

IMPROVEMENTS IN BLOCK-KRYLOV RITZ VECTORS AND THE  
BOUNDARY FLEXIBILITY METHOD OF COMPONENT SYNTHESIS

by

KELLY SCOTT CARNEY

Submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy

Thesis Adviser: Dr. Arthur A. Huckelbridge

Department of Civil Engineering  
CASE WESTERN RESERVE UNIVERSITY

May, 1997



# IMPROVEMENTS IN BLOCK-KRYLOV RITZ VECTORS AND THE BOUNDARY FLEXIBILITY METHOD OF COMPONENT SYNTHESIS

Abstract

by

KELLY SCOTT CARNEY

A method of dynamic substructuring is presented which utilizes a set of static Ritz vectors as a replacement for normal eigenvectors in component mode synthesis. This set of Ritz vectors is generated in a recurrence relationship, proposed by Wilson, which has the form of a block-Krylov subspace. The initial seed to the recurrence algorithm is based upon the boundary flexibility vectors of the component. Improvements have been made in the formulation of the initial seed to the Krylov sequence, through the use of block-filtering. A method to shift the Krylov sequence to create Ritz vectors that will represent the dynamic behavior of the component at target frequencies, the target frequency being determined by the applied forcing functions, has been developed. A method to terminate the Krylov sequence has also been developed. Various orthonormalization schemes have been developed and evaluated, including the Cholesky/QR method. Several auxiliary theorems and proofs

which illustrate issues in component mode synthesis and loss of orthogonality in the Krylov sequence have also been presented.

The resulting methodology is applicable to both fixed and free-interface boundary components, and results in a general component model appropriate for any type of dynamic analysis. The accuracy is found to be comparable to that of component synthesis based upon normal modes, using fewer generalized coordinates. In addition, the block-Krylov recurrence algorithm is a series of static solutions and so requires significantly less computation than solving the normal eigenspace problem. The requirement for less vectors to form the component, coupled with the lower computational expense of calculating these Ritz vectors, combine to create a method more efficient than traditional component mode synthesis.

## Acknowledgements

I would like to thank my wife, Dorothy, and my children, Colin, Molly, and Abigail for their many years of patience and support. The same years of patience and support were given to me by my advisor, Dr. Arthur Huckelbridge, in addition to his guidance and many suggestions.

I would like to thank Dr. Charles Lawrence and Dr. Ayman Abdallah for their beneficial reviews of an early version of this dissertation. I would also like to thank Isam Yunis for engaging in many thought provoking discussions concerning Ritz vectors and Krylov blocks.

My supervisors, Dr. James McAleese, and Dr. Francis Shaker both gave me many years of mentoring and support which enabled me to pursue this work.

I would like to thank the members of my dissertation committee, Dr. Arthur Huckelbridge, Dr. Robert Mullen, Dr. Dario Gasparini, and Dr. Maurice Adams.

I would also like to thank William Hughes for his encouragement and Anne McNelis for the EPS Radiator applied load vector.

## Table of Contents

	page
Abstract .....	ii
Acknowledgements .....	iv
Table of Contents .....	v
List of Figures .....	ix
List of Tables .....	xii
Chapter 1. Introduction .....	1
Chapter 2. Literature Review .....	7
2.1. Introduction .....	7
2.2. Review of Boundary Flexibility Component Mode Synthesis .....	12
2.2.1. Fixed Interface Methodology .....	12
2.2.2. Fixed Interface Methodology for Components with No Rigid Body Modes .....	16
2.2.3. Free Interface Methodology for Components with Rigid Body Modes .....	19
2.3. Relationship to Lanczos Eigenvalue Extraction .....	22
2.4. Orthogonalization .....	24
2.4.1. Loss of Orthogonality .....	25
2.4.2. Orthonormalization .....	27
2.4.3. Cholesky/QR Decomposition .....	28

2.5.	Shifting .....	29
2.6.	Spectrum Slicing .....	30
2.7.	Krylov Sequence Termination Techniques .....	31
Chapter 3.	Theoretical Development .....	33
3.1.	Introduction .....	33
3.2.	The Exact Nature of the Methodology .....	33
Chapter 4.	Orthogonalization, the Krylov Sequence and Static Ritz Vectors .	39
4.1.	Introduction .....	39
4.2.	Orthogonalization .....	39
4.2.1.	Linear Independence and the Loss of Orthogonality .....	41
4.2.2.	Use of the Euclidean Norm for Normalization .....	45
4.2.3.	Gram-Schmidt Failure and Reorthogonalization .....	46
4.2.4.	Various Gram-Schmidt Orthogonalization Strategies .....	48
4.2.5.	Cholesky/QR Orthogonalization .....	49
4.3.	Issues Concerning the Use of Blocks in the Boundary	
	Flexibility Method .....	54
4.3.1	Dependence of Vectors Within the Block .....	55
4.3.2	Block Filtering .....	58
4.4.	Summary and Revised Orthogonalization Algorithm .....	62
Chapter 5.	Numerical Examples of Orthonormalization .....	68
5.1.	Introduction .....	68

5.2.	Tools and Programming .....	68
5.3.	Finite Element Models .....	69
5.3.1.	Simple Beam Model .....	69
5.3.2.	Space Station Electrical Power System Radiator .....	70
5.3.3.	Cassini Spacecraft .....	70
5.4.	Numerical Results .....	71
5.4.1.	Numerical Illustrations of Theoretical Properties .....	71
5.4.2.	Numerical Illustrations of Block Issues .....	73
5.4.3.	Simple Beam Numerical Results .....	74
5.4.4.	Timing Comparisons .....	77
Chapter 6.	Target Shifting .....	106
6.1.	Introduction .....	106
6.2.	Theory .....	107
6.3.	Illustrative Example .....	109
6.4.	Targeted Shifting .....	111
6.5.	Shift Strategy .....	114
6.6.	Applied Dynamic Loading of Example Structural Models .....	116
Chapter 7.	Krylov Sequence Termination Techniques .....	128
7.1.	Introduction .....	128
7.2.	Error Criteria and Effective Mass .....	129
7.3.	Modal Density Truncation .....	132

7.4. Determination of Modal Density and Truncation .....	134
Chapter 8. Numerical Examples of Targeted Shifting and Modal Density Truncation .....	137
8.1. Algorithm with Targeted Shifting and Modal Density Truncation	137
8.2. Numerical Results and the Determination of the Fraction $\psi$ .....	137
Chapter 9. Summary and Conclusions .....	159
References .....	162
Appendix A1 .....	165
Appendix A2 .....	170
Appendix B .....	175

## List of Figures

Figure	Page
4.1 Normalized Vectors Obtained from a Krylov Sequence of Order Nine, with $x^2$ as the Iteration Matrix .....	67
5.1 Space Station Electrical Power System Deployed Radiator .....	95
5.2 Space Station EPS Radiator Undeformed Finite Element Model .....	96
5.3 EPS Radiator First Normal Mode, Fixed Interface .....	97
5.4 EPS Radiator Second Normal Mode, Fixed Interface .....	98
5.5 EPS Radiator Third Normal Mode, Fixed Interface .....	99
5.6 Cassini Spacecraft .....	100
5.7 Cassini Spacecraft Finite Element Model .....	101
5.8 Fixed Interface Ritz Vector, Block One, Vector One .....	102
5.9 Fixed Interface Ritz Vector, Block One, Vector Two .....	103
5.10 Fixed Interface Ritz Vector, Block Two, Vector One .....	104
5.11 Fixed Interface Ritz Vector, Block Two, Vector Two .....	105
6.1 Time Domain Beam Model Applied Dynamic Loading .....	118
6.2 Response Spectrum of the Beam Model Applied Dynamic Loading .....	119
6.3 Plume Impingement Dynamic Loading on the EPS Radiator .....	120
6.4 Response Spectrum of the Plume Impingement Dynamic Loading .....	121
6.5 Cassini Spacecraft Pitch Translation Interface Acceleration .....	122
6.6 Cassini Spacecraft Yaw Translation Interface Acceleration .....	123

6.7	Cassini Spacecraft Longitudinal Interface Acceleration .....	124
6.8	Response Spectrum of the Pitch Interface Acceleration .....	125
6.9	Response Spectrum of the Yaw Interface Acceleration .....	126
6.10	Response Spectrum of the Longitudinal Interface Acceleration .....	127
8.1	Beam Example Tip Axial Acceleration Generated Using a 250 Hz Eigenvector Cutoff .....	147
8.2	Beam Example Tip Axial Acceleration Generated Using a $\psi = 1.0$ Ritz Vector Representation .....	148
8.3	Beam Example Tip Lateral Acceleration Generated Using a $\psi = 1.0$ Ritz Vector Representation .....	149
8.4	Beam Example Base Bending Moment Generated Using a $\psi = 1.0$ Ritz Vector Representation .....	150
8.5	EPS Radiator X Tip Acceleration Generated Using a $\psi = 1.0$ Ritz Vector Representation .....	151
8.6	EPS Radiator Z Tip Acceleration Generated Using a $\psi = 1.0$ Ritz Vector Representation .....	152
8.7	EPS Radiator Scissor Beam Bending Moment 1 Generated Using a $\psi = 1.0$ Ritz Vector Representation .....	153
8.8	EPS Radiator Scissor Beam Bending Moment 2 Generated Using a $\psi = 1.0$ Ritz Vector Representation .....	154

8.9	Cassini Oxidizer Tank X Acceleration Generated Using a $\psi = 1.0$	
	Ritz Vector Representation .....	155
8.10	Cassini Oxidizer Tank Y Acceleration Generated Using a $\psi = 1.0$	
	Ritz Vector Representation .....	156
8.11	Cassini Oxidizer Tank Z Acceleration Generated Using a $\psi = 1.0$	
	Ritz Vector Representation .....	157
8.12	High Gain Antenna Strut Axial Force Generated Using a $\psi = 1.0$	
	Ritz Vector Representation .....	158

## List of Tables

Table		Page
4.1	Revised Boundary Flexibility Algorithm Using Orthonormalization With Respect to the Mass Matrix .....	63
4.2	Revised Boundary Flexibility Algorithm Using Orthonormalization With the Euclidean Vector Norm .....	65
5.1	A Comparison of Cantilevered Beam Frequencies (Hz) Obtained From a Static Ritz Component Transformed into Normal Eigenvalues and a Direct Normal Eigenvalue Solution .....	80
5.2	The Largest Off-Diagonal Terms From an Orthogonality Check of the Static Ritz Vectors Representing a Cantilevered Beam, Using Orthonormalization With Respect to the Mass Matrix .....	82
5.3	The Largest Off-Diagonal Terms From an Orthogonality Check of the Static Ritz Vectors Representing a Cantilevered Beam, Using Orthonormalization With the Euclidean Vector Norm .....	83
5.4	The Cross Orthogonality of Static Ritz Vectors Created by Orthonormalizing With the Previous Two Blocks, Using Orthonormalization With Respect to the Mass Matrix .....	84
5.5	The Cross Orthogonality of Static Ritz Vectors Created by Orthonormalizing With the Previous Two Blocks, Using Orthonormalization With the Euclidean Vector Norm .....	85

5.6	The Cross Orthogonality of the First Krylov Block of the EPS Radiator	86
5.7	The Cross Orthogonality of the First Krylov Block of the EPS Radiator, After Block Filtering	89
5.8	Comparison Between the Frequencies of a Cantilevered Beam Obtained From Fixed Interface Component Normal Modes, Ritz Vectors, and Finite Elements	90
5.9	Comparison Between the Frequencies of a Cantilevered Beam Obtained From Free Interface Component Ritz Vectors, and Finite Elements	91
5.10	Comparison Between the Frequencies of a Free-Free Beam Obtained From Free Interface Component Ritz Vectors, and Finite Elements	92
5.11	Comparisons of the Computer Time (seconds) Required to Form a Component of the EPS Radiator Finite Element Model (4000 DOF), Using Selected Orthonormalization Options	93
5.12	Comparisons of the Computer Time Required to Form a Component of the Cassini Spacecraft Finite Element Model (11,100 DOF), Using Selected Orthonormalization Options	94
7.1	Beam Finite Element Model Effective Masses	136
8.1	Revised Boundary Flexibility Algorithm With Targeted Shifting and Modal Density Truncation	142

8.2	Ratios of Responses of the Beam Example Model Using Various Representations to the 250 HZ Cutoff Eigenvector Representation Responses .....	145
8.3	Ratios of Responses of the EPS Radiator Model Using Various Representations to the 150 HZ Cutoff Eigenvector Representation Responses .....	145
8.4	Ratios of Responses of the Cassini Model Using Various Representations to the 150 HZ Cutoff Eigenvector Representation Responses .....	146



## Chapter 1

### Introduction

Component mode synthesis is a methodology for analyzing large structures by separating them into smaller components, reduced representations of which can then be recombined to analyze the entire system. This methodology has become well established and widely used in structural dynamic analysis. The advantages of component mode synthesis include lower computation costs associated with smaller components, and the flexibility of data management gained by working with the discrete components.

The typical component mode synthesis algorithm is briefly described<sup>8</sup>. A large structure is broken into components, with each component having a set of boundary, or interface, points. At these interface points, fixed or free boundary conditions are assumed, and a corresponding set of component normal mode shapes, or eigenvectors, is determined. The eigenvectors are augmented by a set of modes which are associated with the component's boundary flexibility. Depending on whether a fixed or free interface is selected, these modes are the constraint modes or the attachment modes, respectively. The combined set of component normal modes and boundary modes are used to represent the component in subsequent system analysis by using the following transformation process. The combined set of modes form a coordinate

transformation matrix which transforms the physical coordinates of the structural model into a combination of modal coordinates and boundary coordinates. The boundary coordinates are retained in the physical space, so they can be used to couple the components for subsequent system analysis.

A component's size, although smaller than that of the complete structural model, can still be large enough to be computationally expensive. The rapid reduction in cost per calculation in today's digital computers has not necessarily led to a reduction in total computation cost. Instead, engineers have exploited the increased computational resources by creating larger structural and component models. The larger models have allowed for more structural details to be represented, as well as more refined data recovery, but they may be expensive to formulate and analyze. In order to reduce the computational cost associated with large component models, it is desirable to develop more efficient methods of formulation. Since the solution of the normal eigensystem problem requires the largest computational effort in component formulation, it is logical to develop alternate methods which circumvent the eigensystem solution entirely.

A method, which does circumvent the eigensystem solution, has been defined in literature and is briefly described<sup>1,10,30</sup>. The boundary flexibility modes, specifically either the same constraint modes or attachment modes that were mentioned previously, are multiplied by the component mass matrix to create a force matrix. Static analysis is then performed, using this force matrix and the component stiffness matrix, to

obtain a matrix, or block, of vector displacements. A recurrence relationship of matrix multiplications, which have been shown to be a Krylov sequence<sup>10,19</sup>, then defines a series of matrices, or blocks, of vector displacements. The calculated vectors are orthogonalized, using normalized Gram-Schmidt orthogonalization<sup>18</sup>. These vectors, which can also be thought of as static modes or static Ritz vectors, replace the normal modes in the component formulation methodology. Because the static Ritz vectors are calculated in blocks and are based on a Krylov sequence, the subspace defined by these vectors is called a block-Krylov subspace.

The work described in this dissertation originated by identifying areas of potential improvement in the implementation of the existing static Ritz vector, block-Krylov, boundary flexibility methodology. Specific potential improvements in the form of equations and software were created, implemented and assessed. If useful, the improvement was adopted. These adopted improvements are briefly discussed in the following paragraphs.

Mathematical theory predicts that the vectors obtained from the Krylov sequence, after orthogonalization with the previous two vectors in the sequence, are independent. However, in practical applications these vectors usually converge, and sometimes quickly, to a nearly dependent state. Using totally dependent or nearly dependent vectors, the Gram-Schmidt orthogonalization algorithm fails. As a result, a set of vectors sufficient to define the dynamics of the component may not be obtainable. A solution to this problem has been available, consisting of repeated

Gram-Schmidt orthogonalization with all previous vectors, whenever that procedure initially fails. This simple solution is usually successful. However, successive re-orthogonalization can be computationally expensive and there is no guarantee that the resulting vectors will have any physical significance, or that this process will not also eventually fail<sup>22</sup>. Several alternate re-orthogonalization procedures have been investigated, evaluated, and discussed in this work.

In addition to the investigation of Gram-Schmidt re-orthogonalization procedures, an alternative procedure for orthogonalization in the block-Krylov sequence has been presented. Rather than the Gram-Schmidt vector by vector numerical scaling approach, a matrix transformation which orthogonalizes the vectors simultaneously can be created, without calculating eigenvectors. This matrix transformation utilizes the transpose of the Cholesky factor of a matrix product of the original dependent set of vectors. This process has been shown to be more efficient than the Gram-Schmidt orthogonalization procedure in this application. Cholesky orthogonalization has been integrated into the boundary flexibility method of component synthesis using generalized static Ritz vectors.

Another enhancement in the boundary flexibility methodology was possible because of the initiation of the Krylov sequence with the boundary flexibility modes. Recall that the boundary flexibility modes are multiplied by the mass matrix to create a force matrix. The size of this force matrix is the same as the size of the component interface, which subsequently determines the block size in the Krylov sequence.

When working with finite element models which have simple interfaces the resulting force matrix and solutions have physical significance and the blocks are convenient to process. However, in typical aerospace structures where a more complicated interface and a large number of interface nodes exists, the Krylov block size will be very large. This can lead to several complications in convergence and truncation. The problem of large block size has been solved in this work by discarding nearly dependent vectors from the block, previous to orthogonalization. Vectors which are nearly dependent are, geometrically, nearly identical to each other. As a result, no particularly useful information is being lost when they are removed from the block. This reduction also reduces the size of subsequent blocks. This method has been identified in this work as block filtering.

A problem which all static Ritz vector, Krylov sequence solutions have is the lack of a sound mathematical basis for judging when to terminate the sequence. The sequence should be terminated when the resulting set of Ritz vectors is sufficient for dynamic representation of the component. When using normal eigenvectors to represent a component, modal truncation, based upon an eigenvalue cutoff, is the most popular basis for judging if the dynamic representation is sufficient. Static Ritz vectors do not have an eigenvalue with which to associate a truncation limit and so an alternate method of the termination of the Krylov sequence was required and developed. This method is an heuristic methodology based upon the density of the

modal space in the component and the demonstrated observation that the most dynamically significant static Ritz are generated early in the Krylov sequence.

While investigating the Lanczos eigenvalue extraction method, done so because of its use of the Krylov sequence and its resulting similarity with static Ritz vectors, the tool of inverse operators was recognized. The use of inverse operators is commonly called sequence shifting. By shifting the sequence, an eigenvalue extraction routine can locate missing eigenvalues within a specific frequency range. The use made of shifting in this work differs from that of eigenvalue extraction in that, instead of targeting ranges of missing eigenvalues, the shift is targeted to the frequency of the applied forcing function. In this manner, vectors which can contribute to an accurate dynamic response prediction are generated. The use of shifting in this manner has been identified in this work as targeted shifting.

Of the various enhancements discussed in the preceding paragraphs, only block filtering is specifically tied to component mode synthesis and the boundary flexibility method. All the other improvements, such as targeted shifting and modal density series truncation, can be used in the general static Ritz vector methodology. Even so, in this work these improvements have primarily been implemented and evaluated in the context of the boundary flexibility method. Several general math proofs, which are also not specific to the boundary flexibility method, have also been derived and presented.

## Chapter 2

### Literature Review

#### 2.1) Introduction:

Wilson, Yuan, and Dickens<sup>29</sup> originally proposed the use of Ritz vectors, based upon external loading, for structural dynamic analysis. This formulation reduced an entire structure, not a component. The algorithm begins with a set of externally applied loads. The displacements from the static solution to the applied loads become the initial Ritz vector. That vector is then multiplied by the mass matrix to become the next force vector. This sequence is repeated to form a recurrence relationship and a series of special Ritz vectors, which are referred to in this work as static Ritz vectors. The proposed recurrence relationship is used in the papers discussed below and throughout this work.

Wilson's methodology was applied, using MSC/NASTRAN, to several large finite element models by Arnold, Citerley, Chargin, and Galant<sup>3</sup>. It was found that Wilson's methodology was computationally more efficient than the standard normal modes procedure. A recent application using a simple model of the space station was presented by Escobedo-Torres and Ricles<sup>12</sup>. This work compared the predicted transient response of the space station due to a docking force using load dependent Ritz vectors with predictions using eigenvectors.

Nour-Omid and Clough<sup>19</sup> investigated Wilson, et al's methodology and found that the proposed recurrence relationship actually generated a Krylov sequence. A Krylov subspace of order  $j$  is a vector space defined by

$$[\phi, A\phi, A^2\phi, \dots, A^{j-1}\phi] \quad (2.1)$$

where  $\phi$  is a column vector and  $A$  is a square, symmetric matrix. If  $A$  is  $n \times n$ , nonsingular, and if  $j = n$ , the Krylov vectors span the  $n$  dimensional space<sup>10</sup>, and an exact solution can be produced. In structural dynamics, the Krylov subspace can be defined by the following Krylov sequence,

$$[r, K^{-1}Mr, (K^{-1}M)^2r, \dots, (K^{-1}M)^{j-1}r] \quad (2.2)$$

where  $K$  is the stiffness matrix,  $M$  is the mass matrix, and  $r$  is a starting vector (or in block-Krylov, a set of vectors). The matrix product  $K^{-1}M$  is not symmetric and its use in the Krylov sequence yields subtle theoretical and practical differences which will be discussed below. The Krylov sequence is also the basis of the Lanczos eigenvalue extraction algorithm, and Nour-Omid and Clough refer to Wilson's static Ritz vectors as Lanczos coordinates. The Lanczos eigenvalue extraction algorithm generates Krylov vectors, which are used to transform the system into a tridiagonal form<sup>15,22</sup>. To extract the eigenvalues, this tridiagonal matrix is diagonalized using a QR, or related, algorithm. Nour-Omid and Clough utilize the tridiagonal system matrices directly to solve the dynamic response problem.

Nour-Omid and Clough extended their work to a more general dynamic loading represented by a linear combination of load vectors<sup>20</sup>, such as time dependent loading. The result was a structural dynamics application of the block-Krylov, or block-Lanczos method for the dynamic analysis of structures. A block is defined as the combination of a set of vectors, or modes, into matrices. They also presented the requirement of using the Gram-Schmidt procedure to orthogonalize the vectors within the Krylov block.

The use of static Ritz vectors was shown to be applicable to component mode synthesis by Wilson and Bayo<sup>28</sup>. The static Ritz vectors calculated were based, once again, upon an external load. Only a formulation for components with fixed interface boundary conditions was presented. This work was also implemented and applied by Léger<sup>16</sup> to an example small beam.

A similar development of static Ritz vectors in component mode synthesis was presented by Allen<sup>2</sup>. This paper provided the basis for the application work performed by Brunty<sup>5</sup>. The transient response of the Space Shuttle vehicle, during liftoff, was calculated using load-dependent static Ritz vectors and compared to the response predicted using eigenvectors in the component mode synthesis. Similar answers were obtained using less computer time.

Abdallah and Huckelbridge<sup>1</sup>, and independently, Craig and Hale<sup>10</sup>, demonstrated a generalized methodology applicable to components with fixed or free interfaces, with or without rigid body modes, and with or without applied loading.

Components having no applied external loading were formed using the boundary flexibility matrix, multiplied by the mass matrix, to form a force matrix. This force matrix produces a set of static Ritz vectors. (Craig and Hale refer to these vectors as Krylov vectors). The boundary flexibility matrix is defined as either the constraint modes or the attachment modes, depending on whether fixed or free interface conditions are selected. The methodology contained in these two papers is reviewed in the next three parts of this section. Abdallah and Huckelbridge also quantified the advantages, in computational effort, that generalized static Ritz vectors have over normal eigenvectors. Carney, Abdallah, and Huckelbridge implemented this methodology in MSC/NASTRAN<sup>6</sup>.

Yiu and Landess<sup>30</sup> also developed a similar methodology for forming a component which does not have an external applied load. However, their formulation is applicable to fixed interface components only. A criteria for concluding the recurrence sequence, based upon the rigid body mass and flexibility represented by the calculated static Ritz vectors, was proposed.

Some applications of static Ritz (referred to in that article as Krylov) vectors, including unsymmetric, damped structural dynamics systems may be found in the work of Craig, Su, and Kim<sup>11</sup>. The focus of this effort is on the control of the flexible structure represented by the Krylov vectors.

Both the Lanczos eigenvalue extraction algorithm and static Ritz vectors calculated using the boundary flexibility method are based upon the Krylov recurrence

sequence. As a result, some of the information and experience available in the published literature concerning Lanczos eigenvalue extraction has relevance to static Ritz vectors. Amongst the wide amount of available literature, the most complete and up to date source of information concerning the Krylov sequence and Lanczos eigenvalue extraction is Parlett's book, *The Symmetric Eigenvalue Problem*<sup>22</sup>. This book also includes information on other eigenvalue extraction algorithms, as well as discussions on general linear algebra tools which are particularly useful in this type of analysis.

In addition, two other excellent sources of information concerning the Lanczos eigenvalue extraction algorithm are the reports, *A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems*<sup>13</sup>, by Grimes, Lewis and Simon of Boeing Computer Services, and the *MSC/NASTRAN Handbook for Numerical Methods*<sup>31</sup>. These reports have a useful emphasis on the practical implementation of the Lanczos algorithm. The same implementation of the shifted block-Lanczos algorithm is presented in both of these documents. The first commercial implementation of Lanczos eigenvalue extraction was accomplished by Boeing Computer Services. MSC subsequently obtained this code from Boeing and implemented it in MSC/NASTRAN, and have since made a number of modifications and improvements to the algorithm.

## 2.2) Review of Boundary Flexibility Component Mode Synthesis:

Almost all of the literature published on the variously titled, Krylov, Lanczos, or static Ritz, vectors focuses on the use of these vectors at the system level. These works deal with either Lanczos eigenvalue extraction, control dynamics, or external load derived Ritz vectors. The emphasis of this dissertation is the use of static Ritz vectors in component mode synthesis. As a result, a detailed review of the small amount of literature describing the static Ritz vector boundary flexibility method of component synthesis, which is the starting point of this work, is warranted.

### 2.2.1) Fixed Interface Methodology:

First, as is standard in component mode synthesis methods, the finite element component mass,  $m$ , and stiffness,  $k$ , matrices are partitioned into internal and external degrees of freedom, denoted by subscripts  $i$  and  $c$ , respectively.

$$m = \begin{bmatrix} m_{cc} & m_{ci} \\ m_{ic} & m_{ii} \end{bmatrix} \quad (2.3)$$

$$k = \begin{bmatrix} k_{cc} & k_{ci} \\ k_{ic} & k_{ii} \end{bmatrix} \quad (2.4)$$

The constraint modes are defined by

$$\Phi_{ic} = -k_{ii}^{-1} k_{ic} \quad (2.5)$$

which is the same definition used in standard component mode synthesis.

For Wilson's method<sup>29</sup>, a set of externally applied loads is required to obtain the initial set of Ritz vectors. For the boundary flexibility method, this set of loads is created by multiplying the constraint modes by the mass matrix. (Craig<sup>10</sup> also included the off-diagonal mass matrix in his formulation.) Since the mass matrix is used to create the loads, they can be considered inertia loads. This set of inertia loads are then used to generate the initial set, or block, of Ritz vectors using the following

$$\mathbf{q}_1^{**} = \mathbf{k}_{ii}^{-1} (\mathbf{m}_{ii} \Phi_{ic} + \mathbf{m}_{ic}) \quad (2.6)$$

where the superscript \*\* indicates that the vectors in the matrix have not been normalized. The first block of vectors is normalized using the following equation, where the subscript  $r$  signifies that the block is normalized vector by vector. There are  $c$  vectors within each matrix, or block.

$$\mathbf{q}_{1,r} = \frac{\mathbf{q}_{1,r}^{**}}{\sqrt{(\mathbf{q}_{1,r}^{**T} \mathbf{m}_{ii} \mathbf{q}_{1,r}^{**})}} \quad r = 1, 2, \dots, c \quad (2.7)$$

Note that the denominator of the right hand side of the equation is merely the inner product norm,  $\|\mathbf{q}\|_m$ , calculated with respect to the mass matrix.

The subsequent sets of static Ritz vectors in the Krylov sequence are generated using the recurrence relationship which was defined in equation (2.2)<sup>19,29</sup>,

$$\mathbf{q}_j^* = \mathbf{k}_{ii}^{-1} \mathbf{m}_{ii} \mathbf{q}_{j-1} \quad (2.8)$$

where the superscript \* signifies that the vectors have not been orthogonalized or normalized. The additional sets of vectors are orthogonalized, with respect to the mass matrix, with all previous vectors. The process used to perform this orthogonalization is a normalized Gram-Schmidt procedure.

$$\mathbf{q}_j^{**} = \mathbf{q}_j^* - \mathbf{q}_{1,j-1} \mathbf{c} \quad (2.9)$$

where

$$\mathbf{c} = \mathbf{q}_{1,j-1}^T \mathbf{m}_{ii} \mathbf{q}_j^* \quad (2.10)$$

and  $\mathbf{q}_{1,j-1}$  is the concatenation of the previous sets of Ritz vectors,

$$\mathbf{q}_{1,j-1} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{j-1}] \quad (2.11)$$

where all vectors have been normalized as follows.

$$\mathbf{q}_{j_r} = \frac{\mathbf{q}_{j_r}^{**}}{\sqrt{(\mathbf{q}_{j_r}^{**T} \mathbf{m}_{ii} \mathbf{q}_{j_r}^{**})}} \quad r = 1, 2, \dots, c \quad (2.12)$$

The complete set of calculated Ritz vectors is included in the transformation matrix as  $\mathbf{Q}_l$ . (The resulting transformation matrix has the same form as that of "Craig-Bampton" component mode synthesis<sup>9</sup>, with the Ritz vectors replacing the normal modes.)

$$\mathbf{\Psi} = \begin{bmatrix} \mathbf{I}_{cc} & \mathbf{0} \\ \mathbf{\Phi}_{ic} & \mathbf{Q}_l \end{bmatrix} \quad (2.13)$$

The physical mass and stiffness matrices are transformed into the component modal matrices

$$\boldsymbol{\mu} = \boldsymbol{\Psi}^T \mathbf{m} \boldsymbol{\Psi} \quad (2.14)$$

$$\boldsymbol{\kappa} = \boldsymbol{\Psi}^T \mathbf{k} \boldsymbol{\Psi} \quad (2.15)$$

The resulting mass submatrices are

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_{cc} & \boldsymbol{\mu}_{cl} \\ \boldsymbol{\mu}_{lc} & \boldsymbol{\mu}_{ll} \end{bmatrix} \quad (2.16)$$

where

$$\begin{aligned} \boldsymbol{\mu}_{cc} &= \boldsymbol{\Phi}_{ic}^T (\mathbf{m}_{ii} \boldsymbol{\Phi}_{ic} + \mathbf{m}_{ic}) + \mathbf{m}_{ci} \boldsymbol{\Phi}_{ic} + \mathbf{m}_{cc} \\ \boldsymbol{\mu}_{lc} &= \boldsymbol{\mu}_{cl}^T = \mathbf{Q}_l^T (\mathbf{m}_{ii} \boldsymbol{\Phi}_{ic} + \mathbf{m}_{ic}) \\ \boldsymbol{\mu}_{ll} &= \mathbf{I}_l = \mathbf{Q}_l^T \mathbf{m}_{ii} \mathbf{Q}_l \end{aligned} \quad (2.17)$$

The resulting stiffness submatrices are

$$\boldsymbol{\kappa} = \begin{bmatrix} \boldsymbol{\kappa}_{cc} & \boldsymbol{\kappa}_{cl} \\ \boldsymbol{\kappa}_{lc} & \boldsymbol{\kappa}_{ll} \end{bmatrix} \quad (2.18)$$

where

$$\begin{aligned} \boldsymbol{\kappa}_{cc} &= \mathbf{k}_{ci} \boldsymbol{\Phi}_{ic} + \mathbf{k}_{cc} \\ \boldsymbol{\kappa}_{lc} &= \boldsymbol{\kappa}_{cl}^T = \mathbf{0} \\ \boldsymbol{\kappa}_{ll} &= \mathbf{Q}_l^T \mathbf{k}_{ii} \mathbf{Q}_l \end{aligned} \quad (2.19)$$

The use of constraint modes in the transformation matrix leads to the null off-diagonal partitions of the component stiffness matrix, just as in the approach based upon normal modes.

### 2.2.2) Free Interface Methodology for Components with No Rigid Body Modes:

When allowing the interface points of a component to be free to deflect while forming the component, a somewhat different basis for the initial vector of the Krylov sequence is required. The attachment modes, rather than the constraint modes are utilized in initial block definition. By definition, the attachment modes are the columns of the flexibility matrix which correspond to the interface degrees of freedom.

$$\mathbf{g} = \mathbf{k}^{-1} \quad (2.20)$$

$$\mathbf{g}_a = \begin{bmatrix} \mathbf{g}_{cc} \\ \mathbf{g}_{ic} \end{bmatrix} \quad (2.21)$$

The initial block of vectors in the free interface formulation is defined as

$$\mathbf{q}_1^{**} = \mathbf{k}^{-1} \mathbf{m} \mathbf{g}_a \quad (2.22)$$

and is normalized as follows.

$$\mathbf{q}_{1,r} = \frac{\mathbf{q}_{1,r}^{**}}{\sqrt{(\mathbf{q}_{1,r}^{**T} \mathbf{m} \mathbf{q}_{1,r}^{**})}} \quad r = 1, 2, \dots, c \quad (2.23)$$

Note that the unpartitioned physical mass and stiffness matrices of the component are used in the free interface formulation. The recurrence algorithm then proceeds in the same manner as in the fixed interface methodology.

$$\mathbf{q}_j^* = \mathbf{k}^{-1} \mathbf{m} \mathbf{q}_{j-1} \quad (2.24)$$

$$\mathbf{q}_j^{**} = \mathbf{q}_j^* - \mathbf{q}_{1,j-1} \mathbf{c} \quad (2.25)$$

$$\mathbf{c} = \mathbf{q}_{1,j-1}^T \mathbf{m} \mathbf{q}_j^* \quad (2.26)$$

$$\mathbf{q}_{j,r} = \frac{\mathbf{q}_{j,r}^{**}}{\sqrt{(\mathbf{q}_{j,r}^{**T} \mathbf{m} \mathbf{q}_{j,r}^{**})}} \quad r = 1, 2, \dots, c \quad (2.27)$$

Formation of a static Ritz vector component then follows the normal component mode synthesis techniques which were presented by MacNeal<sup>17</sup> and Rubin<sup>24</sup>. To combine the "Rubin-MacNeal" method with the presented method, the normal eigenvectors are simply replaced with the static Ritz vectors, as in the fixed interface methodology. The free interface methodology uses residual flexibility terms, which fully define the stiffness missing from the modal space due to excluded modes. The flexibility contained in the calculated Ritz vectors is given by the following equation.

$$\mathbf{g}_k = \mathbf{Q}_l (\mathbf{Q}_l^T \mathbf{k} \mathbf{Q}_l)^{-1} \mathbf{Q}_l^T \quad (2.28)$$

The unrepresented flexibility, or residual flexibility, is defined as

$$\mathbf{g}_d = \mathbf{g} - \mathbf{g}_k \quad (2.29)$$

The residual flexibility matrix is then partitioned in the same manner as the flexibility matrix was in equation (2.21), when the attachment modes were created for initial vector calculation. The result is the residual attachment modes.

$$\mathbf{g}_{a_d} = \begin{bmatrix} \mathbf{g}_{cc_d} \\ \mathbf{g}_{ic_d} \end{bmatrix} \quad (2.30)$$

When the residual attachment modes,  $\mathbf{g}_{a_d}$ , are added to the Ritz vectors,  $\mathbf{Q}_l$ , the complete flexibility of the component is represented.

The residual attachment modes and the Ritz vectors are used to form the component transformation matrix. This matrix transforms the physical subspace,  $\mathbf{u}$ , to the modal subspace,  $\mathbf{p}$ , and is defined by the following equation.

$$\begin{bmatrix} \mathbf{u}_c \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{g}_{cc_d} & \mathbf{Q}_{cl} \\ \mathbf{g}_{ic_d} & \mathbf{Q}_{il} \end{bmatrix} \begin{bmatrix} \mathbf{p}_c \\ \mathbf{p}_l \end{bmatrix} \quad (2.31)$$

In order to provide physical interface degrees of freedom, for use in component coupling,  $\mathbf{p}_c$  in the above equation is back-transformed to eliminate it from the right-hand side of the equation. This results in the following transformation matrix,

$$\begin{bmatrix} \mathbf{u}_c \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{cc} & \mathbf{0} \\ \mathbf{g}_{ic}^* & \mathbf{Q}_{il}^* \end{bmatrix} \begin{bmatrix} \mathbf{u}_c \\ \mathbf{p}_l \end{bmatrix} \quad (2.32)$$

where

$$\begin{aligned} \mathbf{g}_{ic}^* &= \mathbf{g}_{ic_d} \mathbf{g}_{cc_d}^{-1} \\ \mathbf{Q}_{il}^* &= \mathbf{Q}_{il} - \mathbf{g}_{ic_d} \mathbf{g}_{cc_d}^{-1} \mathbf{Q}_{cl} \end{aligned} \quad (2.33)$$

The transformation of the component mass and stiffness matrices then proceeds in a similar manner as shown in equations (2.14) through (2.19), with the following differences. The  $\mathbf{Q}_{il}^*$  matrix partition replaces the  $\mathbf{Q}_l$  matrix partition. The  $\mathbf{g}_{ic}^*$  matrix partition replaces the  $\Phi_{ic}$  matrix partition. In the fixed-interface methodology, the definition of the constraint modes,  $\Phi_{ic}$ , leads to terms in the component stiffness matrix which cancel out. In the free-interface methodology, the definition of the transformation submatrices has changed and so this cancellation does not occur. Therefore, the corresponding equations in (2.19) are replaced by the following equations, respectively.

$$\begin{aligned} \kappa_{cc} &= \mathbf{g}_{ic}^{*T} (\mathbf{k}_{ii} \mathbf{g}_{ic}^* + \mathbf{k}_{ic}) + \mathbf{k}_{ci} \mathbf{g}_{ic}^* + \mathbf{k}_{cc} \\ \kappa_{jc} &= \kappa_{cj}^T = \mathbf{Q}_{il}^{*T} (\mathbf{k}_{ii} \mathbf{g}_{ic}^* + \mathbf{k}_{ic}) \end{aligned} \quad (2.34)$$

### 2.2.3) Free Interface Methodology for Components with Rigid Body Modes:

When a component has rigid body modes, the associated stiffness matrix is singular. The inverse of the stiffness matrix, the flexibility matrix, cannot be directly obtained, and therefore the attachment modes cannot be directly obtained. To circumvent this problem, Rubin<sup>24</sup> presented the following method for obtaining the residual elastic attachment modes of a component with rigid body modes.

First, the stiffness matrix is constrained from rigid body motion by partitioning out  $r$  degrees of freedom, where  $r$  is the number of rigid body modes.

$$\mathbf{k} = \begin{bmatrix} \mathbf{k}_{ww} & \mathbf{k}_{wr} \\ \mathbf{k}_{rw} & \mathbf{k}_{rr} \end{bmatrix} \quad (2.35)$$

The remaining partition is then inverted.

$$\mathbf{g}_{ww} = \mathbf{k}_{ww}^{-1} \quad (2.36)$$

This flexibility matrix is then expanded back to  $n$  ( $w + r$ ) size.

$$\mathbf{g}_c = \begin{bmatrix} \mathbf{g}_{ww} & \mathbf{0}_{wr} \\ \mathbf{0}_{rw} & \mathbf{0}_{rr} \end{bmatrix} \quad (2.37)$$

A square projection matrix is defined by

$$\mathbf{A} = \mathbf{I}_{nn} - \mathbf{m} \Phi_r \Phi_r^T \quad (2.38)$$

where  $\Phi_r$  is the rigid body modes matrix. The elastic flexibility matrix,  $\mathbf{g}_e$ , with rigid body motion removed, is shown in reference [24] to be

$$\mathbf{g}_e = \mathbf{A}^T \mathbf{g}_c \mathbf{A} \quad (2.39)$$

Now the analysis proceeds in a similar fashion to the previously discussed methodology of the free interface component with no rigid body motion. The major

difference between the two approaches is that the elastic flexibility matrix is used in place of the general flexibility matrix. The inertia relief attachment modes are

$$\mathbf{g}_{a_e} = \begin{bmatrix} \mathbf{g}_{cc_e} \\ \mathbf{g}_{ic_e} \end{bmatrix} \quad (2.40)$$

The initial block of vectors is calculated using the inertia relief attachment modes and the elastic flexibility matrix.

$$\mathbf{q}_1^{**} = \mathbf{g}_e \mathbf{m} \mathbf{g}_{a_e} \quad (2.41)$$

The subsequent static Ritz vectors are calculated, orthogonalized, and normalized as shown in equations (2.24) through (2.27). The residual elastic flexibility terms are also calculated as shown in the free interface with no rigid body modes discussion, equations (2.28) through (2.30). Creation of the transformation matrix, equations (2.31) through (2.34), is also similar to when no rigid body modes are present. The one exception is that the rigid body modes must be included in the transformation matrix. Therefore, equation (2.31) is replaced by

$$\begin{bmatrix} \mathbf{u}_c \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{g}_{cc_d} & \mathbf{Q}_{cl} & \Phi_{cr} \\ \mathbf{g}_{ic_d} & \mathbf{Q}_{il} & \Phi_{ir} \end{bmatrix} \begin{bmatrix} \mathbf{p}_c \\ \mathbf{p}_l \\ \mathbf{p}_r \end{bmatrix} \quad (2.42)$$

Formation of the final transformation matrix, and subsequently the component mass and stiffness matrices, is then performed as described in the previous section.

### 2.3) Relationship to Lanczos Eigenvalue Extraction:

In addition to being the result of the fundamental recurrence equation when calculating static Ritz vectors, a Krylov sequence is also the result of the fundamental recurrence relation used in the Lanczos eigenvalue extraction algorithm<sup>15,22</sup>. The Lanczos eigenvalue extraction algorithm uses a Krylov sequence to generate terms which can be used to transform the given system into a tridiagonal form. This tridiagonal matrix is then diagonalized using a QR, or related, eigenvalue extraction algorithm and the resulting diagonal terms are the eigenvalues of the original system. A detailed presentation of the Lanczos algorithm is beyond the scope of this work. Since Krylov vectors are the foundation of the Lanczos eigenvalue extraction algorithm, some of the existing literature which investigates orthogonalization, shifting, and practical implementation of the Lanczos algorithm is applicable to static Ritz vectors, since they are also derived from the Krylov sequence.

There are also significant differences between the Lanczos eigenvalue extraction algorithm<sup>15,22</sup> and the use of static Ritz vectors directly. These differences are summarized as follows. In the selection of the initial seed to start the Krylov sequence, Lanczos starts with a random vector. This is done to prevent the Krylov sequence from converging to a particular class of eigenvectors and skipping other eigenvectors entirely. Since another transformation will take place converting the tridiagonal matrix into the system eigenvalues, the vectors from the sequence need not have a particular physical significance. Wilson's methodology initiates the sequence

with a matrix which has physical significance, i.e., a load vector. In the boundary flexibility method that load vector is the constraint or attachment modes multiplied by the component mass matrix, forming an inertia load vector block. That leads the vectors which result from subsequent iterations of the sequence to tend to also have physical significance. This difference alters the Krylov sequence from being a purely mathematical tool for eigenvalue extraction, into a mechanically based application in structural dynamics.

These differences between the two methodologies stem from the differing goals of the two algorithms. As discussed above, the Lanczos algorithm searches for the eigensystem by first transforming the original system matrices into tridiagonal form, and then diagonalizing that tridiagonal form to obtain the eigenvalues. Some applications of Wilson's methodology make use of the tridiagonal form, but the orthonormalized vectors obtained from the sequence are instead used to transform and reduce the original system directly. The eigenvalues and eigenvectors are not obtained, and both transformed mass and stiffness matrices do not assume a diagonal form. In all component mode synthesis the reduced matrices are not diagonal anyway, due to the coupling mass and stiffness partitions of the final reduced matrices (equations 2.17 and 2.34). Therefore, the non-diagonal form obtained from the use of static ritz vectors is not a disadvantage when used in component mode synthesis. In addition, the vector space in which a transformed system is tridiagonal is not the

same as in the component mode synthesis algorithm, and, as a result, is not useful as a final goal.

Vector blocks are also utilized in the practical implementation of the Lanczos eigenvalue extraction algorithms<sup>13,22,31</sup>. The original Lanczos algorithm had difficulty in determining a complete set of multiple eigenvalues. Using vector blocks in the Krylov sequence allows the algorithm to determine multiple roots, up to the dimension of the block. When blocks are used in the Krylov sequence, the tridiagonal form of the transformed system also assumes a blocked format. The bandwidth of the tridiagonal form is then determined by the dimension of the block. The block format complicates orthogonalization and sequence truncation schemes. The block format is a natural feature in the boundary flexibility method because of the multidimensional inertia load vector block.

#### 2.4) Orthogonalization:

The most computationally expensive aspect of the formulation of static Ritz vectors is the process of orthogonalization (equations (2.9) through (2.12)). As a result, efficient orthogonalization is critical in determining this method's efficiency when compared to the use of eigenvectors. Since Ritz vectors are not inherently independent, for them to be used in a similarity transformation, they must be orthogonalized. Obviously if this process is more computationally expensive than the calculation of eigenvectors, then it's usefulness is limited.

#### 2.4.1) Loss of Orthogonality:

It has been shown in several works<sup>15,19,22</sup>, that if a Ritz vector obtained from the Krylov sequence is orthogonalized, using Gram-Schmidt, with the two previous Krylov vectors, it is theoretically orthogonal to all previously generated vectors. The orthonormalizing coefficients are assembled into a tridiagonal matrix. The assembled tridiagonal matrix is also the result of the Krylov coordinate transformation on the original system matrix ( $Q^T A Q$ ). In other words, use of the Krylov sequence and the properties of orthogonality, allow the system to also assume a tridiagonal form. Unfortunately, in practice the theoretical orthogonality that each new Krylov vector has with all vectors, after orthogonalization with the previous two, is usually lost due to either even minimal computational round-off error, or other factors which will be discussed in chapter 3. A mathematically rigorous explanation of this phenomena is given in references [21],[22] and [26]. As a result, additional orthogonalization and sometimes re-orthogonalization is required in order to perform a correct transformation and to maintain the tridiagonal form.

A brief clarification of terms found in the existing literature and used in this work follows. When Gram-Schmidt procedures are required, they are sometimes referred to as orthogonalization and sometimes as re-orthogonalization, varying with author. In this work, orthogonalization refers to the initial Gram-Schmidt process, even if performed with all previous vectors. Re-orthogonalization refers to any additional orthogonalization steps following the initial orthogonalization.

The initial solution to the loss of orthogonality, which occurs in the Krylov sequence, was to simply explicitly orthogonalize with all previous vectors, rather than just the previous two. This orthogonalization scheme is computationally more expensive than orthogonalizing with the previous two vectors, especially as the number of previously determined vectors becomes large. The computational expense of orthogonalization is what limited Lanczos eigenvalue extraction use for many years. Many enhancements to the basic Lanczos eigenvalue extraction algorithm have appeared in the literature through the years, some of which were directed toward insuring orthogonality. Paige<sup>21</sup> established a theorem, using matrix norms and terms from the tridiagonal matrix, which yields a numerical criterion for determining when re-orthogonalization is required. The orthogonality of the obtained vectors is not explicitly calculated. When the requirement for orthogonalization does arise, the new vector would be orthogonalized with respect to all previous vectors. Parlett and Scott<sup>23</sup> also use a numerical criterion, derived using matrix norms and terms from the tridiagonal matrix, to determine when re-orthogonalization is required. They proposed the simple modification of orthogonalizing, using modified Gram-Schmidt<sup>4</sup>, with only those previous vectors with which the new vector is not orthogonal. Simon<sup>26</sup> clarified issues dealing with the loss of orthogonality, developed an additional re-orthogonalization scheme, and investigated the complete orthogonalization issue in depth.

### 2.4.2 Orthonormalization:

In general, the existing literature describes the Lanczos eigenvalue algorithm as extracting the eigenvalues of a single symmetric matrix,  $\mathbf{A}$ .

$$\mathbf{A} \mathbf{x} = \lambda \mathbf{x} \quad (2.43)$$

The vectors obtained from the Krylov sequence are orthonormalized using the Euclidean vector norm, which is defined as  $\|\mathbf{x}\|_2 = (|x_1|^2 + |x_2|^2 + \dots + |x_p|^2)^{1/2}$ . Because of the spectral theorem<sup>22</sup>, the eigenvectors are also orthonormal with respect to the  $\mathbf{A}$  matrix. However, as discussed above, in structural dynamics, the eigenproblem which is being solved is a system with both mass and stiffness matrices.

$$\mathbf{K} \mathbf{x} = \lambda \mathbf{M} \mathbf{x} \quad (2.44)$$

In the MSC/NASTRAN application of the Lanczos eigenvalue extraction algorithm, the mass and stiffness matrices are used directly in the Krylov sequence and the Krylov vectors are orthonormalized with respect to the mass matrix<sup>31</sup>.

In the use of static Ritz vectors for structural dynamics applications, as described in detail in section 2.2, the mass and stiffness are used in the Krylov sequence and the vectors are orthonormalized with respect to the mass matrix. There is one exception to the use of the mass matrix. In the work of Su and Craig<sup>27</sup>, the static Ritz (referred to as Krylov) vectors are orthonormalized with respect to the stiffness matrix. The result of orthonormalizing with respect to the mass matrix will

be that the modal partition of the reduced system mass matrix is the identity matrix. One can orthonormalize with respect to the mass matrix, the stiffness matrix, or using the Euclidean vector norm, but not all three, with a single Gram-Schmidt orthogonalization procedure. Orthonormalizing with respect to mass (or stiffness) allows some flexibility in dealing with the reduced mass matrix in some applications and is required to generate the tridiagonal form when using the Krylov sequence of equation (2.2). This is a direct result of the matrix product,  $K^{-1}M$ , not being symmetric. As previously discussed, in component mode synthesis the resulting transformed mass and stiffness matrices are, by definition, not diagonal, and exist in a different space than that of the tridiagonal form. Therefore, the matrix form of the transformed component matrices is not an important issue, and it is not necessary to maintain the ability to generate the tridiagonal form.

#### 2.4.3) Cholesky/QR Decomposition:

Parlett<sup>22</sup> presents the following discussion, relating to orthogonalization. Any non-null rectangular  $m$  by  $n$  matrix  $B$  can be written as  $B = QR$  with  $m$  by  $r$   $Q$  satisfying  $Q^T Q = I_r$ , and  $r$  by  $n$   $R$  upper triangular with non-negative diagonal elements. The  $QR$  factorization is the matrix formulation of the Gram-Schmidt process for orthonormalizing the columns of  $B$ . When  $B$  has full rank, then  $R$  is the upper Cholesky factor of  $B^T B$  since

$$R^T R = R^T Q^T Q R = B^T B \quad (2.45)$$

Use of the upper Cholesky factor of a matrix for orthogonalization will be presented in section 3.3.5.

### 2.5) Shifting:

A technique defined by Scott<sup>25</sup> shows how an inverse operator, either applied explicitly while using subspace iteration or applied implicitly by while using the Lanczos algorithm, can direct a solution to particular frequency range. The use of the inverse operator in the Lanczos eigenvalue extraction algorithm is commonly called shifting. Shifting has been implemented with success in the commercially available Lanczos eigenvalue extraction routine<sup>13,31</sup>. The inverse operator is applied to the Krylov sequence within the Lanczos algorithm, and therefore can be applicable to any Krylov sequence based solution.

In the Scott paper, the problem of computing some eigenpairs of the generalized eigenvalue problem is considered,

$$(\mathbf{A} - \lambda \mathbf{M}) \mathbf{x} = 0 \quad (2.46)$$

with  $\lambda$  being the eigenvalues and  $\mathbf{x}$  the eigenvectors of the pencil  $(\mathbf{A}, \mathbf{M})$ . A system is created using the operator

$$(\mathbf{A} - \sigma \mathbf{M})^{-1} \mathbf{M}$$

that has the same eigenvectors as (2.46). The shifted system eigenvalues are transformed to  $1/(\lambda - \sigma)$ . This means that the eigenvalue nearest  $\sigma$  becomes the

dominant eigenvalue and the sequence will converge to the corresponding eigenvector. In the commercially available Lanczos routines the algorithm is applied to the shifted and inverted eigenvalue problem of the following equation

$$\mathbf{M}(\mathbf{A} - \sigma \mathbf{M})^{-1} \mathbf{M} \mathbf{x} = \frac{1}{\lambda - \sigma} \mathbf{M} \mathbf{x} \quad (2.47)$$

The use of the shifted eigenvalue form allows for good approximations to eigenvalues within specific ranges, even if they are clustered. The cost for having the advantage of shifting is the factorization of  $(\mathbf{A} - \sigma \mathbf{M})^{-1}$ .

### 2.6) Spectrum Slicing:

Parlett<sup>22</sup> presents the following theorem. When the triangular factorization of  $(\mathbf{A} - \sigma \mathbf{M})$  is calculated, if  $\mathbf{A}$  is symmetric then

$$(\mathbf{A} - \sigma \mathbf{M}) = \mathbf{L} \mathbf{D} \mathbf{L}^T \quad (2.48)$$

where  $\mathbf{D}$  is diagonal and  $\mathbf{M}$  is positive definite. Then

$$\nu(\Lambda - \sigma \mathbf{I}) = \nu(\mathbf{A} - \sigma \mathbf{M}) = \nu(\mathbf{D}) \quad (2.49)$$

where  $\nu$  is number of negative eigenvalues and  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ . The number of negative elements of  $\mathbf{D}$  equals the number of eigenvalues of the pencil  $(\mathbf{A}, \mathbf{M})$  which are less than  $\sigma$ . As a result, whenever a shift is undertaken the number of eigenvalues below the shift frequency is determined.

### 2.7) Krylov Sequence Termination Techniques:

A large uncertainty in the use of static Ritz vectors and the boundary flexibility method is the lack of a criteria for terminating the Krylov sequence. When eigenvectors are used to form a component, typically a frequency range is defined and all eigenvectors with eigenvalues within that given range are determined. That option does not exist in the use of static Ritz vectors. A number of error criterions have been proposed which truncate the sequence when a somewhat arbitrary variable reaches a arbitrary value.

Wilson, et. al.<sup>29</sup>, used a definition of the modal participation factor to define an error term. This factor is equivalent to the dependence coefficient in a Gram-Schmidt orthogonalization procedure. The error term is the linearly independent portion of the force vector, with respect to the Ritz vectors. The linearly independent portion of the force vector should be zero if the complete static solution is desired. No account is made for the dynamic response of the system in the error term. A similar error term was defined by Léger<sup>16</sup>.

Nour-Omid and Clough also used a modal participation factor as a sequence truncation criterion<sup>19,20</sup>. However, they did not define or use an error term based upon the desired dependence of the force vector. They proposed a simple cut off when the mode participation factor (or dependence coefficient) reached an arbitrary numerical value. No account is made for the dynamic response of the system.

Su and Craig<sup>27</sup> stated that a modal participation factor, such as used by Nour-Omid and Cough, was not valid for a non-diagonal representations, which are the result of the representations not being based upon normal modes. Therefore, they proposed that the norm of the off-diagonal submatrices in the transformed mass and stiffness matrices should be used as sequence truncation criterion. Again, the sequence would be terminated when the norm reached an arbitrary numerical value.

Yiu and Landess<sup>30</sup> proposed two similar sequence truncation criteria. The first used a flexibility convergence criteria, similar to the error term proposed by Wilson, et. al. A mass convergence criteria was also proposed. It was based upon the amount of the rigid body mass, rather than the force vector, which would be represented by Ritz vectors. The percentage of rigid body mass represented by Ritz vectors is commonly referred to as effective mass<sup>14</sup>. When the percentage of rigid body mass represented reaches an arbitrary numerical value the sequence is terminated.

## Chapter 3

### Theoretical Development

#### 3.1) Introduction:

This short chapter presents two theorems which pertain to the use of static Ritz vectors in the boundary flexibility method of component mode synthesis. These theorems concern whether or not static Ritz vectors can be used to represent a component in a mathematically rigorous fashion. Proofs are presented which demonstrate the fact that static Ritz vectors can be used to correctly represent a component.

#### 3.2) The Exact Nature of the Methodology:

Currently, most component mode synthesis applications use the normal eigenvectors of the substructure to form the component. If all the eigenvectors of a system are used to form the component, the complete dynamic properties of the component are represented and an "exact" finite element solution may be obtained. This is because, if all the modes are used, the component representation is not a Ritz vector approximation but is instead a complete linear coordinate transformation. The same principle holds true for components based upon a block-Krylov sequence. It was proven in references [10] and [27] that an  $n$  size Krylov subspace spans the entire  $n$  dimensional space. The following theorem is an alternative demonstration of the proposition that, if  $n$  Krylov vectors are used to form a component from a system of

size  $n$ , then the same complete solutions, as found directly from finite elements or normal modes, can be obtained.

Theorem 3.1 - Given that  $x$  is an eigenvector of  $A$ , and that  $\lambda$  is the corresponding eigenvalue, then for  $B$ , where  $B = P^T A P$ ,  $P^T x$  is the associated eigenvector of  $B$  and  $\lambda$  is the invariant eigenvalue of both  $B$  and  $A$ .

Proof - The eigensystem of  $A$  is defined as

$$A x = \lambda x \quad (3.1)$$

and since, from the definition of  $B$ ,

$$A = P B P^{-1} \quad (3.2)$$

equation (3.1) can be re-written as

$$P B P^{-1} x = \lambda x \quad (3.3)$$

Premultiplying by  $P^T$  yields

$$B P^{-1} x = \lambda P^{-1} x \quad (3.4)$$

If  $y$  is defined as  $P^T x$ , then

$$B y = \lambda y \quad (3.5)$$

and it is evident that the eigenvalue of this system is  $\lambda$ , and that the associated eigenvector is  $P^1x$ . ■

It is important to note that the only constraint on  $P$ , implicit or explicit, is that it is a nonsingular matrix, with the same dimension as  $A$ . For  $P$  to be nonsingular it must be of full rank, or equivalently stated, its columns must be linearly independent. Therefore, in order for an exact coordinate transformation to be accomplished, the vectors from which the transformation matrix is assembled must be linearly independent. These vectors do not need to be the eigenvectors, and they do not need to be orthogonal.

The fact that the vectors in a transformation matrix need only be linearly independent, and not orthogonal, is already utilized in component synthesis based upon normal eigenvectors. The transformation matrices which result from normal mode component synthesis consist of linearly independent vectors, not orthogonal ones. This can be demonstrated simply by inspecting the result of the matrix transformation, as shown in equations (2.14) through (2.19). The complete reduced mass and stiffness matrices are not diagonal. Only the modal partition of the matrices, which does result from an orthogonal transformation, is diagonal. Therefore, even when compared to normal eigenvectors, the use of static Ritz vectors contains no inherent disadvantages of matrix form or accuracy, since the resulting complete component mass and stiffness matrices in either case are not diagonal.

In the proof of theorem 3.1, the inverse of the transformation matrix was used in the pre-multiply position. Typically, as in equations (2.14) and (2.15), the transpose of the transformation matrix is used in this position. If a set of orthogonal vectors make up a transformation matrix, then the inverse and the transpose of the matrix are identical. If the vectors are only linearly independent, and not orthogonal, then the transpose and the inverse of the transformation matrix are not identical. However, in structural dynamics, this does not necessitate using the inverse of the transformation matrix. In structural dynamics, the eigensystem represents the space of the matrix multiplication  $M^{-1}K$ , which is derived from  $Kx = \lambda Mx$ . The following theorem shows that for the  $M^{-1}K$  space, use of the inverted transformation matrix and the transposed transformation matrix is interchangeable.

Theorem 3.2 - Specifying that  $P^T \neq P^{-1}$ , so that  $P$  is not orthonormal, and if  $M_a = P^T M P$  and  $K_a = P^T K P$ , which represent a transformation using the transpose, and  $M_b = P^{-1} M P$  and  $K_b = P^{-1} K P$ , which represents a transformation using the inverse, then  $M_a^{-1} K_a = M_b^{-1} K_b$ .

Proof - This may be proven by substitution. The proposition is that

$$M_a^{-1} K_a = M_b^{-1} K_b \quad (3.6)$$

and since, by definition,

$$\begin{aligned} M_a &= P^T M P & M_b &= P^{-1} M P \\ K_a &= P^T K P & K_b &= P^{-1} K P \end{aligned} \quad (3.7)$$

These equations may be substituted into equation (3.6), yielding

$$(\mathbf{P}^T \mathbf{M} \mathbf{P})^{-1} \mathbf{P}^T \mathbf{K} \mathbf{P} = (\mathbf{P}^{-1} \mathbf{M} \mathbf{P})^{-1} \mathbf{P}^{-1} \mathbf{K} \mathbf{P} \quad (3.8)$$

Post-multiplying by  $\mathbf{P}^T \mathbf{K}^{-1} \mathbf{P}$ , and canceling, results in

$$(\mathbf{P}^T \mathbf{M} \mathbf{P})^{-1} \mathbf{P}^T \mathbf{P} = (\mathbf{P}^{-1} \mathbf{M} \mathbf{P})^{-1} \quad (3.9)$$

and then post-multiplying by  $\mathbf{P}^T \mathbf{M} \mathbf{P}$  yields,

$$(\mathbf{P}^T \mathbf{M} \mathbf{P})^{-1} \mathbf{P}^T \mathbf{M} \mathbf{P} = \mathbf{I} \quad (3.10)$$

which reduces to

$$\mathbf{I} = \mathbf{I} \quad \blacksquare \quad (3.11)$$

In summary, it was proven in theorem 3.1 that orthogonal vectors are not required for an exact transformation, i.e., linearly independent vectors suffice. This proof used the inverse of the transformation matrix in the pre-multiply position. With a non-orthogonal transformation matrix, the transpose and the inverse of the transformation matrix are, by definition, not identical. As a result, a transformation which uses the transpose of a non-orthogonal matrix results in a mass and stiffness matrix different from the result obtained from a transformation using the inverse of the transformation matrix in the pre-multiply position. In theorem 3.2 it was shown that when the transformed mass and stiffness matrices, which result from the use of

the non-orthogonal transpose, are combined, the result is identical to the result obtained from using the inverse in the pre-multiply position. Obviously, the eigensystem of the two transformation results are then also identical. Therefore, if the dimension of the transformation matrix is equal to the dimension of the component matrix, then Ritz vectors, or any linearly independent set of vectors, can form an exact transformation. Also, if the system is a dual matrix system, such as one consisting of a mass and stiffness matrix, then the transpose of the linearly independent set of vectors may be used in the pre-multiply position of the transformation.

## Chapter 4

### Orthogonalization, the Krylov Sequence and Static Ritz Vectors

#### 4.1) Introduction:

This chapter examines several related issues pertaining to the orthogonalization of static Ritz vectors in the boundary flexibility method of component mode synthesis. Efficient orthogonalization is essential to the efficiency of the methodology because the bulk of the computing effort required to produce a static Ritz vector component is in vector orthogonalization. Section 4.2 examines general issues concerning orthogonalization of the vectors obtained from the Krylov sequence. The vectors derived from this sequence must be orthogonalized with respect to previously obtained vectors, usually using a Gram-Schmidt approach, to insure linear independence in the transformation matrix. Various orthogonalization schemes are proposed and examined for their accuracy, robustness, and efficiency. In section 4.3 issues concerning the use of blocks with the Krylov sequence are discussed. These topics include orthogonality within the block, and reducing the block to a manageable size. Algorithms containing the new orthogonalization schemes are presented in section 4.4

#### 4.2) Orthogonalization:

As presented in chapter 2, the static Ritz boundary flexibility method of component synthesis is based upon the Krylov sequence. The vectors derived from this sequence must be orthogonalized with respect to previously obtained vectors to insure linear independence in the transformation matrix. If linear independence, or

orthogonality is not maintained, numerical errors in the transformation are a certainty. In addition, for a system of size  $n$ , the orthogonalization methodology should be robust enough to obtain  $n$  linearly independent vectors, to insure completeness in the representation. Orthogonalization routines such as Gram-Schmidt or modified Gram-Schmidt can be extremely expensive computationally if implemented inefficiently. Lanczos eigenvalue extraction was not used widely until efficient orthogonalization schemes were implemented within the algorithm. Consistent with this, for static Ritz vectors to be practical, the orthogonalization scheme must be accurate, robust, and efficient.

Issues examined in this section include the reason why the loss of orthogonality occurs, checking of vector orthogonality, various Gram-Schmidt orthogonalization schemes, orthonormalization options, and alternate orthogonalization methodologies. An appropriate, workable orthogonalization scheme is suggested. This proposed scheme is contrasted with those used with Lanczos eigenvalue extraction. In Lanczos eigenvalue extraction, it is also important to maintain orthogonality to insure linearly independent vectors. In addition, maintaining the orthogonality also aids in determining multiple eigenvalues, as well as maintaining the tridiagonal form<sup>22</sup>. As discussed in chapter 2, maintaining the tridiagonal form is not important in the use of static Ritz vectors, when used in the boundary flexibility method of component synthesis.

#### 4.2.1) Linear Independence and the Loss of Orthogonality:

It was shown in chapter 3.2 that for an  $n$  size system,  $n$  independent vectors will form a transformation matrix which will allow an exact linear transformation. The boundary flexibility algorithms presented in chapter 2 would not be successful in obtaining  $n$  independent vectors without additional enhancements. Specifically, the normalized Gram-Schmidt orthogonalization technique utilized in these algorithms is not robust enough to obtain independent vectors which span the entire  $n$  space. It must be said that, in typical applications, there is no requirement for the entire space of size  $n$  to be represented. One major advantage of component mode synthesis is a reduction of system size. However, in some cases this loss of orthogonality begins quite early within the Krylov sequence. As a result, no guarantee can be made that a sufficient number of vectors, to adequately represent the component, is obtainable. In addition, whether the component is reduced by means of sequence truncation, vector selection, or any other approach, a correct reduction can not be guaranteed, unless the entire dynamic space is obtainable. Therefore, the static Ritz vector algorithm should be able to yield  $n$  independent vectors for an  $n$  size system.

Theoretically, each Krylov vector is linearly independent of previous vectors. This can be demonstrated by inspection of equation (2.1). Since each vector is a product of multiplication of previous vectors, except for the special cases of null or unity spaces, each vector can not also be defined as a linear combination of previous vectors. However, in practice, numerical dependence does occur in vectors obtained

from a Krylov sequence. Due to this numerical dependence,  $n$  independent vectors, describing an  $n$  size system, can not be directly obtained from the Krylov sequence. Understanding of this problem can be aided by considering a Krylov sequence based upon a space described by the normalized vector,  $x^2$ . Assume, in equation (2.1), that  $A$ , the iteration matrix, is  $x^2$  and that  $\phi$ , the initialing vector, is  $e_1$ , the unit column matrix of order one. The Krylov sequence becomes

$$[e_1, x^2 e_1, x^4 e_1, \dots, x^{2(j-1)} e_1] \quad (4.1)$$

Figure (4.1) shows a plot of the resulting vectors, to the ninth order, normalized to unity. In other words, the sequence of functions  $x^2, x^4, x^6, x^8, x^{10}, x^{12}, x^{14}$  and  $x^{16}$ , have been plotted. The unit column matrix,  $e_1$ , was not plotted. It can be seen that as this series continues to a higher order, the vectors become nearly dependent because of computational roundoff error, and eventually, numerically indistinguishable, despite theoretical independence. The Gram-Schmidt algorithm is not able to orthogonalize a vector which is numerically dependent on previous vectors.

An analytical source of vector dependency, not dependent on computational roundoff, is the situation when the Krylov sequence converges to a normal eigenvector. This state might seem to be desirable, considering the traditional use of eigenvectors in dynamic analysis and component mode representation. (In this circumstance the Krylov sequence acts similar to a power method of eigenvalue extraction.) Unfortunately, when the Krylov vectors have converged and produced

an eigenvector, the next vector in the Krylov sequence is the same eigenvector, scaled, which is linearly dependent. Theorem 4.1 demonstrates the convergence.

Theorem 4.1 - If an eigenvector of the system  $Kx = \lambda Mx$ ,  $x_i$ , appears in the Krylov sequence,  $x_{i+j} = K^{-j} M x_i$ , then the resulting vector,  $x_{i+j}$ , is linearly dependent upon  $x_i$ , and differs only by a scale factor of  $1/\lambda$ .

Proof - If

$$x_{i+1} = \frac{1}{\lambda} x_i \quad (4.2)$$

and since the Krylov sequence is defined as

$$x_{i+1} = K^{-1} M x_i \quad (4.3)$$

then

$$\frac{1}{\lambda} x_i = K^{-1} M x_i \quad (4.4)$$

which leads to

$$K x_i = \lambda M x_i \quad (4.5)$$

which is true, from the definition of an eigenvector. ■

The potential convergence of the Krylov sequence to system eigenvectors has several implications. It is a reason, in addition to not prejudicing the sequence towards a certain eigenvector, that in Lanczos eigenvalue extraction, the Krylov

sequence should be initiated by a random vector. In Lanczos eigenvalue extraction, the system will be re-transformed to a diagonal form and so the eigenvectors are not required at that step. The potential convergence of the Krylov sequence to eigenvectors also restricts shifting of the sequence<sup>23,31</sup>. Shifting the sequence so that it would produce a vector close to an eigenvector would cause the dependency problem discussed above.

In the use of generalized static Ritz vectors, where the vectors which initiate the sequence and the vectors resulting from the sequence have physical significance, the convergence of the Krylov sequence to eigenvectors can cause problems of linear dependency. Consider a cantilevered beam under a gravity load. The deflected shape of the beam is very close to the first normal eigenvector. (A cantilevered beam under a gravity load is used as an example and is illustrated in chapter 5.) If that vector initiates the sequence, subsequent vectors will be nearly linearly dependent on the previous vector. The greater the similarity a static Ritz vector has to an eigenvector, the greater will be the dependency of the subsequent vector in the Krylov sequence. In practice, total dependency does not occur on digital computers due to the same class of roundoff errors that lead to the requirement for re-orthogonalization discussed in section 2.4.1. When two Ritz vectors are nearly linearly dependent, the differences between the two vectors will tend to be random, and so physically significant information that the vector will contribute, after orthogonalization, would be minimal.

This random element resulting from orthogonalization does have some benefit. In Parlett<sup>22</sup>, the process of randomization is discussed. There, randomization is defined as a process of creating a set of random vectors, after several blocks of static Ritz vectors have been obtained, and then orthogonalizing these random vectors with the previous static Ritz vectors. A Krylov sequence initiated with a vector orthogonal to a space requiring representation may have difficulty in producing  $n$  independent vectors for a system of size  $n$ . For example, if the initial vector does not contain a displacement in one of the three ordinate directions, then a pure Krylov sequence would not generate a vector representing the system in that direction. Introducing a random element into all degrees of freedom contained in a vector, and then orthogonalizing, would allow all possible directions and shapes to be represented.

It has been found that due to computational roundoff error an explicit randomization routine is not necessarily required. As discussed above, when a vector is orthogonalized with a vector upon which it is nearly dependent, the purified vector will contain a random element. As a result, randomization occurs to some degree in all Krylov processes implemented on a digital computer. In this manner, vectors may be obtained which are orthogonal to the initiating vector in the Krylov sequence and the complete  $n$  size component space may be spanned.

#### 4.2.2) Use of The Euclidean Norm for Normalization:

In section 2.4.2, it was discussed that in the creation of static Ritz vectors, where the Krylov sequence of equation (2.2) is used to generate the vectors, previous

Gram-Schmidt algorithms orthonormalized the vectors with respect to the mass matrix. This is required if obtaining a tridiagonal form is desired. If the Krylov sequence is based upon equation (2.1), then the Euclidean norm is used. In the implementation of the Gram-Schmidt orthogonalization procedure, it was found that the use of the mass matrix in orthonormalization becomes comparatively and extremely costly for anything other than a small problem size. As a result, alternatives were examined. The tridiagonal form is not a particular advantage using Ritz vectors, as opposed to eigenvectors, in component mode synthesis. In component mode synthesis, by definition, the transformed matrices have large off-diagonal components. Therefore, it is possible to orthonormalize using the Euclidean norm, even though doing this will not produce a diagonal modal mass matrix and a tridiagonal modal stiffness matrix. Again, this feature is not a disadvantage in component mode synthesis and very large savings in computational cost are achievable, as documented in chapter 5.

#### 4.2.3) Gram-Schmidt Failure and Reorthogonalization:

The simple normalized Gram-Schmidt orthonormalization procedure outlined in section 2.2 is inadequate to guarantee a set of orthogonal and linearly independent set of Ritz vectors. In the previous section, it was discussed how linear dependence can arise in vectors generated by a Krylov sequence. It is well documented that the Gram-Schmidt orthogonalization algorithm is not successful at producing an orthogonal vector from a nearly dependent vector<sup>18</sup>. The Gram-Schmidt procedure will fail on occasions when vectors, although theoretically independent, are dependent

within the numerical constraints of current digital computers. These vectors cannot then be made orthogonal, using the single normalized Gram-Schmidt step described in the chapter 2.

One option for orthogonalizing numerically nearly dependent vectors is the modified Gram-Schmidt algorithm. Modified Gram-Schmidt is a computationally expensive procedure which is very successful at orthogonalizing nearly dependent vectors<sup>4,18</sup>. Implementation of modified Gram-Schmidt will be discussed in the next section and its computational expense will be discussed in the next chapter. A potentially less expensive option is to repeat the Gram-Schmidt orthonormalization, if it has been unsuccessful in the first attempt. Reorthogonalization will work, even though the first Gram-Schmidt attempt has been unsuccessful at producing an orthogonal vector, because it modifies the vector enough so that it is no longer numerically nearly dependent. The second normalized Gram-Schmidt step, or reorthogonalization, is then usually successful at producing a orthogonal vector. Two normalized Gram-Schmidt orthogonalizations will typically be less expensive than one modified Gram-Schmidt orthogonalization, especially considering that the reorthogonalization step will not be required for every Krylov vector. Computational costs comparisons for various models will be presented and discussed in chapter 5.

If Gram-Schmidt orthonormalization is to be repeated when unsuccessful, a assessment of the Ritz vectors' orthogonality is required. Upon completion of each Krylov vector calculation, and the associated normalized Gram-Schmidt

orthogonalization, each vector is checked for orthogonality. The orthogonality of the block-Krylov vectors is checked using one of the following equations; either

$$\mathbf{L}_m = \mathbf{q}_{1,j-1}^T \mathbf{m}_{ii} \mathbf{q}_j \quad (4.6)$$

or

$$\mathbf{L}_E = \mathbf{q}_{1,j-1}^T \mathbf{q}_j \quad (4.7)$$

depending on whether the vectors have been orthogonalized with respect to the mass matrix or using the Euclidean vector norm. (The mass matrix used in equation (4.6) is appropriate for fixed interface modes. For the free interface approach, the complete physical mass matrix is used.) If the new vector,  $\mathbf{q}_j$ , is orthogonal to all previously calculated vectors,  $\mathbf{L}_m$  or  $\mathbf{L}_l$  will be a  $j$  by  $l$  size null vector. The infinity norm of the  $\mathbf{L}$  vector, which is defined as

$$\|\mathbf{L}\|_\infty = \max \{ |l_1|, |l_2|, \dots, |l_j| \} \quad (4.8)$$

is then calculated and compared to a specified value,  $e$ . If  $\mathbf{L}_\infty > e$ , then the associated vector is judged to be non-orthogonal and the Gram-Schmidt algorithm is repeated.

#### 4.2.4) Various Gram-Schmidt Orthogonalization Strategies:

A number of combinations of Gram-Schmidt orthonormalization and reorthogonalization strategies are possible. These strategies, for the initial Gram-Schmidt step, include complete orthogonalization and orthogonalization with the

previous two blocks only, a possibility which was discussed in section 2.4. It is possible to use modified Gram-Schmidt exclusively, however that option is not competitive computationally. Complete reorthogonalization, selective reorthogonalization, and selective reorthogonalization using modified Gram-Schmidt were the reorthogonalization options investigated. Orthonormalizing with respect to the mass matrix or using the Euclidean vector norm can be performed with any of the above possibilities. The total number of possibilities investigated, amongst the different combinations possible, is twelve. Of these twelve, the options which were examined and presented in the next chapter are, total initial orthogonalizations with all three reorthogonalization options, and initial orthogonalization with the previous two blocks and with selective reorthogonalization, for a subtotal of four cases. Orthonormalization with respect to the mass matrix and the Euclidean norm for the above four cases was also performed for a total of eight Gram-Schmidt options considered. The computational time required for creating component mode models from various finite element models, using the various Gram-Schmidt options discussed above, is presented in chapter 5.

#### 4.2.5) Cholesky/QR Orthogonalization:

Alternatives to Gram-Schmidt orthonormalization exist which perform the orthogonalization in a matrix format, rather than a vector by vector format. Use of explicit matrix orthogonalization can be an advantage in certain programming applications. Gram-Schmidt is classified as a method of performing a  $\mathbf{B} = \mathbf{QR}$

decomposition, where  $\mathbf{B}$  is a set of vectors (such as unorthogonalized Krylov vectors),  $\mathbf{Q}$  is the orthonormalized vector set, and  $\mathbf{R}$  is an upper triangular matrix, which can be assembled from the coefficients in the Gram-Schmidt algorithm. In typical Gram-Schmidt the  $\mathbf{R}$  matrix is not assembled or used explicitly. The QR decomposition can be performed explicitly by using a Householder technique<sup>18</sup> or by recognizing that the  $\mathbf{R}$  matrix is also the upper Cholesky factor of the  $\mathbf{B}^T\mathbf{B}$  subspace<sup>22</sup>. It is instructive to note the similarity of the  $\mathbf{B}^T\mathbf{B}$  matrix multiplication with equation (4.6), the differences being that all vectors, new and previous, are included in the  $\mathbf{B}$  matrix. Cholesky/QR orthogonalization may be understood as a algorithm which, first, locating the non-orthogonal vectors by the  $\mathbf{B}^T\mathbf{B}$  multiplication, and second, determines a transformation which will shift those vectors to an orthogonal space.

That the  $\mathbf{R}$  transformation matrix can be determined by Cholesky decomposition of the  $\mathbf{B}^T\mathbf{B}$  subspace is demonstrated in the following equations. First, the Cholesky factor is defined as follows: if  $\mathbf{A}$  is positive definite and symmetric then the  $\mathbf{LU}$  decomposition,

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{U} \quad (4.9)$$

is equivalent to

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T \quad (4.10)$$

which may be written as

$$\mathbf{A} = \mathbf{L}\mathbf{D}^{1/2}(\mathbf{L}\mathbf{D}^{1/2})^T = \mathbf{C}^T\mathbf{C} \quad (4.11)$$

where  $\mathbf{C} = \mathbf{D}^{1/2}\mathbf{L}^T$ , and is called the Cholesky factor. That  $\mathbf{B}^T\mathbf{B}$  is symmetric can be shown as follows

$$\mathbf{B}^T\mathbf{B} = (\mathbf{B}^T\mathbf{B})^T = \mathbf{B}^T(\mathbf{B}^T)^T \quad (4.12)$$

Since the  $QR$  decomposition is defined by  $\mathbf{B} = \mathbf{Q}\mathbf{R}$ , then the  $\mathbf{B}^T\mathbf{B}$  matrix product is

$$\mathbf{B}^T\mathbf{B} = (\mathbf{Q}\mathbf{R})^T\mathbf{Q}\mathbf{R} = \mathbf{R}^T\mathbf{Q}^T\mathbf{Q}\mathbf{R} \quad (4.13)$$

and since, by definition,  $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ , then equation (3.24) becomes

$$\mathbf{B}^T\mathbf{B} = \mathbf{R}^T\mathbf{R} = \mathbf{C}^T\mathbf{C} \quad (4.14)$$

and because both  $\mathbf{R}$  and  $\mathbf{C}$  are upper triangular matrices then  $\mathbf{R} = \mathbf{C}$ , and therefore  $\mathbf{R}$  is the upper Cholesky factor of  $\mathbf{B}^T\mathbf{B}$ .

The steps in the Cholesky/QR orthogonalization algorithm can be summarized as follows:

- 1) The matrix product  $\mathbf{B}^T\mathbf{B}$ , or using the notation of chapter 2,  $\mathbf{q}_{1,j}^{*T}\mathbf{q}_{1,j}^*$ , is calculated, where  $\mathbf{q}_{1,j}^*$  is the concatenation of the vectors  $\mathbf{q}_{1,j-1}$  defined in equation (2.11) and the vector block  $\mathbf{q}_j^*$  defined in equation (2.8),

- 2) The matrix product  $B^T B$ , is decomposed into,  $R^T R$ , where  $R$  is upper triangular.
- 3) The equation,  $R^T Q^T = B$ , is solved, with only one forward substitution required since  $R^T$  is lower triangular.

The Cholesky/QR decomposition algorithm can also be used to orthonormalize vectors with respect to the mass matrix. If equation (4.13) is re-written as

$$B^T M B = (QR)^T M QR = R^T Q^T M QR \quad (4.15)$$

and since  $Q^T M Q = I$ , if orthonormalizing with respect to the mass matrix, then equation (4.14) can be re-written as

$$B^T M B = R^T R = C^T C \quad (4.16)$$

which would yield a different transformation matrix,  $R$ , and a different set of orthogonal vectors,  $Q$ , than the previous example of orthonormalizing with respect to the Euclidean vector norm. As discussed in section 2.4, the vectors must be orthonormalized with respect to the mass matrix for the tridiagonal form to be achieved, if the Krylov sequence of equation 2.2 is used. The practical aspect of the tridiagonal form in boundary flexibility component synthesis is that each new vector block theoretically only needs to be orthogonalized with the previous two vector blocks.

It is evident, upon examination of the above algorithm, that Cholesky/QR decomposition orthonormalizes a set, or subset, of vectors simultaneously. This set,

or subset, would include all the vectors within a specific Krylov block, and these vectors are nonorthogonal. However, it is important to note that the previous vectors, which have been successfully orthonormalized, would produce a partition of  $\mathbf{B}^T\mathbf{B}$  that is the identity matrix. Decomposing and solving an identity partition produces a transformation which does not alter vectors which formed the mutually orthonormal subset. As a result, the theoretical description of the tridiagonal form is applicable to vectors orthonormalized using Cholesky/QR decomposition, if, as presented above, they have been orthonormalized with respect to the mass matrix. Theoretically then, it is possible to orthonormalize, using Cholesky/QR, each new vector, or block, with the previous two vectors, or vector blocks, and be orthonormal to all previous vectors, in an identical manner as in the Gram-Schmidt algorithm. Practically, the Cholesky/QR algorithm has an advantage over Gram-Schmidt in that, when the loss of orthogonality occurs within the Lanczos algorithm, the inevitability of which is discussed in section 4.2.1, it does not need to be eliminated immediately. All non-orthogonal vectors can be reorthogonalized simultaneously at intervals, as required, and at the termination of the Krylov sequence. The numerous repetitions of the orthogonality checks required in some implementations of the Gram-Schmidt algorithm is not required in the Cholesky/QR algorithm. However, a method to automatically determine when a reorthogonalization is required, such as presented in reference [26] for Gram-Schmidt, would be required if orthogonalization with the mass matrix and the previous two blocks only were to be implemented practically.

#### 4.3) Issues Concerning the Use of Blocks in the Boundary Flexibility Method:

In the boundary flexibility method of component synthesis, several issues of orthogonalization and usage present themselves, due to the use of blocks in the Krylov sequence. Nour-Omid and Clough<sup>20</sup> described how vector blocks may be used with Wilson's algorithm and that the results pertaining to the tridiagonal form and orthogonalization with the previous two blocks, described in section 2.3, are applicable. A difference resulting from the use of blocks is that orthogonalization must be performed with the previous two blocks, not merely the previous two vectors. Those equations will not be repeated here. The need for orthogonalization within each block was also discussed by Nour-Omid and Clough. However, it was not presented that, depending on the choice of the initializing vectors (forces), the vectors within the block may be approaching linear dependency. In the boundary flexibility method of component synthesis, nearly dependent vectors within a block can and do occur. This is not a numerical convergence of the Krylov sequence, as discussed in section 4.2.1. The nearly dependent vectors within a block is the natural result of the static solution of a structure under generalized loading, such as the mass matrix multiplied by the boundary flexibility matrix as in this dissertation. An illustrating example, is presented in section 4.3.1.

As presented in chapter 2, the initializing block of forces in the boundary flexibility method is the mass matrix multiplied by the constraint modes, or the flexibility modes. The size of these matrices is therefore dependent on the size of the

of the boundary set of the structural component under consideration. In typical aerospace applications the boundary set of these models can be quite large, larger than can be practically used in the block-Krylov sequence. Block size reduction by filtering can correct an excessive block size and this addition to the boundary flexibility method of component synthesis is presented in section 4.3.2.

#### 4.3.1) Dependence of Vectors Within the Block:

In the boundary flexibility method of component synthesis, nearly dependent vectors within a block result from the static solution of a structure under generalized loading. Chapter 2 details this generalized loading as the mass matrix multiplied by the boundary flexibility matrix. The initial set of Ritz vectors is the static response the component exhibits for the generalized loading. There is no theoretical basis to expect that this set of vectors within the initial, or any other, block should be linearly independent. The boundary flexibility algorithm, as presented in reference [1] and reviewed in chapter 2, made no orthogonality check of the vectors within the Krylov block. Furthermore, if internal block orthogonalization is not performed, subsequent Gram-Schmidt steps are ineffective because the blocks are orthogonalized with a set of vectors that are not orthogonal.

Consider the case of a simple beam represented by a finite element model consisting of only two nodes and one element. This single element model is used as an example to demonstrate that the vectors within the initial Krylov block can be almost linearly dependent. The beam element will be processed as a component with

a fixed node interface. The interior partition of the component stiffness and mass matrix, for the simple beam element model, may be obtained directly from beam theory<sup>7</sup>.

$$\mathbf{k}_{ii} = \frac{EI}{L^3} \begin{bmatrix} 12 & -6L \\ -6L & 4L^2 \end{bmatrix} \quad (4.17)$$

$$\mathbf{k}_{ic} = \frac{EI}{L^3} \begin{bmatrix} -12 & -6L \\ 6L & 2L^2 \end{bmatrix} \quad (4.18)$$

$$\mathbf{m}_{ii} = \frac{m}{420} \begin{bmatrix} 156 & -22L \\ -22L & 4L^2 \end{bmatrix} \quad (4.19)$$

The inverse of  $\mathbf{k}_{ii}$  is as follows,

$$\mathbf{k}_{ii}^{-1} = \frac{L^3}{EI} \begin{bmatrix} \frac{1}{3} & \frac{1}{2L} \\ \frac{1}{2L} & \frac{1}{L^2} \end{bmatrix} \quad (4.20)$$

The constraint modes are given by equation (2.5), as

$$\Phi_{ic} = \mathbf{k}_{ii}^{-1} \mathbf{k}_{ic} = \begin{bmatrix} -1 & -L \\ 0 & -1 \end{bmatrix} \quad (4.21)$$

Because the component interface is statically determinate, the constraint modes are identical to the rigid body displacement matrix. The initial block of static Ritz vectors,  $\mathbf{q}_1^{**}$ , as given by equation (2.6), is

$$\mathbf{q}_1^{**} = \mathbf{k}_{ii}^{-1} \mathbf{m}_{ii} \Phi_{ic} = -\frac{mL^3}{420EI} \begin{bmatrix} 41 & 35.666L \\ \frac{56}{L} & 49 \end{bmatrix} \quad (4.22)$$

assuming a lumped mass approach. The vectors contained in  $\mathbf{q}_1^{**}$  can be examined for dependency. For the first vector,  $q_{11}$  over  $q_{21}$  is equal to  $.7321L$ . For the second vector,  $q_{12}$  over  $q_{22}$  is equal to  $.7279L$ . These two vectors are, once normalized, almost identical and linearly dependent.

A large amount of dependency can also occur in larger blocks. Consider a structure with a statically indeterminate interface. In many cases the nodes may be positioned closely together, or in a symmetric fashion, either of which may result in some of the constraint modes being nearly identical. Since the initial set of Ritz vectors is the static displacement of the component to the mass matrix times the constraint modes, it is obvious that many component models will yield nearly dependent vectors, in the initial block.

As mentioned previously, Nour-Omid and Clough<sup>20</sup> presented the requirement for orthogonalization within each block of vectors. They suggested that the Gram-Schmidt orthogonalization procedure, as shown in equations (2.9) through (2.12) be applied in a two step process. First, the Gram-Schmidt orthonormalizing is applied

to each vector within each Krylov block. Then, orthogonalization is performed with the previous vector blocks. An alternate solution to the problem of dependent vectors within the blocks, was presented by Su and Craig in reference [27]. In this solution a singular value decomposition is performed on the  $\mathbf{q}_j^{**T} \mathbf{m} \mathbf{q}_j^{**}$  subspace. The obtained transformation matrix orthogonalizes the block.

The recommended method for orthonormalizing the vectors within the Krylov block has been discussed previously in section 4.2.5, Cholesky/QR orthogonalization. Some advantages of using Cholesky/QR decomposition to obtain an orthonormalizing transformation matrix have already been discussed. In relation to vector blocks, this methodology can orthogonalize a new block separately, or with the previous two blocks, or with all previous blocks, simultaneously. One potential disadvantage is that, if the vectors are almost linearly dependent, than the matrix to be decomposed is numerically singular. As discussed above, this near dependence is to be expected in even simple problems. The potential singularity of the  $\mathbf{B}^T \mathbf{B}$  matrix product can be solved by block filtering, which will be discussed next.

#### 4.3.2) Block Filtering:

Block filtering is a procedure by which nearly identical, or dependent, vectors are removed from a vector block. It is based upon a standard orthogonality check. The use of block filtering simultaneously solves two problems. First, by filtering the vectors, the size of the block created by the boundary flexibility method can be

reduced to a manageable level. Second, it can eliminate the potential singularities of the  $\mathbf{B}^T\mathbf{B}$  matrix product which is used in Cholesky/QR orthonormalization.

The first function of block filtering is block size reduction which corrects excessive block size. The size of the block in the boundary flexibility method is determined by the size of the boundary set of the structural component. The initial block of forces in the boundary flexibility method is the mass matrix multiplied by the constraint modes, or the flexibility modes. For the fixed interface method the equation establishing the first block, (2.6), is repeated here.

$$\mathbf{q}_1^{**} = \mathbf{k}_{ii}^{-1}(\mathbf{m}_{ii}\Phi_{ic} + \mathbf{m}_{ic}) \quad (2.6)$$

The boundary set of practical structural models, which require the use of component mode synthesis, is usually large enough to cause problems in use of the algorithm. For instance, a typical Space Shuttle cargo element component model might have a boundary set of forty-eight degrees of freedom, eight nodes with six degrees of freedom each. This would lead to a block size of forty-eight. In contrast, the default block size in the MSC/NASTRAN implementation of the Lanczos eigenvalue extraction method<sup>32</sup> is seven, with a maximum of fifteen.

There are several reasons why a large block size is a disadvantage. First, as discussed in the previous section, many of the vectors in the blocks may be nearly identical. The information retained after orthogonalizing these nearly dependent vectors may not be significant, and in extreme cases may only be the product of

numerical roundoff. In these cases the Ritz vectors obtained from the second Krylov sequence iteration are more likely to be dynamically significant than the product of numerical roundoff. Second, truncation of the Krylov sequence and final resulting set of vectors becomes very imprecise. The static Ritz vectors are generated block by block and so if the block size is very large many more vectors than desired may be obtained. Finally, with large blocks, orthogonalization within each block becomes more expensive and can become numerically difficult due to the previously discussed dependencies.

The second function of block filtering is the elimination the singularities in the  $B^T B$  matrix product, which is used in Cholesky/QR orthonormalization. If two vectors in the  $B$  matrix are nearly identical than, after normalization, the  $B^T B$  matrix will have a unity term on both (lower and upper) off-diagonal positions corresponding to the column number of the identical vectors. Since each row of the  $B^T B$  matrix has a unity term in the diagonal position, the two rows corresponding to the two identical vectors will be dependent, and Cholesky decomposition becomes problematic. If one of the two vectors is eliminated by block filtering than decomposition can be accomplished. No information is lost in the block filtering because, by definition, the vectors in question are nearly identical, and so the discarded vector is a duplication.

The block filtering procedure can be summarized as follows:

- 1) The vectors within the block,  $q_j^*$ , are normalized as follows

$$q_j^{**} = q_j^* [ \langle \text{diag}(q_j^{*T} m_{ii} q_j^*) \rangle_{cc}^{-1/2} ] \quad (4.23)$$

or, depending on the orthonormalization selection,

$$\mathbf{q}_j^{**} = \mathbf{q}_j^* [ \langle \mathbf{diag}(\mathbf{q}_j^{*T} \mathbf{q}_j^*) \rangle_{cc}^{-1/2} ] \quad (4.24)$$

where *diag* is defined as the diagonal terms of the matrix product and the exponent,  $^{-1/2}$ , is applied to each term within the resulting diagonal matrix, not to the complete matrix itself.

2) The cross-orthogonality of the vectors is calculated

$$\mathbf{L}_m = \mathbf{q}_j^{*T} \mathbf{m}_{ii} \mathbf{q}_j^* \quad (4.25)$$

or, depending on the orthonormalization selection,

$$\mathbf{L}_E = \mathbf{q}_j^{*T} \mathbf{q}_j^* \quad (4.26)$$

At this point no orthogonalization has occurred and so there is no reason to expect the matrix product to result in the identity matrix. In this way, the above equations differ from equation (4.6) and (4.7).

3) The *L* matrix is partitioned into its lower triangular portion, excluding the diagonal terms.

4) The infinity norm, as defined in equation (4.8), of each column in the resulting lower triangular matrix is calculated. These terms are compared to an arbitrary filter value, *e*, determined by practice to be initially set to .995. Vectors with associated terms greater than this value are partitioned from the normalized block.

5) The size of the revised block is determined and if the block is too small or too large then the filter value is raised or lowered, respectively, and step 4) is subsequently repeated. The minimum value for a block has been set at six and the maximum at eighteen.

#### 4.4) Summary and Revised Orthogonalization Algorithm:

The following tables contain revised orthogonalization algorithms for the boundary flexibility method of component synthesis, using static Ritz vectors. These algorithms are a synthesis of the basic methodology described in section 2.2 and the revisions and additions to the method which have been presented in this chapter. Table (4.1) presents the algorithm using the mass matrix for orthonormalization. This algorithm initially orthonormalizes with the previous two vector blocks, and then at intervals, and at the termination of the sequence, full reorthogonalization occurs. Table (4.2) presents the algorithm using the Euclidean vector norm for orthonormalization. Both algorithms are presented for fixed interface components, however, for free interface components the body of algorithms presented are identical. The initialization and transformation of the free interface component are different, as documented in section 2.2.

After the assembly of the component mass,  $m$ , and stiffness,  $k$ , matrices:

1) Initialization

$$\Phi_{ic} = -k_{ii}^{-1} k_{ic} \quad \text{[create boundary flexibility matrix, } k_{ii} \text{ and } k_{ic} \text{ were defined in equation (2.3)]}$$

$$q_1^* = k_{ii}^{-1} (m_{ii} \Phi_{ic} + m_{ic}) \quad \text{[create the initial block]}$$

(For a free interface component, (If the component has rigid body modes then the elastic flexibility matrix,  $g_e$ , defined by (2.35) through (2.39) is used):

$$g_a = \begin{bmatrix} g_{cc} \\ g_{ic} \end{bmatrix} \quad \text{where } g = k^{-1} \quad \text{[boundary flexibility matrix]}$$

$$q_1^* = g m g_a \quad \text{[create the initial block]}$$

2) Filtering of Initial Block

$$q_1^{**} = q_1^* [\langle \text{diag}(q_1^{*T} m_{ii} q_1^*) \rangle_{cc}]^{-1/2} \quad \text{[normalization]}$$

$$L_m = q_1^{**T} m_{ii} q_1^{**} \quad \text{[cross orthogonality]}$$

$$L_m \rightarrow \begin{bmatrix} \setminus & \mathbf{0} \\ \bar{L}_m & \setminus \end{bmatrix} \quad \text{[partition into lower triangular]}$$

$$\text{If } (\langle \|\bar{L}_m\|_{\infty} \rangle_c) \geq e, \text{ then dependent} \quad \text{[infinity norm of each vector]}$$

$$q_1^{**} \rightarrow [q_1^{***} \quad q_{DEP}] \quad \text{[partition out dependent vectors]}$$

Table (4.1) - Revised Boundary Flexibility Algorithm  
Using Orthonormalization With Respect to the Mass Matrix

## 3) Orthonormalization of the Initial Block

$$\mathbf{L}_m^* = \mathbf{q}_1^{***T} \mathbf{m}_{ii} \mathbf{q}_1^{***} \quad [B^T M B \text{ matrix product}]$$

$$\mathbf{L}_m^* = \mathbf{R}^T \mathbf{R} \quad [\text{Cholesky factor decomposition}]$$

$$\mathbf{R}^T \mathbf{q}_1^T = \mathbf{q}_1^{***} \quad [\text{Solve by forward substitution}]$$

4) For Blocks  $j = 2, 3, \dots, l$ 

$$\mathbf{q}_j^* = \mathbf{k}_{ii}^{-1} \mathbf{m}_{ii} \mathbf{q}_{j-1} \quad [\text{Krylov sequence}]$$

$$\mathbf{q}_j^{**} = \mathbf{q}_j^* [ \langle \text{diag}(\mathbf{q}_j^{*T} \mathbf{m}_{ii} \mathbf{q}_j^*) \rangle_{cc} ]^{-1/2} \quad [\text{normalization}]$$

$$\mathbf{L}_m = \mathbf{q}_j^{**T} \mathbf{m}_{ii} \mathbf{q}_j^{**} \quad [\text{cross orthogonality}]$$

$$\mathbf{L}_m \rightarrow \begin{bmatrix} \setminus & \mathbf{0} \\ \bar{\mathbf{L}}_m & \setminus \end{bmatrix} \quad [\text{partition into lower triangular}]$$

If  $(\langle \|\bar{\mathbf{L}}_m\|_\infty \rangle_c) \geq e$ , *than dependent* [infinity norm of each vector]

$$\mathbf{q}_j^{**} \rightarrow [ \mathbf{q}_j^{***} \quad \mathbf{q}_{DEP} ] \quad [\text{partition out dependent vectors}]$$

5) Orthonormalization (At intervals, orthonormalize with all previous blocks)  
(when  $j = l$ , transform system to form component)

$$\mathbf{L}_m^* = \mathbf{q}_{j,j-1,j-2}^{***T} \mathbf{m}_{ii} \mathbf{q}_{j,j-1,j-2}^{***} \quad [B^T M B \text{ matrix product}]$$

$$\mathbf{L}_m^* = \mathbf{R}^T \mathbf{R} \quad [\text{Cholesky factor decomposition}]$$

$$\mathbf{R}^T \mathbf{q}_{j,j-1,j-2}^T = \mathbf{q}_{j,j-1,j-2}^{***} \quad [\text{Solve by forward substitution}]$$

Table (4.1) (Continued) - Revised Boundary Flexibility Algorithm  
Using Orthonormalization With Respect to the Mass Matrix

After the assembly of the component mass,  $m$ , and stiffness,  $k$ , matrices:

1) Initialization

$$\Phi_{ic} = -k_{ii}^{-1} k_{ic} \quad \text{[create boundary flexibility matrix, } k_{ii} \text{ and } k_{ic} \text{ were defined in equation (2.3)]}$$

$$q_1^* = k_{ii}^{-1} (m_{ii} \Phi_{ic} + m_{ic}) \quad \text{[create the initial block]}$$

[For a free interface component, (If the component has rigid body modes then the elastic flexibility matrix,  $g_e$ , defined by (2.35) through (2.39) is used):

$$g_a = \begin{bmatrix} g_{cc} \\ g_{ic} \end{bmatrix} \quad \text{where } g = k^{-1} \quad \text{[boundary flexibility matrix]}$$

$$q_1^* = g m g_a \quad \text{[create the initial block]}$$

2) Filtering of Initial Block

$$q_1^{**} = q_1^* [\langle \text{diag}(q_1^{*T} q_1^*) \rangle_{cc}]^{-1/2} \quad \text{[normalization]}$$

$$L_E = q_1^{**T} q_1^{**} \quad \text{[cross orthogonality]}$$

$$L_E \rightarrow \begin{bmatrix} \setminus & \mathbf{0} \\ \bar{L}_E & \setminus \end{bmatrix} \quad \text{[partition into lower triangular]}$$

$$\text{If } (\langle \|\bar{L}_E\|_\infty \rangle_c) \geq e, \text{ then dependent} \quad \text{[infinity norm of each vector]}$$

$$q_1^{**} \rightarrow [q_1^{***} \quad q_{DEP}] \quad \text{[partition out dependent vectors]}$$

Table (4.2) - Revised Boundary Flexibility Algorithm  
Using Orthonormalization With the Euclidean Vector Norm

## 3) Orthonormalization of the Initial Block

$$\mathbf{L}_E^* = \mathbf{q}_1^{***T} \mathbf{q}_1^{***} \quad [B^T B \text{ matrix product}]$$

$$\mathbf{L}_E^* = \mathbf{R}^T \mathbf{R} \quad [\text{Cholesky factor decomposition}]$$

$$\mathbf{R}^T \mathbf{q}_1^T = \mathbf{q}_1^{***} \quad [\text{Solve by forward substitution}]$$

4) For Blocks  $j = 2, 3, \dots, l$ 

$$\mathbf{q}_j^* = \mathbf{k}_{\ddot{u}}^{-1} \mathbf{m}_{\ddot{u}} \mathbf{q}_{j-1} \quad [\text{Krylov sequence}]$$

$$\mathbf{q}_j^{**} = \mathbf{q}_j^* [ < \mathbf{diag}(\mathbf{q}_j^{*T} \mathbf{q}_j^*) >_{cc} ]^{-1/2} \quad [\text{normalization}]$$

$$\mathbf{L}_E = \mathbf{q}_j^{**T} \mathbf{q}_j^{**} \quad [\text{cross orthogonality}]$$

$$\mathbf{L}_E \rightarrow \begin{bmatrix} \setminus & \mathbf{0} \\ \bar{\mathbf{L}}_E & \setminus \end{bmatrix} \quad [\text{partition into lower triangular}]$$

$$\text{If } ( < \|\bar{\mathbf{L}}_E\|_{\infty} >_c ) \geq e, \text{ then dependent} \quad [\text{infinity norm of each vector}]$$

$$\mathbf{q}_j^{**} \rightarrow [ \mathbf{q}_j^{***} \quad \mathbf{q}_{DEP} ] \quad [\text{partition out dependent vectors}]$$

5) Orthonormalization (each block)  
(when  $j = l$ , transform system to form component)

$$\mathbf{L}_E^* = \mathbf{q}_{1,j}^{***T} \mathbf{q}_{1,j}^{***} \quad [B^T B \text{ matrix product}]$$

$$\mathbf{L}_E^* = \mathbf{R}^T \mathbf{R} \quad [\text{Cholesky factor decomposition}]$$

$$\mathbf{R}^T \mathbf{q}_{1,j}^T = \mathbf{q}_{1,j}^{***} \quad [\text{Solve by forward substitution}]$$

Table (4.2) (Continued) - Revised Boundary Flexibility Algorithm  
Using Orthonormalization With Respect the Euclidean Vector Norm

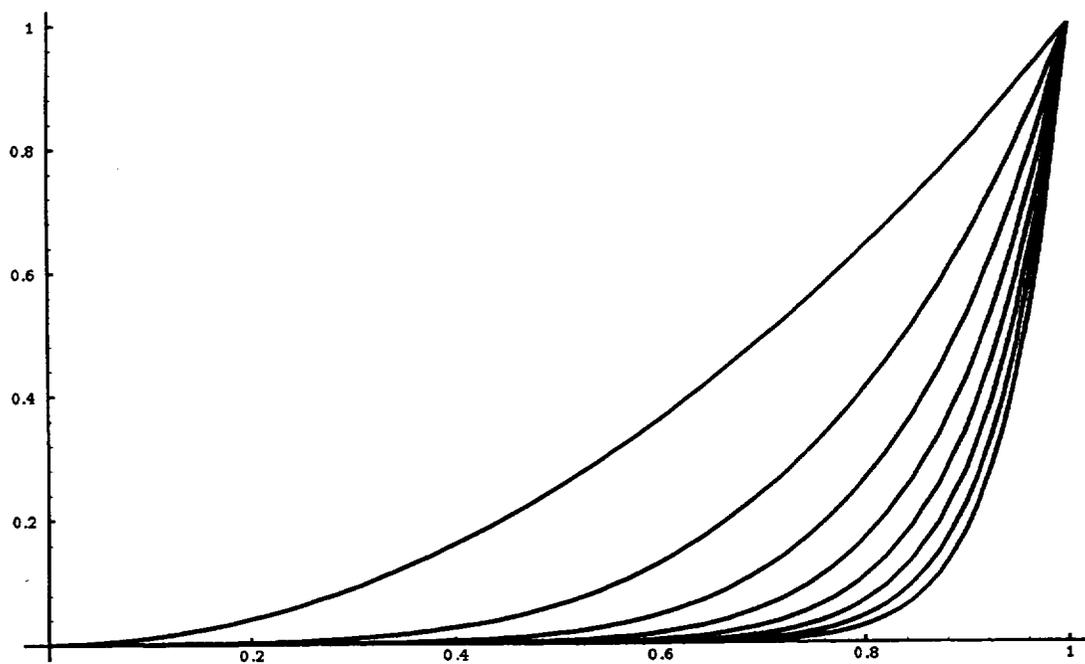


Figure (4.1) - Normalized Vectors Obtained from a Krylov Sequence of Order Nine, with  $x^2$  as the Iteration Matrix

## Chapter 5

### Numerical Examples of Orthonormalization

#### 5.1) Introduction:

The following chapter describes the method with which the algorithms presented in chapter 4 were implemented. Also included are a number of illustrative examples of theoretical concepts, timing comparisons, and descriptions of algorithm trials on practical models. The software and hardware, and finite element models used in these examples is also described.

#### 5.2) Tools and Programming:

The algorithms presented in section 4.3 were implemented in MSC/NASTRAN<sup>32</sup>, Version 67. The use of a standard, commercially available computer program allows the results of this work to be transferred easily to other structural dynamists. Adding the static Ritz vector algorithms to MSC/NASTRAN is implemented by the use of the internal programming language called DMAP (Direct Matrix Abstraction Programming). The standard solution sequences of MSC/NASTRAN are written in DMAP, and the source code of MSC/NASTRAN is available at the DMAP level. For example, equations (2.3) through (2.5), (creation of structural mass and stiffness matrices, partition, and constraint mode creation) and (2.16) through (2.23) (transformation into the modal component) are currently contained in the standard MSC/NASTRAN normal modes solution sequence. The equations presented in Tables (4.1) and (4.2) were written using DMAP, and were

then incorporated into the MSC/NASTRAN normal modes solution sequence. In addition, Gram-Schmidt orthonormalization was also implemented, to allow comparisons with the presented Cholesky/QR decomposition algorithm.

A Cray X-MP running under the UNICOS operating system was the resident hardware running the version of MSC/NASTRAN utilized. This fact is relevant when considering the timing numbers presented in section 5.4.

### 5.3) Finite Element Models:

All finite elements models were created using standard MSC/NASTRAN. They required no special processing. The normal modes solution sequence created the finite element component mass and stiffness matrices and performed the partitions into internal and external degrees of freedom. They were then ready to be processed by the boundary flexibility algorithm using Ritz vectors.

#### 5.3.1) Simple Beam Model:

The first example case of a simple beam was derived from a finite element model of the Space Station Freedom photovoltaic array central mast. The length of the beam was 1179.9 inches. The modulus of elasticity,  $E$ , was  $10.1 \times 10^6$  lbs/in<sup>2</sup> and the moment of inertia of the cross section was 108.9 in<sup>4</sup>. Its weight per unit length was .2296 lbs/in. The simple beam was modeled with eleven nodes and ten beam finite elements. Several different boundary conditions, both at the component and system level, were imposed upon this beam, yielding cantilevered and free-free conditions. The different boundary conditions cases will be described in section 5.4.

### 5.3.2) Space Station Electrical Power System Radiator:

The next example used was a finite element model of the Space Station Electrical Power System (EPS) Radiator (Figure (5.1)). The main contractor constructing the EPS Radiator is Loral Vought Systems. Its purpose is to expel excess heat created by Space Station Electrical Power System into space. The structure weighs approximately 1440 pounds and, when deployed, is approximately 50 feet long and 12 feet wide. The finite element model representing this structure was created by Loral Vought Systems (Figure (5.2)). This finite element model contains approximately 4000 degrees of freedom. With the boundary degrees of freedom fixed, the EPS Radiator finite element produces eight normal modes below 5 Hz, the first three being at .19, .73, and .94 Hz (Figures (5.3), (5.4), and (5.5), respectively).

### 5.3.3) Cassini Spacecraft:

The third example used was a finite element model of the Cassini Spacecraft deep space probe (Figure (5.6)). The primary organization responsible for the Cassini spacecraft is the Jet Propulsion Laboratory of the California Institute of Technology and NASA. This spacecraft will be launched upon a Titan IV launch vehicle and will explore the Saturn planetary system. The structure weighs approximately 12,890 pounds and is approximately 23 feet long and 14 feet wide. The finite element model representing this structure was created by the Jet Propulsion Laboratory (Figure (5.7)). This finite element model contains approximately 11,100 degrees of freedom. With the boundary degrees of freedom fixed, this Cassini finite element model produces

sixty-three normal modes below 50 Hz, the first three primary structural modes being at 7.36 (bending), 7.70 (bending), and 15.78 Hz (torsion).

#### 5.4) Numerical Results:

The following numerical examples can be divided into two groups. The first, contained in sections 5.4.1 and 5.4.2, are illustrations of some of the theoretical properties, presented in chapter 4, of generalized static Ritz vectors in the boundary flexibility method. These examples also serve to validate the correctness of the implementation of the presented theory. Sections 5.4.3 and 5.4.4 contain examples of a more practical nature. These examples serve to provide a physical understanding of the methodology and its benefits in terms of computer cost.

##### 5.4.1) Numerical Illustrations of Theoretical Properties:

If a number of Ritz vectors, equal to the number of degrees of freedom in a finite element model, are used to form a component, then those vectors do not represent a Ritz approximation but are an exact transformation. That a component so formed is exact was proven in section 3.2. A demonstration of that proof, and of the correct implementation of the boundary flexibility/static Ritz vector methodology and algorithms presented in chapter 4, is shown in Table (5.1). A complete set of Ritz vectors, equal to the number of degrees of freedom in the beam model, were calculated. The interface was assumed to be fixed and so the beam was cantilevered. A boundary flexibility component model was subsequently formed. The normal eigenvalues of the transformed component were then calculated and compared to the

eigenvalues obtained directly from the finite element model of the cantilevered beam. This is the comparison contained in Table (5.1).

The cross orthogonality of the Ritz vectors, as defined in equations (3.6) and (3.7), used in the exact transformation discussed in the preceding paragraph, were calculated. The six largest pairs of off-diagonal terms resulting from the cross orthogonality calculation for two different cases are given in Tables (5.2) and (5.3). The vectors used to form the matrix, from which the terms contained in Table (5.2) were extracted, were orthonormalized with respect to mass matrix. In order to facilitate obtaining the exact transformation, orthonormalization was performed on all previous vectors at each step in the Krylov sequence, not with just the previous two vector blocks. The vectors used to form the matrix, from which the terms contained in Table (5.3) were extracted, were orthonormalized using the Euclidean vector norm.

The orthogonality properties of the Krylov sequence are illustrated in Tables (5.4) and (5.5). As was discussed in section 2.3, theoretically, if Ritz vectors are orthonormalized with respect to the mass matrix and the preceding two blocks of vectors, then they are orthogonal with all previously calculated vectors. Table (5.4) contains the orthogonality check of the first five blocks of vectors produced for the cantilevered beam example described above, orthonormalized with respect to the mass matrix. Theoretically, all vectors should be mutually orthonormal and this matrix should be the identity matrix. The extremely large terms at position (1,30) and (30,1) of the matrix are illustrative of the inherent breakdown of orthogonality which was

discussed in section 4.2.1, and demonstrated by Theorem 4.1. The rapid creation of non-orthogonal vectors in this small idealized example clearly demonstrates the requirement for re-orthogonalization. When calculating Ritz vectors for a more realistic problem, the requirement for continual re-orthogonalization can outweigh the advantages of orthogonalization with just the previous two blocks and it can become more efficient to orthogonalize with all previous vector blocks.

Table (5.5) also contains an orthogonality check with the previous two vector blocks. However, in this example the vectors have been orthogonalized using the Euclidean vector norm. There is no theoretical reason why these vectors should be orthogonal with all previous vectors and they are not. The vectors are orthogonal with the previous two blocks, but non-orthogonality is manifest between the other vectors. Specifically, the vectors in block four of the example are not orthogonal with those in block one and the vectors in block five are not orthogonal with the vectors in blocks one and two. As a result, whatever advantages exist for using orthonormalization with the Euclidean vector norm must be weighed against the requirement for orthogonalization with all previous vectors or more difficult re-orthogonalization.

#### 5.4.2) Numerical Illustrations of Block Issues:

The Krylov block issues of dependence, size, and filtering, which were discussed in section 4.3, can be illustrated by the following example, using the EPS Radiator finite element model. This model has a boundary set of six nodes, which is

a typical size in practical aerospace applications. The six nodes each have six degrees of freedom and so the boundary set in this component model has thirty-six degrees of freedom, which in turn leads to thirty-six constraint modes and an initial block size of thirty-six. Clearly, as discussed in section 4.3, this is an unwieldy amount of vectors to process. The cross-orthogonality of the initial block is presented in Table (5.6). As can be seen, this is a large matrix which contains many nearly identical vectors, as identified by the many cross-orthogonality terms approaching unity. This is a further demonstration of the initial nonorthogonality of vectors within a block. The initial, large block of vectors was filtered to produce a new set of eight vectors. The cross-orthogonality of the filtered vector block presented in Table (5.7). Since only nearly identical vectors were removed, the filtered vector block contains almost the identical response information as the original block, is a more convenient size to work with, and is numerically cleaner and easier to orthonormalize.

#### 5.4.3) Simple Beam Numerical Results:

A component representation of the ten element beam finite element model was created using the boundary flexibility method and static Ritz vectors. The fixed interface approach, with constraint modes and two Krylov blocks, was used to form the component. The interface of the component consisted of one node and six degrees of freedom. The number of constraint nodes is equal to the number of interface degrees of freedom, and the size of the Krylov block is equal to the number of constraint modes. Therefore, each Krylov block contained six Ritz vectors. Since the

component model was formed with two Krylov blocks, it contained a total of twelve generalized coordinates.

Plots of the lateral static Ritz vectors, which represented the cantilevered beam, are shown in Figures (5.8-11). The unorthogonalized vectors, as produced by equations (2.6) and (2.8), or as given in Table (4.1), are shown in Figures (5.8-9)(a) and (5.10-11)(a), respectively. The first normalized vector, as produced by equation (2.7), is given in Figure (5.8)(b). (The first vector does not need to be orthogonalized.) The remaining orthogonalized and normalized vectors, as produced by equation (2.12), are given in Figures (5.9-11)(b). The first two unorthogonalized vectors (given in Figures (5.8-9)(a)), which are in the first Krylov block, appear to be nearly identical. The first static Ritz vector produced by the boundary flexibility method is similar to the classic first bending normal mode shape of a beam. After orthonormalization, the second static Ritz vector has been modified into the classic second bending normal mode shape (as shown in Figures (5.8-9)(b)).

The similarity of the Ritz vectors to the normal modes of a cantilevered beam provides an important insight into the numerical difficulties of orthogonalizing vectors which result from the Krylov sequence. It was shown in theorem 4.1, section 4.2.1, that the Krylov sequence converges to an eigenvector. In other words, when an eigenvector is input into the sequence, it produces the same eigenvector. The resulting complete vector dependency will cause numerical orthogonalization to be very difficult. Even when the Ritz vectors closely resemble the normal modes, but are not

exactly the normal modes, near duplication of these vectors from the subsequent Krylov iteration occurs. This convergence to normal modes is the potentially important disadvantage of creating boundary flexibility static Ritz vectors which closely resemble normal modes. This convergence also partially explains the high degree of nonorthogonality evident in Tables (5.4) and (5.5).

The eigenvalues of the reduced component subspace were then calculated to evaluate the static Ritz vector representation. The first five single plane natural frequencies from this reduced system are shown in Table (5.8). For comparison, Table (5.8) also includes the frequencies of a reduced system where the component was formed using traditional normal modes. This component was also formed with a fixed interface, but thirteen normal modes were used for numerical convenience. The full, or "exact", finite element eigenvalue solution is also shown. In the case of the Ritz vectors, no modal selection of any kind was used. For the case of the normal modal component, modal selection by truncation was used. The superior accuracy of the normal modal component, in the fourth bending mode, does not necessarily represent a limitation of the boundary flexibility methodology, but instead demonstrates the requirement for an adequate Krylov sequence truncation criteria. This subject will be discussed in chapter 7.

In addition to the fixed interface example, two free interface examples were created. Both were based on the same ten element beam finite element model, but with different boundary conditions. The first free component model considered was

free-fixed, with the component interface being the free boundary condition, and hence it had no rigid-body modes. Equations (2.20) through (2.34) define the formulation of this free interface component. This component was also formed with two Krylov blocks, and therefore twelve Ritz vectors. Table (5.9) presents the first five single plane system natural frequencies of the reduced component subspace, compared to the natural frequencies of the full finite element model. The second free component model considered consisted of the ten element beam with free-free boundary conditions and six rigid body modes. Equations (2.35) through (2.42) define the formulation of this component. Table (5.10) contains the six rigid body frequencies and the first four single plane elastic frequencies of the reduced component subspace, compared to the frequencies of the full finite element model. In the two free interface cases, there is no comparison with the normal component mode synthesis. This is because standard free-interface MSC/NASTRAN routine does not use the "Rubin-MacNeal" method, and so a direct comparison was not performed.

#### 5.4.4) Timing Comparisons:

Static Ritz vector component models were formed using the boundary flexibility methodology for the EPS Radiator finite element model and the Cassini spacecraft finite element model. A variety of options was used in performing these computer runs. Orthonormalization was performed using Cholesky/QR and Gram-Schmidt, and with respect to the mass matrix and with the Euclidean vector norm. When orthonormalizing with respect to the mass matrix, orthonormalization was

performed both with all previous blocks and with the previous two blocks, as allowed theoretically. When Gram-Schmidt was used, reorthogonalization was performed with all vectors, with selected vectors, and with selected vectors using modified Gram-Schmidt. The timing comparisons are presented in Table (5.11) for the EPS Radiator, and Table (5.12) for the Cassini. In these comparisons a specified, consistent number of Ritz vectors was created. The number of vectors specified was sufficient to create an accurate component, for an arbitrary frequency cutoff. A Cray X-MP, using the UNICOS operating system, was the computer hardware system used to perform these timing comparisons.

The algorithms were implemented using DMAP, in version 67 of MSC/NASTRAN. As a result, some caution should be used in interpreting the timing comparison data. Each DMAP module calls an independent set of compiled fortran routines, and each call takes a certain amount of computer time, which is essentially overhead. As a result, a large sequence of DMAP statements, especially a loop which will be repeated many times, will not be efficient as programs written in some compiled computer languages, such as FORTRAN. The results of the modified Gram-Schmidt reorthogonalization option would be particularly misleading, because of the large number of separate DMAP calls. On the positive side, the results for static Ritz vector, boundary flexibility component synthesis represent a lower bound estimate of the likely improvement in computer time. If implemented more efficiently, such as in the FORTRAN code of NASTRAN, this methodology should demonstrate a greater

time savings in forming the component than the already significant amount shown for the Cassini model in Table (5.12). The Cassini spacecraft model was the largest finite element model considered.

<u>Mode</u>	<u>Ritz</u>	<u>Normal Modes</u>
1	.5463995	.5463995
2	.5463995	.5463995
3	3.414649	3.414649
4	3.414649	3.414649
5	9.522067	9.522067
6	9.522067	9.522067
7	18.56508	18.56508
8	18.56508	18.56508
9	27.57835	27.57835
10	30.53347	30.53347
11	30.53347	30.53347
12	45.41006	45.41006
13	45.41006	45.41006
14	63.18172	63.18172
15	63.18172	63.18172
16	81.89710	81.89710
17	83.59296	83.59296
18	83.59296	83.59296
19	84.63568	84.63568
20	104.4086	104.4086
21	104.4086	104.4086
22	133.7274	133.7274
23	151.3342	151.3342
24	151.3342	151.3342
25	181.4946	181.4946
26	182.6918	182.6918
27	182.6918	182.6918
28	223.0661	223.0661
29	223.0661	223.0661
30	223.7470	223.7470
31	251.5984	251.5984
32	259.2011	259.2011
33	271.7271	271.7271
34	271.7271	271.7271
35	286.7794	286.7794

Table (5.1) - A Comparison of Cantilevered Beam Frequencies (Hz) Obtained From a Static Ritz Component Transformed into Normal Eigenvalues and a Direct Normal Eigenvalue Solution

<u>Mode</u>	<u>Ritz</u>	<u>Normal Modes</u>
36	305.6441	305.6441
37	315.2220	315.2220
38	329.7375	329.7375
39	329.7375	329.7375
40	397.8221	397.8221
41	397.8221	397.8221
42	411.6982	411.6982
43	473.4397	473.4397
44	473.4397	473.4397
45	543.8633	543.8633
46	543.8633	543.8633
47	560.5679	560.5679
48	692.9518	692.9518
49	692.9518	692.9518
50	694.1468	694.1468
51	808.7912	808.7912
52	901.3738	901.3738
53	969.3694	969.3694
54	1010.923	1010.923

Table (5.1)(Continued) - A Comparison of Cantilevered Beam Frequencies (Hz)  
Obtained From a Static Ritz Component Transformed into Normal Eigenvalues  
and a Direct Normal Eigenvalue Solution

Row 1, Selected Columns

1,1) 1.0000e+00 1,54) 1.9043e-14

Row 6, Selected Columns

6,6) 1.0000e+00 6,54) -1.4917e-14

Row 12, Selected Columns

12,12) 1.0000e+00 12,54) -1.6464e-14

Row 13, Selected Columns

13,13) 1.0000e+00 13,54) 1.3798e-14

Row 39, Selected Columns

39,39) 1.0000e+00 39,51) -1.3741e-14

Row 49, Selected Columns

49,49) 1.0000e+00 49,54) -2.2577e-14

Row 51, Selected Columns

51,39) -1.3741e-14 51,51) 1.0000e+00

Row 54, Selected Columns54,1) 1.9043e-14 54,6) -1.4917e-14 54,12) -1.6464e-14 54,13) 1.3798e-14  
54,49) -2.2577e-14 54,54) 1.0000e+00

Table (5.2) - The Largest Off-Diagonal Terms From an Orthogonality Check of the Static Ritz Vectors Representing a Cantilevered Beam, Using Orthonormalization With Respect to the Mass Matrix

Row 39, Selected Columns

39,39) 1.0000e+00 39,51) 9.4066e-15

Row 40, Selected Columns

40,40) 1.0000e+00 40,51) 8.4148e-15

Row 42, Selected Columns

42,42) 1.0000e+00 42,54) -1.7574e-14

Row 43, Selected Columns

43,43) 1.0000e+00 43,54) 1.1849e-14

Row 48, Selected Columns

48,48) 1.0000e+00 48,54) 1.2303e-14

Row 49, Selected Columns

49,49) 1.0000e+00 49,54) 2.1308e-14

Row 51, Selected Columns

51,39) 9.4066e-15 51,40) 8.4148e-15 51,51) 1.0000e+00

Row 54, Selected Columns54,42) -1.7574e-14 54,43) 1.1849e-14 54,48) 1.2303e-14  
54,49) 2.1308e-14 54,54) 1.0000e+00

Table (5.3) - The Largest Off-Diagonal Terms From an Orthogonality Check of the Static Ritz Vectors Representing a Cantilevered Beam, Using Orthonormalization With the Euclidean Vector Norm

Block One

1,1)	1.0000e+00	1,25)	1.6720e-02	1,30)	5.8706e-01
2,2)	1.0000e+00				
3,3)	1.0000e+00	3,28)	9.8957e-02		
4,4)	1.0000e+00				
5,5)	1.0000e+00				
6,6)	1.0000e+00				

Block Two

7,7)	1.0000e+00
8,8)	1.0000e+00
9,9)	1.0000e+00
10,10)	1.0000e+00
11,11)	1.0000e+00
12,12)	1.0000e+00

Block Three

13,13)	1.0000e+00
14,14)	1.0000e+00
15,15)	1.0000e+00
16,16)	1.0000e+00
17,17)	1.0000e+00
18,18)	1.0000e+00

Block Four

19,19)	1.0000e+00
20,20)	1.0000e+00
21,21)	1.0000e+00
22,22)	1.0000e+00
23,23)	1.0000e+00
24,24)	1.0000e+00

Block Five

25,1)	1.6720e-02	25,25)	1.0000e+00
26,26)	1.0000e+00		
27,27)	1.0000e+00		
28,3)	9.8957e-02	28,28)	1.0000e+00
29,29)	1.0000e+00		
30,1)	5.8706e-01	30,30)	1.0000e+00

Table (5.4) - The Cross Orthogonality of Static Ritz Vectors Created by Orthonormalizing With the Previous Two Blocks, Using Orthonormalization With Respect to the Mass Matrix

Block One

1,1)	1.0000e+00	1,19)	-9.9924e-01	1,24)	-3.8427e-02		
2,2)	1.0000e+00	2,20)	-9.9135e-01	2,26)	-1.2688e-01		
3,3)	1.0000e+00	3,21)	-9.9924e-01	3,22)	3.8427e-02		
4,4)	1.0000e+00	4,21)	-3.9085e-02	4,22)	-9.8199e-01	4,27)	1.8463e-01
5,5)	1.0000e+00						
6,6)	1.0000e+00	6,19)	3.9085e-02	6,24)	-9.8199e-01	6,25)	-1.8463e-01

Block Two

7,7)	1.0000e+00	7,25)	-4.9289e-02	7,30)	9.9601e-01		
8,8)	1.0000e+00	8,26)	-2.5613e-01				
9,9)	1.0000e+00	9,27)	-4.9289e-02	9,28)	-9.9601e-01		
10,10)	1.0000e+00	10,28)	-7.4190e-02				
11,11)	1.0000e+00						
12,12)	1.0000e+00	12,30)	-7.4190e-02				

Block Three

13,13)	1.0000e+00						
14,14)	1.0000e+00						
15,15)	1.0000e+00						
16,16)	1.0000e+00						
17,17)	1.0000e+00						
18,18)	1.0000e+00						

Block Four

19,1)	-9.9924e-01	19,6)	3.9085e-02	19,19)	1.0000e+00		
20,2)	-9.9135e-01	20,20)	1.0000e+00				
21,3)	-9.9924e-01	21,4)	-3.9085e-02	21,21)	1.0000e+00		
22,3)	3.8427e-02	22,4)	-9.8199e-01	22,22)	1.0000e+00		
23,23)	1.0000e+00						
24,1)	-3.8427e-02	24,6)	-9.8199e-01	24,24)	1.0000e+00		

Block Five

25,6)	-1.8463e-01	25,7)	-4.9289e-02	25,25)	1.0000e+00		
26,2)	-1.2688e-01	26,8)	-2.5613e-01	26,26)	1.0000e+00		
27,4)	1.8463e-01	27,9)	-4.9289e-02	27,27)	1.0000e+00		
28,9)	-9.9601e-01	28,10)	-7.4190e-02	28,28)	1.0000e+00		
29,29)	1.0000e+00						
30,7)	9.9601e-01	30,12)	-7.4190e-02	30,30)	1.0000e+00		

Table (5.5)- The Cross Orthogonality of Static Ritz Vectors Created by Orthonormalizing With the Previous Two Blocks, Using Orthonormalization With the Euclidean Vector Norm

Row 1, Columns 1 Thru 36

1) 1.0000e+00 -9.9999e-01 -9.9604e-01 -9.9996e-01 9.5350e-01 -9.9999e-01 -9.7551e-01 -9.9976e-01 9.9978e-01 -9.9964e-01  
 11) 7.5250e-01 -1.6221e-01 -9.9941e-01 -9.9906e-01 -9.9289e-01 -9.9924e-01 -9.4312e-01 9.9905e-01 -9.9947e-01 9.9967e-01  
 21) -9.9854e-01 -9.9967e-01 9.9860e-01 9.9959e-01 -9.7430e-01 -9.9972e-01 9.9975e-01 -9.9973e-01 7.8313e-01 1.3869e-02  
 31) 9.9951e-01 9.9954e-01 -9.9908e-01 -9.9963e-01 -9.9900e-01 -9.9952e-01

Row 2, Columns 1 Thru 36

1) -9.9999e-01 1.0000e+00 9.9601e-01 9.9995e-01 -9.5348e-01 1.0000e+00 9.7619e-01 9.9971e-01 -9.9973e-01 9.9959e-01  
 11) -7.5179e-01 1.5939e-01 9.9931e-01 9.9896e-01 9.9282e-01 9.9916e-01 9.4310e-01 -9.9895e-01 9.9954e-01 -9.9971e-01  
 21) 9.9867e-01 9.9969e-01 -9.9872e-01 -9.9964e-01 9.7493e-01 9.9968e-01 -9.9971e-01 9.9968e-01 -7.8530e-01 -1.5068e-02  
 31) -9.9937e-01 -9.9941e-01 9.9888e-01 9.9951e-01 9.9879e-01 9.9938e-01

Row 3, Columns 1 Thru 36

1) -9.9604e-01 9.9601e-01 1.0000e+00 9.9615e-01 -9.7650e-01 9.9589e-01 9.8395e-01 9.9409e-01 -9.9425e-01 9.9473e-01  
 11) -6.9905e-01 1.0225e-01 9.9328e-01 9.9217e-01 9.8160e-01 9.9282e-01 9.2030e-01 -9.9210e-01 9.9320e-01 -9.9385e-01  
 21) 9.9044e-01 9.9359e-01 -9.9057e-01 -9.9359e-01 9.8418e-01 9.9381e-01 -9.9433e-01 9.9407e-01 -8.1058e-01 -8.9784e-02  
 31) -9.9442e-01 -9.9452e-01 9.9475e-01 9.9450e-01 9.9456e-01 9.9483e-01

Row 4, Columns 1 Thru 36

1) -9.9996e-01 9.9995e-01 9.9615e-01 1.0000e+00 -9.5398e-01 9.9994e-01 9.7488e-01 9.9975e-01 -9.9979e-01 9.9979e-01  
 11) -7.5412e-01 1.6492e-01 9.9947e-01 9.9915e-01 9.9343e-01 9.9937e-01 9.4482e-01 -9.9913e-01 9.9946e-01 -9.9968e-01  
 21) 9.9843e-01 9.9965e-01 -9.9848e-01 -9.9961e-01 9.7366e-01 9.9968e-01 -9.9981e-01 9.9975e-01 -7.8143e-01 -1.1353e-02  
 31) -9.9956e-01 -9.9959e-01 9.9901e-01 9.9963e-01 9.9892e-01 9.9958e-01

Row 5, Columns 1 Thru 36

1) 9.5350e-01 -9.5348e-01 -9.7650e-01 -9.5398e-01 1.0000e+00 -9.5308e-01 -9.7285e-01 -9.4745e-01 9.4793e-01 -9.4997e-01  
 11) 5.4647e-01 5.0223e-02 -9.4547e-01 -9.4255e-01 -9.2212e-01 -9.4442e-01 -8.3595e-01 9.4234e-01 -9.4538e-01 9.4702e-01  
 21) -9.3835e-01 -9.4610e-01 9.3865e-01 9.4636e-01 -9.7639e-01 -9.4656e-01 9.4828e-01 -9.4744e-01 8.5363e-01 2.7095e-01  
 31) 9.4900e-01 9.4928e-01 -9.5089e-01 -9.4899e-01 -9.5045e-01 -9.5032e-01

Row 6, Columns 1 Thru 36

1) -9.9999e-01 1.0000e+00 9.9589e-01 9.9994e-01 -9.5308e-01 1.0000e+00 9.7610e-01 9.9973e-01 -9.9974e-01 9.9959e-01  
 11) -7.5237e-01 1.5987e-01 9.9933e-01 9.9898e-01 9.9290e-01 9.9918e-01 9.4331e-01 -9.9897e-01 9.9957e-01 -9.9974e-01  
 21) 9.9874e-01 9.9972e-01 -9.9879e-01 -9.9967e-01 9.7481e-01 9.9970e-01 -9.9972e-01 9.9969e-01 -7.8513e-01 -1.4196e-02  
 31) -9.9937e-01 -9.9940e-01 9.9886e-01 9.9951e-01 9.9878e-01 9.9937e-01

Row 7, Columns 1 Thru 36

1) -9.7551e-01 9.7619e-01 9.8395e-01 9.7488e-01 -9.7285e-01 9.7610e-01 1.0000e+00 9.7086e-01 -9.7096e-01 9.7087e-01  
 11) -6.0711e-01 -5.8122e-02 9.6749e-01 9.6537e-01 9.4835e-01 9.6691e-01 8.6941e-01 -9.6530e-01 9.7294e-01 -9.7297e-01  
 21) 9.7127e-01 9.7230e-01 -9.7154e-01 -9.7271e-01 9.9978e-01 9.7074e-01 -9.7084e-01 9.7077e-01 -8.9989e-01 -2.0921e-01  
 31) -9.6844e-01 -9.6868e-01 9.6838e-01 9.6942e-01 9.6805e-01 9.6889e-01

Row 8, Columns 1 Thru 36

1) -9.9976e-01 9.9971e-01 9.9409e-01 9.9975e-01 -9.4745e-01 9.9973e-01 9.7086e-01 1.0000e+00 -9.9999e-01 9.9979e-01  
 11) -7.6648e-01 1.8253e-01 9.9989e-01 9.9975e-01 9.9490e-01 9.9980e-01 9.4872e-01 -9.9975e-01 9.9967e-01 -9.9982e-01  
 21) 9.9902e-01 9.9988e-01 -9.9905e-01 -9.9978e-01 9.6940e-01 9.9999e-01 -9.9997e-01 9.9996e-01 -7.7111e-01 0.0000e+00  
 31) -9.9979e-01 -9.9980e-01 9.9922e-01 9.9986e-01 9.9916e-01 9.9973e-01

Row 9, Columns 1 Thru 36

1) 9.9978e-01 -9.9973e-01 -9.9425e-01 -9.9979e-01 9.4793e-01 -9.9974e-01 -9.7096e-01 -9.9999e-01 1.0000e+00 -9.9985e-01  
 11) 7.6607e-01 -1.8215e-01 -9.9990e-01 -9.9974e-01 -9.9492e-01 -9.9981e-01 -9.4877e-01 9.9974e-01 -9.9966e-01 9.9982e-01  
 21) -9.9895e-01 -9.9987e-01 9.9898e-01 9.9978e-01 -9.6951e-01 -9.9998e-01 9.9999e-01 -9.9998e-01 7.7128e-01 0.0000e+00  
 31) 9.9981e-01 9.9982e-01 -9.9921e-01 -9.9987e-01 -9.9915e-01 -9.9975e-01

Row 10, Columns 1 Thru 36

1) -9.9964e-01 9.9959e-01 9.9473e-01 9.9979e-01 -9.4997e-01 9.9959e-01 9.7087e-01 9.9979e-01 -9.9985e-01 1.0000e+00  
 11) -7.6582e-01 1.8200e-01 9.9975e-01 9.9960e-01 9.9536e-01 9.9980e-01 9.5065e-01 -9.9958e-01 9.9952e-01 -9.9969e-01  
 21) 9.9856e-01 9.9967e-01 -9.9856e-01 -9.9966e-01 9.6944e-01 9.9972e-01 -9.9991e-01 9.9987e-01 -7.7124e-01 0.0000e+00  
 31) -9.9968e-01 -9.9970e-01 9.9885e-01 9.9967e-01 9.9876e-01 9.9965e-01

Row 11, Columns 1 Thru 36

1) 7.5250e-01 -7.5179e-01 -6.9905e-01 -7.5412e-01 5.4647e-01 -7.5237e-01 -6.0711e-01 -7.6648e-01 7.6607e-01 -7.6582e-01  
 11) 1.0000e+00 -7.1435e-01 -7.7317e-01 -7.7935e-01 -8.2077e-01 -7.7680e-01 -9.0551e-01 7.7955e-01 -7.6740e-01 7.6551e-01  
 21) -7.7476e-01 -7.6670e-01 7.7381e-01 7.6684e-01 -5.9718e-01 -7.6694e-01 7.6661e-01 -7.6683e-01 2.5430e-01 -6.4781e-01  
 31) 7.6653e-01 7.6609e-01 -7.5886e-01 -7.6493e-01 -7.5924e-01 -7.6421e-01

Row 12, Columns 1 Thru 36

1) -1.6221e-01 1.5939e-01 1.0225e-01 1.6492e-01 5.0223e-02 1.5987e-01 -5.8122e-02 1.8253e-01 -1.8215e-01 1.8200e-01  
 11) -7.1435e-01 1.0000e+00 1.9580e-01 2.0415e-01 2.5665e-01 1.9830e-01 3.9512e-01 -2.0440e-01 1.7371e-01 -1.7391e-01  
 21) 1.7840e-01 1.7672e-01 -1.7734e-01 -1.7494e-01 -6.4138e-02 1.8295e-01 -1.8267e-01 1.8287e-01 4.7562e-01 9.0808e-01  
 31) -1.9090e-01 -1.9001e-01 1.8765e-01 1.8719e-01 1.8867e-01 1.8869e-01

Table (5.6) - The Cross Orthogonality of the First Krylov Block of the EPS Radiator

Row 13, Columns 1 Thru 36

1) -9.9941e-01 9.9931e-01 9.9328e-01 9.9947e-01 -9.4547e-01 9.9933e-01 9.6749e-01 9.9989e-01 -9.9990e-01 9.9975e-01  
 11) -7.7317e-01 1.9580e-01 1.0000e+00 9.9995e-01 9.9574e-01 9.9993e-01 9.5135e-01 -9.9995e-01 9.9935e-01 -9.9955e-01  
 21) 9.9864e-01 9.9964e-01 -9.9866e-01 -9.9951e-01 9.6604e-01 9.9987e-01 -9.9989e-01 9.9987e-01 -7.6210e-01 1.8510e-02  
 31) -9.9989e-01 -9.9989e-01 9.9931e-01 9.9989e-01 9.9927e-01 9.9982e-01

Row 14, Columns 1 Thru 36

1) -9.9906e-01 9.9896e-01 9.9217e-01 9.9915e-01 -9.4255e-01 9.9898e-01 9.6537e-01 9.9975e-01 -9.9974e-01 9.9960e-01  
 11) -7.7935e-01 2.0415e-01 9.9995e-01 1.0000e+00 9.9651e-01 9.9995e-01 9.5402e-01 -1.0000e+00 9.9925e-01 -9.9941e-01  
 21) 9.9867e-01 9.9951e-01 -9.9867e-01 -9.9939e-01 9.6377e-01 9.9974e-01 -9.9975e-01 9.9973e-01 -7.5718e-01 2.8265e-02  
 31) -9.9974e-01 -9.9973e-01 9.9903e-01 9.9971e-01 9.9899e-01 9.9963e-01

Row 15, Columns 1 Thru 36

1) -9.9289e-01 9.9282e-01 9.8160e-01 9.9343e-01 -9.2212e-01 9.9290e-01 9.4835e-01 9.9490e-01 -9.9492e-01 9.9536e-01  
 11) -8.2077e-01 2.5665e-01 9.9574e-01 9.9651e-01 1.0000e+00 9.9660e-01 9.7569e-01 -9.9650e-01 9.9525e-01 -9.9505e-01  
 21) 9.9533e-01 9.9503e-01 -9.9515e-01 -9.9526e-01 9.4543e-01 9.9477e-01 -9.9518e-01 9.9500e-01 -7.2508e-01 9.7575e-02  
 31) -9.9459e-01 -9.9458e-01 9.9220e-01 9.9431e-01 9.9210e-01 9.9424e-01

Row 16, Columns 1 Thru 36

1) -9.9924e-01 9.9916e-01 9.9282e-01 9.9937e-01 -9.4442e-01 9.9918e-01 9.6691e-01 9.9980e-01 -9.9981e-01 9.9980e-01  
 11) -7.7680e-01 1.9830e-01 9.9993e-01 9.9995e-01 9.9660e-01 1.0000e+00 9.5435e-01 -9.9994e-01 9.9946e-01 -9.9960e-01  
 21) 9.9880e-01 9.9964e-01 -9.9879e-01 -9.9959e-01 9.6528e-01 9.9975e-01 -9.9986e-01 9.9981e-01 -7.6135e-01 2.4039e-02  
 31) -9.9970e-01 -9.9971e-01 9.9884e-01 9.9967e-01 9.9877e-01 9.9961e-01

Row 17, Columns 1 Thru 36

1) -9.4312e-01 9.4310e-01 9.2030e-01 9.4482e-01 -8.3595e-01 9.4331e-01 8.6941e-01 9.4872e-01 -9.4877e-01 9.5065e-01  
 11) -9.0551e-01 3.9512e-01 9.5135e-01 9.5402e-01 9.7569e-01 9.5435e-01 1.0000e+00 -9.5400e-01 9.5099e-01 -9.4993e-01  
 21) 9.5281e-01 9.4963e-01 -9.5217e-01 -9.5072e-01 8.6322e-01 9.4831e-01 -9.4967e-01 9.4907e-01 -6.0968e-01 2.8219e-01  
 31) -9.4782e-01 -9.4779e-01 9.4127e-01 9.4685e-01 9.4106e-01 9.4686e-01

Row 18, Columns 1 Thru 36

1) 9.9905e-01 -9.9895e-01 -9.9210e-01 -9.9913e-01 9.4234e-01 -9.9897e-01 -9.6530e-01 -9.9975e-01 9.9974e-01 -9.9958e-01  
 11) 7.7955e-01 -2.0440e-01 -9.9995e-01 -1.0000e+00 -9.9650e-01 -9.9994e-01 -9.5400e-01 1.0000e+00 -9.9925e-01 9.9940e-01  
 21) -9.9868e-01 -9.9951e-01 9.9869e-01 9.9939e-01 -9.6370e-01 -9.9974e-01 9.9974e-01 -9.9973e-01 7.5704e-01 -2.8583e-02  
 31) 9.9973e-01 9.9972e-01 -9.9902e-01 -9.9970e-01 -9.9899e-01 -9.9962e-01

Row 19, Columns 1 Thru 36

1) -9.9947e-01 9.9954e-01 9.9320e-01 9.9946e-01 -9.4538e-01 9.9957e-01 9.7294e-01 9.9967e-01 -9.9966e-01 9.9952e-01  
 11) -7.6740e-01 1.7371e-01 9.9935e-01 9.9925e-01 9.9525e-01 9.9946e-01 9.5099e-01 -9.9925e-01 1.0000e+00 -9.9997e-01  
 21) 9.9970e-01 9.9994e-01 -9.9971e-01 -9.9998e-01 9.7105e-01 9.9966e-01 -9.9967e-01 9.9966e-01 -7.7909e-01 0.0000e+00  
 31) -9.9897e-01 -9.9900e-01 9.9790e-01 9.9911e-01 9.9781e-01 9.9885e-01

Row 20, Columns 1 Thru 36

1) 9.9967e-01 -9.9971e-01 -9.9385e-01 -9.9968e-01 9.4702e-01 -9.9974e-01 -9.7297e-01 -9.9982e-01 9.9982e-01 -9.9969e-01  
 11) 7.6551e-01 -1.7391e-01 -9.9955e-01 -9.9941e-01 -9.9505e-01 -9.9960e-01 -9.4993e-01 9.9940e-01 -9.9997e-01 1.0000e+00  
 21) -9.9951e-01 -9.9998e-01 9.9953e-01 1.0000e+00 -9.7124e-01 -9.9980e-01 9.9983e-01 -9.9981e-01 7.7823e-01 0.0000e+00  
 31) 9.9927e-01 9.9930e-01 -9.9834e-01 -9.9939e-01 -9.9826e-01 -9.9918e-01

Row 21, Columns 1 Thru 36

1) -9.9854e-01 9.9867e-01 9.9044e-01 9.9843e-01 -9.3835e-01 9.9874e-01 9.7127e-01 9.9902e-01 -9.9895e-01 9.9856e-01  
 11) -7.7476e-01 1.7840e-01 9.9864e-01 9.9867e-01 9.9533e-01 9.9880e-01 9.5281e-01 -9.9868e-01 9.9970e-01 -9.9951e-01  
 21) 1.0000e+00 9.9953e-01 -1.0000e+00 -9.9957e-01 9.6897e-01 9.9908e-01 -9.9892e-01 9.9896e-01 -7.7756e-01 2.0543e-02  
 31) -9.9797e-01 -9.9799e-01 9.9670e-01 9.9818e-01 9.9663e-01 9.9774e-01

Row 22, Columns 1 Thru 36

1) -9.9967e-01 9.9969e-01 9.9359e-01 9.9965e-01 -9.4610e-01 9.9972e-01 9.7230e-01 9.9988e-01 -9.9987e-01 9.9967e-01  
 11) -7.6670e-01 1.7672e-01 9.9964e-01 9.9951e-01 9.9503e-01 9.9964e-01 9.4963e-01 -9.9951e-01 9.9994e-01 -9.9998e-01  
 21) 9.9953e-01 1.0000e+00 -9.9955e-01 -9.9997e-01 9.7060e-01 9.9988e-01 -9.9986e-01 9.9986e-01 -7.7626e-01 0.0000e+00  
 31) -9.9937e-01 -9.9939e-01 9.9854e-01 9.9950e-01 9.9846e-01 9.9927e-01

Row 23, Columns 1 Thru 36

1) 9.9860e-01 -9.9872e-01 -9.9057e-01 -9.9848e-01 9.3865e-01 -9.9879e-01 -9.7154e-01 -9.9905e-01 9.9898e-01 -9.9856e-01  
 11) 7.7381e-01 -1.7734e-01 -9.9866e-01 -9.9867e-01 -9.9515e-01 -9.9879e-01 -9.5217e-01 9.9869e-01 -9.9971e-01 9.9953e-01  
 21) -1.0000e+00 -9.9955e-01 1.0000e+00 9.9958e-01 -9.6927e-01 -9.9912e-01 9.9894e-01 -9.9899e-01 7.7812e-01 -1.9052e-02  
 31) 9.9801e-01 9.9802e-01 -9.9679e-01 -9.9823e-01 -9.9672e-01 -9.9779e-01

Row 24, Columns 1 Thru 36

1) 9.9959e-01 -9.9964e-01 -9.9359e-01 -9.9961e-01 9.4636e-01 -9.9967e-01 -9.7271e-01 -9.9978e-01 9.9978e-01 -9.9966e-01  
 11) 7.6684e-01 -1.7494e-01 -9.9951e-01 -9.9939e-01 -9.9526e-01 -9.9959e-01 -9.5072e-01 9.9939e-01 -9.9998e-01 1.0000e+00  
 21) -9.9957e-01 -9.9997e-01 9.9958e-01 1.0000e+00 -9.7091e-01 -9.9976e-01 9.9979e-01 -9.9977e-01 7.7785e-01 0.0000e+00  
 31) 9.9919e-01 9.9922e-01 -9.9819e-01 -9.9931e-01 -9.9810e-01 -9.9909e-01

Table (5.6)(Cont.) - The Cross Orthogonality of the First Krylov Block of the EPS Radiator

Row 25, Columns 1 Thru 36

1) -9.7430e-01 9.7493e-01 9.8418e-01 9.7366e-01 -9.7639e-01 9.7481e-01 9.9978e-01 9.6940e-01 -9.6951e-01 9.6944e-01  
 11) -5.9718e-01 -6.4138e-02 9.6604e-01 9.6377e-01 9.4543e-01 9.6528e-01 8.6322e-01 -9.6370e-01 9.7105e-01 -9.7124e-01  
 21) 9.6897e-01 9.7060e-01 -9.6927e-01 -9.7091e-01 1.0000e+00 9.6926e-01 -9.6937e-01 9.6930e-01 -9.0022e-01 -2.2244e-01  
 31) -9.6730e-01 -9.6753e-01 9.6765e-01 9.6825e-01 9.6734e-01 9.6783e-01

Row 26, Columns 1 Thru 36

1) -9.9972e-01 9.9968e-01 9.9381e-01 9.9968e-01 -9.4656e-01 9.9970e-01 9.7074e-01 9.9999e-01 -9.9998e-01 9.9972e-01  
 11) -7.6694e-01 1.8295e-01 9.9987e-01 9.9974e-01 9.9477e-01 9.9975e-01 9.4831e-01 -9.9974e-01 9.9966e-01 -9.9980e-01  
 21) 9.9908e-01 9.9988e-01 -9.9912e-01 -9.9976e-01 9.6926e-01 1.0000e+00 -9.9994e-01 9.9996e-01 -7.7091e-01 0.0000e+00  
 31) -9.9976e-01 -9.9976e-01 9.9922e-01 9.9984e-01 9.9917e-01 9.9968e-01

Row 27, Columns 1 Thru 36

1) 9.9975e-01 -9.9971e-01 -9.9433e-01 -9.9981e-01 9.4828e-01 -9.9972e-01 -9.7084e-01 -9.9997e-01 9.9999e-01 -9.9991e-01  
 11) 7.6661e-01 -1.8267e-01 -9.9989e-01 -9.9975e-01 -9.9518e-01 -9.9986e-01 -9.4967e-01 9.9974e-01 -9.9967e-01 9.9983e-01  
 21) -9.9892e-01 -9.9986e-01 9.9894e-01 9.9979e-01 -9.6937e-01 -9.9994e-01 1.0000e+00 -9.9997e-01 7.7105e-01 0.0000e+00  
 31) 9.9979e-01 9.9981e-01 -9.9911e-01 -9.9983e-01 -9.9904e-01 -9.9974e-01

Row 28, Columns 1 Thru 36

1) -9.9973e-01 9.9968e-01 9.9407e-01 9.9975e-01 -9.4744e-01 9.9969e-01 9.7077e-01 9.9996e-01 -9.9998e-01 9.9987e-01  
 11) -7.6683e-01 1.8287e-01 9.9987e-01 9.9973e-01 9.9500e-01 9.9981e-01 9.4907e-01 -9.9973e-01 9.9966e-01 -9.9981e-01  
 21) 9.9896e-01 9.9986e-01 -9.9899e-01 -9.9977e-01 9.6930e-01 9.9996e-01 -9.9997e-01 1.0000e+00 -7.7095e-01 0.0000e+00  
 31) -9.9977e-01 -9.9977e-01 9.9912e-01 9.9983e-01 9.9906e-01 9.9970e-01

Row 29, Columns 1 Thru 36

1) 7.8313e-01 -7.8530e-01 -8.1058e-01 -7.8143e-01 8.5363e-01 -7.8513e-01 -8.9989e-01 -7.7111e-01 7.7128e-01 -7.7124e-01  
 11) 2.5430e-01 4.7562e-01 -7.6210e-01 -7.5718e-01 -7.2508e-01 -7.6135e-01 -6.0968e-01 7.5704e-01 -7.7909e-01 7.7823e-01  
 21) -7.7756e-01 -7.7626e-01 7.7812e-01 7.7785e-01 -9.0022e-01 -7.7091e-01 7.7105e-01 -7.7095e-01 1.0000e+00 5.2942e-01  
 31) 7.6361e-01 7.6423e-01 -7.6291e-01 -7.6626e-01 -7.6212e-01 -7.6457e-01

Row 30, Columns 1 Thru 36

1) 1.3869e-02 -1.5068e-02 -8.9784e-02 -1.1353e-02 2.7095e-01 -1.4196e-02 -2.0921e-01 0.0000e+00 0.0000e+00 0.0000e+00  
 11) -6.4781e-01 9.0808e-01 1.8510e-02 2.8265e-02 9.7575e-02 2.4039e-02 2.8219e-01 -2.8583e-02 0.0000e+00 0.0000e+00  
 21) 2.0543e-02 0.0000e+00 -1.9052e-02 0.0000e+00 -2.2244e-01 0.0000e+00 0.0000e+00 0.0000e+00 5.2942e-01 1.0000e+00  
 31) 1.0000e+00 1.0000e+00 -9.9970e-01 -9.9998e-01 -9.9966e-01 -9.9999e-01

Row 31, Columns 1 Thru 36

1) 9.9951e-01 -9.9937e-01 -9.9442e-01 -9.9956e-01 9.4900e-01 -9.9937e-01 -9.6844e-01 -9.9979e-01 9.9981e-01 -9.9968e-01  
 11) 7.6653e-01 -1.9090e-01 -9.9989e-01 -9.9974e-01 -9.9459e-01 -9.9970e-01 -9.4782e-01 9.9973e-01 -9.9897e-01 9.9927e-01  
 21) -9.9797e-01 -9.9937e-01 9.9801e-01 9.9919e-01 -9.6730e-01 -9.9976e-01 9.9979e-01 -9.9977e-01 7.6361e-01 0.0000e+00  
 31) 1.0000e+00 1.0000e+00 -9.9970e-01 -9.9998e-01 -9.9966e-01 -9.9999e-01

Row 32, Columns 1 Thru 36

1) 9.9954e-01 -9.9941e-01 -9.9452e-01 -9.9959e-01 9.4928e-01 -9.9940e-01 -9.6868e-01 -9.9980e-01 9.9982e-01 -9.9970e-01  
 11) 7.6609e-01 -1.9001e-01 -9.9989e-01 -9.9973e-01 -9.9458e-01 -9.9971e-01 -9.4779e-01 9.9972e-01 -9.9900e-01 9.9930e-01  
 21) -9.9799e-01 -9.9939e-01 9.9802e-01 9.9922e-01 -9.6753e-01 -9.9976e-01 9.9981e-01 -9.9977e-01 7.6423e-01 0.0000e+00  
 31) 1.0000e+00 1.0000e+00 -9.9969e-01 -9.9998e-01 -9.9964e-01 -9.9999e-01

Row 33, Columns 1 Thru 36

1) -9.9908e-01 9.9888e-01 9.9475e-01 9.9901e-01 -9.5089e-01 9.9886e-01 9.6838e-01 9.9922e-01 -9.9921e-01 9.9885e-01  
 11) -7.5886e-01 1.8765e-01 9.9931e-01 9.9903e-01 9.9220e-01 9.9884e-01 9.4127e-01 -9.9902e-01 9.9790e-01 -9.9834e-01  
 21) 9.9670e-01 9.9854e-01 -9.9679e-01 -9.9819e-01 9.6765e-01 9.9922e-01 -9.9911e-01 9.9912e-01 -7.6291e-01 0.0000e+00  
 31) -9.9970e-01 -9.9969e-01 1.0000e+00 9.9971e-01 1.0000e+00 9.9975e-01

Row 34, Columns 1 Thru 36

1) -9.9963e-01 9.9951e-01 9.9450e-01 9.9963e-01 -9.4899e-01 9.9951e-01 9.6942e-01 9.9986e-01 -9.9987e-01 9.9967e-01  
 11) -7.6493e-01 1.8719e-01 9.9989e-01 9.9971e-01 9.9431e-01 9.9967e-01 9.4685e-01 -9.9970e-01 9.9911e-01 -9.9939e-01  
 21) 9.9818e-01 9.9950e-01 -9.9823e-01 -9.9931e-01 9.6825e-01 9.9984e-01 -9.9983e-01 9.9983e-01 -7.6626e-01 0.0000e+00  
 31) -9.9998e-01 -9.9998e-01 9.9971e-01 1.0000e+00 9.9967e-01 9.9997e-01

Row 35, Columns 1 Thru 36

1) -9.9900e-01 9.9879e-01 9.9456e-01 9.9892e-01 -9.5045e-01 9.9878e-01 9.6805e-01 9.9916e-01 -9.9915e-01 9.9876e-01  
 11) -7.5924e-01 1.8867e-01 9.9927e-01 9.9899e-01 9.9210e-01 9.9877e-01 9.4106e-01 -9.9899e-01 9.9781e-01 -9.9826e-01  
 21) 9.9663e-01 9.9846e-01 -9.9672e-01 -9.9810e-01 9.6734e-01 9.9917e-01 -9.9904e-01 9.9906e-01 -7.6212e-01 0.0000e+00  
 31) -9.9966e-01 -9.9964e-01 1.0000e+00 9.9967e-01 1.0000e+00 9.9971e-01

Row 36, Columns 1 Thru 36

1) -9.9952e-01 9.9938e-01 9.9483e-01 9.9958e-01 -9.5032e-01 9.9937e-01 9.6889e-01 9.9973e-01 -9.9975e-01 9.9965e-01  
 11) -7.6421e-01 1.8869e-01 9.9982e-01 9.9963e-01 9.9424e-01 9.9961e-01 9.4686e-01 -9.9962e-01 9.9885e-01 -9.9918e-01  
 21) 9.9774e-01 9.9927e-01 -9.9779e-01 -9.9909e-01 9.6783e-01 9.9968e-01 -9.9974e-01 9.9970e-01 -7.6457e-01 0.0000e+00  
 31) -9.9999e-01 -9.9999e-01 9.9975e-01 9.9997e-01 9.9971e-01 1.0000e+00

Table (5.6)(Cont.) - The Cross Orthogonality of the First Krylov Block of the EPS Radiator

Row 1, Columns 1 Thru 8

1.0000e+00 9.5350e-01 -9.7551e-01 7.5250e-01 -1.6221e-01 -9.4312e-01 7.8313e-01 1.3869e-02

Row 2, Columns 1 Thru 8

9.5350e-01 1.0000e+00 -9.7285e-01 5.4647e-01 5.0223e-02 -8.3595e-01 8.5363e-01 2.7095e-01

Row 3, Columns 1 Thru 8

-9.7551e-01 -9.7285e-01 1.0000e+00 -6.0711e-01 -5.8122e-02 8.6941e-01 -8.9989e-01 -2.0921e-01

Row 4, Columns 1 Thru 8

7.5250e-01 5.4647e-01 -6.0711e-01 1.0000e+00 -7.1435e-01 -9.0551e-01 2.5430e-01 -6.4781e-01

Row 5, Columns 1 Thru 8

-1.6221e-01 5.0223e-02 -5.8122e-02 -7.1435e-01 1.0000e+00 3.9512e-01 4.7562e-01 9.0808e-01

Row 6, Columns 1 Thru 8

-9.4312e-01 -8.3595e-01 8.6941e-01 -9.0551e-01 3.9512e-01 1.0000e+00 -6.0968e-01 2.8219e-01

Row 7, Columns 1 Thru 8

7.8313e-01 8.5363e-01 -8.9989e-01 2.5430e-01 4.7562e-01 -6.0968e-01 1.0000e+00 5.2942e-01

Row 8, Columns 1 Thru 8

1.3869e-02 2.7095e-01 -2.0921e-01 -6.4781e-01 9.0808e-01 2.8219e-01 5.2942e-01 1.0000e+00

Table (5.7) - The Cross Orthogonality of the First Krylov Block  
of the EPS Radiator, After Block Filtering

Modes	Finite Element	Ritz Vectors	Percent Difference	Normal Modes	Percent Difference
1st Bend	.5464 Hz	.5464 Hz	0. %	.5464 Hz	0. %
2nd Bend	3.415 Hz	3.415 Hz	0. %	3.415 Hz	0. %
3rd Bend	9.522 Hz	9.610 Hz	.92 %	9.522 Hz	0. %
4th Bend	18.57 Hz	24.24 Hz	31. %	18.57 Hz	0. %
1st Tors	27.58 Hz	27.58 Hz	0. %	27.58 Hz	0. %

Table (5.8) - Comparison Between the Frequencies of a Cantilevered Beam Obtained From Fixed Interface Component Normal Modes, Ritz Vectors, and Finite Elements

Modes	Finite Element	Ritz Vectors	Percent Difference
1st Bend	.5464 Hz	.5464 Hz	0. %
2nd Bend	3.415 Hz	3.415 Hz	0. %
3rd Bend	9.522 Hz	9.524 Hz	.02 %
4th Bend	18.57 Hz	19.23 Hz	3.6 %
1st Tors	27.58 Hz	27.58 Hz	0. %

Table (5.9) - Comparison Between the Frequencies of a Cantilevered Beam Obtained From Free Interface Component Ritz Vectors, and Finite Elements

Modes	Finite Element	Ritz Vectors	Percent Difference
1st Rigid	0. Hz	0. Hz	0. %
2nd Rigid	0. Hz	0. Hz	0. %
3rd Rigid	0. Hz	0. Hz	0. %
4th Rigid	0. Hz	0. Hz	0. %
5th Rigid	0. Hz	0. Hz	0. %
6th Rigid	0. Hz	0. Hz	0. %
1st Bend	3.474 Hz	3.474 Hz	0. %
2nd Bend	9.549 Hz	9.549 Hz	0. %
3rd Bend	18.65 Hz	18.69 Hz	.21 %
4th Bend	30.70 Hz	33.15 Hz	8. %

Table (5.10) - Comparison Between the Frequencies of a Free-Free Beam Obtained From Free Interface Component Ritz Vectors, and Finite Elements

## Various Normalization Methods

Various Orthogonalization Methods	Orthonormalized wrt the Euclidean Norm	Orthonormalized wrt the Mass Matrix, and all previous blocks	Orthonormalized wrt the Mass Matrix, previous two blocks only
Cholesky/QR Decomposition	97.0	140.9	112.4
Gram-Schmidt, Reorthogonalize with all vectors	103.9	379.2	385.6
Gram-Schmidt, Reorthogonalize with selected	116.0	409.1	468.9
Gram-Schmidt, Reorthogonalize using modified Gram-Schmidt	461.0	3747.0	4196.9

## Various Eigenvalue Extraction Methods

Normal Eigenvalues	Lanczos	Modified Givens	Inverse Power
	115.5	5758.8	6455.5

Table (5.11) - Comparisons of the Computer Time (seconds) Required to Form a Component of the EPS Radiator Finite Element Model (4000 DOF), Using Selected Orthonormalization Options

## Various Normalization Methods

Various Orthogonalization Methods	Orthonormalized wrt the Euclidean Norm	Orthonormalized wrt the Mass Matrix, and all previous blocks	Orthonormalized wrt the Mass Matrix, previous two blocks only
Cholesky/QR Decomposition	323.2	399.5	376.1
Gram-Schmidt, Reorthogonalize with all vectors	341.6	534.3	542.1
Gram-Schmidt, Reorthogonalize with selected	348.5	558.3	565.2
Gram-Schmidt, Reorthogonalize using modified Gram-Schmidt	559.5	3062.5	3095.3

## Various Eigenvalue Extraction Methods

Normal Eigenvalues	Lanczos	Modified Givens	Inverse Power
	743.7		

Table (5.12) - Comparisons of the Computer Time Required to Form a Component of the Cassini Spacecraft Finite Element Model (11,100 DOF), Using Selected Orthonormalization Options

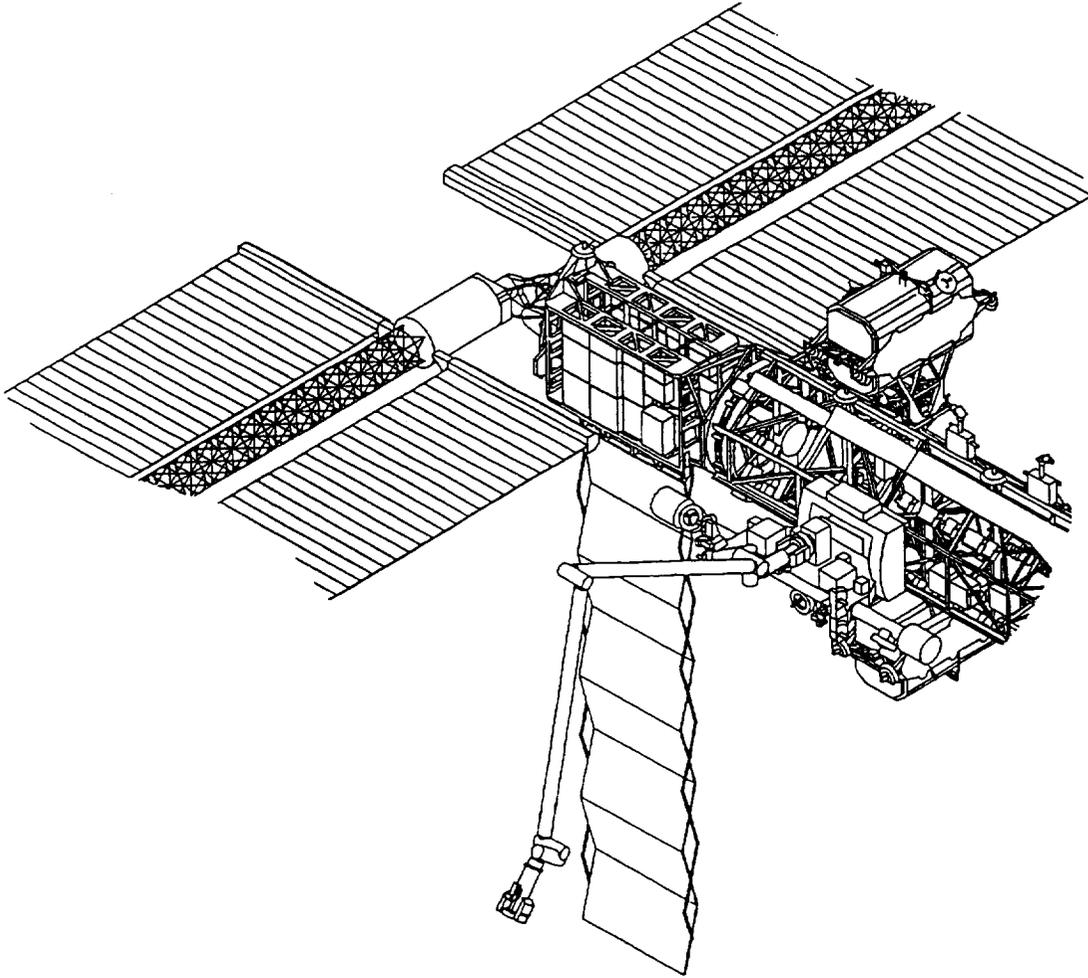


Figure (5.1) - Space Station Electrical Power System Deployed Radiator

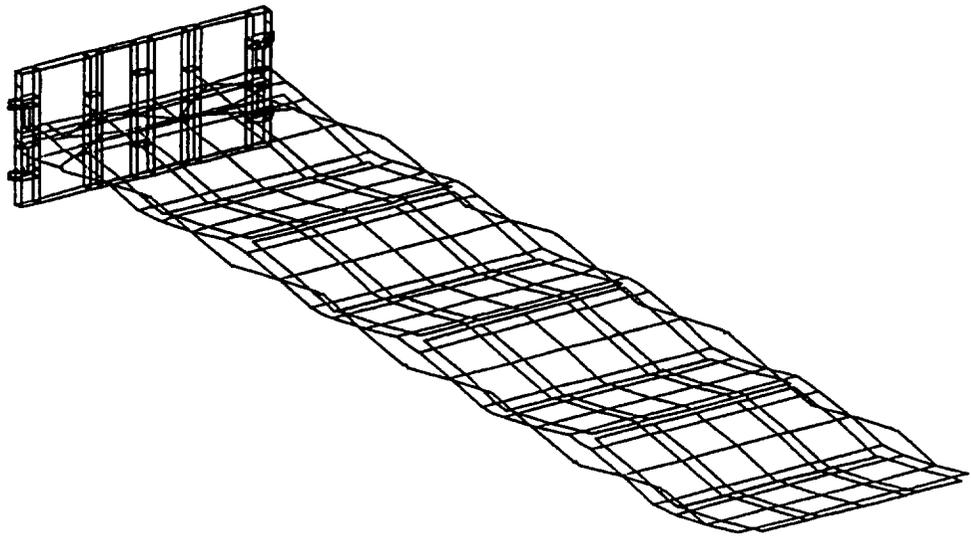


Figure (5.2) - Space Station EPS Radiator Undeformed Finite Element Model

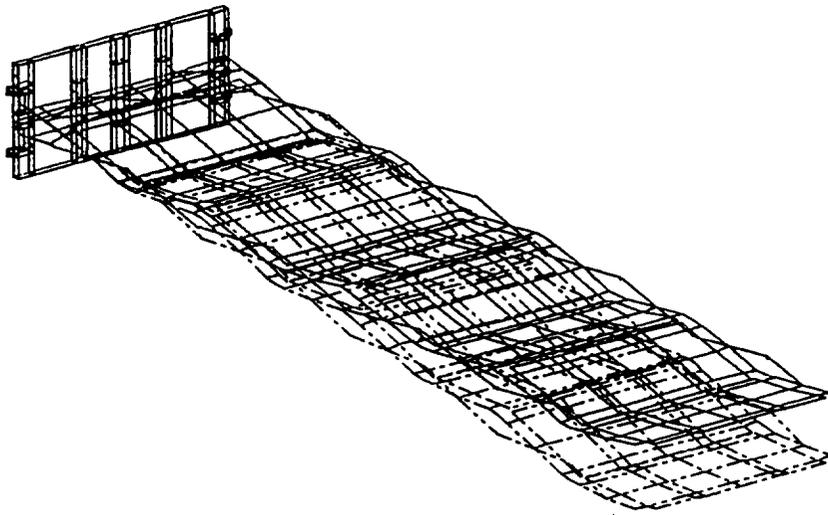


Figure (5.3) - EPS Radiator First Normal Mode, Fixed Interface

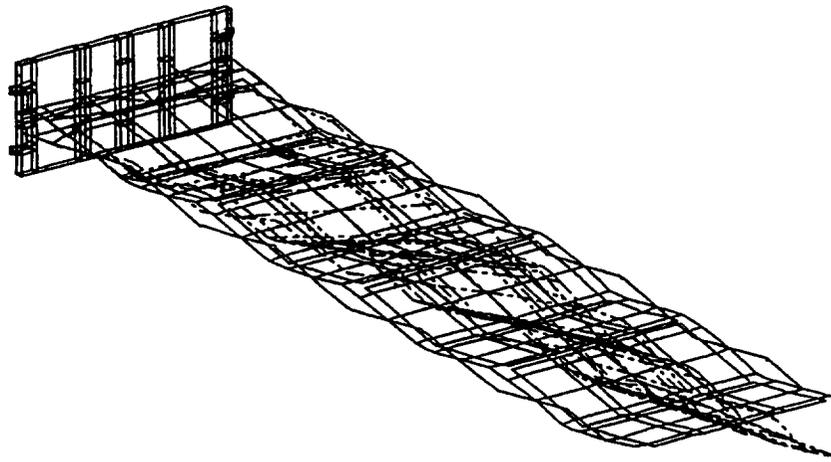


Figure (5.4) - EPS Radiator Second Normal Mode, Fixed Interface

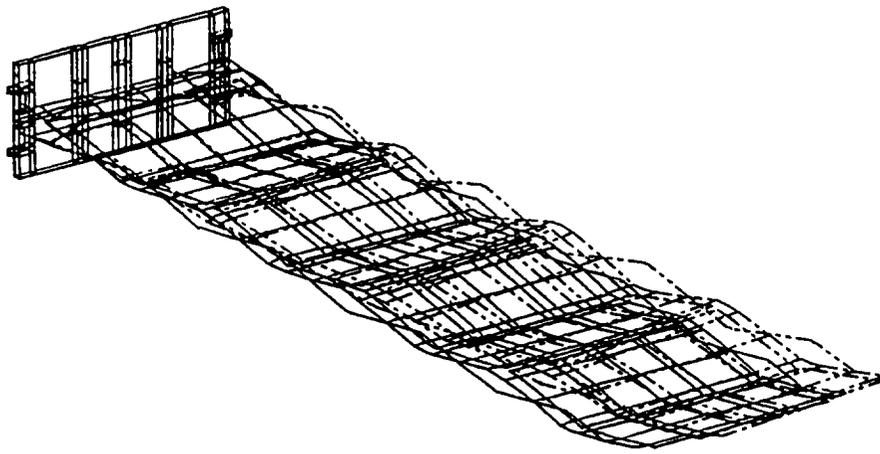


Figure (5.5) - EPS Radiator Third Normal Mode, Fixed Interface

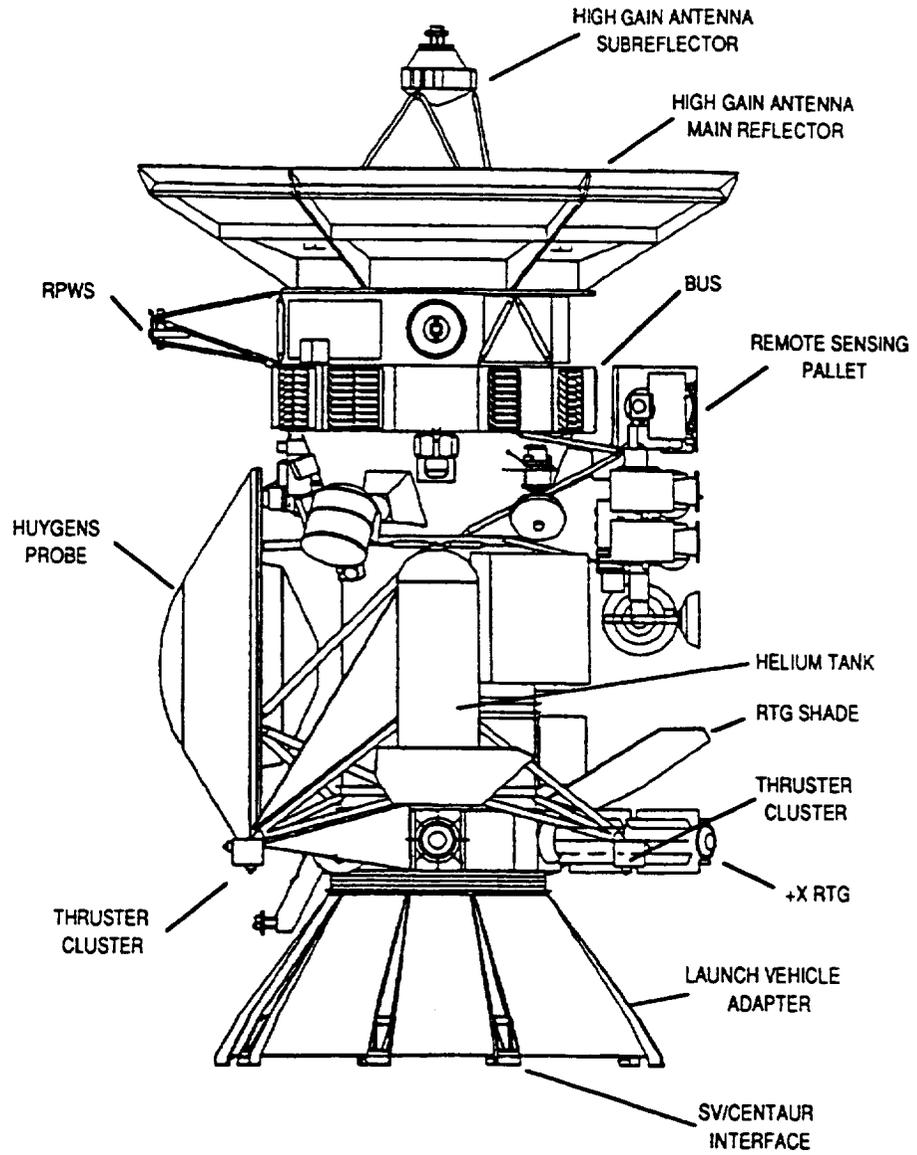


Figure (5.6) - Cassini Spacecraft

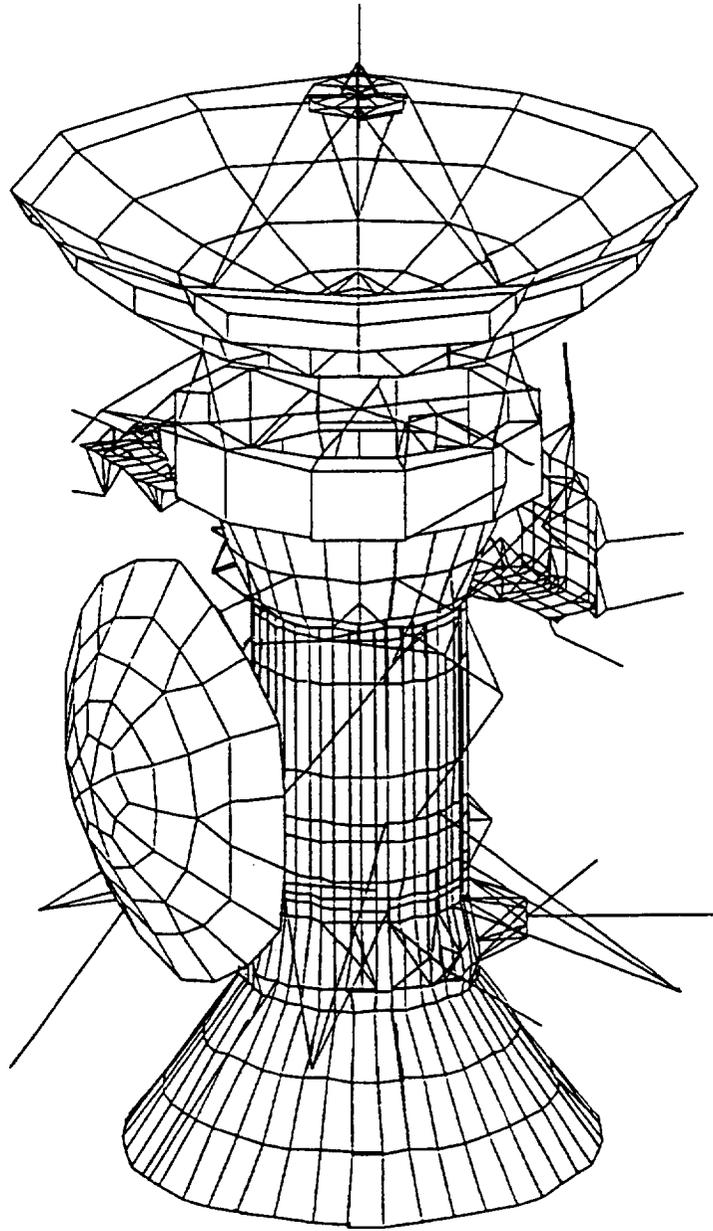
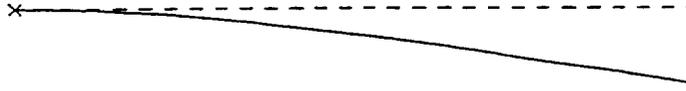


Figure (5.7) - Cassini Spacecraft Finite Element Model

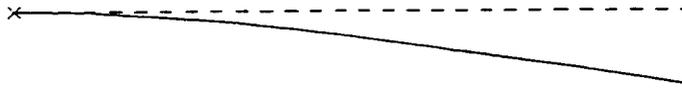


(a) Initial Calculation

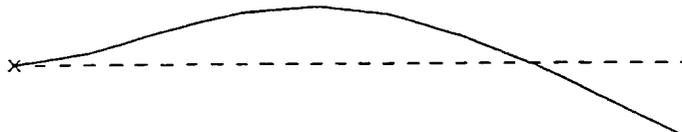


(b) After Gram-Schmidt Orthogonalization

Figure (5.8) - Fixed Interface Ritz Vector,  
Block One, Vector One

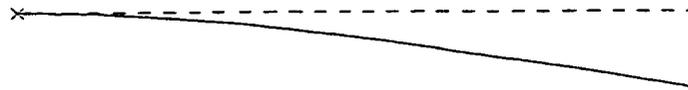


(a) Initial Calculation

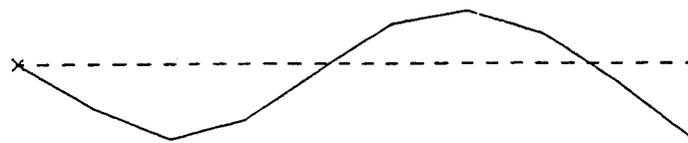


(b) After Gram-Schmidt Orthogonalization

Figure (5.9) - Fixed Interface Ritz Vector,  
Block One, Vector Two



(a) Initial Calculation

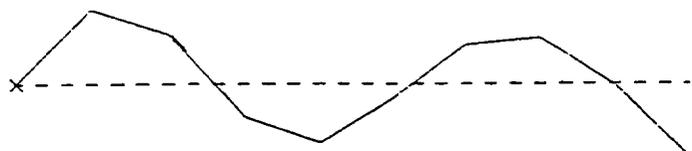


(b) After Gram-Schmidt Orthogonalization

Figure (5.10) - Fixed Interface Ritz Vector,  
Block Two, Vector One



(a) Initial Calculation



(b) After Gram-Schmidt Orthogonalization

Figure (5.11) - Fixed Interface Ritz Vector,  
Block Two, Vector Two

## Chapter 6

### Targeted Shifting

#### 6.1) Introduction:

Inverse operators have been widely used in a variety of eigenvalue extraction routines. The use of inverse operators is commonly called sequence shifting. By shifting the dominant frequency sought by an algorithm, an eigenvalue extraction routine can locate missing eigenvalues within a specific frequency range. In Lanczos eigenvalue extraction, the inverse operator shifts the dominant frequency of the Krylov sequence to a range of frequency where missing eigenvalues are located. By doing so, the number of eigenvalues beyond the range of interest is minimized while insuring that interesting modes are calculated.

The utility made of the ability of the inverse operator to shift the dominant frequency of the Krylov sequence is not limited to searching for missing eigenvalues. This chapter presents shifting that is targeted to the frequency of the dominant applied dynamic load vector, rather than to missing eigenvalues. By shifting the dominant frequency sought by the Krylov sequence, static Ritz vectors which can contribute to an accurate dynamic response prediction are generated. The use of the inverse operator in this manner has been identified in this work as targeted shifting.

6.2) Theory:

As discussed in section 2.5, an inverse operator can direct a solution to particular frequency range. The problem discussed in section 2.5 was the computation of selected eigenpairs of the generalized eigenvalue problem,

$$(\mathbf{K} - \lambda \mathbf{M}) \mathbf{x} = 0 \quad (6.1)$$

with  $\lambda$  being the eigenvalues and  $\mathbf{x}$  the eigenvectors, the matrix  $\mathbf{A}$  of equation (2.46) being re-written to the common structural dynamics usage of  $\mathbf{K}$ . A system is created using the operator

$$(\mathbf{K} - \sigma \mathbf{M})^{-1} \mathbf{M} \quad (6.2)$$

which has the same eigenvectors as (6.1). The shifted system eigenvalues are transformed to  $1/(\lambda - \sigma)$ , with  $\sigma$  being defined as the shift frequency. When the inverse operator is applied the eigenvalue nearest  $\sigma$  becomes the dominant one and the sequence will converge to the corresponding eigenvector. The cost for performing the shift is the factorization of  $(\mathbf{K} - \sigma \mathbf{M})^{-1}$ .

If the selected shift frequency,  $\sigma$ , is coincidentally an exact eigenvalue, then  $(\mathbf{K} - \sigma \mathbf{M})^{-1}$  is, by definition, singular. To obtain a non-singular matrix the shift value is merely altered by a small constant value, such as .1, to move the shift away from the eigenvalue frequency.

To add an inverse operator to the static Ritz vector calculation the Krylov sequence presented in chapter 2 must be modified. Several different inverse operators could be used. Some inverse operators, such as that defined by equation (6.2), transform the system equations so that they produce transformed eigenvalues, but the produced eigenvectors are not transformed. Others inverse operators transform the system equations such that both the eigenvalues and eigenvectors are transformed. (In eigenvalue extraction routines this transformation necessitates a back transformation to obtain a final solution.) The operator defined by equation (6.2) was selected for this work because this operator does not transform vectors resulting from the Krylov sequence when shifting is applied.

When this operator is applied to the Krylov sequence presented in Table (4.1) and chapter 2, equation (2.8),

$$\mathbf{q}_j^* = \mathbf{k}_{ii}^{-1} \mathbf{m}_{ii} \mathbf{q}_{j-1} \quad (2.8)$$

is revised to become

$$\mathbf{q}_j^* = (\mathbf{k}_{ii} - \sigma \mathbf{m}_{ii})^{-1} \mathbf{m}_{ii} \mathbf{q}_{j-1} \quad (6.3)$$

Note that if a shift frequency is defined as zero, then equation (6.3) becomes equation (2.8). The standard Krylov sequence can be considered to be a shifted sequence with the shift frequency permanently defined as zero.

If the inverse operator is applied to the equation used to generate the initial block of Ritz vectors, equation (2.6)

$$\mathbf{q}_1^{**} = \mathbf{k}_{ii}^{-1}(\mathbf{m}_{ii} \Phi_{ic} + \mathbf{m}_{ic}) \quad (2.6)$$

would be revised to become

$$\mathbf{q}_1^{**} = (\mathbf{k}_{ii} - \sigma \mathbf{m}_{ii})^{-1} \mathbf{m}_{ii} (\mathbf{m}_{ii} \Phi_{ic} + \mathbf{m}_{ic}) \quad (6.4)$$

### 6.3) Illustrative Example:

A short numerical example is given to illustrate the effectiveness of shifting.

Consider a system with the following defining matrices,

$$\mathbf{K} = \begin{bmatrix} 4 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad (6.5)$$

and

$$\mathbf{M} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.6)$$

In this example, a load, rather than the boundary flexibility matrix, will be used to initiate the sequence. The initial load dependent static Ritz vector is defined by the following equation

$$\mathbf{q}_1^{**} = \mathbf{K}^{-1} \mathbf{p} \quad (6.7)$$

If  $\mathbf{p}$  is defined as

$$\mathbf{p} = [1 \ 1 \ 1]^T \quad (6.8)$$

then the normalized initial Ritz vector,  $\mathbf{q}_1$ , is

$$\mathbf{q}_1 = [.408 \ .408 \ .816]^T \quad (6.9)$$

The Rayleigh quotient of a vector derived from a two matrix, or general system, can be defined as

$$\rho_i = \frac{\mathbf{q}_i^T \mathbf{K} \mathbf{q}_i}{\mathbf{q}_i^T \mathbf{M} \mathbf{q}_i} \quad (6.10)$$

When the vector used with equation (6.10) is an eigenvector, then the Rayleigh quotient is the eigenvalue associated with that eigenvector. The square root of a vector's Rayleigh quotient implies, as a eigenvalue does, a frequency at which the vector is most likely to respond. The initial static Ritz vector,  $\mathbf{q}_1$ , has a Rayleigh quotient of  $\rho = 2.857$ .

The eigenvalues of the system defined in equations (6.5) and (6.6) are .775, 2., and 3.225. The system eigenvectors,  $\phi$ , are

$$\phi = \begin{bmatrix} -.309 & .707 & .312 \\ .756 & 0.000 & .764 \\ -.617 & -.707 & .624 \end{bmatrix} \quad (6.11)$$

The cross orthogonality of the third eigenvector and the first Ritz vector,  $\phi_3^T \mathbf{q}_1$ , is .948. The initial static Ritz vector, produced by equation (6.8), is most numerically similar to the third eigenvector, both by mode and by frequency.

The shifted initial load dependent static Ritz vector is defined by the following equation

$$\mathbf{q}_1^{**} = (\mathbf{K} - \sigma \mathbf{M})^{-1} \mathbf{M} \mathbf{p} \quad (6.12)$$

Using the same force vector,  $\mathbf{p}$ , defined by equation (6.8), and a shift with an arbitrary frequency of 1.0, (an assumption being made that the force vector,  $\mathbf{p}$ , has a dominant frequency component of 1.0) the normalized initial Ritz vector,  $\mathbf{q}_1$ , is

$$\mathbf{q}_1 = [0.000 \quad .894 \quad -.447]^T \quad (6.13)$$

The Rayleigh quotient of this vector is  $\rho = 1.199$ , relatively close to the shift value of 1.0. The cross orthogonality of the first eigenvector and the first Ritz vector,  $\phi_1^T \mathbf{q}_1$ , is .952. The shifted initial static Ritz vector's frequency content is closer to a target frequency than that of the vector (equation 6.9) produced without the shift. Rather than resembling the third eigenvector of the system, the shifted initial Ritz vector resembles the first eigenvector.

#### 6.4) Targeted Shifting:

After considering the example presented in section 6.3, it is reasonable to consider using the frequency content of an applied dynamic loading in the shift

methodology. This can be accomplished by determining the frequency content of the applied dynamic load and using the dominant frequencies as shift values. The resulting shifted static Ritz vectors will be more likely to accurately predict system responses than static Ritz vectors which have a frequency content far removed from that of the applied dynamic load. In addition, it is also reasonable to augment the static Ritz vectors derived from the boundary flexibility matrices with static load dependent Ritz vectors.

Each separate targeted shift that is performed entails the computation cost of decomposing  $(\mathbf{k}_{ii} - \sigma \mathbf{m}_{ii})^{-1}$  into the upper and lower triangular factors  $U$  and  $L$ . In a large finite element model the computational cost to perform this factorization can be significant and over-aggressive shifting can make the computational cost of this procedure exceed that of the orthonormalization and make the static Ritz vector calculation inefficient. As a result it is beneficial to minimize the amount of shifts in the overall targeted shifting strategy. The overall targeted shifting strategy will be discussed in the next section.

Since relatively few frequencies can be targeted in a computationally efficient shift strategy, the method chosen to select frequencies important to the dynamic response should seek, at most, several dominant frequencies. The absolute magnitude of any particular frequency is unimportant since the relative magnitude of a frequency is used to identify the dominant frequencies. The choice of which method is used to determine the frequency content of the time domain dynamic loading can be made

somewhat subjectively. Most analytical procedures which transform a time domain response into the frequency domain, such as calculating a power spectral density, or a frequency response function, etc. would be sufficient for identifying several dominant frequencies. Alternately, if frequency information is available describing the dynamic environment from non-analytical experience, then that information can also easily be used. A shift can also easily be targeted to test derived frequency information describing the structure itself. There is, of course, a possibility in some applications, due to the nature of the dynamic excitation, that no dominant frequencies can be identified. In those circumstances, the unshifted Krylov sequence is sufficient.

Static Ritz vectors derived from the spatial distribution of the applied dynamic load, in conjunction with targeted shifting, can augment the static Ritz vectors derived from the boundary flexibility matrices. A load dependent static Ritz vector may be calculated from a dynamic load by using the spatial distribution of the dynamic load at a single time step to create a representative static load. Selecting a time step where the applied dynamic load, at a particular node, is at a maximum, is one possible criteria. Consideration should be given, in the time step selection, to the dominant frequency chosen for shifting. Another possible method for creating a representative static load derived from the applied dynamic load would be to, determine the maximum applied load over all time steps for each node, and from these maximums synthesize a single static force vector. This representative static vector may have no physical relation to the dynamic spatial characteristics of the applied loading, and the

resulting static Ritz vector may not resemble a response to dynamic excitation. Once a representative static load has been created, static Ritz vectors may be calculated by using equation (6.12) with the selected dominant frequency. This load dependent static Ritz vector is appended to the vectors calculated by the algorithm given in Tables (4.1) and (4.2).

### 6.5) Shift Strategy:

A definite theory does not exist which describes an optimal ordering and value of shifts in Krylov sequence methodologies. In Lanczos eigenvalue extraction a fairly complicated heuristic approach, which is successful in extracting eigenvalues, has been developed<sup>13</sup>. That particular strategy has limited applicability to the creation of boundary flexibility method static Ritz vectors. As a result, an alternate heuristic strategy has been developed specifically to create accurate static Ritz vectors as early in the Krylov sequence as possible.

The new shift strategy for static Ritz vectors is introduced as follows. No shift (or equivalently, a shift of zero) is applied to the initial block of vectors. The representative static load, derived from applied the dynamic loading, is also not applied to calculate the initial block. Therefore, equation (2.6), as previously presented, is used to generate the initial block of Ritz vectors

$$\mathbf{q}_1^{**} = \mathbf{k}_{ii}^{-1} (\mathbf{m}_{ii} \Phi_{ic} + \mathbf{m}_{ic}) \quad (2.6)$$

The non-zero shift equation, (6.4), is not used in the computer implementation of this work. The initial zero shift is performed because a basic set of vectors targeted to the lowest frequencies is desired. These vectors are desired because, with only a limited amount of shifts targeting the several selected dominant frequencies, they describe other important dynamic characteristics not in the frequency spectrum of the targeted shifts.

Two selected dominant frequencies are allowed, and a cutoff frequency is also used as a shift frequency, for a total of three non-zero targeted shifts. When approximately one quarter of the total required vectors have been calculated (the next chapter presents a method to determine this value) a shift targeted to the first selected frequency is performed. At this shift the representative static load, derived from the applied dynamic loading, is also used by appending it to the previous vector block. The resulting modified Krylov sequence equation (6.3) becomes

$$\mathbf{q}_j^* = (\mathbf{k}_{\ddot{u}} - \sigma_k \mathbf{m}_{\ddot{u}})^{-1} \mathbf{m}_{\ddot{u}} [ \mathbf{q}_{j-1} \quad \mathbf{p}_k ] \quad (6.14)$$

with the addition of the representative static load,  $\mathbf{p}_k$ , and with the subscript  $k$  indicating the shift number, initially  $k = 1$ . The vector block,  $\mathbf{q}_j^*$ , will have one more column in it than the vector block,  $\mathbf{q}_{j-1}$ , due to the augmentation of  $\mathbf{q}_{j-1}$  with the representative static load,  $\mathbf{p}_k$ . Subsequent to the block increment,  $j$ , at which the shift is applied, equation (6.3) is used to calculate additional vector blocks. When approximately one half of the total required vectors have been calculated a second

shift,  $k = 2$ , is applied with the second selected target frequency. The third shift is applied when approximately three quarters of the vectors required have been generated. The frequency used for the shift is a user defined cutoff frequency which will be discussed in the next chapter.

This strategy was developed through trial and error and is as a result heuristic. There may be other strategies which would be successful at creating a set of static Ritz vectors which accurately represent a substructure. The effectiveness of the presented strategy will not be demonstrated until chapter 8, after a sequence termination methodology has also been presented.

#### 6.6) Applied Dynamic Loading of Example Structural Models:

Several examples applied to the structural examples given in chapter 5 are presented. For the beam example the time domain loading is shown in Figure (6.1). The response spectrum of this time domain load is shown in Figure (6.2). The dominant frequency selected is the frequency where the response spectrum reaches its peak, at .546 HZ. Because the time domain dynamic load is a rectangular impulse, this frequency is equal to the inverse of twice the impulse width. The dynamic loading is applied to a single node, the end point. As a result, the maximum spatial load for the structure is the maximum applied dynamic load at that node and a representative static load is easily created.

For the EPS Radiator the spatial distribution of the dynamic loading is the same at every time step that the load is applied. As a result, all choices for an

appropriate time step from which to derive a representative static load are identical. The load is applied in a series of rectangular impulses as shown in Figure (6.3). The response spectrum of this applied dynamic load is shown in Figure (6.4). The peak of the response spectrum, at a frequency of approximately 5 HZ, is a result of the pulse width of .08 sec of the applied dynamic load. This applied dynamic loading is derived from a plume impingement loading event. Plume impingement loading occurs when the space shuttle approaches the space station, directing its reaction control system jets toward the station. It is the critical loading event for the space station structure in its on-orbit configuration. The specific loading case shown in Figure (6.3) is a result of the shuttle translating in yaw along the main station axis.

All Cassini spacecraft loading comes through the spacecraft/launch vehicle interface. As a result, the boundary flexibility vectors are the only available loading vectors. (A separate applied dynamic load vector does not exist.) Figures (6.5), (6.6), and (6.7) show the interface acceleration for a liftoff loading event in the three translational directions. Response spectra of this inertia loading are shown in Figures (6.8), (6.9), and (6.10). Dominant frequencies of 5 HZ and 13 HZ were selected for the two targeted shifts. The 13 HZ frequency was derived from the pitch and yaw excitation response spectrum and the 5 HZ frequency was selected from the longitudinal excitation response spectrum. No representative static load was appended, because of the lack of a spatial distribution for the applied dynamic load.

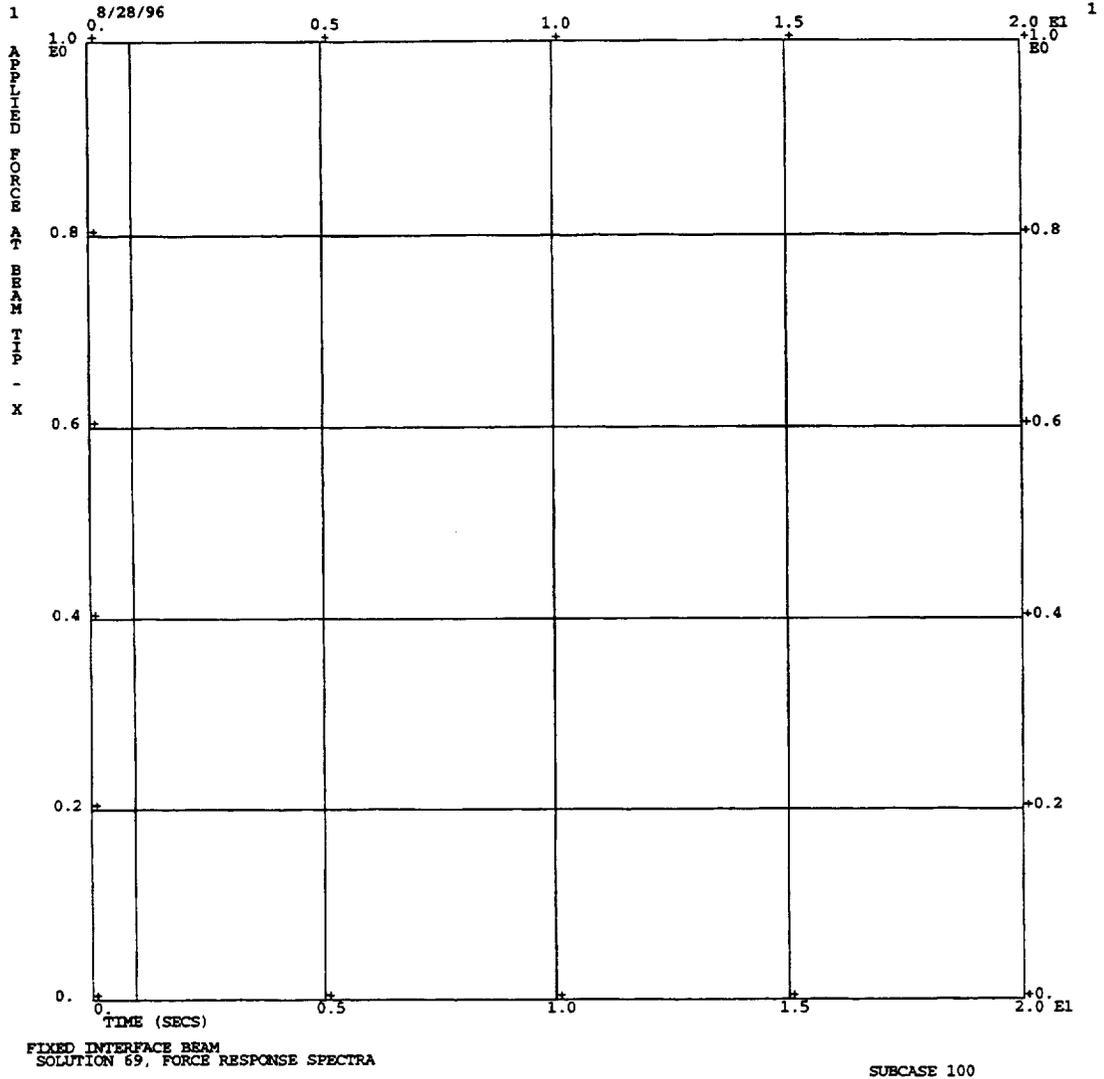


Figure (6.1) - Time Domain Beam Model Applied Dynamic Loading

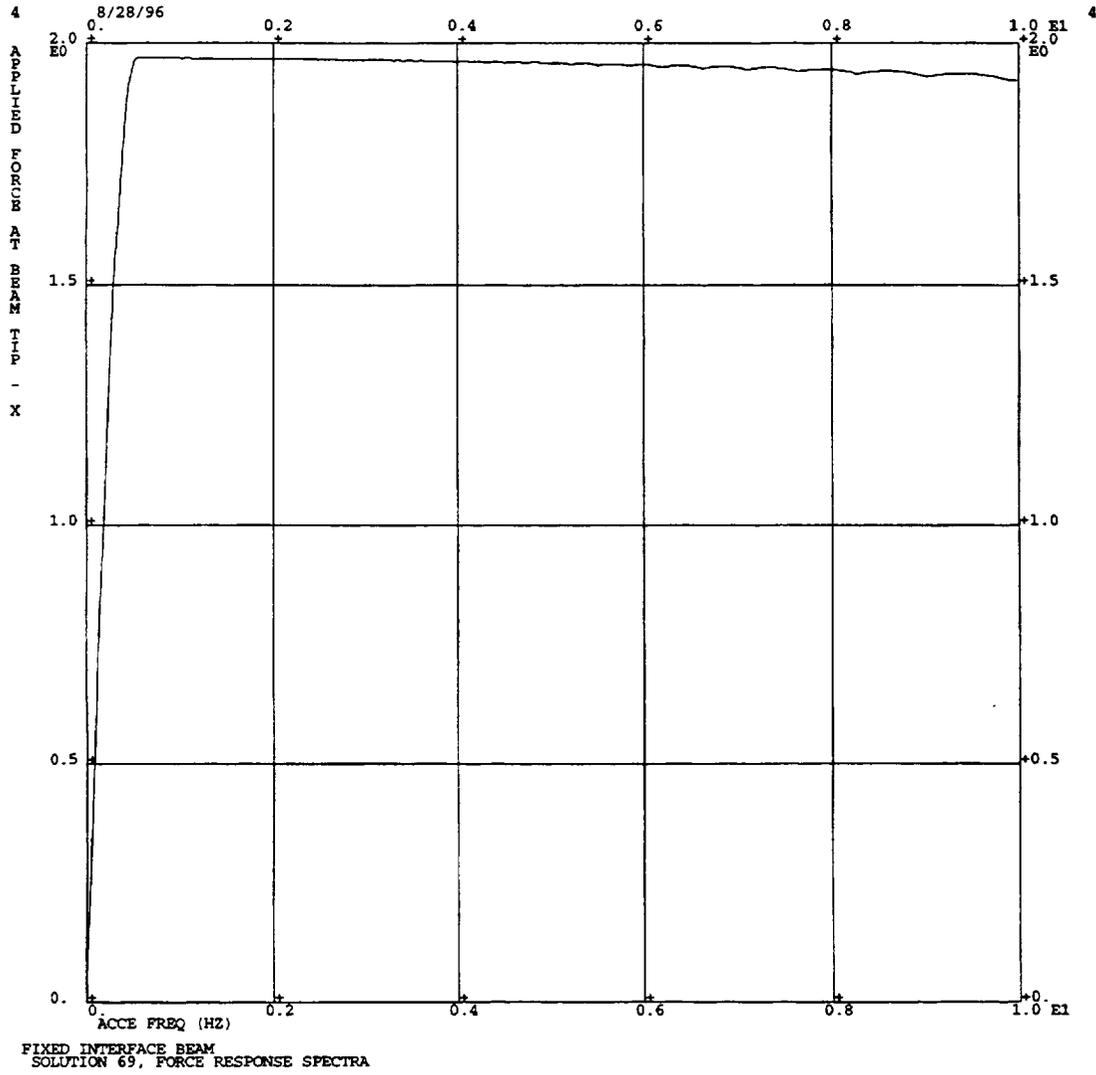


Figure (6.2) - Response Spectrum of the Beam Model Applied Dynamic Loading

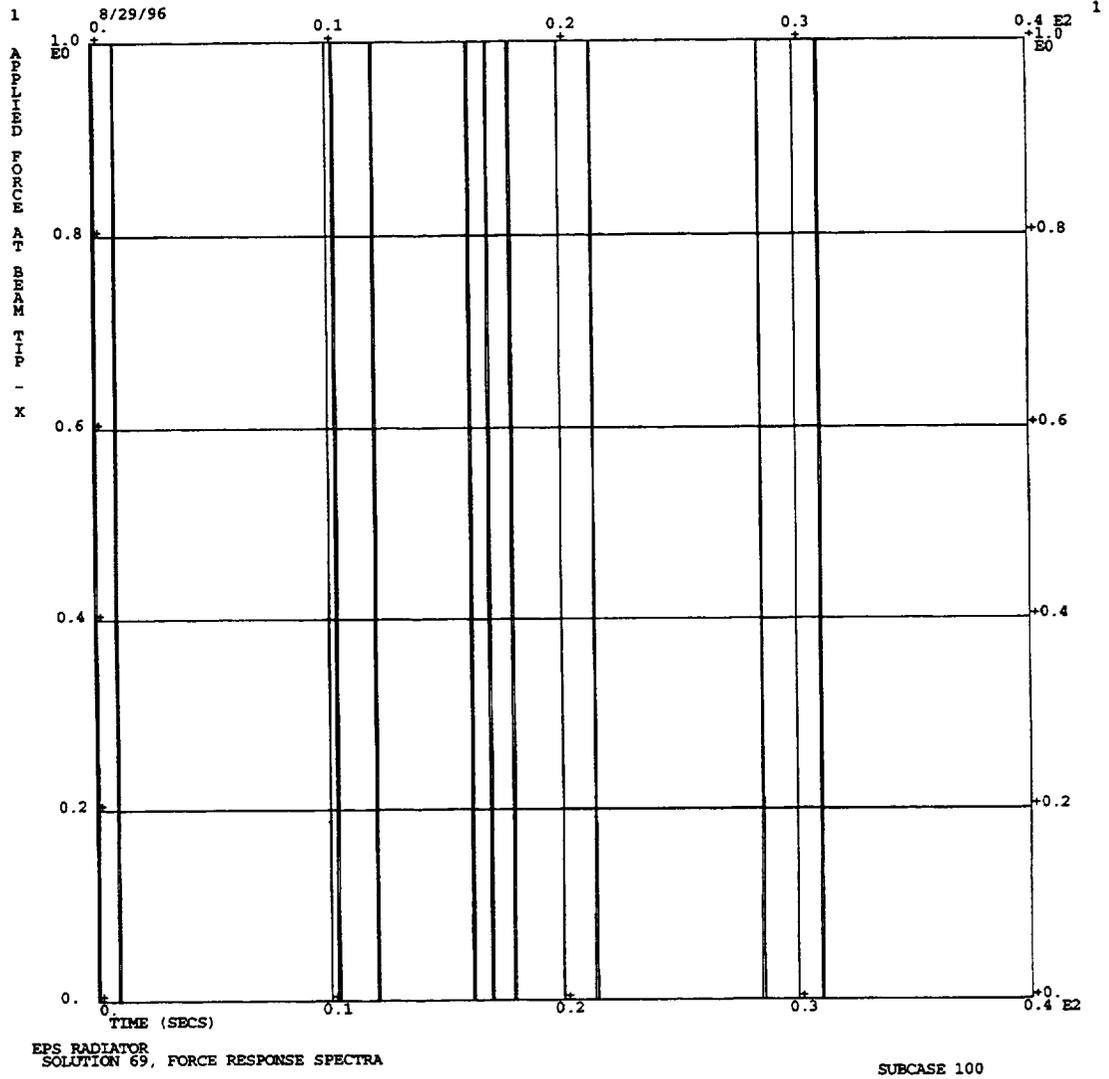


Figure (6.3) - Plume Impingement Dynamic Loading on the EPS Radiator

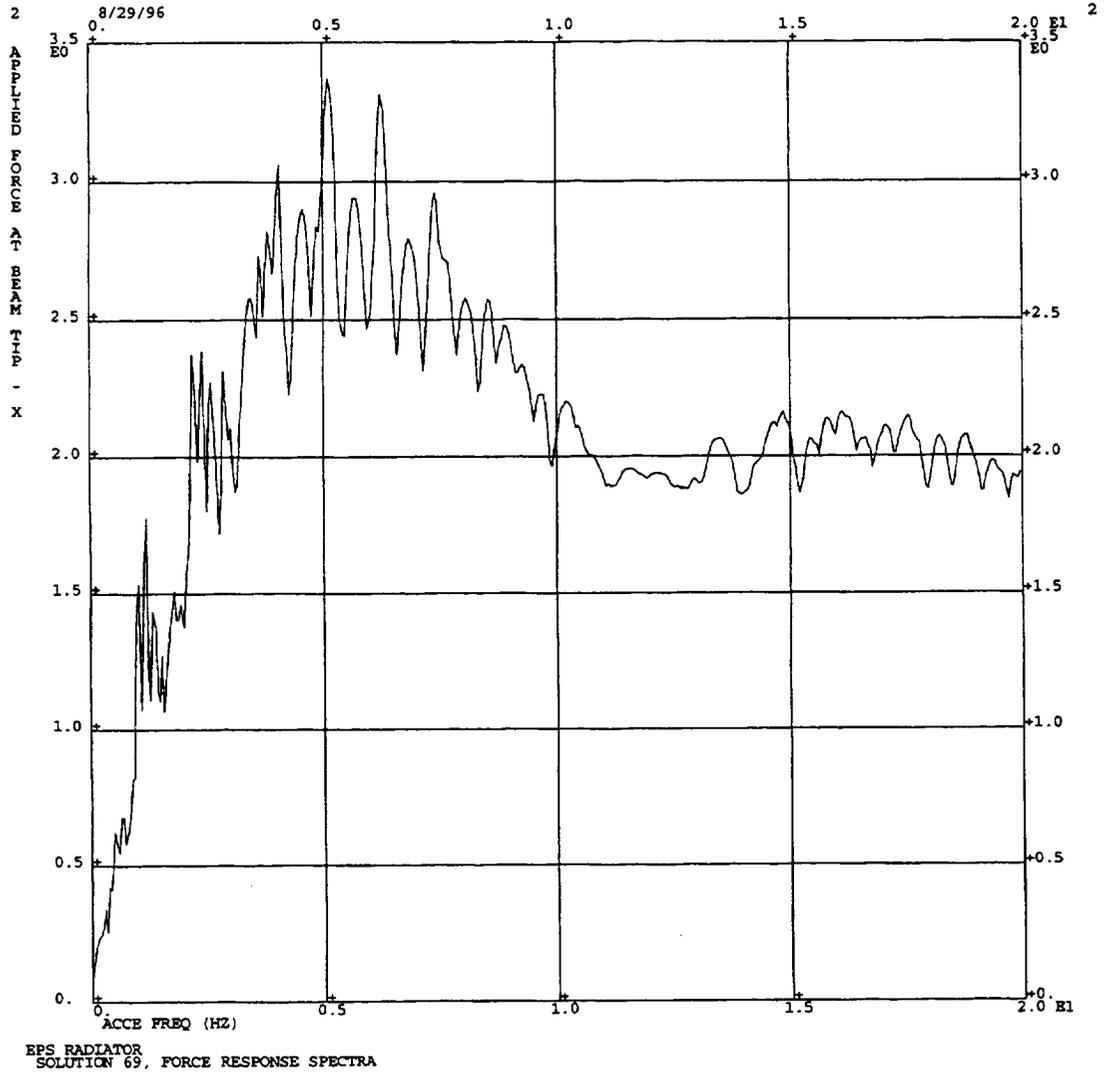


Figure (6.4) - Response Spectrum of the Plume Impingement Dynamic Loading

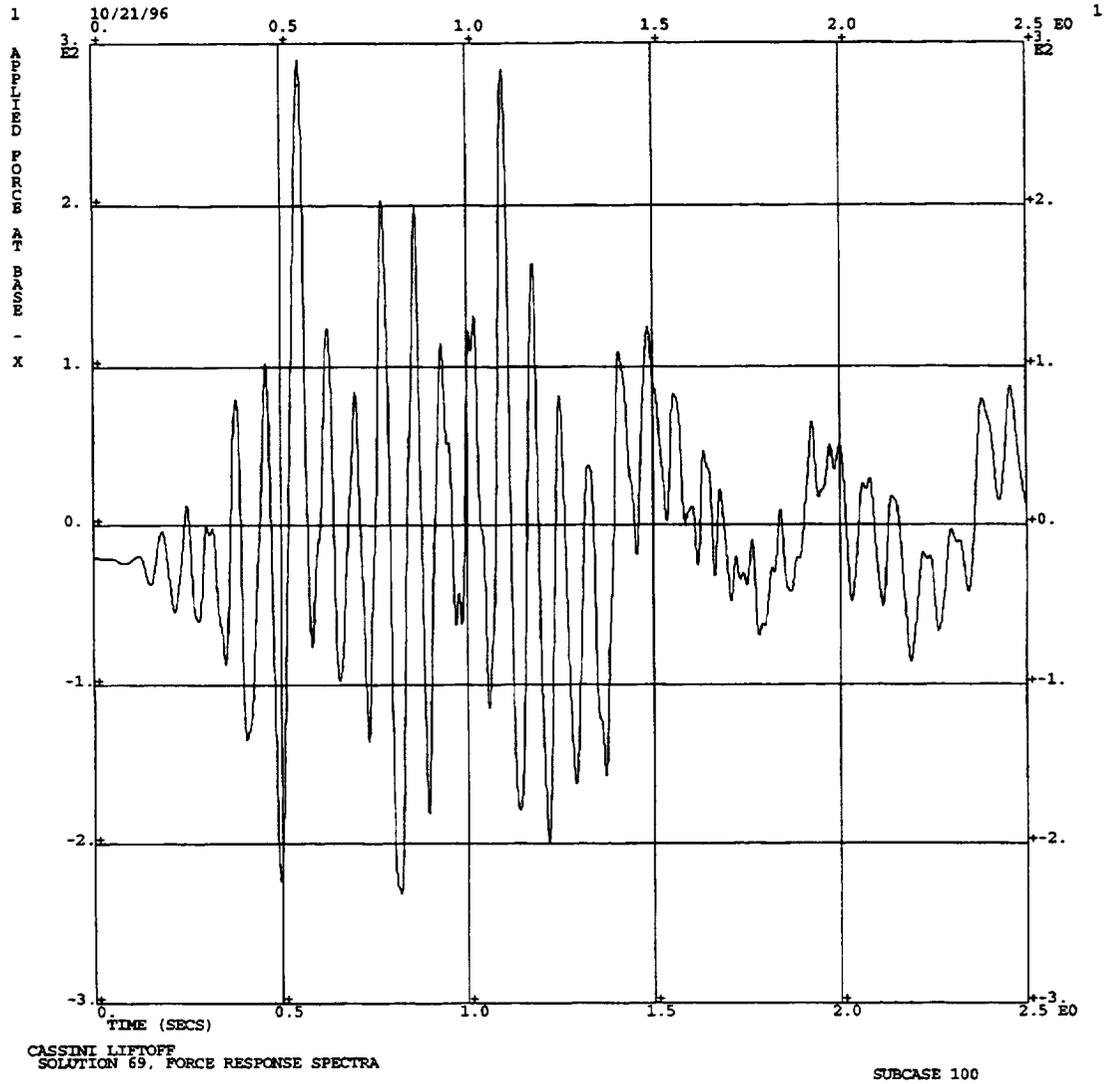


Figure (6.5) - Cassini Spacecraft Pitch Translation Interface Acceleration

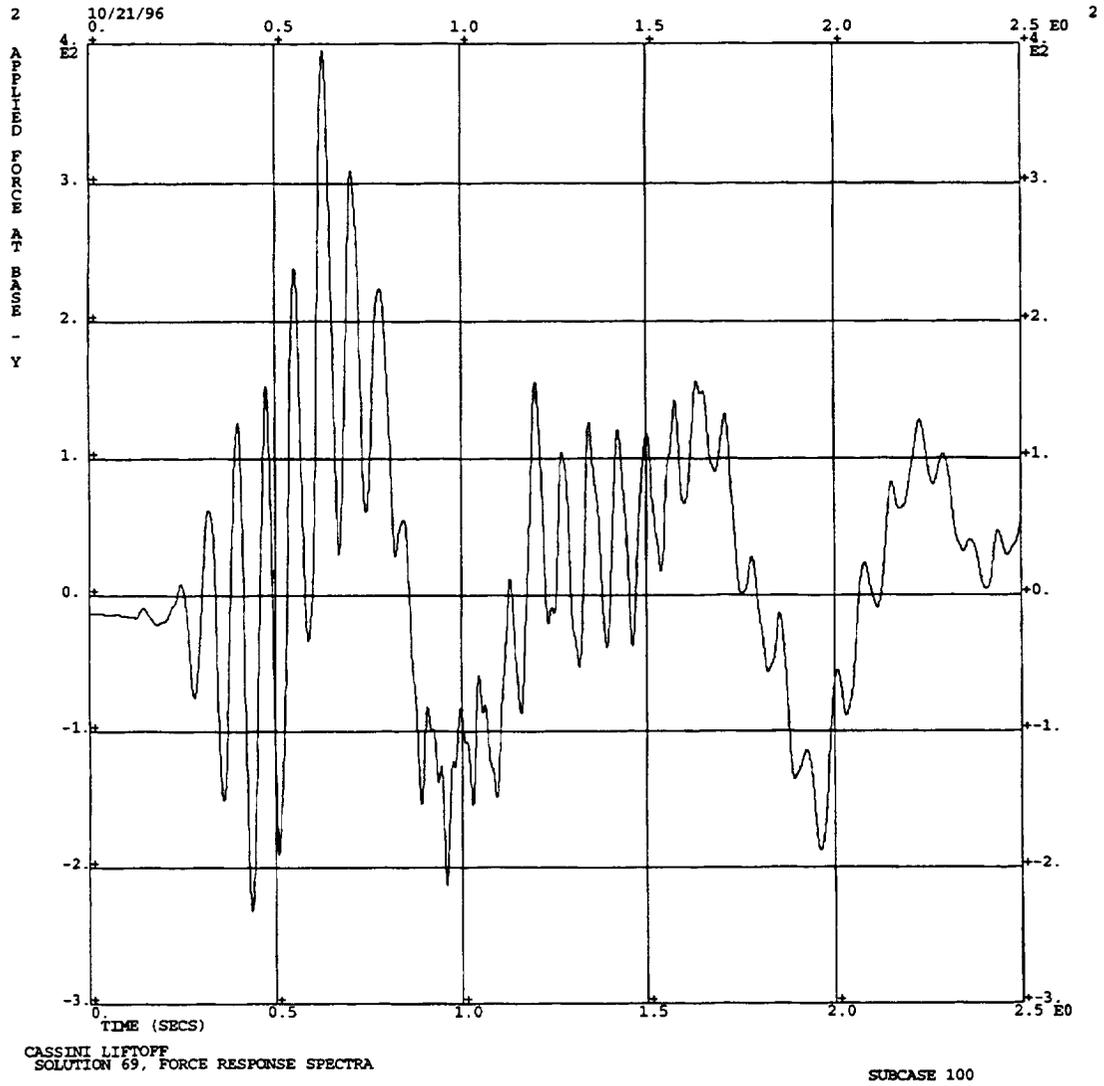


Figure (6.6) - Cassini Spacecraft Yaw Translation Interface Acceleration

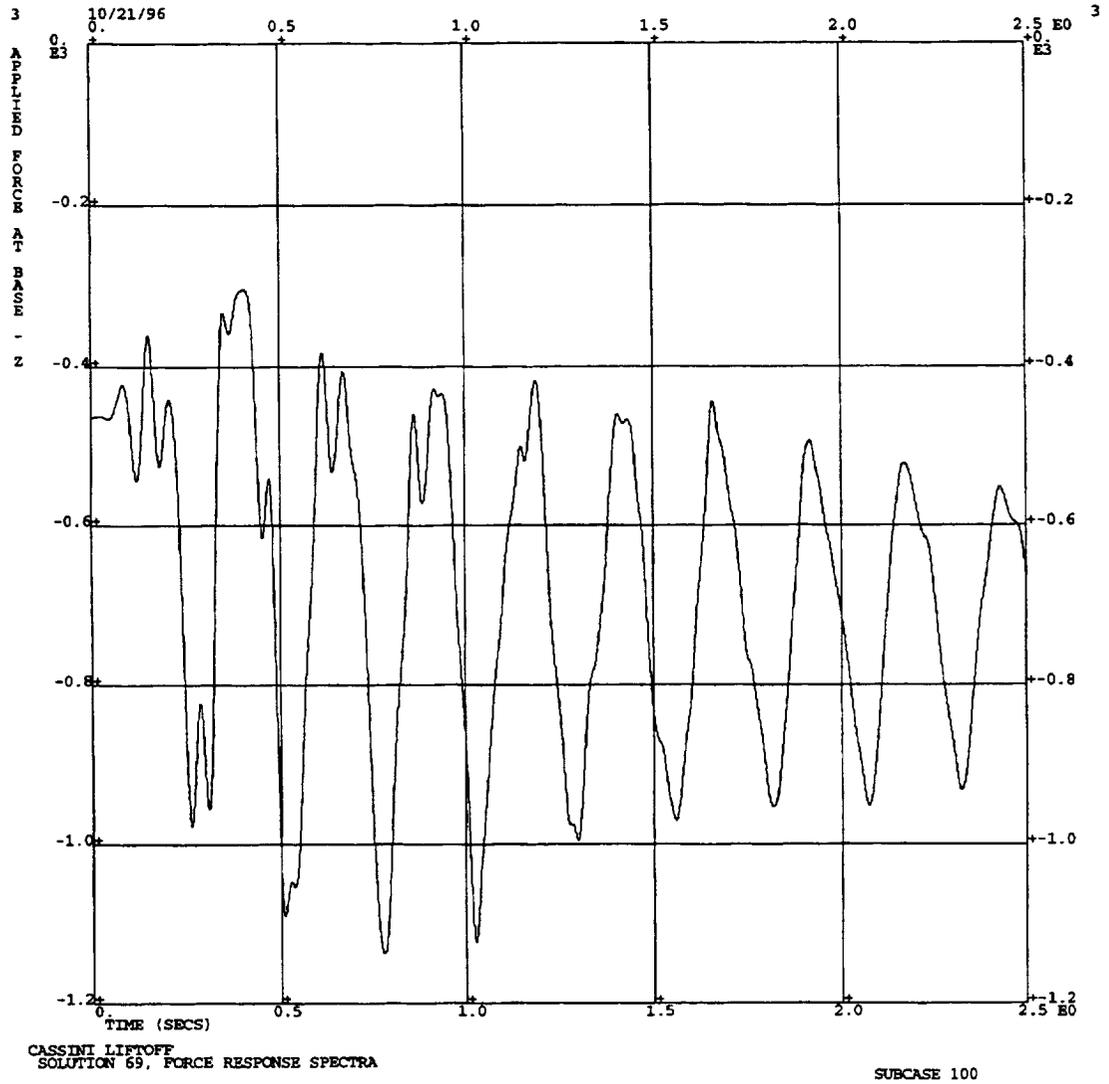


Figure (6.7) - Cassini Spacecraft Longitudinal Interface Acceleration

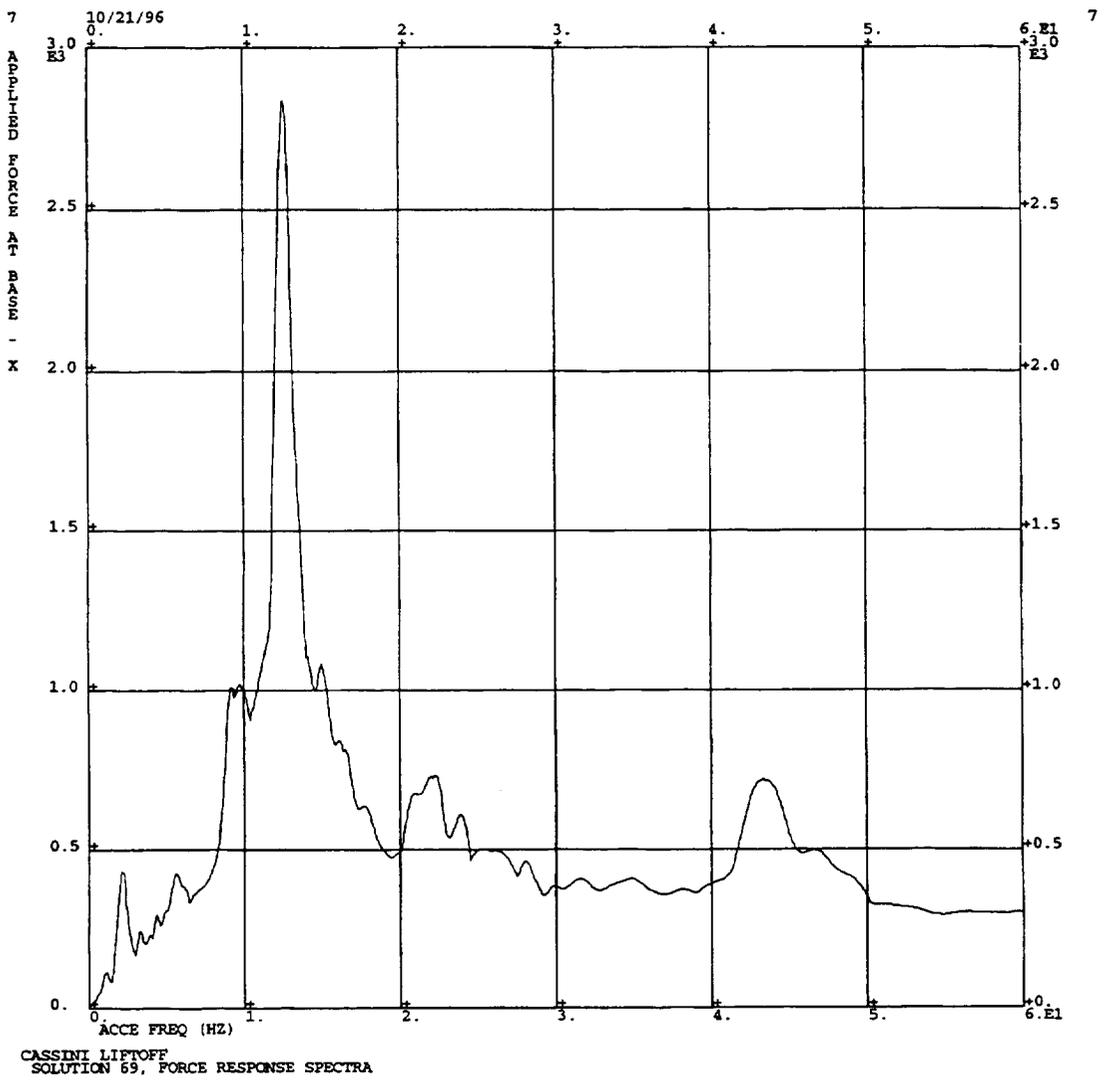


Figure (6.8) - Response Spectrum of the Pitch Interface Acceleration

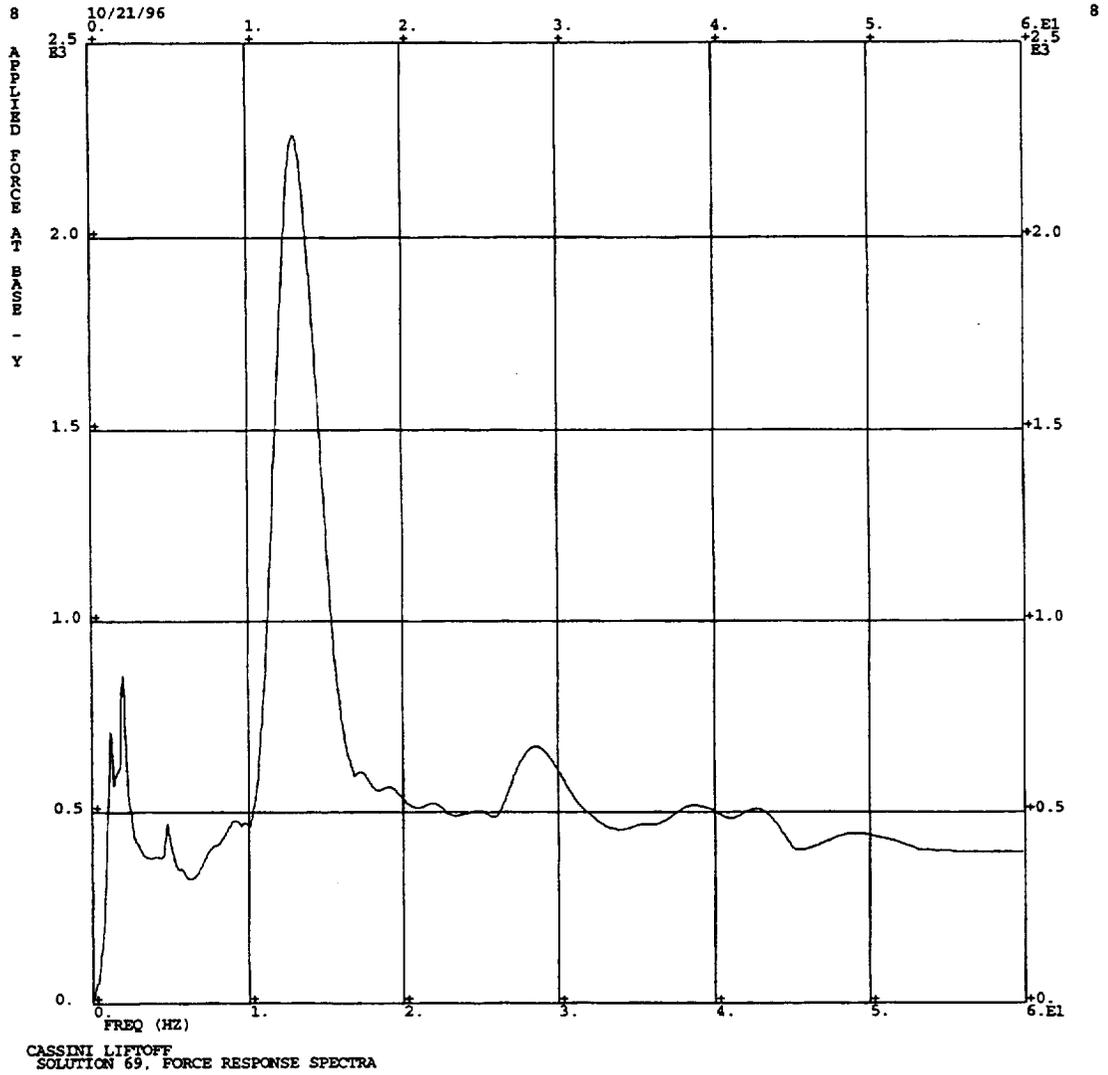


Figure (6.9) - Response Spectrum of the Yaw Interface Acceleration

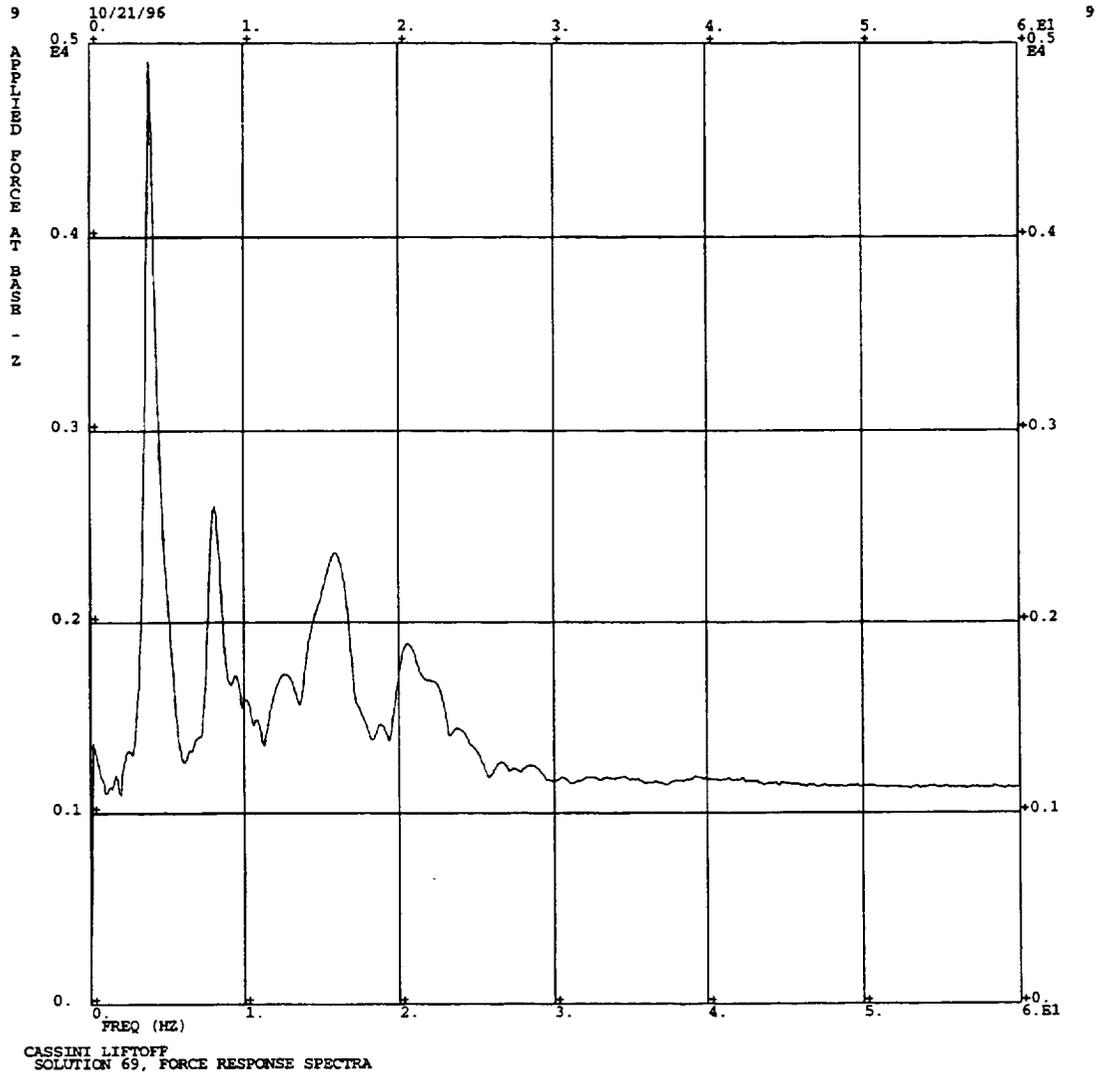


Figure (6.10) - Response Spectrum of the Longitudinal Interface Acceleration

## Chapter 7

### Krylov Sequence Termination Techniques

#### 7.1) Introduction:

Determining when a sufficient number of static Ritz vectors have been calculated is a problem which all static Ritz vector methods, including the boundary flexibility method, share. When enough static Ritz vectors have been calculated to accurately represent the dynamics of the given finite element model the Krylov sequence can be terminated. A heuristic method for sequence termination has been developed which is based, not on rigorous mathematics, but on the observed properties of load dependent and boundary flexibility static Ritz vector creation.

A mathematically rigorous basis for judging when the sequence can be terminated was extensively sought for, both in the literature and by analytical investigation. It is possible that such a solution exists, however, a rigorous solution was neither located nor could be created here. It is also possible that a mathematically rigorous sequence termination methodology, which would be independent of physical knowledge of typical structural dynamic systems (this knowledge was used to create the heuristic method presented in this chapter), cannot be created.

The mathematically rigorous solution sought was analogous to modal truncation. Modal truncation, based upon an eigenvalue cutoff, is the most popular basis for determining if a sufficient number of normal eigenvectors have been calculated. Static Ritz vectors do not have an eigenvalue with which to associate a

truncation limit, but they do have a Rayleigh value, as presented in chapter 6. Using Rayleigh values as a partial substitute for eigenvalues was explored and was found to not be practical.

### 7.2) Error Criteria and Effective Mass:

Several error criteria have been proposed which truncate the Krylov sequence when a somewhat arbitrary variable reaches a arbitrary value. These proposed criteria were briefly described in chapter 2. Most of the proposed termination criteria are applicable only to load dependent static Ritz vectors, with a goal of obtaining an accurate solution to the given static problem. Error criteria which are completely aimed toward the solution of static problems have a limited direct utility in their application to dynamic substructuring and boundary flexibility static Ritz vectors.

The single proposed error criteria for Krylov sequence termination which is directed to dynamic problems is that proposed by Yiu and Landress<sup>30</sup>. This criteria is based on a parameter, effective mass, commonly used in structural dynamics for the identification of globally important normal eigenvectors of a structure. Effective mass is a measure of the amount of the total structural mass represented in each individual eigenvector. Effective mass is calculated as follows, beginning with an intermediate matrix,  $M_{ER}$ , being defined as

$$M_{ER} = \Phi^T M \Phi_{RB} \quad (7.1)$$

where  $\Phi$  is the matrix of eigenvectors, and  $\Phi_{RB}$  is the rigid body transformation matrix of the structure. Each term within the resulting matrix of equation (7.1) is squared to form the effective mass of each eigenvector.

$$\textit{Effective Mass} = \langle M_{ER} \rangle^2 \quad (7.2)$$

Note that eigenvalues are not used in the effective mass calculation, allowing static Ritz vectors to directly replace the eigenvectors in equation (7.1). Each eigenvector has a total of six effective mass values, three for the translations and three for the rotations. The individual effective mass terms of the eigenvectors can, separately in each translation and rotation coordinate, be summed and the sum can be compared to the rigid body mass of the structure. The percentage of mass represented by eigenvectors, in each direction, is referred to as the total effective mass.

Yiu and Landress proposed that the effective mass calculation can be made using static Ritz vectors, and that when the total effective mass reaches an arbitrary percentage, the sequence can be terminated. The recommended arbitrary percentage was a minimum of 90% total effective mass, which is consistent with standard aerospace practice when using normal eigenvectors. In standard aerospace practice this cutoff value is used to determine if the important dynamics of a structure has been test verified, and in some cases even a greater percentage than 90% is specified.

The primary assumption made when using total effective mass as an error criteria is that, when enough vectors have been calculated to achieve the selected

arbitrary percentage of represented mass, an accurate solution is guaranteed. Engineering experience has demonstrated that if this percentage is set at a high enough value an accurate dynamic solution will often be possible. However, usually an accurate solution is possible with far fewer vectors than the amount required to reach the cutoff value. The effective mass of the normal eigenvectors, from the beam example model presented in chapter 5, is shown in Table (7.1). The effective mass of the normal eigenvectors, from the EPS Radiator and the Cassini finite element models, is shown in appendix A. These are typical effective mass results for typical finite element models. The total effective mass of these models does not converge monotonically, nor does it reach any particular percentage within the frequency range of interest. Table (7.1) demonstrates that, for the torsional rotation (R2) of the beam model, eigenvectors with eigenvalues up to 150 HZ yield a total effective mass of 76.7%. The contribution of the 22nd mode at 133.73 HZ was required to bring the axial translation (T2) total effective mass above 90%. Eigenvectors with eigenvalues in this frequency range are not required to allow an accurate dynamic solution for the excitation described in chapter 6. The rectangular impulse excitation has a relatively broad frequency spectrum, and excitation at a particular high frequency would require eigenvectors in that range for an accurate solution, but in any case a required total effective mass of 90% is arbitrary.

For the EPS Radiator model, there are 175 modes below 140 HZ. These modes yield a total effective mass of 78.1% for axial translation and 75.6% for out-

of-plane translation. The in-plane rotational total effective mass for this set of modes is 75.7%. If the desired total effective mass was set at a value of 90%, hundreds of vectors would be generated, but most are not required for an accurate dynamic solution, at frequencies of interest in loads calculation. The Cassini model total effective mass reaches 90% much quicker than the example beam and the EPS Radiator. Even here, if the desired value is set higher than 90%, many unnecessarily vectors might be calculated. In conclusion, effective mass, while a useful tool, was investigated but was not implemented in this work. Other similar error criteria were also found to not exhibit rapid, monotonic or well behaved convergence.

### 7.3) Modal Density Truncation:

A new method has been developed that is based upon the density of the modal space in a finite element model and the observation that the most dynamically significant static Ritz vectors are generated near the beginning of the Krylov sequence. This method was developed from the observation that almost all of the boundary flexibility static Ritz vectors calculated early in the Krylov sequence are required to obtain an accurate solution to the dynamic response problem. This includes static Ritz vectors with Rayleigh values which are high relative to the frequency range of interest, which might erroneously imply relatively low importance to dynamic response prediction. It was also observed, in general, for the test examples utilized and from the current literature, that accurate dynamic response predictions are possible with

fewer load or boundary flexibility derived static Ritz vectors than with normal eigenvectors.

Krylov sequence termination using modal density truncation can be described in detail as follows. First, a cutoff frequency is defined by the user. This frequency should be based upon the dynamic range of interest and the frequency spectrum of the forcing function. Next, the number of normal eigenvectors with eigenvalues below the cutoff frequency is determined, which is referred to here as the modal density. The actual eigenvectors and eigenvalues of the model are not required, just their number. The Krylov sequence is then initiated and when the number of static Ritz vectors created is equal to a fraction of the modal density, the sequence is terminated and the component is formed. A value for this fraction has been determined by trial and error. The effectiveness of this sequence termination strategy will be demonstrated in the next chapter.

Modal density truncation is not mathematically rigorous, but it is practical. It is effective because the order of the system, within the frequency range of interest, is related to the number of static Ritz which are needed to represent this model. This algorithm is simple, and does not pursue unobtainable arbitrary parameter values, which can be the case when using error functions. The computational cost of this method is the factorization of the mass and stiffness matrices, shifted to the cutoff frequency. The factorization for determining the modal density can be reused as a

shift in the Krylov sequence, targeting static Ritz vectors near the cutoff frequency at no additional computational cost.

#### 7.4) Determination of Modal Density and Truncation:

A method, called spectrum slicing, for determining the number of eigenvalues in a system, below a certain value, was presented in section 2.6. Equation (2.48) from that section, has been rewritten below in a form consistent with the equations in Tables (4.1) and (4.2). The triangular factorization of the matrices  $\mathbf{k}_{ii}$  and  $\mathbf{m}_{ii}$ , shifted to a cutoff value  $\sigma_c$ , is calculated

$$(\mathbf{k}_{ii} - \sigma_c \mathbf{m}_{ii}) = \mathbf{L} \mathbf{D} \mathbf{L}^T \quad (7.3)$$

where  $\sigma_c$  is calculated from the user defined cutoff frequency, by

$$\sigma_c = (2\pi f_c)^2 \quad (7.4)$$

Equation (2.49) can then rewritten as

$$\mathbf{v}(\Lambda - \sigma_c \mathbf{I}) = \mathbf{v}(\mathbf{k}_{ii} - \sigma_c \mathbf{m}_{ii}) = \mathbf{v}(\mathbf{D}) \quad (7.5)$$

where  $\mathbf{v}$  is number of negative eigenvalues and  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ . The number of negative elements of  $\mathbf{D}$  is equal to the number of eigenvalues,  $n_c$ , of the matrices  $\mathbf{k}_{ii}$  and  $\mathbf{m}_{ii}$ , which are less than  $\sigma_c$ .

The Krylov sequence is continued until the total number of calculated static Ritz vectors, here defined as  $n_r$ , is greater than the number of eigenvalues below  $\sigma_c$ ,

$n_c$ , multiplied by the fraction  $\psi$ . The fraction  $\psi$  was established by trial and error. Writing the termination logic in the form of the sub-indices from the equations of Tables (4.1) and (4.2) yields

$$\textit{if } n_r > \psi n_c, \textit{ then } j = l \quad (7.6)$$

All calculated static Ritz vectors,  $n_r$ , are retained for subsequent dynamic analysis. The boundary flexibility algorithm, with targeted shifting and modal density truncation included, and numerical examples will be presented in the next chapter.

Mode No.	Freq. (Hz)	Effective Masses (%)					
		T1	T2	T3	R1	R2	R3
1	0.55	0.0	0.0	61.3	97.0	0.0	0.0
2	0.55	61.3	0.0	0.0	0.0	0.0	97.0
3	3.41	18.1	0.0	0.8	0.1	0.0	2.4
4	3.41	0.8	0.0	18.1	2.4	0.0	0.1
5	9.52	2.8	0.0	3.7	0.2	0.0	0.1
6	9.52	3.7	0.0	2.8	0.1	0.0	0.2
7	18.57	3.2	0.0	0.1	0.0	0.0	0.1
8	18.57	0.1	0.0	3.2	0.1	0.0	0.0
9	27.58	0.0	80.6	0.0	0.0	0.0	0.0
10	30.53	2.0	0.0	0.0	0.0	0.0	0.0
11	30.53	0.0	0.0	2.0	0.0	0.0	0.0
12	45.41	1.3	0.0	0.1	0.0	0.0	0.0
13	45.41	0.1	0.0	1.3	0.0	0.0	0.0
14	63.18	0.7	0.0	0.3	0.0	0.0	0.0
15	63.18	0.3	0.0	0.7	0.0	0.0	0.0
16	81.90	0.0	8.6	0.0	0.0	0.0	0.0
17	83.59	0.4	0.0	0.3	0.0	0.0	0.0
18	83.59	0.3	0.0	0.4	0.0	0.0	0.0
19	84.64	0.0	0.0	0.0	0.0	76.7	0.0
20	104.41	0.3	0.0	0.0	0.0	0.0	0.0
21	104.41	0.0	0.0	0.3	0.0	0.0	0.0
22	133.73	0.0	2.8	0.0	0.0	0.0	0.0
		Total Effective Mass (%)					
		T1	T2	T3	R1	R2	R3
		95.4	92.1	95.4	100.0	76.7	100.0

Table (7.1) - Beam Finite Element Model Effective Masses

## Chapter 8

### Numerical Examples of Targeted Shifting and Modal Density Truncation

#### 8.1) Algorithm with Targeted Shifting and Modal Density Truncation:

The complete static Ritz vector, boundary flexibility algorithm, with targeted shifting and modal density truncation included, is shown in Table (8.1). This algorithm was extended from that presented in chapter 4, which included block filtering and Cholesky/QR orthonormalization. Only the revised boundary flexibility algorithm for euclidean vector orthonormalization is presented in this chapter, because there is no fundamental difference in the application of targeted shifting and modal density truncation to the various orthonormalization options presented in chapter 4.

#### 8.2) Numerical Results and the Determination of the Fraction $\psi$ :

Three sets of time response problems were performed for the three example models described in chapter 5 using the applied loading described in chapter 6. These direct transient response problems were all performed using dynamically reduced models, a structural damping ration of 2%, with the beam example and the EPS Radiator having a fixed interface. The Cassini model was excited using enforced acceleration on a seismic mass. Two comparison cases for each model were generated using eigenvectors, the first using a modal truncation frequency greater than the dynamic range of interest, and the second using a frequency cutoff at the frequency range of interest. The beam example cutoff was 250 HZ for the high frequencies and 100 HZ for the representative frequencies. The EPS Radiator and the Cassini cutoff

frequencies were 150 HZ for the high frequencies and 60 HZ for the representative frequencies. Three time response problems, using three different values for the fraction  $\psi = 1.0, .5$  and  $.25$ , for each example model created with static Ritz vectors, were performed.

A selected set of physical accelerations and loads for each of these models was recovered. Table (8.2) shows the ratios of the minimum and maximum peak responses of the beam example, formulated using the options discussed above, to the prediction using the high frequency cutoff eigenvalue model. Tables (8.3) and (8.4) present similar tables for the EPS Radiator and the Cassini model transient solutions. There is no guarantee that the high frequency cutoff eigenvalue model represents a completely converged solution. The cutoff values were selected by multiplying the frequency at the cutoff of the dynamic range of interest by 2.5, and, considering the large size of the Cassini and the EPS Radiator models, this cutoff produced a size dynamic model for which a transient solution could still conveniently be obtained.

Several observations can be made of the beam example ratios presented in Table (8.2). First, in the case of the axial tip acceleration, which is dependent on high frequency beam dynamics, the  $\psi = 1.0$  Ritz vector response predicted a more complete solution than did the eigenvector response. Figures (8.1) and (8.2) show axial tip acceleration time history plots predicted by the 250 Hz cutoff eigenvector and the  $\psi = 1.0$  Ritz vector models. The ratios for the  $\psi = .50$  and the  $\psi = .25$  beam examples are identical. This is because the truncation criteria algorithm produced

identical representations. There are so few eigenvectors in the frequency range of interest that, after the initial two vector blocks are calculated, there are already enough vectors to satisfy the truncation criteria. As a result, no more vectors were calculated for the  $\psi = .50$  beam example than were calculated for the  $\psi = .25$  beam example, and targeted shifting and the representative static load is not utilized for either of these two cases. Not utilizing targeted shifting and a representative static load vector partially explain the relatively low response predictions of the  $\psi = .50$  and the  $\psi = .25$  cases. The lateral tip acceleration time history for the  $\psi = 1.0$  case is illustrated in Figure (8.3), and the base bending moment for the same case is illustrated in Figure (8.4). The time histories of the other cases are not included because they are essentially identical.

Table (8.3) presents the time response ratios generated using the EPS Radiator model. All the results from the various cases are reasonably consistent with the 150 Hz frequency cutoff case. There begins to be some divergence from the 150 Hz case in the responses generated using the  $\psi = .25$  model. This would suggest that a fraction value of  $\psi = .50$  might be appropriate for accurate, but low computational cost, response predictions. In general, the responses predicted using the  $\psi = 1.0$  Ritz vector component is a marginally closer match to the high frequency cutoff case than the 60 Hz eigenvector model.

Figures (8.5), (8.6), (8.7), and (8.8) present the time response of the X acceleration, the Z acceleration, and the two bending moments, for the  $\psi = 1.0$  case.

As in the beam example, the time histories plots of the EPS Radiator from the various cases are virtually identical. Examining the X acceleration time history explains the somewhat lower ratios obtained for the  $\psi = .50$  and the  $\psi = .25$  maximum values comparisons. The minimum value is much greater than the maximum value, which represents a relatively small overshoot, and the ratio of the maximums represent a comparison of relatively small values.

Table (8.4) presents the ratios of the time responses of the Cassini model using various  $\psi$ 's, compared to the 150 Hz cutoff eigenvector component. There is more divergence in the results of the various cases using the Cassini model than was in the previously discussed models. It was more difficult for the static Ritz vectors to match appendage accelerations than other responses. Once again, the results suggest that fraction value of approximately  $\psi = .50$  might be the appropriate choice for reasonably accurate response predictions. The High Gain Antenna Strut response predictions of the 60 Hz cutoff eigenvector component indicate that, for this element, the 60 Hz cutoff was too low. The Ritz vector models, whose vector size is based upon the number of vectors in the 60 Hz model, were also unable to accurately predict these strut loads. However, for the element forces in the main body stringer, the Ritz vectors were able to obtain more accurate response predictions than the 60 Hz eigenvector component. In conclusion, the use of a fraction value of  $\psi = .50$  may not provide enough Ritz vectors for accurate response predictions if the frequency cutoff of interest selected excludes important dynamic effects.

Figures (8.9), (8.10), (8.11), and (8.12) present the time response of the X, Y, and Z acceleration, and the strut axial load, for the  $\psi = 1.0$  case. The ratios of the maximum values of the accelerations in the Z direction were not presented in Table (8.4). An examination of Figure (8.11), the Z direction acceleration, will show that the maximum value occurs near the initial time step and its value is near zero. Ratios of these small response values was not meaningful and were not calculated. Appendix B contains the minimum and maximum responses of all the models and cases, and it was from this data that Tables (8.2), (8.3), and (8.4) were generated.

After the assembly of the component mass,  $\mathbf{m}$ , and stiffness,  $\mathbf{k}$ , matrices:

1) Modal Density Calculation

$$\sigma_c = (2 \pi f_c)^2 \quad \text{[shift value is calculated from user defined cutoff frequency, } f_c \text{]}$$

$$(\mathbf{k}_{ii} - \sigma_c \mathbf{m}_{ii}) = \mathbf{L}_c \mathbf{D}_c \mathbf{L}_c^T = \mathbf{L}_c \mathbf{U}_c \quad \text{[decompose shifted matrices]}$$

$$\sigma_c = \sigma_c + .1 \quad \text{[if and only if } (\mathbf{k}_{ii} - \sigma_c \mathbf{m}_{ii}) \text{ is singular]}$$

$$v(\mathbf{D}) = n_c \quad \text{[count the eigenvalues below } \sigma_c \text{]}$$

2) Initialization

$$\Phi_{ic} = -\mathbf{k}_{ii}^{-1} \mathbf{k}_{ic} \quad \text{[create boundary flexibility matrix, } \mathbf{k}_{ii} \text{ and } \mathbf{k}_{ic} \text{ were defined in equation (2.3)]}$$

$$\mathbf{q}_1^* = \mathbf{k}_{ii}^{-1} (\mathbf{m}_{ii} \Phi_{ic} + \mathbf{m}_{ic}) \quad \text{[create the initial block]}$$

[For a free interface component, (If the component has rigid body modes then the elastic flexibility matrix,  $\mathbf{g}_e$ , defined by (2.35) through (2.39) is used):

$$\mathbf{g}_a = \begin{bmatrix} \mathbf{g}_{cc} \\ \mathbf{g}_{ic} \end{bmatrix} \quad \text{where } \mathbf{g} = \mathbf{k}^{-1} \quad \text{[boundary flexibility matrix]}$$

$$\mathbf{q}_1^* = \mathbf{g} \mathbf{m} \mathbf{g}_a \quad \text{[create the initial block]}$$

3a) Filtering of Initial Block

$$\mathbf{q}_1^{**} = \mathbf{q}_1^* [\langle \mathbf{diag}(\mathbf{q}_1^{*T} \mathbf{q}_1^*) \rangle_{cc}]^{-1/2} \quad \text{[normalization]}$$

$$\mathbf{L}_E = \mathbf{q}_1^{**T} \mathbf{q}_1^{**} \quad \text{[cross orthogonality]}$$

Table (8.1) - Revised Boundary Flexibility Algorithm  
With Targeted Shifting and Modal Density Truncation

## 3a) (Continued) Filtering of Initial Block

$$L_E \rightarrow \begin{bmatrix} \backslash & \mathbf{0} \\ \bar{L}_E & \backslash \end{bmatrix} \quad [\text{partition into lower triangular}]$$

If  $(\|\bar{L}_E\|_\infty) \geq e$ , *then dependent* [infinity norm of each vector]

$$q_1^{**} \rightarrow [q_1^{***} \quad q_{DEP}] \quad [\text{partition out dependent vectors}]$$

## 3b) Orthonormalization of the Initial Block

$$L_E^* = q_1^{***T} q_1^{***} \quad [B^T B \text{ matrix product}]$$

$$L_E^* = R^T R \quad [\text{Cholesky factor decomposition}]$$

$$R^T q_1^T = q_1^{***} \quad [\text{Solve by forward substitution}]$$

4) For Blocks  $j = 2, 3, \dots, l$ , until  $n_r > \psi n_c$ 4a) Set Shift Variables, Based on Count of  $n_r$ 

$$\text{If } n_r < \frac{\psi n_c}{4}, \text{ then } k = 0, \sigma_k = 0 \quad [\text{no shift}]$$

$$\text{If } n_r > \frac{\psi n_c}{4}, \text{ then } k = 1, \sigma_k = \sigma_{T1}, \sigma_k = \frac{1}{3}\sigma_c \quad [\text{use } \sigma_{T1}, \text{ if defined}]$$

$$\text{If } n_r > \frac{\psi n_c}{2}, \text{ then } k = 2, \sigma_k = \sigma_{T2}, \sigma_k = \frac{2}{3}\sigma_c \quad [\text{use } \sigma_{T2}, \text{ if defined}]$$

$$\text{If } n_r > \frac{3\psi n_c}{4}, \text{ then } k = 3, \sigma_k = \sigma_c \quad [\text{reuse } L_c \text{ and } U_c \text{ from modal density calculation}]$$

Table (8.1) (Continued) - Revised Boundary Flexibility Algorithm  
With Targeted Shifting and Modal Density Truncation

## 4b) Vector Calculation and Filtering

$(\mathbf{k}_{ii} - \sigma_k \mathbf{m}_{ii}) = \mathbf{L}_s \mathbf{U}_s$	[if $k = 3$ , then $\mathbf{L}_s = \mathbf{L}_c$ and $\mathbf{U}_s = \mathbf{U}_c$ ]
$\sigma_k = \sigma_k + .1$	[if and only if $(\mathbf{k}_{ii} - \sigma_k \mathbf{m}_{ii})$ is singular]
$\mathbf{L}_s \mathbf{U}_s \mathbf{q}_j^* = \mathbf{m}_{ii} \mathbf{q}_{j-1}$	[create new vector block, unless $k = 1$ ]
$\mathbf{L}_s \mathbf{U}_s \mathbf{q}_j^* = \mathbf{m}_{ii} [\mathbf{q}_{j-1} \mathbf{p}_k]$	[create new vector block, if $k = 1$ ]
$\mathbf{q}_j^{**} = \mathbf{q}_j^* [ < \mathbf{diag}(\mathbf{q}_j^{*T} \mathbf{q}_j^*) >_{cc} ]^{-1/2}$	[normalization]
$\mathbf{L}_E = \mathbf{q}_j^{**T} \mathbf{q}_j^{**}$	[cross orthogonality]
$\mathbf{L}_E \rightarrow \begin{bmatrix} \setminus & \mathbf{0} \\ \bar{\mathbf{L}}_E & \setminus \end{bmatrix}$	[partition into lower triangular]
If $( < \ \bar{\mathbf{L}}_E\ _{\infty} >_c ) \geq e$ , than dependent	[infinity norm of each vector]
$\mathbf{q}_j^{**} \rightarrow [ \mathbf{q}_j^{***} \quad \mathbf{q}_{DEP} ]$	[partition out dependent vectors]

## 4c) Orthonormalization

$\mathbf{L}_E^* = \mathbf{q}_{1,j}^{***T} \mathbf{q}_{1,j}^{***}$	[ $\mathbf{B}^T \mathbf{B}$ matrix product]
$\mathbf{L}_E^* = \mathbf{R}^T \mathbf{R}$	[Cholesky factor decomposition]
$\mathbf{R}^T \mathbf{q}_{1,j}^T = \mathbf{q}_{1,j}^{***}$	[solve by forward substitution]
$\text{columns}(\mathbf{q}_{1,j}) = n_r$	[count total calculated vectors]

5) When  $n_r > \psi n_c$ , Transform System to Form Component

Table (8.1) (Continued) - Revised Boundary Flexibility Algorithm  
With Targeted Shifting and Modal Density Truncation

		Eigen - 100 Hz		Ritz - $\psi = 1.0$		Ritz - $\psi = .50$		Ritz - $\psi = .25$	
		Min	Max	Min	Max	Min	Max	Min	Max
Tip Acceleration									
Lat		1.00	1.00	1.00	1.00	.88	.85	.88	.85
Axial		0.87	0.89	1.11	1.13	.85	.85	.85	.85
Element Forces at the Base of the Beam									
M		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
V		1.00	1.00	1.00	1.00	.99	1.00	.99	1.00

Table (8.2) - Ratios of Responses of the Beam Example Model Using Various Representations to the 250 HZ Cutoff Eigenvector Representation Responses

		Eigen - 60 Hz		Ritz - $\psi = 1.0$		Ritz - $\psi = .50$		Ritz - $\psi = .25$	
		Min	Max	Min	Max	Min	Max	Min	Max
Tip Acceleration									
X		1.00	1.01	1.00	.98	.97	.88	.97	.87
Y		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Z		1.00	1.01	.99	.99	.98	1.02	.97	1.00
Element Forces at the Base of the Scissors Beam									
M1		1.02	1.01	1.01	1.01	1.00	1.00	1.01	1.01
M2		1.02	1.02	1.00	1.01	1.02	1.04	1.07	1.11
V		1.02	1.02	1.01	1.00	1.04	1.02	1.11	1.07
P		1.02	1.00	1.00	1.00	1.02	1.00	1.12	1.10
T		1.02	1.01	1.01	1.01	1.00	1.00	1.01	1.01

Table (8.3) - Ratios of Responses of the EPS Radiator Model Using Various Representations to the 150 HZ Cutoff Eigenvector Representation Responses

	Eigen - 60 Hz		Ritz - $\psi = 1.0$		Ritz - $\psi = .50$		Ritz - $\psi = .25$	
	Min	Max	Min	Max	Min	Max	Min	Max
Acceleration at the Oxidizer Tank cg								
X	1.00	1.00	1.07	1.03	1.06	1.20	.77	.77
Y	1.00	1.00	.97	1.02	.97	.99	.96	1.09
Z	1.00		1.02		.93		.92	
Acceleration at the Probe cg								
X	1.00	1.00	1.00	.98	1.16	1.07	1.03	.96
Y	1.00	1.00	.98	1.00	.97	1.01	.97	.92
Z	1.00		1.00		1.02		1.01	
High Gain Antenna Strut Beam Element Forces								
M1	1.00	1.00	1.00	.96	1.08	.93	.87	.94
M2	.98	1.02	1.00	.95	.92	.88	.84	.89
P	1.05	.90	.98	.94	.89	.82	.90	.84
Main Body Stringer Beam Element Forces								
M1	1.06	.94	.99	1.01	.99	.99	.97	1.02
M2	.99	1.00	.99	1.03	.93	1.00	.99	.99
P	1.02	.99	.99	1.01	1.00	1.01	.96	1.03
Lower Equipment Module Skin Membrane Forces								
Fx	1.00	1.00	1.00	1.00	1.01	1.04	.99	.99
Fy	.99	1.01	1.00	1.00	.99	1.00	1.00	1.00
Fxy	.99	1.00	.99	1.00	1.00	1.01	.92	1.07

Table (8.4) - Ratios of Responses of the Cassini Model Using Various Representations to the 150 HZ Cutoff Eigenvector Representation Responses

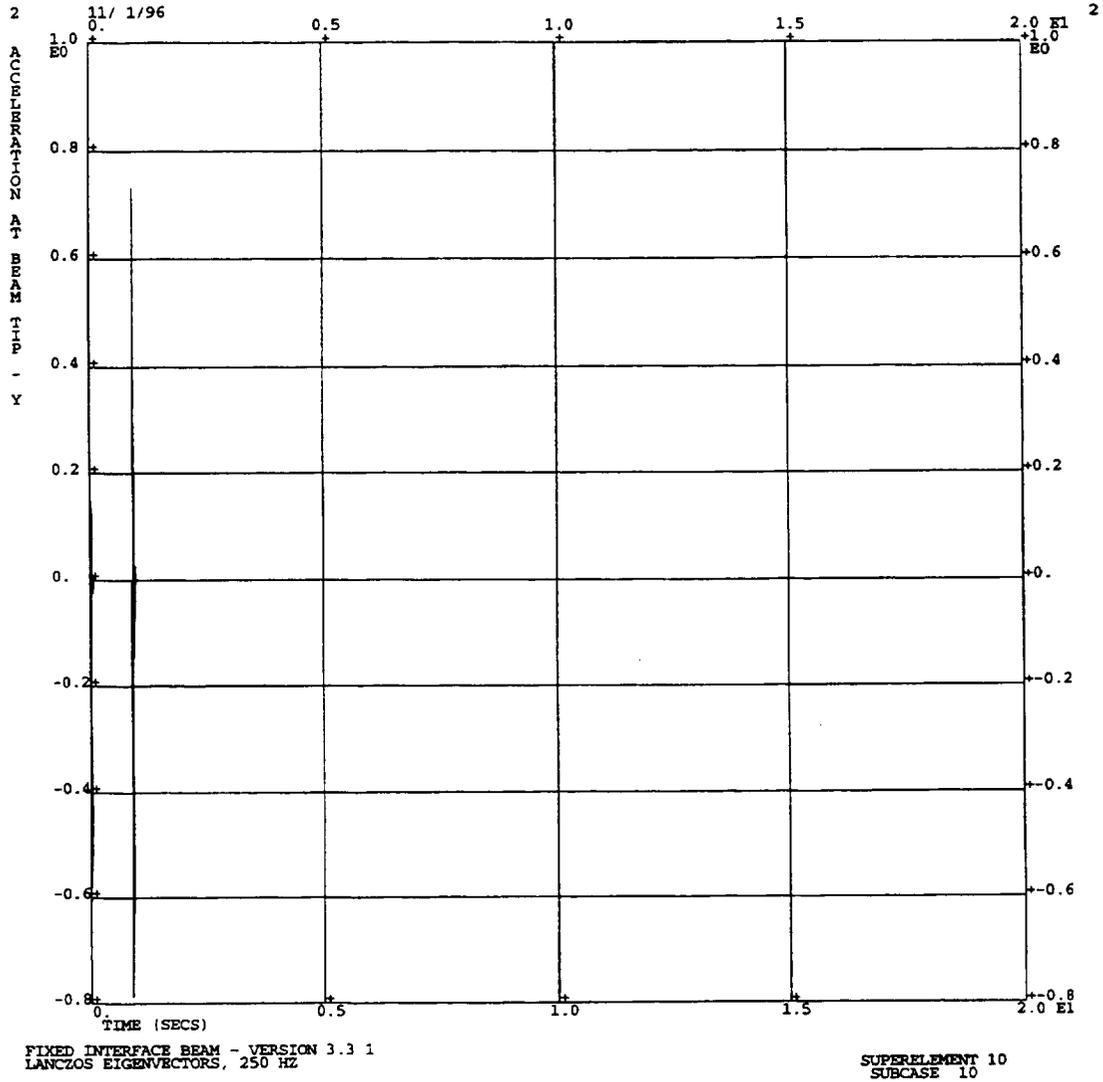


Figure (8.1) - Beam Example Tip Axial Acceleration  
Generated Using a 250 Hz Eigenvector Cutoff

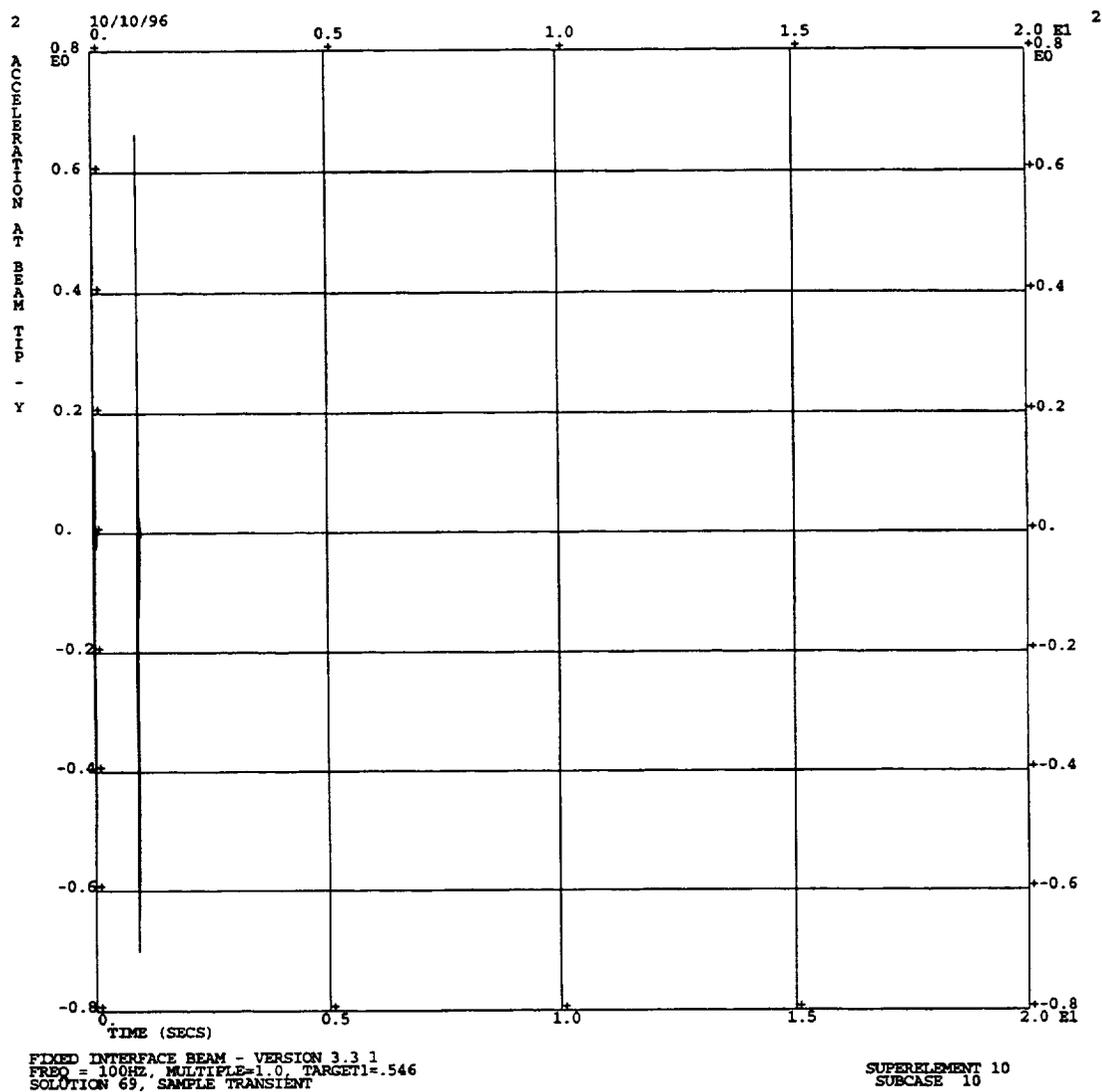


Figure (8.2) - Beam Example Tip Axial Acceleration  
Generated Using a  $\psi = 1.0$  Ritz Vector Representation

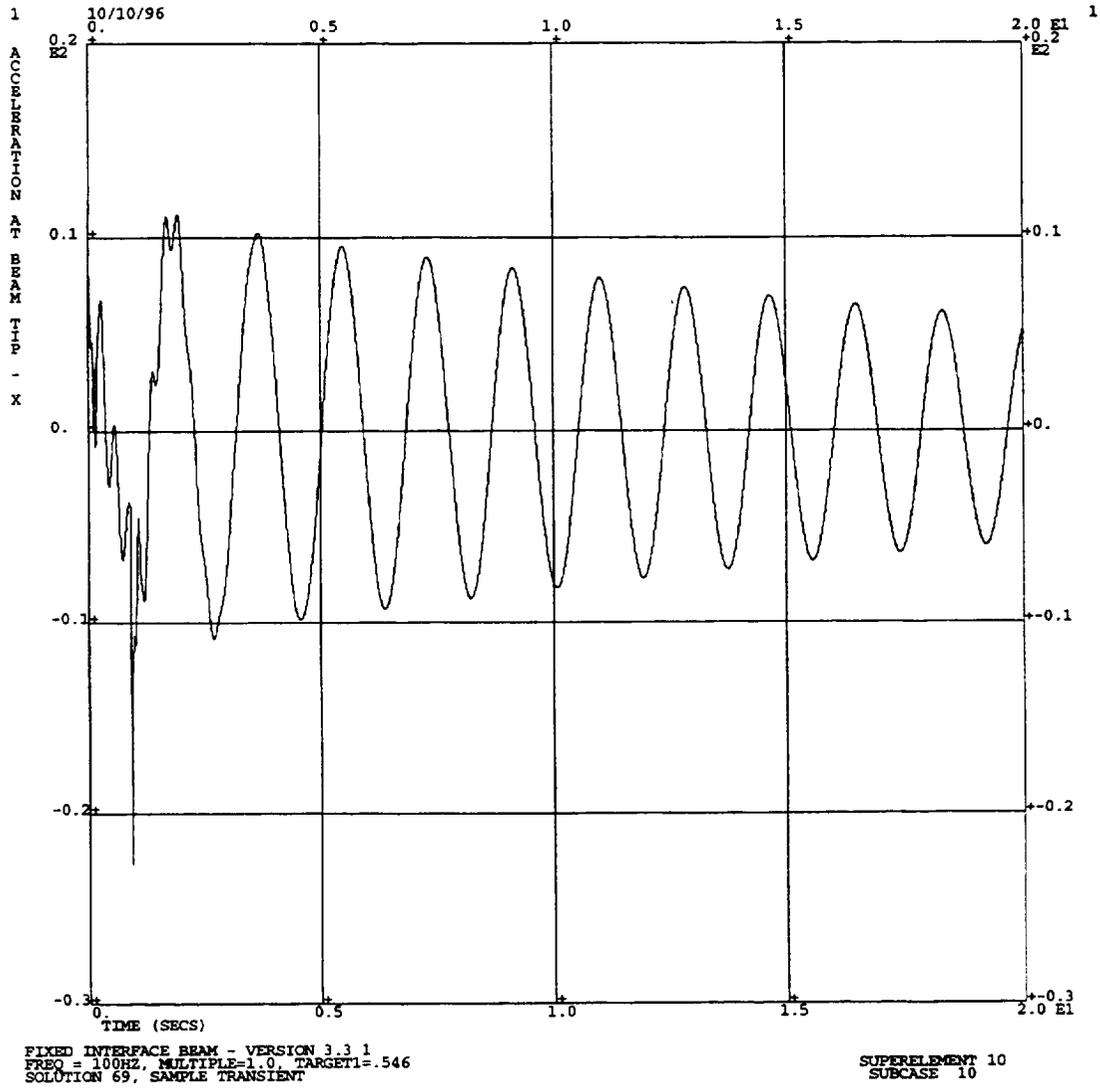


Figure (8.3) - Beam Example Tip Lateral Acceleration  
Generated Using a  $\psi = 1.0$  Ritz Vector Representation

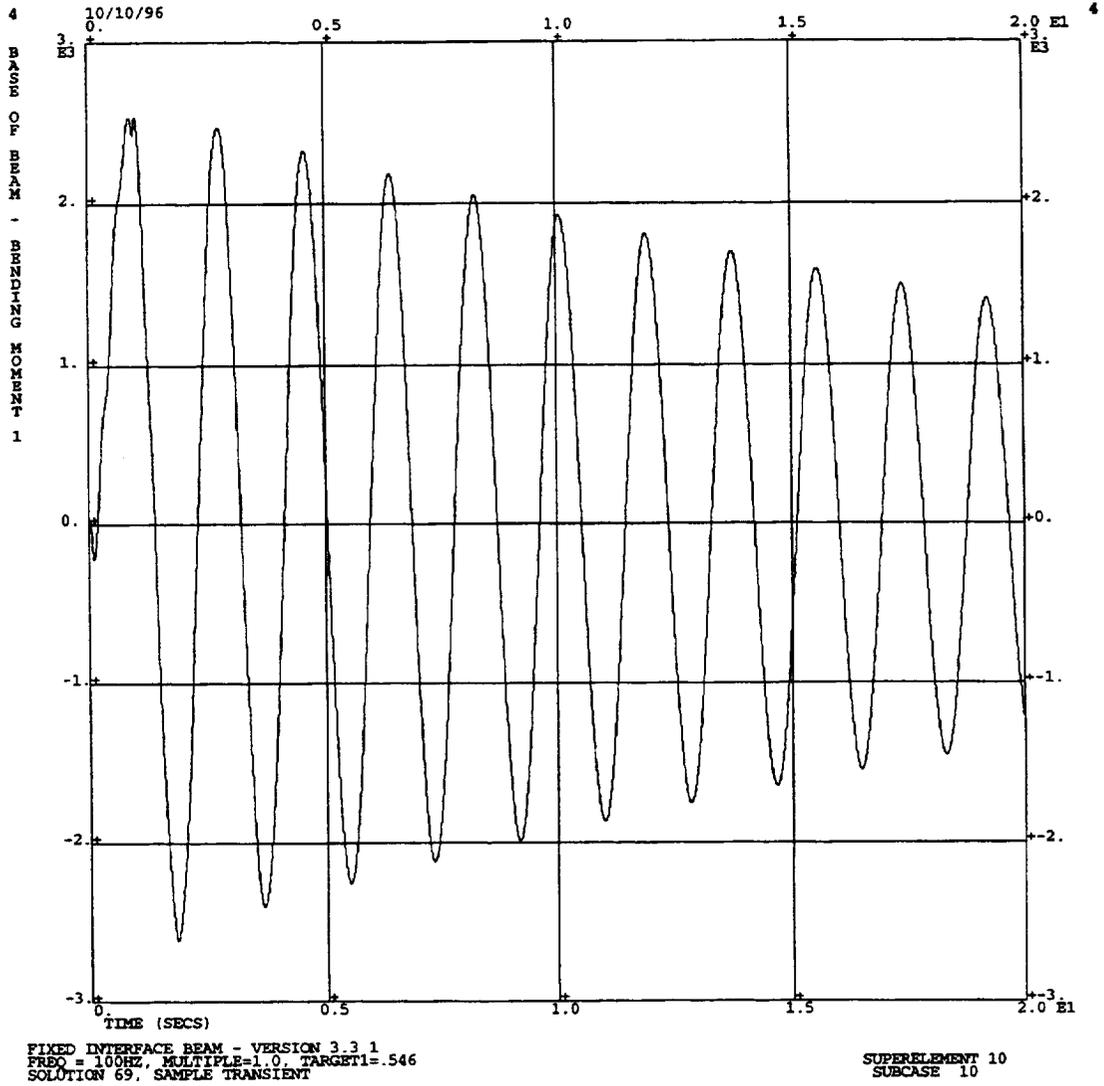


Figure (8.4) - Beam Example Base Bending Moment  
Generated Using a  $\psi = 1.0$  Ritz Vector Representation

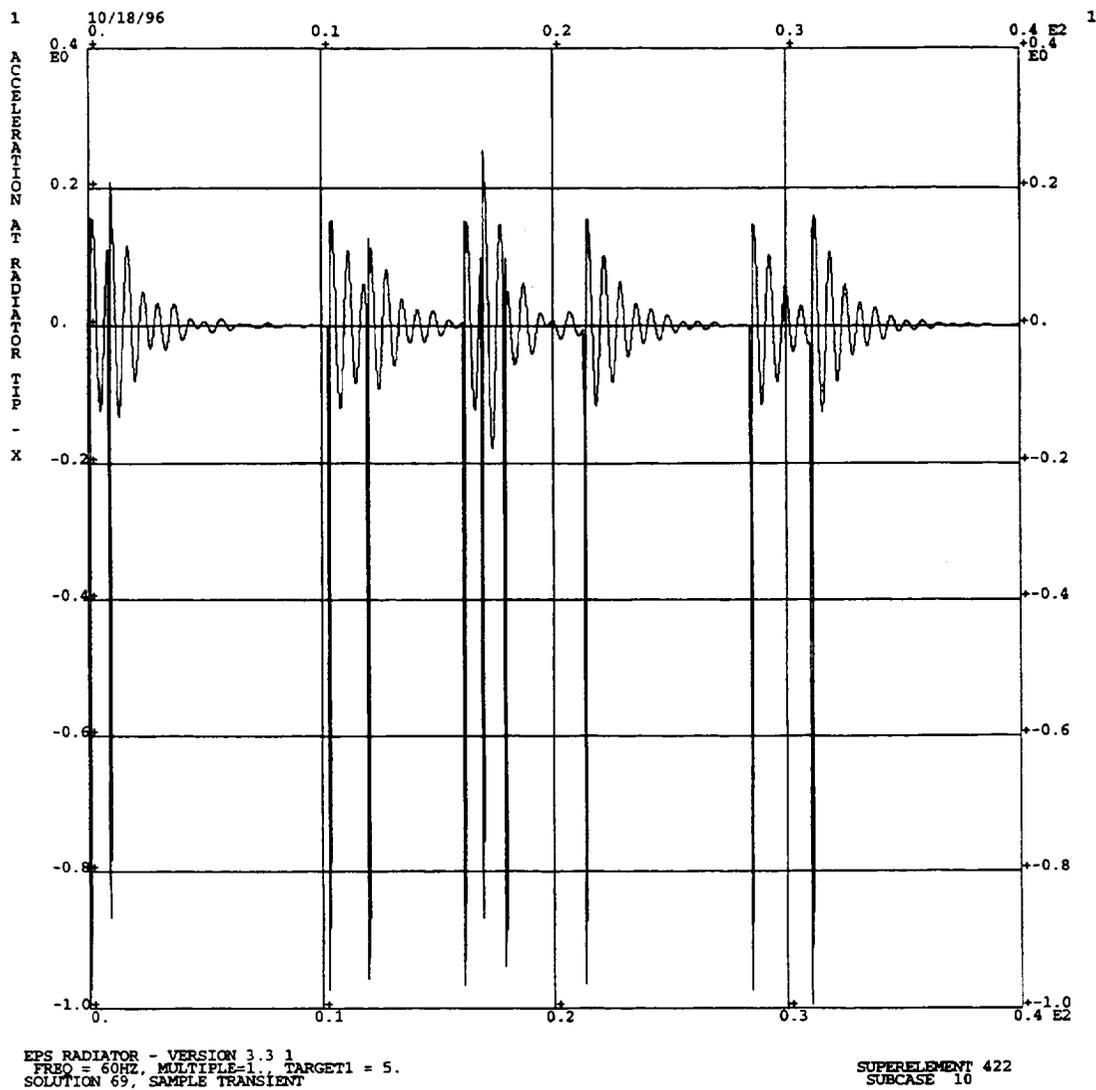


Figure (8.5) - EPS Radiator X Tip Acceleration  
Generated Using a  $\psi = 1.0$  Ritz Vector Representation

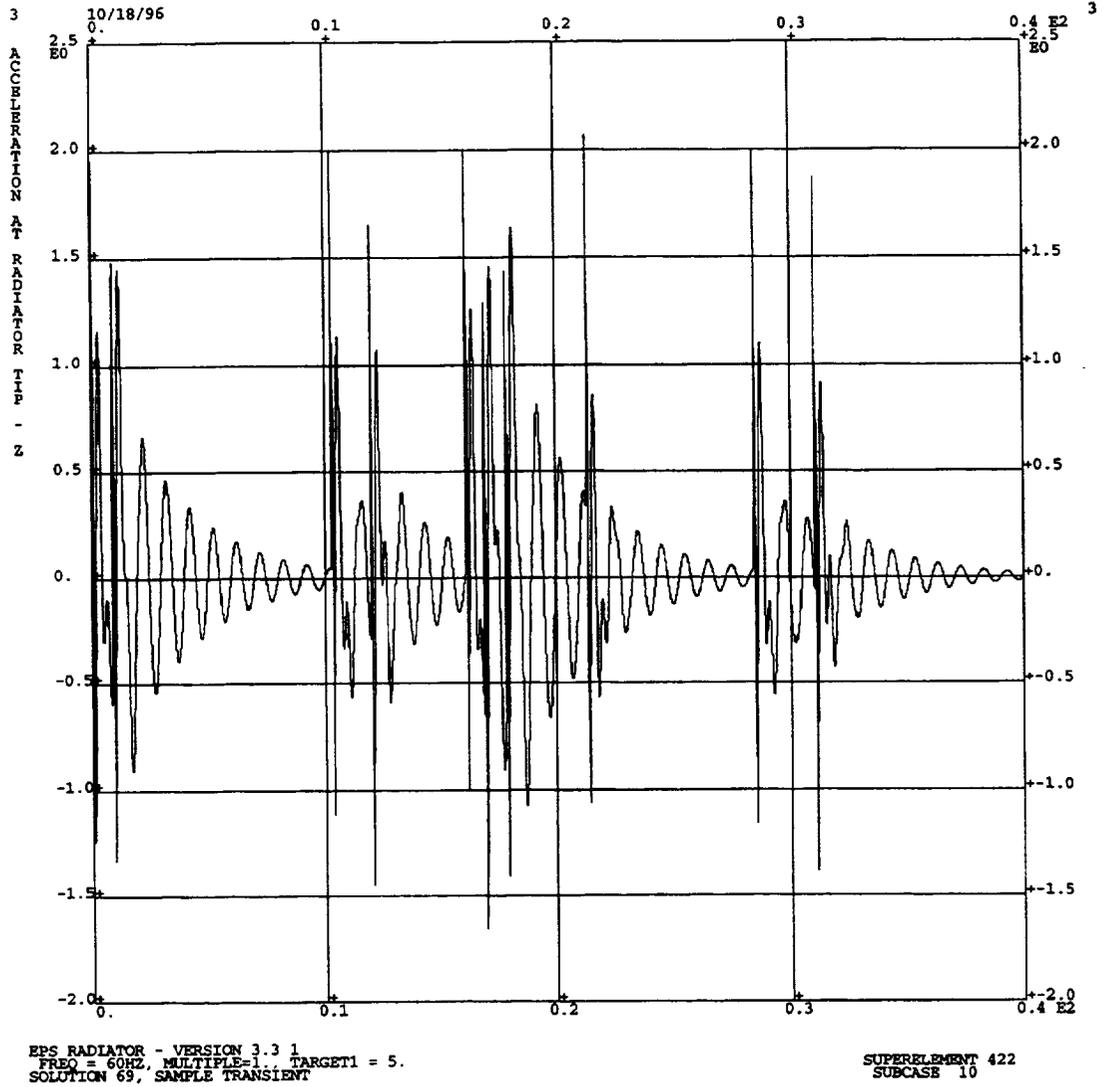


Figure (8.6) - EPS Radiator Z Tip Acceleration  
Generated Using a  $\psi = 1.0$  Ritz Vector Representation

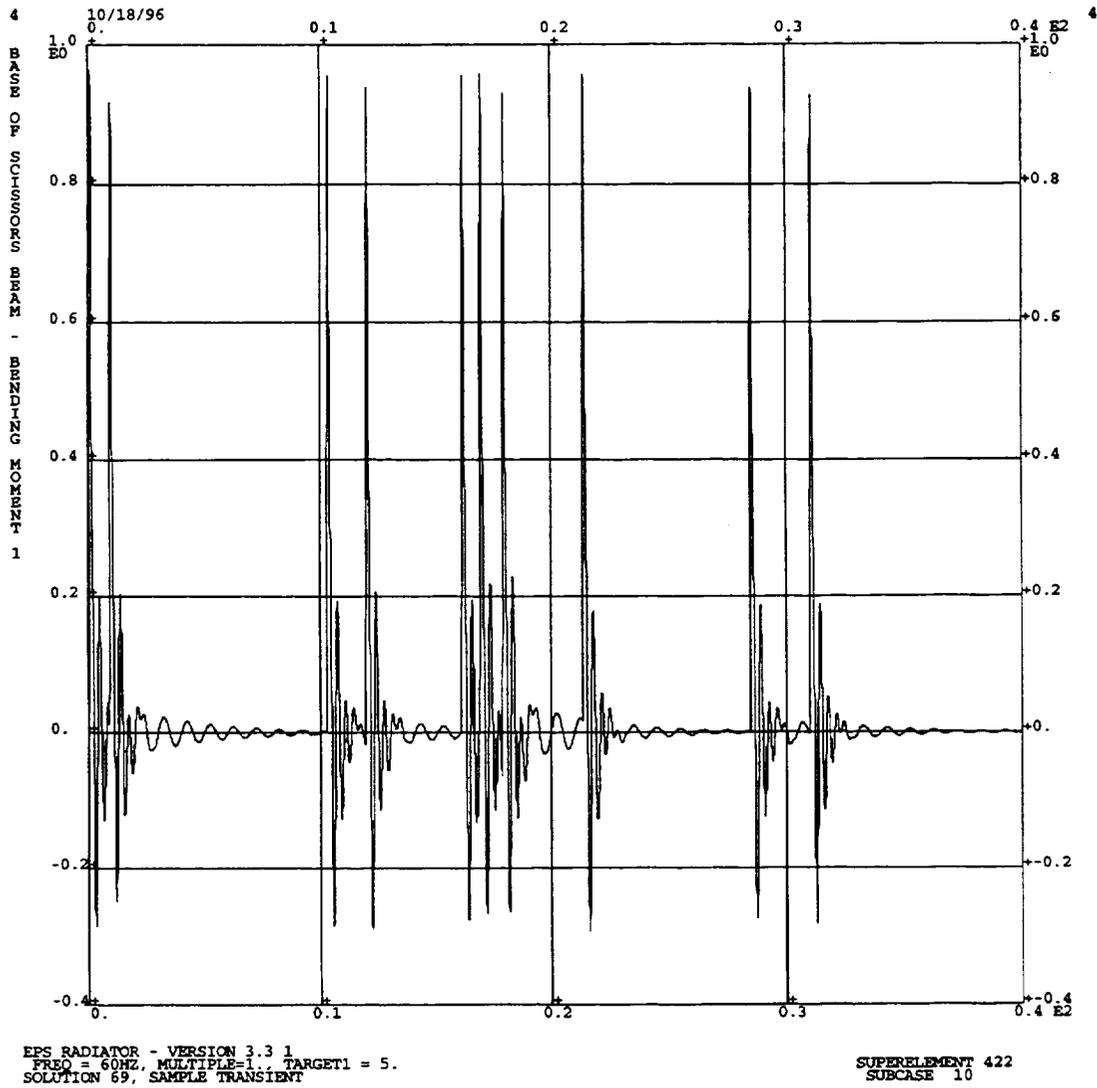


Figure (8.7) - EPS Radiator Scissor Beam Bending Moment 1  
Generated Using a  $\psi = 1.0$  Ritz Vector Representation

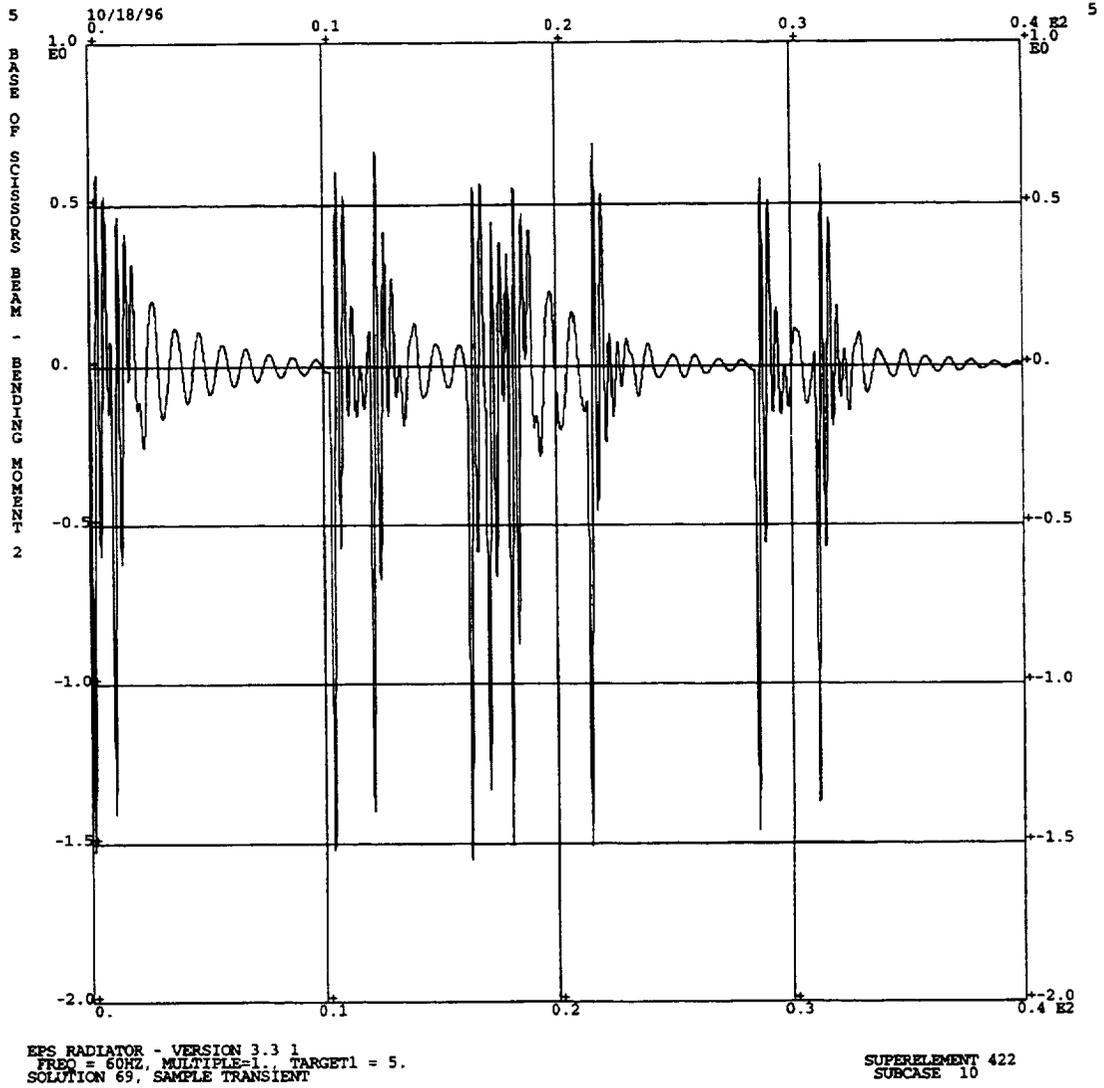


Figure (8.8) - EPS Radiator Scissor Beam Bending Moment 2  
Generated Using a  $\psi = 1.0$  Ritz Vector Representation

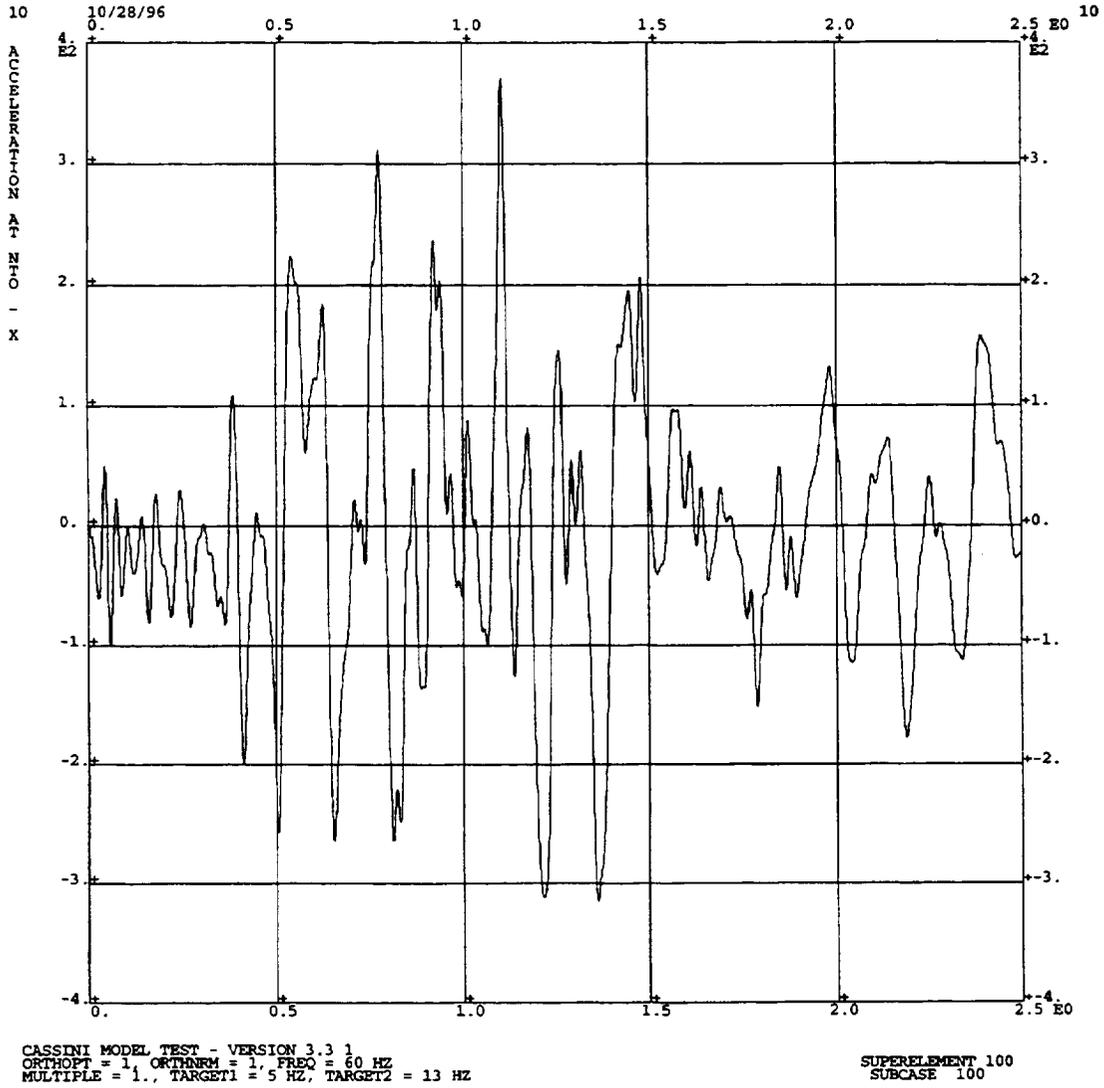


Figure (8.9) - Cassini Oxidizer Tank X Acceleration  
Generated Using a  $\psi = 1.0$  Ritz Vector Representation

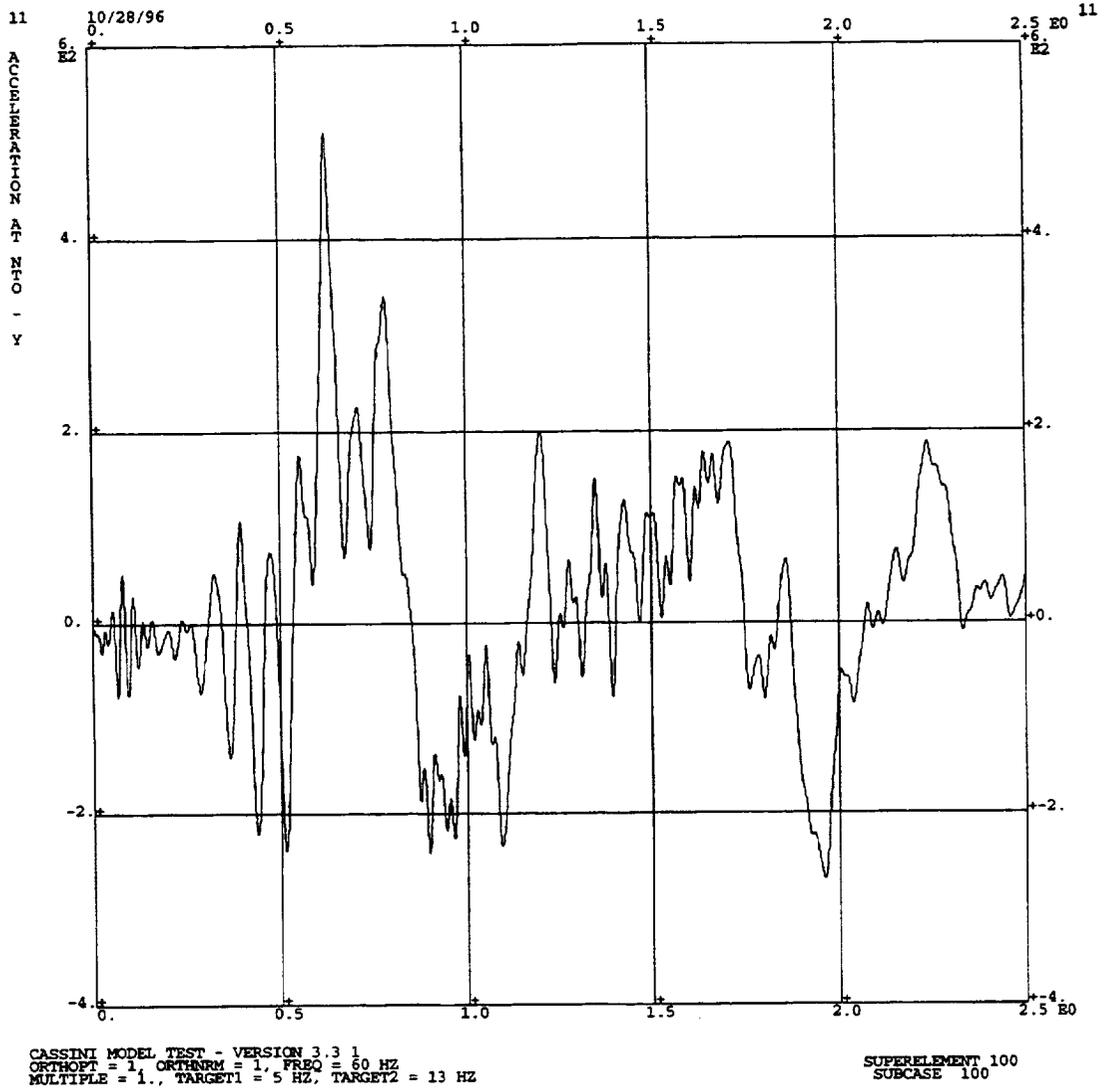


Figure (8.10) - Cassini Oxidizer Tank Y Acceleration  
 Generated Using a  $\psi = 1.0$  Ritz Vector Representation

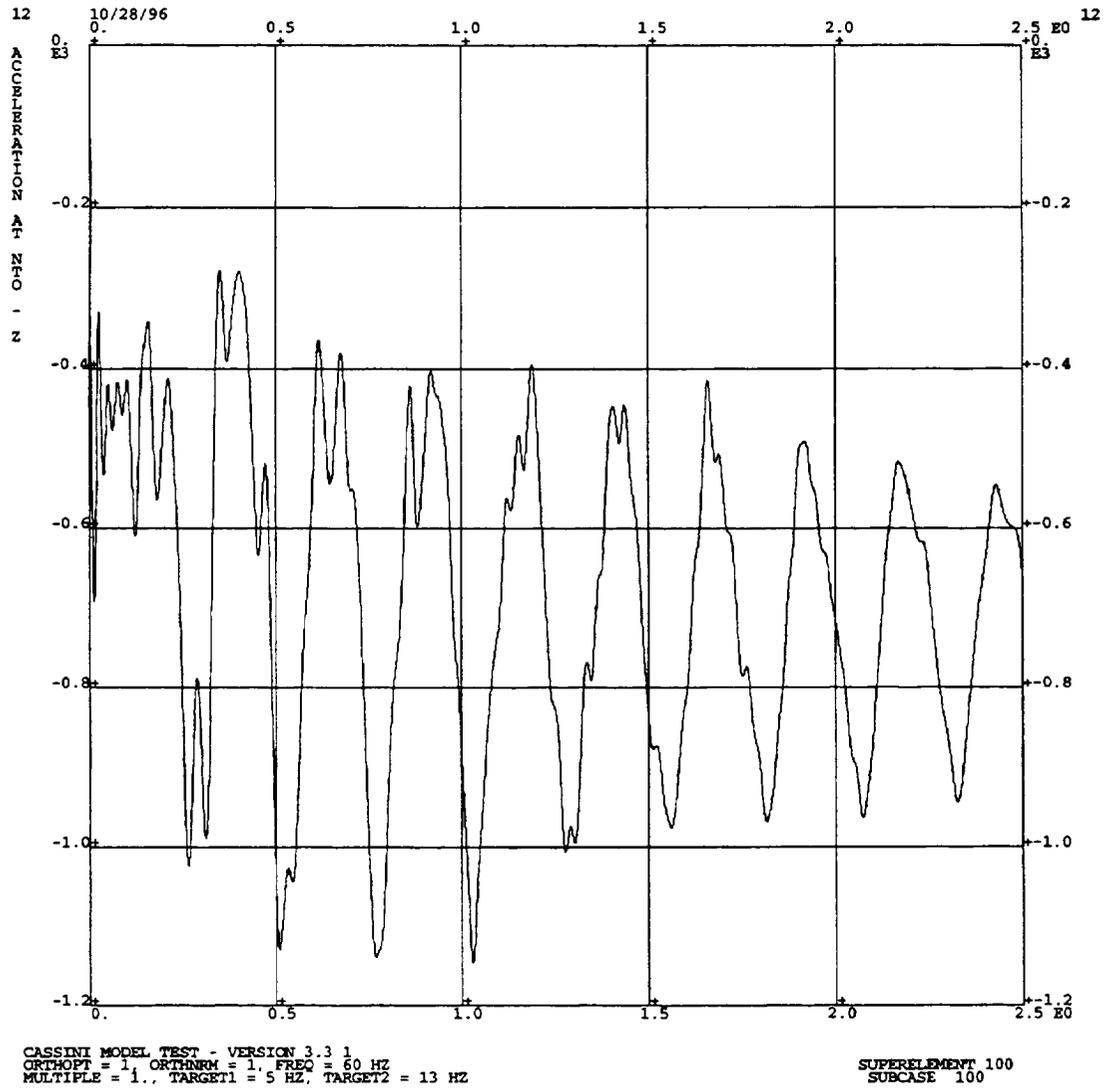


Figure (8.11) - Cassini Oxidizer Tank Z Acceleration  
 Generated Using a  $\psi = 1.0$  Ritz Vector Representation

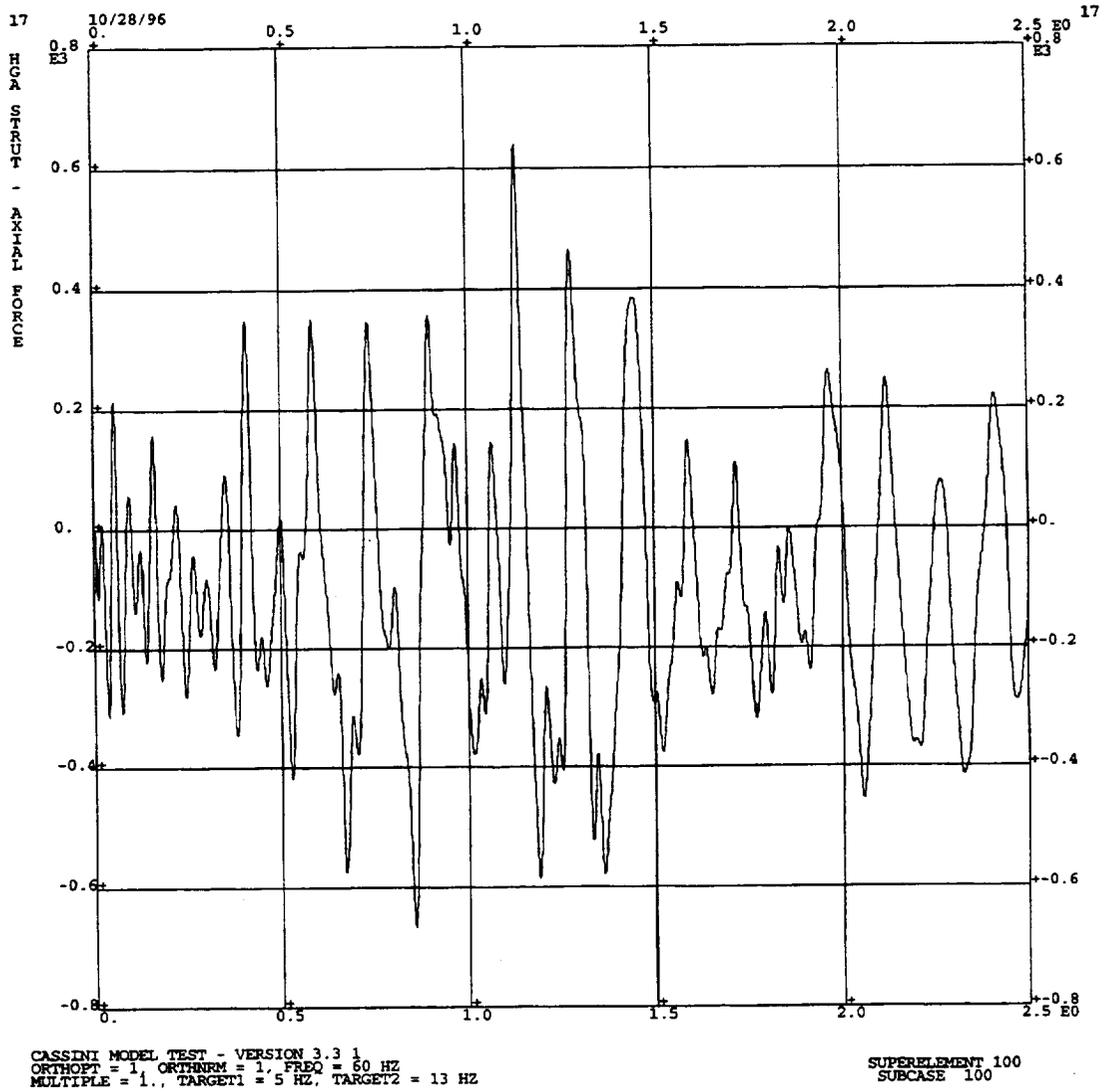


Figure (8.12) - High Gain Antenna Strut Axial Force  
Generated Using a  $\psi = 1.0$  Ritz Vector Representation

## Chapter 9

### Summary and Conclusions

Using the boundary flexibility method to initialize the block-Krylov recurrence algorithm provides an efficient and simple method for generating static Ritz vectors. Static Ritz vectors so generated accurately can represent the dynamics of a substructure. Because this methodology does not require the solution of the component eigenvalue problem, a component can be formed with a significant decrease in computational cost. The component formed using static Ritz vectors can include fewer vectors than the comparable eigensolution, for similar accuracy, and the computational cost of the transient solution is then also reduced.

This dissertation presents new developments in several areas related to static Ritz vector calculation using the boundary flexibility method. It has been shown that the loss of orthogonality, discussed in the literature, is directly related to convergence to an eigenvector in a power extraction method. Orthonormalization using the euclidean norm rather than the mass matrix has been demonstrated. The replacement of Gram-Schmidt with Cholesky/QR orthonormalization has also been demonstrated. These two modifications to the orthonormalization algorithm were developed to decrease the primary computational cost of the block-Krylov sequence.

The Krylov blocks produced by the boundary flexibility method are initially too large for efficient handling and computation. As a result, a method of block filtering has been developed which retains the physically significant information

contained in the vector block, while eliminating the redundant information. Block filtering reduces the block size to that commonly used in commercial Lanczos eigenvalue extraction routines.

The use of inverse operators, or shifting, commonly used in Lanczos eigenvalue extraction routines to search for missing eigenvectors, has been applied to the boundary flexibility method. Shifting alters the Krylov sequence so that vectors near a selected frequency are created. Targeted shifting utilizes the frequency spectrum of the applied dynamic loading to select a targeted frequency. The spatial distribution of the applied dynamic loading can be used in conjunction with targeted shifting to further refine the static Ritz vector creation.

Determining when a sufficient number of static Ritz vectors have been calculated to accurately represent a component has been difficult. Truncation of the Krylov sequence based upon the modal density of the given finite element model has been developed and presented. This heuristic method is based on the observation that dynamically significant static Ritz vectors are calculated early in the Krylov sequence, and that fewer static Ritz vectors than eigenvectors are necessary to accurately represent a component.

Potential future work on the boundary flexibility method of static Ritz vector creation could include the use of synthetic load vectors to supplement the spatial distribution of the applied dynamic load. A synthetic load could be created which would cause a particular element, or sets of elements, to deflect. Applying this load

to the Krylov sequence would create static Ritz vectors which should guarantee accurate response predictions for specific, critical data recovery items. This technique could be investigated with no additional theoretical development.

The shift strategy developed, and the value of the fraction  $\psi$ , could be tested with a larger number of example models. This might allow for either a simplification, or a greater sophistication of the shift strategy. In particular, an alteration of the shift strategy, to allow the use of the applied load and targeted shifting, on models with a number of eigenvectors in the frequency range of interest smaller than the block size, should be investigated.

## References

- 1) Abdallah, A.A. and Huckelbridge, A.A., "Boundary Flexibility Method of Component Mode Synthesis Using Static Ritz Vectors", *Computers & Structures*, Vol. 35, No. 1, 1990.
- 2) Allen, J.J., "A Component Synthesis Method Using Lanczos Vectors", Sixth International Modal Analysis Conference, Vol. 1, 1988.
- 3) Arnold, R., Citerley, R., Chagrin, M., and Galant, D., "Application of Ritz Vectors for Dynamic Analysis of Large Structures", *Computers and Structures*, Vol. 21, No. 3, pp. 461-467, 1985.
- 4) Bronson, R., "*Matrix Operations*", McGraw-Hill, New York, N.Y., 1989.
- 5) Brunty, J.A., "A Transient Response Analysis of the Space Shuttle Vehicle During Liftoff", NASA TM-103505, 1990.
- 6) Carney, K.S., Abdallah, A.A., and Huckelbridge, A.A., "Implementation of the Block-Krylov Boundary Flexibility Method of Component Synthesis", NASA TM-106065, 1993.
- 7) Cook, R.D., "*Concepts and Applications of Finite Element Analysis*", Wiley, New York, N.Y., 1981.
- 8) Craig, R.R., "*Structural Dynamics, An Introduction to Computer Methods*", Wiley, New York, N.Y., 1981.
- 9) Craig, R.R. and Bampton, M.C.C., "Coupling of Substructures for Dynamic Analysis", *AIAA Journal*, Vol. 6, 1968.
- 10) Craig, R.R. and Hale, A.L., "Block-Krylov Component Synthesis Method for Structural Model Reduction", *Journal of Guidance, Control and Dynamics*, Vol. 11, No. 6, 1988.
- 11) Craig, R.R., Su, T.J, and Kim, H.M., "Some Experiences with Krylov Vectors and Lanczos Vectors", Fifth Aerospace Computational Control Conference, 1992.

- 12) Escobedo-Torres, J., and Ricles, J.M., "Improved Dynamic Analysis Method Using Load-Dependent Ritz Vectors", 34th AIAA Structures, Structural Dynamics, and Materials Conference, 1993.
- 13) Grimes, R.G., Lewis, J.G., and Simon, H., "A Shifted Block LANCZOS Algorithm for Solving Sparse Symmetric Generalized Eigenproblems", *AMS-TR-166*, Boeing Computer Services, July 1991.
- 14) Kammer, D.C., Flanigan, C.C., and Dreyer, W., "A Superelement Approach to Test-Analysis Model Development," Fourth International Modal Analysis Conference, Los Angeles, CA., Feb, 1986.
- 15) Lanczos, C., "An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators", *Journal Res. Nat. Bur. Standards*, Sect. B Vol. 45, pp. 225-280, 1950.
- 16) Léger, P., "Application of Load-Dependent Vectors Bases for Dynamic Substructure Analysis", *AIAA Journal*, Vol. 28, No. 1, pp. 177-179, 1990.
- 17) MacNeal, R.H., "A Hybrid Method of Component Mode Synthesis", *Computers and Structures*, Vol. 1, pp. 581-601, 1971.
- 18) Noble, B.N. and Daniel, J.W., "*Applied Linear Algebra*", Prentice-Hall, Englewood Cliffs, N.J., 1988.
- 19) Nour-Omid, B. and Clough R.W., "Dynamic Analysis of Structures Using Lanczos Co-Ordinates", *Earthquake Engineering and Structural Dynamics*, Vol. 12, pp. 565-577, 1984.
- 20) Nour-Omid, B. and Clough R.W., "Block Lanczos Method For Dynamic Analysis of Structures", *Earthquake Engineering and Structural Dynamics*, Vol. 13, pp. 271-275, 1985.
- 21) Paige, C.C., "Practical Use of the Symmetric Lanczos Process with Re-Orthogonalization", *BIT*, Vol. 10, pp.183-195, 1970.
- 22) Parlett, B.N., "*The Symmetric Eigenvalue Problem*", Prentice-Hall, Englewood Cliffs, N.J., 1980.
- 23) Parlett, B.N., and Scott, D.S., "The Lanczos Algorithm With Selective Orthogonalization", *Mathematics of Computation*, Vol. 33, pp. 217-238, 1979.

- 24) Rubin S., "Improved Component-Mode Representation for Structural Dynamic Analysis", *AIAA Journal*, Vol. 13, No. 8, 1975.
- 25) Scott, D.S., "The Advantages of Inverted Operators in Rayleigh-Ritz Approximations", *SIAM Journal of Sci. Stat. Computations*, Vol. 3, No. 1, pp. 68-75, March 1982.
- 26) Simon, H.D., "The Lanczos Algorithm for Solving Symmetric Linear Systems", *Technical Report PAM-74*, University of California, Berkeley, 1982.
- 27) Su, T.J. and Craig R.R., "Model Reduction and Control of Flexible Structures Using Krylov Vectors", *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 2, 1991.
- 28) Wilson, E.L. and Bayo, E.P., "Use of Special Ritz Vectors in Dynamic Substructure Analysis", *ASCE Journal of Structural Engineering*, Vol. 112, No. 8, 1986.
- 29) Wilson, E.L., Yuan, M.W. and Dickens, J.M., "Dynamic Analysis by Direct Superposition of Ritz Vectors", *Earthquake Engineering Structural Dynamics*, Vol. 10, 1982.
- 30) Yiu, Y.C. and Landess, J.D., "A Reduced Order Method for Passive Substructures", 31th AIAA Structures, Structural Dynamics, and Materials Conference, 1990.
- 31) "*MSC/NASTRAN Handbook for Numerical Methods*", Komzsik, L., Editor, The MacNeal-Schwendler Corporation, Los Angeles, CA, April 1990.
- 32) "*MSC/NASTRAN User's Manual, Version 67*", The MacNeal-Schwendler Corporation, Los Angeles, CA, August 1991.

Appendix A1 - EPS Radiator Finite Element Model Effective Masses

Mode No.	Frequency (Hz)	Effective Masses (%)					
		T1	T2	T3	R1	R2	R3
1	0.19	0.0	0.0	46.8	93.7	35.8	0.0
2	0.70	0.1	0.0	0.0	0.0	11.2	0.1
3	0.91	43.4	0.0	0.0	0.0	1.0	85.0
4	1.20	0.0	35.7	0.1	0.0	0.1	2.2
5	1.30	0.0	0.3	12.3	3.6	9.4	0.0
6	2.59	0.4	0.0	0.0	0.0	1.6	0.2
7	3.32	19.2	0.0	0.0	0.0	0.6	6.1
8	3.65	0.0	0.1	3.7	0.7	2.8	0.0
9	5.15	0.0	9.9	0.0	0.0	0.0	0.6
10	5.78	0.2	0.0	0.0	0.0	0.6	0.0
11	6.73	0.0	0.5	1.7	0.3	1.3	0.0
12	6.88	4.3	0.0	0.0	0.0	0.2	0.4
13	8.92	0.0	0.5	0.4	0.0	0.3	0.0
14	9.72	0.9	0.0	0.0	0.0	0.0	0.1
15	9.96	0.0	1.0	1.2	0.2	0.9	0.1
16	10.33	0.8	0.0	0.0	0.0	0.5	0.1
17	10.72	0.0	0.2	5.3	0.5	4.0	0.0
18	12.08	1.1	0.0	0.0	0.0	0.1	0.1
19	13.15	0.0	0.0	0.0	0.0	0.0	0.0
20	14.51	0.8	0.0	0.0	0.0	0.0	0.1
21	15.05	0.0	0.0	0.0	0.0	0.2	0.0
22	16.34	0.7	0.0	0.0	0.0	0.0	0.0
23	16.34	0.0	0.0	0.0	0.0	0.0	0.0
24	16.91	0.0	0.0	0.0	0.0	0.1	0.0
25	17.83	0.7	0.0	0.0	0.0	0.0	0.0
26	18.71	0.7	0.0	0.0	0.0	0.0	0.0
27	19.15	0.0	1.2	0.1	0.0	0.1	0.1
28	20.25	0.0	0.3	0.0	0.0	0.0	0.0
29	21.03	0.1	0.0	0.0	0.0	1.2	0.0
30	22.80	0.0	0.0	0.0	0.0	0.0	0.0
31	22.94	0.0	0.1	0.0	0.0	0.0	0.0
32	23.13	0.0	0.0	0.0	0.0	0.0	0.0
33	24.03	0.0	0.0	0.0	0.0	0.0	0.0
34	24.04	0.0	0.0	0.0	0.0	0.0	0.0

35	24.80	0.0	0.0	0.0	0.0	0.0	0.0
36	25.52	0.0	0.5	0.1	0.0	0.1	0.0
37	25.64	0.0	0.1	0.0	0.0	0.0	0.0
38	25.75	0.0	0.0	0.0	0.0	0.0	0.0
39	26.23	0.0	0.0	0.0	0.0	0.0	0.0
40	26.34	0.0	0.0	0.0	0.0	0.0	0.0
41	27.01	0.0	1.1	0.5	0.0	0.4	0.1
42	27.43	0.0	0.0	0.0	0.0	0.0	0.0
43	27.69	0.0	0.0	0.0	0.0	0.0	0.0
44	28.19	0.0	0.0	0.1	0.0	0.1	0.0
45	28.46	0.0	0.3	0.0	0.0	0.0	0.0
46	28.91	0.0	1.7	0.8	0.1	0.6	0.1
47	29.61	0.0	0.0	0.0	0.0	0.0	0.0
48	30.23	0.0	0.0	0.5	0.0	0.4	0.0
49	30.88	0.0	0.0	0.0	0.0	0.0	0.0
50	30.89	0.0	0.0	0.0	0.0	0.0	0.0
51	31.13	0.0	2.2	0.0	0.0	0.0	0.1
52	33.69	0.0	0.0	0.0	0.0	0.0	0.0
53	33.79	0.0	0.4	0.0	0.0	0.0	0.0
54	34.35	0.0	2.1	0.0	0.0	0.0	0.1
55	34.71	0.0	0.0	0.0	0.0	0.0	0.0
56	36.57	0.0	0.3	0.0	0.0	0.0	0.0
57	37.37	0.0	0.0	0.0	0.0	0.0	0.0
58	37.51	0.0	1.6	0.0	0.0	0.0	0.1
59	37.53	0.0	0.0	0.0	0.0	0.0	0.0
60	38.26	0.0	0.4	0.0	0.0	0.0	0.0
61	38.61	0.0	0.1	0.0	0.0	0.0	0.0
62	39.79	0.0	0.0	0.0	0.0	0.1	0.1
63	39.83	0.0	0.0	0.0	0.0	0.0	0.0
64	40.88	0.0	9.5	0.0	0.0	0.0	0.6
65	41.08	0.0	0.2	0.0	0.0	0.0	0.0
66	42.45	0.0	0.0	0.0	0.0	0.0	0.0
67	44.29	0.0	0.0	0.0	0.0	0.0	0.0
68	44.52	0.0	0.0	0.0	0.0	0.0	0.0
69	45.73	0.0	0.0	0.0	0.0	0.0	0.0
70	46.20	0.0	0.0	0.0	0.0	0.0	0.0
71	46.53	0.0	0.0	0.0	0.0	0.0	0.0
72	46.74	0.0	2.6	0.1	0.0	0.0	0.2
73	48.69	0.0	0.3	0.0	0.0	0.0	0.0
74	49.37	0.0	1.8	0.0	0.0	0.0	0.1
75	49.70	0.0	0.0	0.0	0.0	0.0	0.0

76	50.80	0.0	0.2	0.0	0.0	0.0	0.0
77	51.21	0.0	0.6	0.0	0.0	0.0	0.0
78	51.81	0.0	0.0	0.0	0.0	0.0	0.2
79	52.04	0.0	0.5	0.0	0.0	0.0	0.0
80	53.69	0.0	1.7	0.1	0.0	0.0	0.1
81	55.50	0.1	0.0	0.0	0.0	0.0	0.1
82	56.10	0.0	0.0	0.0	0.0	0.0	0.0
83	56.65	0.0	4.7	0.0	0.0	0.0	0.3
84	56.97	0.1	0.1	0.0	0.0	0.0	0.0
85	57.69	0.0	0.0	0.0	0.0	0.0	0.0
86	58.28	0.0	0.1	0.0	0.0	0.0	0.0
87	59.33	0.0	0.5	0.0	0.0	0.0	0.0
88	60.24	0.0	0.0	0.0	0.0	0.0	0.0
89	61.50	0.0	0.6	0.0	0.0	0.0	0.0
90	62.01	0.0	0.8	0.1	0.0	0.1	0.0
91	62.45	0.0	0.9	0.0	0.0	0.0	0.1
92	63.12	0.0	0.0	0.0	0.0	0.0	0.0
93	63.86	0.0	0.2	0.0	0.0	0.0	0.0
94	64.84	0.0	0.0	0.0	0.0	0.0	0.0
95	67.15	0.0	0.5	0.0	0.0	0.0	0.0
96	70.52	0.0	0.0	0.0	0.0	0.0	0.0
97	70.53	0.0	0.0	0.4	0.0	0.3	0.0
98	71.34	0.0	0.0	0.0	0.0	0.0	0.0
99	73.06	0.0	0.0	0.0	0.0	0.0	0.0
100	74.40	0.0	0.0	0.0	0.0	0.0	0.0
101	75.66	0.0	0.0	0.0	0.0	0.0	0.0
102	76.92	0.0	0.0	0.0	0.0	0.0	0.0
103	79.49	0.0	0.0	0.0	0.0	0.0	0.0
104	79.89	0.0	0.0	0.0	0.0	0.0	0.0
105	80.46	0.0	0.0	0.0	0.0	0.0	0.0
106	82.72	0.0	0.0	0.0	0.0	0.0	0.0
107	83.36	0.0	0.0	0.0	0.0	0.0	0.0
108	84.53	0.0	0.0	0.0	0.0	0.0	0.0
109	85.43	0.0	1.5	0.0	0.0	0.0	0.1
110	86.57	0.0	0.0	0.0	0.0	0.0	0.0
111	87.00	0.0	0.0	0.0	0.0	0.0	0.0
112	88.86	0.0	0.0	0.0	0.0	0.0	0.0
113	89.31	0.0	0.1	0.0	0.0	0.0	0.0
114	89.90	0.0	0.0	0.0	0.0	0.0	0.0
115	92.33	0.0	0.0	0.0	0.0	0.0	0.0
116	92.75	0.0	0.0	0.0	0.0	0.0	0.0

117	94.21	0.0	0.0	0.0	0.0	0.0	0.0
118	96.19	0.0	0.0	0.0	0.0	0.0	0.0
119	96.53	0.1	0.0	0.0	0.0	0.0	0.0
120	98.07	1.0	0.1	0.0	0.0	0.1	0.1
121	98.77	0.0	0.0	0.0	0.0	0.0	0.0
122	99.72	0.1	0.0	0.0	0.0	0.0	0.0
123	100.06	0.0	0.0	0.0	0.0	0.0	0.0
124	100.19	0.0	0.0	0.0	0.0	0.0	0.0
125	100.46	0.1	0.0	0.0	0.0	0.0	0.0
126	101.95	0.0	0.0	0.0	0.0	0.0	0.0
127	102.02	0.0	0.0	0.0	0.0	0.0	0.0
128	102.69	0.0	0.0	0.0	0.0	0.0	0.0
129	103.97	0.0	0.0	0.0	0.0	0.0	0.0
130	104.88	0.0	0.1	0.5	0.0	0.4	0.0
131	106.50	0.0	0.2	0.1	0.0	0.1	0.0
132	107.69	0.0	0.7	0.0	0.0	0.0	0.1
133	107.93	0.1	0.6	0.0	0.0	0.0	0.0
134	108.61	0.0	0.0	0.0	0.0	0.0	0.0
135	108.78	0.0	0.0	0.0	0.0	0.0	0.0
136	109.35	0.0	0.1	0.0	0.0	0.0	0.0
137	110.03	0.0	0.0	0.0	0.0	0.0	0.0
138	111.76	0.0	0.0	0.0	0.0	0.0	0.0
139	111.93	0.0	0.0	0.0	0.0	0.0	0.0
140	113.42	0.0	0.0	0.0	0.0	0.0	0.0
141	113.43	0.0	0.0	0.0	0.0	0.0	0.0
142	113.70	0.0	0.0	0.0	0.0	0.0	0.0
143	113.70	0.0	0.0	0.0	0.0	0.0	0.0
144	114.31	0.0	0.0	0.0	0.0	0.0	0.0
145	114.41	0.0	0.0	0.0	0.0	0.0	0.0
146	115.00	0.0	0.0	0.0	0.0	0.0	0.0
147	115.16	0.0	0.0	0.0	0.0	0.0	0.0
148	116.00	0.0	0.0	0.0	0.0	0.0	0.0
149	117.72	0.0	0.0	0.0	0.0	0.0	0.0
150	117.95	0.2	0.0	0.0	0.0	0.0	0.0
151	118.99	0.1	0.0	0.0	0.0	0.0	0.0
152	121.38	0.0	0.0	0.0	0.0	0.0	0.0
153	122.25	0.0	0.0	0.0	0.0	0.0	0.0
154	123.42	0.0	0.0	0.0	0.0	0.0	0.0
155	123.68	0.0	0.0	0.0	0.0	0.0	0.0
156	124.66	0.0	0.0	0.0	0.0	0.0	0.0
157	125.34	0.0	0.0	0.0	0.0	0.0	0.0

158	125.37	0.0	0.0	0.0	0.0	0.0	0.0
159	126.61	0.0	0.0	0.0	0.0	0.0	0.0
160	127.27	0.0	0.0	0.0	0.0	0.0	0.0
161	128.22	1.0	0.0	0.0	0.0	0.0	0.0
162	128.68	0.6	0.0	0.0	0.0	0.0	0.0
163	129.60	0.2	0.0	0.0	0.0	0.0	0.0
164	131.13	0.0	0.0	0.0	0.0	0.0	0.0
165	131.41	0.2	0.0	0.0	0.0	0.0	0.0
166	133.15	0.0	0.0	0.0	0.0	0.0	0.0
167	133.57	0.0	0.0	0.0	0.0	0.0	0.0
168	133.86	0.2	0.0	0.0	0.0	0.0	0.0
169	134.39	0.0	0.0	0.0	0.0	0.0	0.0
170	135.99	0.0	0.0	0.0	0.0	0.0	0.0
171	136.67	0.0	0.0	0.3	0.0	0.2	0.0
172	137.31	0.0	0.0	0.0	0.0	0.0	0.0
173	137.41	0.0	0.0	0.0	0.0	0.0	0.0
174	138.89	0.0	0.0	0.0	0.0	0.0	0.0
175	139.59	0.0	0.0	0.0	0.0	0.0	0.0

Total Effective Mass (%)					
T1	T2	T3	R1	R2	R3
78.1	90.3	75.6	99.4	75.7	98.8

Appendix A2 - Cassini Finite Element Model Effective Masses

Mode No.	Frequency (Hz)	Effective Masses (%)					
		T1	T2	T3	R1	R2	R3
1	7.36	49.2	8.6	0.0	13.2	78.7	0.0
2	7.70	7.7	50.9	0.0	78.8	12.5	0.0
3	12.19	0.0	0.0	0.0	0.0	0.0	0.2
4	14.89	0.0	0.0	0.0	0.0	0.0	0.0
5	14.91	0.0	0.0	0.0	0.0	0.0	0.0
6	15.50	0.0	0.0	0.0	0.0	0.0	13.9
7	15.75	0.0	0.2	0.0	0.0	0.0	60.5
8	15.88	0.0	0.0	0.0	0.0	0.0	0.0
9	15.90	0.0	0.0	0.0	0.0	0.0	0.0
10	17.85	0.3	0.0	5.2	0.0	0.1	0.0
11	18.28	0.0	0.7	0.0	0.1	0.0	1.0
12	19.01	3.0	0.2	0.4	0.0	0.7	2.2
13	19.08	4.3	0.0	0.5	0.0	0.9	0.8
14	19.21	0.1	0.9	0.0	0.2	0.0	0.0
15	19.54	1.7	0.1	5.2	0.0	0.6	0.2
16	19.81	0.5	0.0	0.5	0.0	0.1	0.4
17	19.96	0.0	0.2	0.0	0.1	0.0	0.0
18	20.04	0.0	1.7	0.0	0.6	0.0	0.5
19	20.34	0.4	2.0	4.1	0.3	0.1	0.9
20	20.47	1.4	1.8	3.5	0.2	0.3	1.2
21	20.63	1.2	0.3	0.0	0.0	0.2	0.1
22	21.39	5.4	0.0	1.1	0.0	1.3	0.0
23	23.44	0.0	0.0	0.0	0.0	0.0	0.7
24	23.73	1.5	0.0	0.8	0.0	0.3	0.0
25	25.55	0.0	9.0	0.4	2.5	0.0	3.4
26	25.94	1.0	0.0	0.1	0.0	0.1	0.0
27	26.54	0.0	0.6	0.1	0.2	0.0	2.5
28	27.14	1.0	0.0	0.0	0.0	0.2	0.0
29	27.23	0.0	0.0	0.1	0.0	0.0	0.1
30	27.45	0.0	0.0	0.0	0.0	0.0	0.0
31	27.52	0.0	0.0	0.2	0.0	0.0	0.0
32	28.33	1.7	0.1	14.2	0.1	0.5	0.1
33	29.60	0.3	0.6	2.8	0.1	0.1	0.0
34	29.70	0.5	1.1	2.3	0.2	0.1	0.0

35	31.27	1.9	2.9	4.6	0.5	0.4	0.1
36	31.51	0.0	0.1	2.1	0.0	0.0	0.0
37	33.53	0.4	0.0	1.9	0.0	0.1	0.0
38	34.49	2.6	0.0	0.0	0.0	0.5	0.0
39	34.99	0.0	0.0	0.0	0.0	0.0	0.1
40	35.23	0.3	0.0	0.3	0.0	0.1	0.0
41	36.45	1.2	3.7	0.5	0.6	0.2	0.0
42	36.73	2.7	1.4	0.1	0.3	0.5	0.1
43	37.53	0.8	2.1	15.6	0.4	0.2	0.1
44	38.04	0.0	1.1	1.6	0.2	0.0	0.0
45	38.29	0.0	0.1	19.6	0.0	0.0	0.0
46	38.53	0.1	0.0	0.1	0.0	0.0	0.1
47	39.26	0.2	0.7	0.0	0.1	0.0	0.2
48	39.39	1.2	0.0	0.5	0.0	0.3	0.1
49	40.65	0.0	3.0	0.0	0.5	0.0	0.0
50	41.70	0.4	0.2	0.2	0.0	0.1	0.0
51	41.97	0.2	0.1	0.4	0.0	0.0	0.0
52	42.69	0.2	0.1	0.1	0.0	0.0	0.1
53	43.38	1.6	0.0	0.1	0.0	0.2	0.1
54	44.23	0.1	0.0	0.5	0.0	0.0	0.0
55	44.58	0.0	0.0	0.1	0.0	0.0	0.0
56	44.98	0.1	0.0	1.3	0.0	0.0	0.0
57	45.11	0.2	0.0	0.9	0.0	0.0	0.0
58	46.03	0.0	1.0	0.1	0.2	0.0	0.0
59	47.12	0.0	0.0	0.0	0.0	0.0	0.3
60	47.71	0.0	0.0	0.2	0.0	0.0	0.0
61	48.40	0.1	0.1	1.3	0.0	0.0	0.0
62	48.89	0.0	0.0	0.0	0.0	0.0	0.0
63	49.13	0.2	0.0	0.1	0.0	0.0	0.7
64	50.11	0.0	0.0	0.0	0.0	0.0	0.0
65	50.11	0.0	0.0	0.0	0.0	0.0	0.0
66	50.12	0.2	0.0	0.0	0.0	0.0	0.0
67	50.93	0.0	0.0	0.0	0.0	0.0	0.0
68	52.72	0.0	0.0	0.0	0.0	0.0	0.0
69	53.08	0.2	0.1	0.0	0.0	0.0	0.0
70	53.31	0.2	0.0	0.0	0.0	0.0	0.0
71	54.10	0.0	0.2	0.0	0.0	0.0	0.0
72	55.12	0.1	0.0	0.0	0.0	0.0	0.0
73	56.42	0.0	0.0	0.1	0.0	0.0	0.0
74	57.31	0.0	0.0	0.0	0.0	0.0	0.0
75	58.35	0.0	0.0	0.1	0.0	0.0	0.0

76	58.75	0.0	0.1	0.1	0.0	0.0	0.0
77	59.10	0.0	0.0	0.0	0.0	0.0	0.0
78	59.12	0.0	0.0	0.0	0.0	0.0	0.0
79	59.12	0.0	0.0	0.0	0.0	0.0	0.0
80	59.13	0.0	0.0	0.0	0.0	0.0	0.0
81	59.30	0.0	0.0	0.0	0.0	0.0	0.0
82	59.72	0.0	0.0	0.0	0.0	0.0	0.0
83	59.93	0.0	0.0	0.0	0.0	0.0	0.0
84	60.32	0.0	0.0	0.1	0.0	0.0	0.0
85	61.40	0.0	0.0	0.0	0.0	0.0	0.0
86	62.01	0.0	0.0	0.3	0.0	0.0	0.0
87	62.77	0.0	0.0	0.0	0.0	0.0	0.0
88	63.21	0.0	0.0	0.0	0.0	0.0	0.0
89	64.52	0.0	0.0	0.3	0.0	0.0	0.0
90	65.18	0.0	0.0	0.0	0.0	0.0	0.0
91	67.86	0.0	0.0	0.0	0.0	0.0	0.0
92	67.88	0.0	0.0	0.0	0.0	0.0	0.0
93	67.88	0.0	0.0	0.0	0.0	0.0	0.0
94	67.88	0.0	0.0	0.0	0.0	0.0	0.0
95	67.91	0.0	0.0	0.0	0.0	0.0	0.0
96	67.94	0.0	0.0	0.0	0.0	0.0	0.0
97	68.75	0.0	0.0	0.0	0.0	0.0	0.0
98	69.50	0.0	0.0	0.1	0.0	0.0	0.0
99	69.86	0.0	0.0	0.0	0.0	0.0	0.1
100	71.31	0.0	0.0	0.0	0.0	0.0	0.0
101	72.05	0.0	0.0	0.1	0.0	0.0	0.0
102	73.13	0.0	0.0	0.8	0.0	0.0	0.0
103	75.57	0.0	0.0	0.0	0.0	0.0	0.0
104	75.57	0.0	0.0	0.0	0.0	0.0	0.0
105	75.59	0.0	0.0	0.0	0.0	0.0	0.0
106	75.59	0.0	0.0	0.0	0.0	0.0	0.0
107	75.64	0.0	0.0	0.0	0.0	0.0	0.0
108	75.68	0.0	0.0	0.0	0.0	0.0	0.0
109	75.84	0.0	0.0	0.0	0.0	0.0	0.0
110	76.07	0.0	0.0	0.0	0.0	0.0	0.2
111	76.43	0.0	0.0	0.0	0.0	0.0	0.0
112	76.81	0.0	0.0	0.0	0.0	0.0	0.0
113	78.81	0.0	0.0	0.0	0.0	0.0	0.5
114	79.64	0.0	0.0	0.0	0.0	0.0	0.1
115	80.36	0.0	0.0	0.0	0.0	0.0	0.3
116	82.57	0.1	0.0	0.0	0.0	0.0	0.3

117	83.03	0.0	0.0	0.3	0.0	0.0	0.0
118	83.48	0.0	0.0	0.0	0.0	0.0	0.0
119	84.86	0.0	0.0	0.1	0.0	0.0	0.0
120	85.93	0.0	0.0	0.0	0.0	0.0	0.0
121	86.70	0.0	0.2	0.0	0.0	0.0	0.0
122	87.09	0.0	0.0	0.6	0.0	0.0	0.0
123	87.80	0.0	0.0	0.0	0.0	0.0	0.0
124	88.35	0.0	0.0	0.0	0.0	0.0	0.0
125	88.41	0.0	0.0	0.1	0.0	0.0	0.0
126	88.81	0.2	0.0	0.3	0.0	0.0	0.0
127	90.55	0.0	0.0	0.0	0.0	0.0	0.0
128	90.56	0.1	0.0	0.1	0.0	0.0	0.0
129	90.70	0.0	0.0	0.0	0.0	0.0	0.0
130	91.26	0.0	0.0	0.0	0.0	0.0	0.0
131	91.73	0.0	0.0	0.0	0.0	0.0	0.0
132	92.47	0.0	0.0	0.0	0.0	0.0	0.0
133	92.88	0.0	0.0	0.0	0.0	0.0	0.0
134	93.82	0.0	0.0	0.0	0.0	0.0	0.0
135	94.24	0.0	0.0	0.0	0.0	0.0	0.0
136	94.37	0.1	0.1	0.0	0.0	0.0	0.0
137	94.99	0.1	0.0	0.0	0.0	0.0	0.0
138	95.03	0.0	0.0	0.1	0.0	0.0	0.0
139	96.00	0.1	0.0	0.0	0.0	0.0	0.1
140	96.25	0.0	0.1	0.0	0.0	0.0	0.0
141	97.55	0.0	0.2	0.1	0.0	0.0	0.0
142	99.16	0.1	0.0	0.0	0.0	0.0	0.0
143	100.37	0.0	0.0	0.0	0.0	0.0	0.1
144	101.77	0.0	0.0	0.0	0.0	0.0	0.0
145	101.93	0.0	0.0	0.0	0.0	0.0	0.0
146	103.05	0.0	0.0	0.1	0.0	0.0	0.0
147	104.03	0.0	0.0	0.0	0.0	0.0	0.0
148	104.08	0.0	0.0	0.0	0.0	0.0	0.0
149	104.62	0.0	0.0	0.0	0.0	0.0	0.0
150	106.66	0.0	0.0	0.0	0.0	0.0	0.0
151	106.66	0.0	0.0	0.0	0.0	0.0	0.0
152	106.69	0.0	0.0	0.0	0.0	0.0	0.0
153	107.17	0.0	0.0	0.0	0.0	0.0	0.0
154	107.17	0.0	0.0	0.0	0.0	0.0	0.0
155	107.85	0.0	0.0	0.0	0.0	0.0	0.0

174

156	109.30	0.0	0.0	0.0	0.0	0.0	0.5
157	109.53	0.0	0.0	0.0	0.0	0.0	0.2

Total Effective Mass (%)

T1	T2	T3	R1	R2	R3
97.8	97.8	97.8	99.8	99.8	93.8

## Appendix B

### M I N / M A X   S U M M A R Y

STRUCTURE: beam		PARAM2: eigen		PARAM3: 250 HZ		RESPONSE: ALL	
ID		VALUE	OCCURRENCE	VALUE	OCCURRENCE	VALUE	OCCURRENCE
ACCE	10	3	-2.25390E+01	.9300	1.79300E+01	.0100	
ACCE	10	4	-7.88126E-01	.9200	8.04331E-01	.0000	
ACCE	10	5	-2.25390E+01	.9300	1.79300E+01	.0100	
EL FOR	1	4	-2.61165E+03	1.8250	2.54222E+03	1.0250	
EL FOR	1	5	-2.54222E+03	1.0250	2.61165E+03	1.8250	
EL FOR	1	6	-3.19838E+00	1.8200	3.31442E+00	1.0400	
EL FOR	1	7	-3.31442E+00	1.0400	3.19838E+00	1.8200	
EL FOR	1	8	-2.29287E-01	.9450	1.27743E+00	.0250	
EL FOR	1	9	-4.65317E-10	.9450	2.67230E-08	.0250	

## M I N / M A X S U M M A R Y

STRUCTURE: beam		PARAM2: eigen		PARAM3: 100 HZ		RESPONSE: ALL	
	ID	VALUE		OCCURRENCE		VALUE	OCCURRENCE
ACCE	10	3	-2.24629E+01	.9300		1.78509E+01	.0100
ACCE	10	4	-6.83512E-01	.9200		6.97566E-01	.0000
ACCE	10	5	-2.24629E+01	.9300		1.78509E+01	.0100
EL FOR	1	4	-2.61165E+03	1.8250		2.54222E+03	1.0250
EL FOR	1	5	-2.54222E+03	1.0250		2.61165E+03	1.8250
EL FOR	1	6	-3.19838E+00	1.8200		3.31442E+00	1.0400
EL FOR	1	7	-3.31442E+00	1.0400		3.19838E+00	1.8200
EL FOR	1	8	-2.28109E-01	.9450		1.08525E+00	.0250
EL FOR	1	9	-2.13987E-11	.1400		1.09748E-13	1.0800

## M I N / M A X S U M M A R Y

STRUCTURE: beam		PARAM2: ritz		PARAM3: frac1.0		RESPONSE: ALL	
	ID	VALUE		OCCURRENCE		VALUE	OCCURRENCE
ACCE	10	3	-2.24504E+01	.9300		1.78359E+01	.0100
ACCE	10	4	-8.72979E-01	.9200		8.90928E-01	.0000
ACCE	10	5	-2.24504E+01	.9300		1.78359E+01	.0100
EL FOR	1	4	-2.61165E+03	1.8250		2.54221E+03	1.0250
EL FOR	1	5	-2.54221E+03	1.0250		2.61165E+03	1.8250
EL FOR	1	6	-3.19838E+00	1.8200		3.31401E+00	1.0400
EL FOR	1	7	-3.31401E+00	1.0400		3.19838E+00	1.8200
EL FOR	1	8	-2.17777E-01	.9450		1.12482E+00	.0250
EL FOR	1	9	0.00000E+00	.0000		0.00000E+00	.0000

## M I N / M A X S U M M A R Y

STRUCTURE: beam	PARAM2: ritz	PARAM3: frac.5	RESPONSE: ALL	
ID	VALUE	OCCURRENCE	VALUE	OCCURRENCE
ACCE 10	3	-1.98163E+01	.9300	1.52155E+01 .0100
ACCE 10	4	-6.70055E-01	.0200	6.68161E-01 .9400
ACCE 10	5	-1.98163E+01	.9300	1.52155E+01 .0100
EL FOR 1	4	-2.61174E+03	1.8250	2.53853E+03 1.0250
EL FOR 1	5	-2.53853E+03	1.0250	2.61174E+03 1.8250
EL FOR 1	6	-3.20066E+00	1.8200	3.27943E+00 1.0450
EL FOR 1	7	-3.27943E+00	1.0450	3.20066E+00 1.8200
EL FOR 1	8	-2.27150E-01	.9450	1.01833E+00 .0250
EL FOR 1	9	0.00000E+00	.0000	0.00000E+00 .0000

## M I N / M A X S U M M A R Y

STRUCTURE: beam	PARAM2: ritz	PARAM3: frac.25	RESPONSE: ALL
ID	VALUE	OCCURRENCE	VALUE OCCURRENCE
ACCE 10	3 -1.98163E+01	.9300	1.52155E+01 .0100
ACCE 10	4 -6.70055E-01	.0200	6.68161E-01 .9400
ACCE 10	5 -1.98163E+01	.9300	1.52155E+01 .0100
EL FOR 1	4 -2.61174E+03	1.8250	2.53853E+03 1.0250
EL FOR 1	5 -2.53853E+03	1.0250	2.61174E+03 1.8250
EL FOR 1	6 -3.20066E+00	1.8200	3.27943E+00 1.0450
EL FOR 1	7 -3.27943E+00	1.0450	3.20066E+00 1.8200
EL FOR 1	8 -2.27150E-01	.9450	1.01833E+00 .0250
EL FOR 1	9 0.00000E+00	.0000	0.00000E+00 .0000

		M I N / M A X S U M M A R Y			
STRUCTURE:	radiator	PARAM2: eigen	PARAM3: 150 HZ	RESPONSE: ALL	
	ID	VALUE	OCCURRENCE	VALUE	OCCURRENCE
ACCE	440827	3	-9.96496E-01	31.0078	2.54486E-01 17.0102
ACCE	440827	4	-5.71353E+00	.1150	1.71285E+00 17.1851
ACCE	440827	5	-1.67945E+00	17.0102	2.06374E+00 21.3244
EL FOR	440155	4	-2.91732E-01	21.6194	9.41954E-01 .1250
EL FOR	440155	5	-1.54566E+00	16.2353	6.89794E-01 21.6094
EL FOR	440155	6	-5.46363E-02	18.1800	7.09203E-02 16.9851
EL FOR	440155	7	-4.14298E-01	21.6094	9.28340E-01 16.2353
EL FOR	440155	8	-9.38668E-01	16.2503	6.09971E-01 18.7948
EL FOR	440155	9	-2.75216E-01	21.6194	9.15571E-01 17.0051

## M I N / M A X S U M M A R Y

STRUCTURE: radiator		PARAM2: eigen		PARAM3: 60 HZ		RESPONSE: ALL	
ID		VALUE		OCCURRENCE		VALUE	OCCURRENCE
ACCE	440827	3	-1.00129E+00	31.0078		2.57273E-01	17.0102
ACCE	440827	4	-5.71337E+00	.1150		1.71283E+00	17.1851
ACCE	440827	5	-1.69103E+00	17.0102		2.07328E+00	21.3244
EL FOR	440155	4	-2.96674E-01	21.6194		9.55594E-01	.1250
EL FOR	440155	5	-1.57609E+00	16.2353		7.06379E-01	21.6094
EL FOR	440155	6	-5.56583E-02	18.1800		7.91698E-02	16.9902
EL FOR	440155	7	-4.24259E-01	21.6094		9.46616E-01	16.2353
EL FOR	440155	8	-9.60995E-01	16.2503		6.08715E-01	18.7948
EL FOR	440155	9	-2.79499E-01	21.6194		9.28780E-01	17.0051

		M I N / M A X S U M M A R Y			
STRUCTURE:	radiator	PARAM2: ritz	PARAM3: frac1.0	RESPONSE: ALL	
	ID	VALUE	OCCURRENCE	VALUE	OCCURRENCE
ACCE	440827	3	-9.92473E-01	31.0078	2.50247E-01 17.0102
ACCE	440827	4	-5.71352E+00	.1150	1.71291E+00 17.1851
ACCE	440827	5	-1.65486E+00	17.0102	2.03754E+00 21.3244
EL FOR	440155	4	-2.93704E-01	21.6194	9.49249E-01 .1250
EL FOR	440155	5	-1.54285E+00	16.2353	6.93919E-01 21.6094
EL FOR	440155	6	-5.56114E-02	18.1800	7.54589E-02 16.9902
EL FOR	440155	7	-4.16776E-01	21.6094	9.26655E-01 16.2353
EL FOR	440155	8	-9.40207E-01	16.2503	6.07573E-01 18.7948
EL FOR	440155	9	-2.76952E-01	21.6194	9.22073E-01 17.0051

## M I N / M A X S U M M A R Y

STRUCTURE: radiator		PARAM2: ritz	PARAM3: frac.5	RESPONSE: ALL	
ID		VALUE	OCCURRENCE	VALUE	OCCURRENCE
ACCE	440827	3	-9.68425E-01	31.0078	2.24475E-01 17.0501
ACCE	440827	4	-5.72016E+00	.1150	1.71146E+00 17.1851
ACCE	440827	5	-1.64152E+00	17.0102	2.09572E+00 21.3244
EL FOR	440155	4	-2.92021E-01	21.6144	9.44502E-01 .1250
EL FOR	440155	5	-1.57293E+00	16.2353	7.15953E-01 21.6094
EL FOR	440155	6	-5.54656E-02	18.1800	7.52191E-02 16.9902
EL FOR	440155	7	-4.30009E-01	21.6094	9.44717E-01 16.2353
EL FOR	440155	8	-9.55818E-01	16.2553	6.10545E-01 18.7948
EL FOR	440155	9	-2.76342E-01	21.6194	9.15235E-01 17.0051

## M I N / M A X S U M M A R Y

STRUCTURE: radiator		PARAM2: ritz	PARAM3: frac.25	RESPONSE: ALL	
ID		VALUE	OCCURRENCE	VALUE	OCCURRENCE
ACCE	440827	3	-9.65517E-01	31.0078	2.21647E-01 17.0551
ACCE	440827	4	-5.72386E+00	.1150	1.71038E+00 17.1851
ACCE	440827	5	-1.62194E+00	17.0102	2.05417E+00 21.3244
EL FOR	440155	4	-2.95189E-01	21.6144	9.55356E-01 .1250
EL FOR	440155	5	-1.65746E+00	16.2453	7.67631E-01 21.6094
EL FOR	440155	6	-5.35842E-02	18.1800	7.54796E-02 16.9851
EL FOR	440155	7	-4.61048E-01	21.6094	9.95492E-01 16.2453
EL FOR	440155	8	-1.04982E+00	16.2553	6.70256E-01 17.9000
EL FOR	440155	9	-2.78004E-01	21.6194	9.23280E-01 17.0051

## M I N / M A X S U M M A R Y

STRUCTURE: cassini		PARAM2: eigen	PARAM3: 150 HZ	RESPONSE: ALL	
ID		VALUE	OCCURRENCE	VALUE	OCCURRENCE
ACCE	10004	3	-5.88261E+02	1.3670	4.88726E+02 .7680
ACCE	10004	4	-4.04763E+02	1.9490	5.37344E+02 .6300
ACCE	10004	5	-1.75641E+03	.5140	3.23883E+02 .6850
ACCE	701	3	-3.12625E+02	1.2130	3.79308E+02 1.1050
ACCE	701	4	-2.75651E+02	1.9610	5.07647E+02 .6280
ACCE	701	5	-1.14259E+03	.7740	-6.93264E+00 .0000
EL FOR	11601	4	-1.04208E+02	.5960	5.69385E+01 .8620
EL FOR	11601	5	-3.49150E+01	1.1870	4.77298E+01 1.1300
EL FOR	11601	8	-6.76765E+02	.8570	6.52884E+02 1.1320
EL FOR	1262	4	-2.22637E+02	1.2840	1.08921E+02 .8560
EL FOR	1262	5	-4.89656E+01	.7670	2.43728E+01 1.9350
EL FOR	1262	8	-1.02894E+02	1.2850	5.93157E+01 .8550
EL FOR	16505	2	-9.22882E+01	1.3640	8.67772E+01 .6080
EL FOR	16505	3	-4.11428E+02	.8510	3.11839E+02 .5970
EL FOR	16505	4	-9.47152E+01	1.1040	9.85466E+01 .6490

## M I N / M A X S U M M A R Y

STRUCTURE: cassini		PARAM2: eigen		PARAM3: 60 HZ		RESPONSE: ALL	
ID		VALUE		OCCURRENCE	VALUE	OCCURRENCE	
ACCE	10004	3	-5.88162E+02	1.3670	4.89098E+02	.7680	
ACCE	10004	4	-4.04593E+02	1.9490	5.36760E+02	.6300	
ACCE	10004	5	-1.75667E+03	.5140	3.23813E+02	.6850	
ACCE	701	3	-3.12834E+02	1.2130	3.78818E+02	1.1050	
ACCE	701	4	-2.75536E+02	1.9610	5.07715E+02	.6280	
ACCE	701	5	-1.14237E+03	.7740	-3.90032E+01	.0000	
EL FOR	11601	4	-1.04183E+02	.5960	5.70208E+01	.8620	
EL FOR	11601	5	-3.40644E+01	1.1870	4.88520E+01	1.1300	
EL FOR	11601	8	-7.10632E+02	.8570	6.11217E+02	1.1320	
EL FOR	1262	4	-2.36321E+02	1.2840	1.02161E+02	.8560	
EL FOR	1262	5	-4.85338E+01	.7670	2.44500E+01	1.9350	
EL FOR	1262	8	-1.04599E+02	1.2850	5.85180E+01	.8550	
EL FOR	16505	2	-9.22948E+01	1.3640	8.67314E+01	.6080	
EL FOR	16505	3	-4.08437E+02	.8510	3.14572E+02	.5970	
EL FOR	16505	4	-9.41314E+01	1.1040	9.87876E+01	.6490	

## M I N / M A X S U M M A R Y

STRUCTURE: cassini		PARAM2: ritz	PARAM3: frac1.0	RESPONSE: ALL	
ID		VALUE	OCCURRENCE	VALUE	OCCURRENCE
ACCE	10004	3	-6.31701E+02	1.3650	5.02735E+02 .7660
ACCE	10004	4	-3.91711E+02	1.9500	5.49198E+02 .6310
ACCE	10004	5	-1.80007E+03	.5140	3.46303E+02 .3990
ACCE	701	3	-3.14088E+02	1.3620	3.71505E+02 1.1050
ACCE	701	4	-2.69071E+02	1.9620	5.09647E+02 .6280
ACCE	701	5	-1.14687E+03	1.0300	-1.60672E+01 .0000
EL FOR	11601	4	-1.03659E+02	.5970	5.43767E+01 .8610
EL FOR	11601	5	-3.48669E+01	.6730	4.51540E+01 1.1320
EL FOR	11601	8	-6.65697E+02	.8570	6.37747E+02 1.1330
EL FOR	1262	4	-2.21476E+02	1.2840	1.09841E+02 .8560
EL FOR	1262	5	-4.85302E+01	.7670	2.49850E+01 1.9370
EL FOR	1262	8	-1.01938E+02	1.2850	5.99338E+01 .8550
EL FOR	16505	2	-9.21733E+01	1.3640	8.67974E+01 .6080
EL FOR	16505	3	-4.10596E+02	.8510	3.12612E+02 .5970
EL FOR	16505	4	-9.37584E+01	1.1040	9.82706E+01 .6490

## M I N / M A X S U M M A R Y

STRUCTURE: cassini		PARAM2: ritz		PARAM3: frac.5		RESPONSE: ALL	
ID		VALUE		OCCURRENCE	VALUE	OCCURRENCE	
ACCE	10004	3	-6.21011E+02	1.2210	5.85796E+02	.7680	
ACCE	10004	4	-3.92906E+02	1.1080	5.29591E+02	.6350	
ACCE	10004	5	-1.62924E+03	.5080	1.67419E+02	.3470	
ACCE	701	3	-3.62351E+02	1.3630	4.04263E+02	1.1050	
ACCE	701	4	-2.68399E+02	1.0970	5.14539E+02	.6290	
ACCE	701	5	-1.16865E+03	1.0300	-3.07396E+01	.0000	
EL FOR	11601	4	-1.12753E+02	.5950	5.31308E+01	.8620	
EL FOR	11601	5	-3.22312E+01	1.1860	4.20591E+01	1.1310	
EL FOR	11601	8	-6.03145E+02	.8530	5.55450E+02	1.1330	
EL FOR	1262	4	-2.20645E+02	1.2860	1.08038E+02	.8550	
EL FOR	1262	5	-4.57440E+01	.7680	2.43067E+01	1.9390	
EL FOR	1262	8	-1.02720E+02	1.2870	5.99723E+01	.8550	
EL FOR	16505	2	-9.30055E+01	1.3620	9.00367E+01	.6070	
EL FOR	16505	3	-4.08634E+02	.8510	3.10949E+02	.5980	
EL FOR	16505	4	-9.48909E+01	1.1030	9.97088E+01	.6450	

## M I N / M A X S U M M A R Y

STRUCTURE: cassini		PARAM2: ritz	PARAM3: frac.25	RESPONSE: ALL	
ID		VALUE	OCCURRENCE	VALUE	OCCURRENCE
ACCE	10004	3	-4.52083E+02	1.3610	3.76871E+02 .9310
ACCE	10004	4	-3.88050E+02	1.9500	5.85286E+02 .6280
ACCE	10004	5	-1.62037E+03	1.2890	2.13292E+02 .6900
ACCE	701	3	-3.23387E+02	1.3680	3.62839E+02 1.1070
ACCE	701	4	-2.68010E+02	.5160	4.66561E+02 .6310
ACCE	701	5	-1.15915E+03	.7770	-4.01343E+01 .0000
EL FOR	11601	4	-9.03325E+01	.5930	5.33065E+01 .8610
EL FOR	11601	5	-3.09406E+01	.6650	4.00999E+01 1.1330
EL FOR	11601	8	-6.11457E+02	.8520	5.70471E+02 1.1370
EL FOR	1262	4	-2.15326E+02	1.2890	1.11135E+02 .8530
EL FOR	1262	5	-4.86087E+01	.7680	2.40371E+01 1.9370
EL FOR	1262	8	-9.91673E+01	1.2890	6.12986E+01 .8530
EL FOR	16505	2	-9.16432E+01	.8510	8.55358E+01 .6020
EL FOR	16505	3	-4.12496E+02	.8510	3.10529E+02 .5990
EL FOR	16505	4	-8.68751E+01	1.1050	1.05739E+02 .6460

# REPORT DOCUMENTATION PAGE

*Form Approved*  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

<b>1. AGENCY USE ONLY</b> ( <i>Leave blank</i> )	<b>2. REPORT DATE</b> May 1997	<b>3. REPORT TYPE AND DATES COVERED</b> Technical Memorandum	
<b>4. TITLE AND SUBTITLE</b> Improvements in Block-Krylov Ritz Vectors and the Boundary Flexibility Method of Component Synthesis		<b>5. FUNDING NUMBERS</b>  WU-066-00-00	
<b>6. AUTHOR(S)</b>  Kelly Scott Carney			
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>  National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191		<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  E-10705	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>  National Aeronautics and Space Administration Washington, DC 20546-0001		<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>  NASA TM-107477	
<b>11. SUPPLEMENTARY NOTES</b> This report was submitted as a dissertation in partial fulfillment of the requirements for the degree Doctor of Philosophy to Case Western Reserve University, Cleveland, Ohio, May 1997. Responsible person, Kelly Scott Carney, organization code 7735 (216) 433-2386.			
<b>12a. DISTRIBUTION/AVAILABILITY STATEMENT</b>  Unclassified - Unlimited Subject Categories 39 and 61  This publication is available from the NASA Center for AeroSpace Information, (301) 621-0390.		<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT</b> ( <i>Maximum 200 words</i> ) A method of dynamic substructuring is presented which utilizes a set of static Ritz vectors as a replacement for normal eigenvectors in component mode synthesis. This set of Ritz vectors is generated in a recurrence relationship, proposed by Wilson, which has the form of a block-Krylov subspace. The initial seed to the recurrence algorithm is based upon the boundary flexibility vectors of the component. Improvements have been made in the formulation of the initial seed to the Krylov sequence, through the use of block-filtering. A method to shift the Krylov sequence to create Ritz vectors that will represent the dynamic behavior of the component at target frequencies, the target frequency being determined by the applied forcing functions, has been developed. A method to terminate the Krylov sequence has also been developed. Various orthonormalization schemes have been developed and evaluated, including the Cholesky/QR method. Several auxiliary theorems and proofs which illustrate issues in component mode synthesis and loss of orthogonality in the Krylov sequence have also been presented. The resulting methodology is applicable to both fixed and free-interface boundary components, and results in a general component model appropriate for any type of dynamic analysis. The accuracy is found to be comparable to that of component synthesis based upon normal modes, using fewer generalized coordinates. In addition, the block-Krylov recurrence algorithm is a series of static solutions and so requires significantly less computation than solving the normal eigenspace problem. The requirement for less vectors to form the component, coupled with the lower computational expense of calculating these Ritz vectors, combine to create a method more efficient than traditional component mode synthesis.			
<b>14. SUBJECT TERMS</b> Block-Krylov; Load-dependent Ritz vectors; Boundary flexibility method; Component mode synthesis; Lanczos coordinates		<b>15. NUMBER OF PAGES</b> 205	<b>16. PRICE CODE</b> A10
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b>