
DETERMINING EIGENVECTORS THROUGH BACK SUBSTITUTION

There are three cases to be considered when solving for the eigenvectors through back substitution. These are:

1. All real eigenvalues and eigenvectors
2. Both real and complex eigenvalues and eigenvectors
3. All complex eigenvalues and eigenvectors

All three cases will be demonstrated with a fourth order system. Case 1, being very straightforward, is covered first. A demonstration of Cases 2 and 3 will follow, each increasing in difficulty.

Case 1. All Real Eigenvalues and Eigenvectors

The eigenvector, \( \{\psi_4\} \), is determined first from the upper triangular matrix, \( T \), and the fourth eigenvalue, \( \lambda_4 \).

\[
[T - \lambda_4 I] \{\psi_4\} = 0
\]

\[
\begin{bmatrix}
t_{11} - \lambda_4 & t_{12} & \cdots & t_{14} \\
t_{22} - \lambda_4 & t_{23} & \cdots & t_{24} \\
\vdots & \vdots & \ddots & \vdots \\
t_{33} - \lambda_4 & t_{34} & \cdots & t_{34} \\
t_{44} - \lambda_4 & \cdots & \cdots & \lambda_4
\end{bmatrix}
\begin{bmatrix}
\psi_{14} \\
\psi_{24} \\
\psi_{34} \\
\psi_{44}
\end{bmatrix} = 0
\]

Recall that the diagonal elements of an upper triangular matrix are the eigenvalues; \( t_{ii} = \lambda_i \).
\[
\begin{bmatrix}
\lambda_1 - \lambda_4 & t_{12} & t_{13} & t_{14} \\
\lambda_2 - \lambda_4 & t_{23} & t_{24} & \\
\lambda_3 - \lambda_4 & t_{34} & & 0 \\
& & & \\
\end{bmatrix}
\begin{bmatrix}
\overline{\psi}_{14} \\
\overline{\psi}_{24} \\
\overline{\psi}_{34} \\
\overline{\psi}_{44}
\end{bmatrix} = 0
\]

In this case, \( \overline{\psi}_{11} \) will always be set to 1. The back substitution to find \( \{ \overline{\psi}_4 \} \) is, therefore:

\[
\overline{\psi}_{44} = 1.0
\]

\[
\overline{\psi}_{34} = -\frac{t_{34}}{(\lambda_3 - \lambda_4)}
\]

\[
\overline{\psi}_{24} = \frac{-t_{24} - t_{23} \overline{\psi}_{34}}{(\lambda_2 - \lambda_4)}
\]

\[
\overline{\psi}_{14} = \frac{-t_{14} - t_{13} \overline{\psi}_{34} - t_{12} \overline{\psi}_{24}}{(\lambda_1 - \lambda_4)}
\]

With \( \{ \overline{\psi}_4 \} \) determined, the fourth row and column of \( T \) are truncated. The third order system is solved for \( \{ \overline{\psi}_3 \} \).

\[
\begin{bmatrix}
\lambda_1 - \lambda_3 & t_{12} & t_{13} \\
\lambda_2 - \lambda_3 & t_{23} & & 0 \\
& & & \\
\end{bmatrix}
\begin{bmatrix}
\overline{\psi}_{13} \\
\overline{\psi}_{23} \\
\overline{\psi}_{33}
\end{bmatrix} = 0
\]

Here again, \( \overline{\psi}_{11} = \overline{\psi}_{33} = 1 \). Also, in this case, it is always true that

\[
\overline{\psi}_{ki} = 0 \quad \text{for} \quad k > i.
\]
With \( \{\bar{\psi}_3\} \) determined, the third row and column are truncated. The second order system is solved for \( \{\bar{\psi}_2\} \).

\[
\begin{bmatrix}
\lambda_1 - \lambda_2 & t_{12} \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\bar{\psi}_{12} \\
\bar{\psi}_{22}
\end{bmatrix} = 0
\]

\( \bar{\psi}_{42} = 0 \)  \hspace{1cm} (F-3)

\( \bar{\psi}_{32} = 0 \)

\( \bar{\psi}_{22} = 1 \)

\( \bar{\psi}_{12} = -t_{12}/(\lambda_1 - \lambda_2) \)

The final modal matrix, \( \bar{\psi} \), is

\[
\bar{\psi} = \begin{bmatrix}
1 & \bar{\psi}_{12} & \bar{\psi}_{13} & \bar{\psi}_{14} \\
1 & \bar{\psi}_{23} & \bar{\psi}_{24} \\
1 & \bar{\psi}_{34} \\
1
\end{bmatrix}
\]
Case 2. Both Real and Complex Eigenvalues and Eigenvectors

T is now a fourth order pseudo upper triangular matrix containing one pair of complex conjugate eigenvalues. The pair of complex eigenvalues is determined from the 2 x 2 diagonal sub-matrix. In this example, $\lambda_2$ and $\lambda_3$ are the complex conjugate eigenvalues.

$$[T - \lambda_4 I]\{\psi_4\} = 0$$

$$\begin{bmatrix}
  t_{11} - \lambda_4 & t_{12} & t_{13} & t_{14} \\
  t_{22} - \lambda_4 & t_{23} & t_{24} \\
  t_{32} & t_{33} - \lambda_4 & t_{34} \\
  0 & t_{44} & 0 & 1
\end{bmatrix} \begin{bmatrix}
  \psi_{14} \\
  \psi_{24} \\
  \psi_{34} \\
  \psi_{44}
\end{bmatrix} = 0$$

The diagonal elements, excluding those within the 2 x 2 sub-matrix, are eigenvalues.

$$\begin{bmatrix}
  \lambda_1 - \lambda_4 & t_{12} & t_{13} & t_{14} \\
  t_{22} - \lambda_4 & t_{23} & t_{24} \\
  t_{32} & t_{33} - \lambda_4 & t_{34} \\
  0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  \psi_{14} \\
  \psi_{24} \\
  \psi_{34} \\
  \psi_{44}
\end{bmatrix} = 0$$

As in Case 1, $\psi_{11} = 1$ for a real eigenvector: $\psi_{44} = 1$. The elements $\psi_{24}$ and $\psi_{34}$ are determined by solving rows two and three simultaneously.
These may be written in matrix form as

\[
\begin{bmatrix}
(t_{22} - \lambda_4) & t_{23} \\
t_{32} & (t_{33} - \lambda_4)
\end{bmatrix}
\begin{bmatrix}
\overline{\psi}_{24} \\
\overline{\psi}_{34}
\end{bmatrix}
= \begin{bmatrix}
-t_{24} \\
-t_{34}
\end{bmatrix}.
\]

Cramer’s Rule is used to determine \( \overline{\psi}_{24} \).

\[
\overline{\psi}_{24} = \frac{(-t_{24})(t_{33} - \lambda_4) - (t_{23})(-t_{34})}{[(t_{22} - \lambda_4)(t_{33} - \lambda_4) - t_{23}t_{32}]}.
\] (F-6)

With \( \overline{\psi}_{24} \) determined, \( \overline{\psi}_{34} \) may be found from either equation (F-5a) or (F-5b).

If \( |t_{23}| > |t_{22} - \lambda_4| \),

\[
\overline{\psi}_{34} = \frac{-t_{24} - (t_{22} - \lambda_4) \overline{\psi}_{24}}{t_{23}}.
\] (F-7a)

If \( |t_{22} - \lambda_4| > |t_{23}| \),

\[
\overline{\psi}_{34} = \frac{-t_{34} - (t_{32} \overline{\psi}_{24})}{(t_{33} - \lambda_4)}.
\] (F-7b)

And, finally
The entire fourth eigenvector, \( \{\Psi_4\} \), has now been determined.

The fourth row and column are truncated, resulting in a third order system. The next eigenvalue is one of a pair of complex conjugate eigenvalues, \( \lambda_2 \) and \( \lambda_3 \). The eigenvectors \( \{\Psi_2\} \) and \( \{\Psi_3\} \) must also be complex conjugates. It is only necessary to solve for one eigenvector, say \( \{\Psi_3\} \). The eigenvector can be conveniently stored with the real part in \( \{\Psi_2\} \) and the imaginary part in \( \{\Psi_3\} \).

\[
\begin{bmatrix}
\lambda_1 - \lambda_3 & t_{12} & t_{13} \\
t_{22} - \lambda_3 & t_{23} & \\
t_{32} & t_{33} - \lambda_3 & \end{bmatrix}
\begin{bmatrix}
\overline{\psi}_{12} + i \overline{\psi}_{13} \\
\overline{\psi}_{22} + i \overline{\psi}_{23} \\
\overline{\psi}_{32} + i \overline{\psi}_{33} \end{bmatrix} = 0
\]

To simplify computations, let

\[
\lambda_3 = p_3 + iq_3
\]

\[
\overline{\psi}_{12} + \overline{\psi}_{13} = x_1
\]

\[
\begin{bmatrix}
(\lambda_1 - p_3) - iq_3 & t_{12} & t_{13} \\
(t_{22} - p_3) - iq_3 & t_{23} & \\
t_{32} & (t_{33} - p_3) - iq_3 & \end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \end{bmatrix} = 0
\]

Assign \( x_3 \) a value of \((1 + i0)\). The second element, \( x_2 \), is determined from row three.
Equating the real part of $x_2$ with $\bar{\psi}_{22}$, and the imaginary part with $\bar{\psi}_{23}$, yields

$$\bar{\psi}_{22} = \frac{-(t_{33} - p_3)}{t_{32}}$$  \hspace{1cm} (F-9)$$

$$\psi_{23} = \frac{q_3}{t_{32}}$$  \hspace{1cm} (F-10)$$

To complete the third eigenvector, $\{\bar{\psi}_3\}$, $\bar{\psi}_{12}$ and $\bar{\psi}_{13}$ must also be determined.

$$[(\lambda_1 - p_3) - i q_3] x_1 = -t_{12} x_2 - t_{13} x_3$$

This equation may be broken down into two equations by equating the real and imaginary portions.

Real:

$$(\lambda_1 - p_3) \bar{\psi}_{12} + q_3 \bar{\psi}_{13} = -t_{12} \bar{\psi}_{22} - t_{13} \bar{\psi}_{32} = r$$

Imaginary:

$$(\lambda_1 - p_3) \bar{\psi}_{13} - q_3 \bar{\psi}_{12} = -t_{12} \bar{\psi}_{23} - t_{13} \bar{\psi}_{33} = s$$
Written in matrix form, it becomes

\[
\begin{bmatrix}
(\lambda_1 - p_3) & q_3 \\
-q_3 & (\lambda_1 - p_3)
\end{bmatrix}
\begin{bmatrix}
\psi_{12} \\
\psi_{13}
\end{bmatrix}
= \begin{bmatrix}
r \\
s
\end{bmatrix}.
\]

Using Cramer's Rule, \(\psi_{12}\) and \(\psi_{13}\) are found to be

\[
\psi_{12} = \frac{r (\lambda_1 - p_3) - q_3 s}{(\lambda_1 - p_3)^2 + (q_3)^2}
\]  \hspace{1cm} (F-11)

and

\[
\psi_{13} = \frac{(\lambda_1 - p_3) s + r q_3}{(\lambda_1 - p_3)^2 + (q_3)^2}.
\]  \hspace{1cm} (F-12)

The entire modal matrix, \(\Psi\), has been determined, and may be written as

\[
\Psi = \begin{bmatrix}
1 & \psi_{12} & \psi_{13} & \psi_{14} \\
\psi_{22} & \psi_{23} & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & \psi_{34} & 1
\end{bmatrix}.
\]  \hspace{1cm} (F-13)

**Case 3. All Complex Eigenvalues and Eigenvectors**

In this case, all the eigenvalues and their corresponding eigenvectors occur as complex conjugate pairs. The pseudo upper triangular matrix, \(T\), contains only \(2 \times 2\) sub-matrices on the diagonal. Using
λ₄, the complex eigenvector \{\bar{\psi}_4\} is found in the same manner as \{\bar{\psi}_3\} was found in Case 2. The real portion of \{\bar{\psi}_4\} is stored in \{\bar{\psi}_3\}; the imaginary portion is stored in \{\psi_4\}. Next, \lambda_2 is used to determine the complex eigenvector \{\bar{\psi}_2\} using the same procedure as outlined above. The final modal matrix, \bar{\psi}, will look like

\[
\bar{\psi} = \begin{bmatrix}
\bar{\psi}_{11} & \bar{\psi}_{12} & \bar{\psi}_{13} & \bar{\psi}_{14} \\
1 & 0 & \bar{\psi}_{23} & \bar{\psi}_{24} \\
& & \bar{\psi}_{33} & \bar{\psi}_{34} \\
& & 1 & 0
\end{bmatrix}
\]  

\text{(F-14)}
REFERENCES


