Computation of a Damping Matrix for Finite Element Model Updating

by

Deborah F. Pilkey

Dissertation submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

in

Engineering Mechanics

Dr. Daniel J. Inman, Co-Chair
Dr. Calvin J. Ribbens, Co-Chair
Dr. Christopher Beattie
Dr. Mehdi Ahmadian
Dr. Romesh Batra

© April 1998, Deborah F. Pilkey, Blacksburg, Virginia
Keywords: Damping, Model Updating, High Performance Computing, Identification

Computation of a Damping Matrix for Finite Element Model Updating

Deborah F. Pilkey

(ABSTRACT)

The characterization of damping is important in making accurate predictions of both the true response and the frequency response of any device or structure dominated by energy dissipation. The process of modeling damping matrices and experimental verification of those is challenging because damping can not be determined via static tests as can mass and stiffness. Furthermore, damping is more difficult to determine from dynamic measurements than natural frequency. However, damping is extremely important in formulating predictive models of structures. In addition, damping matrix identification may be useful in diagnostics or health monitoring of structures.

The objective of this work is to find a robust, practical procedure to identify damping matrices. All aspects of the damping identification procedure are investigated. The procedures for damping identification presented herein are based on prior knowledge of the finite element or analytical mass matrices and measured eigendata. Alternately, a procedure is based on knowledge of the mass and stiffness matrices and the eigendata. With this in mind, an exploration into model reduction and updating is needed to make the problem more complete for practical applications. Additionally, high performance computing is used as a tool to deal with large problems. High Performance Fortran is exploited for this purpose. Finally, several examples, including one experimental example are used to illustrate the use of these new damping matrix identification algorithms and to explore their robustness.

Acknowledgments

First, I would like to express my thanks and gratitude to my two advisors, Dr. Daniel Inman and Dr. Calvin Ribbens. For the last four years, Dan Inman has supplied an endless amount of moral support and encouragement. Cal Ribbens has advised me for two years, always providing rational, realistic views and thoughts.

My degree has been funded by Virginia Tech, the Army Research Office, the Virginia Space Grant Consortium, and the Institute for Computer Applications in Science and Engineering (ICASE), through their VILaP HPCC program. ICASE has also provided two summers of research and education at NASA Langley Research Center. My gratitude goes out to David Keyes and Manny Salas at ICASE for believing in me, my abilities, my integrity, and my future. My summers were made complete by the generosity and friendship offered by all the ICASE employees, especially my fellow graduate students, Kevin Roe and Dave Cronk.

My friendships along the way have been invaluable. For this I thank all of the MSL gang (old and new) and the new CIMSS group. I owe a special thanks to all of my office-mates along the way: Eric Austin, Sergio Carneiro, Jens Cattarius, Clay Carter, Myung-Hyun Kim, Mauro Atalla, Brett Pokines, Greg Agnes, and my good friends Ralph Rietz, Dino Sciulli, Sean Curran, and Ami Leighter. Gyuhae Park was generous enough to provide experimental data. Joel Grasmeyer deserves many thanks for sharing with me his powerful and addicting love of life and the outdoors.

My parents have been there for me throughout my life with advice, encouragement, love and support. Thank you.

Contents

List of Figures ..... **vii**

List of Tables ..... **x**

Nomenclature ..... **xi**

1. Introduction **1**

1.1. Motivation 1
1.2. Problem Definition 2
1.3. Objectives 2
1.4. Background Survey 4
1.4.1. Minas & Inman 4
1.4.2. Beliveau 5
1.4.3. Caravani & Thomson 6
1.4.4. Chen, Ju & Tsuei 7
1.4.5. Fabunmi, Chang And Vorwald 8
1.4.6. Fritzen 10
1.4.7. Gaylard 11
1.4.8. Hasselman 12
1.4.9. Ibrahim 12
1.4.10. Lancaster 13
1.4.11. Roemer And Mook 14
1.4.12. Starek And Inman 15
1.4.13. Wang 16
1.4.14. Others 17

2. Damping Identification Methods **18**

2.1. Iterative Damping Matrix Identification Routine 18
2.2. Direct Method 22
2.3. Discussion of Positive Definiteness 24

3. Data Incompleteness and Model Updating **26**

3.1. Introduction 26
3.2. Spatial Incompleteness 28
3.2.1. Static Reduction and Expansion 29
3.2.2. Dynamic Reduction and Expansion 30
3.2.3. Improved Reduced System (IRS) 30
3.2.4. Iterated IRS 30
3.2.5. Comparison of Methods 31
3.2.6. Choice of Master Degrees of Freedom 32
3.2.7. Inclusion of Damping Term in Model Reduction and Expansion 32
3.2.7.1. Static Expansion including Damping 32
3.2.7.2. Dynamic Reduction/Expansion and IRS Including Damping 33
3.3. Model Updating of Damping and Stiffness Matrices 34

4. Computational Issues 38

4.1. Introduction 38
4.2. High Performance Fortran 39
4.3. Application of HPC to model expansion/reduction 39
4.3.1. Illustrative Examples 41
4.3.1.1. Beam Example 42
4.3.1.2. Plate Example 42
4.3.2. Results and Conclusions 43
4.4. Iterative method vs. least square - a computational look 46
4.4.1. Programming Methodology 46
4.4.2. Results 47
4.4.2.1. Problem Size - Memory Requirements 47
4.4.2.2. Timing 48
4.4.2.3. Performance 48
4.4.3. Parallel Results on the SP-2 49
4.5. Computational Aspects of Direct and Iterative Damping Identification Methods 50

5. Examples 59

5.1. Introduction 59
5.2. Lumped Mass System Example 59
5.3. Damping Identification of lumped mass system 61
5.4. Plate Example 63
5.5. Obtaining Results 65
5.5.1. Data Generation 65
5.5.2. Solution Methods 67
5.6. Results of plate example 67
5.6.1. Plots - Iterative Method 68
5.6.2. Plots - Direct Method 80

6. Experimental Verification and Example of Use 92

6.1. Introduction 92
6.2. Experimental Setup 92
6.3. Finite Element Model 94
6.4. Damping Identification Procedure 95
6.5. Results and Discussion 96

7. Conclusions 99

References 102

Appendices 108

Appendix A: Flops Count 108
Appendix B: Data Mapping 110
Appendix C: HPF Attributes 113

List of Figures

Figure 2-1: Schematic of iterative method 22
Figure 2-2 Schematic of direct damping identification method 23
Figure 3-1 Illustration of spatial incompleteness 27
Figure 3-2 Illustration of modal incompleteness 28
Figure 4-1 Least squares method (Chen, Ju & Tseui, 1997) 47
Figure 4-2: Time for one processor to complete multiple degree of freedom problem 49
Figure 4-3 Direct and iterative damping matrix identification algorithm 52
Figure 5-1 Lumped mass system 59
Figure 5-2 Mesh of damping matrix 65
Figure 5-3 (a) Solution method where the eigendata is obtained before the model reduction is performed. (b) Solution method where the reduction is performed before the eigendata is generated 66
Figure 5-4 Plot of damping matrix found using the iterative method with 40 DOF and 40 modes. 68
Figure 5-5 Difference between Figure 5-4 and the target damping matrix. 68
Figure 5-6 Plot of damping matrix found using the iterative method with 40 DOF 10 percent fewer modes 69
Figure 5-7 Difference between Figure 5-6 and the target damping matrix. 69
Figure 5-8 Plot of damping matrix found using the iterative method with 40 DOF 33 percent fewer modes 70
Figure 5-9 Difference between Figure 5-8 and the target damping matrix. 70
Figure 5-10 Plot of damping matrix found using the iterative method with 40 DOF and 50 percent fewer modes 71
Figure 5-11 Difference between Figure 5-10 and the target damping matrix. 71
Figure 5-12 FRF plots of plate using iterative damping ID, method a, and only 30 DOF. 72
Figure 5-13 Difference between the FRF's of Figure 5-12 and the expected FRF 72
Figure 5-14 FRF plots of plate using iterative damping ID, method a, and only 20 DOF. 73

Figure 5-15 Difference between the FRF's of Figure 5-14 and the expected FRF..... 73
Figure 5-16 FRF plots of plate using iterative damping ID, method a, and only 10 DOF.74
Figure 5-17 Difference between the FRF's of Figure 5-16 and the expected FRF..... 74
Figure 5-18 FRF plots of plate using iterative damping ID, method b, and only 10 modes.75
Figure 5-19 Difference between the FRF's of Figure 5-18 and the expected FRF..... 75
Figure 5-20 Mesh of damping matrix found using iterative method of identification with noise added to the system. 76
Figure 5-21 Difference between the above plot and the actual damping matrix..... 76
Figure 5-22 Mesh of damping matrix found using iterative method of identification with 10 percent fewer modes and noise added to the system. 77
Figure 5-23 Difference between the above plot and the actual damping matrix..... 77
Figure 5-24 Mesh of damping matrix found using iterative method of identification with 33 percent fewer modes and noise added to the system. 78
Figure 5-25 Difference between the above plot and the actual damping matrix..... 78
Figure 5-26 Mesh of damping matrix found using iterative method of identification with 50 percent fewer modes and noise added to the system. 79
Figure 5-27 Difference between the above plot and the actual damping matrix..... 79
Figure 5-28 Plot of damping matrix found using the direct method with 40 DOF and 40 modes. 80
Figure 5-29 Difference between Figure 5-28 and the target damping matrix. 80
Figure 5-30 Plot of damping matrix found using the direct method with 40 DOF and 10 percent fewer modes. 81
Figure 5-31 Difference between Figure 5-30 and the target damping matrix. 81
Figure 5-32 Plot of damping matrix found using the direct method with 40 DOF and 33 percent fewer modes. 82
Figure 5-33 Difference between Figure 5-32 and the target damping matrix. 82
Figure 5-34 Plot of damping matrix found using the direct method with 40 DOF and 50 percent fewer modes. 83
Figure 5-35 Difference between Figure 5-34 and the target damping matrix. 83
Figure 5-36 FRF plots of plate using direct damping ID, method a, and only 30 DOF.... 84

Figure 5-37 Difference between the FRF's of Figure 5-36 and the expected FRF..... 84
Figure 5-38 FRF plots of plate using direct damping ID, method a, and only 20 DOF.... 85
Figure 5-39 Difference between the FRF's of Figure 5-38 and the expected FRF..... 85
Figure 5-40 FRF plots of plate using direct damping ID, method a, and only 10 DOF.... 86
Figure 5-41 Difference between the FRF's of Figure 5-40 and the expected FRF..... 86
Figure 5-42 FRF plots of plate using iterative damping ID, method b, and only 10 modes.87
Figure 5-43 Difference between the FRF's of Figure 5-42 and the expected FRF..... 87
Figure 5-44 Mesh of damping matrix found using direct method of identification with noise added to the system..... 88
Figure 5-45 Difference between the above plot and the actual damping matrix..... 88
Figure 5-46 Mesh of damping matrix found using direct method of identification with 10 percent fewer modes and noise added to the system..... 89
Figure 5-47 Difference between the above plot and the actual damping matrix..... 89
Figure 5-48 Mesh of damping matrix found using direct method of identification with 33 percent fewer modes and noise added to the system..... 90
Figure 5-49 Difference between the above plot and the actual damping matrix..... 90
Figure 5-50 Mesh of damping matrix found using direct method of identification with 50 percent fewer modes and noise added to the system..... 91
Figure 5-51 Difference between the above plot and the actual damping matrix..... 91
Figure 6-1 Schematic of bolted beam used in the example..... 92
Figure 6-2 Experimental setup..... 93
Figure 6-3 Experimental data..... 94
Figure 6-4 Coherence plot for experimental data..... 94
Figure 6-5 Experimental procedure..... 96
Figure 6-6 Comparison of FRFs..... 97
Figure 6-7 error of the above plots..... 97

List of Tables

Table 3-1 First 10 analytical natural frequencies of a 40 DOF plate, and the natural frequencies of the same plate when reduced to 10 degrees of freedom using static reduction, IRS and Iterated IRS (5 iterations) 32
Table 4-1: Results for parallelized model reduction algorithms applied to 1000 degree of freedom beam example 45
Table 4-2: Results for parallelized model reduction algorithms applied to 1012 degree of freedom plate example. 45
Table 4-3: Computational rate (MFLOPS per processor) for static and IRS reduction applied to two test problems. 45
Table 4-4 Results for the iterative method on the SP-2, using several different problem sizes. 50
Table 4-5 Computational results for direct method applied to a 100 DOF lumped mass example. 55
Table 4-6 Computational results for direct method applied to a 480 DOF plate example 55
Table 4-7 Computational results for direct method applied to a 1012 DOF plate example. 56
Table 4-8 Computational results for iterative method (one iteration) applied to a 100 DOF lumped mass example. 56
Table 4-9 Computational results for iterative method (one iteration) applied to a 480 DOF plate example 57
Table 4-10 Computational results for iterative method (one iteration) applied to a 1012 DOF plate example 57
Table 4-11 Computational rate (MFLOPS per processor) for the three test problems using direct and iterative damping identification. 58

Nomenclature

M : Mass matrix (n x n)

C : Damping matrix (n x n)

K : Stiffness matrix (n x n)

d : Vector containing the $(n^2+n)/2$ unique elements of **C**

p : Vector containing elements of **M, C,** and **K**

$\mathbf{u}_i$: Right eigenvector

$\mathbf{G}_i$: Matrix containing the real and imaginary parts if $\mathbf{u}_i$

$\mathbf{H}^c$: Complex frequency response function

$\mathbf{H}^n$: Normal frequency response function

J : Jacobean

T : Transformation matrix

Λ : Diagonal matrix of eigenvalues

Φ : Matrix of right eigenvectors

x : Displacement

$\dot{x}$: Velocity

$\ddot{x}$: Acceleration

$\mathbf{X}(s)$: Laplace transform of displacement

n : Number of DOFs in the FE model

s : Complex Laplace variable

λ_i : Eigenvalue

ω_i : Natural frequency

γ : damping loss factor

Subscripts

RE : Real part of a matrix

IM : Imaginary part of a matrix

m : Master (measurable) DOF

s : Slave DOF

Operators

* : Complex conjugate transpose

T : Transpose of a matrix

- : Complex conjugate of a matrix

Chapter 1

Introduction

1.1. Motivation

The field of damping matrix identification is one which still holds quite a bit of intrigue in the engineering community. This is because the modeling of damping is very complex and is still considered somewhat of an unknown or gray area. The effects of damping are clear, but the characterization of damping is a puzzle waiting to be solved.

The synthesis of damping in structural systems and machines is extremely important if a model is to be used in predicting transient responses, transmissibility, decay times or other characteristics in design and analysis that are dominated by energy dissipation. Methods for determining the mass and stiffness matrices of a system are more straight forward than those for determining the damping matrix as they represent quantities which can be measured and evaluated by static tests. Damping, on the other hand must be determined by dynamic testing. This makes the process of modeling and experimental verification difficult. It is assumed here that acceptable models of the mass and stiffness matrices are available and that it is desired to use the eigenvalue and eigenvector information to construct a damping matrix. This is known as an inverse problem.

One application of the inverse problem is diagnostics. This idea is to test for changes in a structure's properties by looking at changes in measurable values such as mode shapes or frequencies. Here the underlying assumption is that changes in the damping values correspond to some sort of change in the structure's health. Banks et al. [1], [2] have shown that damping is much more sensitive to change in a structure than stiffness is.

1.2. Problem Definition

The problem being investigated assumes a structural system consisting of mass (**M**), damping (**C**), and stiffness (**K**) matrices such that the response *x(t)* satisfies the homogeneous equation of motion

$$\mathbf{M}\ddot{x} + \mathbf{C}\dot{x} + \mathbf{K}x = \mathbf{0} \tag{1-1}$$

where x is an n by 1 vector varying with time, representing the displacements of the masses in a lumped mass system (n is the number of degrees of freedom). The vectors $\ddot{x}$ and $\dot{x}$ represent the acceleration and velocity respectively of the lumped masses. The eigenvalue equation (also called the quadratic pencil) for the system is written as

$$(\lambda_i^2\mathbf{M} + \lambda_i\mathbf{C} + \mathbf{K})\mathbf{u}_i = \mathbf{0}\,. \tag{1-2}$$

In the frequency domain, the equation of motion is written as

$$(-\omega_i^2\mathbf{M} + i\omega_i\mathbf{C} + \mathbf{K})\mathbf{v}_i = \mathbf{f}(\omega_i)\,. \tag{1-3}$$

Again, the equation of motion can be written in the Laplace domain as

$$(s^2\mathbf{M} + s\mathbf{C} + \mathbf{K})\mathbf{X}(s) = \mathbf{F}(s) \tag{1-4}$$

where s is the Laplace variable, and the initial conditions are assumed to be zero.

The idea behind an inverse problem is to find the physical parameters of a system (mass, damping, and stiffness) from its behavior using measurements such as forced responses and natural frequencies. Damping identification is an inverse problem in which the damping matrix is the desired result.

1.3. Objectives

Two damping matrix identification methods are developed which produce accurate representative damping matrices. An in depth investigation into model expansion, reduction and updating serves to integrate the theory and practical application of damping matrix identification. Another practical issue is addressed by utilizing High Performance Fortran to show the benefits high performance computing for the solution of larger, more

realistic sized problems. Examples show that an experimental system can be characterized , and that the damping matrix identification routines proposed in this work are indeed robust.

The following sections and chapters will provide a thorough investigation into damping matrix identification. First, an in depth background survey will give the reader a detailed summary of the past and present efforts of researchers to characterize a damping matrix. Then, two new methods which out-perform all of the previous attempts are introduced.

Next, some of the more practical issues of verification are introduced. These include model reduction or expansion, methods which account for the difference in size between the experimental data and finite element or analytical model. In addition, simultaneous model updating of stiffness and damping matrices is explored as a way to correlate experimental and analytical information.

Following the theory, Chapter 4 discusses the computational issues faced in solving the damping identification problem. This is an important aspect of the problem because often models are large, and must be solved in the most efficient manner. The tool to do this is high performance computing. This chapter investigates high performance computing issues associated with the entire problem including model expansion/reduction and both types of damping identification explored in this work. Finally, a computational comparison is made between the iterative damping identification method and another current method. This clearly illustrates the computational benefits of using the iterative method

Finally, in Chapter 5 examples are presented which clearly show the robustness of the two damping matrix identification routines. Not only are straightforward solutions plotted, but results can be seen for problems with noisy data, and for problems with various combinations of spatial and modal incompleteness. Then Chapter 6 presents an example of the use of the damping matrix identification algorithms on a bolted beam experiment.

1.4. Background Survey

When all the information in the above equations is considered known and accurate, save the damping matrix, there are straightforward methods to obtain the unknown matrix. One such method is presented in the frequency domain by Chen, Ju and Tsuei [3], whose equation is based on knowledge of a complex measured frequency response function, $\mathbf{H}^c$.

$$\mathbf{C} = -\frac{1}{\omega}\left\{\mathbf{H}^c_{\mathrm{Re}}(\omega) + \mathbf{H}^c_{\mathrm{Im}}(\omega)[\mathbf{H}^c_{\mathrm{Re}}(\omega)]^{-1}\mathbf{H}^c_{\mathrm{Im}}(\omega)\right\}^{-1}\mathbf{H}^c_{\mathrm{Im}}(\omega)[\mathbf{H}^c_{\mathrm{Re}}(\omega)]^{-1}. \qquad (1\text{-}5)$$

Unfortunately, it is unrealistic to assume that all pertinent information is given to solve for a damping matrix. Actually, data from testing is neither complete nor is it error free. The methods described below all attempt to find the best method for damping matrix identification given various levels of incomplete or noisy data [4].

1.4.1. Minas & Inman

The method of Minas and Inman [5] assumes that analytical mass and stiffness matrices are determined *a priori* from a finite element model. Eigenvalues and eigenvectors are obtained experimentally, and are allowed to be incomplete, as would be expected from modal testing. The damping matrix under investigation is assumed to be real, symmetric and positive definite. The mass and stiffness matrices are reduced to the size of the modal data available. Minas and Inman start by rearranging the eigenvalue problem of equation (1-2), such that

$$\mathbf{C}\mathbf{u}_i = -(1/\lambda_i)(\lambda_i^2\mathbf{M} + \mathbf{K})\mathbf{u}_i. \qquad (1\text{-}6)$$

Taking the complex conjugate transpose leads to

$$\mathbf{u}_i^*\mathbf{C} = \mathbf{f}_i^*, \qquad (1\text{-}7)$$

where,

$$\mathbf{f}_\mathbf{i} = -(1/\lambda_i)(\lambda_i^2\mathbf{M} + \mathbf{K})\mathbf{u}_\mathbf{i} \qquad (1\text{-}8)$$

The solution to (1-7) is made possible by separating the real and imaginary parts to create

$$\mathbf{G_i d} = \mathbf{b_i} \tag{1-9}$$

where **d** is a vector containing the $(n^2+n)/2$ unique real elements of the damping matrix. The vector $\mathbf{b}_i$ contains the real and imaginary parts of $\mathbf{f}_i$,

$$\mathbf{b_i} = \begin{Bmatrix} \mathrm{Re}(\mathbf{f}_i) \\ \mathrm{Im}(\mathbf{f}_i) \end{Bmatrix}, \tag{1-10}$$

and the matrix $\mathbf{G}_i$ contains the real and imaginary parts of $\mathbf{u}_i$.

The row dimension of **G** is determined by the number of linearly independent equations provided by the eigenvectors. When the system is overdetermined, a least squares approach is in order. A determined problem with a square **G** matrix can be solved with a straightforward inverse. An underdetermined problem requires the use of the Moore-Penrose inverse or some other optimization procedure. The result is a unique identification of a damping matrix. The structure must exhibit complex modes for this procedure, and the solution is limited to real symmetric positive (semi-) definite damping matrices.

1.4.2. Beliveau

Beliveau [6] uses natural frequencies, damping ratios, mode shapes and phase angles to identify parameters of a viscous damping matrix. The method uses a Bayesian framework based on eigenvector and eigenvalue perturbations and a Newton-Raphson scheme. The identification is performed iteratively.

$$\mathbf{Q} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & -\mathbf{M} \end{bmatrix} \tag{1-11}$$

$$\mathbf{B} = \begin{bmatrix} -\mathbf{C} & -\mathbf{M} \\ -\mathbf{M} & \mathbf{0} \end{bmatrix} \tag{1-12}$$

The perturbation of the eigenvalues and eigenvectors satisfy:

$$\frac{\partial \lambda_i}{\partial \theta} = \frac{\mathbf{u}_{Li}^T \left\{ \left[\frac{\partial \mathbf{Q}}{\partial \theta} \right] - \lambda_i \left[\frac{\partial \mathbf{B}}{\partial \theta} \right] \right\} \mathbf{u}_{Ri}}{\mathbf{u}_{Li}^T \mathbf{B} \mathbf{u}_{Ri}} \tag{1-13}$$

$$\left[\mathbf{Q}^T - \lambda_i \mathbf{B}^T \vdots \mathbf{u}_{Ri} \right] \left[\frac{\mathbf{Q} - \lambda_i \mathbf{B}}{\mathbf{u}_{Ri}^T} \right] \left\{ \frac{\partial \mathbf{u}_{Ri}}{\partial \theta} \right\} = -\left[\mathbf{Q}^T - \lambda_i \mathbf{B}^T \vdots \mathbf{u}_{Ri} \right] \times \left[\frac{\frac{\partial \mathbf{Q}}{\partial \theta} - \lambda_i \frac{\partial \mathbf{B}}{\partial \theta} - \frac{\partial \lambda_i}{\partial \theta} \mathbf{B}}{0^T} \right] \mathbf{u}_{Ri} \tag{1-14}$$

where $\mathbf{u}_R$ and $\mathbf{u}_L$ are the right and left eigenvectors of the system described in equations (1-11) and (1-12) above. This method involves solving an nth order system of linear equations for each eigenvector, making equation (1-14) fairly inefficient. An objective function, which is formulated at length by the author, is iterated upon until convergence is achieved. The variance of parameters is obtainable, although a model is not always forthcoming.

1.4.3. Caravani & Thomson

Caravani and Thomson [7] introduce a numerical technique which identifies damping coefficients when the frequency response is known. This paper is specific to viscous damping. The identification is done in an iterative manner. A distinction is made between the ideal case and the real case. The goal is to chose a set of damping coefficients that will minimize ($\mathbf{v}_{real}$ - $\mathbf{v}_{ideal}$), where $\mathbf{v}$ is defined in equation (1-3).

The algorithm includes real and ideal displacement measures.

$$\mathbf{L_k} = \sum_{i=1}^{k} (\mathbf{v}_{i,k} - \bar{\mathbf{v}}_i)^T \mathbf{A}_i (\mathbf{v}_{i,k} - \bar{\mathbf{v}}_i) + (\mathbf{d}_k - \mathbf{d}_0)^T \mathbf{B}_i (\mathbf{d}_k - \mathbf{d}_0), \tag{1-15}$$

where $\mathbf{v}$ is the response vector, $\mathbf{d}$ is a vector containing the unknown damping values from the damping matrix, $\mathbf{A}_i$ is the degree of confidence, and $\mathbf{B}_i$ in this case is a penalty function.

The algorithm requires an initial guess for the damping values, $\mathbf{d}_0$. $\mathbf{v}_{ideal}$ is taken as a function of **d** at every iteration. Thus, it is necessary to have a model for the system so that the response at each step can be obtained. $\mathbf{A}_i$ can be seen as an inverse covariance matrix. The iteration on $\mathbf{d}_k$ and $\mathbf{v}_{i.k}$ is defined by $\mathbf{d}_k = \mathbf{d}_{k-1} + f[\mathbf{J,A,B,v,d_{k-1},d_0}]$ and $\mathbf{v}_{i.k} = \mathbf{v}_{i,k-1} + \mathbf{J}_{i,k-1}(\mathbf{d}_k - \mathbf{d}_{k-1})$, where $\mathbf{J}_{i,k-1}$ is the Jacobian [d**v**/dc]. Usually **B** is set to zero, so that you get stationary values for the coefficients. If the coefficients do not "look" physically correct, then one must choose a new **B** matrix and repeat the process. Examples are shown where the system model is known, and thus the ideal model is exact. The "measured" converges to the exact response. This method is meant to solve relatively simple problems, as there are complications when the number of dampers exceeds the number of lumped masses.

1.4.4. Chen, Ju & Tsuei

Chen, Ju, and Tsuei [3] use a frequency domain method to estimate the damping matrix of a structure. The method begins with the knowledge of an experimentally obtained complex frequency response function, $\mathbf{H}^c$. A normal frequency response function ($\mathbf{H}^n$) is derived from the complex function with

$$\mathbf{H}^n(\omega) = \mathbf{H}^c_{\mathbf{Re}}(\omega) - \mathbf{T}(\omega)\mathbf{H}^c_{\mathbf{Im}}(\omega), \tag{1-16}$$

where the transformation matrix **T** is defined as

$$\mathbf{T}(\omega) = -\mathbf{H}^c_{\mathbf{Im}}(\omega)[\mathbf{H}^c_{\mathbf{Re}}(\omega)]^{-1}. \tag{1-17}$$

When there is no noise, the exact damping matrix can be solved directly with

$$\mathbf{C} = \frac{1}{\omega_i}[\mathbf{H}^n(\omega_i)]^{-1}\mathbf{T}(\omega_i). \tag{1-18}$$

To account for the noise problem, equation (1-18) must be solved at several frequencies. This is done by creating a vector **d**, containing the $(n^2+n)/2$ elements of the symmetric damping matrix. This vector is multiplied with the collection of natural frequencies and normal FRF matrices, and the other side of the equation contains the

collection of transformation matrices for each frequency. This is solved in a least squares sense for the parameters of the damping matrix. If the damping matrix is known to be banded or diagonal, this can be accounted for to reduce the amount of computation.

1.4.5. Fabunmi, Chang And Vorwald

Fabunmi, Chang, and Vorwald [8] present a damping matrix identification scheme that uses forced response data in the frequency domain. They assume knowledge of a structural mass matrix, stiffness matrix and frequency response data. The damping matrix is first formulated as

$$\mathbf{C}\mathbf{v}(\omega_j) = \frac{-i}{f(\omega_j)}[\mathbf{f}(\omega_j) + (\omega_j{}^2\mathbf{M} - \mathbf{K})\mathbf{v}(\omega_j)] \tag{1-19}$$

where $f(\omega) = \omega$ for viscous damping and $f(\omega) = 1$ for hysteretic damping. The damping matrix is then transformed into a vector, $\mathbf{d}_c$ which contains all the elements of the damping matrix. The response vectors $\mathbf{v}(\omega_j)$ are transformed into matrices containing the response information along the diagonal, for each frequency value, j. The right hand side of equation (1-19) is referred to as the mobility, and is assembled into a Jn x 1vector containing the mobility for each frequency, j, where J is the total number of sampled frequencies. The resulting equation is,

$$\begin{bmatrix} diag[\mathbf{v}(\omega_1)] \\ diag[\mathbf{v}(\omega_2)] \\ \vdots \\ diag[\mathbf{v}(\omega_J)] \end{bmatrix} \mathbf{d} = \begin{bmatrix} \frac{-i}{f(\omega_1)}[\mathbf{f}(\omega_1) + (\omega_1^2\mathbf{M} - \mathbf{K})\mathbf{v}(\omega_1)] \\ \frac{-i}{f(\omega_2)}[\mathbf{f}(\omega_2) + (\omega_2^2\mathbf{M} - \mathbf{K})\mathbf{v}(\omega_2)] \\ \vdots \\ \frac{-i}{f(\omega_J)}[\mathbf{f}(\omega_J) + (\omega_J^2\mathbf{M} - \mathbf{K})\mathbf{v}(\omega_J)] \end{bmatrix}. \tag{1-20}$$

Now, $\mathbf{v}$ can be written as a linear combination of K orthonormal basis vectors, ϕ_j. A Gram-Schmidt process can be used to find these, such that

$$\alpha_{nj} = \phi_j^T \mathbf{v}_n \qquad j = 1, \ldots, K \quad n = 1, \ldots, \text{n} \tag{1-21}$$

A transformation is defined such that the nonzero terms of the resulting vector are,

$$\left[\mathrm{L}_k(X_n)\right]_{(i-1)n+k} = X_n^i \qquad i = 1,...,J. \tag{1-22}$$

This is performed on ϕ such that $\psi_j^k = \mathrm{L}_k(\phi_j)$. Then,

$$\Phi = \left(\psi_1^1|\psi_2^1|\cdots|\psi_K^1|\psi_1^2|\psi_2^2|\cdots|\psi_K^2|\cdots|\psi_K^n\right). \tag{1-23}$$

A matrix **A** is defined such that $\mathbf{A}_{ij}$=0 for $i \neq j$, and $\mathbf{A}_{jj}$ is a diagonal submatrix whose entries are α_{n1}, $\alpha_{n2,}$ ... $\alpha_{nj.}$ The problem can now be redefined as

$$\mathbf{Ad} = \Phi^* \begin{bmatrix} \dfrac{-i}{f(\omega_1)}[\mathbf{f}(\omega_1) + (\omega_1^2\mathbf{M} - \mathbf{K})\mathbf{v}(\omega_1)] \\ \dfrac{-i}{f(\omega_2)}[\mathbf{f}(\omega_2) + (\omega_2^2\mathbf{M} - \mathbf{K})\mathbf{v}(\omega_2)] \\ \vdots \\ \dfrac{-i}{f(\omega_J)}[\mathbf{f}(\omega_J) + (\omega_J^2\mathbf{M} - \mathbf{K})\mathbf{v}(\omega_J)] \end{bmatrix}. \tag{1-24}$$

The elements of vector **d** are solved as a set of linear equations, and an "untreated" damping matrix, $\hat{\mathbf{C}}$ is assembled. This untreated matrix will often be non-symmetric and contain imaginary parts. These undesirable qualities are removed by

$$\mathbf{C} = \frac{1}{4}\left(\hat{\mathbf{C}} + \hat{\mathbf{C}}^T + \overline{\hat{\mathbf{C}}} + \hat{\mathbf{C}}^*\right), \tag{1-25}$$

where the overbar represents the complex conjugate, and the * represents the complex conjugate transpose.

It is possible by using this method to obtain a nonunique damping matrix which reproduces the measured data, but which does not resemble the true damping matrix. This occurs when one or more of the basis vectors is "trivial", meaning in this case that its contribution to the mode is smaller than the noise level.

1.4.6. Fritzen

A loss function must be optimized (i.e., minimize the error) to estimate the system parameters in Fritzen's 1986 [9] damping identification method. Output error (OE) refers to comparing the output signals with a given input signal ($\varepsilon_{OE} = \mathbf{X}_{system} - \mathbf{X}_{model}$). This requires nonlinear optimization and can have problems such as computation time, convergence, and initial models required. The output error method yields good results even with the presence of noise. The method that is used in this paper is called equation error (EE) or input error. It is the error of the equation of motion:

$$\varepsilon_{EE} = \mathbf{F}(s) - (s^2\mathbf{M} + s\mathbf{C} + \mathbf{K})\,\mathbf{X}(s). \tag{1-26}$$

Equation (1-26) is linear and easier to deal with than nonlinear equations. A least squares method is used in conjunction with the equation error method because it is very straightforward. The procedure involves defining a large matrix equation by combining known vectors and matrices into larger ones. A vector **p** is defined by the elements of the mass, stiffness and damping matrices. **M**, **C**, and **K** must be symmetric with some elements known *a priori*. The dimension of **p** is defined to be $N \leq 3n^2$. The equation to be investigated can be summarized as

$$\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{p}, \tag{1-27}$$

where **e** = error, **b** = forces,

$$\mathbf{A} = \begin{bmatrix} s_1^2 X_1^T(s_1) & s_1 X_1^T(s_1) & X_1^T(s_1) \\ s_2^2 X_1^T(s_2) & s_2 X_1^T(s_2) & X_1^T(s_2) \\ \vdots & \vdots & \vdots \\ s_\nu^2 X_\mu^T(s_\nu) & s_\nu X_\mu^T(s_\nu) & X_\mu^T(s_\nu) \end{bmatrix}. \tag{1-28}$$

The method of least squares minimizes the loss function (min $\|\mathbf{e}^T\mathbf{e}\|$). The equation can be written in normal form as $\mathbf{A}^T\mathbf{A}\mathbf{p} = \mathbf{A}^T\mathbf{b}$ where **p** is the solution vector. There are different ways to solve this. First, solve the normal equations by standard elimination methods (with iterative improvement). Second, use Householder transformations, Gram-Schmidt orthogonalization, or pseudo-inverse techniques to solve $\mathbf{0} = \mathbf{b} - \mathbf{A}\mathbf{p}$. A method

is sought that preserves the linear nature of the problem and is still less sensitive to noise. This least squares type of method can have bias problems when faced with noise.

The Instrument Variable (IV) method is able to reduce or eliminate bias when noise is present. **W** is known as the Instrument Variable Matrix with the properties,

$$\underset{l\to\infty}{p\lim}\left\{\frac{1}{l}W^T e\right\} = 0 \tag{1-29}$$

$$\underset{l\to\infty}{p\lim}\left\{\frac{1}{l}W^T A\right\} \text{ is nonsingular,} \tag{1-30}$$

and has restrictions such that **W** and **e** are not correlated. *l* is the number of observations in the preceding equations. Creation of auxiliary matrices becomes necessary. An iterative procedure is used to solve for $\mathbf{p}_{IV}$. One advantage is that a noise model is not needed. The method can be described in a few steps. First, a least squares estimate is used to find an initial **p**. This is used to calculate the instrument variables, *X*. $\mathbf{W}^T\mathbf{A}$ and $\mathbf{W}^T\mathbf{b}$ are generated and a new **p** is estimated with $\mathbf{p} = (\mathbf{W}^T\mathbf{A})^{-1}\mathbf{W}^T\mathbf{b}$. Next, **p** is compared with **p** in the previous step. If they do not match up sufficiently, then the processes is iterated with the new **p** vector. If convergence has been achieved then the damping matrix is assembled from the elements of **p**. One problem with the Instrument Variable method is that the choice of the **W** matrix is not straightforward. After several examples, Fritzen concludes that the accuracy of the IV method is better than that of the least squares. Also, systems with more damping are better suited for this method.

1.4.7. Gaylard

Gaylard [10] presents a unique method of damping identification using mass weighted coproducts. This is a time domain approach, and can be related to autocorrelation style analysis methods. Computational demands of this approach are heavy, and deconvolutions are required in the analysis.

Results of an example show that using the coproduct method along with identified rather than true mass decreases the error sensitivity of the damping matrix significantly

over recurrence identification and true mass. The coproduct method also does not produce instable behavior, which can be a problem with the recurrence method. In another example, the author adds damping to a beam, and is able to pick up the increase in damping, but unable to locate it. Thus, this method is only good for Rayleigh style damping.

1.4.8. Hasselman

Hasselman [11] assumes linear viscous damping and investigates two types of damping: proportional damping, where damping is a linear combination of the mass and stiffness matrices, and nonproportional damping, which usually does not diagonalize.

The method of normal modes can be used for the analysis of structures. First, the equations of motion are written in terms of modal coordinates. Usually damping is introduced after this and it is assumed to be diagonal.

Hasselman's method uses the phase differences between the coincident and the quadrature components of the acceleration response to construct the off diagonal terms of the damping matrix. The damping identification reduces to,

$$\mathbf{C}_{jj} = -2\sigma_j \mathbf{M}_{jj} \tag{1-31}$$

$$\mathbf{C}_{jk} = \omega_j \delta\phi_{\mathrm{Im}j}{}^{\mathrm{T}} m\, \phi_{\mathrm{Re}k} + \omega_k \phi_{\mathrm{Re}j}{}^{\mathrm{T}} m\, \delta\phi_{\mathrm{Im}k} \quad , \quad j \neq k \tag{1-32}$$

where $x(t) = \phi_{\mathrm{Re}j} e^{(\sigma j + i\omega j)1/2}$ and $\phi = \phi_{\mathrm{Re}} + \delta\phi_{\mathrm{Re}} + i\delta\phi_{\mathrm{Im}}$. This can be done only if "pure" modes are obtainable.

1.4.9. Ibrahim

Ibrahim [12] assumes that we are given an analytical model of the system as well as complex measured modes. The first step is to calculate normal modes from the given complex modes. These normal modes are then used in a mass orthogonality condition,

$$\mathbf{u}_N^T \mathbf{M} \mathbf{u}_N = \mathbf{M}_A \ , \tag{1-33}$$

where $\mathbf{M}_A$ is the analytical mass matrix, **M** is an improved mass matrix, and $\mathbf{u}_N$ are the normal modes. The damping matrix can then be given by

$$\mathbf{C} = \mathbf{M}[\mathbf{M}^{-1}\mathbf{C}]. \tag{1-34}$$

The improved mass, **M**, is found in equation (1-33). The $[\mathbf{M}^{-1}\ \mathbf{C}]$ term is found through

$$\left[\mathbf{M}^{-1}\mathbf{K} \quad \mathbf{M}^{-1}\mathbf{C}\right]\begin{Bmatrix}\mathbf{u}_i \\ \lambda_i\mathbf{u}_i\end{Bmatrix} = \left\{\lambda_i^2\mathbf{u}_i\right\} \quad (i = 1,2, \ldots \text{m}) \tag{1-35}$$

$$\left[\mathbf{M}^{-1}\mathbf{K} \quad \mathbf{M}^{-1}\mathbf{C}\right]\begin{Bmatrix}\mathbf{u}_{iN} \\ \lambda_i\mathbf{u}_{Ni}\end{Bmatrix} = \left\{\lambda_i^2\mathbf{u}_{iN}\right\} \quad (i = \text{m+1}, \ldots \text{n}), \tag{1-36}$$

where m is the number of modes available experimentally, and $\mathbf{u}_i$ is a measured complex mode shape. Equation (1-36) uses analytical values for the higher degrees-of-freedom. The limiting factor here is that the order of the improved model can be no more than the number of elements in the measured eigenvectors.

1.4.10. Lancaster

Lancaster's [13] formulation is intended to compute the mass, stiffness and damping matrices of a system directly given only the eigenvalues and eigenvectors. The input data must be normalized in a very specific way for the method to work. In particular, his formulation requires that the mass and damping matrices be used to normalize the eigenvectors, which is then used to calculate the damping matrix.

The Lancaster method is specific to system with only viscous damping where **M, C,** and **K** are symmetric. All of the zeros of the quadratic pencil arise in complex conjugate pairs. That is, the system must be underdamped.

If the eigenvectors are normalized such that

$$\mathbf{u}_i^T(2\mathbf{M}\lambda_i + \mathbf{C})\mathbf{u}_i = 1, \tag{1-37},$$

then the damping matrix can be solved directly with

$$\mathbf{C} = -\mathbf{M}(\Phi\Lambda^2\Phi^T + \overline{\Phi}\overline{\Lambda}^2\Phi^*)\mathbf{M} \,. \tag{1-38}$$

The overbar represents the complex conjugate, and the * is the complex conjugate transpose. Λ is the matrix of eigenvalues, and Φ the matrix of eigenvectors.

Lancaster concludes by stating, "the theory is there, should the experimental techniques ever become available." It is still not possible to measure normalized eigenvectors. The shortfall of this method comes in normalizing the eigenvectors, which requires knowledge of the very same damping matrix which we wish to find in the end.

1.4.11. Roemer And Mook

The Roemer and Mook [14] paper identifies mass, damping and stiffness matrices associated with lumped parameter systems given noisy measurement data. Two time-domain techniques and one estimation technique are combined for an optimal solution. Eigenvalue Realization Algorithm (ERA) uses singular value decomposition (SVD) to make it less noise sensitive, but still has a pretty high threshold. Some noise filtering techniques assume that noise is white or Gaussian, which is not entirely realistic. Fritzen's Instrument Variable method was an improvement, but a problem exists in finding the weighting matrix for his IV method.

SVD is used as a tool to determine model order even with noise. ERA uses the frequency response function to generate a state matrix using Markov parameters. Then, once it has the state matrix it solves for eigenvalues and eigenvectors numerically. In this sense there is no modal synthesis. Minimum Model Error (MME) is used to optimize the combined algorithms. In the Impulse Response Method, the transfer function matrix must be determined with Markov parameters to in turn find **M**, **C**, and K. When noise is present, mass can still be identified, but damping and stiffness are extremely sensitive using this technique. ERA can identify modal properties in low noise situations. The state space model can be transformed into a continuous time model easily, but it is only good for little or no noise.

The state space estimation used is called "Minimum Model Error" (MME) estimation. This assumes that the model error is of unknown quantity (i.e., not like white noise which

is assumed to have a known covariance). The basis of this method is called the "covariance constraint", and requires the following approximation to be satified:

$$R_j = \left\{ [x(t_j) - \mathbf{C}\hat{x}(t_j)][x(t_j) - \mathbf{C}\hat{x}(t_j)]^T \right\}, \tag{1-39}$$

where $\hat{x}$ is an estimated measurement. Once R has been satisfied, the state estimate is optimized. These state estimates are used as new measurement inputs $x(t)$ for the time domain algorithms. An unknown disturbance vector **d** is added to the right hand side of the state matrix. A cost function J = J[c,C,*x*,R,d,W] is minimized, where **W** is a weight matrix (not yet known).

The three methods are combined in this way: original measurements are taken and used in ERA to get a state space form and then used in an impulse response technique (IRT) to estimate **M**. Theses two results are combined to get **M**, **C**, and **K**. The state space formulation is given to MME to find a new estimate for the response which is returned to the beginning. This is an iterative process which continues until a tolerance is converged upon. This combined approach reduces the noise sensitivity seen in other methods. Since the type of error is an unknown in real problems, MME includes model error as part of the solution. It is suggested that the initial estimate of error covariance should be low, and increased until the best "modal amplitude covariance" is found.

In a simulated problem, a cantilevered beam was discretized into a 4 d.o.f. system. The exact solution is known *a priori* so that "white noise" can be added and results compared. The IRT was able consistently to identify correct mass matrices (thus, its purpose in the combined routine). Alone, neither IRT nor ERA could identify all of the components of the damping matrix accurately. Combined, the results are much better.

1.4.12. Starek And Inman

The paper of Starek and Inman [15] presents a solution for the damping identification problem when the desired matrices are both symmetric and positive definite. This method is useful because it preserves the positive definiteness of the resulting matrix. It is based on Lancaster's work using the strict theory of inverse eigenvalue problems for second order systems.

This largely theoretical result presents an inverse eigenvalue problem approach to computing the mass, damping and stiffness matrices for systems of equation (1-1). The results use a mass normalized system so that only the stiffness and damping matrices are determined in the symmetric form $\mathbf{K'}=\mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2}$ and $\mathbf{C'}=\mathbf{M}^{-1/2}\mathbf{C}\mathbf{M}^{-1/2}$. The resulting main contribution is to specify the condition that leads to identifying a nonproportional damping matrix. The formula for computing the mass normalized damping matrix is

$$\mathbf{C'} = -\mathbf{Z} - \mathbf{Z}^* \tag{1-40}$$

where the complex matrix $\mathbf{Z}$ is computed from factoring the matrix polynomial of equation (1-2), written in terms of measured eigenvectors and eigenvalues of the second order form. The matrix $\mathbf{Z}$ is given by

$$\mathbf{Z} = \mathbf{X}_z \mathbf{J}_z (\mathbf{X}_z)^{-1} \tag{1-41}$$

where $\mathbf{X}_z$ is computed from solving a Lyapunov type equation involving an arbitrarily chosen orthogonal matrix Δ of order n, and certain combinations of the measured eigenvectors (complex). The matrix $\mathbf{J}_z$ is essentially the Jordan matrix containing the measured complex eigenvalues.

The method avoids many of the practical questions of noise, incomplete eigenvectors, etc., but does give a firm theoretical foundation useful for small order systems such as machines. In addition, the matrix Δ is diagonal if and only if the system has normal modes, so that non-diagonal choices of Δ will yield complex modes and coupling.

1.4.13. Wang

Wang [16] makes a comparison between a least squares routine and an instrument variable method for parameter identification. The bias problem refers to the effect of noise on the results of the least squares method. Instrument variable methods are used to obtain unbiased estimates. The disadvantages of the instrument variable method are that it is more computationally intensive, because it theoretically needs frequency response functions at an infinite number of frequencies. Also, an iterative procedure is used to obtain a consistent estimation. Practically, it can be very time consuming to measure

many frequency response functions. Wang combines the least squares and the instrument variable methods to form a more accurate method.

1.4.14. Others

Several damping identification routines have been explored. It is appropriate to compare these methods in context of the availability of modal information and computational needs. Factors to consider are reliability, computational intensity, complexity, and methodology. It is hoped that, given all the above information, the reader can make an informed exploration into the field of damping matrix identification. Other papers that address the identification of damping include: Alvin, Park and Peterson [17] (1993), Beattie and Smith [18] (1990), Liang and Lee [19] (1991), Link [20] (1985), Milne [21] (1960), Mottershead and Foster [22] (1988), He and Ewins [23] (1992), and Tseng, Longman and Juang [24] (1993).

One factor that each of these methods has neglected to mention is the ease of implementation for larger (non-trivial sized) problems, and the possibilities for real time implementation. In fact, previous methods are not suitable for real time implementation. In the following chapters, two methods will be presented which do posses these qualities.

Chapter 2

Damping Identification Methods

Two methods of damping matrix identification are presented in this chapter. Both are original methods, serving two different purposes, depending on the amount of reliable data that is available. These methods are not only suitable for use with large problems, but also are able to be entirely automated, making them ideal for real time practical implementation. The first method is based on prior knowledge of a mass matrix and eigenvalues and eigenvectors. The second method requires more information, but is less computationally intensive. This method requires knowledge of the mass and stiffness matrices as well as the eigendata. The tradeoff in methods is one that must be made based on the availability of reliable analytical or finite element models. The information required is from both experimental testing and finite element modeling. The eigenvalues and eigenvectors are data that can be obtained experimentally through modal testing. The mass and stiffness matrices are formed with the knowledge of material properties and system geometry. It can be assumed that an accurate mass matrix is available for most systems. It is also possible to accurately model the stiffness matrix.

2.1. Iterative Damping Matrix Identification Routine

A theory was developed from concepts that Lancaster [13] introduced which were introduced in Chapter 1. The idea is to find a damping matrix starting with normalized data. The concept is clever, but is flawed by the application process. The obstacle turned out to be the very criteria that validated the equation. That is, the normalization that is required before the damping and stiffness matrix may be found is not possible to obtain in practice. This renders the method in its original state virtually useless.

The theory behind the method is described here. The theorems are from an early Lancaster paper [25] on inversion of lambda matrices (also commonly known as the eigenvalue problem). To understand these methods, a bit of notation must first be introduced. Lancaster defines the problem of inverting the general matrix polynomial:

$$\mathbf{D}^{(l)}(\lambda) \equiv (\mathbf{A}_0\lambda^l + \mathbf{A}_1\lambda^{l-1} + \cdots + \mathbf{A}_{l-1}\lambda + \mathbf{A}_l). \tag{2-1}$$

In the following theorems, the notation has been changed from its original definition above to be easily recognized by the structural dynamist and to be consistent with the notation of this work. The problem of interest is the case when l = 2. This gives the quadratic pencil

$$\mathbf{D}^{(2)}(\lambda) \equiv (\mathbf{M}\lambda^2 + \mathbf{C}\lambda + \mathbf{K}), \tag{2-2}$$

which leads to the theorems behind the methodology.

Theorem 1:

If **M** and **K** are real, symmetric n by n matrices, **M** is nonsingular

$$\begin{aligned}(\mathbf{M}\lambda + \mathbf{K})q &= \mathbf{0}\\ (\mathbf{M}^T\lambda + \mathbf{K}^T)r &= \mathbf{0}\end{aligned} \tag{2-3}$$

Then they can be normalized such that

$$\mathbf{R}^T\mathbf{M}\mathbf{Q} = \mathbf{I} \quad \text{and} \quad \mathbf{R}^T\mathbf{K}\mathbf{Q} = -\Lambda. \tag{2-4}$$

In which case,

$$(\mathbf{M}\lambda + \mathbf{K})^{-1} = \mathbf{Q}(\mathbf{I}\lambda - \Lambda)^{-1}\mathbf{R}^T. \tag{2-5}$$

Theorem2:

If M is nonsingular and $D^{(2)}(\lambda)$ has degeneracy equal to the multiplicity of λ_i for i = 1,2,…,2n, then the latent vectors (or eigenvectors) can be normalized in a way such that

$$\lambda^{0,1}\left[\mathbf{D}^2(\lambda)\right]^{-1} = \mathbf{Q}\Lambda^{0,1}(I\lambda - \Lambda)^{-1}\mathbf{R}^T \tag{2-6}$$

and

$$\lambda^2\left[\mathbf{D}^2(\lambda)\right]^{-1} = \mathbf{Q}\Lambda^2(I\lambda - \Lambda)^{-1}\mathbf{R}^T + \mathbf{M}^{-1} \tag{2-7}$$

In the above equations **Q** and **R** are matrices containing the right and left eigenvectors of the quadratic pencil described above. The superscripts 0,1 and 2 are exponents on the eigenvectors. For example, λ^0 is 1 and λ^1 is λ.

By realizing that the right and left matrices of eigenvectors are identical and that the eigenvectors occur in complex conjugate pairs, it is possible using these theorems to generate the equations (1-37) and (1-38). Lancaster states of his own method, "the theory is there, should the experimental techniques ever become available." It is still not possible to measure normalized eigenvectors. The shortfall of this method comes in normalizing the eigenvectors, which requires knowledge of the very same damping matrix which we wish to find in the end. Thus, the obstacle is the very criteria that validates the equations.

The work described from here onward is an extension of what we have just seen. This extension creates a robust viable method from the method left behind by Lancaster. With the implementation of an iterative process [26], it is possible to correctly normalize a system that meets the original criteria set forth by Lancaster. Thus, by bringing in unnormalized data, it is still possible to generate a full damping matrix. This is a more robust method than any found in the literature to date. The method is pictured in Figure 2-1. It should be noted that in addition to the damping matrix, the iterative method can be used to simultaneously solve for a stiffness matrix. Furthermore, this method has another advantage in that it can handle small amounts of noise in the experimental data, and is able to produce reasonable results for reduced systems, as will be illustrated in Chapter 5.

Starting with calculated or experimental values of the mass and the eigensystem, the first step in the procedure involves guessing an initial damping matrix. For an nth order system, this can be any appropriately scaled n dimensional matrix, such as the identity matrix or a modal damping matrix. Next, the eigenvectors must be normalized using

$$\phi_i^T\left(2\mathbf{M}\lambda_i + \mathbf{C}_0\right)\phi_i = 1 \tag{2-8}$$

C is then solved for using equation (1-38). Since the initial guess for **C** is not going to match the new value of **C**, it is necessary to iterate. In the next iteration, the eigenvectors are again normalized, this time using the initial mass matrix and the updated **C** matrix:

$$\phi_i^T \left(2\mathbf{M}\lambda_i + \mathbf{C}_1\right)\phi_i = 1. \tag{2-9}$$

The damping is again calculated using equation (1-38). The iterative procedure continues using an updated damping matrix each time to normalize the eigenvectors until the error between successive damping matrices is small enough to declare convergence.

Most structural systems can be solved using this method with only a few exceptions. The system should be underdamped (in other words, eigenvectors and eigenvalues must occur in complex conjugate pairs). In our experience, the only case where the iterative procedure diverges occurs when the difference between the damping and the mass matrices is small. If the values of the damping matrix are too close to the values of the stiffness matrix, then the iterative procedure will produce a damping matrix that oscillates between two solutions, both near the expected value.

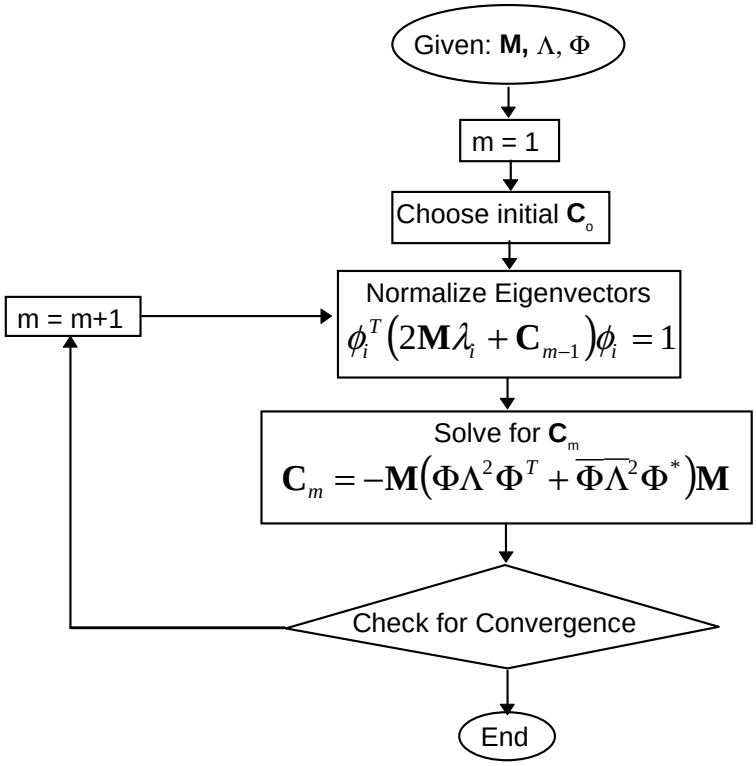


Figure 2-1: Schematic of iterative method

2.2. Direct Method

Given the information, the damping matrix **C** can be computed directly, avoiding the iteration described in the previous section. This direct method also relies on properly normalizing the eigenvectors, and proceeds from the previous normalization equation:

$$\phi_i^T\left(2\mathbf{M}\lambda_i + \mathbf{C}\right)\phi_i = 1 \tag{2-10}$$

Solving for the damping matrix in the time domain can be performed assuming accurate knowledge of the symmetric mass and stiffness matrices, as well as the eigensystem. The eigenvalue problem for the equation of motion can be written as

$$\phi_i^T\mathbf{C}\phi_i = -\phi_i^T\left(\mathbf{K}(1/\lambda_i) + \mathbf{M}\lambda_i\right)\phi_i \tag{2-11}$$

which, when substituted into the previous normalization (Eq. 2-10) yields a new normalization condition. For an underdamped system, if the eigenvectors are normalized such that,

$$\phi_i^T(\mathbf{M}\lambda_i^2 - \mathbf{K})\phi_i = \lambda_i, \tag{2-12}$$

then the symmetric damping matrix may be found through:

$$\mathbf{C} = -\mathbf{M}(\Phi\Lambda^2\Phi^T + \overline{\Phi}\overline{\Lambda}^2\Phi^*)\mathbf{M}, \tag{2-13}$$

where the overbar represents the complex conjugate and * represents the complex conjugate transpose. Φ is a matrix of eigenvectors, and Λ is a diagonal matrix containing the eigenvalues.

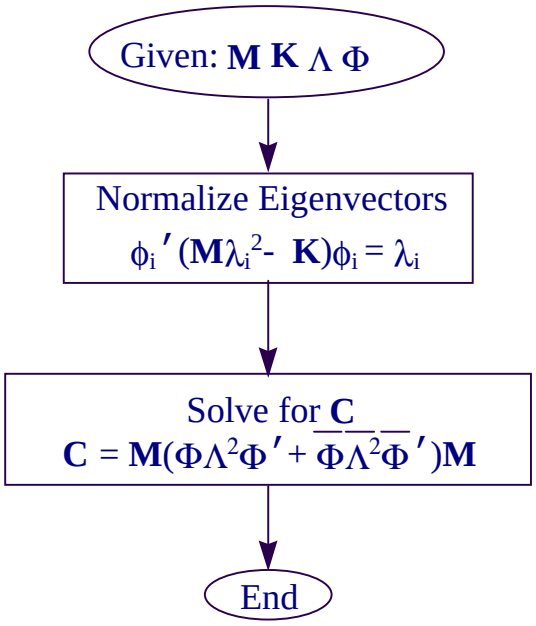


Figure 2-2 Schematic of direct damping identification method

This method will be denoted throughout this work as the "direct method" because it involves no iteration, yet still produces a damping matrix. Although the result is similar, the direct and iterative methods solve different problems because they start with different initial data. While the direct method requires prior knowledge of eigendata as well as

mass and stiffness matrices, the iterative method requires prior knowledge of only an accurate mass matrix and eigendata. Thus, the two methods are not interchangeable.

2.3. Discussion of Positive Definiteness

When identifying a damping matrix, preserving properties of the matrix such as symmetry and positive definiteness becomes an issue. The symmetry is easily seen in figures 2-1 and 2-2, although a positive definite resultant matrix is not as obvious. Because the eigenvectors of the entire structure are preserved using these methods, the definiteness of the structure undergoes no changes through the identification process. The definiteness of the damping matrix, though becomes questionable as the number of available modes decreases. To illustrate this, a lumped mass example similar to that in Figure 5-1 can be used. For a simple ten degree of freedom problem, when ten modes are assumed known, then the resulting damping matrix is both symmetric and positive definite. When less than half of the modes are assumed known, then the identified damping matrix is still symmetric, but is no longer positive definite.

The damping matrix **C** is positive definite when

$$x^T \mathbf{C} x > 0 \qquad \forall \quad x \neq 0 . \tag{2-14}$$

But,

$$x^T \mathbf{C} x = -x^T \mathbf{M} (\Phi \Lambda^2 \Phi^T + \overline{\Phi} \overline{\Lambda}^2 \Phi^*) \mathbf{M} x , \tag{2-15}$$

and, if we substitute

$$y = \mathbf{M} x \tag{2-16}$$

in equation. (2-15), then,

$$x^T \mathbf{C} x = -y^T (\Phi \Lambda^2 \Phi^T + \overline{\Phi} \overline{\Lambda}^2 \Phi^*) y . \tag{2-17}$$

So for **C** to be positive definite we must have

$$y^T (\Phi \Lambda^2 \Phi^T + \overline{\Phi} \overline{\Lambda}^2 \Phi^*) y < 0 \qquad \forall y \in \Re^n . \tag{2-18}$$

One very significant flaw comes when the number of eigenvectors is less than half of the size of the system (n). In equation (2-18), when there are few eigenvectors ϕ, then y could be orthogonal to a vector in Φ, making the quotient zero. Thus, the resulting damping matrix could be semi-definite, or in the worst case, indefinite. When this occurs, it can not be proven that a positive definite damping matrix will ensue. If this is the case, it is best to consider the damping matrix generated by these methods a good first guess, and continue to find the nearest positive definite matrix using the method of Beattie and Smith [18]

Chapter 3

Data Incompleteness and Model Updating

It is important in the investigation of mechanical systems to compare analytical finite element models with experimentally obtained information. The comparison between analytical and experimental data is challenging due to the differences in size of the two types of models. A finite element model may have many thousands of degrees of freedom. Experimental verification is limited due to the physical constraints of modal analysis. Grid points of the experimental model are only available where transducers can be placed and responses measured. This chapter investigates the methods available to compare measured and numerical mode shapes. Model updating is also used to compare finite element models and measure data, and will be discussed in Section 3.3.

3.1. Introduction

From Newton's Law, the equation of motion for an undamped mechanical system can be written as

$$\mathbf{M}\ddot{x} + \mathbf{K}x = \mathbf{f} \tag{3-1}$$

where **M** is the mass of the system, **K** is the stiffness, x is the displacement and **f** is the external force applied to the system. A finite element model can be developed to generate the mass and stiffness matrices based on material properties and geometry of the test system. This can be used to solve for mode shapes and natural frequencies.

Validation of the finite element model is necessary to ensure accuracy and to test any assumptions in the model. An actual test structure is used for this verification, in a procedure known as modal analysis [30]. Laboratory testing produces mode shapes and

natural frequencies of the test structure, which are then compared to the finite element model. The finite element model is often complex, to account for areas of particular interest in the structure. The modal model, on the other hand, is only as large as testing allows. It can be limited by the number of transducers available, and the data analysis hardware capabilities at the testing facility. Model reduction or expansion is the tool used to compare the two models.

There are two ways in which incomplete modal data sets present themselves. The first type, called spatial incompleteness, occurs when the number of degrees of freedom that can be measured is fewer than the number of degrees of freedom in the analytical or the finite element model. This is illustrated in Figure 3-1, where the information inside the brackets represents the matrix of mode shapes or eigenvectors. Each row of the matrix contains information for one of the many degrees of freedom. The total number of rows in the matrix is an indication of the number of degrees of freedom included in the model.

In Figure 3-1, the finite element or analytical model has more degrees of freedom than the experimentally obtained information. Each yellow row represents a "master" degree of freedom, or one which can be measured. The green rows represent degrees of freedom that are not measurable, or are just excluded from the experiment; these are commonly known as the "slave" degrees of freedom.

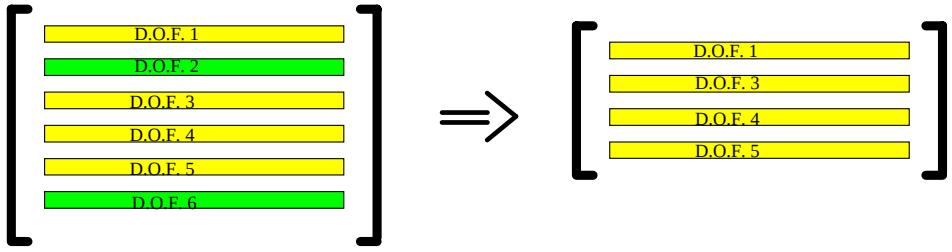
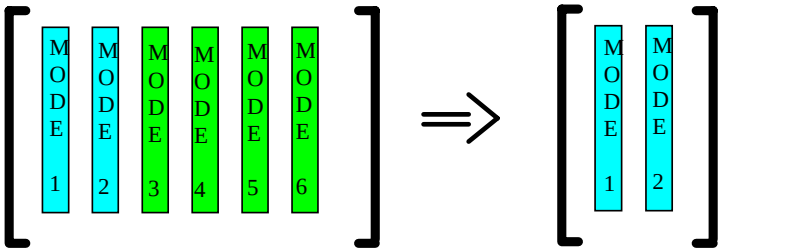


Figure 3-1 Illustration of spatial incompleteness

The second type of data incompleteness occurs because the number of modes that can be accurately measured is far fewer than the number of modes that an analytical or finite element model contains. In Figure 3-2, the information inside the brackets once again

represents the modes shapes or eigenvectors of a system. Each column of the matrix refers to one of several mode shapes. In general, only the lower mode shapes (shown in blue) can be accurately measured. It is commonly accepted that only the lower one third to one half of the mode shapes can be accurately represented. Notice the significantly reduced size of the experimentally obtainable information.



FEM / Analytical Matrix of Eigenvectors **Experimentally Measured Matrix of Eigenvectors**

Figure 3-2 Illustration of modal incompleteness

A complete understanding of data incompleteness becomes apparent when the two examples above are combined. Figure 3-1 and Figure 3-2 can be overlaid, so that not only are the rows of slave degrees of freedom eliminated, but also columns of higher modes are removes, then the final matrix is smaller than the original matrix in two dimensions. This illustrates the true size discrepancy faced when comparing finite element models and experimental data.

3.2. Spatial Incompleteness

Model reduction or expansion first involves partitioning the larger, finite element model into measured and unmeasured degrees of freedom. Automated procedures are available to aid in choosing optimal measurement locations on the experimental model [31]. Equation (3-2) shows the partitioned equation of motion:

$$\left[\begin{array}{c|c} \mathbf{M}_{mm} & \mathbf{M}_{ms} \\ \hline \mathbf{M}_{sm} & \mathbf{M}_{ss} \end{array}\right]\left\{\begin{array}{c} \ddot{\mathbf{x}}_m \\ \hline \ddot{\mathbf{x}}_s \end{array}\right\}+\left[\begin{array}{c|c} \mathbf{K}_{mm} & \mathbf{K}_{ms} \\ \hline \mathbf{K}_{sm} & \mathbf{K}_{ss} \end{array}\right]\left\{\begin{array}{c} \mathbf{x}_m \\ \hline \mathbf{x}_s \end{array}\right\}=\left\{\begin{array}{c} \mathbf{f}_m \\ \hline \mathbf{0} \end{array}\right\}. \tag{3-2}$$

The subscript *m* refers to the measured degrees of freedom, and *s* to the unmeasured degrees of freedom, where $n = m + s$.

The partitioning (3-2), is used as the basis for all of the methods for model reduction and expansion described below.

3.2.1. Static Reduction and Expansion

Static reduction was first introduced by Guyan [32]. It is used most frequently by finite element packages because of its relative simplicity. Static reduction is so named because it neglects the inertia term in the equation of motion. Neglecting the inertia term in equation (3-2), we are left with the two expressions:

$$\mathbf{K}_{sm}\mathbf{x}_m + \mathbf{K}_{ss}\mathbf{x}_s = \mathbf{0} \tag{3-3}$$

and

$$\begin{Bmatrix} \mathbf{x}_m \\ \hline \mathbf{x}_s \end{Bmatrix} = \mathbf{T_s}\{\mathbf{x}_m\} \tag{3-4}$$

where $\mathbf{T}_s$ is the static reduction / expansion transformation matrix defined by:

$$\mathbf{T_s} = \begin{bmatrix} \mathbf{I} \\ \hline -\mathbf{K}_{ss}^{-1}\mathbf{K}_{sm} \end{bmatrix}. \tag{3-5}$$

Equation (3-5) can be used to either expand the mode shape vector from *m* degrees of freedom to the full *n* degrees of freedom, or it can be used to create reduced mass ($\mathbf{M}_r$) and stiffness ($\mathbf{K}_r$) matrices as follows:

$$\mathbf{M}_r = \mathbf{T}_s^t\mathbf{M}\mathbf{T}_s \tag{3-6}$$

and

$$\mathbf{K}_r = \mathbf{T}_s^t\mathbf{K}\mathbf{T}_s. \tag{3-7}$$

These last two expressions are used in model reduction.

3.2.2. Dynamic Reduction and Expansion

By including the effects of inertia, the accuracy of the expansion process is increased. In dynamic expansion, a chosen frequency of interest can be used to create an accurate transformation matrix. It is also possible to create a separate transformation matrix for each natural frequency measured. This increases accuracy significantly, but at a considerable cost computationally. The dynamic transformation [33] is given by:

$$\mathbf{T_d} = \begin{bmatrix} \mathbf{I} \\ \hline -(\mathbf{K}_{ss} - \omega_i^2 \mathbf{M}_{ss})^{-1}(\mathbf{K}_{sm} - \omega_i^2 \mathbf{M}_{sm}) \end{bmatrix} \tag{3-8}$$

where $\mathbf{T_d}$ is the transformation matrix for dynamic expansion. The reduced mass and stiffness matrices are formed in the same manner seen in section 3.2.1.

3.2.3. Improved Reduced System (IRS)

The Improved Reduced System (IRS) method is modeled after static condensation. Although more computationally intensive, IRS provides a better approximation of the model by including an extra term that makes some allowance for the inertia forces. The transformation [34] is given by:

$$\mathbf{T}_{IRS} = \mathbf{T}_s + \mathbf{SMT}_s\mathbf{M}_r^{-1}\mathbf{K}_r \tag{3-9}$$

where the matrix **S** is singular and given by

$$\mathbf{S} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{ss}^{-1} \end{bmatrix}. \tag{3-10}$$

$\mathbf{T}_{\text{IRS}}$ is the transformation matrix for the IRS method.

3.2.4. Iterated IRS

Recently, the IRS method has been improved and extended to form an iterated IRS method [35] [36]. The basic transformation for this method comes from dynamic reduction, as opposed to static reduction for the traditional IRS method. In addition, a corrective term is generated iteratively using the best estimate for the reduced model at

each iteration. Friswell, et al. [37] have been able to demonstrate and prove that the natural frequencies of the reduced model converge to those of the full model.

The transformation for the iterated IRS method is obtained from

$$\mathbf{T}_{i+1} = \begin{bmatrix} \mathbf{I} \\ \mathbf{t}_{i+1} \end{bmatrix} \tag{3-11}$$

where,

$$\mathbf{t}_{i+1} = \mathbf{t}_s + \mathbf{K}_{ss}^{-1}\begin{bmatrix}\mathbf{M}_{sm} & \mathbf{M}_{ss}\end{bmatrix}\mathbf{T}_i\mathbf{M}_{Ri}^{-1}\mathbf{K}_{Ri} \tag{3-12}$$

with,

$$\mathbf{t}_0 = \mathbf{t}_s = -\mathbf{K}_{ss}^{-1}\mathbf{K}_{sm}\,. \tag{3-13}$$

The reduced mass and stiffness matrices at the ith iteration are defined as

$$\mathbf{M}_{Ri} = \mathbf{T}_i^T\mathbf{M}\mathbf{T}_i \tag{3-14}$$

and

$$\mathbf{K}_{Ri} = \mathbf{T}_i^T\mathbf{K}\mathbf{T}_i\,. \tag{3-15}$$

3.2.5. Comparison of Methods

We tested three methods of model reduction for accuracy. Static reduction, Improved Reduced System (IRS), and the Iterated IRS methods have been applied to several problems, with varying numbers of master and slave degrees of freedom. The Iterated IRS method consistently out performed the others when a comparison of natural frequencies of the full and reduced systems was made. The data in Table 3-1, based on a 40 degree of freedom plate problem, is typical of the relative performance of the three methods. Other examples tested include simple lumped mass models, and beam problems. Similar results are found in literature [35] [36].

Table 3-1 First 10 analytical natural frequencies of a 40 DOF plate, and the natural frequencies of the same plate when reduced to 10 degrees of freedom using static reduction, IRS and Iterated IRS (5 iterations)

natural frequencies			
Analytical	Static	IRS	IIRS
1.4539	1.6092	1.4597	1.4539
1.7328	1.8967	1.7612	1.7328
2.1124	2.5745	2.1478	2.1124
2.9843	4.1614	3.1871	2.9843
3.8415	5.5350	4.3389	3.8417
4.7414	6.3513	5.0971	4.7420
4.8596	7.3763	6.1126	4.8646
5.8576	8.3759	7.2305	5.8632
5.9945	11.0274	8.0586	5.9984
6.3137	13.0187	11.0303	6.3488

3.2.6. Choice of Master Degrees of Freedom

Methods exist to aid in making an optimal choice of master degrees of freedom [38]. This is done by removing the least significant degree of freedom as determined by the ratio of stiffness to mass. The slave degree of freedom is chosen where the stiffness is high and the inertia is low, so that if the ratio of the diagonal terms k_{ii}/m_{ii} is large, then the ith coordinate is selected as the slave coordinate. After each selection, a reduced model is calculated before the next slave degree-of-freedom is removed.

3.2.7. Inclusion of Damping Term in Model Reduction and Expansion

Model reduction and expansion for damped systems is rarely addressed, and is a relatively immature area of study. The following sections describe some general thoughts on expanding the previously described reduction methods to include a damping term. This is not a highly publicized topic because finite element models currently generally exclude damping.

3.2.7.1.Static Expansion including Damping

Hysteretic damping (also known as structural damping) is developed as follows:

$$m\ddot{x} + (\gamma k / \omega)\dot{x} + kx = fe^{i\omega t} \tag{3-16}$$

where $\dot{x} = i\omega x$. γ is the structural damping factor (sometimes called damping loss factor). ω is a natural frequency. The equation above can be written as

$$m\ddot{x} + k(1 + i\gamma)x = fe^{i\omega t} \tag{3-17}$$

or,

$$\mathbf{M}\ddot{x} + (\mathbf{K} + i\mathbf{C})x = fe^{i\omega t} \tag{3-18}$$

Thus, traditional static expansion is modified in the following ways:

$$\left[\begin{array}{c|c} \mathbf{M_{mm}} & \mathbf{M_{sm}} \\ \hline \mathbf{M_{ms}} & \mathbf{M_{ss}} \end{array}\right]\left\{\begin{array}{c} \ddot{x}_m \\ \hline \ddot{x}_s \end{array}\right\} + \left[\begin{array}{c|c} \mathbf{K_{mm}} + i\mathbf{C_{mm}} & \mathbf{K_{sm}} + i\mathbf{C_{sm}} \\ \hline \mathbf{K_{ms}} + i\mathbf{C_{ms}} & \mathbf{K_{ss}} + i\mathbf{C_{ss}} \end{array}\right]\left\{\begin{array}{c} x_m \\ \hline x_s \end{array}\right\} = \left\{\begin{array}{c} f_m \\ \hline 0 \end{array}\right\} \tag{3-19}$$

Neglecting the inertia term, we are left with:

$$(\mathbf{K}_{sm} + i\mathbf{C}_{sm})x_m + (\mathbf{K}_{ss} + i\mathbf{C}_{ss})x_s = 0 \tag{3-20}$$

$$\left\{\begin{array}{c} x_m \\ \hline x_s \end{array}\right\} = \mathbf{T}_s\left\{x_m\right\} \tag{3-21}$$

where $\mathbf{T}_s$ is the new static expansion transformation matrix.

$$\mathbf{T_s} = \left[\begin{array}{c} \mathbf{I} \\ \hline -(\mathbf{K_{ss}} + i\mathbf{C_{ss}})^{-1}(\mathbf{K_{sm}} + i\mathbf{C_{sm}}) \end{array}\right] \tag{3-22}$$

3.2.7.2.Dynamic Reduction/Expansion and IRS Including Damping

Similarly, a dynamic transformation matrix can be formed.

$$\mathbf{T}_d = \left[\begin{array}{c} \mathbf{I} \\ \hline -(\mathbf{M}_{ss}\lambda^2 + \mathbf{C}_{ss}\lambda + \mathbf{K}_{ss})^{-1}(\mathbf{M}_{sm}\lambda^2 + \mathbf{C}_{sm}\lambda + \mathbf{K}_{sm}) \end{array}\right] \tag{3-23}$$

Where $\mathbf{T_d}$ is the new transformation matrix for dynamic expansion or reduction.

The IRS transformation matrix could include a damping term by simply using equation (3-22) in place of $\mathbf{T}_s$ of equation (3-9).

3.3. Model Updating of Damping and Stiffness Matrices

Model updating is often used to correlate finite element models and measured data. This is described several excellent works by Friswell and Mottershead [39] [40]. Some of the earliest methods were introduced by Baruch [41], who assumed that the mass matrix was known exactly, and that only the stiffness matrix needed to be updated. This type of method is described as a reference basis method [42] because one quantity (the mass matrix) is assumed to be exact, and the other quantities - the modal data and the stiffness matrix - are updated.

A recent paper by Friswell, Inman and Pilkey [43] concentrates on a direct updating method based on measured modal data. It assumes that the mass matrix is correct, and the damping and stiffness matrices are updated simultaneously, so that the updated model reproduces the measured modal data. The theory behind this method follows.

Baruch's method updated only the stiffness matrix. Following this method, the penalty or objective function *J* of equation (3-24) minimizes the difference between the initial and updated damping and stiffness matrices simultaneously. The constraints of the eigenvalue equation are satisfied and the damping and stiffness matrices are symmetric and real. So, the function to minimize is

$$J = \frac{1}{2}\left\|\mathbf{N}^{-1}\left[\mathbf{K}-\mathbf{K}_a\right]\mathbf{N}^{-1}\right\|^2 + \frac{1}{2}\mu\left\|\mathbf{N}^{-1}\left[\mathbf{C}-\mathbf{C}_a\right]\mathbf{N}^{-1}\right\|^2 \tag{3-24}$$

subject to

$$\mathbf{M}_a\Phi\Lambda^2 + \mathbf{C}\Phi\Lambda + \mathbf{K}\Phi = \mathbf{0} \tag{3-25}$$

$$\mathbf{C} = \mathbf{C}^T \qquad \mathbf{K} = \mathbf{K}^T \tag{3-26}$$

where $\mathbf{N} = \mathbf{M}_a^{1/2}$, $\mathbf{M}_a$, $\mathbf{C}_a$ and $\mathbf{K}_a$ are the initial, analytical mass, damping and stiffness matrices, $\mathbf{C}$ and $\mathbf{K}$ are the updated damping and stiffness matrices, and Φ and Λ

are the measured eigenvector and eigenvalue matrices. A full set of modes are not measured, so Φ is not square, but all degrees of freedom are assumed measured (otherwise mode shape expansion or model reduction must first be used). Λ is a diagonal matrix with the measured eigenvalues on the diagonal. The parameter μ in equation (3-24) is to enable the damping and stiffness terms to be weighted. Often the magnitude of the stiffness terms are far greater than the damping terms, and so if μ was not present, more weight would be given to the stiffness terms, leading to a poor estimate of the damping matrix. The value of μ can be selected based on experience. Otherwise, the norms of the change in damping and stiffness matrices is plotted, and an L shaped curve will result. A range of values is tried and the optimum value is chosen based on the L curve type criterion in regularization [44]. Note that Wei [45]-[47] produced a similar method for updating mass and stiffness simultaneously, although he didn't include a weighting factor similar to μ.

The Lagrange multiplier method is used to solve the optimization problem. The augmented penalty function based on equation (3-24) and the constraints is

$$\begin{aligned} J = \frac{1}{2}\left\|\mathbf{N}^{-1}\left[\mathbf{K}-\mathbf{K}_a\right]\mathbf{N}^{-1}\right\|^2 + \frac{1}{2}\mu\left\|\mathbf{N}^{-1}\left[\mathbf{C}-\mathbf{C}_a\right]\mathbf{N}^{-1}\right\|^2 + \\ \sum_{i,j=1}^{n}\gamma_{\mathrm{K}ij}\left(k_{ij}-k_{ji}\right) + \sum_{i,j=1}^{n}\gamma_{\mathrm{C}ij}\left(c_{ij}-c_{ji}\right) + \\ 2\sum_{i=1}^{n}\sum_{j=1}^{m}\gamma_{\Lambda ij}\sum_{h=1}^{n}\left(k_{ih}\phi_{hj}+c_{ih}\phi_{hj}\lambda_j+m_{ih}\phi_{hj}\lambda_j{}^2\right) + \\ 2\sum_{i=1}^{n}\sum_{j=1}^{m}\bar{\gamma}_{\Lambda ij}\sum_{h=1}^{n}\left(k_{ih}\bar{\phi}_{hj}+c_{ih}\bar{\phi}_{hj}\bar{\lambda}_j+m_{ih}\bar{\phi}_{hj}\bar{\lambda}_j{}^2\right) \end{aligned} \tag{3-27}$$

where k_{ij} is the (i,j) element of **K** and similarly for **C** and Φ, λ_j is the jth eigenvalue (or the (j,j) element of Λ), n is the number of degrees of freedom, and m is the number of measured modes. The third and fourth terms ensure the updated damping and stiffness matrices are symmetric, and the last two terms ensure that the eigenvalue equation is satisfied. The Lagrange multipliers may be formed into three matrices in obvious notation: Γ_{K} and Γ_{C} which are real and skew-symmetric (since otherwise they would

not be unique) and Γ_Λ which is complex. Because the eigenvalues and eigenvectors are complex, so must the corresponding Lagrange multiplier, Γ_Λ, and the last term in equation (3-28) ensures that *J* is real.

Differentiating *J* with respect to each element of **K** and assembling the results into a matrix gives the following equation,

$$\mathbf{M}_a^{-1}\left[\mathbf{K}-\mathbf{K}_a\right]\mathbf{M}_a^{-1}+2\Gamma_K+2\Gamma_\Lambda\Phi^T+2\overline{\Gamma}_\Lambda\overline{\Phi}^T=\mathbf{0} \tag{3-28}$$

Similarly differentiating with respect to the damping matrix gives,

$$\mu\,\mathbf{M}_a^{-1}\left[\mathbf{C}-\mathbf{C}_a\right]\mathbf{M}_a^{-1}+2\Gamma_C+2\Gamma_\Lambda\,\Lambda\Phi^T+2\overline{\Gamma}_\Lambda\,\overline{\Lambda}\,\overline{\Phi}^T=\mathbf{0} \tag{3-29}$$

Using the skew symmetry of Γ_K and the symmetry of the mass and stiffness matrices, Γ_K can be eliminated from equation (3-29) to give,

$$\mathbf{K}=\mathbf{K}_a-\mathbf{M}_a\left[\Gamma_\Lambda\Phi^T+\Phi\Gamma_\Lambda{}^T+\overline{\Gamma}_\Lambda\overline{\Phi}^T+\overline{\Phi}\overline{\Gamma}_\Lambda{}^T\right]\mathbf{M}_a \tag{3-30}$$

Similarly for the damping matrix

$$\mathbf{C}=\mathbf{C}_a-\frac{1}{\mu}\mathbf{M}_a\left[\Gamma_\Lambda\,\Lambda\Phi^T+\Phi\Lambda\Gamma_\Lambda{}^T+\overline{\Gamma}_\Lambda\,\overline{\Lambda}\,\overline{\Phi}^T+\overline{\Phi}\,\overline{\Lambda}\,\overline{\Gamma}_\Lambda{}^T\right]\mathbf{M}_a \tag{3-31}$$

If we knew the Lagrange multiplier matrix Γ_Λ we could calculate the updated damping and stiffness matrices from equations (3-30) and (3-31). We can obtain a set of equations for this Lagrange multiplier matrix by combining equations (3-30) and (3-31) with the constraint Equation (3-25), to give,

$$\begin{aligned}&\mathbf{M}_a\left[\Gamma_\Lambda\,\Phi^T+\Phi\Gamma_\Lambda{}^T+\overline{\Gamma}_\Lambda\,\overline{\Phi}^T+\overline{\Phi}\,\overline{\Gamma}_\Lambda{}^T\right]\mathbf{M}_a\,\Phi+\\&\frac{1}{\mu}\mathbf{M}_a\left[\Gamma_\Lambda\,\Lambda\Phi^T+\Phi\Lambda\Gamma_\Lambda{}^T+\overline{\Gamma}_\Lambda\,\overline{\Lambda}\,\overline{\Phi}^T+\overline{\Phi}\,\overline{\Lambda}\,\overline{\Gamma}_\Lambda{}^T\right]\mathbf{M}_a\,\Phi\Lambda=\\&\qquad\qquad\mathbf{M}_a\,\Phi\Lambda^2+\mathbf{C}_a\,\Phi\Lambda+\mathbf{K}_a\,\Phi\end{aligned} \tag{3-32}$$

Equation (3-32) is a set of 2*nm* equations (real and imaginary parts) for the 2*nm* elements of the matrix Γ_Λ. Baruch (and Wei) simplified these equations to give closed form solutions, but because of the factor μ and also because the eigenvector

normalization is not so straight forward we do not attempt a closed form solution. Even so we have an equation to obtain Γ_Λ and then the updated damping and stiffness matrices may be obtained from equations (3-30) and (3-31).

This section has outlined the extension to direct methods of model updating to estimate both the damping and stiffness matrices in a structure. The algorithm does not guarantee the preservation of positive definiteness. Also, the connectivity of the finite element model is not necessarily preserved. Even so, by minimizing the change in damping and stiffness matrices, and ensuring that the measured modal data is reproduced complete structural matrices can be updated.

An in depth investigation of this method, would include several examples of updating and formation of L curves to aid in determination of the μ factor. Completeness may also include a computational investigation of the algorithm. This method of model updating is not explored further, and is left as future work.

Chapter 4

Computational Issues

4.1. Introduction

When faced with potentially large problems such as damping matrix identification, it is necessary to investigate the practical issues of solving the problem. An important practical consideration in evaluating algorithms for damping matrix identification is the performance on high performance computer architectures, which are required to solve large problems in "reasonable" time. As part of this investigation, we have evaluated the performance of our algorithms on multiple computers, including personal PC's, SUN Sparc stations running UNIX, IBM SP-2's at NASA Ames and Langley Research Centers with over 100 processors (which were later reduced to fewer than 30 and then decommissioned altogether), and an Intel Paragon located on the campus of Virginia Tech. Results are not reported for all of the above machines, but only those deemed appropriate for each investigation, and where a complete set of data is available.

A distinguishing characteristic of today's high performance systems (e.g., the SP-2's and the Paragon) is parallelism. In parallelizing the routines, some of the issues to consider are problem size, sparsity patterns, memory constraints, language limitations and available linear solvers. The problem size, as well as the type and number of computations that are required, will be a determining factor in the choice of computer architecture and number of processors needed. One important issue when investigating available computing resources is memory constraints. The amount of data stored on each processor is limited by the memory constraints. Thus, larger problems will tend to work most efficiently with more processors.

The goal of these investigations is to study and optimize the available capabilities of high performance computing (HPC) for this particular problem. The hope is that a 'real time' solution will ultimately be reasonable for large problems. The investigation includes a comparison of model expansion/reduction routines, as well as a computational investigation into both the direct and iterative methods of damping matrix identification. Finally, a comparison between the iterative damping identification method and a least squares method introduced in Chapter 1 is made to illustrate the computational benefits of the methods of this work over others.

4.2. High Performance Fortran

High Performance Fortran (HPF) [48] was chosen as the programming language for our implementations because of its relatively low learning curve, making it ideal for engineering problems. Also, according to M. Wilkes [49], "Fortran is still by far the most popular language for numerical computation", making it the ideal language for this study. High Performance Fortran is one of the most widely used parallel computing languages. It is best suited for large "data-parallel" applications such as is common in dense numerical linear algebra. Hence, it is an obvious candidate for implementing expansion/reduction and damping matrix identification algorithms. Generally, to develop an HPF code, a Fortran 90 code is written first to run the program sequentially, and HPF directives are added which help the HPF compiler parallelize the program. Since HPF is a relatively modest extension to Fortran 90 (from a programmers point of view), it is not difficult to get a parallel "first working version" of a code implemented in Fortran. Developing a more efficient parallel code is then a matter of working with details of data distribution and parallelizing loops. Some of these details are explained in the appendices.

4.3. Application of HPC to model expansion/reduction

There are several issues to consider when looking into the applicability of high performance computing to the model reduction problem [50]. The first is problem size. In the comparison of finite element models and modal data, the finite element model can be extremely large, and the modal data relatively small. In addition, the number of

modes, and the number of elements of a mode are generally smaller than the analytical model. Sparsity is another issue to deal with. In general, we can assume certain sparsity patterns in each finite element model, but can expect the pattern to change from model to model.

In this section, we study parallel performance of straightforward implementations of the model reduction algorithms using the High Performance Fortran [48] (HPF) programming language. We report performance results on an Intel Paragon distributed memory parallel computer with up to 32 processors.

The problem size of interest for this application is one in which the number of measured degrees of freedom (*m*) is significantly smaller than the number of unmeasured degrees of freedom (*s*). High performance computing is only necessary when the finite element model is large.

All reduction methods considered require the solution of linear systems of equations involving the *s* x *s* matrices $\mathbf{K}_{ss}$ and/or $\mathbf{M}_{ss}$ (see Eq. (3-2)). The sparsity pattern of the mass and stiffness matrices will be a determining factor in the choice of inverse solvers. It is expected that the matrices in a structural problem will be banded, and that the bandwidth will be different for different types of problems. Thus, in each problem the sparsity pattern should be considered, as it is a major factor in parallelization. It is also important for reducing the overall computational cost that the bandwidth be as small as possible. Thus, when the finite element model is generated, a bandwidth optimizer should be used.

Static reduction / expansion and the Improved Reduced System (IRS) methods are investigated in this work. The Iterated IRS method discussed in Chapter 3 is computationally similar at each iteration to the IRS method, as is dynamic to static reduction. The methods considered in this work require the solution of one or more matrix equations $\mathbf{AX}=\mathbf{B}$, where $\mathbf{A}$ and $\mathbf{B}$ are known and $\mathbf{X}$ is unknown. Interestingly, although these steps are expensive, they are not necessarily the dominant step in the computation because the matrices $\mathbf{A}$ are either sparse or small. Other steps involving large dense matrix-matrix multiplications are generally even more expensive than the linear solves. Nevertheless, it is important to implement these linear solves efficiently.

In the static and dynamic methods, we must solve a matrix equation with a large (s x s) banded symmetric positive definite matrix. Because Cholesky factorization on narrowly banded matrices is not an easily parallelized computation, our implementation factors these matrices redundantly (on every processor). Then, given the factorization, the triangular solves needed to compute the columns of **X** can all be done in parallel and with no communication, assuming corresponding columns of **X** and **B** are located on the same processor. Since the factorization of the banded matrix is relatively inexpensive, this extra non-parallelized work is not significant for modest numbers of processors.

Compared to the static or dynamic algorithms, the IRS expansion algorithm requires several additional matrix-matrix multiplications, which parallelize well, and an additional matrix solve involving a small (m x m) dense matrix. We factor this matrix redundantly as well, and then do the forward and back solves in parallel.

There are several data mapping strategies to choose from in HPF (see Appendix B). The intrinsic `distribute` allows one to control the distribution of arrays on to processors by specifying the block or cyclic distributions. Our experiments show that the `(*,block)` distribution, where each processor holds a block column of the matrices, is close to optimal for the matrix-matrix multiplication (the `(block,block)` distribution was found to be at most 20% better in our experiments). And, since `(*, block)` allows the matrix solves to be done in parallel with no communication, it is the distribution we use for all the spatial incompleteness algorithms.

4.3.1. Illustrative Examples

Two examples are used to show the application of reduction algorithms to structural mechanics. First, a beam is generated, and artificially enlarged so as justify the use of high performance computing. The matrices arising from the model of the beam are banded with a very small bandwidth. Next, a plate is modeled with quadrilateral plate elements. This is an excellent example of how structural finite element problems can become large, when a fine mesh is desired to ensure accuracy. The beam reflects problems with small bandwidths, and the plate, larger bandwidths.

4.3.1.1. Beam Example

As an illustration of the computational aspects of the methods described, we consider a Bernoulli - Euler cantilevered beam. The beam is modeled with Hermitian shape functions, and for the purpose of this example has 500 elements. Obviously, a simple beam can be modeled with far fewer elements, but because our purpose is computational in nature, this example was chosen and artificially enlarged. Other problems, which are more difficult to generate, but require more degrees of freedom to accurately model, may include damaged structures, or those with known irregularities. The computational characteristics of more realistic examples of this kind should be similar to the present example.

We assume that of the 1000 degrees of freedom, only 100 are measured. In addition, the 'measured' eigenvalues occur in the first third of the frequency band, so there are 334 measured eigenvectors of interest. This restriction is necessary because the finite element model can only predict the lower portion of the natural frequencies and mode shapes accurately. In addition, model expansion methods are most accurate in the lower frequency range.

The 100 'measured' degrees of freedom are chosen so that an equal number of translational and rotational degrees of freedom are chosen in pairs. So, if degree of freedom i is measured and is translational, then degree of freedom i+1 is also measured. The pairs are equally spaced, so that measurements are taken along the entire beam. As a result of these assumptions, the matrix $\mathbf{K}_{ss}$ is banded, with half bandwidth 3.

4.3.1.2.Plate Example

A rectangular steel block is used as an example plate problem to test the reduction routines. The symmetry conditions of the rectangular plate allow it to be divided into four equal sections. The boundary conditions reflect such a simplified problem. The system is modeled with four noded quadrilateral elements. The consistent mass matrix and stiffness matrix are assembled in a way that allows for two degrees of freedom at each node. The global matrices are assembled with the use of a destination array. This array makes use of the homogeneous essential boundary conditions to distinguish between the active and passive degrees of freedom.

The beam is assembled so that it has 1012 active degrees of freedom. It is assumed that 102 of those are 'measured'. In addition, the 'measured' eigenvalues occur in the first third of the frequency band, giving 338 eigenvectors of interest. The bandwidth of the resulting partitioned matrix $\mathbf{K}_{ss}$ is 477. This example allows us to make a comparison between model reduction methods based on bandwidth.

4.3.2. Results and Conclusions

Table 4-1 and Table 4-2 report time, parallel speedup, and parallel efficiency (defined as the ratio of speedup to number of processors) for the static and IRS methods on the two example problems. We do not report any results for dynamic reduction because with ω_i fixed, its performance is identical to the static method. Also, the Iterated IRS method is similar computationally to the IRS, so it was also neglected in this study.

The static code achieves better than 60% parallel efficiency on the beam problem up through 16 processors, as is evident in Table 4-1. The performance for the plate problem (Table 4-2) trails off more quickly because of the cost of the redundant factorization for the large bandwidth problem. This redundant factorization becomes the parallel bottleneck for the large bandwidth problem, as it takes over 50% of the total solution time on 16 and 32 processors. For the smaller bandwidth problem, the matrix-matrix multiplication takes the bulk of the time - over 80% on any number of processors - but it parallelizes well, as it is an intrinsic HPF function.

Using the IRS code, good parallel performance is achieved up through 16 processors for both examples. Better than 74% efficiency is realized. The IRS method is dominated by matrix multipications and multiple triangular solves, making the cost of the redundant factorizations less significant.

A comparison between the two methods shows that the IRS method benefits relatively more from parallel computing than the static method. For the larger bandwidth plate problem, IRS is 11.2 times slower than the static solution on one processor. When running on 16 processors IRS is only 4.8 times slower than the static solution. For the small bandwidth problem, the change in the gap between the two methods is not as dramatic. Running on one processor, IRS is 16.6 times slower than the static solution to

the beam example. On 16 processors, IRS is 13.7 times slower than the static solution. Clearly, the IRS method of model reduction is more computationally intensive than the either static or dynamic reduction. The issue of speed versus accuracy must be considered. IRS is considerably more accurate than static condensation [51]. The tradeoff will have to be determined by the user's specific needs for accuracy and speed. Problems of industrial interest tend to not only be large, but possess large bandwidth, making the IRS method much more competitive with static condensation in the high performance computing context.

Besides looking at traditional measures such as speedup and efficiency, parallel performance can also be evaluated by considering computational rate --- specifically, MFLOPS per processor (see Appendix A) ---- to see how the performance scales as the number of processors grows. In Table 4-3 we report computational rates for the two methods on our two test problems. Note that we are using a fixed problem size, whereas problem size is often scaled with the number of processors in traditional scalability experiments. However, we can still learn something from this data. Although the computational rates degrade only moderately up through 16 processors in all cases, the actual MFLOP rates are disappointing - compared with a peak rate of 50 MFLOPS for one Intel Paragon processor. A simple experiment shows that the major reason for this low computational rate is the inefficiency of HPF's intrinsic matrix multiplication routine. For example, on one processor, multiplying a 900 x 100 matrix by a 100 x 334 matrix takes over 13 times as long using the HPF intrinsic `matmul` (as implemented by the pghpf compiler and run-time library) as it does using the BLAS [52] [53] routine `dgemm`. The corresponding MFLOP rates are 2.6 and 35.8 respectively. Unfortunately, the Portland Group, developers of the HPF compiler for the Paragon, have indicated that they have no immediate plans for fixing this problem [54]. Since matrix-matrix multiplication is relatively more important for the IRS method, we see lower computational rates there than for the static method. The static method achieves a higher computational rate for the plate problem than for the beam because the triangular solves, which run very efficiently, are a larger percentage of the overall computation for this larger bandwidth case.

Using the above examples, decent parallel performance can only be expected up through 16 processors because of the problem size. For example, with just over 300 eigenvectors divided across 32 processors, each processor is left with just over 10 columns. This means that the computation time of operations such as matrix-matrix multiplication will be dominated by communication costs. Larger problems would be expected to produce better efficiency using more processors.

Table 4-1: Results for parallelized model reduction algorithms applied to 1000 degree of freedom beam example

	Static			IRS		
Processors	time(sec)	speedup	efficiency	time(sec)	speedup	efficiency
1	26.08	-	-	432.16	-	-
2	13.28	1.97	0.99	213.41	2.03	1.01
4	6.89	3.80	0.95	104.09	4.15	1.04
8	3.94	6.65	0.83	66.49	6.50	0.81
16	2.61	9.98	0.62	35.69	12.11	0.76
32	2.65	9.89	0.31	32.82	13.17	0.41

Table 4-2: Results for parallelized model reduction algorithms applied to 1012 degree of freedom plate example.

	Static			IRS		
Processors	time(sec)	speedup	efficiency	time(sec)	speedup	efficiency
1	47.80	-	-	533.30	-	-
2	26.10	1.82	0.91	249.44	2.14	1.07
4	15.66	3.04	0.76	124.29	4.29	1.07
8	10.94	4.40	0.55	74.83	7.13	0.89
16	9.25	5.14	0.32	45.11	11.82	0.74
32	8.42	5.63	0.18	36.87	14.47	0.45

Table 4-3: Computational rate (MFLOPS per processor) for static and IRS reduction applied to two test problems.

	static		IRS	
processors	beam	plate	beam	plate
1	2.44	17.93	2.21	3.68
2	2.40	16.42	2.24	3.93
4	2.31	13.68	2.30	3.94
8	2.02	9.79	1.80	3.27
16	1.53	5.79	1.67	2.72
32	0.75	3.18	0.91	1.66

4.4. Iterative method vs. least square - a computational look

A factor not yet investigated in damping identification is how the problem size affects the ability to solve the inverse problem. Although example problems used in published papers are able to prove that a method is capable of producing the desired results for small problems, it is usually not mentioned whether a particular method is still viable when the problem size is increased to practical limits with many degrees of freedom. Most inverse problems in damping identification require some sort of iteration or optimization procedure. These can become very costly with the increasing problem size in regards to execution time and computer memory limitations [55].

The increased execution time and required memory creates a need for high performance computing. Two recent representative damping matrix identification routines are compared in this section. The first, developed by Chen, Ju and Tsuei [3] is discussed in Chapter 1, and is based on a least squares solve. The other is the iterative routine described in Section 2.1, and involving many matrix multiplications.

4.4.1. Programming Methodology

HPF directives were added to the least squares method described in Figure 4-1. No other modifications were made to the code. The benefits of parallelization, if any, would be small for this least squares method. Also, the code's parallelization would not allow a much larger degree-of-freedom problem to be considered even with the additional memory available with multiple processors, since the memory requirements for this algorithm grow like $O(mn^4)$, where n is the order of the system, and m is the number of sample frequencies.

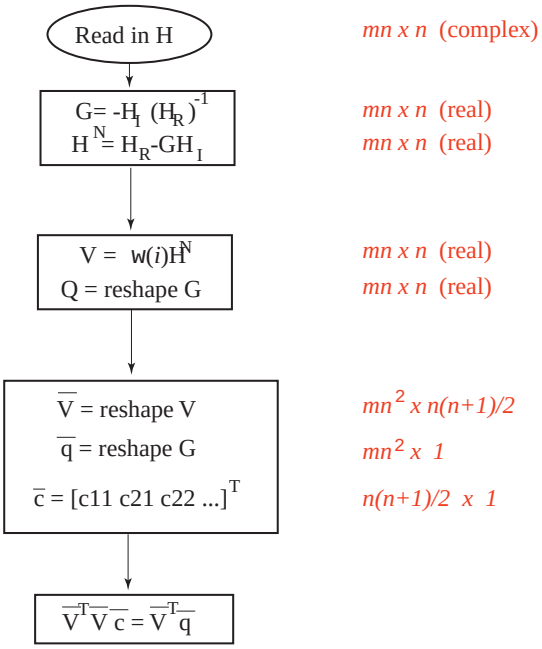


Figure 4-1 Least squares method (Chen, Ju & Tseui, 1997)

4.4.2. Results

Both methods are tested and timed for the problem using the IBM SP-2 at NASA Ames Research Center. The maximum problem size is determined for each method based on the memory of the majority of IBM SP-2 nodes (256 Megabytes). The performance of each method for different problem sizes is examined.

4.4.2.1.Problem Size - Memory Requirements

The nodes used on the IBM SP-2 at NASA Ames Research Center have 256 Megabytes of memory. This is the limiting factor for each method and determines how effectively the machine's memory is utilized. The least squares method use of memory increases at a rather rapid rate ($O(mn^4)$). Because of this, a 35 degree-of-freedom problem fills the machine's memory and larger problems can not be tested without utilizing the memory of multiple processors. However, this will only allow slightly larger problem sizes to be tested because the amount of memory available will grow linearly with the number of processors used, while the amount of memory required will grow on the $O(mn^4)$.

The iterative method is more modest in its use of memory. Problem sizes over 500 degrees of freedom can be accommodated since the memory required will grow on the $O(n^2)$. The slower increase in memory usage allows problems with a high number of degrees of freedom to be run on multiple processors.

4.4.2.2.Timing

Certain conventions are used in timing the results. First, 5 to 10 timings are taken for each test to get an average since there were fluctuations in the timings on the IBM SP-2. Second, only the actual computation is timed; reading the data files is not included in the timings. This is omitted because it is inherently sequential and we have not currently examined parallel I/O. Future work may include parallel I/O since the amount of time to read 500+ degree of freedom data sets can become significant.

4.4.2.3.Performance

The problem is tested for many different degrees of freedom of the type of problem in Figure 5-1 using both methods. Tests conducted on the IBM SP-2 using sequential Fortran 90 are presented in Figure 4-2. The results show that the least squares method is competitive with the iterative method only up to 20 degrees of freedom. After that, the execution time increases at a much higher rate for the least squares method than the iterative method until it no longer fits in memory. The iterative method is capable of solving much larger problems than this least squares method.

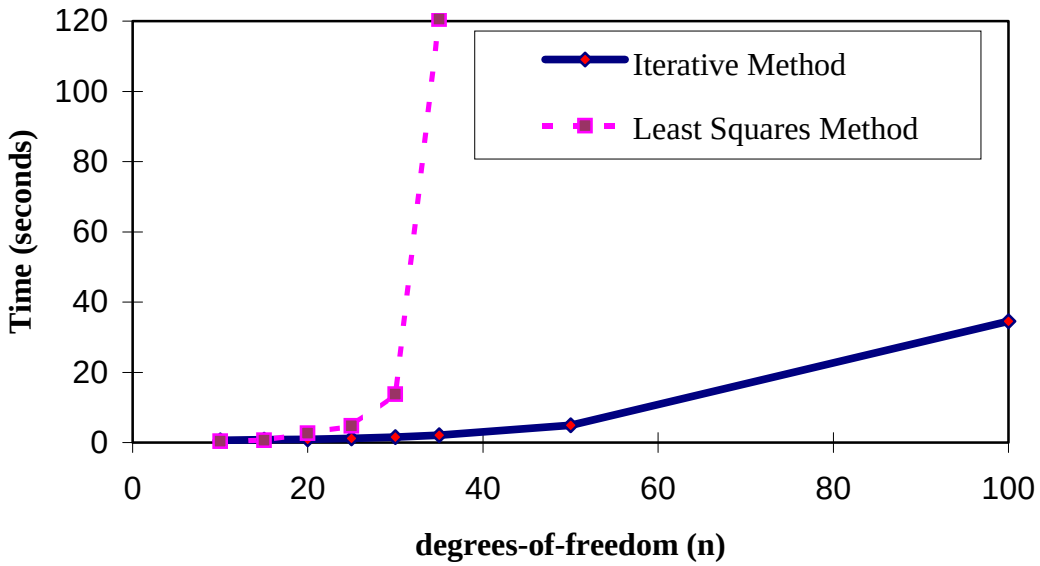


Figure 4-2: Time for one processor to complete multiple degree of freedom problem

4.4.3. Parallel Results on the SP-2

A sequential Fortran 90 code was written for the iterative method. To parallelize it, the sequential code is modified by adding HPF directives. This code is then tested to see which distribution achieves the best performance. The optimal distribution is found to be `(*, block)`. Only this distribution is used for the final simulation.

The algorithm for the iterative method was tested for many different degrees of freedom. Tests conducted on the IBM SP-2 using sequential Fortran 90 and HPF are presented in Table 4-4. Note that parallel speedup refers to the *p* processor HPF time relative to the one processor HPF time. Sequential speedup refers to the *p* processor HPF time compared to the sequential Fortran 90 time. The maximum speedup obtained for: n=100 is 1.70 using 4 processors, n=250 is 3.79 for 8 processors, and n=500 is 6.36 for 16 processors. More extensive parallel results are seen in the following section using the Intel Paragon.

Calculations for accuracy and convergence are made using the Frobenius norm (referred to as F-norm in Table 4-4). The Frobenius norm is defined as

$$|\mathbf{C}|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^2} \ . \qquad (4\text{-}1)$$

With this measure, the results present negligible error for each test.

Table 4-4 Results for the iterative method on the SP-2, using several different problem sizes.

problem size, n = 100				
procs	time(s)	F-norm	parallel speedup	sequential speedup
1	19.53	2.43568	1.00	0.82
2	12.22	2.43568	1.60	1.32
4	9.44	2.43568	2.07	1.70
8	11.26	2.43568	1.73	1.43
problem size, n = 250				
procs	time(s)	F-norm	parallel speedup	sequential speedup
1	286.53	3.86427	1.00	0.85
2	152.36	3.86427	1.88	1.59
4	87.59	3.86427	3.27	2.78
8	64.18	3.86427	4.46	3.79
16	73.47	3.86427	3.90	3.31
problem size, n = 500				
procs	time(s)	F-norm	parallel speedup	sequential speedup
1	2267.43	5.47106	1.00	0.83
2	1134.71	5.47106	2.00	1.68
4	611.52	5.47106	3.71	3.11
8	369.58	5.47106	6.14	5.15
16	298.58	5.47106	7.59	6.36
32	440.14	5.47106	5.14	4.32

4.5. Computational Aspects of Direct and Iterative Damping Identification Methods

One aspect of the damping matrix identification problem that has been neglected in the literature to date is the computational side of the problem. This section takes a close look at the direct and iterative damping matrix identification routines in a computational sense.

As in the previous section, we evaluate an HPF implementation of these algorithms on an Intel Paragon.

The direct method is computationally very similar to one iteration of the iterative damping identification method. Essentially, developing a computationally efficient code involves the following steps:

- Dividing the steps of the algorithm so that each operation (multiplication, addition, transpose, conjugate, etc.) is performed separately.
- Recognizing the independent loops.
- Distributing the data and aligning the matrices properly.

Essentially, the algorithm (see Figure 2-1and Figure 2-2) is divided into the steps seen in Figure 4-3.

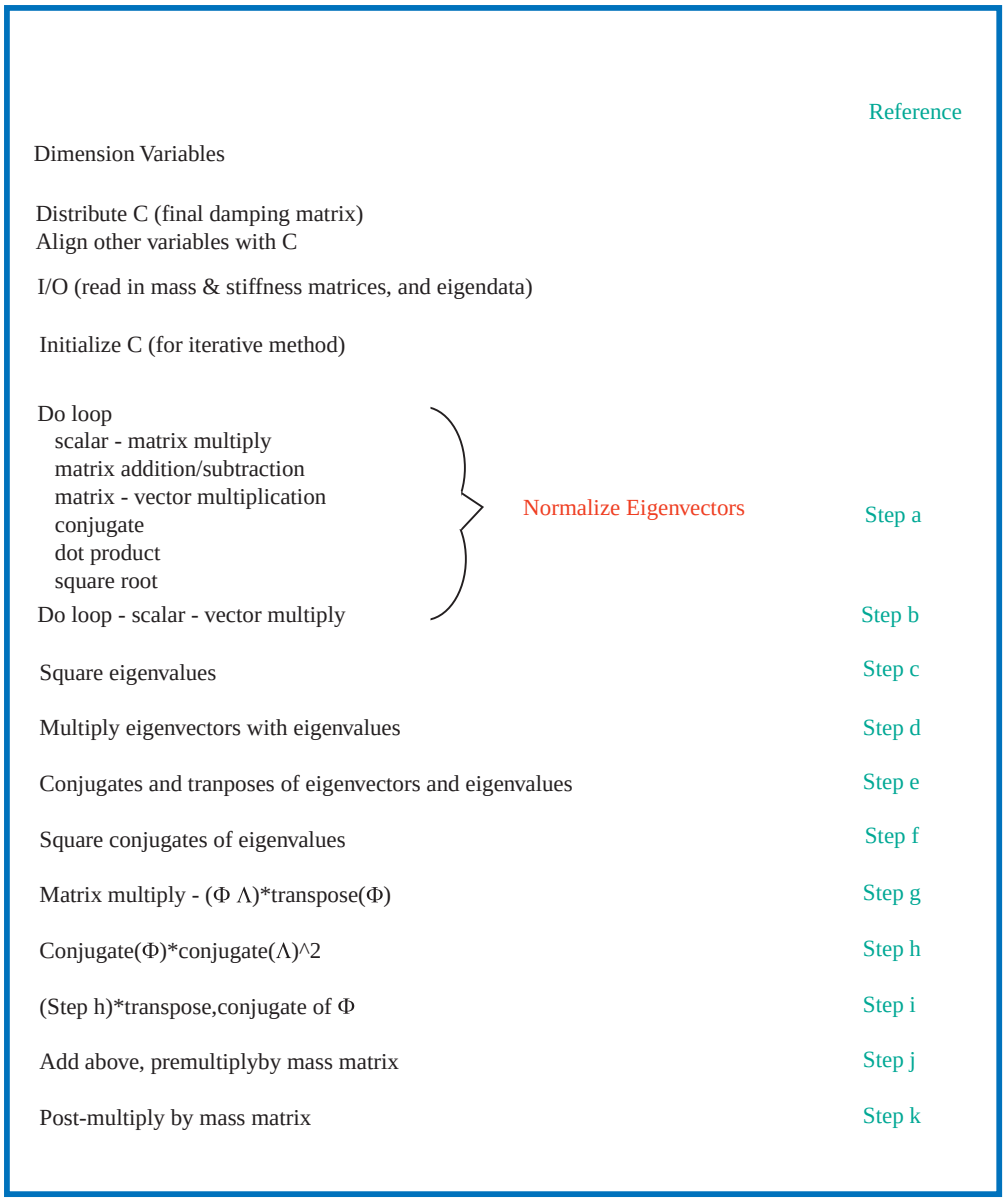


Figure 4-3 Direct and iterative damping matrix identification algorithm

It was found that the code performance is dramatically increased when the steps are divided up, so that each step is a relatively simple operation. Originally, the code was written just as the algorithm in Chapter 2 indicates, with the final step of the damping identification all on one line of code. It was found that the code ran significantly faster if each operation was coded on a separate line. This allows the HPF compiler to generate efficient communicate (i.e. data movement) strategies for each step of the operation.

Several `independent do` loops are used in the code. The `independent` directive can be added to a `do` loop when it is certain that no iteration is dependent another iteration. For example, when each element of a vector must be squared, as in the case of the eigenvectors of the damping identification algorithm, then the work can be divided among several processors. It was found that HPF can override the independent statement when the matrices within the loop are distributed in such a way to make the loop's independence impractical. This was the case for step a of Figure 4-3. The data distribution overrides the independent aspect of the do loop, and it is performed sequentially.

The data is optimally mapped when the final damping matrix **C** is distributed in `(*,block)` form. This means that blocks of columns are assigned to each of the processors. Most of the remaining variables are aligned with the damping matrix. For example, the mass, stiffness and eigenvector matrix are aligned in this way,

```
!HPF$ align ( : , : ) with c( : , : )
```

This means that the rows and columns of these matrices are aligned with the rows and columns of the damping matrix. Essentially, they are distributed in the same manner. The scale factors used in the normalization and the eigenvalues are aligned with the columns of the damping matrix. Several other intermediate matrices are also aligned similarly.

After eliminating all of the bottlenecks through the methods just described, the one final remaining bottleneck turned out to be the HPF intrinsic `matmul`, which is responsible for the matrix multiplications. Since this is an intrinsic in HPF, it is not possible for the user to improve upon its performance, once the matrices have been optimally distributed. The compiler is responsible for such operations. Because of this, and because those in charge of the creation and maintenance of the compiler are unresponsive to this problem, an improved code could be derived by moving a more open environment. One option for algorithms such as those of this work, and in general structural engineering types of problems which tend to deal primarily with matrices, is to choose a more open environment where the user has greater control over (and responsibility for) data distribution and low-level operations such as matrix

multiplication. The standard example of such an environment is Fortran plus MPI [56] or PVM [57].

Table 4-5 through Table 4-11 show the results of this study for a 100 degree of freedom lumped mass model similar to the one in Figure 5-1, as well as 480 and 1012 degree of freedom plate examples. By looking at the timing and speedup numbers, it is evident that as the problem size increases, so does the effectiveness of the parallel features. This is because of the tradeoff between computation and communication. The parallel speedup results are excellent, with a speedup of approximately 30 for both methods on 32 processors for the largest problem, corresponding to a parallel efficiency rate of over 90%. A look at computational rates shows that for the direct method, the 100 degree of freedom problem starts out at 2.87 for one processor, but quickly degrades to 0.06 at 32 processors. The 480 degree of freedom problem remains steady around 1.65 FLOPS through 8 processors, and degrades to 1.14 at 32 processors. The 1012 DOF problem keeps a roughly constant FLOPS rate between 1.43 and 1.51 for all numbers of processors used, except for 32 processors which has a slightly lower FLOPS rate of 1.35. The iterative method shows similar trends.

Table 4-5 Computational results for direct method applied to a 100 DOF lumped mass example.

100 DOF spring mass damper, direct method						
	number of processors					
	1	2	4	8	16	32
ta	1.81E+00	1.06E+00	7.65E-01	7.01E-01	8.15E-01	1.15E+00
tb	1.50E-02	1.70E-02	2.60E-02	3.60E-02	4.50E-02	5.30E-02
tc	1.00E-03	0.00E+00	1.00E-03	1.00E-03	0.00E+00	0.00E+00
td	1.10E-02	2.20E-02	2.80E-02	3.80E-02	4.30E-02	5.20E-02
te	2.60E-02	1.90E-02	1.60E-02	1.80E-02	2.60E-02	3.70E-02
tf	1.00E-03	1.00E-03	0.00E+00	0.00E+00	0.00E+00	1.00E-03
tg	7.54E-01	3.87E-01	2.06E-01	1.37E-01	9.30E-02	1.09E-01
th	1.10E-02	2.10E-02	3.40E-02	3.70E-02	4.40E-02	5.70E-02
ti	7.51E-01	3.77E-01	1.92E-01	1.10E-01	8.40E-02	9.80E-02
tj	7.68E-01	3.88E-01	1.99E-01	1.15E-01	8.60E-02	1.00E-01
tk	7.46E-01	3.77E-01	1.94E-01	1.11E-01	8.40E-02	9.90E-02
total time	4.89E+00	2.66E+00	1.66E+00	1.30E+00	1.32E+00	1.75E+00
speedup	1.00	1.83	2.94	3.75	3.70	2.79
matmul	61.8%	57.4%	47.6%	36.3%	26.3%	23.1%
loop a	36.9%	39.6%	46.1%	53.8%	61.7%	65.5%
other	1.3%	3.0%	6.3%	10.0%	12.0%	11.4%

Table 4-6 Computational results for direct method applied to a 480 DOF plate example

480 DOF plate, direct method						
	number of processors					
	1	2	4	8	16	32
ta	1.76E+02	8.91E+01	4.57E+01	2.44E+01	1.46E+01	1.17E+01
tb	1.56E-01	2.05E-01	2.46E-01	2.87E-01	3.25E-01	3.62E-01
tc	2.00E-03	9.99E-04	9.99E-04	2.00E-03	1.00E-03	1.00E-03
td	2.75E-01	3.52E-01	3.79E-01	4.16E-01	4.56E-01	4.85E-01
te	9.18E-01	6.68E-01	3.61E-01	1.60E-01	9.10E-02	9.40E-02
tf	2.00E-03	2.00E-03	9.99E-04	1.00E-03	1.00E-03	1.00E-03
tg	1.86E+02	9.31E+01	4.71E+01	2.43E+01	1.33E+01	9.05E+00
th	2.69E-01	4.02E-01	4.86E-01	4.39E-01	4.84E-01	5.13E-01
ti	1.86E+02	9.28E+01	4.67E+01	2.36E+01	1.21E+01	6.74E+00
tj	1.97E+02	9.30E+01	4.68E+01	2.37E+01	1.22E+01	6.77E+00
tk	1.93E+02	9.27E+01	4.67E+01	2.36E+01	1.21E+01	6.75E+00
total time	9.40E+02	4.62E+02	2.34E+02	1.21E+02	6.57E+01	4.24E+01
speedup	1.00	2.03	4.01	7.76	14.29	22.14
matmul	81.1%	80.4%	79.9%	78.7%	75.7%	69.1%
loop a	18.8%	19.3%	19.5%	20.2%	22.3%	27.5%
other	0.2%	0.4%	0.6%	1.1%	2.1%	3.4%

Table 4-7 Computational results for direct method applied to a 1012 DOF plate example.

1012 dof plate - direct method						
	number of processors					
	1	2	4	8	16	32
ta	1.71E+03	8.40E+02	4.23E+02	2.16E+02	1.15E+02	6.91E+01
tb	2.30E+01	7.96E-01	8.85E-01	9.67E-01	1.05E+00	1.13E+00
tc	4.80E-02	3.00E-02	2.99E-03	2.99E-03	3.00E-03	3.00E-03
td	1.92E+01	4.58E+00	1.49E+00	1.51E+00	1.58E+00	1.64E+00
te	5.25E+01	2.96E+01	1.84E+00	8.97E-01	4.07E-01	2.29E-01
tf	1.58E-01	4.40E-02	4.00E-03	3.01E-03	3.00E-03	3.00E-03
tg	1.95E+03	1.01E+03	5.71E+02	2.38E+02	1.25E+02	7.42E+01
th	2.92E+01	5.96E+01	3.15E+01	2.07E+00	2.08E+00	2.08E+00
ti	1.91E+03	9.61E+02	4.75E+02	2.34E+02	1.18E+02	6.22E+01
tj	2.01E+03	1.03E+03	5.41E+02	2.47E+02	1.19E+02	6.23E+01
tk	1.96E+03	9.85E+02	5.00E+02	2.35E+02	1.18E+02	6.22E+01
total time	9.67E+03	4.92E+03	2.55E+03	1.18E+03	6.01E+02	3.35E+02
speedup	1.00	1.96	3.80	8.22	16.09	28.84
matmul	81.0%	81.0%	82.0%	81.1%	80.0%	77.9%
loop a	17.7%	17.1%	16.6%	18.4%	19.2%	20.6%
other	1.3%	1.9%	1.4%	0.5%	0.9%	1.5%

Table 4-8 Computational results for iterative method (one iteration) applied to a 100 DOF lumped mass example.

100 DOF lumped mass example, iterative method						
	number of processors					
	1	2	4	8	16	32
ta	1.90E+00	1.07E+00	7.17E-01	6.21E-01	6.89E-01	9.66E-01
tb	8.00E-03	1.70E-02	2.60E-02	3.50E-02	4.30E-02	5.30E-02
tc	0.00E+00	1.00E-03	1.00E-03	0.00E+00	0.00E+00	0.00E+00
td	7.00E-03	1.60E-02	2.50E-02	3.30E-02	4.30E-02	5.00E-02
te	1.10E-02	7.00E-03	6.00E-03	6.00E-03	8.00E-03	1.10E-02
tf	0.00E+00	0.00E+00	1.00E-03	0.00E+00	1.00E-03	0.00E+00
tg	7.47E-01	3.77E-01	1.94E-01	1.15E-01	8.70E-02	1.03E-01
th	8.00E-03	1.70E-02	2.70E-02	3.40E-02	4.30E-02	5.50E-02
ti	7.47E-01	3.77E-01	1.92E-01	1.11E-01	8.30E-02	9.80E-02
tj	7.56E-01	3.81E-01	1.94E-01	1.12E-01	8.40E-02	1.00E-01
tk	7.46E-01	3.76E-01	1.93E-01	1.12E-01	8.40E-02	9.90E-02
total time	4.93E+00	2.64E+00	1.58E+00	1.18E+00	1.16E+00	1.54E+00
speedup	1.00	1.87	3.13	4.18	4.23	3.21
matmul	60.8%	57.2%	49.0%	38.2%	29.0%	26.1%
loop a	38.6%	40.6%	45.5%	52.7%	59.1%	62.9%
other	0.7%	2.2%	5.5%	9.2%	11.8%	11.0%

Table 4-9 Computational results for iterative method (one iteration) applied to a 480 DOF plate example

480 DOF plate, iterative method						
	number of processors					
	1	2	4	8	16	32
ta	2.35E+02	9.55E+01	4.88E+01	2.59E+01	1.54E+01	1.19E+01
tb	1.71E-01	2.05E-01	2.46E-01	2.85E-01	3.24E-01	3.63E-01
tc	2.08E-03	2.01E-03	1.98E-03	2.00E-03	1.01E-03	2.00E-03
td	6.29E+00	1.91E-01	2.34E-01	2.75E-01	3.18E-01	3.56E-01
te	1.57E+01	2.31E-01	1.26E-01	4.90E-02	3.20E-02	3.30E-02
tf	9.77E-04	1.95E-03	1.98E-03	2.00E-03	9.99E-04	2.00E-03
tg	2.06E+02	9.36E+01	4.65E+01	2.36E+01	1.21E+01	6.76E+00
th	5.96E+00	1.93E-01	2.80E-01	3.11E-01	3.50E-01	3.86E-01
ti	1.91E+02	9.36E+01	4.65E+01	2.36E+01	1.21E+01	6.75E+00
tj	2.02E+02	9.36E+01	4.65E+01	2.37E+01	1.21E+01	6.76E+00
tk	1.91E+02	9.36E+01	4.65E+01	2.36E+01	1.21E+01	6.75E+00
total time	1.05E+03	4.71E+02	2.36E+02	1.21E+02	6.49E+01	4.01E+01
speedup	1.00	2.24	4.47	8.67	16.24	26.29
matmul	75.0%	79.5%	78.9%	77.9%	74.7%	67.4%
loop a	22.3%	20.3%	20.7%	21.4%	23.7%	29.7%
other	2.7%	0.2%	0.4%	0.8%	1.6%	2.8%

Table 4-10 Computational results for iterative method (one iteration) applied to a 1012 DOF plate example

1012 DOF plate, iterative method						
	number of processors					
	1	2	4	8	16	32
ta	1.87E+03	9.44E+02	5.01E+02	2.31E+02	1.23E+02	7.29E+01
tb	2.25E+01	9.03E-01	9.93E-01	9.68E-01	1.05E+00	1.13E+00
tc	2.73E-02	2.49E-02	2.93E-03	1.95E-03	2.99E-03	2.99E-03
td	2.56E+01	2.39E+01	2.74E+01	9.12E-01	9.95E-01	1.08E+00
te	6.77E+01	4.25E+01	3.63E+01	3.29E-01	1.42E-01	8.40E-02
tf	1.30E-01	5.52E-02	2.93E-03	3.05E-03	2.99E-03	2.99E-03
tg	1.99E+03	1.06E+03	5.84E+02	2.34E+02	1.19E+02	6.22E+01
th	5.75E+01	8.74E+01	6.60E+01	1.14E+00	1.09E+00	1.19E+00
ti	1.93E+03	9.75E+02	4.84E+02	2.34E+02	1.19E+02	6.22E+01
tj	2.13E+03	1.09E+03	5.74E+02	2.34E+02	1.19E+02	6.23E+01
tk	1.91E+03	9.69E+02	4.99E+02	2.34E+02	1.19E+02	6.22E+01
total time	1.00E+04	5.19E+03	2.77E+03	1.17E+03	6.00E+02	3.25E+02
speedup	1.00	1.93	3.61	8.55	16.67	30.75
matmul	79.6%	78.8%	77.2%	79.9%	79.0%	76.5%
loop a	18.7%	18.2%	18.1%	19.8%	20.4%	22.4%
other	1.7%	3.0%	4.7%	0.3%	0.5%	1.1%

Table 4-11 Computational rate (MFLOPS per processor) for the three test problems using direct and iterative damping identification.

	Direct Method			Iterative Method		
processors	100 DOF	480 DOF	1012 DOF	100 DOF	480 DOF	1012 DOF
1	2.87	1.65	1.50	2.85	1.47	1.45
2	2.63	1.68	1.47	2.65	1.64	1.39
4	1.72	1.65	1.43	2.22	1.64	1.29
8	0.82	1.60	1.54	1.48	1.59	1.55
16	0.26	1.47	1.51	0.75	1.49	1.51
32	0.06	1.14	1.35	0.28	1.21	1.39

Chapter 5

Examples

5.1. Introduction

The objective of this chapter is to provide a better understanding of the issues discussed in the previous chapters through illustrative examples. Two examples are presented. The first is a simple lumped mass-spring-damper system that is used to illustrate the steps of the iterative damping matrix identification routine. It is a good example to show the potential for damping matrix identification in the area of diagnostics of structures. The second example is a larger, more complex plate problem. It is used to illustrate the performance of both the direct and iterative methods with modal incompleteness, spatial incompleteness, and noise in the system.

5.2. Lumped Mass System Example

Consider the multi-degree-of-freedom lumped mass system shown in Figure 5-1.

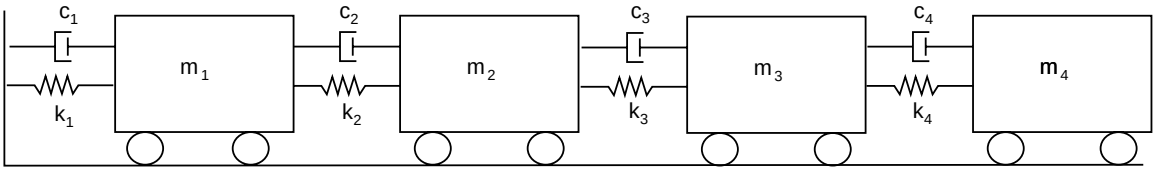


Figure 5-1 Lumped mass system

The purpose of this example is to understand and visualize the steps involved in the iterative damping matrix identification routine. The mass and stiffness matrices for this simple system are provided. The desired damping matrix is also provided to create a

better understanding of the process. Because this is a contrived example, the desired result is known ahead of time. This model is described by:

$$\mathbf{M} = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0.02 & -0.01 & 0.00 & 0.00 \\ -0.01 & 0.02 & -0.01 & 0.00 \\ 0.00 & -0.01 & 0.02 & -0.01 \\ 0.00 & 0.00 & -0.01 & 0.01 \end{bmatrix}$$

For this example, we assume the mass of the system is known. The eigenvalues and eigenvectors can be obtained experimentally, although in this case we use exact values obtained analytically. In this method, the eigenvalues and eigenvectors are assumed to occur in complex conjugate pairs. Since only one root from each pair is needed, we are left with four eigenvalues and eigenvectors,

$$\Lambda = diag(-0.0000 + 0.1256i, -0.0001 + 0.3864i, -0.0002 + 0.5922i, -0.0002 + 0.6959i)$$

$$\Phi = \begin{bmatrix} 0.0100 - 0.2290i & 0.1030 - 0.4144i & 0.0259 - 0.3431i & -0.0042 + 0.7385i \\ 0.0192 - 0.4399i & 0.1292 - 0.5194i & 0.0064 - 0.0846i & 0.0018 - 0.3110i \\ 0.0253 - 0.5815i & -0.0376 + 0.1511i & -0.0356 + 0.4705i & -0.0008 + 0.1454i \\ 0.0275 - 0.6313i & -0.1482 + 0.5960i & 0.0472 - 0.6245i & 0.0006 - 0.1023i \end{bmatrix}.$$

This forms the data to be used in the construction of a damping matrix.

5.3. Damping Identification of lumped mass system

The first step in the iterative damping matrix identification process (see Figure 2-1) is to choose an initial value for the damping matrix, C. In this case the identity matrix is the initial guess.

$$\mathbf{C}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Next, the eigenvectors are normalized according to equation (1-37). A first damping matrix is generated. After just one iteration the damping matrix and eigenvectors are,

$$\mathbf{C} = \begin{bmatrix} 0.3994 & -0.0495 & -0.0068 & -0.0112 \\ -0.0495 & 0.6930 & -0.0855 & -0.0292 \\ -0.0068 & -0.0855 & 0.6483 & -0.0632 \\ -0.0112 & -0.0292 & -0.0632 & 0.3387 \end{bmatrix}$$

$$\Phi = \begin{bmatrix} 0.1337 - 0.0814i & 0.1526 - 0.1263i & 0.1073 - 0.0944i & -0.2347 + 0.2078i \\ 0.2568 - 0.1564i & 0.1913 - 0.1583i & 0.0265 - 0.0233i & 0.0988 - 0.0875i \\ 0.3394 - 0.2067i & -0.0556 + 0.0460i & -0.1472 + 0.1295i & -0.0462 + 0.0409i \\ 0.3685 - 0.2244i & -0.2195 + 0.1817i & 0.1954 - 0.1719i & 0.0325 - 0.0288i \end{bmatrix}.$$

Using the new damping matrix, the eigenvectors are again normalized, and the process repeats in this manner. After five iterations,

$$\mathbf{C} = \begin{bmatrix} 0.0267 & -0.0042 & -0.0005 & -0.0007 \\ -0.0042 & 0.0445 & -0.0067 & -0.0020 \\ -0.0005 & -0.0067 & 0.0415 & -0.0052 \\ -0.0007 & -0.0020 & -0.0052 & 0.0218 \end{bmatrix}$$

$$\Phi = \begin{bmatrix} 0.1188 - 0.1160i & 0.1421 - 0.1405i & 0.1019 - 0.1011i & -0.2233 + 0.2217i \\ 0.2282 - 0.2228i & 0.1782 - 0.1762i & 0.0251 - 0.0249i & 0.0940 - 0.0933i \\ 0.3017 - 0.2945i & -0.0518 + 0.0512i & -0.1397 + 0.1386i & -0.0440 + 0.0436i \\ 0.3275 - 0.3197i & -0.2044 + 0.2021i & 0.1855 - 0.1840i & 0.0309 - 0.0307i \end{bmatrix}.$$

It should be noted that using the same methodology, the stiffness matrix could be found simultaneously. After fifteen iterations the damping matrix is

$$\mathbf{C} = \begin{bmatrix} 0.0200 & -0.0100 & 0.0000 & 0.0000 \\ -0.0100 & 0.0200 & -0.0100 & 0.0000 \\ 0.0000 & -0.0100 & 0.0200 & -0.0100 \\ 0.0000 & 0.0000 & -0.0100 & 0.0100 \end{bmatrix}$$

$$\Phi = \begin{bmatrix} 0.1174 - 0.1174i & 0.1413 - 0.1413i & 0.1015 - 0.1015i & -0.2225 + 0.2225i \\ 0.2256 - 0.2256i & 0.1772 - 0.1772i & 0.0250 - 0.0250i & 0.0937 - 0.0937i \\ 0.2981 - 0.2981i & -0.0515 + 0.0515i & -0.1392 + 0.1392i & -0.0438 + 0.0438i \\ 0.3237 - 0.3237i & -0.2033 + 0.2033i & 0.1847 - 0.1847i & 0.0308 - 0.0308i \end{bmatrix}.$$

This is the expected solution to the problem. As more iterations are performed, the values tend toward the exact solution with higher precision. Convergence to the desired solution has been achieved.

Next we show that by assuming knowledge of the structure (Figure 5-1), we can use the iterative method above as a diagnostic tool. In a case such as the above example where the model is known, the damping values at specific locations of the structure can be determined. The example presented will have a damping matrix of the form:

$$\mathbf{C} = \begin{bmatrix} c_1 + c_2 & -c_2 & 0 & 0 \\ -c_2 & c_2 + c_3 & -c_3 & 0 \\ 0 & -c_3 & c_3 + c_4 & -c_4 \\ 0 & 0 & -c_4 & c_4 \end{bmatrix} \tag{5-1}$$

It becomes evident that the values for c_i are 0.0100 for all *i*. This is a valuable tool in diagnostics of structures. If a real system were involved, the test would be run again at a

later time to determine any difference in the resulting damping matrix. Since the location of excess energy dissipation can be determined by comparing damping matrices, the damage to the structure can be identified.

For example, the mass matrix is known to remain constant, and at a later point in time, the measurements taken for the same system shown in Figure 5-1 are different than those above. In this case, the data looks like:

$$\Lambda = diag(-0.0001 + 0.1256i, -0.0009 + 0.3864i, -0.0019 + 0.5922i, -0.0031 + 0.6959i)$$

$$\Phi = \begin{bmatrix} -0.0919 + 0.2100i & 0.3532 - 0.2398i & -0.3297 + 0.0984i & 0.0877 - 0.7333i \\ -0.1767 + 0.4033i & 0.4420 - 0.3018i & -0.0818 + 0.0225i & -0.0353 + 0.3090i \\ -0.2337 + 0.5331i & -0.1228 + 0.0874i & 0.4529 - 0.1323i & 0.0157 - 0.1446i \\ -0.2537 + 0.5787i & -0.5074 + 0.3459i & -0.6010 + 0.1761i & -0.0107 + 0.1018i \end{bmatrix}.$$

Using the known mass matrix, along with this measurement data, the iterative routine is once again used to determine the damping matrix.

$$\mathbf{C} = \begin{bmatrix} 0.0300 & -0.0100 & 0.0000 & 0.0000 \\ -0.0100 & 0.0200 & -0.0100 & 0.0000 \\ 0.0000 & -0.0100 & 0.0200 & -0.0100 \\ 0.0000 & 0.0000 & -0.0100 & 0.0100 \end{bmatrix}$$

It is easily determined that the over time, c_1 has doubled to 0.02. This illustrates the possibilities for damage detection in systems with all known measurable data. The above is also an example of the successful use of the iterative damping matrix identification procedure for non-normal mode damping.

5.4. Plate Example

To illustrate the damping identification routines discussed in this work, a finite element example has been contrived. For the purpose of illustration an analytical model of a plate is given non-proportional damping. Using only the eigensystem and the mass matrix, or the mass and stiffness matrices, a damping matrix is identified for the plate. Results are presented with the assumption of knowledge of the full eigensystem, and then it is

assumed that there is a deficit in either the modes or the degrees of freedom measured. Then, the effectiveness of the algorithms is illustrated when it is assumed that the deficit in the eigensystem is affected by both the modes and degrees of freedom measured. Finally, noise is added to the system to illustrate the robustness of the algorithms.

A finite element model is formulated using four noded quadrilateral elements. Quadratic shape functions are used. The consistent mass matrix and stiffness matrix are assembled in a way that allows for two degrees of freedom at each node. The global matrices are assembled with the use of a destination array. This array makes use of the homogeneous essential boundary conditions to distinguish between the active and passive degrees of freedom. The plate has 40 degrees of freedom for this example.

Non-proportional damping is added to the plate by first creating proportional damping at each iteration of the assembly of the global matrices, except in the fifth element, where the contribution of the stiffness to the damping matrix is reduced by fifty percent. It is easily verified that non-Rayleigh style damping ensues with the simple equation [58]

$$\mathbf{C}\mathbf{M}^{-1}\mathbf{K} \neq \mathbf{K}\mathbf{M}^{-1}\mathbf{C}. \tag{5-2}$$

A mesh of the expected damping matrix is shown below.

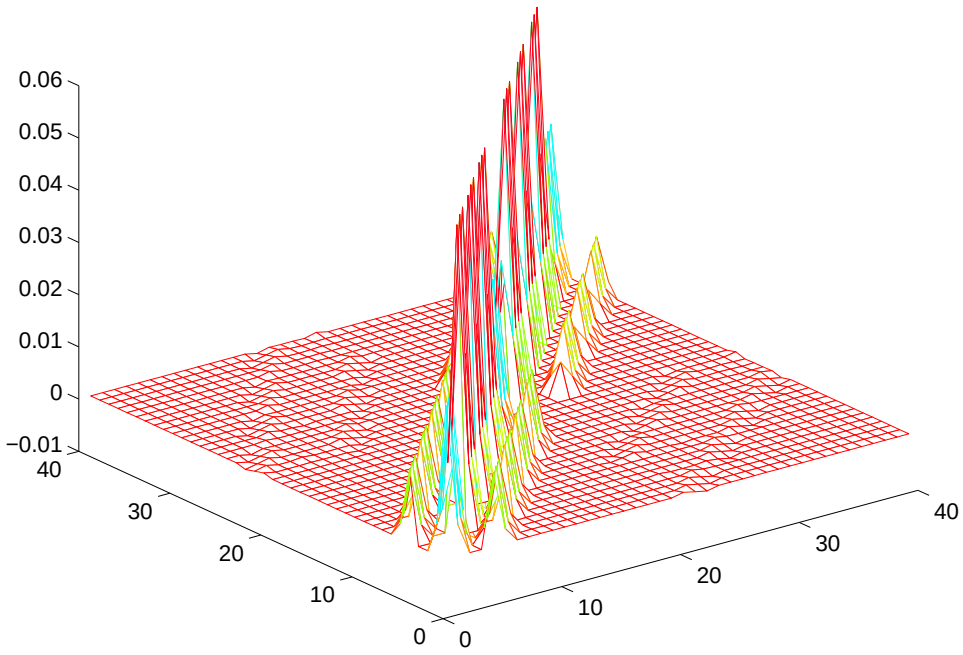


Figure 5-2 Mesh of damping matrix

5.5. Obtaining Results

5.5.1. Data Generation

Two methods are used to generate the data for the following 'test' situations. The first method, shown in Figure 5-3a, begins with the full system matrices, including the damping, which is the ultimate goal of the identification procedure. The eigenvalues and eigenvectors are generated from this full system. Next, a specified number of columns for the higher modes are removed from the matrix of eigenvectors to simulate modal incompleteness. Then, rows are removed from the already reduced matrix of eigenvectors to simulate spatial incompleteness. These rows, representing degrees of freedom of the plate, can be eliminated in an optimal fashion by making a good choice of master degrees of freedom as discussed in Chapter 3. Finally, the system matrices are reduced using the Iterated IRS technique of model reduction, and the damping matrix is

identified using either the direct or iterative methods of this work. This method will be denoted as method (a) throughout this chapter.

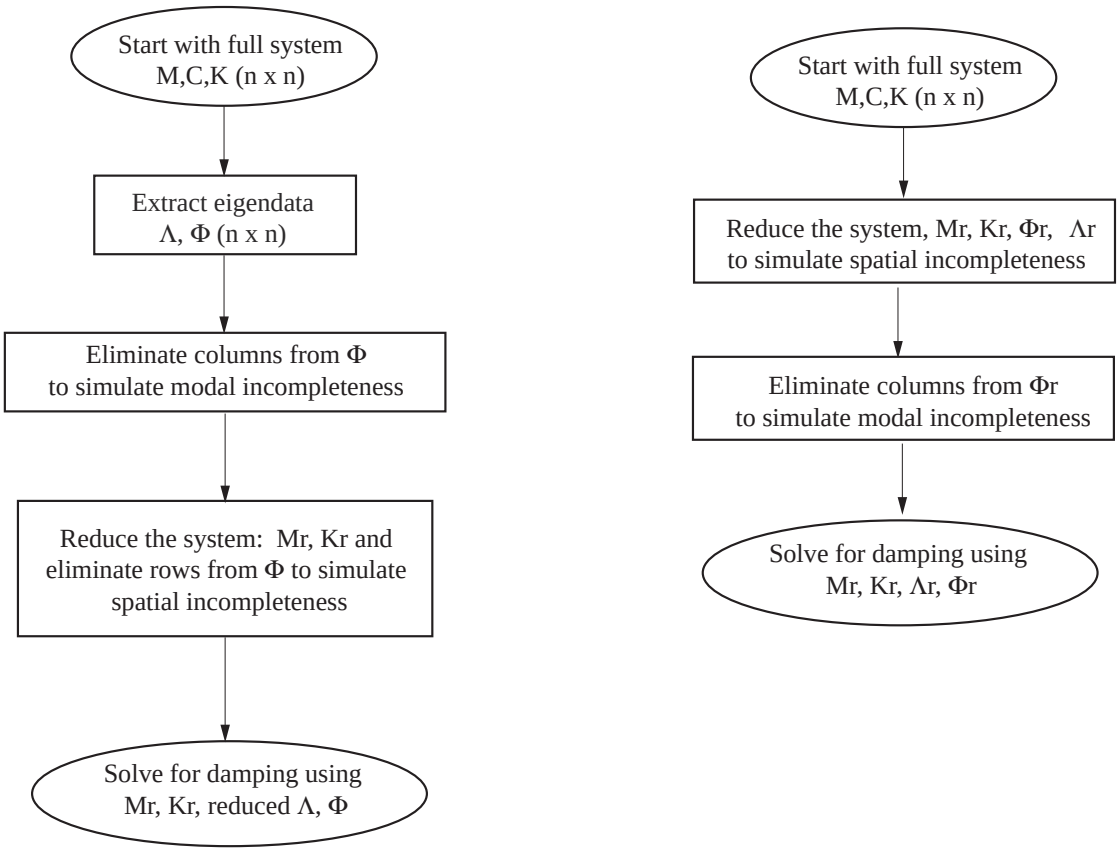


Figure 5-3 (a) Solution method where the eigendata is obtained before the model reduction is performed. (b) Solution method where the reduction is performed before the eigendata is generated.

The second solution method is shown in Figure 5-3b, and will be denoted as method (b) throughout this chapter. Once again, the full system matrices of the plate are generated as discussed above using a finite element model. In this case, the next step involves choosing the master degrees of freedom, and reducing the system matrices using the Iterated IRS technique. Then, a reduced set of eigenvectors is generated using the reduced system matrices. From this new Φr, columns are eliminated to simulate modal incompleteness. Finally, the damping matrix is identified using all of the reduced information via the direct or iterative methods.

5.5.2. Solution Methods

Both the direct and iterative methods presented in Chapter 2 are used here to illustrate the example. It is possible to impose sparsity constraints when using the iterative method. This can be done at every iteration when normalizing the eigenvectors by imposing the condition on the damping matrix.

5.6. Results of plate example

Mesh plots, showing the entries of the damping matrix graphically, are used to compare the predicted damping matrices with the expected results. This type of comparison can only be made for systems that are spatially complete. Figure 5-4 - Figure 5-11 show the mesh of the predicted damping matrix using the iterative damping identification method, and the difference between the predicted and expected damping matrices. The results are excellent when all 40 modes are included in the estimation. The resulting damping matrix is still acceptable when a small percentage of these modes are removed. When fifty percent (or more) modes are removed, the resulting damping matrix is much further from the expected value. Figure 5-28 - Figure 5-35 show similar results for the direct method of damping matrix identification.

FRF plots are used to evaluate the effectiveness of the algorithms when spatial incompleteness is a factor. Figure 5-12 - Figure 5-17 show plots with 30, 20 and 10 degrees of freedom (where these master degrees of freedom were chosen using the method described in Chapter 3), with various number of modes available. Using method (a) and the Iterated IRS method of reduction, it can be seen that as the number of degrees of freedom is reduced, the FRF plots diverge from the expected ones. This is due, in part to the reduction procedure. Similar results are seen for the direct method in Figure 5-36 - Figure 5-41 Method (b) is shown with only 10 modes available (Figure 5-18 and Figure 5-43). Again, the FRF's for large spatial incompleteness have large error when compared to the expected FRF.

Figures 5-29 to 5-27 and 5-44 to 5-51 show the results of the damping matrices for spatially complete systems with five percent normally distributed noise added to the eigenvectors. The results look good when a small percentage of the modes are removed.

5.6.1. Plots - Iterative Method

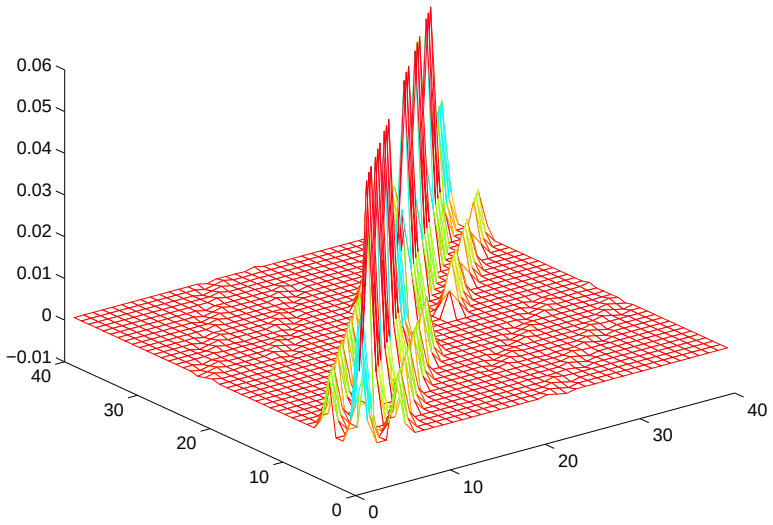


Figure 5-4 Plot of damping matrix found using the iterative method with 40 DOF and 40 modes.

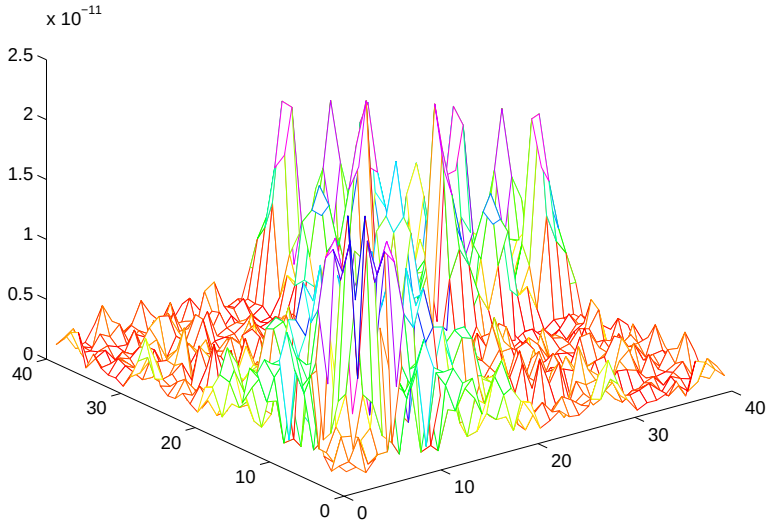


Figure 5-5 Difference between Figure 5-4 and the target damping matrix.

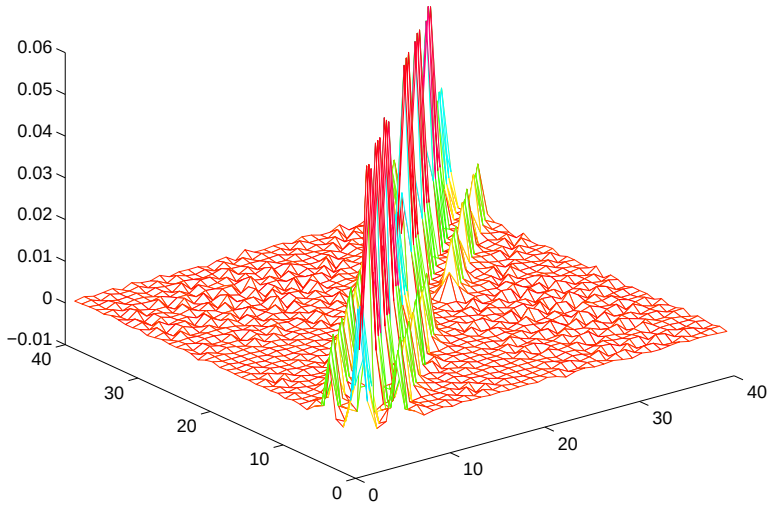


Figure 5-6 Plot of damping matrix found using the iterative method with 40 DOF 10 percent fewer modes

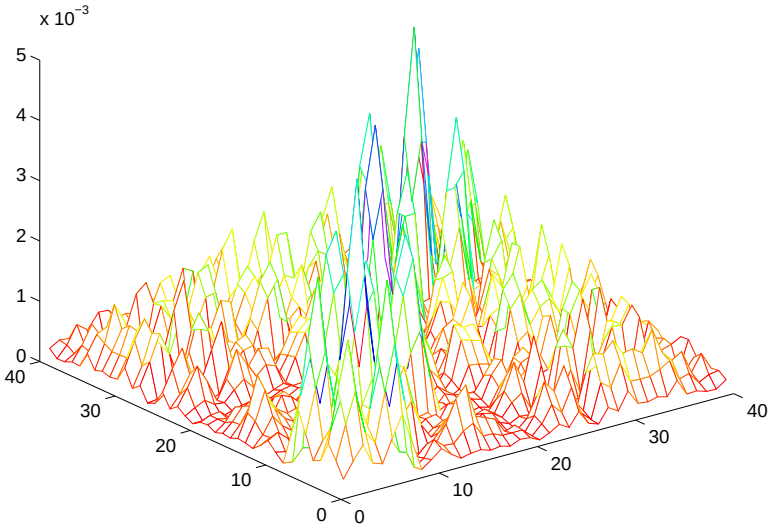


Figure 5-7 Difference between Figure 5-6 and the target damping matrix.

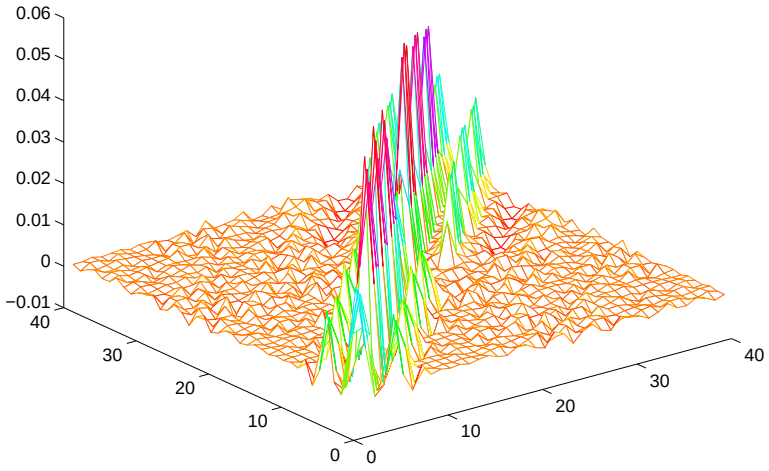


Figure 5-8 Plot of damping matrix found using the iterative method with 40 DOF 33 percent fewer modes.

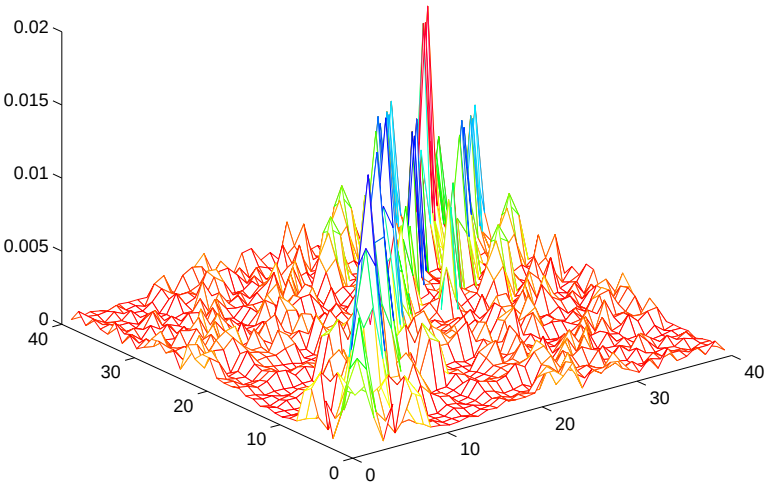


Figure 5-9 Difference between Figure 5-8 and the target damping matrix.

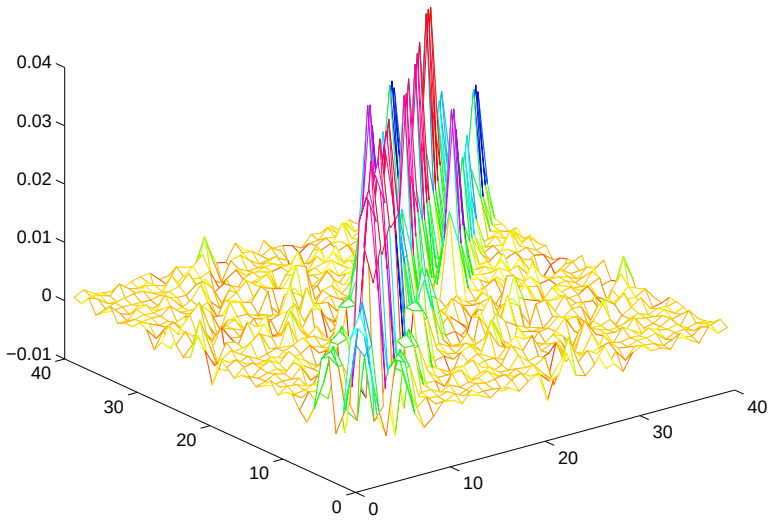


Figure 5-10 Plot of damping matrix found using the iterative method with 40 DOF and 50 percent fewer modes.

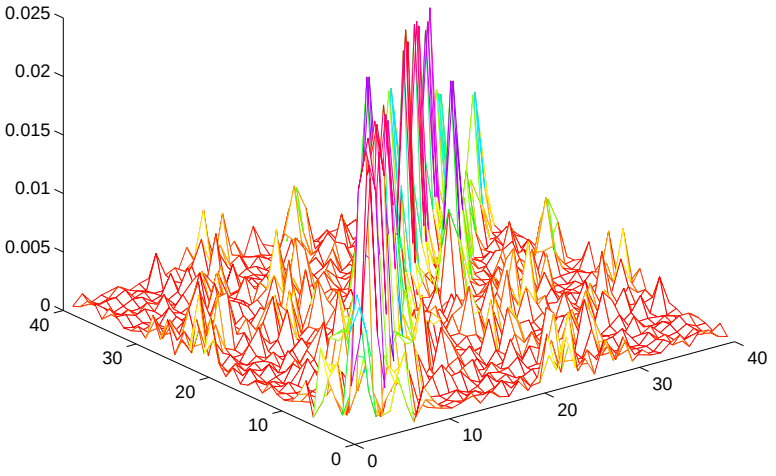


Figure 5-11 Difference between Figure 5-10 and the target damping matrix.

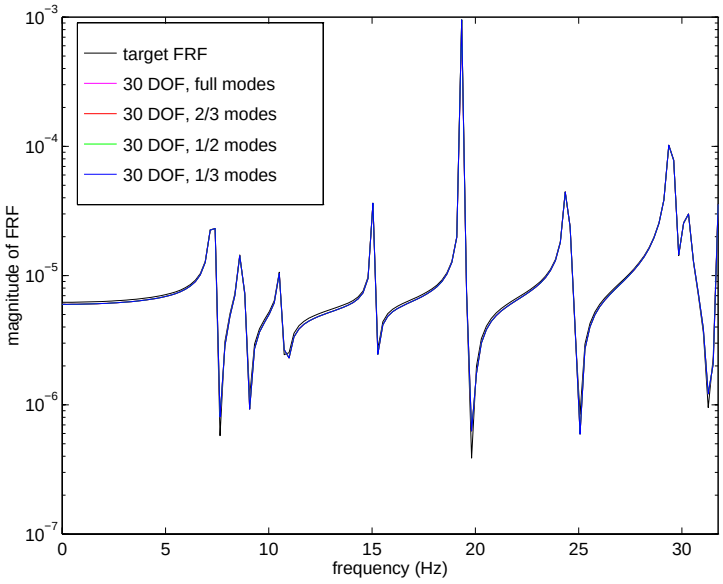


Figure 5-12 FRF plots of plate using iterative damping ID, method a, and only 30 DOF.

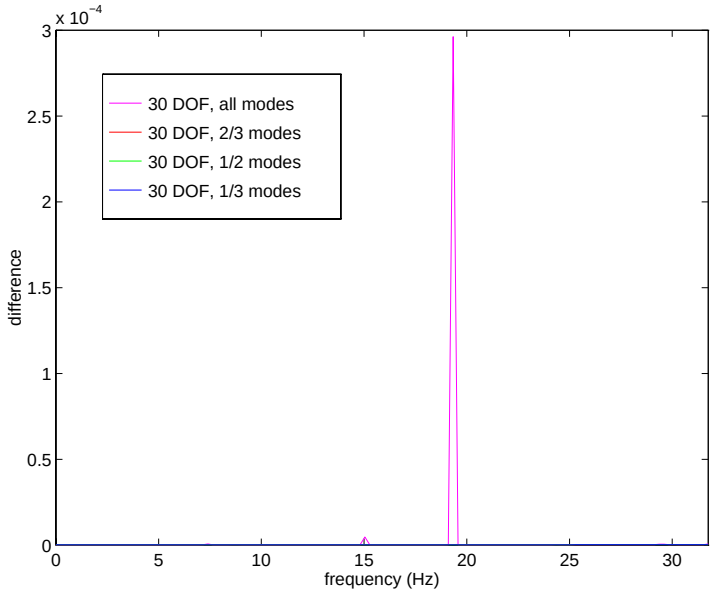


Figure 5-13 Difference between the FRF's of Figure 5-12 and the expected FRF.

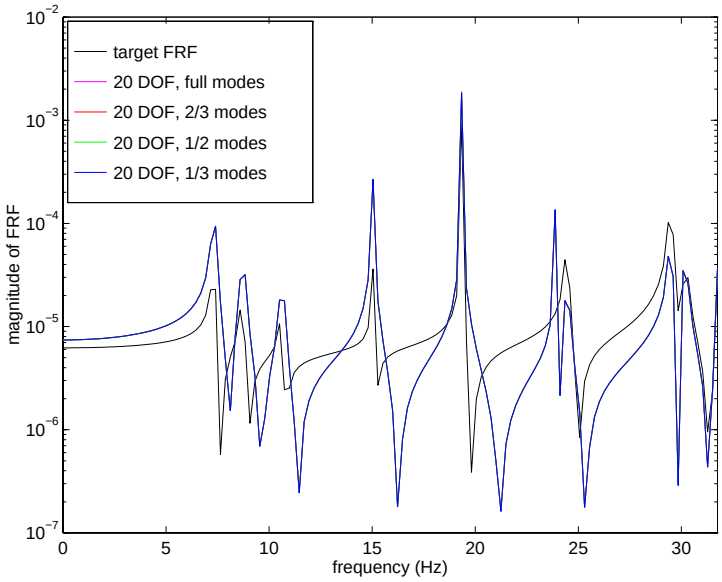


Figure 5-14 FRF plots of plate using iterative damping ID, method a, and only 20 DOF.

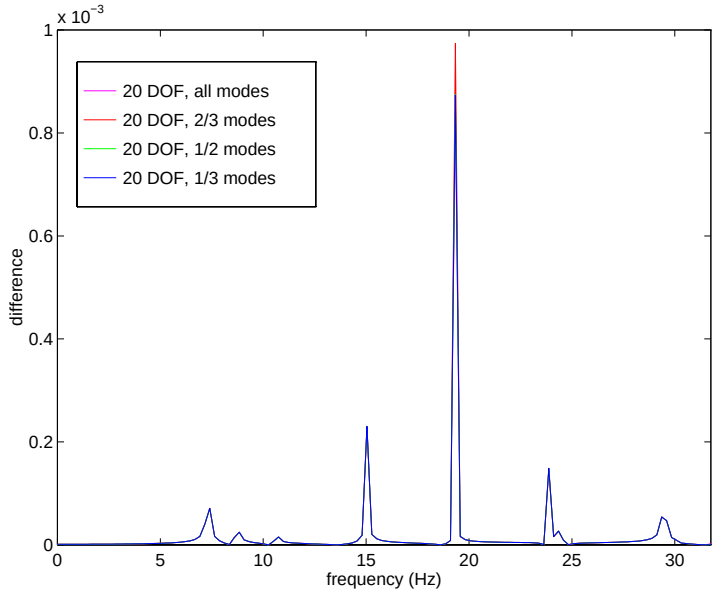


Figure 5-15 Difference between the FRF's of Figure 5-14 and the expected FRF.

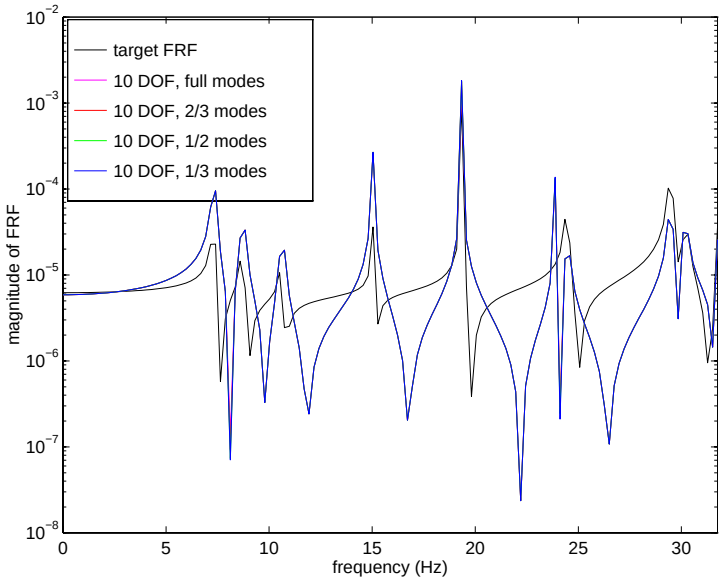


Figure 5-16 FRF plots of plate using iterative damping ID, method a, and only 10 DOF.

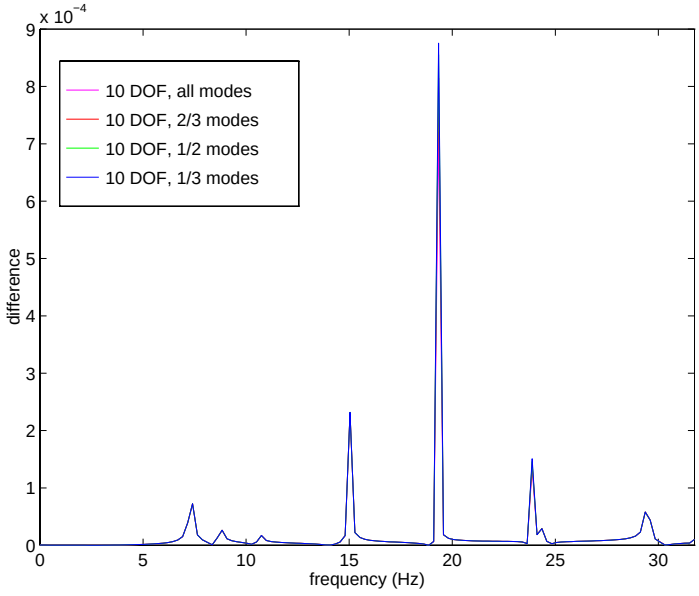


Figure 5-17 Difference between the FRF's of Figure 5-16 and the expected FRF.

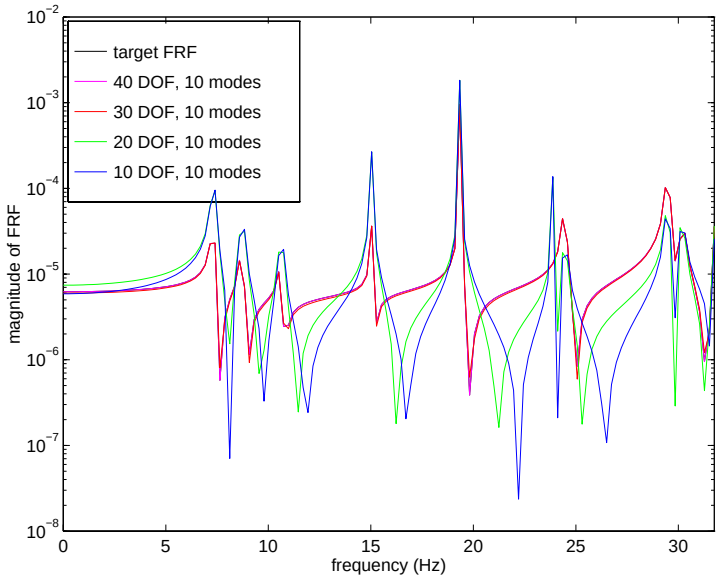


Figure 5-18 FRF plots of plate using iterative damping ID, method b, and only 10 modes.

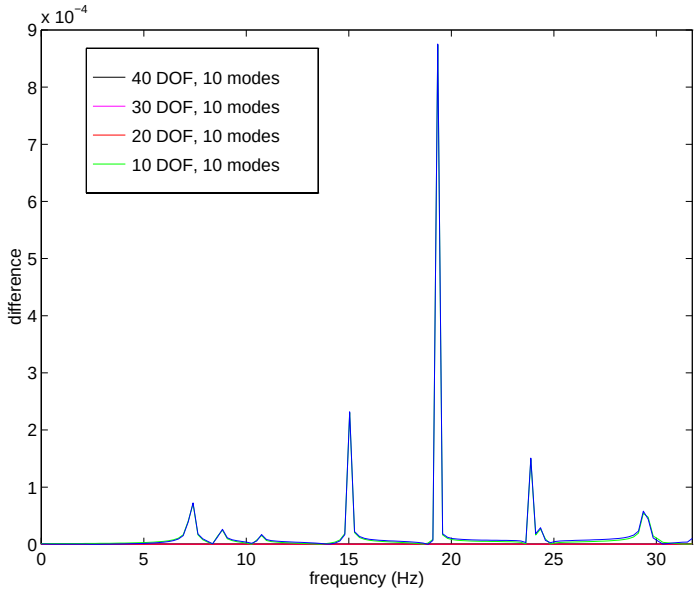


Figure 5-19 Difference between the FRF's of Figure 5-18 and the expected FRF.

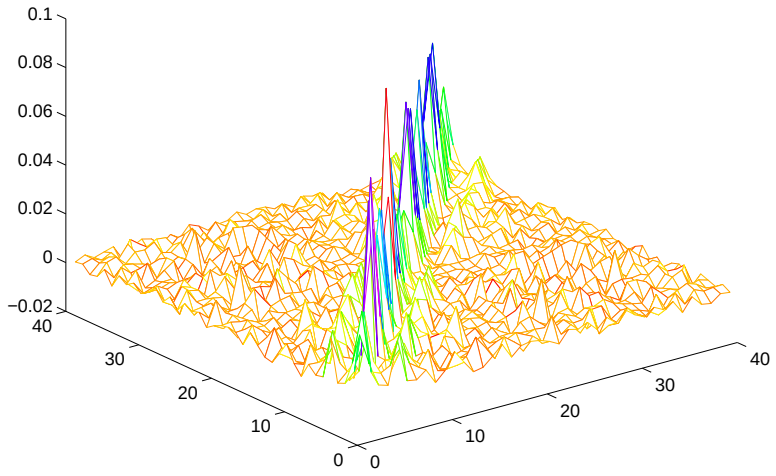


Figure 5-20 Mesh of damping matrix found using iterative method of identification with noise added to the system.

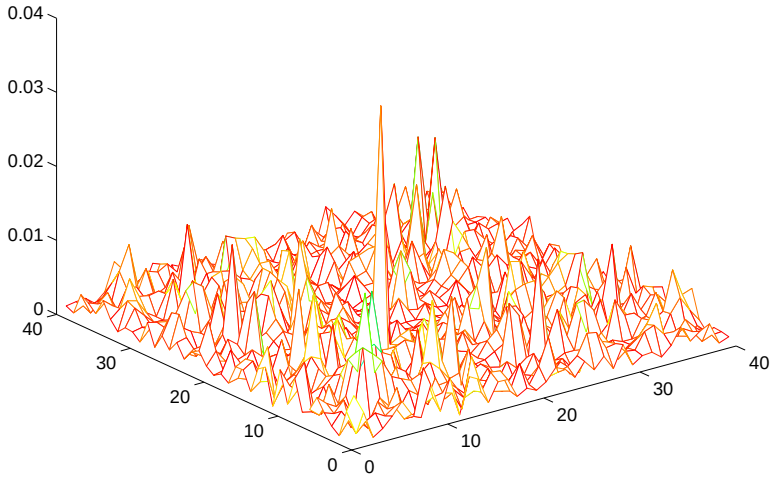


Figure 5-21 Difference between the above plot and the actual damping matrix.

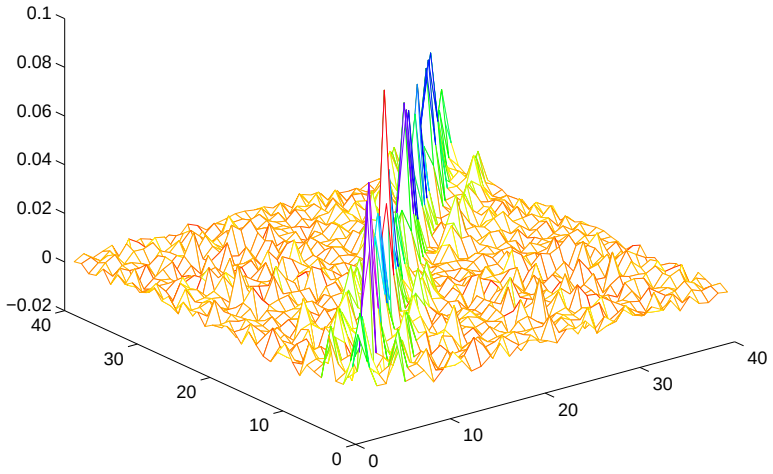


Figure 5-22 Mesh of damping matrix found using iterative method of identification with 10 percent fewer modes and noise added to the system.

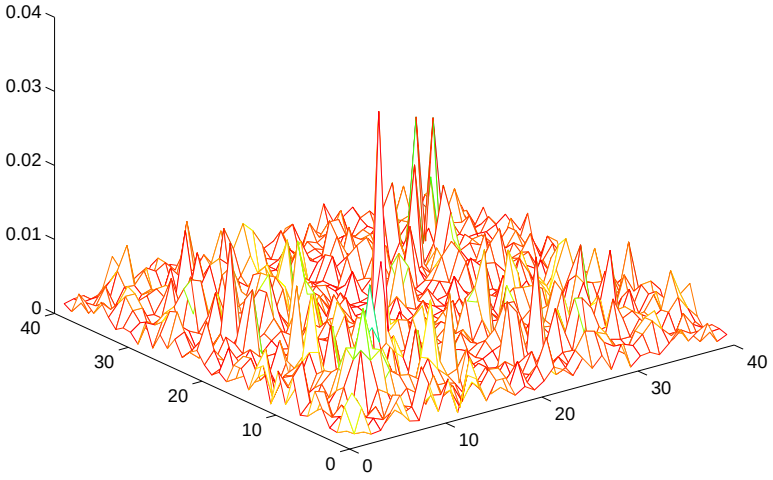


Figure 5-23 Difference between the above plot and the actual damping matrix.

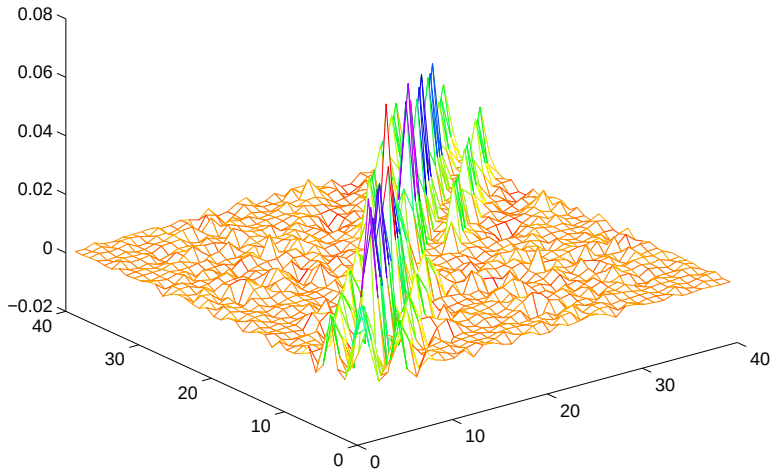


Figure 5-24 Mesh of damping matrix found using iterative method of identification with 33 percent fewer modes and noise added to the system.

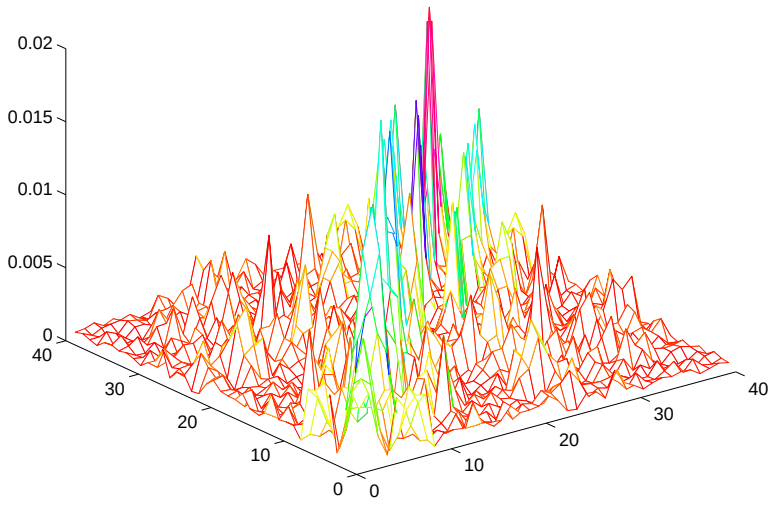


Figure 5-25 Difference between the above plot and the actual damping matrix.

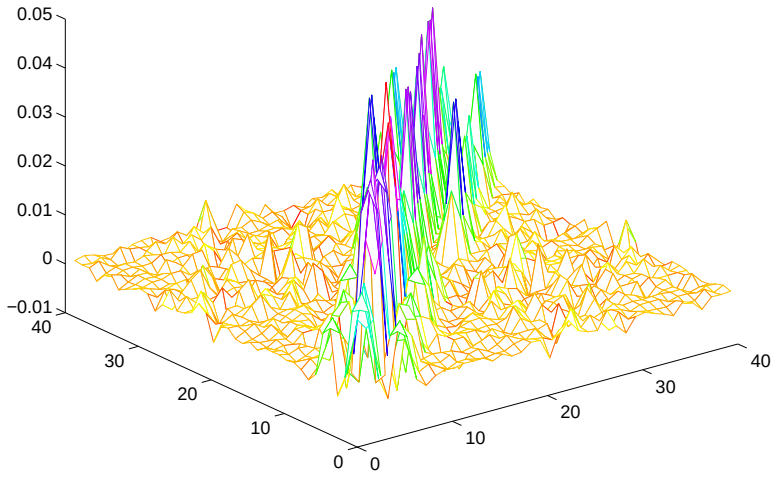


Figure 5-26 Mesh of damping matrix found using iterative method of identification with 50 percent fewer modes and noise added to the system.

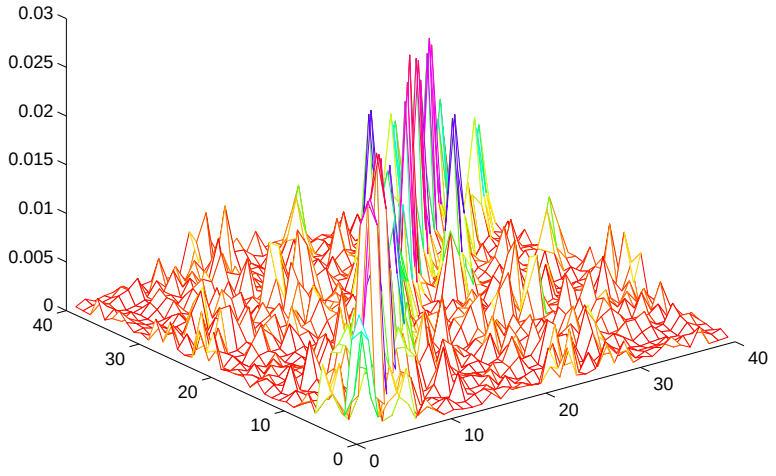


Figure 5-27 Difference between the above plot and the actual damping matrix.

5.6.2. Plots - Direct Method

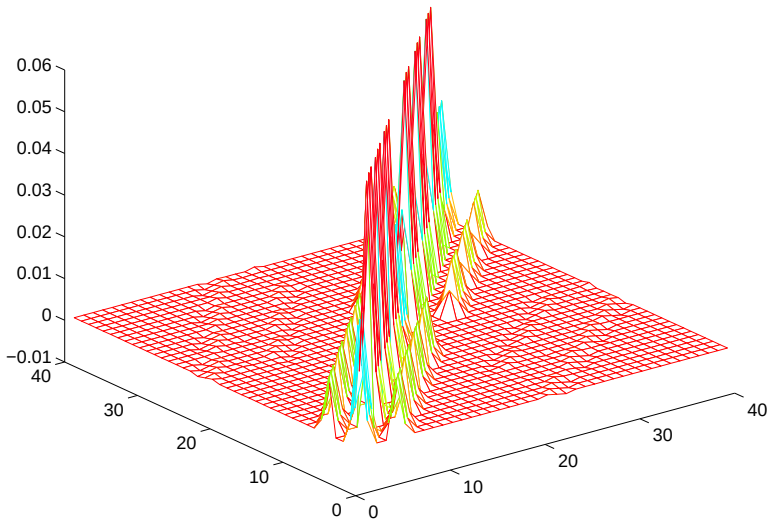


Figure 5-28 Plot of damping matrix found using the direct method with 40 DOF and 40 modes.

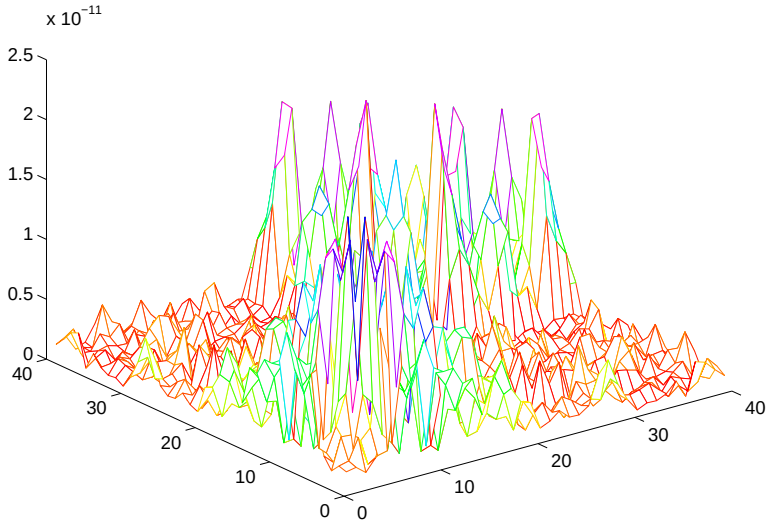


Figure 5-29 Difference between Figure 5-28 and the target damping matrix.

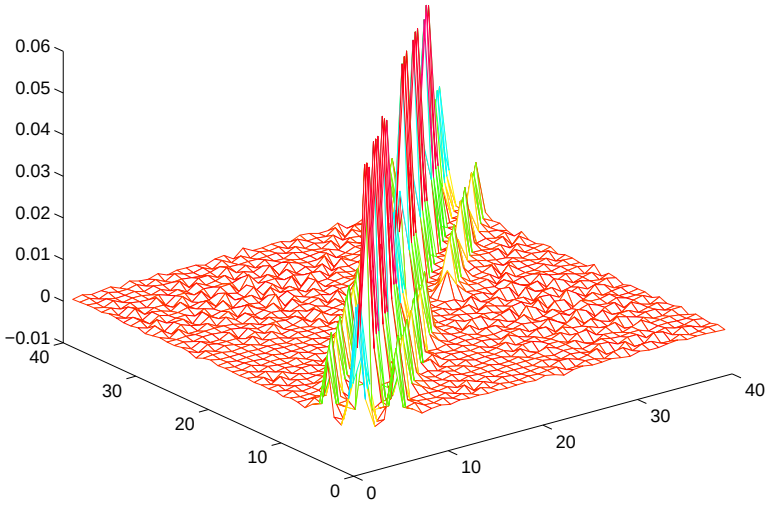


Figure 5-30 Plot of damping matrix found using the direct method with 40 DOF and 10 percent fewer modes.

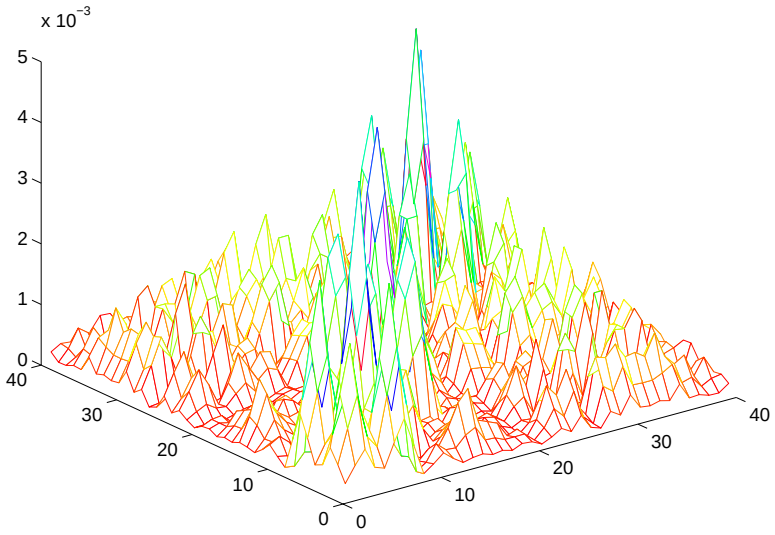


Figure 5-31 Difference between Figure 5-30 and the target damping matrix.

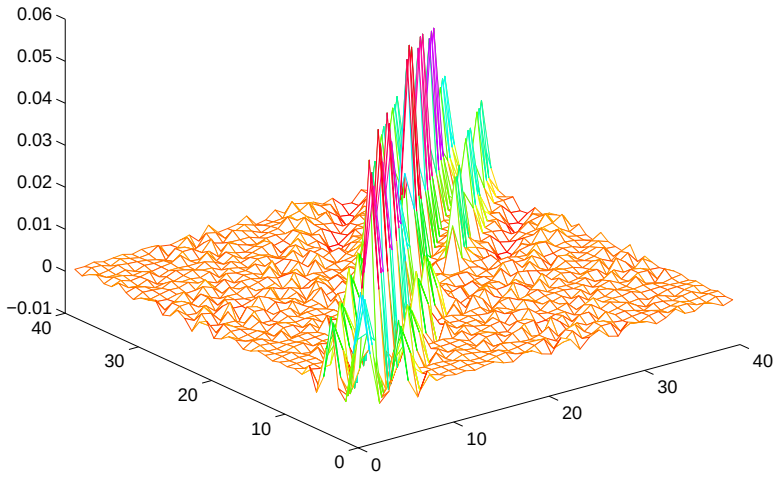


Figure 5-32 Plot of damping matrix found using the direct method with 40 DOF and 33 percent fewer modes.

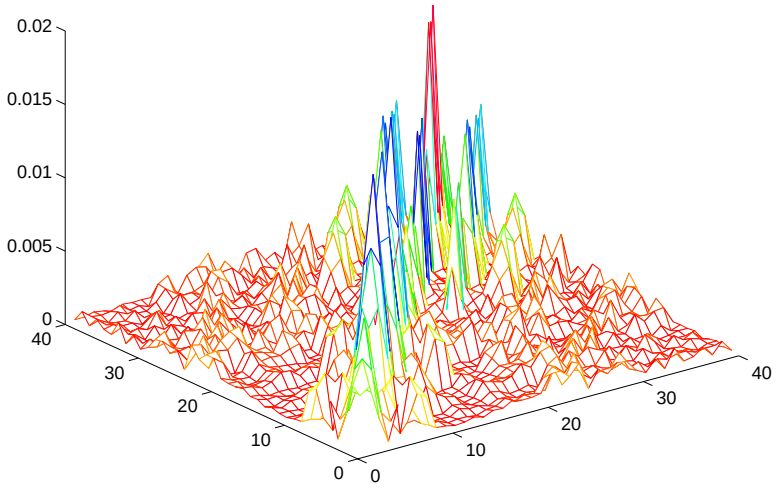


Figure 5-33 Difference between Figure 5-32 and the target damping matrix.

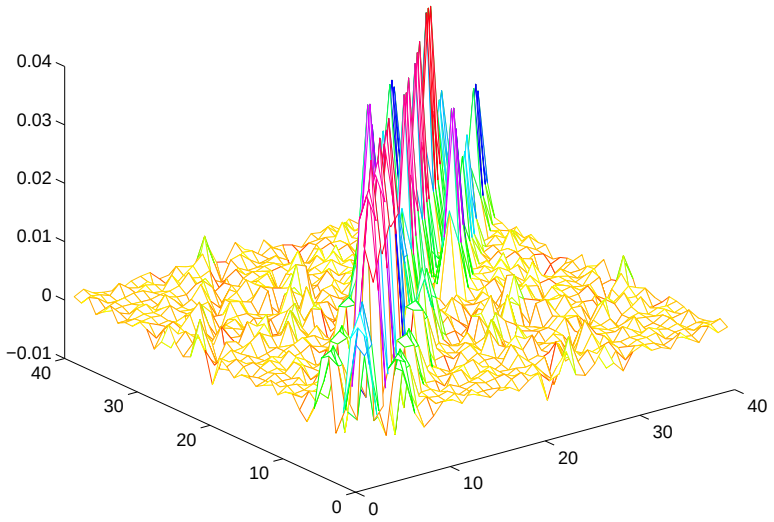


Figure 5-34 Plot of damping matrix found using the direct method with 40 DOF and 50 percent fewer modes.

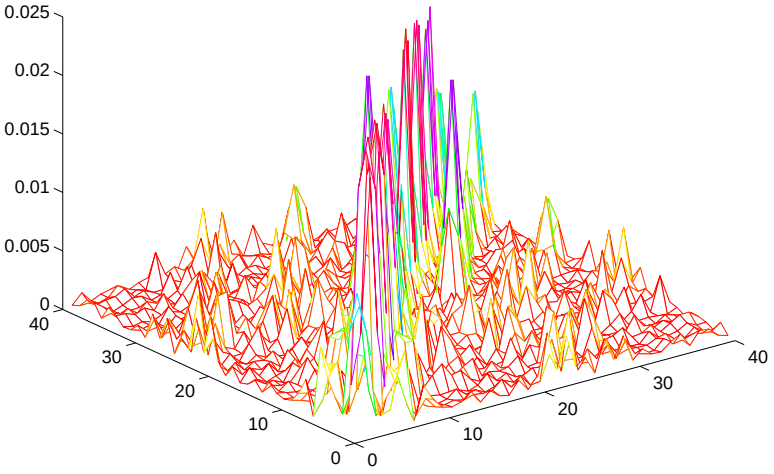


Figure 5-35 Difference between Figure 5-34 and the target damping matrix.

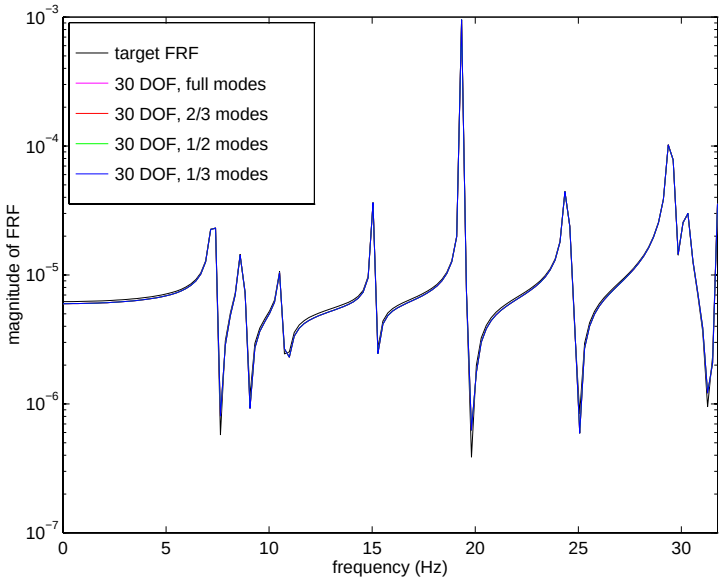


Figure 5-36 FRF plots of plate using direct damping ID, method a, and only 30 DOF.

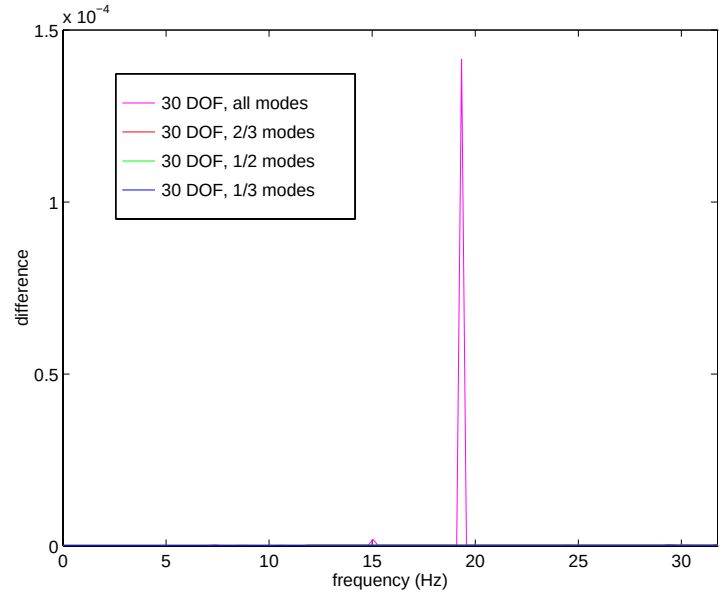


Figure 5-37 Difference between the FRF's ofFigure 5-36 and the expected FRF.

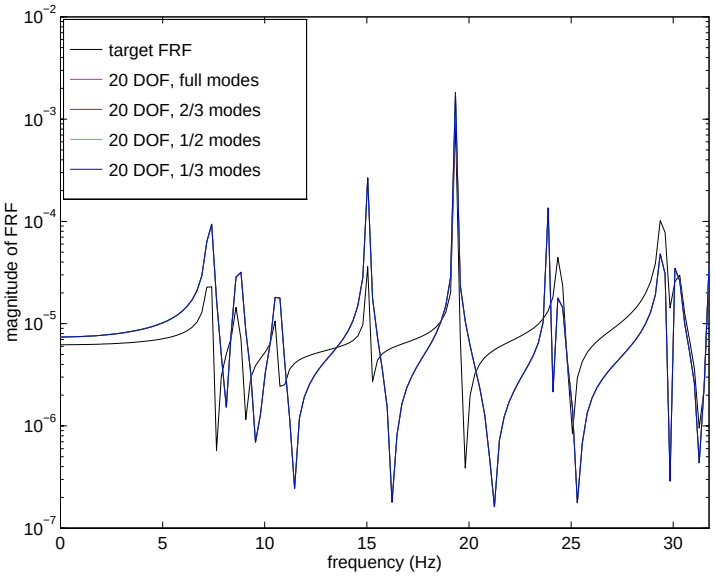


Figure 5-38 FRF plots of plate using direct damping ID, method a, and only 20 DOF.

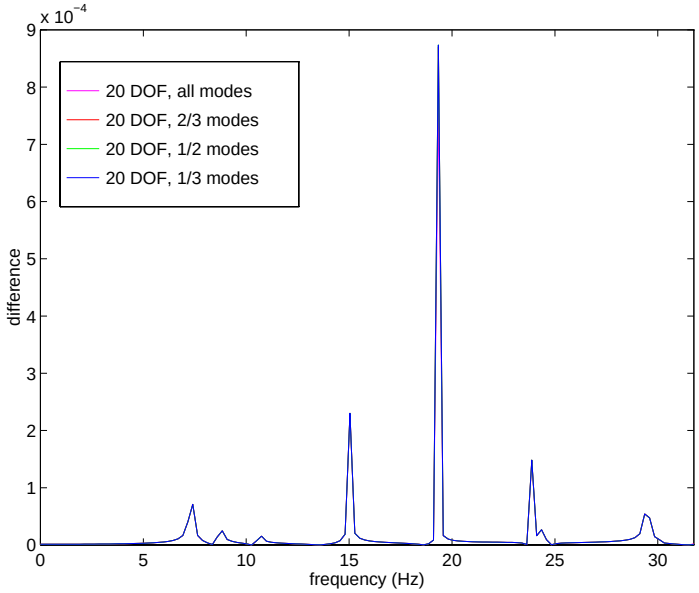


Figure 5-39 Difference between the FRF's of Figure 5-38 and the expected FRF.

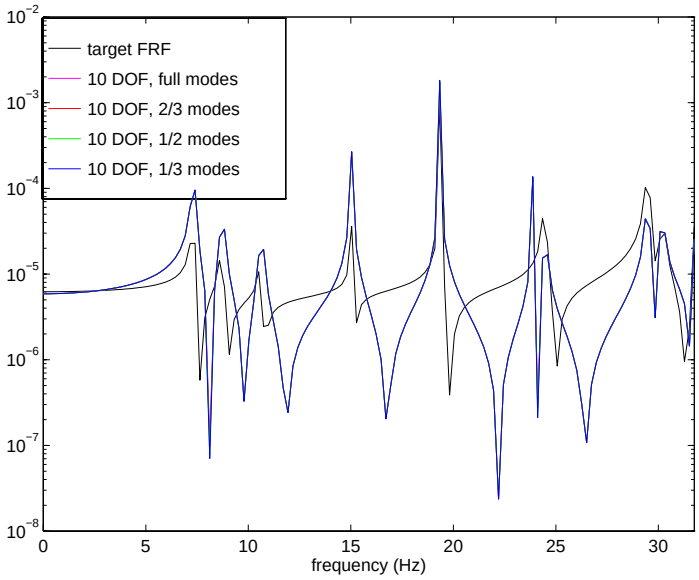


Figure 5-40 FRF plots of plate using direct damping ID, method a, and only 10 DOF.

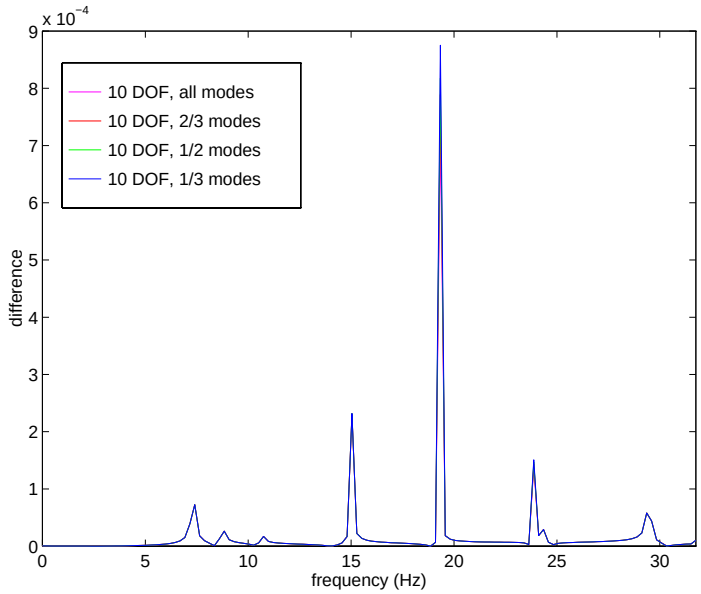


Figure 5-41 Difference between the FRF's ofFigure 5-40 and the expected FRF.

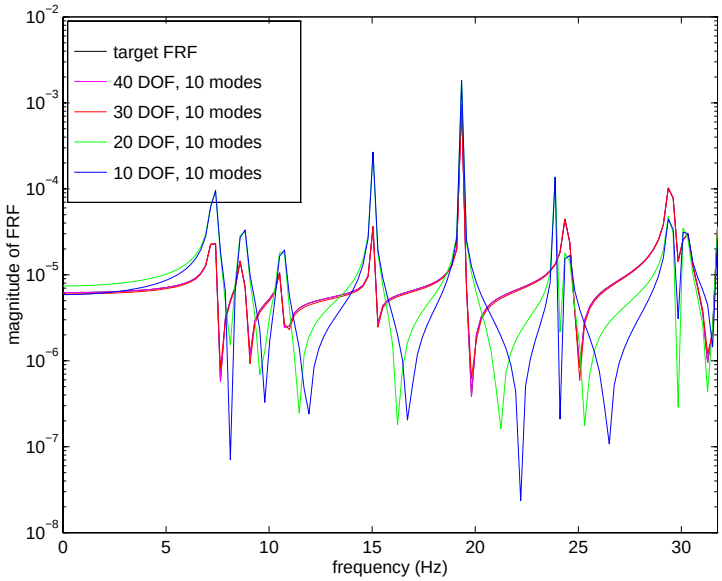


Figure 5-42 FRF plots of plate using iterative damping ID, method b, and only 10 modes.

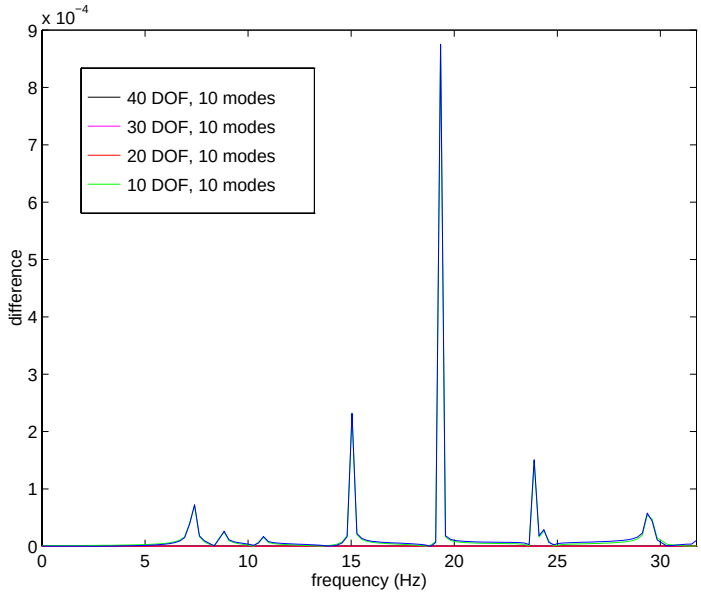


Figure 5-43 Difference between the FRF's ofFigure 5-42 and the expected FRF.

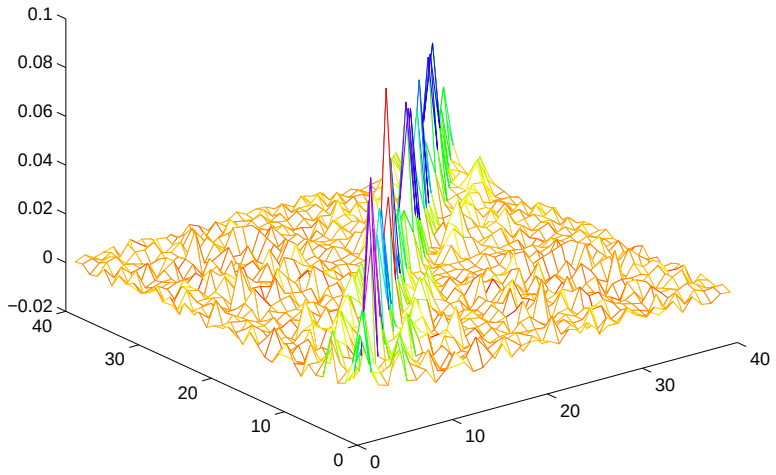


Figure 5-44 Mesh of damping matrix found using direct method of identification with noise added to the system

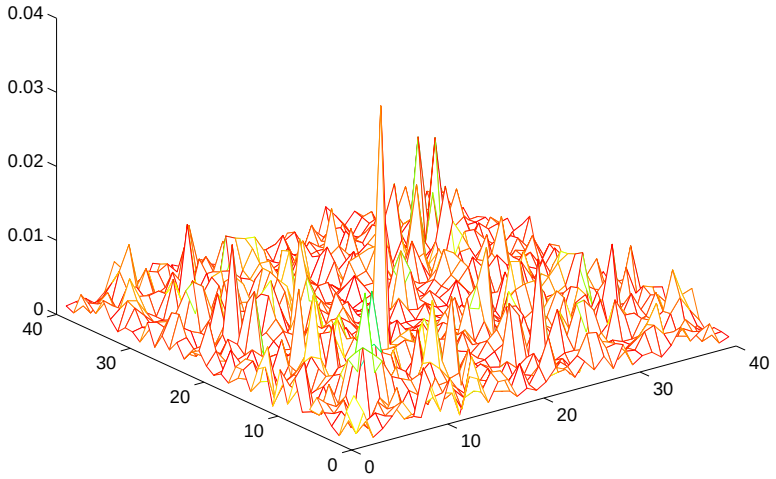


Figure 5-45 Difference between the above plot and the actual damping matrix

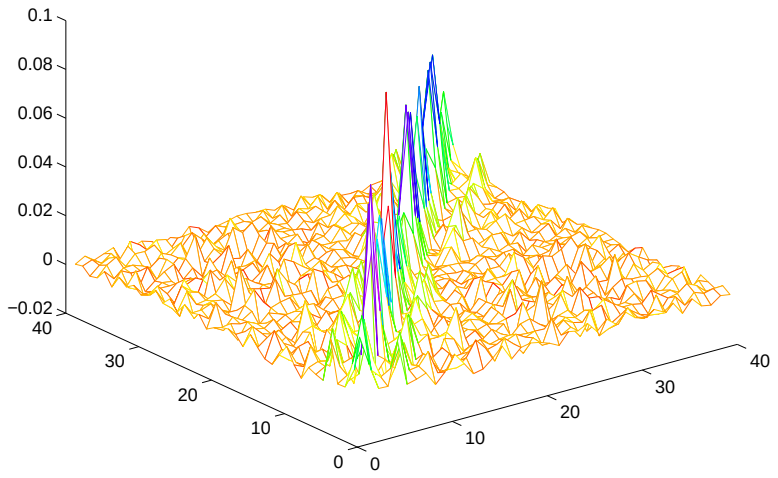


Figure 5-46 Mesh of damping matrix found using direct method of identification with 10 percent fewer modes and noise added to the system

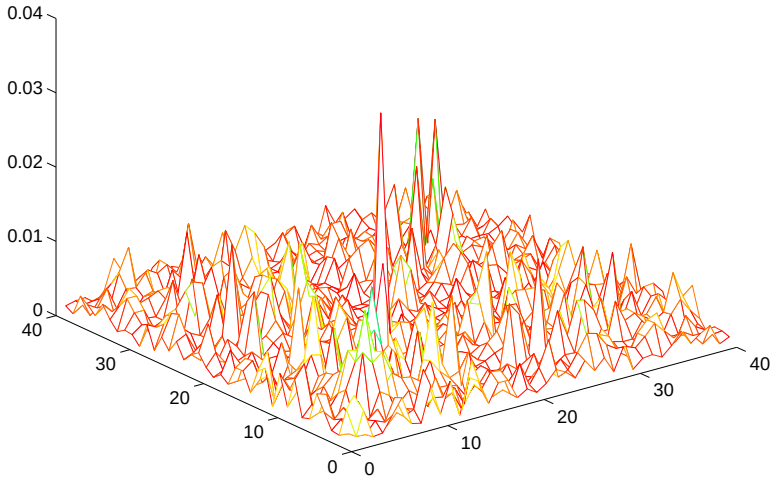


Figure 5-47 Difference between the above plot and the actual damping matrix

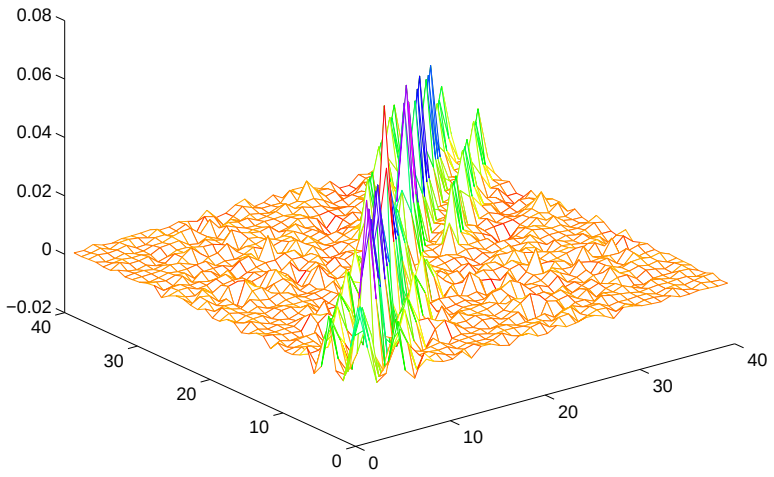


Figure 5-48 Mesh of damping matrix found using direct method of identification with 33 percent fewer modes and noise added to the system

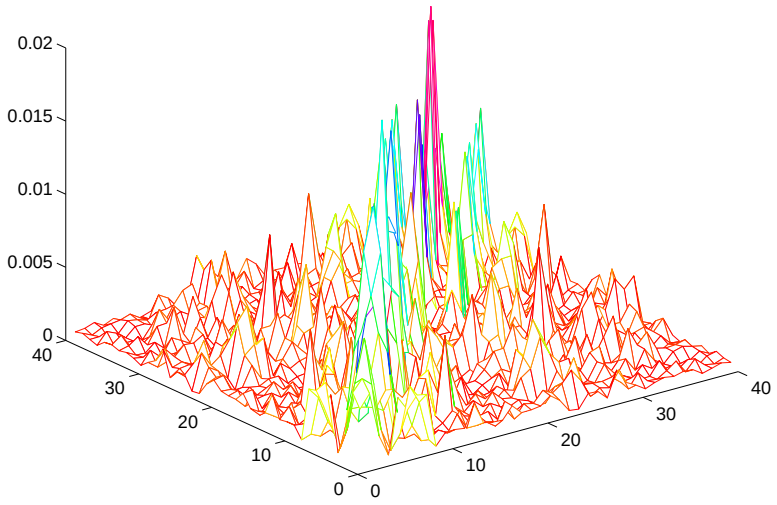


Figure 5-49 Difference between the above plot and the actual damping matrix

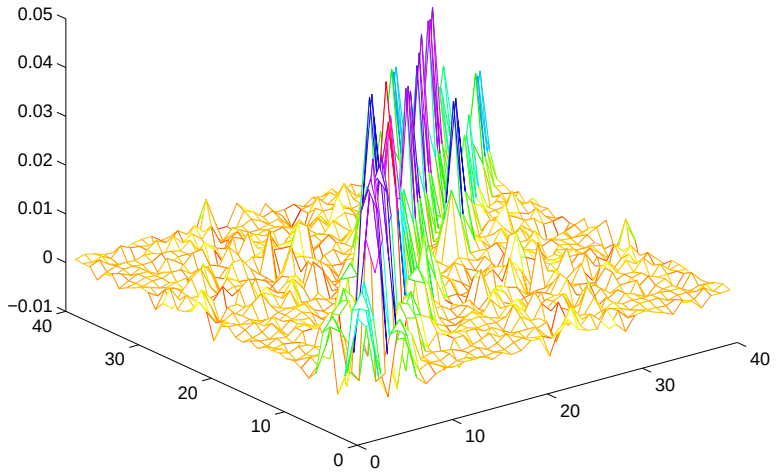


Figure 5-50 Mesh of damping matrix found using direct method of identification with 50 percent fewer modes and noise added to the system

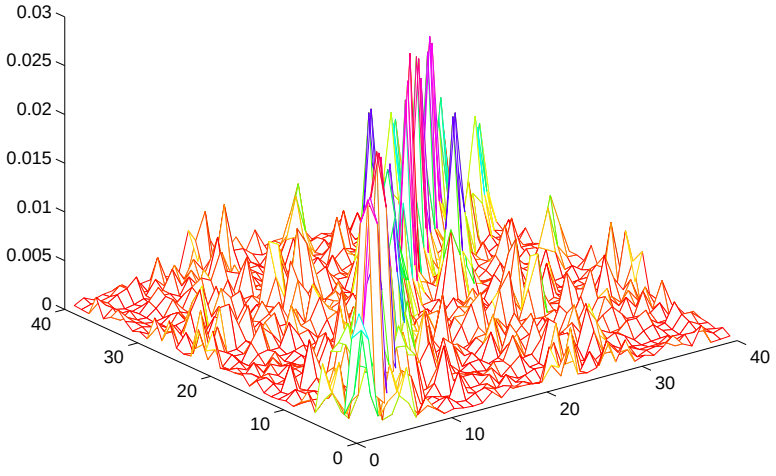


Figure 5-51 Difference between the above plot and the actual damping matrix

Chapter 6

Experimental Verification and Example of Use

6.1. Introduction

The purpose of this chapter is to illustrate how the proposed procedure works with experimental data and how it is used in conjunction with a finite element model to produce a damped model of the system. It will be illustrated that actual experimental test data can be combined with a finite element model of a structure or device, to which the damping matrix identification routines are applied. The resulting system produces a frequency response function that is comparable to that of the measured data.

Figure 6-1 Schematic of bolted beam used in the example

6.2. Experimental Setup

For this example, a modal test was performed on a two overlaid beams connected with bolts. The beam is suspended in a free-free state using fishing wire at one end. An accelerometer is attached to the last node point of the beam. Using excitation provided by an impact hammer, data is collected at several points along the beam. As seen in Figure 6-2, the accelerometer is connected to an amplifier, which is connected to the Tektronix signal analyzer. The impact hammer is connected in a similar manner.

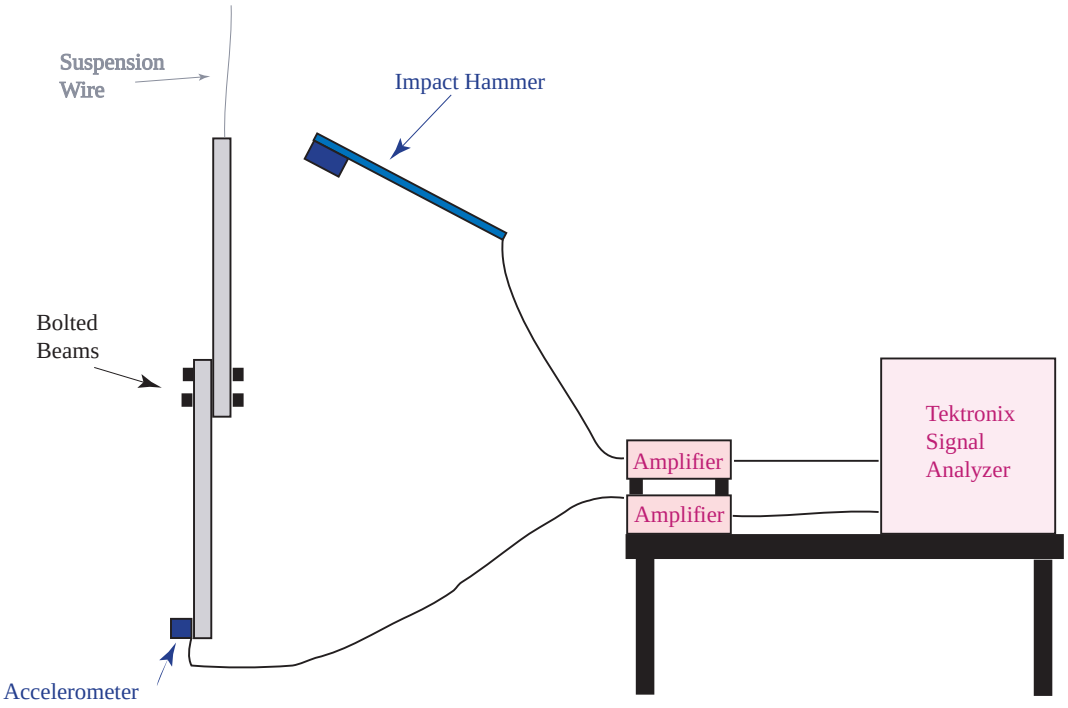


Figure 6-2 Experimental setup

The experimental setup and data collection is attributed to Gyuhae Park of The Center for Intelligent Material Systems and Structures. Figure 6-3 shows an example of the experimental test data collected on the form of the FRF, where the structure was excited at the first node with the impact hammer, and the accelerometer is positioned at the last node. The coherence plot is seen in Figure 6-4.

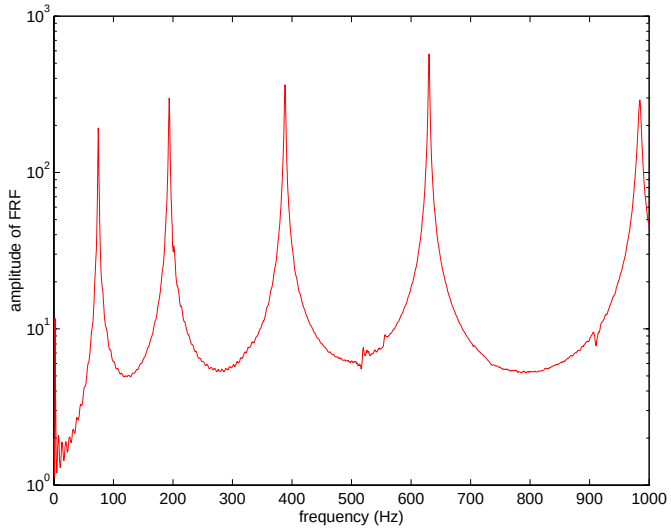


Figure 6-3 Experimental data

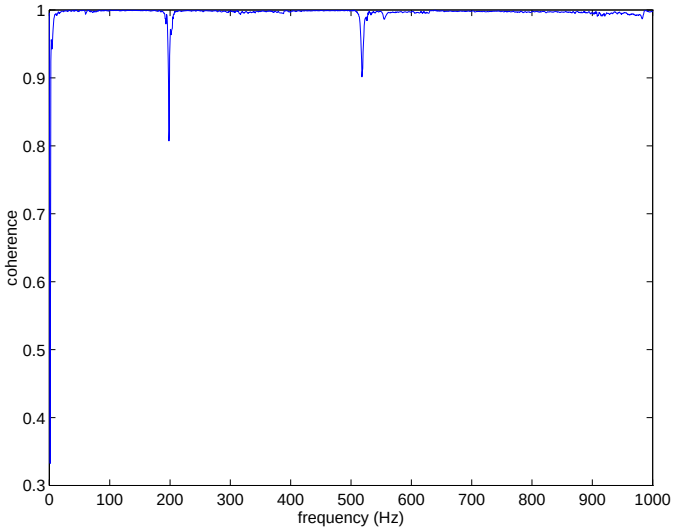


Figure 6-4 Coherence plot for experimental data

6.3. Finite Element Model

A finite element model of the bolted beam is necessary to generate mass and stiffness matrices, which are needed for the damping matrix identification procedure. The beams are modeled using Bernoulli-Euler beam theory.

$$m_{element} = \frac{Ah\rho}{420}\begin{bmatrix} 156 & 22h & 54 & -13h \\ 22h & 4h^2 & 13h & -3h^2 \\ 54 & 13h & 156 & -22h \\ -13h & -3h^2 & -22h & 4h^2 \end{bmatrix} \quad (6\text{-}1)$$

$$k_{element} = \frac{EI}{h^3}\begin{bmatrix} 12 & 6h & -12 & 6h \\ 6h & 4h^2 & -6h & 2h^2 \\ -12 & -6h & 12 & -6h \\ 6h & 2h^2 & -6h & 4h^2 \end{bmatrix} \quad (6\text{-}2)$$

In the above matrices, *h* is the element length, which is determined by specifying the number of elements desired in the problem, and the division of the beam. The mass and stiffness elements are assembled to form the global matrix.

For the aluminum beams, the Young's modulus, E is 69 x 10^9, and density is 2.715 x 10^3. The steel bolts each have Young's modulus of 2.1 x 10^{11} and density of 7.87 x 10^3. The entire bolted structure is modeled with 16 elements.

6.4. Damping Identification Procedure

Once the experimental test data is collected, complex eigenvalues and eigenvectors are generated using modal analysis software developed by Mr. Shawn Fahey of the Electric Boat Company, and based on standard modal parameter estimation theory [59], [60]. Because only a limited number of degrees of freedom can be measured (for example, no translational degrees of freedom can be measured with the standard accelerometer), the finite element mass and stiffness matrices must be reduced. The matrices are reduced from 34 degrees of freedom to only 7 measured degrees of freedom. This is done using the iterated IRS technique. At this point, it can be noted that only 5 modes have been captured experimentally.

Finally, a damping matrix is generated using the procedures defined in Chapter 2. Both the direct method (where knowledge of both mass and stiffness matrices is necessary), and the iterative method, requiring knowledge of the mass matrix, are used to

generate the damping matrices. The identification is then considered complete, and results can be compared with the initial information.

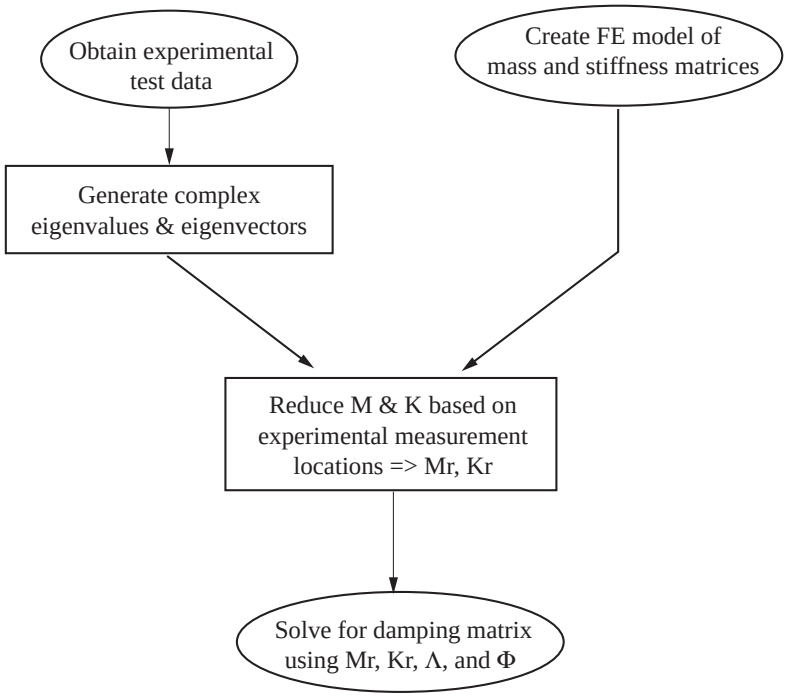


Figure 6-5 Experimental procedure

6.5. Results and Discussion

A comparison is necessary to determine the success of the method. In an experimental situation such as this, it is not possible to compare the resulting damping matrix with a known damping matrix. Instead, a clear purpose must be defined before a comparison can take place. It is the purpose of this test to characterize the experimental and finite element data. Thus, a comparison of frequency responses is in order. Figure 6-6 shows the experimental frequency response plotted with the one obtained by finite element model and those obtained using both the direct and iterative methods of damping matrix identification. It can be seen that both identification procedures produce a frequency response plot that characterizes the bolted beam adequately. This can also be seen in Figure 6-7, which shows the error of the two methods verses the experimental data.

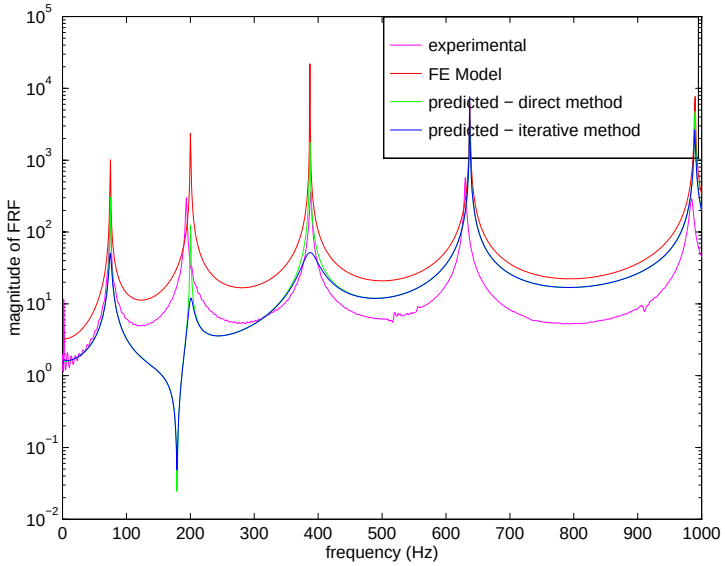


Figure 6-6 Comparison of FRFs

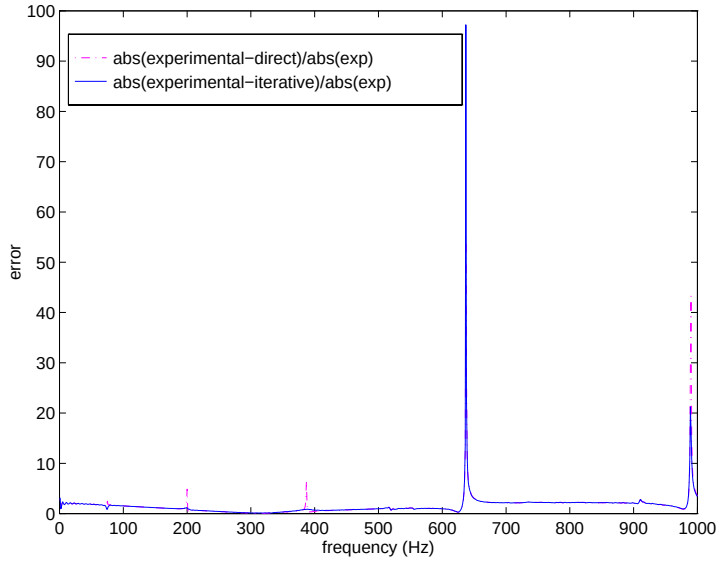


Figure 6-7 error of the above plots

It can not be expected that this example will produce an exact damping matrix. The fundamental assumptions of the identification method are violated by the fact that the experimental frequency response and the one generated by the finite element mass and stiffness matrices don't match up exactly. Added to this is the issue of reducing an imperfect model. This procedure adds additional error into the model, and creates a

situation where the experimental eigenvalues and eigenvectors do not necessarily satisfy the equation of motion. All of this, on top of the experimental and modeling errors can have a significant effect on the outcome of the methods.

Better results can be realized with careful collection of measurement data, accurate finite element models, and improved model reduction techniques. This is left as an area of further study. Future study for this type of experiment can also include the effects on the damping matrix of different torque settings on the bolts.

Chapter 7

Conclusions

The focus of this work is an in depth investigation into damping matrix identification. First, the inverse problem was introduced and interpreted in several ways. Following this, a detailed survey was made into the damping matrix identification problem. Each of the methods surveyed attempted to solve some aspect of damping matrix identification, and often allowed for at least one 'practical' issue such as noisy data or incomplete modal information. None of the methods that have previously attempted to solve this problem possess the robustness necessary to be considered complete. To the contrary, the field of damping matrix identification is not only one that holds quite a bit of intrigue in the engineering community, but is an area that is still in need of a robust, reliable solution. Also, this work is the first to address the performance of the procedures from a computational standpoint.

The solution to the damping matrix identification problem lies in two original methods that are introduced in Chapter 2. These methods include a direct method capable of solving for a damping matrix given accurately modeled mass and stiffness matrices as well as eigendata, and an iterative method, able to solve for damping with more limited information - mass matrices, eigenvalues and eigenvectors. A brief derivation of these methods is followed by a discussion of positive definiteness for underdamped systems. These two methods advance the literature in the damping matrix identification field through their simplicity and robustness.

Several practical issues needed to compare large finite element and smaller experimental models are introduced and compared. Chapter 3 introduced ideas that are used later in the work. These concepts are needed to link the mathematics behind the

inverse eigenvalue problem with the more practical test engineering issues such as spatial and modal data incompleteness. Included in this chapter is a section which advances the model updating literature by simultaneously updating stiffness and damping matrices.

Next, an investigation into the computational issues of all aspects of the damping identification problem including model reduction is performed. It is found that using high performance computing can greatly benefit all aspects of this problem through efficient use of High Performance Fortran intrinsics and data mapping capabilities. A careful explanation is made of the procedures used to parallelize static and IRS reduction as well as those for the direct and iterative methods of damping identification which lead to the most computationally efficient solution to the problem. Finally, the iterative method is compared in a computational sense to another recent method to show that with large problem sizes, the iterative method of this work is not only more practical, but also essentially the only viable solution. This computational investigation is deemed necessary because problem sizes and computing power both tend to increase significantly as more accurate and timely solutions are demanded. This is the first investigation of its kind into this type of inverse problem and its surrounding complexities.

Two example problems are presented to illustrate the procedures and their robustness. The first is a simple lumped mass system to illustrate the steps of the iterative damping matrix identification method. This simple example illustrates the potential for damage detection and diagnostics of structures. The second, an example of a forty degree of freedom plate, is presented to illustrate the robustness of both the iterative and direct methods of damping matrix identification. Results are shown for the cases of spatial and model incompleteness, as well as for noisy input data.

As an example of the use of the method, data was provided for a set of bolted beams. A finite element model was generated, as well. Using this information, the system was regenerated and compared with the experimental results.

To summarize the contributions and conclusions,

- Identification routines were developed to produce an accurate, representative damping matrix by both iterative and direct methods.
- Theory and application of damping matrix identification are integrated through and in depth investigation into model reduction, expansion and updating.
- Speedup rates for the model reduction methods show that these procedures benefit from the use of high performance computing.
- The iterative (and thus the direct) methods presented herein were shown to be computationally a significant improvement over other common methods which require solving larger systems than even the order of the original problem.
- High Performance Fortran features were implemented in an investigation of high performance computing issues associated with the iterative and direct damping identification routines. It was found that excellent speedup is available with careful attention to the details of coding. All the bottlenecks were able to be eliminated, until only the HPF intrinsics themselves became the slowest elements of the routine. In this way, it was determined that an improvement should be sought from the creators of the HPF matrix multiply intrinsic.
- The potential for these damping matrix identification procedures in the areas of damage detection and diagnostics of structures is illustrated through an example.
- The model updating method of Baruch is extended to include damping.
- Through examples, the robustness, accuracy, and use of the direct and iterative damping matrix identification routines were illustrated. In the application of damping matrix identification to actual data, an unknown amount of error exists both in the finite element model and the measurement data, and additional error is added by significantly reducing the model. It was discovered that an experimental system can be reasonably characterized.

References

[1] Banks, H.T., Inman, D.J., Leo, D.J., and Wang, Y., "An Experimentally Validated Damage Detection Theory in Smart Structures", *Journal of Sound and Vibration*, Vol. 191, No. 5, pp. 859-880, 1996

[2] Banks, H.T., and Inman, D.J., "On The Significance of Modeling Internal Damping in the Control of Structures", *Journal of Guidance, Control and Dynamics*, Vol 15, No. 6, pp. 1509-1512, 1992.

[3] Chen, S.Y., Ju, M.S., and Tsuei, Y.G., "Estimation of Mass, Stiffness, and Damping Matrices from Frequency Response Functions", *Journal of Vibration and Acoustics*, vol. 118, pp. 78-82, 1996.

[4] Pilkey, D. and Inman, D., "A Survey of Damping Matrix Identification", *Proceedings of the 16th International Modal Analysis Conference*, pp. 104-110, February 1998.

[5] Minas, C., and Inman, D.J., "Identification of a Nonproportional Damping Matrix from Incomplete Modal Information," *Journal of Vibration and Acoustics*, Vol. 113, pp. 219 - 224, 1991.

[6] Beliveau, J.-G., "Identification of Viscous Damping in Structures From Modal Information," *ASME Journal of Applied Mechanics*. Vol. 43, pp. 335-338, 1976.

[7] Caravani, P. and Thomson, W.T., "Identification of Damping Coefficients in Multidimensional Linear Systems," *ASME Journal of Applied Mechanics*, vol. 41, pp. 379-382, 1974.

[8] Fabunmi, J. Chang, P., and Vorwald, J., "Damping Matrix Identification Using the Spectral Basis Technique," *ASME Journal of Vibration, Acoustics, Stress, and Reliability in Design*, vol. 110, pp. 332-337, 1988.

[9] Fritzen, C.-P., "Identification of Mass, Damping, and Stiffness Matrices of Mechanical Systems," *ASME Journal of Vibration, Acoustics, Stress, and Reliability in Design*, vol. 108, pp. 9-16, 1986.

[10] Gaylard, M.E., "Identification of Damping Matrices: An Autocorrelation-Style Technique," *Identification in Engineering Systems. Proceedings of the Conference*, pp. 225-237, 1996.

[11] Hasselman, T.K., "A Method of Constructing a Full Modal Damping Matrix from Experimental Measurements," *AIAA Journal*, Vol. 10, pp. 526-527, 1972.

[12] Ibrahim, S.R., "Dynamic Modeling of Structures from Measured Complex Modes", *AIAA Journal*, Vol. 21, no. 6, pp. 898-901, 1983.

[13] Lancaster, P., "Expression for damping Matrices in Linear Vibration Problems," *Journal of the Aerospace Sciences*, p. 256, 1961.

[14] Roemer, M.J. and Mook, D.J., "Mass, Stiffness, and Damping Matrix Identification: An Integrated Approach," *ASME Journal of Vibration and Acoustics*, vol. 114, pp. 358-363, 1992.

[15] Starek, L., and Inman, D.J., "A Symmetric Inverse Vibration Problem for Non-proportional, Underdamped Systems", *ASME Journal of Applied Mechanics*, Vol. 64, no. 3, 1997.

[16] Wang, J.-H., "Mechanical Parameter Identification with Special Consideration of Noise Effects," *Journal of Sound and Vibration*, Vol. 125, no. 1, pp. 151-167, 1988.

[17] Alvin, K.F., Park, K.C., Peterson, L.D., "Extraction of Undamped Normal Modes and Nondiagonal Damping Matrix from Damped System Realization Parameters," *Journal of Sound and Vibration*, pp. 151-167, 1993.

[18] Beattie, C., and Smith, S.W., "Matrix Approximation in the Identification of Vibrating Structures", *Proceedings of the Conference on New Developments in Structural Mechanics*, pp. 271-287, 1990.

[19] Liang, Z. and Lee, G.C., "Representation of a Damping Matrix," *Journal of Engineering Mechanics*, Vol. 117, no. 5, pp. 1005-1020, 1991.

[20] Link, M., "Theory of a Method for Identifying Incomplete System Matrices Using Vibration Test Data," *Z. Flugwiss. Weltroumforsch*, Vol. 9, pp. 76-82, 1985.

[21] Milne, R.D., "On the Estimation of Structural Damping from Aircraft Resonance Tests," *Journal of the Aerospace Sciences*, Vol. 27, No. 5, pp. 339-343, 1960.

[22] Mottershead, J.E. and Foster, C.D., "An Instrument variable Method for the Estimation of Mass, Stiffness, and Damping Parameters From Measured frequency Response Functions," *Mechanical Systems and Signal Processing*, Vol. 2, No. 4, pp. 379-390, 1988.

[23] He, J., and Ewins, D.J., "Identification of Damping Properties in Vibrating Structures," *Proceedings MOVIC*, Japan, September 1992.

[24] Tseng, D.-H., Longman, R.W., and Juang, J.-N., "Identification of Second Order Mechanical System Model From State Space Realization," *Advances in the Aerospace Sciences*, Vol. 82, pp. 347-360, 1993.

[25] Lancaster, P. "Inversion of Lambda-Matrices and Application to the Theory of Linear Vibrations," *Archive for rational Mechanics and Analysis*, Vol. 6, pp. 105-114, 1960.

[26] Pilkey, D. and Inman, D., "An Iterative Approach to Viscous Damping Matrix Identification", *Proceedings of the 15th International Modal Analysis Conference*, pp. 1152-1159, February 1997

[27] Ortega, J.M., *Matrix Theory, A Second Course*. Plenum Press, New York, 1987.

[28] Starek, L., and Inman, D.J., "A Symmetric Inverse Vibration Problem With Overdamped Modes", *Journal of Sound and Vibration*, Vol. 181, No. 5, pp. 893-903, 1995.

[29] Gohberg, I., Lancaster, P., and Rodman, L., *Matrix Polynomials*, Academic Press, New York, 1982.

[30] D. J. Ewins. *Modal Testing: Theory and Practice*, Research Studies Press Ltd., England 1984.

[31] Penny, J.E.T., Friswell, M.I., and Garvey, S.D., "Automatic Choice of Measurement Locations for Dynamic Testing", *AIAA Journal*, Vol. 33, No. 2, 407-414, 1992.

[32] Guyan, R.J. , "Reduction of Stiffness and Mass Matrices," *AIAA Journal*. Vol. **3,** No. 2, p.280, 1965.

[33] Paz, M., "Dynamic Condensation," *AIAA Journal*, Vol.22, No. 5, pp.724-727, 1984.

[34] O'Callahan, J.C. , "A procedure for an Improved Reduced System (IRS) Model," *Proceedings of the Seventh International Modal Analysis Conference*, pp. 17-21, 1989.

[35] Friswell, M.I., Garvey, S.D. and Penny, J.E.T., "Model Reduction using Dynamic and Iterated IRS Techniques", *Journal of Sound and Vibration*, Vol. 186, No. 2, pp. 311-323, 1995.

[36] Friswell, M.I., Garvey, S.D. and Penny, J.E.T., "Using Iterated IRS Model Reduction Techniques to Calculate Eigensolutions", *Proceedings of the 15th International Modal Analysis Conference*, pp.1537-1543, 1997.

[37] Friswell, M.I., Garvey, S.D. and Penny, J.E.T., "The Convergence of the Iterated IRS Method", *Journal of Sound and Vibration*, Vol. 211, No. 1, pp. 123-132, 1998.

[38] Henshell, R.D. and Ong, J.H. "Automatic Masters for Eigenvalue Economisation", *Earthquake Engineering and Structural Dynamics*. Vol. 3, 375-383. 1975.

[39] Friswell, M.I., and Mottershead, J.E**.,** *Finite Element Model Updating in Structural Dynamics*, Kluwer Academic, Norwell, MA, 1995.

[40] Mottershead, J.E., and Friswell, M.I., "Model Updating in Structural Dynamics: A Survey", *Journal of Sound and Vibration*, Vol. 167, No. 2, pp. 347-375, 1993.

[41] Baruch, M., "Optimization Procedure to Correct Stiffness and Flexibility Matrices Using Vibration Data", *AIAA Journal*, Vol. 16, No. 11, pp. 1208-1210, 1978.

[42] Baruch, M., "Methods of Reference Basis fot Identification of Linear Dynamic Structures", *Proceedings of the AIAA 23rd Structures, Structural Dynamics, and Materials Conference*, pp.557-563, 1982 (AIAA paper 82-0769).

[43] Friswell, M.I., Inman, D.J., and Pilkey, D.F., "Direct Updating of Damping and Stiffness Matrices", *AIAA Journal*, Vol. 36, No. 3, pp. 491-493, 1998.

[44] Ahmadian, J.E., Mottershead, H., Friswell, M.I., "Regularisation Methods for Finite Element Model Updating", *Mechanical Systems and Signal Processing*, Vol 12, No. 1, pp. 47-64, 1998.

[45] Wei, F.-S., "Structural Dynamic Model Identification Using Vibration Test Data", ", *Proceedings of the 7th International Modal Analysis Conference*, pp.562-567, 1989.

[46] Wei, F.-S., "Structural Dynamic Model Improvement Using Vibration Test Data", *AIAA Journal*, Vol. 28, No. 1, pp. 175-177, 1990.

[47] Wei, F.-S., "Mass and Stiffness Interaction Effects on Analytical Model Modification", *AIAA Journal*, Vol. 28, No. 9, pp. 1686-1688, 1990.

[48] High Performance Fortran Forum. "High performance Fortran Language Specification, version 1.0". Technical report CRPC-TR92225, Center for Research on Parallel Computing, Rice University, Houston, TX, 1992.

[49] Wilkes, M.J., "From Fortran and Algol to Object-Oriented Languages", *Communications of the ACM*, Vol. 36, No. 7, pp. 21-23, July 1993.

[50] Pilkey, D., Ribbens, C., and Inman, D." High Performance Computing Issues For Model Reduction/Expansion", *Advances in Engineering Software*, Vol. 29, No. 2, March 1998.

[51] O'Callahan, J.C., Avitable, P., Reimer, R., "System Equivalent Reduction Expansion Process," *Proceedings of the Seventh International Modal Analysis Conference*, pp. 29-37, 1989.

[52] Lawson, C., Hanson, R., Kincaid, D., and Krogh F., "Basic Linear Algebra Subprograms for Fortran Usage", *ACM Transactions on Mathematical Software*, Vol. 5, pp. 308-323, 1979.

[53] Dongarra, J.J., Du Croz, J., Hammarling, S., and Hansen, R.J., "An Extended Set of Fortran Basic Linear Algebra Subroutines", *ACM Transactions on Mathematical Software*, Vol. 14, No. 1, pp. 1-17, 1988.

[54] Personal Communication from Portland Group, Inc. April 5, 1998.

[55] Pilkey, D., Roe, K. and Inman, D., "Computational Issues in Damping Identification for Large Scale Problems", *Proceedings 1997 ASME Design Technical Conference*, paper No. DETC97/VIB-3835, September 1997.

[56] Gropp, W., Lusk, E., and Skjellum, A., *Using MPI*, MIT Press, Boston, MA, 1994.

[57] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., and Sunderam, V., *PVM: Parallel Virtual Machine*, MIT Press, Cambridge, MA, 1994.

[58] Inman, D. J., *Vibration with Control Measurement and Stability*, Prentice Hall, 1989.

[59] Richardson M., Formenti D.L., Parameter estimation from Frequency Response Measurements using RFP", *Proceeding of the International Modal Analysis Conference*, pp.167-182, 1982.

[60] Allemang R.J., Brown D.L., Fladung W., Modal Parameter Estimation: UMPA, ", *Proceeding of the International Modal Analysis Conference*, pp.501-514, 1994.

[61] Golub, G.H., and Van Loan, C.F. *Matrix Computations, Third Edition*, Johns Hopkins University Press, Baltimore, 1997.

[62] Anderson, E., Bai, Z, Bischof, C., Demmel, J., Dongarra, J., DuCroz, J., Greenbaum, A., Hammerling, S., McKenney, A., and Sorensen, D., "LAPACK: A Portable Linear Algebra Library for High - Performance Computers", University of Tennessee technical report No. UT CS 90-105, May 1990.

[63] Koelbel, C. H. , Loveman D. B. , Schreiber, R. S. , Steele, G. L. Jr., and Zosel, M. E. , *The High Performance Fortran Handbook*, The MIT Press, Cambridge, Massachusetts, 1994.

Appendices

Appendix A: Flops Count

One way to quantify the arithmetic complexity of sections of an algorithm or code is to count the number of flops. A flop is defined as a floating point operation [61]. Flop counting can provide insight into the issue of program efficiency. It is considered a necessary procedure, although somewhat crude because of the omission of effects and issues of program execution such as processor memory constraints, communication costs, etc.

Several examples of flop counts follow that aided in the analysis found in Chapter 4. The basic matrix-matrix multiplication,

$$\mathbf{C}_{(l \text{ x } n)}=\mathbf{A}_{(l \text{ x } m)}\mathbf{B}_{(m \text{ x } n)} \tag{A1-1}$$

requires $2lmn$ flops. The basic matrix addition,

$$\mathbf{C}_{(m \text{ x } n)}=\mathbf{A}_{(m \text{ x } n)}+\mathbf{B}_{(m \text{ x } n)} \tag{A1-2}$$

where the matrices **A** and **B** are m x n requires mn flops. Matrix - scalar multiplication,

$$\mathbf{C}_{(m \text{ x } n)}=\alpha\mathbf{A}_{(m \text{ x } n)} \tag{A1-3}$$

where the matrix **A** is m x n requires mn flops. Several more complex counts that were implemented include factoring and solving matrices, using LAPACK [62] subroutines. An LU factorization for an n x n matrix requires $^2/_3\ n^3$ flops. The banded Cholesky factorization for an n x n symmetric, positive definite matrix with half bandwidth k requires $n(k^2+k)$ flops. A banded triangular solve for

$$\mathbf{A}_{(n \times n)}\mathbf{x}_{(n \times m)}=\mathbf{B}_{(n \times m)} \qquad \text{(A1-4)}$$

where **A** is n x n, **x** is n x m and **A** is symmetric and positive definite with half bandwidth k, and **A** is already factored required $4mnk$ flops.

There are several operations used in the codes for this dissertation that cost nothing in terms of flops. Included in these are matrix transposes and complex conjugates of a matrix.

Appendix B: Data Mapping

In the High Performance Fortran programming language [63], matrices and vectors can be mapped in certain ways onto multiple processors in ways that optimize the use of the parallel processors. There are two stages involved in mapping data to the computer's processors. The first is `distribute`, which describes how a matrix or vector is divided into evenly sized pieces and then distributed to the processors. Data can be distributed in variations of block or cyclic patterns. The second stage in data mapping is to `align` arrays with each other. If two arrays are always distributed the same, they can be lined up with the `align` statement.

The following matrix vector multiplication is a good example of data mapping. In matrix multiplication, recall that each row of the left-hand matrix is multiplied with the corresponding column of the right hand matrix. It is possible to reduce the communication required by each processor by distributing the data correctly. For example, the n x n matrix **A** is multiplied with vector **b**. The matrix **A** can be row distributed by using (block,*) distribution, and **b** can be replicated, so that a copy exists on each processor.

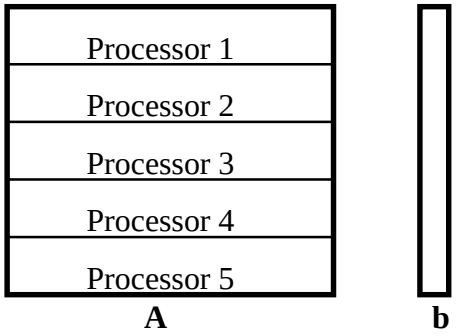


Figure B-1: Data distribution of matrix **A** (block, *) and vector **b** (replicated).

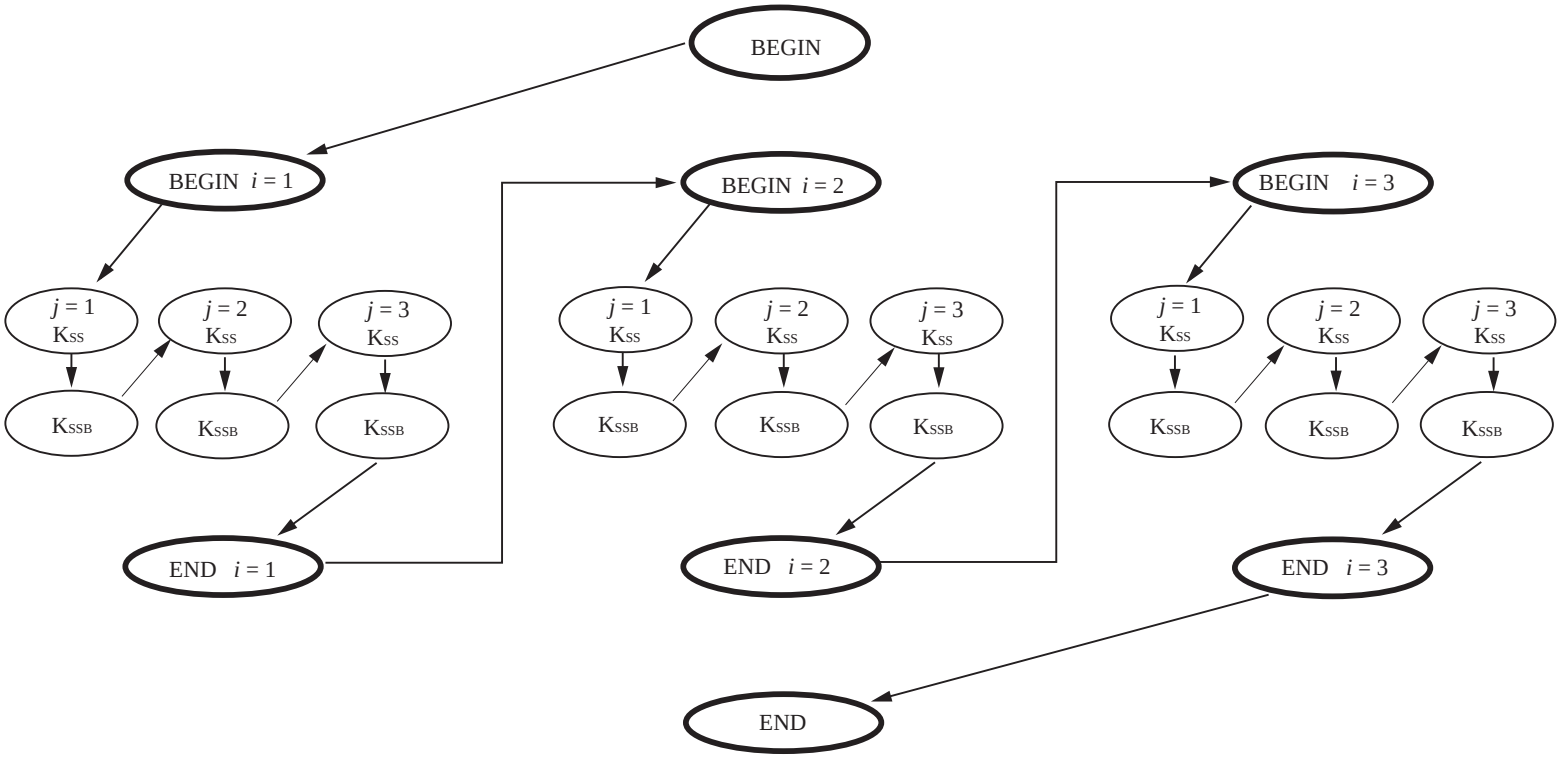
The above figure illustrates the distribution process. Assuming that we have 5 processors, the (block,*) distribution shown maps groups (or blocks) of rows to each of the five processors. Because the vector **b** is replicated on each processor, the amount of communication is minimized for this example

Appendix C: HPF Attributes

This simple example is used to show the difference between a `do` statement, and a `forall` statement, and then the potential benefits of the HPF independent intrinsic. The section of code describe in this appendix was taken from codes used for model reduction found in Chapter 4. The objective of this small section is to rearrange the elements of a matrix. The code is intended for use with large matrices, but for this illustration, a 3 x 3 matrix is sufficient to make the point.

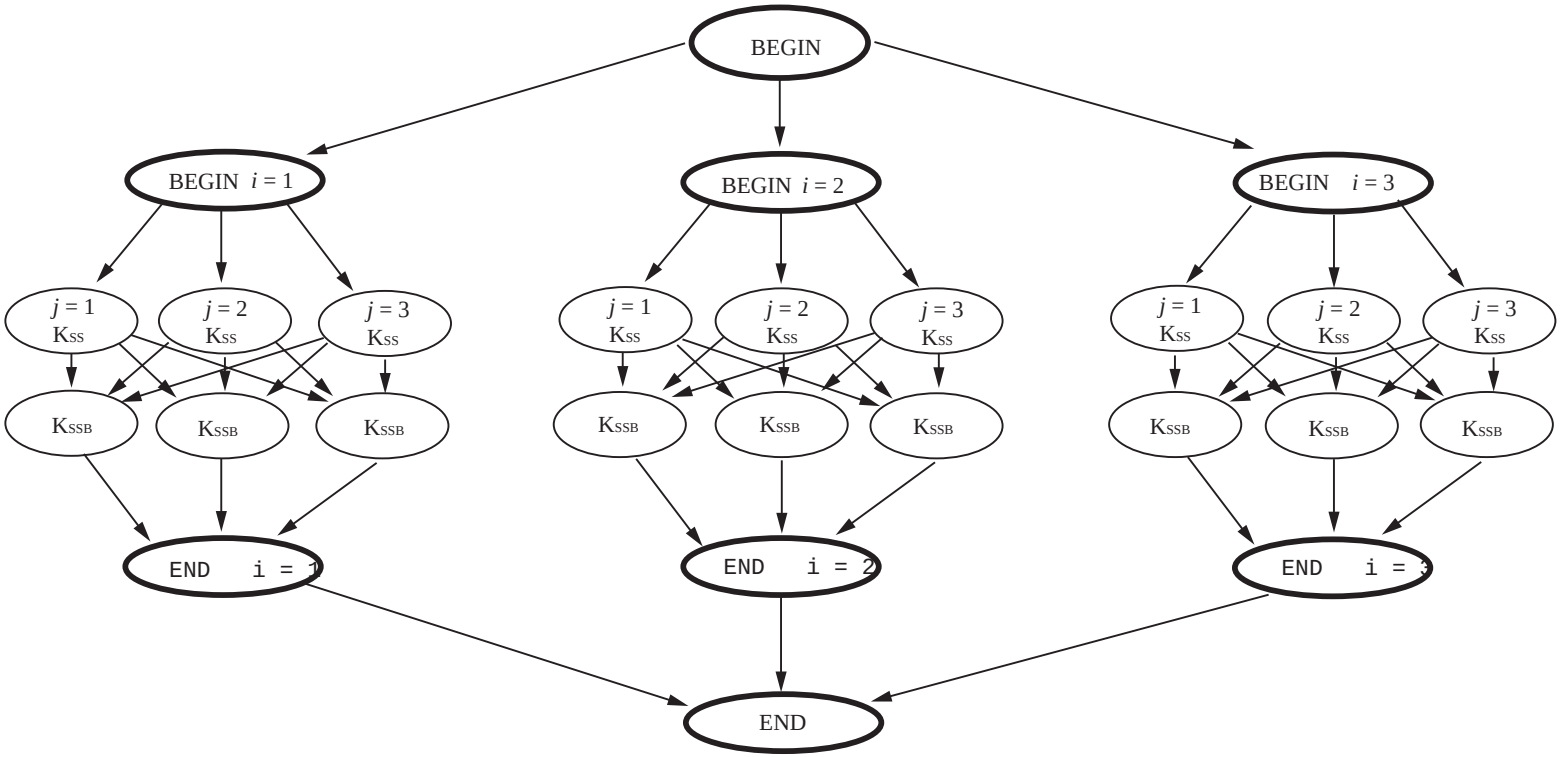
Below is a section of code that was improved by exploiting the features of HPF discussed above. This loop is intended to take a full matrix ($\mathbf{K}_{ss}$) and move each element so that it is in a form specified by LAPACK which allow the bandedness to be exploited. The original Fortran 90 code was written with nested do loops, and a separate nested if statement.

```
do i=1,ns
 do j=1,ns
  if ((max(1,j-kd) .le. i) .and. (i .le. j)) then
    Kssb(kd+1+i-j,j) = Kss(i,j)
  end if
 end do
end do
```



The improved version exploits several aspects of HPF. The forall statement can handle all three conditions in one line, and is able to exploit the multiple processors. Adding an independent statement indicates that each element of the new matrix (Kssb) does not depend on any of the others. This is the most efficient use of HPF for this problem.

```
forall(i=1:ns,j=1:ns,((max(1,j-kd).le.i).and.(i.le.j)))Kssb(kd+1+I-j,j)=Kss(i,j)
```


BEGIN
BEGIN i = 1
BEGIN i = 2
BEGIN i = 3
j = 1
Kss
j = 2
Kss
j = 3
Kss
KSSB
KSSB
KSSB
END i = 1
END i = 2
END i = 3
END

Vita

Deborah F. Pilkey was born on March 22, 1971 to Barbara and Walter Pilkey in Charlottesville, Virginia. She obtained her undergraduate education at Duke University in the Civil Engineering Department, from which she received a BSE with distinction in May 1993. She then moved to Palo Alto, California to attend Stanford University, where she received a masters degree in Civil Engineering Structures in June 1994. Most recently, she received a Ph.D. from the Engineering Science & Mechanics Department at Virginia Tech in Blacksburg, VA. Debbie will greatly miss the freedom of academia and the natural beauty of Blacksburg when she finally enters the "real world".