Research on Numerical Algorithms for Large Space Structures

Final Report

NASA Grant NSG-1603

March 15, 1980 - March 14, 1981

Department of Electrical Engineering

Cullen College of Engineering

University of Houston

Research on Numerical Algorithms for Large Space Structures

Final Report

NASA Grant NSG-1603

March 15, 1980 - March 14, 1981

Prepared by

Eugene D. Denman, Principal Investigator

University of Houston

Houston, Texas 77004

for

NASA Langley Research Center

Hampton, Virginia 23665

Table of Contents

1. Introduction

This report is the annual report on work done under NASA Grant NSG-1603, NASA Langley Research Center, Hampton, Virginia. The grant task is to investigate numerical algorithms for analysis, and design of large space structures. The report covers activities during the period March 16, 1980 to March 15, 1981. The report for the previous period, March 16, 1979 to March 15, 1980 was on matrix polynomials and their relationship to matrices of the form sI-A. The report during the more recent period uses that material to derive new results.

Section 2 of this report describes the sign algorithm and the application to decoupling of differential equations such as the finite element equation

$$M\ddot{x}(t)+C\dot{x}(t)+Kx(t) = u(t) \tag{1.1}$$

The algorithm is useful for the decoupling procedure but does not appear to be efficient for large scale structures. Work is continuing on finding new approaches to the decoupling (or model reduction) problem.

The work given in Section 2 is extended in Section 3 with new results. The generalized sign algorithm is given and its application to several problems discussed.

Sections 4 and 5 discusses the Laplace transforms of matrix functions and the diagonalization procedure for the equation given in (1.1). It is shown that the matrix

$$= \begin{bmatrix} 0 & I \\ -\overline{K} & -\overline{C} \end{bmatrix} \tag{1.2}$$

can be transformed to the form

$$BD = \begin{bmatrix} 0 & I \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \tag{1.3}$$

by a constant matrix Q. The matrix polynomial requires a matrix polynomial to diagonalize with

$$(Is^2+2\zeta\omega s+\omega^2) = Q(s)[Is^2+\bar{C}s+\bar{K}]P(s) \tag{1.4}$$

Extensions of the results in Section 5 will be studied during the period of the present grant.

Identification of the mass, damping and stiffness matrices, M, C and K, is considered in Sections 6 and 7. It is shown that the quadrature integration algorithm can be used to determine the elements of the matrices. It would appear that the algorithm is suitable for large space structures although the algorithm requires further examination. Actual data will be used with the algorithm to determine the accuracy of the procedure.

Some results have been obtained on the computation of the damping matrix C for prescribed eigenvalue locations. The work on this matter is not reported due to the timing of the report as well as the need for additional research. This problem will also be explored during the remaining period of the present grant.

An interim report was filed earlier on the Ibrahamin identification algorithm. Readers interested in that work should secure a copy from NASA Langley Research Center.

2. The Sign Matrix and Decoupling

There has been considerable interest in the analysis of dynamic systems by decoupling or order reduction, [1]-[4]. The decoupling of a system into subsystems with different time scales has been described earlier by Yoo, [5] and by Popeeva and Lupas, [6] where a block diagonalization scheme was used to separate the modes. The mathematical theory of the block diagonalization procedure was given by Russell, [7] in a earlier paper. The concept of separating "slow" and "fast" modes was the basic motivation for the work by Lee, [3], and Yackel and Kokotović [8] although their approach differs from the procedure given in [1]-[6] and the algorithm to be presented.

This section will describe a general procedure for decoupling of a system into subsystems. The number of modes in each subsystem is arbitrary with the eigenvalues arranged according to their magnitudes. The resulting subsystems will not be stiff even though the system may be stiff. The system dynamics is recovered from the subsystem dynamics by using the decoupling matrices. The method is closely related to the work of Kron, [9], and system decomposition, [10]. The concept as given here was first given by Yoo, [5].

The algorithm to be presented is based on the solution of algebraic Riccati equations and the sign function, [4], conceived by Roberts. The sign function and its application to system analysis has been investigated by Beavers and Denman [11]-[15], as well as by others, [16]-[21]. It can be shown that the spectral domain $\rho(A)$ of a system matrix can be decomposed into spectral subdomains, $\rho_i(A)$, by the sign matrix of A with the union of $\rho_i(A)$ covering the domain $\rho(A)$. The eigenvalues of A remain invariant under the decomposition operation and the eigenprojectors [22], for the subdomains

can also be generated as a byproduct of the decomposition.

The sign function for a system matrix will be described in this papers as well as the decomposition procedure. The concept of eigenprojectors will be introduced as well as the numerical use of eigenprojectors.

The system equation to be considered will be of the form

$$M\frac{d^2x(t)}{dt^2} + C\frac{dx(t)}{dt} + K\,x(t) = f(t) \tag{2.1}$$

where M, C and K are the mass, damping and stiffness matrices respectively. The vectors x(t) and f(t) are n×1 with M, C , and K n×n matrices. The system matrix will be written as

$$A = \begin{bmatrix} 0 & I \\ -\bar{K} & -\bar{C} \end{bmatrix} \tag{2.2}$$

where A is 2n×2n with $\bar{K} = -M^{-1}K$, and $\bar{C} = M^{-1}C$. The system matrix A will have 2n eigenvalues and it will be assumed that all eigenvalues in the spectrum $\rho(A)$ occur in conjugate pairs with an even number of zero values.

The form of system equation assumed occurs in structural dynamics. The procedure will be valid for A in general form, the above form was chosen since the procedure is applied to structural dynamics in the current research effort.

Section 2.1 will describe the sign function which has an important role in the decoupling algorithm. The eigenprojectors, or projectors, will be introduced in Section 2.2 along with the application of eigenprojectors to numerical analysis. Decomposition of the spectrum of A into subdomains will be discussed in Section 2.3. The application of the decoupling algorithm to system equations is given in Section 2.4.

2.1 The Matrix Sign Function and Projectors

Let A be a 2n×2n matrix with 2n complex eigenvalues where λ_i is the ith eigenvalue. Assume that J denotes the Jordan form of the eigenvalue matrix with Jordan blocks J_i for $i = 1,2,\ldots,s\leq 2n$. An eigenvalue may be repeated with multiplicity r_i and the A matrix may have a null space less than 2n requiring $s\ell_i$ generalized eigenvectors where ℓ_i denotes the number of generalized eigenvectors for λ_i and s indicates the number of Jordan blocks of A.

The definition of a functior of a matrix is given by Gantmacher, [21]. If A and J are similar matrices such that Φ transforms J into A,

$$A = \Phi J \Phi^{-1} \quad . \tag{2.1.1}$$

Then the matrix functions f(A) and f(J) are similar and Φ transforms f(J) into f(A);

$$f(A) = \Phi f(J) \Phi^{-1} \quad . \tag{2.1.2}$$

Defining the sign of a complex function as the sign of the real part by the complex function, then sign (A) is given by

$$S = \text{sign}(A) = \Phi \text{sign}(J) \Phi^{-1} \tag{2.1.3}$$

where Φ is the matrix of eigenvectors of A. Since J is the Jordan form, the eigenvalues of A will be located on the diagonal of J with plus ones on the off-diagonal of J provided that such is required for multiple eigen-

values. The sign of J will be defined as

$$\text{sign}(J) = \text{sign}(\Lambda) \tag{2.1.4}$$

where $\Lambda = \text{diag}[J]$ thus S is defined as

$$S = \text{sign}(A) = \Phi\, \text{sign}[\text{Re}(\Lambda)]\Phi^{-1} \tag{2.1.5}$$

Roberts, [4], has shown that S can be computed by the iterative algorithm

$$S^{i+1} = \frac{1}{2}\,[S^i + (S^i)^{-1}] \qquad S^0 = A \tag{2.1.6}$$

which is the Newton algorithm for $S^2 = I$. This algorithm will have quadratic convergence and the iterative process can be terminated when trace $(S^2)-n<\varepsilon$. It should be noted that the sign of $\text{Re}(\lambda_i)$ does not exist when $\lambda_i = 0$ or $\lambda_i = j\omega_i$. A method of computing the sign under such an eigenvalue assignment will be described later.

The eigenprojectors of a matrix are defined as the matrices

$$P_{i0} = \Phi E_{i0} \Phi^{-1} \qquad i = 1,2,\ldots,s\leq 2n \tag{2.1.7}$$

for the S Jordan blocks and secondary eigenprojectors P_{ij} with

$$P_{ij} = \Phi E_{ij} \Phi^{-1} \qquad j = 1,2,\ldots,\ell_i \tag{2.1.8}$$

for the repeated eigenvalues λ_i where ℓ_i is the number of generalized eigenvectors for that eigenvalue. The matrix E_{i0} is given by

(2.1.9) $$E_{i0} = \text{diag}[0\ 0\ \ldots\ 0\ 1\ \ldots\ 1\ 0\ \ldots\ 0]$$

where the ones occupy the same positions as λ_i does in the ith Jordan block. The matrices E_{ij} are defined as the set of matrices with ones on the off-diagonals of the matrix. To illustrate, let J_i be defined as

(2.1.10) $$J_i = \begin{bmatrix} \lambda_i & 1 & 0 \\ 0 & \lambda_i & 1 \\ 0 & 0 & \lambda_i \end{bmatrix}$$

then

(2.1.11) $$E_{i0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad E_{i1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad E_{i2} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The E_{i0} matrix can be expressed in the compact notation

(2.1.12) $$E_{i0} = \text{diag}[0,\ 0,\ 0\ \ldots\ 0,\ \hat{E}_{i0},\ 0,\ \ldots\ 0]$$

with a similar expression for E_{ij} but where E_{ij} will have ℓ_i representations, $E_{i0}, E_{i1}, \ldots, E_{i,\ell_i}$.

The primary eigenprojectors, P_{i0} will have rank r_i where r_i is the multiplicity of the λ_i eigenvalue. The secondary eigenprojectors, P_{ij} $j = 1,2,\ldots,\ell_i$, will have rank $r_i - j$. The eigenprojectors are idempotent and have properties;

$$\text{2P1.} \quad \sum_{i=1}^{S} P_{i0} = I$$

$$\text{2P2.} \quad P_{i0}P_{i0} = P_{i0} \qquad \text{(idempotent)}$$

$$\text{2P3.} \quad P_{i0}P_{j0} = 0 \qquad i \neq j$$

$$\text{2P4.} \quad P_{ij}P_{ij} = P_{i,j+1} \qquad j = 1,2,\ldots,\ell_i-1$$

$$\text{2P5.} \quad P_{ij}P_{i,j+1} = 0 \qquad j = 1,2,\ldots,\ell_i-1 \ .$$

It is not difficult to show that any square matrix A has a spectral decomposition

$$\text{(2.1.13)} \qquad A = \sum_{i=1}^{s} A_i = A \sum_{i=1}^{s} P_{i0} = \sum_{i=1}^{s} \lambda_i P_{i0} + \sum_{i=1}^{s} P_{i1}$$

where P_{i0} is the set of primary eigenprojectors and P_{i1} is the first secondary eigenprojector of a Jordan block if it exist.

The eigenprojectors of A, which is assumed to be time-invariant, can be computed by repeated computations of the sign matrix. Assume that m eigenvalues of A have eigenvalues with $\mathrm{Re}(\lambda_i)>0$ and 2n-m eigenvalues have $\mathrm{Re}(\lambda_i)<0$. This spectral splitting of the spectral domain can be achieved in several ways, origin shifts, bilinear transformations, etc. The sign of A will then have the form

$$\text{(2.1.14)} \qquad S = \Phi \begin{bmatrix} I_{m\times m} & 0 \\ 0 & -I_{2n-m\times 2n-m} \end{bmatrix} \Phi^{-1}$$

The lower identity matrix will vanish if 0.5(I+S) is computed whereas the upper identity block will not be present in 0.5(I-S). The projector for

the positive and negative spectra are then given by

$$P^{+} = \frac{1}{2}(I+S) \tag{2.1.15}$$

$$P^{-} = \frac{1}{2}(I-S) \ . \tag{2.1.16}$$

The products AP^{+} and AP^{-} gives A^{+} and A^{-} respectively where the nonzero eigenvalues of A^{+} have $Re(\lambda_i)>0$ and those of A^{-} have $Re(\lambda_i)<0$. The spectra of A^{+} and A^{-} can now be altered by an origin shift and the signs computed. Repeated computation of the sign matrix, origin shift, and spectral decomposition will give the eigenprojectors of all Jordan blocks if desired.

2.2 Decomposition of the Spectrum of A

The spectrum of a general n×n matrix A can be decomposed into m matrices with subspectra $\rho_i(A)$ such that $\rho(A) = \rho_1(A) \cup \rho_2(A) \ldots \cup \rho_m(A)$ where $m \leq s$. Each matrix A_i will have an associated set of r finite eigenvalues depending on the spectral decomposition as well as n-r zero eigenvalues. The decomposition of A is given by

$$A = \sum_{i=1}^{s} \lambda_i P_{i0} + \sum_{i=1}^{s} P_{i1} = \sum_{i=1}^{m} A_i \tag{2.2.1}$$

where A_i can be any partial sum over a selected set of eigenvalues. The decomposition of A may be by eigenvalue magnitudes, reals, complex, imaginary or any other desired eigenvalue property.

Suppose that all eigenvalues with magnitudes $\rho_{i-1} < |\lambda_j| < \rho_i$ are to be assigned to a spectral domain $\rho(A_j)$, then A can be decomposed into m matrices A_i such that

$$A_i = \sum_{j=k}^{p} A_j \qquad i = 1,2,\ldots,m \tag{2.2.2}$$

with

$$A = \sum_{i=1}^{m} A_i = A \sum_{i=1}^{m} P_i \tag{2.2.3}$$

Each A_i will be 2n×2n with 2n-(p-k) zero eigenvalues and p-k nonzero eigenvalues in the spectral domain $\rho_i(A)$.

The spectral decomposition described above is a decomposition of A into m 2n×2n matrices. It is also possible to find m square matrices of lower

dimension that will preserve the spectral domain of A. Let the eigenvector matrix Φ be partitioned as

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \tag{2.2.4}$$

where Φ_{11} and Φ_{22} are square matrices of dimension m×m and 2n-m×2n-m respectively. Define the transformation matrix T as

$$T = \frac{1}{2}\,\Phi\{\mathrm{diag}[\Phi_{11}^{-1} \cdots \Phi_{22}^{-1}]\} = \frac{1}{2}\begin{bmatrix} I & -\Phi_{12}\Phi_{22}^{-1} \\ \Phi_{21}\Phi_{11}^{-1} & -I \end{bmatrix} \tag{2.2.5}$$

The similarity transformations $T^{-1}AT$ and TAT^{-1} will produce block diagonalized matrices. The first transformation $T^{-1}AT$ gives

$$\begin{aligned} A_B = T^{-1}AT &= \mathrm{diag}[\Phi_{11}\text{-}\Phi_{22}]J\ \mathrm{diag}[\Phi_{11}^{-1}\text{-}\Phi_{22}^{-1}] \\ &= \mathrm{diag}[A_{B1}\ A_{B2}] \end{aligned} \tag{2.2.6}$$

The A_{Bi} submatrices are equal to $\Phi_{ii}\ J_{Bi}\ \Phi_{ii}^{-1}$ where J_{Bi} contains the Jordan blocks associated with the eigenvalues belonging to the set of eigenvectors in Φ_{ii}. If J_{B1} has eigenvalues $|\lambda_i|>\rho_1$ and J_{B2} has all eigenvalues with $|\lambda_i|<\rho_1$ then A_{B1} has only "fast" modes with the "slow" modes in A_{B2}.

The transformation matrix T can be constructed from the eigenprojectors of A. Let P^+ denote the sum of the eigenprojectors P_{i0} for eigenvalues $|\lambda_i|>\rho_1$ and P^- the sum of the remaining eigenprojectors. The block eigenprojector P^+ is given by

$$P^{+} = \sum_{i=1}^{k} P_{i0} = \Phi E_{B1} \Phi^{-1} = \Phi I_1 \Phi^{-1} \tag{2.2.7}$$

where $I_1 = \text{diag}[I,0] = \text{diag}[E_{B1},0] = \text{diag}[\sum_{i=1}^{k} E_{i0},0]$. The indexing over the summation assumes that the eigenvalues of A have been arranged by descending magnitudes. Since the sum of the eigenprojectors must be equal to the identity matrix, then

$$P^{-} = \sum_{i=k+1}^{2n} P_{i0} = \Phi I_2 \Phi^{-1} = (I-P^{+}) \tag{2.2.8}$$

It is not difficult to show that the sign of A can be computed from the eigenprojector, or the eigenprojectors can be determined from the sign matrices. Assume that k eigenvalues have $\text{Re}(\lambda_i)>0$ and 2m-k eigenvalues have $\text{Re}(\lambda_i)<0$. If P^{+} is the sum of the eigenprojectors for $\text{Re}(\lambda_i)>0$ and P^{-} the sum of the eigenprojectors for $\text{Re}(\lambda_i)<0$ then

$$S = \text{sign}(A) = P^{+}-P^{-} = \sum_{i=0}^{k} P_{i0} - \sum_{i=k+1}^{2n} P_{i0} \tag{2.2.9}$$

where P_{i0} is the individual eigenprojector for a Jordan block. Noting that $P^{-} = I-P^{+}$, then

$$S = \text{sign}(A) = 2P^{+}-I \tag{2.2.10}$$

or

$$P^{+} = \frac{1}{2}(S+I) \tag{2.2.11}$$

with P^{-} determined from P^{+}. The eigenprojectors P_{i0} can be computed by

utilizing origin shifts or bilinear transformations. These procedures will be discussed later.

The transformation matrix T was defined in (2.2.5) and this matrix can be computed from the sign matrix of the eigenprojectors. According to (2.2.9), the sign of A is given by P^+-P^- thus

$$S = P^+-P^- = \Phi \text{ sign}[\text{Re}(\Lambda)]\Phi^{-1} \tag{2.2.12}$$

provided that $\text{Re}(\lambda_i) \neq 0$ for all i. Let sign $[\text{Re}(\Lambda)] = E = \text{diag}[I_1,-I_2]$ then

$$T = [S+E]^{-1} = [\Phi E\Phi^{-1}+E]^{-1} = \Phi[\Phi E+E\Phi]^{-1} \tag{2.2.13}$$

since $[\Phi E+E\Phi]^{-1} = \text{diag}[\Phi_{11}^{-1}-\Phi_{22}^{-1}]$. It should be pointed out that the signs of (2.2.5) and (2.2.6) can be reversed.

The analysis thus far has considered only block diagonalization for two subblocks. It is not difficult to show that the procedure is valid for s blocks, the number of Jordan blocks in A. The limiting case is equivalent to scalar diagonalization of A in which case $A_B = \text{diag}[\lambda_1,\lambda_2,\ldots,\lambda_{2n}]$ when the eigenvalues are distinct.

Knowledge of any one of the matrices, T, S, P^+ or P^-, is sufficient to completely determine the other three matrices for two block diagonalization. If m blocks are to be found, then additional information must be available from the eigenprojectors or the sign matrix. Details on the spectral decomposition will be found in [23].

2.3 Matrix Sign Functions and Block Diagonalization

It was shown in the previous section that spectral decomposition and block diagonalization is possible if the eigenprojectors are known. It will be shown in this section how the matrix sign function is utilized for carrying out the block diagonalization.

Assume that A has real and complex eigenvalues and A is to be block diagonalized. It will be assumed that the spectral domains are such that $|\lambda_i|<\alpha$ will be in $\rho(A_{B1})$ and $|\lambda_i|>\alpha$ will belong to $\rho(A_{B2})$. The only restriction on the procedure is that α be selected such that there is no eigenvalue with magnitude equal to α. The bilinear transformation will map

$$A_D = (A-\alpha I)(A+\alpha I)^{-1} \tag{2.3.1}$$

all eigenvalues $|\lambda_i|<\alpha$ into the left half plane of the complex space and all $|\lambda_i|>\alpha$ into the right half plane. The spectral domain of $\rho_1(A_D)$ is the entire left half plane and that of $\rho_2(A_D)$ is the right half plane with $\rho(A_D) = \rho_1(A_D) \cup \rho_2(A_D)$. The sign of A_D is then computed by (2.6) with

$$S = \text{sign}\,(A_D) = \Phi\{\text{diag}[\hat{I}_1 \quad \hat{I}_2]\}\Phi^{-1} . \tag{2.3.2}$$

The matrices $\hat{I}_1$ and $\hat{I}_2$ will not be ordered in the general case with ± 1 intermixed along the diagonal of the two matrices. The reason for this is that the eigenvalues of A are not ordered and the iterative algorithm for the sign of A_D is independent of the order.

Assuming that I_1 has only plus ones which indicates an eigenvalue $|\lambda_i|>\alpha$ and I_2 has only minus ones for $|\lambda_i|<\alpha$ then S can be written as

$$S = \text{sign}\,(A_D) = \Phi E \Phi^{-1} \tag{2.3.3}$$

where E is an elementary matrix. It follows that

$$S+E = \Phi E \Phi^{-1} + E = [\Phi E + E\Phi]\Phi^{-1} \tag{2.3.4}$$

$$= \begin{bmatrix} 2\Phi_{11} & 0 \\ 0 & -2\Phi_{22} \end{bmatrix} \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}^{-1}$$

and the inverse of S+E is

$$T = [S+E]^{-1} = \frac{1}{2}\,\Phi \begin{bmatrix} \Phi_{11}^{-1} & 0 \\ 0 & -\Phi_{22}^{-1} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} I & -R_{12} \\ R_{21} & -I \end{bmatrix} \tag{2.3.5}$$

where $R_{12} = \Phi_{12}\Phi_{22}^{-1}$ and $R_{12} = \Phi_{21}\Phi_{11}^{-1}$.

The similarity transformation $T^{-1}AT$ will produce a block diagonalized matrix similar to A, [24], with

$$A_B = \begin{bmatrix} A_{B1} & 0 \\ 0 & A_{B2} \end{bmatrix} = \begin{bmatrix} A_{11}+A_{12}R_{21} & A_{12}+A_{11}R_{12}-R_{12}A_{22}-R_{12}A_{21}R_{12} \\ A_{21}+A_{22}R_{21}-R_{21}A_{11}-R_{21}A_{12}R_{21} & A_{22}+A_{21}R_{12} \end{bmatrix} \tag{2.3.6}$$

where R_{12} and R_{21} are the solutions to the two algebraic Riccati equations, [13],

$$A_{12}+A_{11}R_{12}-R_{12}A_{22}-R_{12}A_{21}R_{12} = 0 \tag{2.3.7}$$

$$A_{21}+A_{22}R_{21}-R_{21}A_{11}-R_{21}A_{12}R_{21} = 0 \tag{2.3.8}$$

The solutions to these equations are given by the off diagonal blocks of T as seen in (2.3.5). It is a simple matter to show that the mode decoupling algorithm of Anderson, [1] is a special case of the above analysis requiring a two step process, computation of R_{12} (or R_{21}) and the solution to an associated Lypanov equation.

The diagonalization procedure given above was for diagonalization of A into two blocks with the eigenvalues $|\lambda_i|>\alpha$ in A_{B1} and all others in A_{B2}. Suppose now that the spectral domain is to be factored into three domains with $|\lambda_i|<\alpha_1$ in $\rho_1(A)$, $\alpha_1<|\lambda_i|<\alpha_2$ in $\rho_2(A)$ and $|\lambda_i|>\alpha_2$ in $\rho_3(A)$. The procedure is the same as described for the two block decomposition except that A_{B1} is now used in the bilinear transformation with

$$A_{D1} = (A_{B1}-\alpha_2 I)(A_{B1}+\alpha_2 I)^{-1} . \tag{2.3.9}$$

The sign of A_{D1} is computed and E_1 for the sign (A_{D1}) is then used to find $\hat{T}_1$ with

$$\hat{T}_1 = (S_1+E_1) \tag{2.3.10}$$

The transformation matrix T_1 is augmented with an identity matrix I having the dimensions of A_{B2} with

$$T_1 = \begin{bmatrix} \hat{T}_1 & 0 \\ 0 & -I \end{bmatrix} \tag{2.3.11}$$

If T_0 denotes the first transformation matrix and T_1 the second, then the

product must be

$$(2.3.12) \qquad T = T_0 T_1 = \frac{1}{4} \begin{bmatrix} I & -R_{12} & R_{13} \\ R_{21} & -I & R_{23} \\ R_{31} & -R_{32} & I \end{bmatrix}$$

from which it follows that

$$(2.3.13) \quad A_B = \begin{bmatrix} A_{11}+A_{12}R_{21}+A_{13}R_{31} & 0 & 0 \\ 0 & A_{22}+A_{21}R_{12}+A_{23}R_{32} & 0 \\ 0 & 0 & A_{33}+A_{31}R_{13}+A_{32}R_{23} \end{bmatrix}$$

The R_{ij} matrices satisfy coupled algebraic Riccati equations, [16].

An example is given to illustrate the above procedure. Let A be defined as [25]

$$A = \begin{bmatrix} 5 & 1 & -2 & 0 & -2 & 5 \\ 1 & 6 & -3 & 2 & 0 & 6 \\ -2 & -3 & 8 & -5 & -6 & 0 \\ 0 & 2 & -5 & 5 & 1 & -2 \\ -2 & 0 & -6 & 1 & 6 & -3 \\ 5 & 6 & 0 & -2 & -3 & 8 \end{bmatrix}$$

with eigenvalues $\lambda_1,\lambda_2 = -1.598734$, $\lambda_3,\lambda_4 = 4.455989$ and $\lambda_5,\lambda_6 = 16.142744$. Values of $\rho_1 = 2$ and $\rho_2 = 8$ were selected to split the spectrum. The block matrices are

$$A_{B1} = \begin{bmatrix} 16.1427 & 0 \\ 0 & 16.1427 \end{bmatrix}$$

$$A_{B2} = \begin{bmatrix} 4.45599 & 0 \\ 0 & 4.45599 \end{bmatrix}$$

$$A_{B3} = \begin{bmatrix} -1.59873 & 0 \\ 0 & -1.59873 \end{bmatrix}$$

The transformation matrix for the diagonalization is

$$T = \begin{bmatrix} 0.25 & 0. & 0.595655 & 0.407491 & 0.530013 & 0.120331 \\ 0. & 0.25 & -0.4012 & -0.431213 & -0.407491 & -0.37346 \\ 1.51992 & -1.2999 & -0.25 & 0. & -0.397965 & -0.167369 \\ -0.116553 & 0.0515402 & 0 & -0.25 & 0.231053 & 0.0710525 \\ -0.678077 & 0.641664 & -0.0845469 & 0.214427 & 0.25 & 0. \\ -0.0496661 & 0.379825 & -0.0885163 & 0.0860365 & 0. & 0.25 \end{bmatrix}$$

which is Φ since A is diagonalized. The computations were made in double precision.

An important aspect of the integration procedure is controllability of a mode. It can be shown that the system modes are controllable if and only if the vectors in B spans the space of A. Let B be $2n \times m$ with vectors b_i such that

$$B = [b_1 \; b_2 \; \ldots \; b_m] \tag{2.3.14}$$

The mode $\exp(\lambda_i t)$ is controllable if the sum of the column vectors in B is linear dependent on the eigenvector for λ_i. To show this, consider the Laplace transform of $\dot{z}(t) = Az(t)+Bf(t)$ which is

$$(2.3.15) \qquad Z(s) = [sI-A]^{-1}BF(s) = \{ \sum_{i=1}^{s} \frac{K_{i0}}{s-s_i} + \sum_{j=1}^{r} \frac{K_{ij}}{(s-s_i)^j} \} BF(s)$$

where K_{i0} denotes the primary matrix residues

$$(2.3.16) \qquad K_{i0} = (s-s_i)^r [sI-A]^{-1} \Big|_{s=s_i} \qquad i = 1,2,\ldots,s$$

and K_{ij} the secondary matrix residues

$$(2.3.17) \qquad K_{ij} = \frac{1}{j!} \frac{d^j}{d\zeta} \{(s-s_i)^r [sI-A]^{-1}\} \Big|_{s=s_i} \qquad j = 1,2,\ldots,r-1$$

where r is the multiplicity of an eigenvalue. It then follows that if $K_{i0}B$ and $K_{ij}B$ are zero for all j, the ith mode will not appear in z(t). Since the columns of B can always be constructed from a linear combination of the eigenvectors of A, which must span a 2n dimensional space, and since the residues are eigenprojectors, the absence of the ith eigenvector in B will make $K_{i0}B$ and $K_{ij}B$ zero. Modes that are not controllable will require the inclusion of f(t).

If f(t) is rapidly varying function with respect to a mode, the integration step will be set accordingly.

Since A has the special form given in (2.2), the first n elements of z(t) will be the displacements x(t) and the last n elements of the vector will be $\dot{x}(t)$. If only x(t) is desired, it is easily shown that

$$x(t) = \frac{1}{2m-1} [v_1(t) - R_{12}v_2(t) + R_{13}v_3(t) \ldots] \tag{2.4.4}$$

where m is the number of diagonal blocks in A_3.

If f(t) is rapidly varying function with respect to a mode, the integration step will be set accordingly.

Since A has the special form given in (2.2), the first n elements of z(t) will be the displacements x(t) and the last n elements of the vector will be $\dot{x}(t)$. If only x(t) is desired, it is easily shown that

$$x(t) = \frac{1}{2^{m-1}} [v_1(t) - R_{12}v_2(t) + R_{13}v_3(t) \ldots] \tag{2.4.4}$$

where m is the number of diagonal blocks in A_3.

2.5 Application to Free-Free Beam Model

The algorithm was applied to a free-free beam model generated by NASA LRC. The beam was 144 inches in length, the mass per unit length was 0.112 lbs with the EI constant equal to 6.25×10^5. The nodes were equi-spaced with spacing ℓ_i with the two end nodes located at a distance of 1/2 ℓ_i from the ends. A total of 21 nodes was assumed although the computer model allowed the number of nodes to be changed.

The decoupling algorithm was used to separate the highest frequency mode from all others. In addition to obtaining the solution by decoupling, the initial conditions were chosen such that only the highest mode was excited. The full set of equations were then integrated to obtain a benchmark solution for comparison to the decoupled solution. A step size of h=0.0002 was selected for the two computed solutions. The displacement for the highest mode is given below at three selected times where v(t) is the transformed solution vector $v(t) = T^{-1}z(t)$, as given in Section 2.4 where z(t) is the displacement of the free-free beam nodes.

t	v_{21}(t)(Full model)	v_{21}(t)(Decoupled)
0.002	0.584225754E-3	0.58425754E-3
0.004	-0.11151369E-2	-0.11151369E-2
0.006	-0.13946460E-2	-0.13946460E-2

The difference between the two solutions was in the 11th digit thus it can be assumed that the decoupling algorithm performs in a satisfactory manner.

Test of the algorithm has been carried out for several other models and the accuracy was excellent. There is no reason to believe that the algorithm fails for any model, the major problem with the algorithm is the computational overhead in computing the decoupling matrix T. Work continues on the algorithm but it appears that the algorithm is not suitable for large space structures in its present form.

$$x(0) = (P^{+}+P^{-}+P^{I}+P^{0})x(0) = x^{+}(0)+x^{-}(0)+x^{I}(0)+x^{0}(0) \tag{3.6.5}$$

The solution to the system equation is then given by

$$x(t) = \exp(A\,t)x^{+}(0)+\exp(A\,t)x^{-}(0)+\exp(A\,t)x^{I}(0)+x^{0}(0) \tag{3.6.6}$$

where the properties of the eigenprojectors has been utilized to obtain (3.6.6) from (3.6.4) and (3.6.5). Equations (3.6.4)-(3.6.6) are the mathematical expressions for the decoupling concept. It is possible to continue the decoupling process further such that the decoupled solution vectors include "fast" and "slow" modes although that procedure will not be given since it has been described elsewhere, [5] and Section 2 of this report.

The following matrix has been used for illustration of the algorithm where A has the following eigenvalues $\lambda_1 = 1.0$ $\lambda_2 = -3.0$ $\lambda_3 = +2j$ $\lambda_4 = -2j$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 12 & -8 & -1 & -2 \end{bmatrix} \tag{3.6.7}$$

and the initial conditions for the system $\dot{X} = AX$ are given by $X(0) = [-1.0, 4.0, 1.0, -3.0]^{T}$. The resultant eigenprojectors are

$$P^{+} = \begin{bmatrix} 0.6 & 0.2 & 0.15 & 0.05 \\ 0.6 & 0.2 & 0.15 & 0.05 \\ 0.6 & 0.2 & 0.15 & 0.05 \\ 0.6 & 0.2 & 0.15 & 0.05 \end{bmatrix}$$

3. Generalized Sign Matrix

Applications of the matrix sign function have been previously given in [15]. The matrix sign function is a useful mathematical tool for spectral decomposition as it provides insight on the spectrum of the matrix even though the eigenvectors and eigenvalues are unknown.

The material given in this paper is an extension of earlier work with the concept of a generalized sign matrix introduced and discussion of several applications. Knowledge of the generalized sign matrix is sufficient to construct the eigenprojectors on the spectral domain of a $n \times n$ matrix. The generalized sign matrix appears to have been introduced by Mattheys, [19] as the extended matrix sign function. The computation of the generalized sign matrix is given in the next section.

3.1 The Generalized Sign Function of a Matrix

A general n×n A matrix can be represented by the form

$$A = MJM^{-1} \tag{3.1.1}$$

where M is the eigenvector matrix of A and J is the Jordan canonical form with the eigenvalues along the diagonal and ones or zeros on the superdiagonal. If J is expressed as

$$J = \text{diag}[J(\lambda_1), J(\lambda_2), \ldots, J(\lambda_r)] \tag{3.1.2}$$

Then any function f(A) which is analytical in the plane containing the eigenvalues λ_i of A is defined [3] by

$$f(A) = M[f(J)]M^{-1} \tag{3.1.3}$$

where

$$f(J) = \text{diag}[f(J_1), f(J_2), \ldots, f(J_r)] \tag{3.1.4}$$

with each Jordan block having the form

$$f(J_i) = \begin{bmatrix} f(\lambda_i) & f'(\lambda_i)/1! & \cdots & f^{(n_i-1)}(\lambda_i)/(n_i-1)! \\ 0 & f(\lambda_i) & \cdots & f^{(n_i-2)}(\lambda_i)/(n_i-2)! \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & f(\lambda_i) \end{bmatrix} \tag{3.1.5}$$

If the sign function for a Jordan block is defined as the function with the sign of the real part of $f(\lambda_i)$, it then follows that

$$\text{sign}(A) = M\ \text{sign}[R_e J]M^{-1} \tag{3.1.6}$$

where sign $[R_e J]$ = diag $[\text{sign}(R_e\lambda_1)\ \text{sign}(R_e\lambda_2)\ldots\text{sign}(R_e\lambda_r)]$.
with

$$\text{sign}(R_e\lambda_i) = \begin{cases} +1 & R_e(\lambda_i)>0 \\ -1 & R_e(\lambda_i)<0 \end{cases} \tag{3.1.7}$$

The definition of the sign of A has a restriction in the sense that the sign of zero or imaginary eigenvalue is not defined. To account for this case define the generalized sign matrix [19] as

$$\text{sign}(\lambda_i) = \begin{cases} 1 & R_e(\lambda_i)>0 \\ 0 & R_e(\lambda_i)=0 \\ -1 & R_e(\lambda_i)<0 \end{cases} \tag{3.1.8}$$

with the generalized sign matrix of A denoted by $\hat{S}(A)$ with

$$\hat{S}(A) = M\ \text{sign}(J)M^{-1} \tag{3.1.9}$$

An efficient algorithm to find the sign of a matrix has been given by Roberts [4] which is based on Newton's algorithm for $S^2 = 1$. The iterative algorithm

$$S_{i+1} = \frac{1}{2}\ (S_i+S_i^{-1}) \qquad S_0 = A \tag{3.1.10}$$

has quadratic convergence with convergence in approximately k+3 iterations where $2^k > \{\max \lambda_i \text{ and } (\min \lambda_i)^{-1}\}$. The stopping criterion $|\text{Tr } S^2 - n| \leq \epsilon$ has been used by the authors where ϵ is a preselected error bounds. Several versions of accelerated convergence algorithms have been given in the literature [4], [20].

It should be noted that the iterative algorithm in (3.1.10) will fail when A has a zero or imaginary eigenvalue. The singularity encountered in (3.1.10)can be avoided by a translation of the spectrum or an origin shift by adding ρI to A with

$$A+\rho I = M\Lambda M^{-1}+\rho MM^{-1} = M(\Lambda+\rho I)M^{-1} \tag{3.1.11}$$

where the direction of the shift is set by the sign of ρ. There is no assurance that a random choice of ρ will not produce a new singularity thus care should be taken in selecting ρ.

Assume that ρ is selected such that $\rho < |R_e \lambda_i|$ for all eigenvalues having a nonzero real part and further, let $\Lambda+\rho I$ and $\Lambda-\rho I$ be nonsingular. Let $S(\Lambda_1)$ and $S(\Lambda_2)$ denote the signs of $A+\rho I$ and $A-\rho I$ respectively. The generalized sign of A is then given by

$$\hat{S}(\Lambda) = \frac{(S(A_1)+S(A_2))}{2} \tag{3.1.12}$$

The computational strategy is to move all eigenvalues along the jw axis into the right half plane for $S(A_1)$ and into the left hand plane for $S(A_2)$. The eigenvalues with $R_e(\lambda_i) \neq 0$ must remain in the same spectral domains under the shifts. The major task associated with computing the generalized sign is the selection of ρ.

3.2 Eigenprojectors of the Sign Spectrum Decomposition

The ability to compute the eigenprojector from the generalized sign matrix makes the sign matrix a useful tool for system analysis. The availability of the eigenprojectors makes it possible to generate new system matrices based on various mathematical operations on the eigenvalues of the system. The eigenvectors will remain invariant under the operations, [22].

Suppose that the eigenprojectors are defined to cover the entire spectrum and let the eigenprojectors be denoted by P^+, P^-, P^0 and P^I with

3.2D.1 P^+ Positive eigenprojector - the projection on the positive real part of the eigenvalues of the matrix; $R_e(\lambda_i)>0$.

3.2D.2 P^- Negative eigenprojector - the projection on the negative real part of the eigenvalues of the matrix; $R_e(\lambda_i)<0$.

3.2D.3 P^0 Null eigenprojector - the projection on the zero eigenvalues of the matrix.

3.2D.4 P^I Imaginary eigenprojector - the projection on the imaginary eigenvalues of the matrix.

when these eigenprojectors are applied to the A matrix, a set of subspaces will be obtained with

$$A^+ = AP^+ = P^+A$$

$$A^- = AP^- = P^-A$$

$$A^0 = AP^0 = P^0A$$

$$A^I = AP^I = P^IA$$

with $A = A^+ + A^- + A^0 + A^I$. It should be obvious that A^0 is a zero matrix but P^0 will not be a zero matrix. The eigenvalues of the eigenprojectors will be zeros and ones with the same eigenvectors as A. The definition of these

projectors as eigenprojectors follows from the eigenvector property. The eigenprojectors have the basic properties

3.2P.1 the sum of the eigenprojectors is the identity matrix, $P^0+P^I+P^-+P^+=I$

3.2P.2 the eigenprojectors are orthogonal, i.e. $P^-P^+ = 0$

3.2P.3 the eigenprojectors are idempotent matrices, i.e. $P^+P^+ = P^+$.

These three properties are useful in system analysis and provide the essential information for spectral decomposition of A.

3.3 Computation of the Eigenprojectors

The eigenprojectors defined in Section3.2can be computed in a straightforward manner from the generalized sign matrix. The generalized sign matrix must be such that:

$$\hat{S}(A) = P^{+}-P^{-} \tag{3.3.1}$$

Since all eigenvalues along the jw axis will have $\text{sign}(R_e\lambda_i) = 0$. The square of the generalized sign matrix must be

$$\hat{S}^2(A) = P^{+}+P^{-} \tag{3.3.2}$$

Since the eigenprojector are orthogonal and idempotent matrices. It then follows that P^{+} and P^{-} are given by

$$P^{+} = \frac{\hat{S}^2(A)+\hat{S}(A)}{2} \tag{3.3.3}$$

$$P^{-} = \frac{\hat{S}^2(A)-\hat{S}(A)}{2} \tag{3.3.4}$$

Recognizing that the sum of the eigenprojectors must give the identity matrix, it follows that

$$P^{0}+P^{I} = I-P^{+}-P^{-} = I-\hat{S}^2(A) \tag{3.3.5}$$

To compute P^{0} and P^{I}, the eigenvalues in the domains of P^{+} and P^{-} are eliminated by multiplying (3.3.5) by A thus

$$\hat{A}_1 = A^0+A^I = A(P^0+P^I) = A(I-\hat{S}^2(A)) \tag{3.3.6}$$

where $A^0 = 0$. The square of $\hat{A}_1$ will have only real eigenvalues therefore the matrix $\hat{A}_2$ defined by

$$\hat{A}_2 = A+\hat{A}_1^2 = A+A^2(I-\hat{S}^2(A))^2 = A+A^2(I-\hat{S}^2(A)) \tag{3.3.7}$$

will have a spectrum with only real, complex and zeroeigenvalues. All imaginary eigenvalues $\lambda_i = j\omega_i$ will have been shifted by $-\omega_i^2$. The original eigenvalues with $R_e(\lambda_i) \neq 0$ will be preserved in the operation. The eigenprojector for the imaginary spectral domain will then be given by

$$P^I = \hat{S}(A)-\hat{S}(\hat{A}_2) \tag{3.3.8}$$

and finally

$$P^0 = I-P^+-P^--P^I = I-\hat{S}(A)-\hat{S}^2(A)+\hat{S}(\hat{A}_2) \tag{3.3.9}$$

The latter eigenprojector, P^0, can be utilized for inverting singular matrices as will be shown later.

Example 1 - The following matrix is used to find its spectral decomposition which has eigenvalues $\lambda_1 = 0$ $\lambda_2 = -2.0$ $\lambda_3 = 3.0$ $\lambda_4 = +j$ $\lambda_5 = -j$

$$A = \begin{bmatrix} 29.2 & -24.2 & 69.5 & 49.8 & 7.0 \\ -9.2 & 5.2 & -18.0 & -16.8 & -2.0 \\ -10.0 & 6.0 & -20.0 & -18.0 & -2.0 \\ -9.6 & 9.6 & -25.5 & -15.4 & -2.0 \\ 9.8 & -4.8 & 18.0 & 18.2 & 2.0 \end{bmatrix}$$

the result of the generalized sign algorithm is

$$\hat{S}(A) = \begin{bmatrix} 11.94930 & -2.24532 & 15.31728 & 21.65328 & -2.24532 \\ -3.84265 & 0.49866 & -4.59065 & -7.18665 & 0.49866 \\ -4.07999 & 0.55999 & -4.91998 & -7.59998 & 0.55999 \\ -4.03465 & 1.04266 & -5.59865 & -7.02664 & 1.04266 \\ 4.15732 & -0.50132 & 4.90931 & 7.81331 & -0.50133 \end{bmatrix}$$

the set of eigenprojectors for the sign spectral decomposition are

$$P^{+} = \begin{bmatrix} 2.08533 & 1.04266 & 0.52132 & 5.21332 & 1.04266 \\ -0.93866 & -0.46933 & -0.23466 & -2.34666 & -0.46933 \\ -0.95999 & -0.48000 & -0.23999 & -2.39999 & -0.48000 \\ -0.36266 & -0.18133 & -0.09066 & -0.90666 & -0.18133 \\ 1.06133 & 0.53066 & 0.26533 & 2.65333 & 0.53066 \end{bmatrix}$$

$$P^{-} = \begin{bmatrix} -9.86397 & 3.28798 & -14.795956 & -16.43995 & 3.28799 \\ 2.90399 & -0.96799 & 4.35598 & 4.83998 & -0.96799 \\ 3.11999 & -1.03999 & 4.67998 & 5.19998 & -1.03999 \\ 3.67199 & -1.22399 & 5.50798 & 6.11998 & -1.22399 \\ -3.09599 & 1.03199 & -4.64398 & -5.15998 & 1.03194 \end{bmatrix}$$

$$P^{I} = \begin{bmatrix} 8.51195 & -6.46416 & 14.40820 & 10.95989 & -5.66391 \\ -2.03199 & 1.90406 & -4.08809 & -2.55997 & 1.10396 \\ -2.15998 & 1.52006 & -3.44009 & -2.79996 & 1.51996 \\ -3.17598 & 2.47203 & -5.48403 & -4.07996 & 2.07198 \\ 1.96798 & -2.09607 & 4.41211 & 2.43996 & -0.89595 \end{bmatrix}$$

$$P^0 = \begin{bmatrix} 0.26668 & 2.13350 & -0.13358 & 0.26674 & 1.33326 \\ 0.06666 & 0.53326 & -0.03323 & 0.06664 & 0.33336 \\ -0.00000 & -0.00006 & 0.00010 & -0.00002 & 0.00003 \\ -0.13334 & -1.06670 & 0.06671 & -0.13335 & -0.66665 \\ 0.06667 & 0.53341 & -0.03345 & 0.06669 & 0.33329 \end{bmatrix}$$

it can be shown this set of eigenprojectors satisfy the properties of the eigenprojectors given earlier.

3.4 Computing the pth Root of a Positive Semidefinite Matrix

Several recent papers have appeared on the computation of roots of matrices that are generalizations of Newton's method. Hoskins and Walton, [5], described a method of computing the pth root of a positive definite matrix with p an integer. Denman [17]described an extension to the Hoskins and Walton algorithm with the only restriction that the matrix not have eigenvalues along the $j\omega$ axis. An algorithm will be given in this paper that removes the restriction on the presence of zero eigenvalues. The procedure can be extended for complete removal of the restriction on eigenvalues along the $j\omega$-axis.

Let A be a general positive semidefinite n×n matrix with the possibility of a zero eigenvalue. Define $\hat{A}$ as the p-th root of A such that

$$A = \hat{A}^P \tag{3.4.1}$$

The Newton algorithm as given by Hoskins and Walton is

$$X_{i+1} = \frac{1}{p}\left[(p-1)X_i + AX_i^{1-P}\right] \qquad X_0 = I \tag{3.4.2}$$

which will converge to $\hat{A}$ in the limit as $\varepsilon \to 0$ with stopping criteria

$$||X_{i+1} - X_i|| < \varepsilon \tag{3.4.3}$$

where ε is a preselected error bound. It can be seen that the above iterative algorithm fails when there is a zero eigenvalue.

The generalized sign matrix provides the mathematical tool to eliminate the problem due to the zero eigenvalue. If A is positive definite, then the

sign of A will be I. The generalized sign matrix will give

$$\hat{S}(A) = P^{+} \tag{3.4.4}$$

from which it follows that the null eigenprojector is given by

$$P^{0} = I - P^{+} = I - \hat{S}(A) \tag{3.4.5}$$

The null eigenprojector will have the same eigenvectors as A but will have eigenvalues +1 for all zero eigenvalues of A and zero eigenvalues for all $R_e(\lambda_i)>0$ of A. Since the eigenvectors of P^0 are the same as those of A, the sum of A and P^0 will have the same eigenvectors as A but $A+P^0$ will be a positive definite matrix with +1 eigenvalues in place of the zero eigenvalues of A. If A_D is defined as

$$A_D = A + P^0 \tag{3.4.6}$$

then the p-th root of A_D will be correct aside from the inclusion of the p-th root of 1 rather than the p-th root of zero. The root due to P^0 is removed from the p-th root of A_D by multiplying $A_D^{1/p}$ by P^+ or

$$\hat{\Lambda} = (\Lambda_D)^{1/p} P^{+} = (\Lambda + P^0)^{1/p} P^{+} \tag{3.4.7}$$

An example is given where A is 4×4 and its eigenvalues are $\lambda_1 = 0$, $\lambda_2 = 0$, $\lambda_3 = 3.258$ and $\lambda_4 = 10.741$

$$A = \begin{bmatrix} 2.0 & -1.0 & 1.0 & -1.0 \\ -1.0 & 4.0 & 3.0 & -3.0 \\ 1.0 & 3.0 & 4.0 & -4.0 \\ -1.0 & -3.0 & -4.0 & 4.0 \end{bmatrix}$$

Using $\rho = 0.5$, the generalized sign matrix is obtained with 7 iterations for the right and left shifts when $\varepsilon = 10^{-6}$ with

$$\hat{s}(A) = P^{+} = \begin{bmatrix} 0.6 & -0.4 & 0.2 & -0.2 \\ -0.4 & 0.6 & 0.2 & -0.2 \\ 0.2 & 0.2 & 0.4 & -0.4 \\ -0.2 & -0.2 & -0.4 & 0.4 \end{bmatrix}$$

The square root, p=2, was obtained from the Newton algorithm in 5 iterations with

$$A^{1/2} = \hat{A} = \begin{bmatrix} 1.091906 & -0.662353 & 0.429553 & -0.429553 \\ -0.662353 & 1.485410 & 0.823057 & -0.823057 \\ 0.429553 & 0.823057 & 1.252610 & -1.252610 \\ -0.429553 & -0.823057 & -1.252610 & 1.252610 \end{bmatrix}$$

The 5-th root was obtained for the same matrix and converged in 7 iterations

$$A^{1/5} = \hat{A} = \begin{bmatrix} 0.761942 & -0.492761 & 0.269181 & -0.269181 \\ -0.492761 & +0.853145 & 0.360339 & -0.360384 \\ 0.269181 & 0.360384 & 0.629565 & -0.629565 \\ -0.269181 & -0.360384 & -0.629565 & 0.629565 \end{bmatrix}$$

The error in reconstructing A from $A^{1/2}$ and $A^{1/5}$ was no greater than 1E-05

for all elements, that is

$$|A_{ij}-(A_{ij}^{1/p})^p|<1.E-05$$

3.5 Computation of the Generalized Inverse Matrix of A

The generalized inverse matrix can also be obtained from the eigenprojectors as given earlier. The null eigenprojector will be used in the same manner as it was used for finding the p-th root.

Let A be an n×n matrix with zero and nonzero eigenvalues and can be represented as

$$A = M\{\text{diag}[\lambda_1,0,0,\ldots,\lambda_r,\lambda_{r+1},\ldots,\lambda_{n-1},\lambda_n]\}M^{-1} \tag{3.5.1}$$

The generalized inverse matrix is obtained when the nonzero eigenvalues are replaced by the reciprocal with the zero eigenvalues unchanged. If $A^{\dagger}$ is the generalized inverse matrix with

$$A^{\dagger} = M\{\text{diag}[\lambda_1^{-1},0,0,\ldots,0,\lambda_r^{-1},\lambda_{r+1}^{-1},\ldots,\lambda_{n-1}^{-1},\lambda_n^{-1}]\}M^{-1} \tag{3.5.2}$$

it follows that $A^{\dagger}$ satisfies the four properties of Penrose [26] which are

$$AA^{\dagger}A = A \qquad \text{3.5P.1}$$

$$A^{\dagger}AA^{\dagger} = A^{\dagger} \qquad \text{3.5P.2}$$

$$(A^{\dagger}A)^{T} = A^{\dagger}A \qquad \text{3.5P.3}$$

$$(AA^{\dagger})^{T} = AA^{\dagger} \qquad \text{3.5P.4}$$

Assume that the generalized sign matrix of A has been found. It then follows that

$$\hat{A} = A+P^{0} \tag{3.5.3}$$

where P^0 has been computed as in (3.3.9) for the most general case. This matrix will have rank n since

$$(3.5.4) \qquad n = \text{rank}(A) + \text{rank}(P^0)$$

and is invertible. The inverse of $\hat{A}$ is given by

$$(3.5.5) \qquad \hat{A}^{-1} = (A+P^0)^{-1} = A^{\dagger}+P^0$$

Since the eigenvectors of P^0 are the same as those of A. The generalized inverse of A is then given by

$$(3.5.6) \qquad A^{\dagger} = \hat{A}^{-1}-P_0$$

or $A^{\dagger} = \hat{A}^{-1}(I-P_0)$.

The matrix given in Section 4 was used to compute $A^{\dagger}$ by this method with

$$A^{\dagger} = \begin{bmatrix} 2.41117 & 19.45559 & -30.77219 & 8.67789 & -11.74442 \\ -1.18890 & -4.34446 & 6.32779 & -3.12226 & 2.85556 \\ -1.00002 & -5.00001 & 7.49999 & -3.00004 & 2.99999 \\ -0.78890 & -7.14446 & 11.42778 & -2.92226 & 4.95554 \\ 1.47780 & 3.98890 & -5.50555 & 3.54449 & -2.81111 \end{bmatrix}$$

It can be shown that the eigenvalues of this matrix are given by $\lambda_1 = 0$, $\lambda_2 = -0.5$, $\lambda_3 = 0.33333$, $\lambda_4 = -j$ and $\lambda_5 = +j$.

3.6 Decoupling of Linear Time-Invariant System Equations

The last topic to be discussed in this paper is the use of the spectral decomposition concept for solving system equations of the form

$$\dot{x}(t) = Ax(t) \qquad x(0) = x \in R^n \tag{3.6.1}$$

where A is n×n and x(t) is n×1. The solution x(t) can be expressed in the form

$$x(t) = \phi(t,0)\, x(0) = \exp(At)x(0) \tag{3.6.2}$$

where $\phi(t,0)$ is the state transition matrix with $\phi(t,t) = I$.

The general matrix A can be decomposed into

$$A = A^{+}+A^{-}+A^{I}+A^{0} \tag{3.6.3}$$

where A^{+}, A^{-}, A^{I} and A^{0} are as defined in Section 3.2. It follows that (3.6.2) can be modified with

$$\begin{aligned} x(t) &= \exp[(A^{+}+A^{-}+A^{I}+A^{0})t]x(0) \\ &= \exp(A^{+}t)\exp(A^{-}t)\exp(A^{I}t)x(0) \end{aligned} \tag{3.6.4}$$

since $A^{0} = 0$. The exponential matrices are commutative as each matrix has the same eigenvectors.

The initial condition vector can also be decomposed with

$$P^{-} = \begin{bmatrix} 0.07692 & -0.07692 & 0.01923 & -0.01923 \\ -0.23076 & 0.23076 & -0.05769 & 0.05769 \\ 0.69230 & -0.69230 & 0.17307 & -0.17307 \\ -2.07692 & 2.07692 & -0.51923 & 0.51923 \end{bmatrix}$$

$$P^{I} = \begin{bmatrix} 0.32307 & -0.12307 & -0.16923 & -0.03076 \\ -0.36923 & 0.56923 & -0.09230 & -0.10769 \\ -1.29230 & 0.49230 & 0.67692 & 0.12307 \\ -1.29230 & -2.27692 & 0.36923 & 0.43076 \end{bmatrix}$$

$$P^{0} = 0$$

the resultant eigenprojectors on the initial condition vector are

$$X^{+}(0) = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix} \quad X^{-}(0) = \begin{bmatrix} -0.307692 \\ 0.923077 \\ -2.769231 \\ 8.307693 \end{bmatrix} \quad X^{I}(0) = \begin{bmatrix} -0.892308 \\ 2.876923 \\ 3.569231 \\ -11.507692 \end{bmatrix} \quad X^{0}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The initial condition vectors given above were used to compute the vectors $x^{+}(t)$, $x^{-}(t)$ and $x^{I}(t)$. The different modes were separated by the algorithm according to the spectral domains. Table 3.1 gives the third element of the vectors as well as the third element of $x(t)$ as computed directly from (3.6.1).

t(sec)	$X_3(t)$	$X_3^+(t)$	$X_3^-(t)$	$X_3^I(t)$
0	1.000000	0.200000	-2.769231	3.569231
0.4	-2.176561	0.298365	-0.834076	-1.640849
0.8	-5.661723	0.445108	-0.251218	-5.855612
1.2	-5.930082	0.664023	-0.075665	-6.518440
1.6	-2.259451	0.990606	-0.022789	-3.227268
2.0	3.492468	1.477811	-0.006863	2.021520
2.4	8.246650	2.204635	-0.002066	6.044082
2.8	9.688690	3.288929	-0.000620	6.400383
3.2	7.780616	4.906506	-0.000183	2.874297
3.6	4.924292	7.319647	-0.000049	-2.395300
4.0	4.707674	10.919630	-0.000006	-6.211941

All computations were in single precision. The value of $x_3(t)$ should be equal to the sum of $x_3^+(t)$, $x_3^-(t)$ and $x_3^I(t)$. The difference between x(t) and the sum of the individual terms was 9×10^{-6} at t = 4 seconds.

4. Laplace Transform of Matrix Functions

Assume that a system is defined by a set of differential equations of the form

$$\ddot{x}(t)+A_1\dot{x}(t)+A_2x(t) = Gu(t) \tag{4.0.1}$$

where A_1, A_2 and G are constant matrices. The matrices A_1 and A_2 are n×n and G is n×m. Since this type of equation is encountered in structures, the closed form solution in the time domain is of interest. The Laplace and inverse-Laplace transforms are therefore of interest.

4.1 Laplace Transform

Consider a system defined as in (4.0.1) with initial conditions x(0) and $\dot{x}(0)$ from which it follows that

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & I \\ -A_2 & -A_1 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u(t) = A\bar{x}(t)+\bar{G}u(t) \tag{4.1.1}$$

with

$$\begin{bmatrix} sX(s)-x(0) \\ s^2X(s)-sx(0)-\dot{x}(0) \end{bmatrix} = \begin{bmatrix} 0 & I \\ -A_2 & -A_1 \end{bmatrix} \begin{bmatrix} X(s) \\ sX(s)-x(0) \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u(s)$$

The second equation of (4.1.2) gives

$$s^2X(s)-sx(0)-\dot{x}(0) = -A_2X(s)-A_1sX(s)+A_1x(0)+GU(s) \tag{4.1.3}$$

which can be rearranged with

$$(4.1.4) \qquad [s^2I+A_1s+A_2]X(s) = [sI+A_1]x(0)+\dot{x}(0)+GU(s)$$

Premultiplying by $[s^2I+A_1s+A_2]^{-1}$ gives

$$(4.1.5) \qquad X(s) = [s^2I+A_1s+A_2]^{-1}\{[sI+A_1]x(0)+\dot{x}(0)+Gu(s)\}$$

Suppose that the inverse of [sI-a] is now considered where

$$(4.1.6) \qquad \begin{bmatrix} sI & -I \\ A_2 & sI+A_1 \end{bmatrix}^{-1} = \begin{bmatrix} s^2I+A_1s+A_2 & 0 \\ 0 & s^2I+A_1s+A_2 \end{bmatrix}^{-1} \begin{bmatrix} sI+A_1 & I \\ -A_2 & sI \end{bmatrix}$$

which will be written in the compact form

$$(4.1.7) \qquad [sI-A]^{-1} = [s^2I+A_1s+A_2]^{-1} \begin{bmatrix} sI+A_1 & I \\ -A_2 & sI \end{bmatrix}$$

for convenience. It should be noted that the upper row of $[sI-a]^{-1}$ contains the two terms of interest in (4.1.5).

If (4.0.1) has a leading term A_0, then (4.1.5) becomes

$$(4.1.8) \qquad X(s) = [A_0s^2+A_1s+A_2]^{-1}\{[SI+A_1]A_0x(0)+A_0\dot{x}(0)+Gu(s)\}$$

which requires that [sI-a] be modified. A general form for A_0 in [sI-a] is not known thus (4.1.7) will be used with A_0 included as shown in (4.1.8).

4.2 Inverse Laplace Transform

The inverse Laplace of the desired matrix equation can be obtained from (4.1.7). Consider the eigenvalue-eigenvector problem for A where

$$\lambda I - A = \begin{bmatrix} \lambda I & -I \\ A_2 & \lambda I + A_1 \end{bmatrix} \tag{4.2.1}$$

where λ and s are interchangeable in the mathematical operations. It is known that the eigenvector matrix of (4.0.1) has the general form

$$\Phi = \begin{bmatrix} \phi_1 & \phi_2 \\ \phi_1\Lambda_1 & \phi_2\Lambda_2 \end{bmatrix} \tag{4.2.2}$$

with $\phi_2 = \phi_1^*$ and $\Lambda_2 = \Lambda_1^*$ for a system with all eigenvalues in pairs, real or complex conjugate. Since the structure problem is of that type, then Φ can be assumed to be

$$\Phi = \begin{bmatrix} \phi_1 & \phi_1^* \\ \phi_1\Lambda_1 & \phi_1^*\Lambda_1^* \end{bmatrix} \tag{4.2.3}$$

It then follows that

$$A = \begin{bmatrix} \phi_1 & \phi_1^* \\ \phi_1\Lambda_1 & \phi_1^*\Lambda_1^* \end{bmatrix} \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_1^* \end{bmatrix} \begin{bmatrix} \phi_1 & \phi_1^* \\ \phi_1\Lambda_1 & \phi_1^*\Lambda_1^* \end{bmatrix}^{-1} \tag{4.2.4}$$

Using the property that a function of a matrix is given by

(4.2.5) $$f(A) = \Phi f(\Lambda)\Phi^{-1}$$

then

(4.2.6) $$\exp(At) = \mathcal{L}^{-1}[sI-a] = \Phi f(s)\Phi^{-1}$$

from which it follows that

(4.2.7) $$\exp(At) = \Phi \begin{bmatrix} \exp(\Lambda_1 t) & 0 \\ 0 & \exp(\Lambda_1^* t) \end{bmatrix} \Phi^{-1}$$

It is not difficult to show from (4.1.8), (4.2.4) and (4.2.7) that

(4.2.8) $$\mathcal{L}^{-1}[s^2 I+A_1 s+A_2]^{-1} = \exp(B_1 t)\sin(B_2 t)B_2^{-1}$$

(4.2.9) $$\mathcal{L}^{-1}[s^2 I+A_1 s+A_2]^{-1} = \exp(B_1 t)[\cos(B_2 t)-\sin(B_2 t)(B_2^{-1}B_1)]$$

where B_1+jB_2 is a spectral factor of $[s^2 I+A_1 s+A_2]$ or

(4.2.10) $$[sI+B_1+jB_2][SI+B_1-jB_2] = [s^2 I+A_1 s+A_2]$$

with B_1+jB_2 in the upper half plane and B_1-jB_2 in the lower half plane. Substituting back into (4.1.5) with $A_0 = I$, x(t) is given by

(4.2.11) $$\begin{aligned} x(t) = {} & \{\exp(B_1 t)[\cos(B_2 t)-\sin(B_2 t)(B_2^{-1}B_1)]\}x(0) \\ & + \{\exp(B_1 t)\sin(B_2 t)(B_2^{-1})\}\dot{x}(0) \\ & + \int_0^t \exp[B_1(t-\tau)]\sin[B_2(t-\tau)](B_2^{-1})Cu(\tau)d\tau \end{aligned}$$

"Page missing from available version"

4.3 Facorization of A(s)

Let A(s) be defined as

$$A(s) = Is^2+A_1s+A_2$$

for which the factors $[sI+B_1+jB_2]$ and $[sI+B_1-jB_2]$ are to be obtained. It can be shown that there exists a matrix R such that

$$R^2+A_1R+A_2 = 0 \tag{4.3.1}$$

where R is given by either

$$R_1 = \phi\Lambda\phi^{-1} = -B_1-jB_2 \tag{4.3.2}$$

or

$$R_2 = \phi^*\Lambda^*\phi^{*-1} = -B_1+jB_2 \tag{4.3.3}$$

The matrix R will be complex and can be computed from the eigenvalues and eigenvectors of A where

$$A = \begin{bmatrix} 0 & I \\ -A_2 & -A_1 \end{bmatrix} \tag{4.3.4}$$

It can also be shown that A(s) can be written as

$$A(s) = (Is+\frac{A_1}{2})^2 + A_2 - \frac{A_1^2}{4} \tag{4.3.5}$$

which factors as

$$A(s) = \{Is+\frac{A_1}{2}+j[A_2-A_1^2/4]^{1/2}\}\{Is+\frac{A_1}{2}-j[A_2-A_1^2/4]^{1/2}\} \tag{4.3.6}$$

therefore

$$B_1 = -\frac{A_1}{2} \tag{4.3.7}$$

$$B_2 = [A_2-A_1^2/4]^{1/2} \tag{4.3.8}$$

provided that $A_2-A_1^2/4$ is a positive definite matrix. Provided that this restriction is satisfied, it is not necessary to compute the algebraic solution to (4.3.1). The square root of $A_2-A_1^2/4$ can be computed by the procedure given by Denman [17] or Hoskins and Walton [20].

The equation given in (4.3.1) is a algebraic matrix Riccati equation and plays an important role in system analysis and particularly in spectral factorization. To show that R is as given (4.3.2) consider (4.3.1) and (4.3.4). It is known that the eigenvector of a is as given in (4.2.2) thus

$$\begin{bmatrix} \lambda I & -I \\ A_2 & \lambda I+A_1 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_1\Lambda_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.3.9}$$

thus if $\lambda = \Lambda_1$ then

$$A_2\phi_1+\Lambda_1\phi_1\Lambda_1+A_1\phi_1\Lambda_1 = 0 \tag{4.3.10}$$

Now if $\Lambda_1\phi_1 = \phi_1\Lambda$ then

$$\phi_1\Lambda_1^2\phi_1^{-1}+A_1\phi_1\Lambda_1\phi_1^{-1}+A_2 = 0 \tag{4.3.11}$$

therefore $R = \phi_1\Lambda_1\phi_1^{-1}$ as given in (3.2). Now from (2.4)

$$\begin{bmatrix} 0 & I \\ -A_2 & -A_1 \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_1^* \\ \phi_1\Lambda_1 & \phi_1^*\Lambda_1^* \end{bmatrix}\begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_1^* \end{bmatrix}\begin{bmatrix} \phi_1 & \phi_1^* \\ \phi_1\Lambda_1 & \phi_1^*\Lambda_1^* \end{bmatrix}^{-1} \tag{4.3.12}$$

thus

$$A_1 = -(\phi_1\Lambda_1\phi_1^{-1}-\phi_1^*\Lambda_1^{*2}\phi_1^{*-1})(\phi_1\Lambda_1\phi_1^{-1}-\phi_1^*\Lambda_1^*\phi_1^{*-1})^{-1} \tag{4.3.13}$$

$$A_2 = -(\phi_1\Lambda_1\phi_1^{-1}-\phi_1^*\Lambda_1^*\phi_1^{*-1})(\phi_1\Lambda_1^{-1}\phi_1^{-1}-\phi_1\Lambda_1^{*-1}\phi_1^{*-1})^{-1} \tag{4.3.14}$$

where it is necessary to prove that

$$R_1 = -\frac{A_1}{2}+j(A_2-A_1^2/4)^{1/2} = \phi_1\Lambda_1\phi_1^{-1} \tag{4.3.15}$$

If $R_1 = R_2^* = -B_1+jB_2$ then

$$A_1 = -[(-B_1+jB_2)^2-(-B_1-jB_2)^2][-B_1+jB_2-B_1+jB_2]^{-1} \tag{4.3.16}$$

$$= -(B_1B_2+B_2B_1(B_2)^{-1} = -2B, \qquad \underline{\text{iff } B_1B_2=B_2B_1}$$

$$A_2 = -[(-B_1+jB_2)-(-B_1-jB_2)][(-B_1+jB_2)^{-1}-(-B_1-jB_2)^{-1}]^{-1} \tag{4.3.17}$$

$$= -2jB_2(-B_1-jB_2)(-2jB_2)^{-1}(-B_1+jB_2) = B_1^2+B_2^2 \quad \underline{\text{iff } B_1B_2=B_2B_1}$$

Substituting back into R_1 for A_1 and A_2.

$$(4.3.18) \qquad R_1 = -\frac{2B_1}{2} + j[B_1^2+B_2^2-B_1^2] = -B_1+jB_2$$

where

$$(4.3.19) \qquad B_1 = -\frac{A_1}{2} \qquad\qquad B_2 = [A_2-A_1^2/4].$$

5. Diagonalization of Matrix Polynomials

There has been numerous comments made at technical meetings regarding the diagonalization of matrix polynomial of the form

$$A(s) = Ms^2+Cs+K = M^{1/2}[Is^2+\bar{C}s+\bar{K}]M^{1/2}. \tag{5.1}$$

It has been stated that the second order matrix polynomials of the general form cannot be diagonalized into the form

$$D(s) = M^{1/2}[Is^2+2\bar{\zeta}\omega s+\omega^2]M^{1/2} = M^{1/2}\bar{D}(s)M^{1/2} \tag{5.2}$$

where $2\zeta\omega$ and ω are diagonal forms. It is true that there does not exist a constant matrix such that

$$D(s)\ Q\bar{A}(s)Q^{-1} \tag{5.3}$$

but it is not true that A(s) cannot be diagonalized. As will be shown in the following sections, polynomials do exist such that

$$D(s)\ Q(s)\bar{A}(s)P(s) \tag{5.4}$$

To show this, it will be assumed that the mass matrix is invertible so that $M^{1/2}$ exists. This assumption may be too restrictive but will be used.

5.1 Diagonalization of Block Companion Matrix

Since it has been assumed that M is invertible, a block companion form matrix a can be defined with

$$(5.1.1) \qquad A = \begin{bmatrix} 0 & I \\ -\bar{K} & -\bar{C} \end{bmatrix}$$

where $\bar{K} = M^{1/2}KM^{1/2}$, and $\bar{C} = M^{1/2}CM^{1/2}$. If K and C are symmetric then $\bar{K}$ and $\bar{C}$ will be symmetric when $M^{1/2}$ is symmetric.

The matrix A can be defined as

$$(5.1.2) \qquad A = \Phi\Lambda\Phi^{-1}$$

where Φ is the n×n matrix of eigenvectors of a and Λ is the eigenvalue matrix. Let A have distinct eigenvalues with real and complex eigenvalues in conjugate pairs. The eigenvalues and eigenvectors of A can always be ordered such that

$$(5.1.3) \qquad A = [y_1 y_1^* y_2 y_2^* \cdots y_m y_m^* y_{m+1} y_{m+2} \cdots y_{2n}]$$
$$\cdot \operatorname{diag}[\lambda_1 \lambda_1^* \lambda_2 \lambda_2^* \cdots \lambda_m \lambda_m^* \lambda_{m+1} \lambda_{m+2} \cdots \lambda_{2n}]$$
$$\cdot [y_1 y_1^* y_2 y_2^* \cdots y_{2n}]^{-1}$$

and thus

$$(5.1.4) \qquad \Lambda = \Phi^{-1}A\Phi = \operatorname{diag}[\lambda_1 \lambda_1^* \lambda_2 \lambda_2^* \cdots \lambda_{2n}]$$

Consider any pair of eigenvalues including the real eigenvalues. The comparison form of the 2×2 subblocks can then be written as

$$(5.1.5) \qquad A_{B_i} = \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\zeta_i\omega_i \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \lambda_i & \lambda_i^* \end{bmatrix} \begin{bmatrix} \lambda_i & \\ 0 & \lambda_i^* \end{bmatrix} \begin{bmatrix} 1 & 1 \\ \lambda_i & \lambda_i^* \end{bmatrix}^{-1}$$

with a new matrix defined as A_{BC} given by

$$(5.1.6)\quad A_{EC} = \begin{bmatrix} A_{B1} & & & 0 \\ & A_{B2} & & \\ & & \ddots & \\ 0 & & & A_{Bn} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ -\omega_1^2 & -2\zeta_1\omega_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & -\omega_2^2 & -2\zeta_2\omega_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \\ 0 & 0 & 0 & 0 & \cdots & -\omega_n^2 & -2\zeta_n\omega_n \end{bmatrix}$$

It follows that a row-column interchange can now be made to transform A_{BC} into the form

$$(5.1.7)\quad A_{DC} = \begin{bmatrix} 0 & I \\ -\omega^2 & -2\zeta\omega \end{bmatrix} = E A_{BC} E^T$$

where E is an elementary matrix. For example, consider the 4×4 matrix A_{BC} with

$$(5.1.8)\quad A_{Bi} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_1^2 & -2\zeta_1\omega_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_2^2 & -2\zeta_2\omega_2 \end{bmatrix}$$

then

$$(5.1.9)\quad E A_{BC} E^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_1^2 & -2\zeta_1\omega_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_2^2 & -2\zeta_2\omega_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation process requires three transformations with

$$A_{DC} = E\Theta\Phi^{-1}A\Phi\Theta^{-1}E^{T} \tag{5.1.10}$$

where E is the elementary matrix, Θ is a block diagonal matrix with

$$\Theta_i = \begin{bmatrix} 1 & 1 \\ \lambda_i & \lambda_i^* \end{bmatrix} -$$

or a suitable pair of real eigenvalues and Φ is the usual eigenvector matrix. The Θ_i matrix must be invertible which is a valid assumption for conjugate pairs of eigenvalues and distinct real eigenvalues.

5.2 Matrix Polynomial Form

The analysis given in Section 5.1 was concerned with the "state variable" or block companion form and does not give the required transformation in matrix polynomial form. The objective of the diagonalization form is to obtain Q(s) and P(s) as given in (5.4). The desired Q(s) and P(s) can be found from the work in Section 5.1. It is not difficult to show that

$$(5.2.1) \qquad [A(s)]^{-1} = \begin{bmatrix} sI & -I \\ K & sI+C \end{bmatrix}^{-1} = \begin{bmatrix} [\bar{A}(s)]^{-1} & \\ 0 & [\bar{A}(s)]^{-1} \end{bmatrix} \begin{bmatrix} sI+C & I \\ -K & sI \end{bmatrix}$$

which was shown in a previous report. The diagonalization process given in the previous section holds for a(s) with

$$(5.2.2) \qquad E\Theta\Phi^{-1}[A(s)]\Phi\Theta^{-1}E^{T} = \begin{bmatrix} sI & -I \\ \omega^2 & sI+2\zeta\omega \end{bmatrix}$$

with

$$(5.2.3) \qquad \begin{bmatrix} sI & -I \\ \omega^2 & sI+2\zeta\omega \end{bmatrix}^{-1} = \begin{bmatrix} [\bar{D}(s)]^{-1} & \\ 0 & [\bar{D}(s)]^{-1} \end{bmatrix} \begin{bmatrix} sI+2\zeta\omega & I \\ -\omega^2 & sI \end{bmatrix}$$

$$= E\Theta\Phi^{-1}[A(s)]^{-1}\Phi\Theta^{-1}E^{T}$$

Suppose that the product $E\Theta\Phi^{-1}$ is defined as Q^{-1} thus

$$(5.2.4) \qquad \begin{bmatrix} [\bar{D}(s)]^{-1} & \\ 0 & [\bar{D}(s)]^{-1} \end{bmatrix} \begin{bmatrix} sI+2\zeta\omega & I \\ -\omega^2 & sI \end{bmatrix} = Q^{-1} \begin{bmatrix} [\bar{A}(s)]^{-1} & 0 \\ 0 & [\bar{A}(s)]^{-1} \end{bmatrix} \begin{bmatrix} sI+C & I \\ -K & sI \end{bmatrix} Q$$

$$D(s) = [(sI+C)Q_{12}+Q_{22}]^{-1}[\bar{A}(s)][sQ_{12}+Q_{11}] \tag{5.2.5}$$

as well as

$$D(s) = [sQ_{22}-KQ_{12}]^{-1}\bar{A}(s)[sQ_{22}+Q_{21}] \tag{5.2.6}$$

for the quadratic matrix polynomial with

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \tag{5.2.7}$$

where Q_{ij} is an n×n partitioned block of Q. It follows from further analysis that

$$\omega^2 = -Q_{12}^{-1}Q_{21} = (CQ_{12}+Q_{22})^{-1}KQ_{11} \tag{5.2.8}$$

and

$$2\zeta\omega = Q_{22}^{-1}[Q_{21}+CQ_{22}-KQ_{12}] = Q_{12}^{-1}[Q_{22}-Q_{11}] \tag{5.2.9}$$

The usual transformation to normal coordinates

$$x(t) = \Phi q(t) \tag{5.2.10}$$

is replaced by a coordinate transformation of the form

(5.2.11) $$Q_{22}\dot{q}(t)+Q_{21}q(t) = x(t)$$

thus

(5.2.12) $$[Is^2+Cs+K][Q_{22}s+Q_{21}]q(s) = U(s)$$

If the force vector u(t) is transformed with

(5.2.13) $$Q_{22}\dot{f}(t)-KQ_{12}f(t) = u(t)$$

then the resulting equation is

(5.2.14) $$[Is^2+\bar{C}s+\bar{K}][Q_{22}s+Q_{21}]q(s) = [Q_{22}s-\bar{K}Q_{12}]f(s)$$

therefore

(5.2.15) $$[Q_{22}s-\bar{\bar{K}}Q_{12}]^{-1}[Is^2+\bar{C}s+\bar{K}][Q_{22}s+Q_{21}]q(s) = f(s)$$

It follows from (5.2.6) that

(5.2.16) $$D(s)q(s) = f(s) = [Is^2+2\zeta\omega s+\omega^2]q(s)$$

A different form of (5.2.15) can be obtained from (5.2.5) although (5.2.16) is the same.

The mass matrix must now be taken back into the equation since $M^{1/2}$ was factored out.

5.3 Examples

As an example of the above procedure, let

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -35 & -40 & 12 & 8 \\ -40 & -35 & 8 & 12 \end{bmatrix}$$

with $\Lambda = [15 \quad 5 \quad 5 \quad -1]$ and

$$\Phi = \begin{bmatrix} -1 & -5 & 103 & -1 \\ -1 & -4 & -23 & 1 \\ -15 & -25 & 515 & 1 \\ -15 & -20 & -115 & -1 \end{bmatrix}$$

as computed with by EISPAK with column vector scaling. The transformed A_{DC} matrix was computed as

$$A_{DC} = \begin{bmatrix} -1.9E-6 & -2.86E-6 & 1 & -4.76E-6 \\ -2.30E-6 & 2.62E-6 & 2.98E-8 & 1 \\ -75.0 & -1.22E-4 & 20. & -7.63E-5 \\ 2.38E-7 & 5 & -7.45E-9 & 4 \end{bmatrix}$$

with

$$-\omega^2 = Q_{12}^{-1}Q_{21} = \begin{bmatrix} -75.0 & -3.81E-6 \\ -1.19E-7 & 5 \end{bmatrix}$$

and

$$2\zeta\omega = Q_{12}^{-1}(Q_{22}-Q_{11}) = \begin{bmatrix} -2.0 & 3.81E\text{-}6 \\ -2.98E\text{-}8 & -4 \end{bmatrix}$$

All computations were in single precision except for the eigenvalue. The Q matrix required for the block diagonalization of A was computed and is given by

$$Q = \begin{bmatrix} -7. & 16.3333 & 0.4 & 17.333 \\ -5.5 & -3 & 0.3 & -4.0 \\ -30 & 86.6667 & 1 & 85.6667 \\ -22.5 & -20 & 0.5 & -19 \end{bmatrix}$$

with

$$Q = \Phi\Theta^{-1}E \ .$$

The example indicates that the block diagonalization of A is valid. The matrices Φ,Θ, and E will always exist thus Q will transform A into a diagonal block companion matrix.

6. Quadrature Method and Laplace Transforms

The quadrature formula for numerical integration is a useful mathematical tool for system analysis. The primary use in the work described in this report is that related to system identification although the procedure has other related uses. Readers that are not familar with the work should consult Bellman, Kalaba and Lockett for details as well as applications, [27]. Numerous papers are also available in the literature on the use of other orthogonal polynomial rather than the Legendre polynomials as used in [28].

Consider a function f(t) that is Laplace transformable, that is

$$\int_0^\infty |f(t)|e^{-\sigma t}dt<\infty \tag{6.1}$$

for a real positive σ. The Laplace transform of f(t) is given by

$$F(s) = \int_0^\infty f(t)e^{-st}dt \tag{6.2}$$

where the inverse Laplace transform is given by

$$f(t) = \frac{1}{2\pi i}\int_{\sigma-i\omega}^{\sigma+i\omega} F(s)e^{st}ds \tag{6.3}$$

The Laplace transform of f(t) will be denoted by F(s) with the notation

$$\mathcal{L}[f(t)] = F(s) \tag{6.4}$$

and the inverse transform by

$$\mathcal{L}^{-1}[F(s)] = f(t) \tag{6.5}$$

The Laplace transform and the inverse Laplace transforms also hold for vector and matrix functions. If f(t) is a n×1 vector then F(s) will be a n×1 vector where

$$F_i(s) = \int_0^\infty f_i(t)e^{-st}dt \tag{6.6}$$

with a similar expression for the inverse transform. Let A(t) represent a matrix, then the elements of the matrix have Laplace transforms $A_{ij}(s)$ defined by

$$A_{ij}(s) = \int_0^\infty A_{ij}(t)e^{-st}dt \tag{6.7}$$

The operations are always performed on each element of the matrix rather than as a matrix operation.

6.1 Properties of Laplace Transforms

Some important properties that will be useful in the work that follows will be given

a) Real differentiation. If $\mathcal{L}[f(t)] = F(s)$ then

$$\mathcal{L}\left[\frac{df}{dt}\right] = SF(s)-f(0-) \tag{6.1.1}$$

and

$$(6.1.2) \qquad \mathcal{L}\left[\frac{d^n f}{dt^n}\right] = s^n F(s) - s^{n-1} f(0-) - s^{n-2} \dot{f}(0-) \ldots - f^{(n-1)}(0-)$$

where $\dot{f}(0-)$ indicates the first derivative with respect to time and $f^{(n-1)}(0-)$ indicates the (n-1)th derivative. The argument 0- represents the initial condition on f(t).

b) Real integration. Let $\mathcal{L}[f(t)] = F(s)$ then

$$(6.1.3) \qquad \mathcal{L}\left[\int_{0^-}^{t} f(\tau)d\tau\right] = \frac{F(s)}{s}$$

c) Differentiation by s. If s is the Laplace variable, then

$$(6.1.4) \qquad \mathcal{L}[tf(t)] = -\frac{dF(s)}{ds}$$

6.2 Laplace Transforms and Quadrature Integration

Let f(t) represent a Laplace transformable variable, scalar or vector, then $[f(t)] = F(s)$ with

$$(6.2.1) \qquad F(s) = \int_0^\infty f(t)e^{-st}dt = \sum_{i=1}^{N} w_i r_i^{s-1} f(t_i) \qquad s = 1,2,3,\ldots,\infty$$

where the summation form is the quadrature integration of f(t). The weights are denoted by w_i and the roots by r_i with $t_i = -\log(r_i)$. Details of the derivation of (6.21) are given in [27]. The weights and roots are given in the appendix to this report for N = 3,4,...,15. Extensive tables can be found elsewhere [29].

The properties given in Section 6.1 can be used to derive other useful expressions. From property a)

$$(6.2.2) \qquad sF(s) = f(0^-) = \sum_{i=1}^{N} w_i r_i^{s-1} \frac{df(t_i)}{dt}$$

with $t(0^-)$ the initial condition on the function f(t). From b)

$$(6.2.3) \qquad \frac{F(s)}{s} = \sum_{i=1}^{N} w_i r_i^{s-1} \left[\int_0^{t_i} f(\tau)d\tau\right]$$

and finally

$$(6.2.4) \qquad \frac{dF(s)}{ds} = - \sum_{i=1}^{N} w_i r_i^{s-1} t_i f(t_i)$$

As examples of each of these, let

$$(6.2.5) \qquad f(t) = 1-e^{-t}$$

which has the Laplace transform of

$$(6.2.6) \qquad F(s) = \frac{1}{s(s+1)}$$

with

$$(6.2.7a) \qquad \frac{df}{dt} = e^{-t}$$

$$(6.2.7b) \qquad \int_0^t (1-e^{-\tau})d\tau = t+e^{-t}-1$$

(6.2.7c) $\qquad tf(t) = t(1-e^{-t})$

The table below gives computed values from (6.2.1), (6.2.2), (6.2.3) and (6.2.4) with exact values for quadrature orders N=7 and N-11. The functions F(s) and sF(s) are correct to six digits whereas dF(s)/ds and F(s)/s have significant errors. The errors in the latter two functions decrease with the quadrature orders. It is therefore necessary to select the quadrature order higher if f(t) is not a smooth function over the range of time used in the procedure and particularily if f(t) has variations between the sampling points. Since the interval time between the sample times t_i increases as i get larger, less information is available for large t_i.

S		F(s)	sF(s)	dF(s)ds	F(s)/s
1	EXACT	0.5	0.5	-0.75	0.5
(N=7)	COMPUTED	0.5	0.5	-0.738658	0.488739
(N=11)	COMPUTED	0.5	0.5	-0.745205	0.495219
2	EXACT	0.166666	0.333333	-0.138888	0.083333
(N=7)	COMPUTED	0.166666	0.333333	-0.138972	0.0834144
3	EXACT	0.083333	0.250000	-0.048611	0.0277777
(N=7)	COMPUTED	0.083333	0.250000	-0.048609	0.0277758
(N=11)	COMPUTED	0.083333	0.250000	-0.048611	0.0277776
4	EXACT	0.050000	0.200000	-0.022500	0.012500
(N=7)	COMPUTED	0.050000	0.200000	-0.022500	0.012500
(N=11)	COMPUTED	0.050000	0.200000	-0.022500	0.012500
5	EXACT	0.033333	0.166666	-0.012222	0.006666
(N=7)	COMPUTED	0.033333	0.166666	-0.012222	0.006666
(N=11)	COMPUTED	0.033333	0.166666	-0.012222	0.006666

Table 6.1 COMPUTED FUNCTIONS

The quadrature formula given in (6.2.1) fails for noninteger values of s. It has been shown in [27] that values of F(s) other than the integer

values can be computed by modifying the formula. Let a be an arbitrary scalar, then

$$F(s/a) = a \sum_{i=1}^{N} w_i r_i^{s-1} f(at_i) \qquad (6.2.8)$$

Equations (6.2.1) and (6.2.8) can be used to compute F(s) such that rapidly changing values of f(t) are included in the data set for F(s). This procedure will assure that variations of f(t) for large t_i has been included in the s-domain information.

6.3 Inverse Laplace Transforms and Quadrature Formulas

The time function f(t) at times t_i can be computed from F(s) by noting that a set of linear equations can be defined from F(s) with

$$F(s) = \sum_{i=1}^{N} w_i r_i^{s-1} f(t_i) \tag{6.3.1}$$

Let S = 1,2,3,...,N, then

$$\begin{bmatrix} F(1) \\ F(2) \\ \vdots \\ F(N) \end{bmatrix} = \begin{bmatrix} w_1 & w_2 & w_3 & \cdots & w_N \\ w_1 r_1 & w_2 r_2 & w_3 r_3 & \cdots & w_N r_N \\ \vdots & \vdots & \vdots & \vdots\vdots\vdots & \vdots \\ w_1 r_1^{N-1} & w_2 r_2^{N-1} & w_3 r_3^{N-1} & \cdots & w_N r_N^{N-1} \end{bmatrix} \begin{bmatrix} f(t_1) \\ f(t_2) \\ \vdots \\ f(t_N) \end{bmatrix} \tag{6.3.2}$$

Equation (6.3.2) can then be considered as having the form Ax = b which can be solved by inverting A with $x = A^{-1}b$. The A matrix is ill-conditioned and will require care in selecting the algorithm to compute f(t).

The computed values from (6.3.2) and the exact values of the f(t) given in (6.2.5) are tabulated in the table below. Single precision calculations were used with a 7th order quadrature formula. The singular value decomposition was used to compute the $f(t_i)$ values.

i	t_i (sec)	$f(t_i)$ Exact	Computed	Error
1	0.025772	0.0254458	0.0254146	3.12E-5
2	0.138382	0.129234	0.129285	-5.1E-5
3	0.352509	0.297077	0.297007	7.E-5
4	0.693147	0.5	0.50009	-9.E-5
5	1.213762	0.702923	0.702812	1.11E-4
6	2.046127	0.870766	0.870889	-1.23E-4
7	3.671195	0.974554	0.97445	1.04E-4

Table 6.2 Compute Value of $f(t_i)$ from F(s)

Additional values of $f(t_i)$ could be obtained by using a higher order quadrature formula or by utilizing (6.2.8). The accuracy of the computed values of $f(t_i)$ will depend upon the order of the quadrature formula as well as the variation of f(t). There is no assurance that a low order quadrature formula will give a unique vector f(t) when the s-domain information is inadequate. For example, a resonance peak may be missed entirely or else the s-domain information may not cover a region in which F(s) varies rapidly as at cutoff. The same is true for the computation of F(s) from f(t).

The above procedure can be utilized to compute the response of the structure to a forcing function. Let X(s) be defined by

$$x(s) = [Ms^2+CS+K]^{-1}Gu(s) \tag{6.3.3}$$

where $[Ms^2+CS+K]^{-1}$ can be expanded as

$$[Ms^2+CS+K]^{-1} = \sum_{i=1}^{2n} \frac{R_i}{s+s_i} \qquad R_i \quad n\times n \tag{6.3.4}$$

where R_i is the matrix residue of the root $s = -s_i$. The residues have rank 1 and are given by the outer product

$$R_i = \hat{y}_i \hat{z}_i^T \tag{6.3.5}$$

where $\hat{y}_i$ is the right latent vector of Ms^2+CS+K and $\hat{z}_i$ is the left latent vector. If Ms^2+CS+K is symmetric then $\hat{z}_i = \hat{y}_i$. The vectors $\hat{y}_i$ and $\hat{z}_i$ must be properly normalized. Assuming that the roots s_i appear in conjugate parrs, the residue for s_i^* must be given R_i^* which means that all of the

residues can be stored in n^2 locations, n vectors of dimension n×1. An additional storage of n locations must be available for the n roots.

The values of X(s/a)/a can therefore be computed by using

$$X(s/a)/a = \sum_{i=1}^{2n} \frac{R_i}{s/a+s_i} Gu(s) \tag{6.3.6}$$

It follows that $x(t_i)$ can be computed by using (6.3.2) with

$$X(-a\ell nr_i) = \sum_{i=1}^{N} P_i X(s/a)/a = X(t_i) \tag{6.3.7}$$

where P_i is the first row of the inverse of the matrix in (6.3.2). Approximately n^2+n storage locations will be required by using the above algorithm which exceeds the storage requirement for a 4th order Runge-Kutta when the sparse matrices M, C, and K are handled properly in the integration algorithm. The sparse nature of M, C and K cannot be used in the quadrature algorithm as described.

7. Identification of Linear Systems by Quadrature Method

The identification or parameter estimation problem in linear system theory is that of determining the system parameters from a sequence of input-output data. In some cases, only statistical properties of the input are known with measured output data available. It will be assumed that the input, u(t), and the output, y(t), are known. Since the identification problem of interest is to estimate the parameters of a vibrating system or a structure, u(t) will be the excitation of the system with y(t) measured displacements of points or nodes of the structure. The variables u(t) and y(t) may be vectors and x(t) will be the vector of all nodes of the structure.

It will be assumed that the excitation u(t) is that signal applied to a set of transducers that apply mechanical forces to the structure whereas y(t) will be the output of sensors that measure the displacement of the nodes. With those assumptions, the vector x(t) can be defined by

$$M\ddot{x}+C\dot{x}+Kx = Gu(t) \tag{7.1}$$

where M is the mass matrix, C the damping matrix, K the stiffness matrix and G the input matrix. The matrices, M, C and K will be $n\times n$, G is $n\times m$, x(t) is $n\times 1$ and u(t) is $m\times 1$. The output vector y(t) will be given by

$$y(t) = Hx(t) \tag{7.2}$$

with H a $\ell\times n$ matrix and y(t) a $\ell\times 1$ vector. G and H will be considered as constant, matrices with known values.

The input and output variables, u(t) and y(t), may be sampled with values taken at a fixed sampling rate, f_s, or at specified time intervals which are not equal. If f_s is fixed, then samples are available every T seconds. The nonequal interval sampling strategy will be to take samples at a set of specified times, t_i, where $t_{i+1}-b_i \neq T$ for all i. The two sampling strategies will be denoted as uniform and nonuniform sampling rates.

If uniform sampling is chosen, the mathematical theory for the linear system should be formulated in the z-domain. It can then be shown that

$$Y(z) = T(z)U(z) = Hx(z) \tag{7.3}$$

where T(z) is the transfer function in the z-domain given by

$$\begin{aligned} T(z) = Z\{H[Ms^2+Cs+K]^{-1}G\} = HM^{1/2}\{&(I-\exp(\beta_1 T)[\cos(\beta_2 T) \\ &- \beta_1\beta_2^{-1}\sin(\beta_2 T)]z+\exp(\beta_1 T)[\exp(\beta_1 T)-(\cos(\beta_2 T)-\beta_1\beta_2^{-1}\sin(\beta_1 T))]\}. \\ &\cdot \{z^2 I-2z\exp(\beta_1 T)\cos(\beta_2 T)+\exp(2\beta_1 T)\}^{-1}M^{1/2}G \ . \end{aligned} \tag{7.4}$$

The details for the derivation of (7.4) are given in the appendix. The major problems associated with this approach is that the parameters of the system are dependent on matrix functions, i.e. $\exp(\beta_1 T)$, $\cos(\beta_2 T)$ and $\sin(\beta_2 T)$. Small errors in the elements of the matrix elements may be magnified in the recovery of β_1 and β_2.

Although the nonuniform sampling strategy appears to be less desirable than the uniform sampling, there exist unique sampling times that leaves

the system in the s-domain. The system parameters can be estimated by application of linear algebra theory. In the Laplace domain, the system equations are

(7.5) $$Y(s) = T(s)U(s) = Hx(s)$$

with

(7.6) $$T(s) = H[Ms^2+Cs+K]^{-1}G$$

The quadrature formulas hold if the sampling times are consistent with the quadrature integration procedures.

7.1 Sampled Values and the s-domain

Assume that the inputs $u(t_i)$ and outputs $y(t_i)$ have been measured at the times t_i specified by the quadrature order N. The system equations can then be defined by

$$(7.1.1) \qquad Y(s) = \sum_{i=1}^{N} w_i r_i^{s-1} y(t_i)$$

$$(7.1.2) \qquad U(s) = \sum_{i=1}^{N} w_i r_i^{s-1} u(t_i)$$

$$(7.1.3) \qquad Y(s) = \sum_{i=1}^{N} w_i r_i^{s-1} Hx(t_i)$$

where H is assumed to be known. The vectors Y(s) and U(s) can be computed from (7.1.1) and (7.1.2) when the sequence of inputs $u(t_i)$ and outputs $y(t_i)$ are known. The state vector $x(t_i)$ can then be computed by use of (7.1.3) although not all $x(t_i)$ can be determined when H is $\ell \times m$ with $Y(s) \neq f[x_1(s), x_2(s), \ldots, x_n(s)]$. Since $y_i(s) \alpha x_i(s)$, then Y(s) will be incomplete in the sense that not all values of $x_i(s)$ are measured.

If H is an identity matrix, then $X(t_i)$ can be determined from (7.1.2) with X(s) then equal to Y(s). It follows that

$$(7.1.4) \qquad [Ms^2+Cs+K]X(s) = Gu(s)$$

where M, C and K are unknown matrices with G known. A set of linear algebraic equations can then be determined with

$$(7.1.5)\quad \begin{bmatrix} X^T(1) & X^T(1) & X^T(1) \\ 4X^T(2) & 2X^T(2) & X^T(2) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ q^2X^T(q) & qX^T(q) & X^T(q) \end{bmatrix} \begin{bmatrix} M^T \\ C^T \\ K^T \end{bmatrix} = \begin{bmatrix} U^T(1)G^T \\ U^T(2)G^T \\ \cdot \\ \cdot \\ \cdot \\ U^T(q)G^T \end{bmatrix}$$

If M, C and K are n×n, then q must be equal to or greater than 3n. Exact values of X(s) and $X(t_1)$ requires that only 3n rows be used as additional information would be redundant. This ideal situation will not occur as the best measuring device can only resolve the measured values to a certain accuracy. Numerical errors will also be introduced. To overcome these sources of errors, a set of overdetermined equations should be used which means that q>3n. Algorithms, such as SVD, are then used to compute the parameters of M, C and K. A version of the SVD algorithm given in Clarebout, [30] has been used with good result, a listing of that program is given in the Appendix.

Measurements of all of the node displacements is uneconomical as well as unfeasible for a large structure. This means that H will be ℓ×n with only ℓ nonzero elements in H under the assumption that ℓ arbitrary node displacements are measured. Suppose that the first ℓ nodes are measured, thus the first ℓ elements of $x(t_1)$ can be computed. The resulting equations from (7.1.3) would then be

$$(7.1.6)\quad \begin{bmatrix} Y(1) \\ Y(2) \\ \vdots \\ Y(q) \end{bmatrix} = \begin{bmatrix} w_1\hat{H} & w_2\hat{H} & w_3\hat{H} & \cdots & w_N\hat{H} \\ w_1r_1\hat{H} & w_2r_1\hat{H} & w_3r_3\hat{H} & \cdots & w_Nr_N\hat{H} \\ \vdots & \vdots & \vdots & \vdots\vdots\vdots & \vdots \\ w_1r_1^{N-1}\hat{H} & w_2r_2^{N-1}\hat{H} & w_3r_3^{N-1}\hat{H} & \cdots & w_Nr_N^{N-1}\hat{H} \end{bmatrix} \begin{bmatrix} \hat{X}(t_1) \\ \hat{X}(t_2) \\ \vdots \\ \hat{X}(t_N) \end{bmatrix}$$

where

$$(7.1.7)\qquad \hat{x}(t) = \begin{bmatrix} x_1(t_1) \\ x_2(t_1) \\ x_3(t_1) \\ \cdot \\ \cdot \\ \cdot \\ x_\ell(t_1) \end{bmatrix} \qquad \hat{H} = \begin{bmatrix} h_{11} & & & & & \\ & h_{22} & & & 0 & \\ & & h_{23} & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & \cdot \\ & 0 & & & & h_{\ell\ell} \end{bmatrix}$$

with the last $n-\ell$ columns of $\hat{H}$ deleted since the elements are zeros. The resulting matrix in (7.1.6) will be $q\times\ell n$ where $q>\ell n$. The matrix $\hat{H}$ is diagonal with sensor gains along the diagonal. As stated earlier, the displacement sensors and force transducers are collocated.

Consider now the mass, damping and stiffness matrices in the form

$$(7.1.8)\qquad M = \text{diag}\langle m_{11}\ \ m_{22}\ \ m_{33}\ \ \ldots\ \ m_{nn}\rangle$$

$$(7.1.9)\qquad C = \text{tridiag}\left\langle \begin{matrix} & c_{12} & & c_{23} & & \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ c_{11} & & c_{22} & & c_{33} & \cdot\ \cdot\ \cdot\ \cdot\ \cdot & c_{nn} \\ & c_{21} & & c_{32} & & \cdot\ \cdot\ \cdot\ \cdot\ \cdot \end{matrix} \right\rangle$$

$$(7.1.10)\qquad K = \text{tridiag}\left\langle \begin{matrix} & k_{12} & & k_{23} & & \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ k_{11} & & k_{22} & & k_{33} & \cdot\ \cdot\ \cdot\ \cdot\ \cdot & k_{nn} \\ & k_{21} & & k_{32} & & \cdot\ \cdot\ \cdot\ \cdot\ \cdot \end{matrix} \right\rangle$$

The first two equations of MS^2+CS+K would then be

$$(7.1.11a)\qquad m_{11}s^2y_1(s)+c_{11}sy_1(s)+c_{12}sy_2(s)+k_{11}y_1(s)+k_{12}y_2(s) = u_1(s)$$

$$(7.1.11b) \qquad m_{22}s^2y_2(s)+c_{21}sy_1(s)+c_{22}sy_2(s)-c_{23}sy_3(s)-k_{21}y_1(s)-k_{22}y_2(s)$$

$$+ k_{23}y_3(s) = u_2(s)$$

where it has been assumed that $y_1(s) = x_1(s)$.

Equation (7.11a) could then be used to write a set of algebraic equations with s as the varying parameter with

$$(7.1.12) \quad \begin{bmatrix} y_1(1) & y_1(1) & y_2(1) & y_1(1) & y_2(1) \\ 4y_1(2) & 2y_1(2) & 2y_2(2) & y_1(2) & y_2(2) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ q^2y_1(q) & qy_1(q) & qy_2(q) & y_1(q) & y_2(q) \end{bmatrix} \begin{bmatrix} m_{11} \\ c_{11} \\ c_{12} \\ k_{11} \\ k_{12} \end{bmatrix} = \begin{bmatrix} u_1(1) \\ u_i(2) \\ \cdot \\ \cdot \\ u_1(q) \end{bmatrix}$$

The values of m_{ii} will be the same, as will c_{ii} and k_{ii}, for the simple structure with evenly spaced nodes, similarily $c_{ij} = c_{ji}$ and $k_{ij} = k_{ji}$. The above equation will be the only one required for complete determination of the structure parameters.

Suppose now that a beam has equally spaced nodes except for the end ones which are located at one-half spacing from the ends of the beam. The elements m_{11}, c_{11}, k_{11}, c_{12}, c_{21}, k_{12} and k_{21} will not be equal to m_{22}, c_{22}, k_{22}, c_{23}, c_{32}, k_{23} and k_{32}. The first two equations are then used to compute the parameters with (7.1.12) being solved first followed by the solution of the algebraic equations of (7.1.11b).

If all of the coefficients in M, C and K are different, a full set of measurements will be required. The important property of the parameter identification procedure is that the structures symmetry can be utilized to reduce the dimensions of the matrices thus permitting large space structures to be identified.

7.2 Numerical Results

The identification algorithm described in the previous section was used to identify the coefficients of the M, C and K matrices as given in Table 7.1. The algorithm was first used to identify all 36 coefficients by equation (7.1.5). The results were fair in the sense that the accuracy was approximately one part in 10^4 or an error of 0.1%. The vectors x(s) and u(s) were exact for the numerical test. The resulting matrix equation Ax = b as given in (7.1.5) was with A a 18×19 matrix, X a 6×6 matrix, and U a 6×19 matrix.

The second test performed used the structure of the matrices M, C and K as explained in the text and given by equation (7.1.12). The estimated parameters are given in Table 7.2 where the diagonal and tridiagonal elements only were computed, all zeros were expressed as zeros. The parameters were correct to 12 digits with all computations in double precision on a Honeywell 60/60 machine (36 bit word length).

The elements of the C matrix were then reduced by two orders of magnitude to test the algorithm for less damping. The results are given in Table 7.3. Table 7.4 gives the estimated values when the stiffness matrix elements were increased by one order of magnitude with C the same as given in Table 7.4. The estimated values of the parameters in C were in error in the eight digit indicating that lightly damped structures will be more difficult to identify by the algorithm.

The test matrices are not representative of any structure. The model used was for convenience. Further studies will be made to test the effect of noise on the estimated parameters.

THE M MATRIX

0.15000000000000D 01	0.	0.	0.	0.	0.
0.	0.10000000000000D 00	0.	0.	0.	0.
0.	0.	0.40000000000000D 00	0.	0.	0.
0.	0.	0.	0.20000000000000D 01	0.	0.
0.	0.	0.	0.	0.80000000000000D 00	0.
0.	0.	0.	0.	0.	0.13000000000000D 01

THE C MATRIX

0.10500000000000D 02	-0.45000000000000D 01	0.	0.	0.	0.
-0.45000000000000D 01	0.14700000000000D 02	-0.10200000000000D 02	0.	0.	0.
0.	-0.10200000000000D 02	0.12600000000000D 02	-0.24000000000000D 01	0.	0.
0.	0.	-0.24000000000000D 01	0.24000000000000D 02	-0.21600000000000D 02	0.
0.	0.	0.	-0.21600000000000D 02	0.22500000000000D 02	-0.90000000000000D 00
0.	0.	0.	0.	-0.90000000000000D 00	0.14100000000000D 02

THE K MATRIX

0.15000000000000D 04	-0.50000000000000D 03	0.	0.	0.	0.
-0.50000000000000D 03	0.12500000000000D 04	-0.75000000000000D 03	0.	0.	0.
0.	-0.75000000000000D 03	0.10500000000000D 04	-0.30000000000000D 03	0.	0.
0.	0.	-0.30000000000000D 03	0.11000000000000D 04	-0.80000000000000D 03	0.
0.	0.	0.	-0.80000000000000D 03	0.10000000000000D 04	-0.20000000000000D 03
0.	0.	0.	0.	-0.20000000000000D 03	0.80000000000000D 03

Table 7.1 The M, C and K Matrices Used in the Identification.

THE M MATRIX

0.15000000000000D 01	0.	0.	0.	0.	0.
0.	0.10000000000000D 00	0.	0.	0.	0.
0.	0.	0.40000000000000D 00	0.	0.	0.
0.	0.	0.	0.20000000000000D 01	0.	0.
0.	0.	0.	0.	0.80000000000000D 00	0.
0.	0.	0.	0.	0.	0.13000000000000 01

THE C MATRIX

0.19500000000000D 02	-0.45000000000000D 01	0.	0.	0.	0.
-0.45000000000000D 01	0.14700000000000D 02	-0.10200000000000D 02	0.	0.	0.
0.	-0.10200000000000D 02	0.12600000000000D 02	-0.24000000000000D 01	0.	0.
0.	0.	-0.24000000000000D 01	0.24000000000000D 02	-0.21600000000000D 02	0.
0.	0.	0.	-0.21600000000000D 02	0.22500000000000D 02	-0.89999999999999D 00
0.	0.	0.	0.	-0.89999999999999D 00	0.14100000000000D 02

THE K MATRIX

0.15000000000000D 04	-0.50000000000000D 03	0.	0.	0.	0.
-0.50000000000000D 03	0.12500000000000D 04	-0.75000000000000D 03	0.	0.	0.
0.	-0.75000000000000D 03	0.10500000000000D 04	-0.30000000000000D 03	0.	0.
0.	0.	-0.30000000000000D 03	0.11000000000000D 04	-0.80000000000000D 03	0.
0.	0.	0.	-0.80000000000000D 03	0.10000000000000D 04	-0.20000000000000D 03
0.	0.	0.	0.	-0.20000000000000D 03	0.60000000000000D 03

Table 7.2 Estimated Values of M, C and K for the Matrices Given in Table 7.1.

THE M MATRIX

0.15000000000000D 01	0.	0.	0.	0.	0.
0.	0.10000000000000D 00	0.	0.	0.	0.
0.	0.	0.40000000000000D 00	0.	0.	0.
0.	0.	0.	0.20000000000000D 01	0.	0.
0.	0.	0.	0.	0.80000000000000D 00	0.
0.	0.	0.	0.	0.	0.13000000000000D 01

THE C MATRIX

0.10499999999998D 00	-0.44999999999992D-01	0.	0.	0.	0.
-0.44999999999992D-01	0.14699999999997D 00	-0.10199999999998D 00	0.	0.	0.
0.	-0.10199999999998D 00	0.12000000000000D 00	-0.24000000000018D-01	0.	0.
0.	0.	-0.24000000000018D-01	0.24000000000008D 00	-0.21600000000000D 00	0.
0.	0.	0.	-0.21600000000000D 00	0.22500000000007D 00	-0.90000000000000D-02
0.	0.	0.	0.	-0.90000000000000D-02	0.14100000000002D 00

THE K MATRIX

0.15000000000000D 04	-0.50000000000000D 03	0.	0.	0.	0.
-0.50000000000000D 03	0.12500000000000D 04	-0.75000000000000D 03	0.	0.	0.
0.	-0.75000000000000D 03	0.10500000000000D 04	-0.30000000000000D 03	0.	0.
0.	0.	-0.30000000000000D 03	0.11000000000000D 04	-0.80000000000000D 03	0.
0.	0.	0.	-0.80000000000000D 03	0.10000000000000D 04	-0.20000000000000D 03
0.	0.	0.	0.	-0.20000000000000D 03	0.80000000000000D 03

Table 7.3 Estimated Values of M, C and K with Less Damping than in Table 7.1.

THE M MATRIX

0.15000000000000D 01	0.	0.	0.	0.	0.
0.	0.99999999999999D-01	0.	0.	0.	0.
0.	0.	0.39999999999140 00	0.	0.	0.
0.	0.	0.	0.20000000000070 01	0.	0.
0.	0.	0.	0.	0.80000000000080 00	0.
0.	0.	0.	0.	0.	0.13000000000010 01

THE C MATRIX

0.10500000016980 00	-0.45000000069790-01	0.	0.	0.	0.
-0.45000000069790-01	0.14700000023960 00	-0.10200000016990 00	0.	0.	0.
0.	-0.10200000016990 00	0.12599999987250 00	-0.23999999702640-01	0.	0.
0.	0.	-0.23999999702640-01	0.23999999913870 00	-0.21599999943600 00	0.
0.	0.	0.	-0.21599999943600 00	0.22499999938100 00	-0.89999999449860-02
0.	0.	0.	0.	-0.89999999449860-02	0.14099999987540 00

THE K MATRIX

0.15000000000000D 06	-0.50000000000000D 05	0.	0.	0.	0.
-0.50000000000000D 05	0.12500000000000D 06	-0.75000000000000D 05	0.	0.	0.
0.	-0.75000000000000D 05	0.10500000000000D 06	-0.30000000000000D 05	0.	0.
0.	0.	-0.30000000000000D 05	0.11000000000000D 06	-0.80000000000000D 05	0.
0.	0.	0.	-0.80000000000000D 05	0.10000000000000D 06	-0.20000000000000D 05
0.	0.	0.	0.	-0.20000000000000D 05	0.60000000000000D 05

Table 7.4 Estimated Values of M, C and K with Very Light Damping.

8. Summary

The research effort under NASA Grant NSG-1602 has continued with the major objective of developing numerical algorithms for large space structures. The research is directed to finding algorithms for analysis, synthesis and identification of large structures.

Since the finite element model gives rise to the equation $M\ddot{x}+Cx+Kx=f(t)$, considerable effort has been expended toward understanding matrix polynomial which play an important role inthe obtaining the dynamics of systems. In addition, the dimensions of the matrix coefficients are lower by one half, than the state variable model. Analysis of the structure from the state variable model appears to be out of the question when one assumes that the matrices M, C and K, although sparse, may exceed the size of 1000×1000. It is difficult to handle matrices of this dimension and much more difficult to manipulate matrices of twice the dimension.

The decoupling algorithm for analysis has continued but the procedure does not appear to have the efficiency required for the desired analysis. The research has been redirected with more emphasis on the theory of matrix polynomials.

This report discusses the decoupling algorithm, the theory of Laplace transforms of matrix polynomials and identification of the three matrices M, C and K. The identification algorithm is promising and work will continue on that aspect of the large space structure. Data from the NASA LRC beam facility will be obtained in the near future and the algorithm will be tested on this data. Work will continue on the assignment of damping to a structure which requires a better understanding of transformations in a multi-dimensional space.

References

1. L. R. Anderson, Decoupling of Two-Time Scale Linear Systems, Proc. 1978 Joint. Auto. Contr. Conf., Vol. 4, Philadelphia, Pa., pp. 153-164.

2. L. R. Anderson, Decoupling and Order Reduction of Two-Time-Scale Control Systems, Proc. Second. Annual Workshop Info. Linkage Appl. Math and Industry, Naval Postgraduate School, Monterey, Cal., Feb. 20-24, 1979.

3. H. B. Lee, Matrix Filtering as an Aid to Numerical Integration, Proc. IEEE, Vol. 55, pp. 1826-1831, 1967.

4. J. D. Roberts, Linear Model Reduction and Solution of Algebraic Riccati Equations by Use of the Sign Function, Cambridge U., Rept. CUED/B-CONTROL, TR-13, 1971.

5. K. S. Yoo, Development of a Numerical Algorithm for Uncoupling Constant Coefficient State Equations of Control Theory, Dissertation, Dept. of Elec. Eng., U. of Houston, May 1974.

6. C. Popeeva and L. Lupas, Decomposition and Reduction of Linear Systems by the Matrix Sun Function, Revue Roumaine des Sciences Tech, Serie Electrotechnique et Energetique, Vol. 20, pp. 567-576, 1975 (English).

7. D. Russell, An Extended Block-Diagonalization Theorem, Proc. US-Japan Seminar on Diff. and Func. Eqns., U. of Minnesota, pp. 567-575, 1967.

8. R. A. Yackel and R. V. Kokotović, A Boundary Layer Method for the Matrix Riccati Equation, IEEE Trans. Auto. Cont., Vol. AC-18, pp. 17-23, 1973.

9. H. Nicholson, Structure of Kron's Polyhedron Model and the Scattering Problem, Proc. IEE, Vol. , pp. , 1973.

10. N. R. Sandell, Jr., Decomposition vs Decentralization in Large-Scale System Theory, Proc. Cont. on Dec. and Cont., pp. 1043-1046, 1976.

11. A. N. Beavers, Jr. and E. D. Denman, A New Solution Method for Quadratic Matrix Equations, Math. Biosci., Vol. 20, pp. 135-143, 1974.

12. A. N. Beavers, Jr., and E. D. Denman, A New Similarity Transformation Method for Eigenvalues and Eigenvectors, Math. Biosci., Vol. 21, pp. 143-169, 1974.

13. A. N. Beavers, Jr., and E. D. Denman, Asymptotic Solutions to the Matrix Riccati Equation, Math. Biosci., Vol. 20, pp. 339-344, 1974.

14. A. N. Beavers, Jr. and E. D. Denman, A New Solution Method for the Lypanov Matrix Equation, SIAM J. Appl. Math., pp. 416-421, 1975.

15. A. N. Beavers, Jr. and E. D. Denman, The Matrix Sign Function and Computations in Systems, Appl. Math. Comp., Vol. 2, pp. 63-94, 1976.

16. E. D. Denman, An Additional Algorithm for a System of Coupled Algebraic Matrix Riccati Equations, IEEE Trans. Comp., Vol. , pp. 91-93, 1976.

17. E. D. Denman, Roots of Real Matrices, Linear Algebra and Appl., (to appear).

18. E. D. Denman and J. Leyva-Ramos, Roots of Real and Complex Matrices (submitted for publication).

19. R. L. Mattheys, Stability Analysis Via the Extended Matrix Sign Function, Proc. IEEE , 1978.

20. W. D. Hoskins and D. J. Walton, A Faster Method for Computing the Square Root of a Matrix, IEEE Trans. Auto. Cont., AC-23, pp. 494-495, 1978.

21. A. Halberberg and V. Bar-Ness, Solution of the Discrete Regulator Problem Using the Sign Matrix Function, Electronic Letters, Vol. 14, pp. 286-287, 1978.

22. F. V. Atkinson, Multiparameter, Eigenvalue Problems, Academic Press, N. Y., 1972.

23. E. D. Denman and J. Leyva-Ramos, Spectral Decomposition of a Matrix Using the Generalized Sign Matrix, (accepted for publication in Applied Math. and Computations).

24. F. R. Gantmacher, Theory of Matrices, Vol. 1, Chelsa Publishing Co., N. Y., pp. 95-100.

25. R. T. Gregory and D. L. Karney, A Collection of Matrices for Testing Computational Algorithms, Wiley-Interscience, N. Y., p. 660, 1969.

26. R. Penrose, A Generalized Inverse of a Matrix, Proc. Cambridge Philos. Soc., Vol. 51, pp. 406-413, 1955.

27. R. E. Bellman, R. E. Kalaba and J. A. Lockett, Numerical Inversion of the Laplace Transform, American Elsevier Publishing Company, Inc., New York, N. Y., 1966.

28. G. E. Cook, E. D. Denman and H. M. Carr, Numerical Inversion of Laplace Transforms by the Laguerre-Gauss Quadratic Approximation, IEEE Transactions on Automatic Control, Vol. AC-12, October, 1967.

29. A. S. Kronrod, Nodes and Weights of Quadrature Formulas, Consultants Bureau Enterprises, Inc., New York, N. Y., 1965.

30. J. F. Clarebout, Fundamentals of Geophysical Data Processing, McGraw-Hill, New York, N. Y., 1976.

31. G. E. Forsythe, M. A. Malcolm, and C. B. Moler, Computer Methods for Mathematical Computations, Prentice-Hall, Englewood Cliffs, N. J., 1977.

A.1 Derivation of the Discrete Transfer Function

Let T(s) denote the transfer function in the Laplace domain with

$$T(s) = H[Ms^2Cs+K]^{-1}G$$
$$= HM^{1/2}[Is^2+M^{-1/2}CM^{-1/2}s+M^{-1/2}KM^{-1/2}]M^{1/2}G \tag{A.1.1}$$
$$= HM^{1/2}[Is^2+A_1s+A_2]^{-1}M^{1/2}G$$

where A_1 and A_2 are as defined. The discrete transfer function T(z) is given by

$$T(z) = HM^{1/2}(1-z^{-1})Z\{s^{-1}[Is^2+A_1s+A_2]^{-1}\}M^{1/2}G \tag{A.1.2}$$

The term inside the braces can be expanded into a partial fraction with

$$s^{-1}[Is^2+A_1s+A_2]^{-1} = A_2^{-1}s^{-1}-A_2^{-1}[s+A_1][Is^2+A_1s+A_2]^{-1} \tag{A.1.3}$$

which can be rewritten as

$$s^{-1}[Is^2+A_1s+A_2]^{-1} = A_2^{-1}\{Is^{-1}-(s+A_1/2)[(Is+A_1/2)^2+A_2-A_1^2/4]$$
$$-A_1/2[A_2-A_1^2/4]^{-1/2}[A_2-A_1^2/4]^{1/2}[(Is+A_1/2)^2+A_2-A_1^2/4]^{-1}\} \tag{A.1.4}$$

for which the z-transform is

$$Z\{s^{-1}[Is^2+A_1s+A_2]^{-1}\} = A_2^{-1}\{z(z-1)^{-1}I-[z^2I-z\exp(-\frac{A_1}{2}T)\cos(A_2-\frac{A_1^2}{4})^{1/2}T]$$
$$\cdot[z^2I-2z\exp(-\frac{A_1}{2}T)\cos(A_2-A_1^2/4)^{1/2}T+\exp(-A_1T)]-\frac{A_1}{2}[A_2-A_1^2/4]^{-1/2} \tag{A.1.5}$$
$$\cdot[z\exp(-\frac{A_1}{2}T)\sin(A_2-\frac{A_1^2}{4})^{1/2}T][z^2I-2z\exp(-\frac{A_1}{2}T)\cos(A_2-\frac{A_1^2}{4})^{1/2}T$$
$$+\exp(-A_1T)]\}$$

Since (z-1) is a scalar, (A.1.5) can be rearranged* as

$$(A.1.5)\quad Z\{s^{-1}[Is^2+A_1s+A_2]\} = A_2^{-1}\{z[z^2I-2z\exp(-\tfrac{A_1}{2}T)\cos(A_2-\tfrac{A_1^2}{4})^{1/2}T + \exp(-A_1T)]-[z^2I-z\exp(-\tfrac{A_1}{2}T)\cos(A_2-\tfrac{A_1^2}{4})^{1/2}T][z-1] - [z\exp(-\tfrac{A_1}{2}T)\sin(A_2-\tfrac{A_1^2}{4})^{1/2}T][z-1]\}[z-1]^{-1}[z^2I-2z\exp(-\tfrac{A_1}{2}T)\cos(A_2-\tfrac{A_1^2}{4})^{1/2}T + \exp(-A_1T)]$$

which is

$$(A.1.6)\quad Z\{s^{-1}[Is^2+A_1s+A_2]\} = \{(I-\exp(-\tfrac{A_1}{2}T)[\cos(A_2-\tfrac{A_1^2}{4})^{1/2}T - \tfrac{A_1}{2}(A_2-\tfrac{A_1^2}{4})^{-1/2}\sin(A_2-\tfrac{A_1^2}{4})^{1/2}T])z+\exp(-\tfrac{A_1}{2}T)(\exp(-\tfrac{A_1}{2}T) - [\cos(A_2-\tfrac{A_1^2}{4})^{1/2}T-\tfrac{A_1}{2}(A_2-\tfrac{A_1^2}{4})^{-1/2}\sin(A_2-\tfrac{A_1^2}{4})^{1/2}T])z\}\{(z-1)^{-1} \cdot [z^2I-2z\exp(-\tfrac{A_1}{2}T)\cos(A_2-\tfrac{A_1^2}{4})^{1/2}T+\exp(-A_1T)]^{-1}\}$$

where

$$(A.1.7)\quad A_1 = M^{-1/2}C_M^{-1/2}$$

$$(A.1.8)\quad A_2 = M^{-1/2}K_M^{-1/2}$$

Defining

$$(A.1.9)\quad \beta_1 = -\frac{A_1}{2} = -\frac{1}{2}M^{-1/2}C_M^{-1/2}$$

(A.1.10) $$\beta_2 = -(A_2 - \frac{A_1^2}{4})^{1/2} = -[M^{-1/2}KM^{-1/2} - \frac{1}{4}M^{-1}CM^{-1}]^{1/2}$$

(A.1.6) can be written in the compact form

(A.1. $$Z\{s^{-1}[Is^2+A_1s+A_2]^{-1}\} = \{(I-\exp(\beta_1T)[\cos(\beta_2T)-\beta_1\beta_2^{-1}\sin(\beta_2T))z^2$$

$$+ \exp(\beta_1T(\exp(\beta_1T)-[\cos(\beta_2T)-\beta_1\beta_2^{-1}\sin(\beta_2T)])z\}\{(z-1)$$

$$\cdot [z^2I-2z\exp(\beta_1T)\cos(\beta_2T)+\exp(2\beta_1T)]\}^{-1}$$

Returning to (A.1.2), the discrete transfer function T(z) is then

(A.1.11) $$T(z) = HM^{1/2}\{(I-\exp(\beta_1T)[\cos(\beta_2T)-\beta_1\beta_2^{-1}\sin(\beta_2T))z$$

$$+ \exp(\beta_1T)(\exp(\beta_1T)-[\cos(\beta_2T)-\beta_1\beta_2^{-1}\sin(\beta_2T))\}\{z^2I-2z\exp(\beta_1T)$$

$$\cdot \cos(\beta_2T)+\exp(2\beta_1T)\}^{-1}M^{1/2}G$$

where β_1 and β_2 are defined as in (A.1.9) and (A.1.10).

APPENDIX 2

ROOTS OF THE SHIFTED LEGENDRE POLYNOMIALS AND CHRISTOFFEL WEIGHTS

ROOTS	WEIGHTS
N = 3	
1.1270166E-1	2.7777777E-1
5.0000000E-1	4.4444444E-1
8.8729833E-1	2.7777777E-1
N = 5	
4.6910077E-2	1.1846344E-1
2.3076534E-1	2.3931433E-1
5.0000000E-1	2.8444444E-1
7.6923465E-1	2.3931433E-1
9.5308992E-1	1.1846344E-1
N = 7	
2.5446043E-2	6.4742483E-2
1.2923440E-1	1.3985269E-1
2.9707742E-1	1.9091502E-1
5.0000000E-1	2.0897959E-1
7.0292257E-1	1.9091502E-1
8.7076559E-1	1.3985269E-1
9.7455395E-1	6.4742483E-2
N = 9	
1.5919880E-2	4.0637194E-2
8.1984446E-2	9.0324080E-2
1.9331428E-1	1.3030534E-1
3.3787328E-1	1.5617353E-1
5.0000000E-1	1.6511967E-1
6.6212671E-1	1.5617353E-1
8.0668571E-1	1.3030534E-1
9.1801555E-1	9.0324080E-2
9.8408011E-1	4.0637194E-2

ROOTS	WEIGHTS
N = 11	
1.0885670E-2	2.7834283E-2
5.6468700E-2	6.2790184E-2
1.3492399E-1	9.3145105E-2
2.4045193E-1	1.1659688E-1
3.6522842E-1	1.3140227E-1
5.0000000E-1	1.3646254E-1
6.3477157E-1	1.3140227E-1
7.5954806E-1	1.1659688E-1
8.6507600E-1	9.3145105E-2
9.4353129E-1	6.2790184E-2
9.8911432E-1	2.7834283E-2
N = 13	
7.9084726E-3	2.0242002E-2
4.1200800E-2	4.6060749E-2
9.9210954E-2	6.9436755E-2
1.7882533E-1	8.9072990E-2
2.7575362E-1	1.0390802E-1
3.8477084E-1	1.1314159E-1
5.0000000E-1	1.1627577E-1
6.1522915E-1	1.1314159E-1
7.2424637E-1	1.0390802E-1
8.2117466E-1	8.9072990E-2
9.0078904E-1	6.9436755E-2
9.5879919E-1	4.6060749E-2
9.9209152E-1	2.0242002E-2
N = 15	
6.0037409E-3	1.5376620E-2
3.1363303E-2	3.5183023E-2
7.5896708E-2	5.3579610E-2
1.3779113E-1	6.9785338E-2
2.1451391E-1	8.3134602E-2
3.0292432E-1	9.3080500E-2
3.9940295E-1	9.9215742E-2
5.0000000E-1	1.0128912E-1
6.0059704E-1	9.9215742E-2
6.9707567E-1	9.3080500E-2
7.8548608E-1	8.3134602E-2
8.6220886E-1	6.9785338E-2
9.2410329E-1	5.3579610E-2
9.6863669E-1	3.5183023E-2
9.9399625E-1	1.5376600E-2

NEGATIVES OF LOGARITHMS OF ROOTS OF SHIFTED LEGENDRE POLYNOMIALS

N = 3

2.183011
0.693147
0.119574

N = 5

3.059523
1.466354
0.693147
0.262359
0.048046

N = 7

3.671195
2.046127
1.213762
0.693147
0.352509
0.138382
0.025775

N = 9

4.140187
2.501226
1.643438
1.085084
0.693147
0.412298
0.214821
0.085541
0.016048

N = 11

4.520308
2.874069
2.003044
1.425235
1.007232
0.693147
0.454490
0.275032
0.144938
0.058126
0.010945

N = 13

4.839821
3.189298
2.310507
1.721346
1.288247
0.955107
0.693147
0.485760
0.322624
0.197019
0.104484
0.042074
0.007940

N = 15

5.115372
3.462117
2.578382
1.982016
1.539381
1.194272
0.917784
0.693147
0.509831
0.360861
0.241453
0.148258
0.078931
0.031866
0.006022

Appendix 3 Ax=b Version of Golub Algorithm

```
        SUBROUTINE GOLUB(A,X,B,M,N)
C       A(M,N) ; B(M) GIVEN WITH M>N SOLVES FOR X(N) SUCH THAT
C            J IS A MINIMUM
C       METHOD OF G. GOLUB, NUMERISCHE MATHEMATIK 7, 206-216 (1965)
        IMPLICIT DOUBLE PRECISION (D)
        REAL A(10,10),X(10),B(10),U(50)
C.....PERFORM N ORTHOGONAL TRANSFORMATIONS TO A(.,.) TO
C.....UPPER TRIANGULARIZE THE MATRIX
        DO 3010 K=1,N
        DSUM=0.0D0
        DO 1010 I=K,M
        DAJ=A(I,K)
 1010   DSUM=DSUM+DAJ**2
        DAI=A(K,K)
        DSIGMA=DSIGN(DSQRT(DSUM),DAI)
        DBI=DSQRT(1.0D0+DAI/DSIGMA)
        U(K)=DBI
        FACT=DFACT
        KPLUS=K+1
        DO 1020 I=KPLUS,M
 1020   U(I)=FACT*A(I,K)
C..... I-U*U**T IS SYMMETRIC, ORTHOGONAL MATRIX WHICH WHEN APPLIED
C.....TO A(.,.) WILL ANNIHILATE THE ELEMENTS BELOW THE DIAGONAL I
        DO 2030 J=K,N
C.....APPLY THE ORTHOGONAL TRANSFORMATION
        FACT=0.0
        DO 2010 I=K,M
 2010   FACT=FACT+U(I)*A(I,J)
        DO 2020 I=K,M
 2020   A(I,J)=A(I,J)-FACT*U(I)
 2030   CONTINUE
        FACT=0.0
        DO 2040 I=K,M
 2040   FACT=FACT+U(I)*B(I)
        DO 2050 I=K,M
 2050   B(I)=B(I)-FACT*U(I)
 3010   CONTINUE
C.....BACK SUBSTITUTE TO RECURSIVELY YIELD X(.)
        X(N)=B(N)/A(N,N)
        LIM=N-1
        DO 4020 I=1,LIM
        IROW=N-I
        SUM=0.0
        DO 4010 J=1,I
 4010   SUM=SUM+X(N-J+1)*A(IROW,N-J+1)
 4020   X(IROW)=(B(IROW)-SUM)/A(IROW,IROW)
        RETURN
        END
```

From Clarebout, [30].