NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CALCULATING REQUIRED SUBSTRUCTURE DAMPING TO MEET PRESCRIBED SYSTEM DAMPING LEVELS

by

Wendel Domingo Penetrante

June 2007

Thesis Advisor: Joshua Gordis

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2007	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Calculating Required Substructure Damping to Meet Prescribed System Damping Levels		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Wendel D Penetrante			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

Structural synthesis is a method of calculating the transient dynamic response of an assemblage of substructures without explicitly assembling and solving a combined system model. While significant computational advantages are provided by this method, the modal parameters of the combined system are not explicitly calculated. Hence, a method is needed to allow the a priori determination of the substructure damping levels such that the synthesized system damping is within user-prescribed bounds. This thesis focuses on the development of such a method.

14. SUBJECT TERMS Substructure Modal Damping Synthesis			**15. NUMBER OF PAGES** 93
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

CALCULATING REQUIRED SUBSTRUCTURE DAMPING TO MEET PRESCRIBED SYSTEM DAMPING LEVELS

Wendel D. Penetrante
Lieutenant, United States Navy
B.S., University of California San Diego, 1999

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2007

Author: Wendel Penetrante

Approved by: Joshua Gordis
Thesis Advisor

Anthony J. Healey
Chairman, Department of Mechanical and Astronautical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Structural synthesis is a method of calculating the transient dynamic response of an assemblage of substructures without explicitly assembling and solving a combined system model. While significant computational advantages are provided by this method, the modal parameters of the combined system are not explicitly calculated. Hence, a method is needed to allow the a priori determination of the substructure damping levels such that the synthesized system damping is within user-prescribed bounds. This thesis focuses on the development of such a method.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

NOMENCLATURE

C = physical damping matrix

M = physical mass matrix

K = physical stiffness matrix

$\mathcal{C}$ = modal damping matrix

$\mathcal{M}$ = modal mass matrix, identity matrix [I]

$\mathcal{K}$ = modal stiffness matrix

T = kinetic energy

D =damping dissipation function

U = potential energy

λ = lagrange multiplier

Φ = mass normalized modal matrix

ϕ = mode shape

β = another way to represent mode shape, individual vector, or element of mode shape

ω = natural frequency Hz

ζ = modal damping

Ω_d = undamped natural frequency

$\sigma = \zeta\omega$, modal damping times natural frequency

x = physical coordinate

q = modal coordinate

[I] = identity matrix

f = interconnecting condition between substructure

t = time

Subscripts

j = mode index

r = number of interconnecting region

i = element of substructure

Superscript

n = substructure designator

’ = transpose

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

To my wife Sheryl and kids Kalana, Kaleb and Kaden

To Professor Gordis

To all of my Mechanical Engineering and Mathematics Professors

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Structural synthesis is the process of calculating the response of an assemblage of two or more substructures where each substructure is represented by impulse response functions calculated at the connection degrees-of-freedom only. In addition, since it is often difficult or even impossible to conduct vibrations tests on the system, it is often desired to extract system's dynamic response by breaking the system into substructures and conduct vibrations test on these individual substructures. The dynamic responses from the substructures are then synthesized to provide the dynamic response for the system. In this thesis we will exploit the concept of substructure coupling which makes use of linear homogenous equations of constraints; such a method enables us to generate the system dynamic response by making use of substructure modal properties (transformed physical properties to modal properties) and coupling constraints. The method that will be implemented to calculate the system response is based on the zero-eigenvalues theorem developed by Walton and Steeves [16].

With the ability to calculate system dynamic response we are now able to go further and investigate a method of determining bounds for substructure damping that would satisfy prescribed bounds for system damping given only substructure physical properties such as mass and stiffness and coupling constraints. In support of the task in hand, we will also make use of other concepts such as Lagrange's equation to attain equations of motion, Lagrange multipliers and holonomic constraints to provide connectivity for each substructure, orthogonal complement and zero-eigenvalue theorem to impose the holonomic constraints and synthesize the substructure and finally an optimization technique to determine the necessary values for substructure damping that would fulfill the required bounds for system damping.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. LAGRANGE

1. Theory and Derivation

Lagrange's equations are commonly used to generate the equations of motion for second-order systems It utilizes the energy functional of kinetic (T), and potential (U) energies, and the damping dissipation function (D) to derive equations of motion for a given system.

A general form of Lagrange's equation is,

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right)-\frac{\partial U}{\partial q}+\frac{\partial D}{\partial \dot{q}}=Q_i$$

where q≡q(t) is a generalized coordinate of the system and Q_i is the generalized non-potential force or moment resulting from excitations that adds energy to the system.

The energies will be derived in modal coordinates. The following will describe how each components of the Lagrange's equation is derived symbolically:

First we must start with the basic equation of motion in physical coordinates,

$$M\ddot{x}+C\dot{x}+Kx=Q_i$$

where, M is the mass matrix, C is the damping matrix, K is the stiffness matrix and x≡x(t). Using M and K matrices, natural frequencies (ω) and mode shapes (ϕ) were calculated by solving the eigensystem, $\left[K-\omega^2 M\right]\{\phi\}=\{0\}$. In MATLAB this eigensystem is calculated by [Φ, ω] = eig(K, M) , where Φ is a mass normalized modal matrix comprised of mode shapes ϕ. This then results in $\Phi'M\Phi=\mathcal{M}=[I]$ and $\Phi'K\Phi=[\mathcal{K}]$, diagonalized K matrix or a diagonal matrix of the structures natural

frequencies, $\begin{pmatrix} \ddots & & \\ & \omega^2 & \\ & & \ddots \end{pmatrix}$, and finally $\Phi' C \Phi = [\mathcal{C}] = \begin{pmatrix} \ddots & & \\ & 2\zeta\omega & \\ & & \ddots \end{pmatrix}$, assuming that our physical damping matrix is proportionally damped. Since we are currently working with physical coordinates, for this thesis it is desired to transform the equation of motion from physical to modal coordinates. To proceed with the transformation, we need to utilize the modal matrix Φ, and the transformation equation is $\{x\} = \Phi\{q\}$, taking the first and second derivatives with respect to time *t* give the following $\{\dot{x}\} = \Phi\{\dot{q}\}$ and $\{\ddot{x}\} = \Phi\{\ddot{q}\}$. As a result the equation of motion is now in the form of,

$$[\mathrm{I}]\ddot{q} + \begin{pmatrix} \ddots & & \\ & 2\zeta_n\omega_n & \\ & & \ddots \end{pmatrix}\dot{q} + \begin{pmatrix} \ddots & & \\ & \omega_n^2 & \\ & & \ddots \end{pmatrix} q = Q_i$$

Derivation of Kinetic Energy using modal coordinates

Generally $T = \frac{1}{2}\mathcal{M}\dot{q}^2$, but since $\mathcal{M}$ is equal to [I], it can be simplified as $T = \frac{1}{2}\dot{q}^2$. As a result,

$$T = \frac{1}{2}\sum_{n=1}^{N}\sum_{j=1}^{\infty}\dot{q}_j^{(n)^2}$$

$$\textit{Therefore, } \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) = \sum_{n=1}^{N}\sum_{j=1}^{\infty}\ddot{q}_j^{(n)}$$

where, superscript *n* is the substructure designator and *j* is the mode index

Derivation of Potential Energy using modal coordinates

Generally $U = \frac{1}{2}\mathcal{K}q^2$ As a result,

$$U = \frac{1}{2}\sum_{n=1}^{N}\sum_{j=1}^{\infty} K_j^{(n)} q_j^{(n)^2}$$

since $[K] = \omega^2$

$$U = \frac{1}{2}\sum_{n=1}^{N}\sum_{j=1}^{\infty} \omega_j^{(n)^2} q_j^{(n)^2}$$

$$\textit{Therefore, } \frac{\partial U}{\partial q} = \sum_{n=1}^{N}\sum_{j=1}^{\infty} \omega_j^{(n)^2} q_j^{(n)}$$

Derivation of Damping Dissipation Function using modal coordinates

Generally $D = \frac{1}{2} C \dot{q}^2$ As a result,

$$D = \frac{1}{2}\sum_{n=1}^{N}\sum_{j=1}^{\infty} C_j^{(n)} \dot{q}_j^{(n)^2}$$

since $[C] = [2\zeta\omega]$

$$D = \frac{1}{2}\sum_{n=1}^{N}\sum_{j=1}^{\infty} 2\zeta_j^{(n)} \omega_j^{(n)} \dot{q}_j^{(n)^2}$$

$$\textit{Therefore, } \frac{\partial D}{\partial \dot{q}} = \sum_{n=1}^{N}\sum_{j=1}^{\infty} 2\zeta_j^{(n)} \omega_j^{(n)} \dot{q}_j^{(n)}$$

Substituting the above-derived energies into Lagrange's equation gives the following result,

$\ddot{q}_j^{(n)} + 2\zeta_j^{(n)}\omega_j^{(n)}\dot{q}_j^{(n)} - \omega_j^{(n)^2} q_j^{(n)} = 0$, since $Q_i = 0$, as a result the j[th] mode of Lagrange's equation for the n[th] substructure is,

$$\ddot{q}_j^{(n)} + 2\zeta_j^{(n)}\omega_j^{(n)}\dot{q}_j^{(n)} - \omega_j^{(n)^2} q_j^{(n)} = 0 .$$

B. LAGRANGE MULTIPLIERS AND HOLONOMIC CONSTRAINTS

1. Theory

Lagrange Multipliers (λ) are included in the generalized form of Lagrange's equation to scale each holonomic constraints. Lagrange Multipliers' value is a measure of the force of constraint. In terms of substructure coupling, the Lagrange multiplier is interpreted as a measure of the coupling force between substructures. On the other hand,

holonomic constraints (f), are constraints on a given coordinate system; it serves as an interconnecting condition between the substructures, and has an overall effect of diminishing the total number of independent degrees of freedom (N) to N – r, where r is the number of constraints associated with the interconnecting region. Adding the Lagrange Multipliers to the generalized Lagrange's equation gives the result as shown in Klein [8],

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right)-\frac{\partial U}{\partial q}+\frac{\partial D}{\partial \dot{q}}+\frac{\partial \sum_{r=1}^{R}\lambda_r f_r}{\partial q}=0 \text{ , where } f_r=\sum_{n=1}^{N}\sum_{j=1}^{\infty}\beta_{rj}^{(n)}q_j^{(n)}=0 \text{ and } r=1,...,R \text{ and the}$$

superscript *n* is the substructure designator and *j* is the element of mode shape or modal coordinate vector. This then gives the final expression for the equation of motion in the form

$$\ddot{q}_j^{(n)}+2\zeta_j^{(n)}\omega_j^{(n)}\dot{q}_j^{(n)}+\omega_j^{(n)^2}q_j^{(n)}-\sum_{r=1}^{R}\lambda_r\beta_{rj}^{(n)}=0$$

Equation 1

To better understand the concept of interconnecting region and how it is formulated from substructures, a simple two degree of freedom mass spring system as shown below will be used as an example.

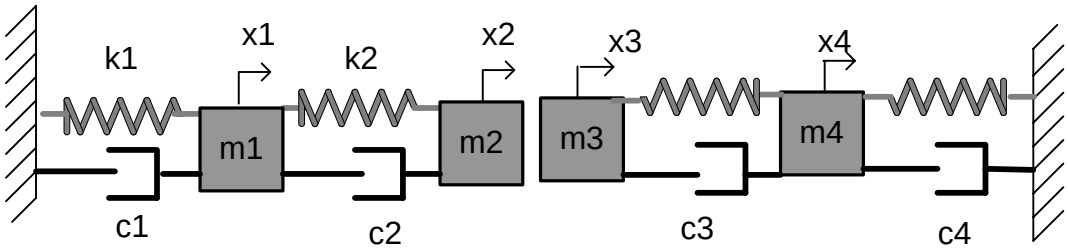

Figure 1. 2 DOF System SubsA (Left), SubsB (Right)

Using physical coordinates to show the connectivity between substructure A and B and knowing that there is only one degree of freedom (r = 1) that is shared by the two substructures after coupling the two; we may now come up with the equation of constraint for the two coordinates to be coupled.

$f_1 = x_2 - x_3$, where x_2 and x_3 are the degrees of freedom of each substructure that are being joined together. Since this thesis mainly works with modal coordinates, we need to transform each of the degrees of freedom in terms of modal coordinate, for example $x_2 = \beta_{1,1}^{(1)} q_1^{(1)} + \beta_{1,2}^{(1)} q_2^{(1)}$, where β is a component of the modal matrix for each substructure; therefore the new equation of interconnecting region becomes

$$f_1 = \beta_{1,1}^{(1)} q_1^{(1)} + \beta_{1,2}^{(1)} q_2^{(1)} - \beta_{1,1}^{(2)} q_1^{(2)} - \beta_{1,2}^{(2)} q_2^{(2)} = 0$$

2. Example of Holonomic Constraint

Two objects each have six coordinates: x, y, z, r, θ, φ, where x, y, and z are the center of mass coordinates, r is the separation between the two objects, θ and φ are their orientation in space. By connecting the two objects with a rigid rod in coordinate r, we are able to decreases the degree of freedom both objects are free to move about by one; therefore, we can say that we placed the two objects under a holonomic constraint.

C. ORTHOGONAL COMPLEMENT AND ZERO-EIGENVALUE THEOREM

1. Theory

In order to understand the concept of orthogonal complement we will first define the four fundamental subspaces that will be covered in the explanation of orthogonal complement. The definitions are extracted from the book "Linear Algebra and its Applications" by Gilbert Strang.

Four Fundamental Subspaces

Starting with an *m* by *n* matrix *A* we must first define the first part of Fundamental Theorem of Linear Algebra Part 1:

1. $\mathbb{R}_n$ (A^T) = row space of *A;* dimension *r*
2. $\mathcal{N}$ (*A)* = nullspace of *A*; dimension *n-r*
3. $\mathbb{R}_m$ (*A)* = column space of *A; dimension r*

4. $\mathcal{N}(A^T)$ = left nullspace of *A;* dimension *m-r*

Applying the first part of the fundamental theorem of Linear Algebra we can now talk about the orthogonal subspace which is a lead in to the concept of the orthogonal complement. The following are definitions from Strang [14] which talks about the relationships between the four subspaces and the relationship to the theory of the orthogonal complement.

1. Two subspaces *V* and *W* of the same space $\mathbb{R}^n$ are called orthogonal if every vector *v* in *V* is orthogonal to every *w* in *W* : $v^T w = 0$ for all *v* and *w.*

2. For any *m* by *n* matrix *A,* the nullspace $\mathcal{N}(A)$ and the row space $\mathcal{R}(A^T)$ are the orthogonal subspaces of $\mathbb{R}^n$. Similarly, the left $\mathcal{N}(A^T)$ and the column space $\mathcal{R}(A)$ are orthogonal subspaces of $\mathbb{R}^m$.

3. Orthogonal complement definition: Given a subspace *V* of $\mathbb{R}^n$, the space of all vectors orthogonal to *V* is called the orthogonal complement of *V*, and denoted by $V^{\perp}$.

Using this terminology, the nullspace $\mathcal{N}(A)$ is the orthogonal complement of $\mathcal{R}(A^T)$: $\mathcal{N}(A) = (\mathcal{R}(A^T))^{\perp}$. At the same time, the opposite relation also holds: The row space $\mathcal{R}(A^T)$ contains all vectors that are orthogonal to the nullspace. This is not so obvious from the construction, since in solving $Ax = 0$ we started with the row space and found all *x* that were orthogonal to it; now we are going in the opposite direction. Suppose, however, that some vector *z* in $\mathbb{R}^n$ is orthogonal to the nullspace but is outside the row space. Then adding *z* as an extra row and *A* would enlarge the row space without changing the null space. But we know that there is a fixed formula: $r+(n-r)=n$, or dim(row space) + dim(nullspace) = number of columns. Since the last two numbers are unchanged when the new row z is added, it is impossible for the first one to change either. We conclude that every vector orthogonal to the nullspace is already in the row space: $\mathcal{R}(A^T) = \mathcal{N}(A)^{\perp}$ (Strang) [14].

The same reasoning applied A^T produces the dual result: *The left nullspace* $\mathcal{N}(A^T)$ *and the column space* $\mathcal{R}(A)$ *are orthogonal complements of one another in* $\mathbb{R}^m$. This completes the second half of the fundamental theorem of linear algebra. The first half gave the dimensions of the four subspaces, including the fact that row rank = column rank, and now we know that they are not only perpendicular; they are orthogonal complements (Strang) [14].

Zero-eigenvalue theorem (Walton and Steeves) [16]: Let E be n x n matrix $A^T A$. If the rank of *E* is m, therefore the eigenvalues of *E* is also m and it has s = n – m "zero eigenvalues". These zero-eigenvalues have corresponding eigenvectors, which is called the nullspace of matrix *E*. Having solved for the set of eigenvectors corresponding to the zero-eigenvalues of matrix E we can call these set of eigenvectors *T*, which is an n x s matrix. Moreover, by post multiplying the equality $E = A^T A$ with the newly generated *T* matrix we will come up with the expression $E\ T = A^T A\ T = 0$ and by premultiplying the said equation with T^T, the equality remained equal to zero as shown by $T^T\ A^T A\ T = 0$; further matrix manipulation will simplify this term to $(AT)^T\ AT = 0$, which finally leads to the fact that matrix *T* is an orthogonal complement of matrix *A* (Kamman) [7].

The two theorems discussed above is one of the central aspects that will be employed in this thesis; the above said matrix T, will be utilized as a transformation matrix of which will enable us to generate coupled system equations of motion given only modal properties of each substructure.

2. Example

MATLAB code was generated to show the usefulness of zero-eigenvalue theorem in substructure coupling using constraints. Using the example from William Walton and Earl Steeves' technical report, the MATLAB code is as follows:

```
clear
clc
%Mass Spring System
%M1==/\/==M2==/\/==M3==/\/==M4==/\/==M5
%Example from Walton and Steeves paper
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
m1 = [1,1,1,1,1];          %mass values for subs1
k1 = [1,1,1,1];            %stiffness values for subs1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

K=zeros(5,5);

%Building of K and M matrices for each substructure;

index = [0 , 1];
for i = 1 : length(k1);
    index = index + 1;
    ka = [k1(1,i) , -k1(1,i) ; -k1(1,i) ,  k1(1,i)];
    K(index,index) = K(index,index) + ka;
end
M = diag(m1);
[Phi,Lam] = eig(K,M);
W = sqrt(Lam)

%Subdivided
Kbar=zeros(8,8);
index = [-1 , 0];
for i = 1 : length(k1);
   index = index + 2;
   ka = [k1(1,i) , -k1(1,i) ; -k1(1,i) ,  k1(1,i)];
   Kbar(index,index) = Kbar(index,index) + ka;
end
Kbar;
m1bar = [1,.5,.5,.5,.5,.5,.5,1];
Mbar = diag(m1bar);

%Equation of Constraint
C = [0,1,-1,0,0,0,0,0; ...
    0,0,0,1,-1,0,0,0; ...
    0,0,0,0,0,1,-1,0];
E = C'*C;

[U,lam] = eig(E);
T = null(E);

%System M and Stiffness matrices
MBAR = T'*Mbar*T
KBAR = T'*Kbar*T

[Phibar,Lambar] = eig(KBAR,MBAR);
Wbar = sqrt(Lambar)
```

```
W =

   0 + 0.0000i        0                  0                  0          0
   0                0.6180               0                  0          0
   0                   0               1.1756                0          0
   0                   0                  0               1.6180        0
   0                   0                  0                  0          1.9021



Wbar =

   0 + 0.0000i        0                  0                  0          0
   0                0.6180               0                  0          0
   0                   0               1.1756                0          0
   0                   0                  0               1.6180        0
   0                   0                  0                  0          1.9021
```

The natural frequencies W (solution using equation of motion for the system) and Wbar (solution from subdividing the system into 5 parts) matches.

D. DECOUPLING APPROXIMATION

Most structures are analyzed via classical damping models, where physical damping matrix C is diagonalized by calculating the triple product $\Phi'C\Phi=[2\zeta_i\omega_i]$ which enables us to extract modal damping ratios for the structure being analyzed. On the other hand, some structures have a dynamic characteristic that is non-classically damped, meaning they are not diagonalized by the mode shapes of the structure; as a result, extracting the modal properties of the structure becomes a tedious task. Structures that exhibit this kind of characteristics are those that come from the coupling of substructures, structures that have different damping characteristics or structures that has highly damped components (Xu) [17]. Fortunately, we may circumvent the problem of having a non-classically damped structure by applying the concept of decoupling approximation, which allows for the decoupling of structures modal matrix by simply ignoring the off-diagonal components of the matrix and using the main diagonal elements to come up with the damping ratios of the substructure. In order for us to implement the decoupling approximation for the structure in hand, we must first satisfy some necessary requirements, the following will explain how we formulate required values such as second-order eigenvalues and eigenvectors and see if they comply with the necessary requirements.

Starting with the general equation of motion $M_o\ddot{x}+C_o\dot{x}+K_ox=0$, as previously shown above we are able to generate a mass normalized modal matrix Φ_o by solving the

eigensystem $\left[K_o - \omega_{jo}^2 M_o\right]\{\Phi_o\} = \{0\}$ in MATLAB. Given Φ_o, and the relationship $x = \Phi_o q$, we can now convert the equation of motion in modal coordinates. Therefore the resulting equation is $I\ddot{q} + \mathcal{C}\dot{q} + \Lambda q = 0$, where I is an identity matrix, $\mathcal{C}$ is the modal damping matrix, which was not diagonalized, and Λ is the decoupled stiffness matrix. The next step is to solve for the second-order eigenvalue problem corresponding to the modal equation of motion derived above. The formulation of the second-order eigenvalue problem is shown in the equation $\left[s_j^2 I + s_j \mathcal{C} + \Lambda\right]\Psi_j = 0$, where s_j and Ψ_j are the system's eigenvalues and eigenvectors. These Ψ_j, of which are mass normalized modal matrix, are then used to obtain the structure's mode shapes Φ_j in the original displacement coordinates x by the relation $\Phi_j = \sum \psi_{ij} \Phi_{io}$, where ψ_j is the ith component of Ψ_j. Moreover, the generated eigenvalues s_j are complex natural frequencies that are comprised of structures natural frequencies ω_j and ζ_j and can be expressed in thee form $s_j = \omega_j\left(-\zeta_j \pm i\sqrt{1-\zeta_j^2}\right)$. To find out whether or not the structure that is currently being worked can be accurately represented by the decoupling approximation, we must see if the generated mode shapes Ψ_j and eigenvalues s_j satisfies the requirement $\psi_j = O(\varepsilon)$ for all values of $i \neq j$ and $s_j \approx s_{jo} = \omega_{jo}\left(-\zeta_{jj} \pm i\sqrt{1-\zeta_{jj}^2}\right)$, where $\zeta_{jj} = \dfrac{\mathcal{C}_{ij}}{\left(\omega_{io} + \omega_{jo}\right)}$ and $\mathcal{C}_{ij}$ is the ijth component of the damping matrix $\mathcal{C}$ and $O(\varepsilon)$ represents a term of the same order of magnitude as ε or smaller, and ε is relatively smaller than one (Xu) [17].

E. GERSCHGORIN'S THEOREMS

1. Theory

The Gerschgorin disk theorem is a theorem which is used to bound eigenvalues of a square matrix. This was shown in the book by Leonard Meirovitch [12] in a more complete detail, the proof is as follows:

"Let us write the eigenvalue problem associated with the *n* x *n* matrix *A* in the form $\sum_{j=1}^{n} a_{kj} x_j = \lambda x_k$, $k = 1, 2, ..., n$ next, let us assume that x_m is a component of the vector **x** with the largest modulus, $|x_m| = \max |x_j|$ $(j = 1, 2, ..., n)$, so that letting $k = m$ in above equation, we can write $(\lambda - a_{mm}) x_m = \sum_{\substack{j=1 \\ j \neq m}}^{n} a_{mj} x_j$ Hence, $|\lambda - a_{mm}| \cdot |x_m| \leq \sum_{\substack{j=1 \\ j \neq m}}^{n} |a_{mj}| \cdot |x_j| \leq |x_m| \cdot \sum_{\substack{j=1 \\ j \neq m}}^{n} |a_{mj}|$ Dividing the inequality through by $|x_m|$, we obtain $|\lambda - a_{mm}| \leq \sum_{\substack{j=1 \\ j \neq m}}^{n} |a_{mj}|$. This inequality is called the first theorem of Gerschgorin and can be stated as follows: Every eigenvalue of the matrix *A* lies in at least one of the circular disks with centers at a_{mm} and radii $r_m = \sum_{\substack{j=1 \\ j \neq m}}^{n} |a_{mj}|$. The disks are sometimes referred to as Gerschgorin's disks." Meirovitch [12]. This theorem seemed useful in coming up with possible bounds for both system and substructure damping but, unfortunately for this theorem to be of much use in estimating modal damping, it is important that the off-diagonal elements of matrix *A* be small relative to the main diagonals, as a result this theorem was not put to use in this thesis.

2. Example

From Meirovitch's book

Consider the matrix $A = \begin{bmatrix} 2.5 & -1 & 0 \\ -1 & 5 & -\sqrt{2} \\ 0 & -\sqrt{2} & 10 \end{bmatrix}$, the centers of Gerschgorin's disk are the diagonals $a_{11} = 2.5,\ \ a_{22} = 5,\ \ a_{33} = 10$ and the corresponding radii are

$$r_1 = |a_{12}| + |a_{13}| = |-1| + |0| = 1$$
$$r_2 = |a_{21}| + |a_{23}| = |-1| + |-\sqrt{2}| = 1 + \sqrt{2},$$
$$r_3 = |a_{31}| + |a_{32}| = |0| + |-\sqrt{2}| = \sqrt{2}$$

The eigenvalues for matrix *A*, are $\lambda_1 = 2.119322,\ \ \lambda_2 = 5,\ \ \lambda_3 = 10.380678$.

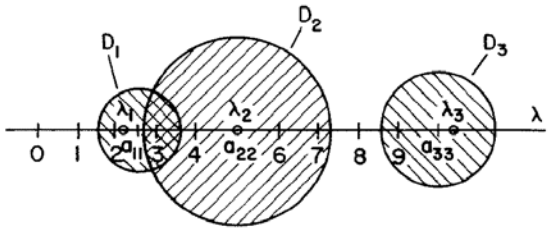

Figure 2. Gerschgorin's Disk shows how each eigenvalues falls within each disk

F. METHODS FOR PRODUCING PROPORTIONALLY DAMPED SUBSTRUCTURES

1. Method A

Modal damping ratio (ζ) is a physical property of a system that is typically measured in a vibration test experiment. When conducting structural coupling we can assume value(s) for a given structure's ζ for each corresponding modes, a typical minimum value used commonly used is around two percent ($\zeta = 0.02$). In doing so, we assume the given structure to be a proportionally damped system.

2. Method B

Typically the way we would come up with proportional damping is by making the physical damping matrix proportional to both mass (M) and stiffness (K) matrices with a given proportionality constants α and β. $C = \alpha K + \beta M$, with this proportionally damped *C* matrix we may now extract damping ratios by first diagonalizing the physical damping matrix *C* by conducting the triple product $\Phi' C \Phi = [2\zeta_i \omega_i]$, where Φ is the modal damping generated from solving the eigensystem consisted of *K* and *M* matrices and by dividing the corresponding values of the diagonal matrix with the corresponding natural frequency multiplied by two in the form, $\zeta_i = \frac{2\zeta_i \omega_i}{2\omega_i}$.

Although we may create proportionally damped substructures by providing values for ζ or extracting ζ from a proportionally damped physical *C* matrix, it is not guaranteed that the final damping matrix *C* will be a proportionally damped one; in fact coupled system in general are those of the non-proportional type Udwadia, F. E. [15].

THIS PAGE INTENTIONALLY LEFT BLANK

III. MILESTONES OF THESIS WORK

A. COUPLING OF MASS SPRING SYSTEM

Synthesizing a simple mass spring system will be used to illustrate the process of constructing system matrix and extracting from it system natural frequencies and modal damping. Figure 1 below is the model that is being synthesized.

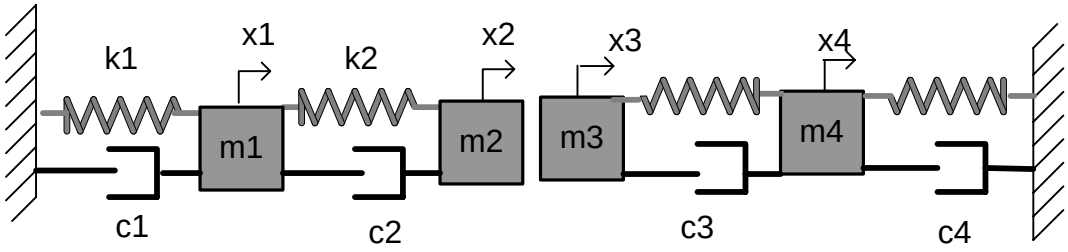

Figure 3. 2 DOF System SubsA (Left), SubsB (Right)

The following will explain the three different methods that were used to calculate system physical properties. Furthermore, coding to come up with physical property values was done in MATLAB and since the method of constructing substructure A (subsA) and substructure B (subsB) matrices are identical, this thesis will focus only on substructure A.

1. Method 1; Basic Structural Coupling

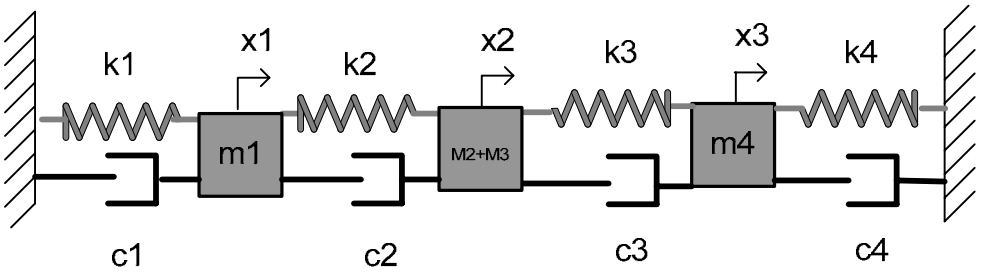

Figure 4. 3 DOF Mass Spring System

SubsA

Initially, physical properties Mass (M), Stiffness (K) and modal damping ratio (ζ) for each mode are given. Using M and K matrices, natural frequencies (ω) and mode shape (Φ) were calculated by solving the eigensystem, $\left[K-\omega^2 M\right]\{\phi\}=\{0\}$. In MATLAB this eigensystem is calculated by $[\Phi, \omega] = \text{eig(K, M)}$, where Φ is a mass normalized modal matrix. This then results in $\Phi'\text{M}\Phi=[\text{I}]$ and $\Phi'\text{K}\Phi=[\mathcal{K}]$, diagonalized K matrix. Since ω is now known, physical damping matrix (C) can now be solved via the equality $\Phi'\text{C}\Phi=[2\zeta\omega]$, where $[2\zeta\omega]$ is a diagonal matrix, this will be a proportionally damped substructure since modal damping is initially given. Finally solving for C, since $\Phi'\text{M}\Phi=[\text{I}]$ and $\Phi'\text{C}\Phi=[2\zeta\omega]$ we solve for the physical damping by conducting the following matrix manipulations,

$\text{C}=\Phi[2\zeta\omega]\Phi'$
since $\text{M}\Phi=\Phi[\text{I}]$; $\text{M}\Phi=\Phi$
and $\Phi'\text{M}=[\text{I}]\Phi'$; $\Phi'\text{M}=\Phi'$
therefore,
$\text{C}=\text{M}\,\Phi[2\zeta\omega]\Phi'\text{M}$

The mass, stiffness and damping matrices, *M, C,* and *K*, for substructure B are calculated in the same way. Having created *Ma, Ka,* and *Ca* for SubsA and *Mb, Kb,* and *Cb* for SubsB, the systems *Ms*, *Ks* and *Cs* can be calculated by adding the terms of the substructure matrix that corresponds to the same position in the system matrix. For example, $\begin{bmatrix} Ma_{11} & Ma_{12} & 0 \\ Ma_{21} & Ma_{22}+Mb_{11} & Mb_{12} \\ 0 & Mb_{21} & Mb_{22} \end{bmatrix}$

Having calculated for system's physical mass, stiffness and damping matrices the next step is to solve the eigensystem $\left[\text{K}_s-\omega^2\text{M}_s\right]\{\Phi_s\}=\{0\}$, where the subscript (s) denotes that we are making use of systems mass and stiffness matrices, to determine the values for ω_s and Φ_s. With these two modal properties in hand we are now be able to solve for system modal damping ratios; in order to do so the triple product $\Phi_s'\text{C}_s\Phi_s$ is

conducted in order to modalize the damping matrix which then enables us to extract said damping ratios from the matrix $\Phi_s' C_s \Phi_s = [2\zeta_s \omega_s]$. As mentioned above in the decoupling approximation, we will ignore the off-diagonal terms of the modal mass matrix when calculating for the damping ratio ζ_s; this assumption was made because we are not guaranteed to have a proportionally damped system via coupling of two sub substructures. (Xu) [17]. Therefore, $\zeta_{ii} = C_{ii} / 2\omega_{ii}$, where i = 1,2,3, to N and ii is the diagonal value for each corresponding system matrix.

2. Method 2; Calculating System's Damping Ratio and Natural Frequencies by Transformation Method Using Zero-Eigenvalue Theorem

In this section structural synthesis is conducted by implementing the Zero-Eigenvalue Theorem. This theorem will provide the necessary transformation matrix that will convert a block matrix of substructures A and B to a synthesized system structure.

Using Lagrange's equation, Lagrange multiplier and holonomic constraints the final equation of motion is in the form $\ddot{q}_j^{(n)} + 2\zeta_j^{(n)}\omega_j^{(n)}\dot{q}_j^{(n)} + \omega_j^{(n)^2} q_j^{(n)} - \sum_{r=1}^{R} \lambda_r \beta_{rj}^{(n)} = 0$ for each substructure.

This example will illustrate how method two is applied to the above mentioned mass spring system. Using the generalized equation

$$\ddot{q}_j^{(n)} + 2\zeta_j^{(n)}\omega_j^{(n)}\dot{q}_j^{(n)} + \omega_j^{(n)^2} q_j^{(n)} - \sum_{r=1}^{R} \lambda_r \beta_{rj}^{(n)} = 0$$

Each substructures equations of motion are as follows

Substructure(*A*)

$$\ddot{q}_1^{(1)} + 2\zeta_1^{(1)}\omega_1^{(1)}\dot{q}_1^{(1)} + \omega_1^{(1)^2} q_1^{(1)} - \lambda_1 \beta_{1,1}^{(1)} = 0$$

$$\ddot{q}_2^{(1)} + 2\zeta_2^{(1)}\omega_2^{(1)}\dot{q}_2^{(1)} + \omega_2^{(1)^2} q_2^{(1)} - \lambda_1 \beta_{1,2}^{(1)} = 0$$

Substructure(*B*)

$$\ddot{q}_1^{(2)} + 2\zeta_1^{(2)}\omega_1^{(2)}\dot{q}_1^{(2)} + \omega_1^{(2)^2} q_1^{(1)} + \lambda_1 \beta_{1,1}^{(2)} = 0$$

$$\ddot{q}_2^{(2)} + 2\zeta_2^{(2)}\omega_2^{(2)}\dot{q}_2^{(2)} + \omega_2^{(2)^2} q_2^{(2)} + \lambda_1 \beta_{1,2}^{(2)} = 0$$

The next step is to put the above equations of motions in the form

$\ddot{q} + 2\zeta\omega\dot{q} + \omega^2 q - [A]^T \lambda = 0$, where A = $\begin{bmatrix} -\beta_{1,1}^{(1)} & -\beta_{1,2}^{(1)} & \beta_{1,1}^{(2)} & \beta_{1,2}^{(2)} \end{bmatrix}$, a matrix of holonomic constraints. The block diagonal consisted of $2\zeta\omega$ shall be called *blockC* and ω^2 as *blockK* for labeling purposes.

$$[I]\begin{Bmatrix} \ddot{q}_1^{(1)} \\ \ddot{q}_2^{(1)} \\ \ddot{q}_1^{(2)} \\ \ddot{q}_2^{(2)} \end{Bmatrix} + \begin{bmatrix} 2\zeta_1^{(1)}\omega_1^{(1)} & 0 & 0 & 0 \\ 0 & 2\zeta_2^{(1)}\omega_2^{(1)} & 0 & 0 \\ 0 & 0 & 2\zeta_1^{(2)}\omega_1^{(2)} & 0 \\ 0 & 0 & 0 & 2\zeta_2^{(2)}\omega_2^{(2)} \end{bmatrix} \begin{Bmatrix} \dot{q}_1^{(1)} \\ \dot{q}_2^{(1)} \\ \dot{q}_1^{(2)} \\ \dot{q}_2^{(2)} \end{Bmatrix} + \begin{bmatrix} \omega_1^{(1)^2} & 0 & 0 & 0 \\ 0 & \omega_2^{(1)^2} & 0 & 0 \\ 0 & 0 & \omega_1^{(2)^2} & 0 \\ 0 & 0 & 0 & \omega_2^{(2)^2} \end{bmatrix} \begin{Bmatrix} q_1^{(1)} \\ q_2^{(1)} \\ q_1^{(2)} \\ q_2^{(2)} \end{Bmatrix} + \begin{bmatrix} -\beta_{1,1}^{(1)} \\ -\beta_{1,2}^{(1)} \\ \beta_{1,1}^{(2)} \\ \beta_{1,2}^{(2)} \end{bmatrix} * \{\lambda_1\} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

Now let $D = A^T A$, and then find the orthogonal complement of *D* matrix by using the "null" function in MATLAB, which gives the basis vectors for the null space of *D* matrix. These basis vectors will now be called matrix *N*, this *N* matrix will be used as a transformation matrix to transform the block diagonal matrices *blockC* and *blockK* to system modal damping matrix and modal stiffness matrix. After the transforming the block matrices to system matrices the systems natural frequencies and damping ratios were then extracted from the system's modal damping and modal stiffness matrices. The following will show the process on how to conduct the transformation.

First, as discussed above, we must extract the block matrices of $2\zeta_j^{(n)}\omega_j^{(n)}$ and $\omega_j^{(n)^2}$ from the above equations of motion; the upper block corresponds to SubsA and the lower block to that of SubsB

$$\begin{bmatrix} 2\zeta_1^{(1)}\omega_1^{(1)} & 0 & 0 & 0 \\ 0 & 2\zeta_2^{(1)}\omega_2^{(1)} & 0 & 0 \\ 0 & 0 & 2\zeta_1^{(2)}\omega_1^{(2)} & 0 \\ 0 & 0 & 0 & 2\zeta_2^{(2)}\omega_2^{(2)} \end{bmatrix} = [blockC] \qquad 4\text{x}4$$

$$\begin{bmatrix} \omega_1^{(1)^2} & 0 & 0 & 0 \\ 0 & \omega_2^{(1)^2} & 0 & 0 \\ 0 & 0 & \omega_1^{(2)^2} & 0 \\ 0 & 0 & 0 & \omega_2^{(2)^2} \end{bmatrix} = [blockK] \quad 4\text{x}4$$

Next, form a triple product to transform matrices *blockC* and *blockK* using matrix N to generate the new matrices *G* and *L* .

$$[G] = N^T [blockC] N \qquad 3\text{x}3$$
$$[L] = N^T [blockK] N \qquad 3\text{x}3$$

With the above given matrices, we can now put it in the first order form by transforming the second order differential equation to first order differential $\begin{Bmatrix} \dot{q}_j^{(n)} \\ \ddot{q}_j^{(n)} \end{Bmatrix} = [E] \begin{Bmatrix} q_j^{(n)} \\ \dot{q}_j^{(n)} \end{Bmatrix}$, where E is equal to the block matrix, $\text{E} = \begin{bmatrix} 0 & \text{I} \\ \text{L} & \text{G} \end{bmatrix}$. The next step is to solve for the complex eigenvalues of matrix E, $\lambda_i = \sigma_i \pm \Omega_{d_i} \text{j}$, where $\sigma_i = \zeta_i \omega_i$ and Ω_{d_i} is the undamped natural frequency of the system. Finally, using the relationships, $\Omega_{d_i} = \omega\sqrt{1-\zeta_i^2}$ and $\sigma_i = \zeta_i \omega_i$, we can now solve for systems modal damping.

The resulting ω and ζ from this method matches the results ω_o and ζ_o of method one above.

3. Method 3 Branches Out of Method 2

Another method that was conducted to get values for ω and ζ branches out from that of the transformation method. In this method, we make use of the already generated $[\text{G}]$ and $[\text{L}]$ matrices. First, system natural frequencies ω and mode shapes Φ are extracted from $[\text{L}]$ by solving for the eigenvalues and corresponding eigenvectors. Upon extraction of $[\text{L}]$'s eigenvectors, the next step is to take the triple product

$\text{New\_G} = \Phi'[G]\Phi$. The matrix New_G will now serve as the modal damping matrix for the system, though it is not a diagonal matrix, since it is not proportionally damped, we are once again making use of the decoupling approximation to calculate the values for the system modal damping. The final step of this method is to individually extract the diagonal values of New_G matrix and divide it with the corresponding system's 2ω, $\zeta_i = \frac{2\zeta_i\omega_i}{2\omega_i}$, by doing so we can make an observation that the solution will indeed be equal to that of the above calculated values from the first two methods.

4. Comparing the Calculation Cost Between Method 2 and 3

Since computational cost is a big issue in solving for numerical problems. We need to investigate how to improve computation time for almost any task that is being done computationally. As a result, for this thesis we need to look at and compare which method, method two or three, would generate a solution in a more economical way. In the two methods discussed above, the main difference in the methodology was the size of the matrix that is being solved. Method two used a matrix of dimension 2*n* by 2*n* in solving for the solution while method three made use of a matrix of size *n* by *n.* Knowing this fact, we further investigated which matrix takes longer to calculate. The first part of the investigation dealt with calculating for the time it takes to compute for the eigenvalues symmetric matrices of varying sizes starting from size *n* by *n* to 16*n* by 16*n*. Using a generated MATLAB code created by Gordis, it is clear that the larger the dimension of a matrix becomes the more time it takes to calculate for the eigensolution. The second part of the investigation called for finding the exact relationship between the cost (time) and the sizes of matrices that are being solved for its eigensolution. Since we are employing MATLAB to solve for the eigenvalue problem, which makes use of QR and QL algorithms, where Q is the orthogonal, R is the upper triangle and L is the lower triangle of any matrix *A*; we can compute the workload of a general matrix to be approximately $O(n^3)$. This then makes method three a better choice in solving for the solution.

B. INITIAL WORK AND THE REASON FOR IMPLEMENTING ZERO EIGENVALUE THEOREM

The main motivation on why this concept is being implemented on this thesis was the fact that it made synthesis of two substructures less tedious in an algebraic stand point. During the initial phase of the thesis work, the method of choice in synthesizing the two substructures was to transform the equations of motion of each substructure from a second order differential equation into a first order differential, $\begin{Bmatrix} \dot{q} \\ \ddot{q} \end{Bmatrix} = [A] \begin{Bmatrix} q \\ \dot{q} \end{Bmatrix}$ where $[A]$ is the system matrix which will be used to calculate for system's dynamic response . To generate this system matrix we must first consider the equations of motion from each substructure that was generated via Lagrange's equations. Using the same mass spring system that was mentioned above and its equation of motions

$$\ddot{q}_1^{(1)} + 2\zeta_1^{(1)}\omega_1^{(1)}\dot{q}_1^{(1)} + \omega_1^{(1)^2} q_1^{(1)} - \lambda_1\beta_{1,1}^{(1)} = 0$$
$$\ddot{q}_2^{(1)} + 2\zeta_2^{(1)}\omega_2^{(1)}\dot{q}_2^{(1)} + \omega_2^{(1)^2} q_2^{(1)} - \lambda_1\beta_{1,2}^{(1)} = 0$$
$$\ddot{q}_1^{(2)} + 2\zeta_1^{(2)}\omega_1^{(2)}\dot{q}_1^{(2)} + \omega_1^{(2)^2} q_1^{(2)} + \lambda_1\beta_{1,1}^{(2)} = 0$$
$$\ddot{q}_2^{(2)} + 2\zeta_2^{(2)}\omega_2^{(2)}\dot{q}_2^{(2)} + \omega_2^{(2)^2} q_2^{(2)} + \lambda_1\beta_{1,2}^{(2)} = 0$$

then solving for $\ddot{q}_j^{(n)}$ gives the following equations

$$\ddot{q}_1^{(1)} = -2\zeta_1^{(1)}\omega_1^{(1)}\dot{q}_1^{(1)} - \omega_1^{(1)^2} q_1^{(1)} + \lambda_1\beta_{1,1}^{(1)} \quad (EOM1)$$
$$\ddot{q}_2^{(1)} = -2\zeta_2^{(1)}\omega_2^{(1)}\dot{q}_2^{(1)} - \omega_2^{(1)^2} q_2^{(1)} + \lambda_1\beta_{1,2}^{(1)} \quad (EOM2)$$
$$\ddot{q}_1^{(2)} = -2\zeta_1^{(2)}\omega_1^{(2)}\dot{q}_1^{(2)} - \omega_1^{(2)^2} q_1^{(2)} - \lambda_1\beta_{1,1}^{(2)} \quad (EOM3)$$
$$\ddot{q}_2^{(2)} = -2\zeta_2^{(2)}\omega_2^{(2)}\dot{q}_2^{(2)} - \omega_2^{(2)^2} q_2^{(2)} - \lambda_1\beta_{1,2}^{(2)} \quad (EOM4)$$

Using the interconnecting conditions between the two substructure components

$f_r = \sum_{n=1}^{N}\sum_{j=1}^{\infty}\beta_{rj}^{(n)} q_j^{(n)} = 0$ *and* $r = 1,...,R$, since r = 1, the explicit form is

$$f_1 = \beta_{1,1}^{(1)} q_1^{(1)} + \beta_{1,2}^{(1)} q_2^{(1)} - \beta_{1,1}^{(2)} q_1^{(2)} - \beta_{1,2}^{(2)} q_2^{(2)} = 0$$

The next step is to take the first and second derivatives of the constraint equation with respect to the modal coordinate q which results in the following equations

$$f_1 = \beta_{1,1}^{(1)} q_1^{(1)} + \beta_{1,2}^{(1)} q_2^{(1)} - \beta_{1,1}^{(2)} q_1^{(2)} - \beta_{1,2}^{(2)} q_2^{(2)} = 0$$
$$\dot{f}_1 = \beta_{1,1}^{(1)} \dot{q}_1^{(1)} + \beta_{1,2}^{(1)} \dot{q}_2^{(1)} - \beta_{1,1}^{(2)} \dot{q}_1^{(2)} - \beta_{1,2}^{(2)} \dot{q}_2^{(2)} = 0$$
$$\ddot{f}_1 = \beta_{1,1}^{(1)} \ddot{q}_1^{(1)} + \beta_{1,2}^{(1)} \ddot{q}_2^{(1)} - \beta_{1,1}^{(2)} \ddot{q}_1^{(2)} - \beta_{1,2}^{(2)} \ddot{q}_2^{(2)} = 0$$

Then substitute in the corresponding values of $\ddot{q}_j^{(n)}$ from the equations of motion above and then solve for the Lagrange multiplier λ, stepping through the process:

$$\beta_{1,1}^{(1)} \ddot{q}_1^{(1)} + \beta_{1,2}^{(1)} \ddot{q}_2^{(1)} - \beta_{1,1}^{(2)} \ddot{q}_1^{(2)} - \beta_{1,2}^{(2)} \ddot{q}_2^{(2)} = 0$$
$$\Rightarrow \textit{Substitute}$$
$$\beta_{1,1}^{(1)} \left(-2\zeta_1^{(1)} \omega_1^{(1)} \dot{q}_1^{(1)} - \omega_1^{(1)^2} q_1^{(1)} + \lambda_1 \beta_{1,1}^{(1)} \right) + \beta_{1,2}^{(1)} \left(-2\zeta_2^{(1)} \omega_2^{(1)} \dot{q}_2^{(1)} - \omega_2^{(1)^2} q_2^{(1)} + \lambda_1 \beta_{1,2}^{(1)} \right) ....$$
$$-\beta_{1,1}^{(2)} \left(-2\zeta_1^{(2)} \omega_1^{(2)} \dot{q}_1^{(2)} - \omega_1^{(2)^2} q_1^{(2)} - \lambda_1 \beta_{1,1}^{(2)} \right) - \beta_{1,2}^{(2)} \left(-2\zeta_2^{(2)} \omega_2^{(2)} \dot{q}_2^{(2)} - \omega_2^{(2)^2} q_2^{(2)} - \lambda_1 \beta_{1,2}^{(2)} \right) = 0$$

$$\Rightarrow \textit{Distribute}$$
$$-\beta_{1,1}^{(1)} 2\zeta_1^{(1)} \omega_1^{(1)} \dot{q}_1^{(1)} - \beta_{1,1}^{(1)} \omega_1^{(1)^2} q_1^{(1)} + \lambda_1 \beta_{1,1}^{(1)^2} - \beta_{1,2}^{(1)} 2\zeta_2^{(1)} \omega_2^{(1)} \dot{q}_2^{(1)} - \beta_{1,2}^{(1)} \omega_2^{(1)^2} q_2^{(1)} + \lambda_1 \beta_{1,2}^{(1)^2} ....$$
$$+\beta_{1,1}^{(2)} 2\zeta_1^{(2)} \omega_1^{(2)} \dot{q}_1^{(2)} + \beta_{1,1}^{(2)} \omega_1^{(2)^2} q_1^{(2)} + \lambda_1 \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)} 2\zeta_2^{(2)} \omega_2^{(2)} \dot{q}_2^{(2)} + \beta_{1,2}^{(2)} \omega_2^{(2)^2} q_2^{(2)} + \lambda_1 \beta_{1,2}^{(2)^2} = 0$$

$$\Rightarrow \textit{Solve}(\lambda)$$
$$\lambda_1 \left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2} \right) = \beta_{1,1}^{(1)} 2\zeta_1^{(1)} \omega_1^{(1)} \dot{q}_1^{(1)} + \beta_{1,1}^{(1)} \omega_1^{(1)^2} q_1^{(1)} + \beta_{1,2}^{(1)} 2\zeta_2^{(1)} \omega_2^{(1)} \dot{q}_2^{(1)} .....$$
$$+\beta_{1,2}^{(1)} \omega_2^{(1)^2} q_2^{(1)} - \beta_{1,1}^{(2)} 2\zeta_1^{(2)} \omega_1^{(2)} \dot{q}_1^{(2)} - \beta_{1,1}^{(2)} \omega_1^{(2)^2} q_1^{(2)} - \beta_{1,2}^{(2)} 2\zeta_2^{(2)} \omega_2^{(2)} \dot{q}_2^{(2)} - \beta_{1,2}^{(2)} \omega_2^{(2)^2} q_2^{(2)}$$

$$\lambda_1 = \frac{\beta_{1,1}^{(1)} 2\zeta_1^{(1)} \omega_1^{(1)} \dot{q}_1^{(1)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2} \right)} + \frac{\beta_{1,1}^{(1)} \omega_1^{(1)^2} q_1^{(1)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2} \right)} + \frac{\beta_{1,2}^{(1)} 2\zeta_2^{(1)} \omega_2^{(1)} \dot{q}_2^{(1)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2} \right)} ....$$
$$+\frac{\beta_{1,2}^{(1)} \omega_2^{(1)^2} q_2^{(1)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2} \right)} - \frac{\beta_{1,1}^{(2)} 2\zeta_1^{(2)} \omega_1^{(2)} \dot{q}_1^{(2)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2} \right)} - \frac{\beta_{1,1}^{(2)} \omega_1^{(2)^2} q_1^{(2)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2} \right)} ....$$
$$-\frac{\beta_{1,2}^{(2)} 2\zeta_2^{(2)} \omega_2^{(2)} \dot{q}_2^{(2)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2} \right)} - \frac{\beta_{1,2}^{(2)} \omega_2^{(2)^2} q_2^{(2)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2} \right)}$$

Re-substituting λ_1 back to equations of motion 1 through 4 (EOM 1 – 4)

$$\ddot{q}_1^{(1)} = \left[\frac{\beta_{1,1}^{(1)^2} 2\zeta_1^{(1)}\omega_1^{(1)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} - 2\zeta_1^{(1)}\omega_1^{(1)} \right] \dot{q}_1^{(1)} + \left[\frac{\beta_{1,1}^{(1)^2}\omega_1^{(1)^2}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} - \omega_1^{(1)^2} \right] q_1^{(1)} .....$$

$$+ \left[\frac{\beta_{1,1}^{(1)}\beta_{1,2}^{(1)} 2\zeta_2^{(1)}\omega_2^{(1)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] \dot{q}_2^{(1)} + \left[\frac{\beta_{1,1}^{(1)}\beta_{1,2}^{(1)}\omega_2^{(1)^2}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] q_2^{(1)} .....$$

$$- \left[\frac{\beta_{1,1}^{(1)}\beta_{1,1}^{(2)} 2\zeta_1^{(2)}\omega_1^{(2)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] \dot{q}_1^{(2)} - \left[\frac{\beta_{1,1}^{(1)}\beta_{1,1}^{(2)}\omega_1^{(2)^2}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] q_1^{(2)} ....$$

$$- \left[\frac{\beta_{1,1}^{(1)}\beta_{1,2}^{(2)} 2\zeta_2^{(2)}\omega_2^{(2)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] \dot{q}_2^{(2)} - \left[\frac{\beta_{1,1}^{(1)}\beta_{1,2}^{(2)}\omega_2^{(2)^2}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] q_2^{(2)}$$

$$\ddot{q}_2^{(1)} = \left[\frac{\beta_{1,2}^{(1)}\beta_{1,1}^{(1)} 2\zeta_1^{(1)}\omega_1^{(1)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] \dot{q}_1^{(1)} + \left[\frac{\beta_{1,2}^{(1)}\beta_{1,1}^{(1)}\omega_1^{(1)^2}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] q_1^{(1)} .....$$

$$+ \left[\frac{\beta_{1,2}^{(1)^2} 2\zeta_2^{(1)}\omega_2^{(1)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} - 2\zeta_2^{(1)}\omega_2^{(1)} \right] \dot{q}_2^{(1)} + \left[\frac{\beta_{1,2}^{(1)^2}\omega_2^{(1)^2}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} - \omega_2^{(1)^2} \right] q_2^{(1)} .....$$

$$- \left[\frac{\beta_{1,2}^{(1)}\beta_{1,1}^{(2)} 2\zeta_1^{(2)}\omega_1^{(2)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] \dot{q}_1^{(2)} - \left[\frac{\beta_{1,2}^{(1)}\beta_{1,1}^{(2)}\omega_1^{(2)^2}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] q_1^{(2)} ....$$

$$- \left[\frac{\beta_{1,2}^{(1)}\beta_{1,2}^{(2)} 2\zeta_2^{(2)}\omega_2^{(2)}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] \dot{q}_2^{(2)} - \left[\frac{\beta_{1,2}^{(1)}\beta_{1,2}^{(2)}\omega_2^{(2)^2}}{\left(\beta_{1,1}^{(1)^2} + \beta_{1,2}^{(1)^2} + \beta_{1,1}^{(2)^2} + \beta_{1,2}^{(2)^2}\right)} \right] q_2^{(2)}$$

$$\ddot{q}_1^{(2)} = -\left[\frac{\beta_{1,1}^{(2)}\beta_{1,1}^{(1)}2\zeta_1^{(1)}\omega_1^{(1)}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)}\right]\dot{q}_1^{(1)} - \left[\frac{\beta_{1,1}^{(2)}\beta_{1,1}^{(1)}\omega_1^{(1)^2}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)}\right]q_1^{(1)}.....$$

$$-\left[\frac{\beta_{1,1}^{(2)}\beta_{1,2}^{(1)}2\zeta_2^{(1)}\omega_2^{(1)}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)}\right]\dot{q}_2^{(1)} - \left[\frac{\beta_{1,1}^{(2)}\beta_{1,2}^{(1)}\omega_2^{(1)^2}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)}\right]q_2^{(1)}.....$$

$$+\left[\frac{\beta_{1,1}^{(2)^2}2\zeta_1^{(2)}\omega_1^{(2)}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)} - 2\zeta_1^{(2)}\omega_1^{(2)}\right]\dot{q}_1^{(2)} + \left[\frac{\beta_{1,1}^{(2)^2}\omega_1^{(2)^2}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)} - \omega_1^{(2)^2}\right]q_1^{(2)}....$$

$$+\left[\frac{\beta_{1,1}^{(2)}\beta_{1,2}^{(2)}2\zeta_2^{(2)}\omega_2^{(2)}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)}\right]\dot{q}_2^{(2)} + \left[\frac{\beta_{1,1}^{(2)}\beta_{1,2}^{(2)}\omega_2^{(2)^2}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)}\right]q_2^{(2)}$$

$$\ddot{q}_2^{(2)} = -\left[\frac{\beta_{1,2}^{(2)}\beta_{1,1}^{(1)}2\zeta_1^{(1)}\omega_1^{(1)}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)}\right]\dot{q}_1^{(1)} - \left[\frac{\beta_{1,2}^{(2)}\beta_{1,1}^{(1)}\omega_1^{(1)^2}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)}\right]q_1^{(1)}.....$$

$$-\left[\frac{\beta_{1,2}^{(2)}\beta_{1,2}^{(1)}2\zeta_2^{(1)}\omega_2^{(1)}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)}\right]\dot{q}_2^{(1)} - \left[\frac{\beta_{1,2}^{(2)}\beta_{1,2}^{(1)}\omega_2^{(1)^2}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)}\right]q_2^{(1)}.....$$

$$+\left[\frac{\beta_{1,2}^{(2)}\beta_{1,1}^{(2)}2\zeta_1^{(2)}\omega_1^{(2)}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)} - 2\zeta_1^{(2)}\omega_1^{(2)}\right]\dot{q}_1^{(2)} + \left[\frac{\beta_{1,2}^{(2)}\beta_{1,1}^{(2)}\omega_1^{(2)^2}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)} - \omega_1^{(2)^2}\right]q_1^{(2)}....$$

$$+\left[\frac{\beta_{1,2}^{(2)^2}2\zeta_2^{(2)}\omega_2^{(2)}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)} - 2\zeta_2^{(2)}\omega_2^{(2)}\right]\dot{q}_2^{(2)} + \left[\frac{\beta_{1,2}^{(2)^2}\omega_2^{(2)^2}}{\left(\beta_{1,1}^{(1)^2}+\beta_{1,2}^{(1)^2}+\beta_{1,1}^{(2)^2}+\beta_{1,2}^{(2)^2}\right)} - \omega_2^{(2)^2}\right]q_2^{(2)}$$

Finally, assemble the system matrix *A*,

$$\begin{Bmatrix} \dot{q}_1^{(1)} \\ \dot{q}_2^{(1)} \\ \dot{q}_1^{(2)} \\ \dot{q}_2^{(2)} \\ \ddot{q}_1^{(1)} \\ \ddot{q}_2^{(1)} \\ \ddot{q}_1^{(2)} \\ \ddot{q}_2^{(2)} \end{Bmatrix} = [A] \begin{Bmatrix} q_1^{(1)} \\ q_2^{(1)} \\ q_1^{(2)} \\ q_2^{(2)} \\ \dot{q}_1^{(1)} \\ \dot{q}_2^{(1)} \\ \dot{q}_1^{(2)} \\ \dot{q}_2^{(2)} \end{Bmatrix}$$

As we can see with only one interconnecting region the calculation process takes an exorbitant amount of time. As a result the process of using zero-eigenvalue proved to be a highly beneficial method of synthesizing substructures.

C. CONSTRAINT ON THE SYSTEM MODAL DAMPING

The method of substructure coupling was put in place in order to determine what some of the possible bounds are for system damping. With this concept in mind, we may now focus on the aspect of predicting the possible bounds for sub structural damping that would suffice the necessary damping ratio for the system.

Knowing what the desired range of system damping should be, of which we can refer to as Zeta_max and Zeta_min, and having calculated systems response from substructure mass and stiffness matrices from either methods two or three mentioned above; we may now proceed in calculating for the main motivation of this thesis which are the necessary bounds for sub structural damping that would not violate the necessary bounds for system damping. This task can be achieved by slightly modifying the above methodology of substructure coupling. The whole process is explained below:

1. First, by choosing either method two or three above, we must calculate substructure response, given initial values for substructure mass and stiffness matrices. As mentioned before, by solving the

eigenvalue problem $\left[K-\omega^2 M\right]\{\phi\}=\{0\}$, we will come up with the substructure's natural frequencies (ω) and mode shapes(Φ).

2. Given substructure mode shapes, Φ_A and Φ_B we are now able to extract from these mode shapes the values for(β), which are elements of each mode shape that corresponds to each interconnecting points for coupling the substructures, to help formulate the constraint matrix$[A]$.

3. This constraint matrix will then be utilized in forming the transformation matrix using zero-eigenvalue theorem. This is shown by first solving for $D=A^T A$, then by taking the orthogonal complement of *D,* we can generate matrix *N* which is consisted of the basis vectors for the null space of matrix *D*. This will then allow for the calculation of system's natural frequencies via transformation of *blockK,* which is a block diagonal matrix consisted of substructure's natural frequencies squared (ω^2). The output from this transformation is exactly what matrix *L* is from method three above, with this matrix we can come up with systems natural frequencies and mode shapes by solving for the eigenvalues and eigenvectors of matrix *L* in MATLAB.

4. After generating system natural frequencies ω_s and mode shapes Φ_s, and prescribing values for system modal damping which falls between desired values Zeta_max and Zeta_min, we can come up with system's physical damping matrix, via the equation $C_s = \Phi_s[2\zeta_s\omega_s]\Phi_s{}^T$

5. Given physical damping matrix C_s and the previously calculated transformation matrix *N* we can decompose this system matrix into

two matrices consisted of block diagonal *blockC* consisted of $2\zeta\omega$ for each substructure.

6. Finally, since the natural frequencies and mode shapes were calculated early on for each substructure, extraction of each substructures modal damping can be performed via $\zeta_i = \frac{2\zeta_i\omega_i}{2\omega_i}$. This then enables us to determine the bounds for each substructure damping that would satisfy the desired Zeta_min and Zeta_max.

D. OPTIMIZATION

1. Work

Now that we have an idea what the necessary bounds are for substructure damping given a constraint on system damping. In this portion of the thesis we are tasked to implement the method of optimization to ease the process in calculating for the values for sub structural damping, bound by a lower and upper limit, which would satisfy the given constraints for system damping for all modes.

Instead of formulating our own code, a built in MATLAB function FMINCON is put to use in finding these substructure damping. In MATLAB, fmincon is a built in function which aids in finding a constrained minimum of a given objective function of several design variables starting at an initial estimate. This is generally referred to as constrained nonlinear optimization or nonlinear programming.

Fmincon will minimize the objective function which is a summation of all substructure damping rations, the design variables (DV), $f(\zeta)=\sum_{i=1}^{N}\zeta_i$, subject to a lower and upper bound for the solution (DV) and a nonlinear constraint equation which then provides upper and lower bounds for system damping.

E. EXAMPLES OF WORK THAT WAS DONE

1. Mass Spring System

All of the work done with this type of system was discussed above.

2. Beam Structure

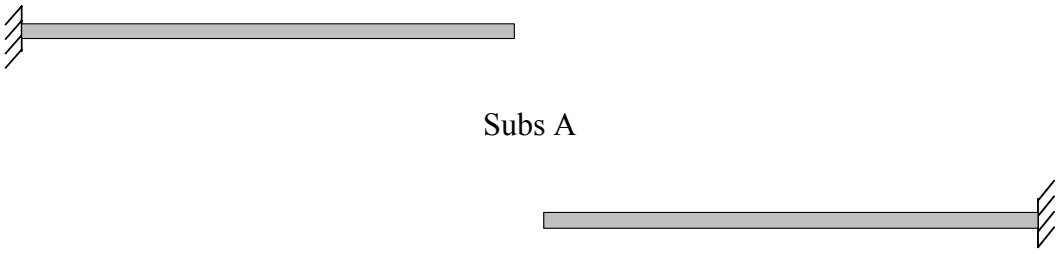

Figure 5. Cantilever Beam

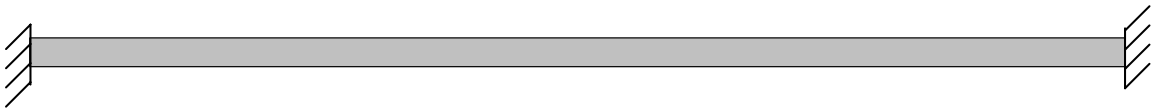

Figure 6. Synthesized Cantilever Beam

The process was conducted in the same fashion as the mass spring system. The major differences are the number of holonomic constraints and our ability to choose how many modes to keep when coupling the system from substructure. The upcoming tables will illustrate the effects of keeping only the first few modes of each substructure to calculate for the dynamic response of the system and compare that to a system solution check that was generated using method one which was the basic method of coupling.

3. Plate Structure

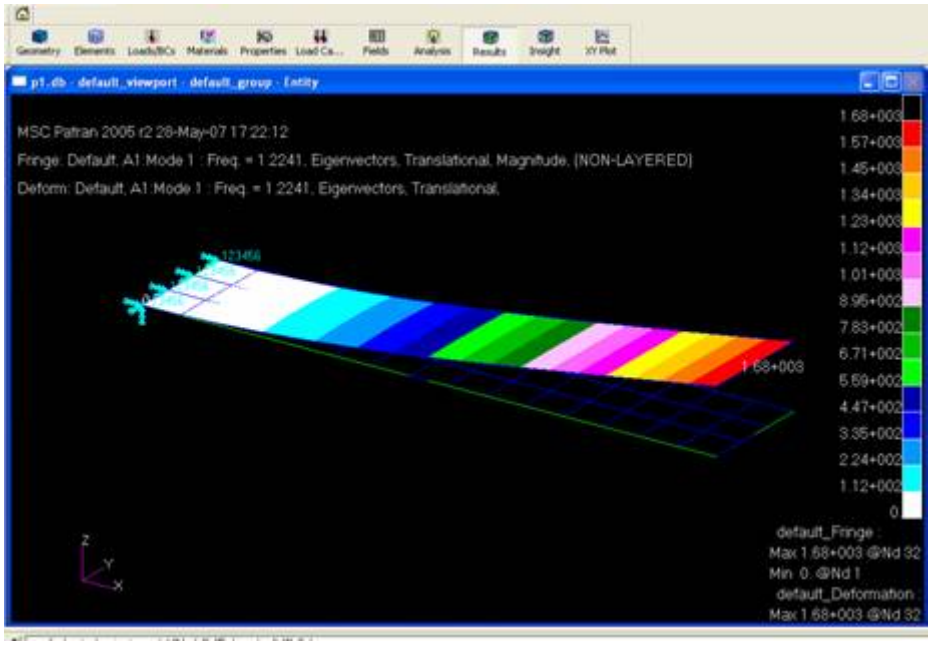

Figure 7. Substructure A Mode 1

In this section Patran and Nastran was put to use in producing natural frequencies and mode shapes for two plate structures. The results from Nastran where then extracted and transferred over to MATLAB where we can make further use of the results to aid couple the two plate structures. As in the mass spring system we make use of the interconnected elements of each substructure to create the transformation matrix that is required to create the system. Once the transformation matrix has been created we can now assume random values for substructure modal damping and proceed with the sequence of events of method two to reproduce system properties such as mode shape, natural frequencies and modal damping. Furthermore, by creating a finite element model of the system in Patran and Nastran, we are able to compare the values of ω from method two, to the solution of the finite element; this then gave us further assurance that the method of substructure coupling was indeed correct. The main motivation in utilizing commercial software was to prevent having to create a program in MATLAB that would generate a Finite Element Model of plate structures which would have been very time consuming.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS

Nomenclature for Tables

ζ_o ub = upper bound for initial damping for each substructure

ζ_o lb = lower bound for initial damping for each substructure

ζ_{subs} ub = upper bound for substructure damping

ζ_{subs} lb = lower bound for substructure damping

ζ_{subs} max = max value of substructure damping that satisfies both substructure and system upper bounds

ζ_{subs} min = min value of substructure damping that satisfies both substructure and system upper bounds

ζ_{sys} ub = upper bound for desired system damping

ζ_{sys} lb = lower bound for desired system damping

ζ_{sys} max = max value of system damping that satisfies system upper bound

ζ_{sys} min = min value of system damping that satisfies system upper bound

A. TABLES

Check Wn	Check Zeta	Method 3 Wn	Method 3 Zeta	Error Wn Check Vs Method 3	Error Zeta Check Vs Method 3
14.0032	0.05	14.00315	0.05	0.00%	0.00%
38.6003	0.04967	38.60027	0.0497	0.00%	0.00%
75.6722	0.05	75.67218	0.05	0.00%	0.00%
125.091	0.04965	125.0908	0.0497	0.00%	0.00%
186.867	0.05	186.8666	0.05	0.00%	0.00%
261.001	0.04983	261.0014	0.0498	0.00%	0.00%
347.5	0.05	347.4996	0.05	0.00%	0.00%
446.368	0.04989	446.3678	0.0499	0.00%	0.00%
557.616	0.05	557.6159	0.05	0.00%	0.00%
681.258	0.04992	681.2576	0.0499	0.00%	0.00%
817.312	0.05	817.3119	0.05	0.00%	0.00%
965.804	0.04994	965.8035	0.0499	0.00%	0.00%
1126.76	0.05	1126.764	0.05	0.00%	0.00%
1300.23	0.04994	1300.234	0.0499	0.00%	0.00%
1486.26	0.05	1486.261	0.05	0.00%	0.00%
1684.91	0.04995	1684.906	0.0499	0.00%	0.00%
1896.24	0.05	1896.24	0.05	0.00%	0.00%
2120.35	0.04995	2120.346	0.0499	0.00%	0.00%
2357.32	0.05	2357.322	0.05	0.00%	0.00%
2607.28	0.04995	2607.28	0.0499	0.00%	0.00%
2870.35	0.05	2870.348	0.05	0.00%	0.00%
3146.67	0.04995	3146.672	0.0499	0.00%	0.00%
3436.41	0.05	3436.412	0.05	0.00%	0.00%
3739.75	0.04995	3739.747	0.0499	0.00%	0.00%
4056.87	0.05	4056.873	0.05	0.00%	0.00%
4388	0.04995	4387.999	0.05	0.00%	0.00%
4733.35	0.05	4733.348	0.05	0.00%	0.00%
5093.15	0.04996	5093.151	0.05	0.00%	0.00%
5467.64	0.05	5467.636	0.05	0.00%	0.00%
5857.02	0.04997	5857.019	0.05	0.00%	0.00%
6261.47	0.05	6261.473	0.05	0.00%	0.00%
6681.09	0.05	6681.085	0.05	0.00%	0.00%
7115.77	0.05	7115.767	0.05	0.00%	0.00%
7565.08	0.05	7565.077	0.05	0.00%	0.00%
8027.82	0.04994	8027.819	0.0499	0.00%	0.00%
8501.04	0.05	8501.042	0.05	0.00%	0.00%
8977.19	0.0498	8977.192	0.0498	0.00%	0.00%
9434.6	0.05	9434.603	0.05	0.00%	0.00%
9806.41	0.04955	9806.414	0.0496	0.00%	0.00%
11155.6	0.04907	11155.64	0.0491	0.00%	0.00%
11598.2	0.05	11598.25	0.05	0.00%	0.00%
12153.2	0.04934	12153.21	0.0493	0.00%	0.00%
12764.6	0.05	12764.62	0.05	0.00%	0.00%
13415.1	0.04946	13415.1	0.0495	0.00%	0.00%
14098.8	0.05	14098.79	0.05	0.00%	0.00%
14813.7	0.0495	14813.73	0.0495	0.00%	0.00%
15559.5	0.05	15559.54	0.05	0.00%	0.00%
16336.6	0.04951	16336.58	0.0495	0.00%	0.00%
17145.6	0.05	17145.62	0.05	0.00%	0.00%
17987.7	0.04949	17987.65	0.0495	0.00%	0.00%
18863.8	0.05	18863.84	0.05	0.00%	0.00%
19775.4	0.04945	19775.41	0.0495	0.00%	0.00%
20723.7	0.05	20723.67	0.05	0.00%	0.00%
21709.9	0.0494	21709.92	0.0494	0.00%	0.00%
22735.4	0.05	22735.44	0.05	0.00%	0.00%

23801.4	0.04932	23801.43	0.0493	0.00%	0.00%
24908.9	0.05	24908.95	0.05	0.00%	0.00%
26058.8	0.04922	26058.84	0.0492	0.00%	0.00%
27251.6	0.05	27251.65	0.05	0.00%	0.00%
28487.5	0.04911	28487.47	0.0491	0.00%	0.00%
29765.9	0.05	29765.85	0.05	0.00%	0.00%
31085.6	0.04898	31085.57	0.049	0.00%	0.00%
32444.4	0.05	32444.44	0.05	0.00%	0.00%
33839.1	0.04885	33839.07	0.0489	0.00%	0.00%
35264.6	0.05	35264.56	0.05	0.00%	0.00%
36714.2	0.04879	36714.16	0.0488	0.00%	0.00%
38179	0.05	38179	0.05	0.00%	0.00%
39647.7	0.04893	39647.66	0.0489	0.00%	0.00%
41106	0.05	41105.96	0.05	0.00%	0.00%
42536.7	0.04955	42536.7	0.0496	0.00%	0.00%
43919.6	0.05	43919.6	0.05	0.00%	0.00%
45231.5	0.05	45231.53	0.05	0.00%	0.00%
46447	0.04863	46446.97	0.0486	0.00%	0.00%
47538.9	0.05	47538.86	0.05	0.00%	0.00%
48479.8	0.04443	48479.83	0.0444	0.00%	0.00%
49243.8	0.04931	49243.85	0.0493	0.00%	0.00%
49808	0.0347	49807.96	0.0347	0.00%	0.00%
50154.1	0.05	50154.14	0.05	0.00%	0.00%

Table 1. 20 Element Per Substructure All modes Kept

Check Wn	Check Zeta	Method 3 Wn	Method 3 Zeta	Error Wn Check Vs Method 3	Error Zeta Check Vs Method 3
14.0032	0.05	14.09988	0.05	0.69%	0.00%
38.6003	0.04967	38.6004	0.0497	0.00%	0.12%
75.6722	0.05	76.3976	0.05	0.96%	0.00%
125.091	0.04965	125.0923	0.0497	0.00%	0.06%
186.867	0.05	188.6633	0.05	0.96%	0.00%
261.001	0.04983	261.0077	0.0498	0.00%	0.03%
347.5	0.05	350.8928	0.05	0.98%	0.00%
446.368	0.04989	446.3863	0.0499	0.00%	0.02%
557.616	0.05	563.1438	0.05	0.99%	0.00%
681.258	0.04992	681.3006	0.0499	0.01%	0.01%
817.312	0.05	825.5409	0.05	1.01%	0.00%
965.804	0.04994	965.89	0.0499	0.01%	0.01%
1126.76	0.05	1138.291	0.05	1.02%	0.00%
1300.23	0.04994	1300.39	0.05	0.01%	0.01%
1486.26	0.05	1501.721	0.05	1.04%	0.00%
1684.91	0.04995	1685.169	0.05	0.02%	0.02%
1896.24	0.05	1916.312	0.05	1.06%	0.00%
2120.35	0.04995	2120.763	0.05	0.02%	0.03%
2357.32	0.05	2382.742	0.05	1.08%	0.00%
2607.28	0.04995	2607.91	0.05	0.02%	0.04%
2870.35	0.05	2901.926	0.05	1.10%	0.00%
3146.67	0.04995	3147.59	0.05	0.03%	0.06%
3436.41	0.05	3475.054	0.05	1.12%	0.00%
3739.75	0.04995	3741.042	0.05	0.03%	0.08%
4056.87	0.05	4103.614	0.05	1.15%	0.00%
4388	0.04995	4389.776	0.05	0.04%	0.10%
4733.35	0.05	4789.407	0.05	1.18%	0.01%
5093.15	0.04996	5095.533	0.05	0.05%	0.09%
5467.64	0.05	5534.501	0.05	1.22%	0.04%
5857.02	0.04997	5860.137	0.05	0.05%	0.06%
6261.47	0.05	6341.06	0.05	1.27%	0.10%
6681.09	0.05	6685.061	0.05	0.06%	0.00%
7115.77	0.05	7210.761	0.0499	1.33%	0.18%
7565.08	0.05	7569.964	0.05	0.06%	0.00%
8027.82	0.04994	8142.548	0.0498	1.43%	0.20%
8501.04	0.05	8506.581	0.05	0.07%	0.00%
8977.19	0.0498	9120.192	0.0497	1.59%	0.24%
9434.6	0.05	9439.023	0.05	0.05%	0.00%
9806.41	0.04955	10043.56	0.0489	2.42%	1.41%
11155.6	0.04907	11174.97	0.0493	0.17%	0.52%
11598.2	0.05	11685.07	0.05	0.75%	0.00%
12153.2	0.04934	12173.57	0.0496	0.17%	0.47%
12764.6	0.05	12906.03	0.05	1.11%	0.00%
13415.1	0.04946	13438.44	0.0497	0.17%	0.50%
14098.8	0.05	14279.65	0.05	1.28%	0.00%
14813.7	0.0495	14841.67	0.0498	0.19%	0.56%
15559.5	0.05	15779.88	0.05	1.42%	0.00%
16336.6	0.04951	16370.63	0.0498	0.21%	0.63%
17145.6	0.05	17410.08	0.05	1.54%	0.00%
17987.7	0.04949	18029.6	0.0498	0.23%	0.72%
18863.8	0.05	19179.88	0.05	1.68%	0.00%
19775.4	0.04945	19827.49	0.0499	0.26%	0.83%
20723.7	0.05	21101.34	0.05	1.82%	0.00%
21709.9	0.0494	21775.06	0.0499	0.30%	0.98%
22735.4	0.05	23187.71	0.05	1.99%	0.00%
23801.4	0.04932	23883.55	0.0499	0.35%	1.15%
24908.9	0.05	25452.22	0.05	2.18%	0.00%
26058.8	0.04922	26163.25	0.0499	0.40%	1.37%
27251.6	0.05	27906.36	0.05	2.40%	0.00%
28487.5	0.04911	28621.42	0.0499	0.47%	1.61%
29765.9	0.05	30557.1	0.05	2.66%	0.00%
31085.6	0.04898	31258.96	0.0499	0.56%	1.90%
32444.4	0.05	33402.51	0.05	2.95%	0.00%
33839.1	0.04885	34065.49	0.0499	0.67%	2.16%
35264.6	0.05	36426.07	0.05	3.29%	0.00%
36714.2	0.04879	37012.26	0.0499	0.81%	2.30%
38179	0.05	39596.4	0.05	3.71%	0.00%
39647.7	0.04893	40045.74	0.0499		
41106	0.05				
42536.7	0.04955				
43919.6	0.05				
45231.5	0.05				
46447	0.04863				
47538.9	0.05				
48479.8	0.04443				
49243.8	0.04931				
49808	0.0347				
50154.1	0.05				

Table 2. 20 Element Per Substructure 35 modes Kept

Check Wn	Check Zeta	Method 3 Wn	Method 3 Zeta	Error Wn Check Vs Method 3	Error Zeta Check Vs Method 3
14.0032	0.05	14.12993	0.05	0.91%	0.00%
38.6003	0.04967	38.60047	0.0497	0.00%	0.16%
75.6722	0.05	76.62695	0.05	1.26%	0.00%
125.091	0.04965	125.0931	0.0497	0.00%	0.07%
186.867	0.05	189.2424	0.05	1.27%	0.00%
261.001	0.04983	261.0113	0.0498	0.00%	0.02%
347.5	0.05	352.0073	0.05	1.30%	0.00%
446.368	0.04989	446.3968	0.0499	0.01%	0.00%
557.616	0.05	564.9938	0.05	1.32%	0.00%
681.258	0.04992	681.3252	0.0499	0.01%	0.02%
817.312	0.05	828.3466	0.05	1.35%	0.00%
965.804	0.04994	965.9393	0.0499	0.01%	0.03%
1126.76	0.05	1142.295	0.05	1.38%	0.00%
1300.23	0.04994	1300.48	0.0499	0.02%	0.03%
1486.26	0.05	1507.19	0.05	1.41%	0.00%
1684.91	0.04995	1685.319	0.0499	0.02%	0.03%
1896.24	0.05	1923.543	0.05	1.44%	0.00%
2120.35	0.04995	2121	0.0499	0.03%	0.03%
2357.32	0.05	2392.067	0.05	1.47%	0.00%
2607.28	0.04995	2608.269	0.0499	0.04%	0.02%
2870.35	0.05	2913.721	0.05	1.51%	0.00%
3146.67	0.04995	3148.112	0.0499	0.05%	0.01%
3436.41	0.05	3489.746	0.05	1.55%	0.00%
3739.75	0.04995	3741.78	0.0499	0.05%	0.00%
4056.87	0.05	4121.701	0.05	1.60%	0.00%
4388	0.04995	4390.79	0.05	0.06%	0.01%
4733.35	0.05	4811.474	0.05	1.65%	0.00%
5093.15	0.04996	5096.892	0.05	0.07%	0.02%
5467.64	0.05	5561.256	0.05	1.71%	0.00%
5857.02	0.04997	5861.919	0.05	0.08%	0.06%
6261.47	0.05	6373.383	0.05	1.79%	0.00%
6681.09	0.05	6687.34	0.05	0.09%	0.00%
7115.77	0.05	7249.799	0.05	1.88%	0.06%
7565.08	0.05	7572.772	0.05	0.10%	0.00%
8027.82	0.04994	8189.856	0.0499	2.02%	0.06%
8501.04	0.05	8509.775	0.05	0.10%	0.00%
8977.19	0.0498	9177.231	0.0498	2.23%	0.02%
9434.6	0.05	9441.573	0.05	0.07%	0.00%
9806.41	0.04955	10190.56	0.0488	3.92%	1.54%
11155.6	0.04907	11185.92	0.0494	0.27%	0.70%
11598.2	0.05	11732.62	0.05	1.16%	0.00%
12153.2	0.04934	12185.58	0.0496	0.27%	0.59%
12764.6	0.05	12979.25	0.05	1.68%	0.00%
13415.1	0.04946	13452.45	0.0497	0.28%	0.42%
14098.8	0.05	14373.69	0.05	1.95%	0.00%
14813.7	0.0495	14858.75	0.0499	0.30%	0.79%
15559.5	0.05	15897.01	0.05	2.17%	0.00%
16336.6	0.04951	16391.98	0.0497	0.34%	0.31%
17145.6	0.05	17555.52	0.05	2.39%	0.00%
17987.7	0.04949	18056.78	0.05	0.38%	1.03%
18863.8	0.05	19361.93	0.05	2.64%	0.00%
19775.4	0.04945	19862.83	0.0497	0.44%	0.58%
20723.7	0.05	21333.36	0.05	2.94%	0.00%
21709.9	0.0494	21822.38	0.0499	0.52%	1.07%
22735.4	0.05	23493.38	0.05	3.33%	0.00%
23801.4	0.04932	23950.06	0.0499	0.62%	1.10%
24908.9	0.05	25885.77	0.05	3.92%	0.00%
26058.8	0.04922	26267.28	0.0499	0.80%	1.36%
27251.6	0.05				
28487.5	0.04911				
29765.9	0.05				
31085.6	0.04898				
32444.4	0.05				
33839.1	0.04885				
35264.6	0.05				
36714.2	0.04879				
38179	0.05				
39647.7	0.04893				
41106	0.05				
42536.7	0.04955				
43919.6	0.05				
45231.5	0.05				
46447	0.04863				
47538.9	0.05				
48479.8	0.04443				
49243.8	0.04931				
49808	0.0347				
50154.1	0.05				

Table 3. 20 Element Per Substructure 30 modes Kept

Check Wn	Check Zeta	Method 3 Wn	Method 3 Zeta	Error Wn Check Vs Method 3	Error Zeta Check Vs Method 3
14.0032	0.05	14.17136	0.05	1.20%	0.00%
38.6003	0.04967	38.60066	0.0498	0.00%	0.20%
75.6722	0.05	76.9464	0.05	1.68%	0.00%
125.091	0.04965	125.0952	0.0497	0.00%	0.07%
186.867	0.05	190.0578	0.05	1.71%	0.00%
261.001	0.04983	261.0204	0.0498	0.01%	0.00%
347.5	0.05	353.5932	0.05	1.75%	0.00%
446.368	0.04989	446.4232	0.0499	0.01%	0.04%
557.616	0.05	567.6543	0.05	1.80%	0.00%
681.258	0.04992	681.3866	0.0499	0.02%	0.08%
817.312	0.05	832.4239	0.05	1.85%	0.00%
965.804	0.04994	966.0627	0.0499	0.03%	0.10%
1126.76	0.05	1148.174	0.05	1.90%	0.00%
1300.23	0.04994	1300.703	0.0499	0.04%	0.11%
1486.26	0.05	1515.305	0.05	1.95%	0.00%
1684.91	0.04995	1685.695	0.0499	0.05%	0.12%
1896.24	0.05	1934.384	0.05	2.01%	0.00%
2120.35	0.04995	2121.595	0.0499	0.06%	0.13%
2357.32	0.05	2406.196	0.05	2.07%	0.00%
2607.28	0.04995	2609.17	0.0499	0.07%	0.13%
2870.35	0.05	2931.781	0.05	2.14%	0.00%
3146.67	0.04995	3149.426	0.0499	0.09%	0.13%
3436.41	0.05	3512.486	0.05	2.21%	0.00%
3739.75	0.04995	3743.639	0.0499	0.10%	0.12%
4056.87	0.05	4149.998	0.05	2.30%	0.00%
4388	0.04995	4393.354	0.0499	0.12%	0.12%
4733.35	0.05	4846.373	0.05	2.39%	0.00%
5093.15	0.04996	5100.35	0.0499	0.14%	0.11%
5467.64	0.05	5604.01	0.05	2.49%	0.00%
5857.02	0.04997	5866.485	0.0499	0.16%	0.12%
6261.47	0.05	6425.508	0.05	2.62%	0.00%
6681.09	0.05	6693.23	0.0499	0.18%	0.14%
7115.77	0.05	7313.077	0.05	2.77%	0.00%
7565.08	0.05	7580.122	0.0499	0.20%	0.11%
8027.82	0.04994	8265.936	0.05	2.97%	0.12%
8501.04	0.05	8518.26	0.05	0.20%	0.00%
8977.19	0.0498	9262.239	0.05	3.18%	0.31%
9434.6	0.05	9448.436	0.05	0.15%	0.00%
9806.41	0.04955	10510.54	0.049	7.18%	1.02%
11155.6	0.04907	11215.53	0.0498	0.54%	1.47%
11598.2	0.05	11846.5	0.05	2.14%	0.00%
12153.2	0.04934	12220.83	0.0495	0.56%	0.24%
12764.6	0.05	13147.17	0.05	3.00%	0.00%
13415.1	0.04946	13496.48	0.05	0.61%	1.09%
14098.8	0.05	14601.45	0.05	3.57%	0.00%
14813.7	0.0495	14918.37	0.0497	0.71%	0.49%
15559.5	0.05	16222.34	0.05	4.26%	0.00%
16336.6	0.04951	16482.55	0.0499	0.89%	0.75%
17145.6	0.05				
17987.7	0.04949				
18863.8	0.05				
19775.4	0.04945				
20723.7	0.05				
21709.9	0.0494				
22735.4	0.05				
23801.4	0.04932				
24908.9	0.05				
26058.8	0.04922				
27251.6	0.05				
28487.5	0.04911				
29765.9	0.05				
31085.6	0.04898				
32444.4	0.05				
33839.1	0.04885				
35264.6	0.05				
36714.2	0.04879				
38179	0.05				
39647.7	0.04893				
41106	0.05				
42536.7	0.04955				
43919.6	0.05				
45231.5	0.05				
46447	0.04863				
47538.9	0.05				
48479.8	0.04443				
49243.8	0.04931				
49808	0.0347				
50154.1	0.05				

Table 4. 20 Element Per Substructure 25 modes Kept

Check Wn	Check Zeta	Method 3 Wn	Method 3 Zeta	Error Wn Check Vs Method 3	Error Zeta Check Vs Method 3
14.0032	0.05	14.21897	0.05	1.54%	0.00%
38.6003	0.04967	38.60105	0.0498	0.00%	0.24%
75.6722	0.05	77.31798	0.05	2.17%	0.00%
125.091	0.04965	125.0995	0.0497	0.01%	0.06%
186.867	0.05	191.0191	0.05	2.22%	0.00%
261.001	0.04983	261.0391	0.0498	0.01%	0.04%
347.5	0.05	355.4874	0.05	2.30%	0.00%
446.368	0.04989	446.4781	0.0498	0.02%	0.11%
557.616	0.05	570.8732	0.05	2.38%	0.00%
681.258	0.04992	681.5146	0.0498	0.04%	0.16%
817.312	0.05	837.4212	0.05	2.46%	0.00%
965.804	0.04994	966.32	0.0498	0.05%	0.19%
1126.76	0.05	1155.473	0.05	2.55%	0.00%
1300.23	0.04994	1301.17	0.0498	0.07%	0.22%
1486.26	0.05	1525.511	0.05	2.64%	0.00%
1684.91	0.04995	1686.48	0.0498	0.09%	0.24%
1896.24	0.05	1948.202	0.05	2.74%	0.00%
2120.35	0.04995	2122.842	0.0498	0.12%	0.25%
2357.32	0.05	2424.451	0.05	2.85%	0.00%
2607.28	0.04995	2611.063	0.0498	0.15%	0.26%
2870.35	0.05	2955.449	0.05	2.96%	0.00%
3146.67	0.04995	3152.203	0.0498	0.18%	0.26%
3436.41	0.05	3542.739	0.05	3.09%	0.00%
3739.75	0.04995	3747.599	0.0498	0.21%	0.26%
4056.87	0.05	4188.262	0.05	3.24%	0.00%
4388	0.04995	4398.88	0.0498	0.25%	0.25%
4733.35	0.05	4894.407	0.05	3.40%	0.00%
5093.15	0.04996	5107.925	0.0498	0.29%	0.27%
5467.64	0.05	5664.005	0.05	3.59%	0.00%
5857.02	0.04997	5876.723	0.0498	0.34%	0.27%
6261.47	0.05	6500.173	0.05	3.81%	0.00%
6681.09	0.05	6706.887	0.0498	0.39%	0.32%
7115.77	0.05	7405.475	0.05	4.07%	0.00%
7565.08	0.05	7597.999	0.0499	0.44%	0.29%
8027.82	0.04994	8377.479	0.05	4.36%	0.12%
8501.04	0.05	8540.406	0.0499	0.46%	0.24%
8977.19	0.0498	9373.524	0.05	4.41%	0.40%
9434.6	0.05	9468.096	0.0499	0.36%	0.14%
9806.41	0.04955				
11155.6	0.04907				
11598.2	0.05				
12153.2	0.04934				
12764.6	0.05				
13415.1	0.04946				
14098.8	0.05				
14813.7	0.0495				
15559.5	0.05				
16336.6	0.04951				
17145.6	0.05				
17987.7	0.04949				
18863.8	0.05				
19775.4	0.04945				
20723.7	0.05				
21709.9	0.0494				
22735.4	0.05				
23801.4	0.04932				
24908.9	0.05				
26058.8	0.04922				
27251.6	0.05				
28487.5	0.04911				
29765.9	0.05				
31085.6	0.04898				
32444.4	0.05				
33839.1	0.04885				
35264.6	0.05				
36714.2	0.04879				
38179	0.05				
39647.7	0.04893				
41106	0.05				
42536.7	0.04955				
43919.6	0.05				
45231.5	0.05				
46447	0.04863				
47538.9	0.05				
48479.8	0.04443				
49243.8	0.04931				
49808	0.0347				
50154.1	0.05				

Table 5. 20 Element Per Substructure 20 modes Kept

Check Wn	Check Zeta	Method 3 Wn	Method 3 Zeta	Error Wn Check Vs Method 3	Error Zeta Check Vs Method 3
14.0032	0.05	14.2942	0.05	2.08%	0.00%
38.6003	0.04967	38.60231	0.0498	0.01%	0.29%
75.6722	0.05	77.91517	0.05	2.96%	0.00%
125.091	0.04965	125.1135	0.0497	0.02%	0.04%
186.867	0.05	192.5928	0.05	3.06%	0.00%
261.001	0.04983	261.0998	0.0498	0.04%	0.11%
347.5	0.05	358.6457	0.05	3.21%	0.00%
446.368	0.04989	446.6556	0.0498	0.06%	0.20%
557.616	0.05	576.3424	0.05	3.36%	0.00%
681.258	0.04992	681.9285	0.0498	0.10%	0.27%
817.312	0.05	846.0811	0.05	3.52%	0.00%
965.804	0.04994	967.1543	0.0498	0.14%	0.31%
1126.76	0.05	1168.397	0.05	3.69%	0.00%
1300.23	0.04994	1302.692	0.0498	0.19%	0.34%
1486.26	0.05	1544.029	0.05	3.89%	0.00%
1684.91	0.04995	1689.065	0.0498	0.25%	0.35%
1896.24	0.05	1974.008	0.05	4.10%	0.00%
2120.35	0.04995	2127.016	0.0498	0.31%	0.36%
2357.32	0.05	2459.782	0.05	4.35%	0.00%
2607.28	0.04995	2617.57	0.0498	0.39%	0.35%
2870.35	0.05	3003.435	0.05	4.64%	0.00%
3146.67	0.04995	3162.139	0.0498	0.49%	0.33%
3436.41	0.05	3608.124	0.05	5.00%	0.00%
3739.75	0.04995	3762.706	0.0498	0.61%	0.31%
4056.87	0.05	4279.266	0.05	5.48%	0.00%
4388	0.04995	4422.318	0.0498	0.78%	0.29%
4733.35	0.05	5029.64	0.05	6.26%	0.00%
5093.15	0.04996	5147.391	0.0498	1.06%	0.25%
5467.64	0.05				
5857.02	0.04997				
6261.47	0.05				
6681.09	0.05				
7115.77	0.05				
7565.08	0.05				
8027.82	0.04994				
8501.04	0.05				
8977.19	0.0498				
9434.6	0.05				
9806.41	0.04955				
11155.6	0.04907				
11598.2	0.05				
12153.2	0.04934				
12764.6	0.05				
13415.1	0.04946				
14098.8	0.05				
14813.7	0.0495				
15559.5	0.05				
16336.6	0.04951				
17145.6	0.05				
17987.7	0.04949				
18863.8	0.05				
19775.4	0.04945				
20723.7	0.05				
21709.9	0.0494				
22735.4	0.05				
23801.4	0.04932				
24908.9	0.05				
26058.8	0.04922				
27251.6	0.05				
28487.5	0.04911				
29765.9	0.05				
31085.6	0.04898				
32444.4	0.05				
33839.1	0.04885				
35264.6	0.05				
36714.2	0.04879				
38179	0.05				
39647.7	0.04893				
41106	0.05				
42536.7	0.04955				
43919.6	0.05				
45231.5	0.05				
46447	0.04863				
47538.9	0.05				
48479.8	0.04443				
49243.8	0.04931				
49808	0.0347				
50154.1	0.05				

Table 6. 20 Element Per Substructure 15 modes Kept

Check Wn	Check Zeta	Method 3 Wn	Method 3 Zeta	Error Wn Check Vs Method 3	Error Zeta Check Vs Method 3
14.0032	0.05	14.45803	0.05	3.25%	0.00%
38.6003	0.04967	38.60776	0.0498	0.02%	0.36%
75.6722	0.05	79.25888	0.05	4.74%	0.00%
125.091	0.04965	125.1741	0.0496	0.07%	0.07%
186.867	0.05	196.2635	0.05	5.03%	0.00%
261.001	0.04983	261.3633	0.0497	0.14%	0.28%
347.5	0.05	366.2861	0.05	5.41%	0.00%
446.368	0.04989	447.4288	0.0496	0.24%	0.49%
557.616	0.05	590.1081	0.05	5.83%	0.00%
681.258	0.04992	683.7469	0.0497	0.37%	0.35%
817.312	0.05	868.9125	0.05	6.31%	0.00%
965.804	0.04994	970.8904	0.0495	0.53%	0.78%
1126.76	0.05	1204.563	0.05	6.90%	0.00%
1300.23	0.04994	1309.773	0.0498	0.73%	0.24%
1486.26	0.05	1600.535	0.05	7.69%	0.00%
1684.91	0.04995	1702.045	0.0496	1.02%	0.64%
1896.24	0.05	2065.763	0.05	8.94%	0.00%
2120.35	0.04995	2151.829	0.0498	1.48%	0.36%
2357.32	0.05				
2607.28	0.04995				
2870.35	0.05				
3146.67	0.04995				
3436.41	0.05				
3739.75	0.04995				
4056.87	0.05				
4388	0.04995				
4733.35	0.05				
5093.15	0.04996				
5467.64	0.05				
5857.02	0.04997				
6261.47	0.05				
6681.09	0.05				
7115.77	0.05				
7565.08	0.05				
8027.82	0.04994				
8501.04	0.05				
8977.19	0.0498				
9434.6	0.05				
9806.41	0.04955				
11155.6	0.04907				
11598.2	0.05				
12153.2	0.04934				
12764.6	0.05				
13415.1	0.04946				
14098.8	0.05				
14813.7	0.0495				
15559.5	0.05				
16336.6	0.04951				
17145.6	0.05				
17987.7	0.04949				
18863.8	0.05				
19775.4	0.04945				
20723.7	0.05				
21709.9	0.0494				
22735.4	0.05				
23801.4	0.04932				
24908.9	0.05				
26058.8	0.04922				
27251.6	0.05				
28487.5	0.04911				
29765.9	0.05				
31085.6	0.04898				
32444.4	0.05				
33839.1	0.04885				
35264.6	0.05				
36714.2	0.04879				
38179	0.05				
39647.7	0.04893				
41106	0.05				
42536.7	0.04955				
43919.6	0.05				
45231.5	0.05				
46447	0.04863				
47538.9	0.05				
48479.8	0.04443				
49243.8	0.04931				
49808	0.0347				
50154.1	0.05				

Table 7. 20 Element Per Substructure 10 modes Kept

Check Wn	Check Zeta	Method 3 Wn	Method 3 Zeta	Error Wn Check Vs Method 3	Error Zeta Check Vs Method 3
14.0032	0.05	15.00189	0.05	7.13%	0.00%
38.6003	0.04967	38.66069	0.0497	0.16%	0.02%
75.6722	0.05	84.17335	0.05	11.23%	0.00%
125.091	0.04965	125.7672	0.05	0.54%	0.70%
186.867	0.05	211.4017	0.05	13.13%	0.00%
261.001	0.04983	264.0277	0.0485	1.16%	2.62%
347.5	0.05	403.839	0.05	16.21%	0.00%
446.368	0.04989	456.284	0.05	2.22%	0.19%
557.616	0.05				
681.258	0.04992				
817.312	0.05				
965.804	0.04994				
1126.76	0.05				
1300.23	0.04994				
1486.26	0.05				
1684.91	0.04995				
1896.24	0.05				
2120.35	0.04995				
2357.32	0.05				
2607.28	0.04995				
2870.35	0.05				
3146.67	0.04995				
3436.41	0.05				
3739.75	0.04995				
4056.87	0.05				
4388	0.04995				
4733.35	0.05				
5093.15	0.04996				
5467.64	0.05				
5857.02	0.04997				
6261.47	0.05				
6681.09	0.05				
7115.77	0.05				
7565.08	0.05				
8027.82	0.04994				
8501.04	0.05				
8977.19	0.0498				
9434.6	0.05				
9806.41	0.04955				
11155.6	0.04907				
11598.2	0.05				
12153.2	0.04934				
12764.6	0.05				
13415.1	0.04946				
14098.8	0.05				
14813.7	0.0495				
15559.5	0.05				
16336.6	0.04951				
17145.6	0.05				
17987.7	0.04949				
18863.8	0.05				
19775.4	0.04945				
20723.7	0.05				
21709.9	0.0494				
22735.4	0.05				
23801.4	0.04932				
24908.9	0.05				
26058.8	0.04922				
27251.6	0.05				
28487.5	0.04911				
29765.9	0.05				
31085.6	0.04898				
32444.4	0.05				
33839.1	0.04885				
35264.6	0.05				
36714.2	0.04879				
38179	0.05				
39647.7	0.04893				
41106	0.05				
42536.7	0.04955				
43919.6	0.05				
45231.5	0.05				
46447	0.04863				
47538.9	0.05				
48479.8	0.04443				
49243.8	0.04931				
49808	0.0347				
50154.1	0.05				

Table 8. 20 Element Per Substructure 5 modes Kept

Modes Kept	ζ_o	ζ_{subs} ub	ζ_{subs} lb	ζ_{sys} ub	ζ_{sys} lb	ζ_{subs} max	ζ_{subs} min	ζ_{sys} max	ζ_{sys} min
ALL	0.2	0.1	0.01	0.05	0.01	0.1	0.0339	0.05	0.034705
35	0.2	0.1	0.01	0.05	0.01	0.0721	0.0501	0.05	0.0489
30	0.2	0.1	0.01	0.05	0.01	0.0722	0.0502	0.05	0.0488
25	0.2	0.1	0.01	0.05	0.01	0.0723	0.05	0.05	0.049
20	0.2	0.1	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497
15	0.2	0.1	0.01	0.05	0.01	0.0725	0.0503	0.05	0.0497
10	0.2	0.1	0.01	0.05	0.01	0.0728	0.0506	0.05	0.0495
5	0.2	0.1	0.01	0.05	0.01	0.0737	0.0521	0.05	0.0485

Table 9. For a 20 Element Beam Structure with varying number of modes retained; Calculated Results for Substructure Damping which meets the prescribed system damping constraint and bounds for the values of substructure damping. The results for system damping are given as well.

ζ_o	ζ_{subs} ub	ζ_{subs} lb	ζ_{sys} ub	ζ_{sys} lb	ζ_{subs} max	ζ_{subs} min	ζ_{sys} max	ζ_{sys} min	Iterations	Function Call
0.001	0.1	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497	2	63
0.005	0.1	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497	2	63
0.01	0.1	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497	2	63
0.1	0.1	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497	2	63
0.2	0.1	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497	2	63
0.5	0.1	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497	2	63
1	0.1	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497	2	63
10	0.1	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497	2	63

Table 10. Using a 20 Element Beam Substructures and Keeping only 20 modes; System and Substructure damping results with varying start points (initial values of the design variable, substructure damping)

ζ_o	ζ_{subs} ub	ζ_{subs} lb	ζ_{sys} ub	ζ_{sys} lb	ζ_{subs} max	ζ_{subs} min	ζ_{sys} max	ζ_{sys} min	Iterations	Function Call
0.2	0.2	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497	2	63
0.2	0.5	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497	2	63
0.2	0.1	0.01	0.05	0.01	0.0724	0.0501	0.05	0.0497	2	63
0.2	0.05	0.01	0.05	0.01	0.0500	0.0500	0.0498	0.0353	1	42
0.2	0.03	0.01	0.05	0.01	0.0300	0.0300	0.0299	0.0212	1	42
0.2	0.05	0.03	0.05	0.01	0.0500	0.0500	0.0498	0.0353	1	42
0.2	0.04	0.03	0.05	0.01	0.0400	0.0400	0.0399	0.0283	1	42
0.2	0.015	0	0.05	0.01	0.0150	0.0150	0.0150	0.0106	1	42

Table 11. Using a 20 Element Beam Substructures and Keeping only 20 modes and Initial value for the design variable kept at 0.2; System and Substructure damping results with varying Upper and Lower Bounds for Substructure damping

ζ_o	ζ_{subs} ub	ζ_{subs} lb	ζ_{sys} ub	ζ_{sys} lb	ζ_{subs} max	ζ_{subs} min	ζ_{sys} max	ζ_{sys} min	Iterations	Function Call
0.2	0.05	0.01	0.1	0.01	0.0500	0.0500	0.0498	0.0353	1	42
0.2	0.05	0.01	0.04	0.01	0.0500	0.0401	0.0400	0.0352	2	63
0.2	0.05	0.01	0.03	0.01	0.0434	0.0301	0.0300	0.0298	2	63
0.2	0.05	0.01	0.03	0.02	0.0434	0.0301	0.0300	0.0298	2	63
0.2	0.05	0.01	0.03	0.03	0.0434	0.0301	0.0300	**0.0291***	2	63
0.2	0.05	0.01	0.015	0.01	0.0217	0.0150	0.0150	0.0149	2	63

Table 12. Using a 20 Element Beam Substructures and Keeping only 20 modes and Initial value for the design variable kept at 0.2; System and Substructure damping results with varying Upper and Lower Bounds for System damping

ζ_o	ζ_{subs} ub	ζ_{subs} lb	ζ_{sys} ub	ζ_{sys} lb	ζ_{subs} max	ζ_{subs} min	ζ_{sys} max	ζ_{sys} Min	Iterations	Function Call
1	.015	0.01	0.05	0.02	**0.0208***	**0.0208***	0.0207	0.0147	2	63
0.2	0.05	0.01	0.045	0.04	**0.0507***	**0.0093***	**0.0456***	**0.0093***	2	63
0.2	0.05	0.045	0.045	0.04	**0.0507***	**0.0443***	**0.0456***	**0.0358***	2	63
0.2	0.05	0.01	0.045	0.038	0.0500	0.0451	0.0450	**0.0353***	2	63
0.2	0.05	0.045	0.045	0.036	0.0500	0.0451	0.0450	**0.0353***	2	63
0.2	0.05	0.045	0.045	0.035	0.0500	0.0451	0.0450	0.0353	2	63
0.2	0.06	0.045	0.045	0.040	0.0600	0.0451	0.0450	0.0419	2	63
0.2	0.1	0.2	0.02	0.05	0.0724	0.0501	0.05	0.0497	2	63

Table 13. Using a 20 Element Beam Substructures and Keeping only 20 modes; System and Substructure damping results with varying Upper and Lower Bounds for System and Substructure damping and Start Point

Size of Matrix	Time In Seconds Trial 1	Time in Seconds $O(n^3)$ Trial 1	Time In Seconds Trial 2	Time in Seconds $O(n^3)$ Trial 2	Time In Seconds Trial 3	Time in Seconds $O(n^3)$ Trial 3
N	0.000433	0.000244	0.000409	0.000244	0.000439	0.000244
2N	0.002612	0.001953	0.002563	0.001953	0.002677	0.001953
4N	0.017596	0.015625	0.017392	0.015625	0.017575	0.015625
8N	0.127353	0.125000	0.126400	0.125000	0.128845	0.125000
16N	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Table 14. Table showing the time it takes to calculate the eigenvectors and eigenvalues of varying matrix sizes.

Using the "tic, toc, rand, and eig" commands in MATLAB, we compared the time it takes to solve for eigvenvalues and eigenvectors of varying size symmetric matrices. The "tic toc" command in MATLAB, measures the amount of time that it takes MATLAB to complete one or more operations, and displays the time in seconds. The "rand" command on the other hand, returns a pseudorandom, scalar value drawn from a

uniform distribution on the unit interval, for this thesis varying matrix of sizes *N, 2N, 4N, 8N and 16N* comprised of these random numbers are compared. The "eig" command solves for the eigenvalues and eigenvector of the given square matrices. Lastly, the table also make use of the generalized equation $O(n^3)$ which solves for the amount of time is takes to solve varying size of symmetric matrices, where n is the dimension of the matrix.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

Since it is not often that we conduct vibrations test on system structure, the method of conducting test on systems sub structural parts then synthesizing the results is an efficient way of predicting or determining system modal properties. In this thesis we will exploit the concept of substructure coupling which makes use of both linear homogenous equations of constraints and generated substructure modal parameters such as natural frequencies, modal damping and mode shapes to determine system's dynamic response. In order to achieve such task, we made use of the zero-eigenvalue theorem, of which, made use of a set of given constraints to generate a transformation matrix which was utilized to convert modal block matrix, either ω^2 and/or $2\zeta\omega$, to solve for system's modal parameters. The results of mentioned coupling are tabulated in tables one through eight of the results section.

Looking at tables one through eight, we can see that by applying more modal parameters from each substructure in solving for the system dynamic properties, the solution converges to the actual system solution, which was generated to check the results from coupling. However, it is not often desired to use all if not most modes in calculating for system's response due to that fact that computational cost becomes a bigger concern when solving higher order matrix problems. Therefore, if we are able to compute the desired output, in this case, system's damping ratios, by making use of only a few modes extracted from each substructure, we are be able to eliminate the economic concerns for this calculation. Furthermore, these tables also gave system bounds given substructure modal properties, which were used as the initial building block in formulating an optimization routine in solving for optimum values for substructure damping which would satisfy prescribed bounds for system damping.

Using a built in command in MATLAB, "fmincon", and by slightly modifying the formulated MATLAB code used for structural coupling, we were able to achieve the goal of being able to produce values for substructure damping which satisfies bounds imposed on system damping. By looking at tables nine through thirteen, which are results taken

from coupling a beam structure with 20 elements, by varying either start points or initial values for the design variable (substructure damping) or the upper and lower bounds for substructure and system damping ratios, we were able to solve for an optimum value for the design variable; therefore we can say that the methodology worked. The first part of the optimization analysis dealt with varying only the start point while holding the values of the bounds for both substructure and system damping. After conducting this analysis we may say that the start point did not have any impact on the optimizer's solution finding since it found the same solution for all of the attempts that was made, this is due to the fact that we have a linear objective function of which had no local extremes within the objective function. Secondly, by looking closely on tables twelve and thirteen, we can see that the optimizer was not able to produce values that met the prescribed bounds. In table twelve row five column nine when we designated a system without bounds in this case both lower and upper bounds were valued at 0.03, the optimizer did not find a feasible solution for the problem; furthermore, looking at table thirteen in the first five rows we observe that values of either substructure's or system's maximum or minimum modal damping solutions does not meet the bounds set in the optimization program. Upon careful inspection of these problems, we conclude that the reason for not having a feasible solution is because the bounds set for these trials are too stringent for the optimizer to solve, the reason being, as we slowly space out the bounds for the system damping the better the solution becomes as shown in rows six and seven, row six showed how expanding the bounds for the system enable us to process feasible results while row seven was gave more separation between the upper bounds of system and substructure damping. The formulation of such method which determines practical spacing for system damping may be further investigated and can be a follow on topic for this thesis.

Lastly, for this thesis we also investigated plate structures that were modeled in PATRAN and NASTRAN. During this process we were able to come up with modal parameters for each substructure after modeling, in fact we were able to convert the files generated in NASTRAN to a useable database file that was easily extracted for use in MATLAB; however upon coupling the two substructures we were not able to match system's modal properties which lead to the conclusion that there might be a mistake

hidden within the MATLAB code that was previously generated. As a result we will continue to do the work for the plate structure; however, the results will not be shown for this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. RECOMMENDATION

Further investigate why it is that system's maximum natural frequency and maximum damping ratio does not exceed that of either substructure's maximum frequency and damping.

Formulate a method which determines practical spacing for system and/or substructure modal damping bounds which will give practical results that may be used in real world scenario.

Apply above concept to more than two substructures. Create a multi jointed structure and divide them into substructure then make use of the transformation matrix and optimizer to produce modal damping for substructures that will then satisfy the given constraints.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX

NOTE: If the main program calls for subprograms, the subprogram used for substructure A is identical substructure B with the exception everything that refers to A must be replaced with B; as a result only subprogram for substructure A are shown.

A-1 INITIAL WORK COUPLING 1 DOF MASS SPRING SYSTEM, WHICH MAKES USE OF A SYSTEM MATRIX THAT WAS CONVETED FROM 2ND ORDER DIFFERENTIAL TO FIRST ORDER DIFFERENTIAL

```
%This will calculate a simple 2DOF mass spring system
%Solution will be system natural freq and damping
%The Solution will be compared to the results gotten from Mathematica
%The equations of motions were transformed from a 2nd order differential to
%a first order differential
%Given
M1 = .01;
M2 = .02;
K1 = 100;
K2 = 150;
zeta_SubA = .02;
zeta_SubB = .01;
 %Calculate Natural Frequencies
Wn_SubA = sqrt(K1/M1)
Wn_SubB = sqrt(K2/M2)
[PhiA,LamA] = eig(K1,M1)
WnA = sqrt(LamA)
[PhiB,LamB] = eig(K2,M2)
WnB = sqrt(LamB)
%Calculate C1 and C2 using Substructure's zeta and Wn
C1 = 2*zeta_SubA*Wn_SubA*M1;
C2 = 2*zeta_SubB*Wn_SubB*M2;
 A = [0, 0, 1, 0; 0, 0, 0, 1;...
     M1\(1/150*(M1\K1))-M1\K1, -M1\(1/150*(M2\K2)), M1\(1/150*(M1\C1))-M1\C1, -
M1\(1/150*(M2\C2));...
      -M2\(1/150*(M1\K1)), M2\(1/150*(M2\K2))-M2\K2, -M2\(1/150*(M1\C1)),
M2\(1/150*(M2\C2))-M2\C2]
 Lam = eig(A);
 sigma = real(Lam);
Wd = imag(Lam);
 for i = 1:1
    WD(i,1) = Wd(i*2-1,1);
    SIGMA(i,1) = sigma(i*2-1,1);
end
 for i =1:1
    Zeta(i,1) = sqrt((((WD(i,1)^2/SIGMA(i,1)^2)+1)^-1));
    WN(i,1) = -SIGMA(i,1)/Zeta(i,1);
end
WN
Zeta
 %Compare with Mathematica
Mathematica_Fn =  91.2871
Mathematica_Zeta = .0136275
 %Check for Mathematica
C3 = C1+C2;
M3 = M1+M2;
K3 = K1+K2;
Check_Fn = sqrt(K3/M3)
Check_Zeta = C3/(2*Check_Fn*M3)
```

A-2 COUPLING 2 DOF MASS SPRING SYSTEM, WHICH MAKES USE OF A SYSTEM MATRIX THAT WAS CONVETED FROM 2ND ORDER DIFFERENTIAL TO FIRST ORDER DIFFERENTIAL

```
%Given Values
nel=2;                    % number of elements
nnel=2;                   % number of nodes per element
ndof=1;                   % number of dofs per node
nnode=(nnel-1)*nel+1;     % total number of nodes in system
sdof=nnode*ndof;          % total system dofs
conek = 1;                %# of connection
num_subs = 2;             %# of substructure
m1 = [.01,.02];             %mass values for subs1
k1 = [100,125];           %stiffness values for subs1
zetas1 = [.01,.02];       %zeta values for subs1
 m2 = [.03,.04];            %mass values for subs2
k2 = [150,175];           %stiffness values for subs2
zetas2 = [.01,.02];       %zeta values for subs2
K1=zeros(sdof,sdof);
K2=zeros(sdof,sdof);
 %Building of K and M matrices for each substructure;
 index = [0 , 1];
%Substructure A
 for i = 1 : length(k1);
     index = index + 1;
     ka = [k1(1,i) , -k1(1,i) ; -k1(1,i) ,  k1(1,i)];
     K1(index,index) = K1(index,index) + ka;
 end
 %Apply boundary conditions for subs A
Keep = 2:sdof;
Ka = K1(Keep, Keep);
Ma = diag(m1);
 %Calculation of Substructure "A"s  mode shapes, natural freq, and
%physical C matrix
 [PhiA,LamA] = eig(Ka,Ma);
Wa = sqrt(LamA);
Check_MassNormA = PhiA'*Ma*PhiA;
 %Substructure A's C matrix
for i = 1:length(zetas1)
    Z_Wn_A(i,i)  = 2*Wa(i,i)*zetas1(1,i);
end
Ca = Ma*PhiA*Z_Wn_A*PhiA'*Ma;
 index = [0 , 1];
%Substructure B
for i = 1 : length(k2);
     index = index + 1;
     kb = [k2(1,i) , -k2(1,i) ; -k2(1,i) ,  k2(1,i)];
     K2(index,index) = K2(index,index) + kb;
end
 %Apply boundary conditions for subs B
Keep = 1:sdof-1;
Kb = K2(Keep, Keep);
Mb = diag(m2);
 %Calculation of Substructure "B"s  mode shapes, natural freq, and
%physical C matrix
[PhiB,LamB] = eig(Kb,Mb);
Wb = sqrt(LamB);
Check_MassNormB = PhiB'*Mb*PhiB;
 %Substructure B's C matrix
for i = 1:length(zetas2)
    Z_Wn_B(i,i)  = 2*Wb(i,i)*zetas2(1,i);
end
Cb = Mb*PhiB*Z_Wn_B*PhiB'*Mb;
 Pa1 = PhiA(:,1);
Pa2 = PhiA(:,2);
Pb1 = PhiB(:,1);
Pb2 = PhiB(:,2);
Ma1 = 1;
Ma2 = 1;
Mb1 = 1;
Mb2 = 1;
Q = Pa1(2,1)^2/Ma1 + Pa2(2,1)^2/Ma2 + Pb1(1,1)^2/Mb1 + Pb2(1,1)^2/Mb2;
 % A matrix used to calculate Wn and zeta for the overall structure
%This matrix is derived from the two substructures
A = [ 0, 0, 0, 0, 1, 0, 0, 0;...
      0, 0, 0, 0, 0, 1, 0, 0;...
```

```
      0, 0, 0, 0, 0, 0, 1, 0;...
      0, 0, 0, 0, 0, 0, 0, 1;...
      Ma1\((Q\(Pa1(2,1)^2))*Wa(1,1)^2) - Wa(1,1)^2,
Ma1\((Q\(Pa1(2,1)*Pa2(2,1)))*Wa(2,2)^2),...
      -Ma1\((Q\(Pa1(2,1)*Pb1(1,1)))*Wb(1,1)^2), -
Ma1\((Q\(Pa1(2,1)*Pb2(1,1)))*Wb(2,2)^2),...
      ...
      Ma1\((Q\(Pa1(2,1)^2))*Z_Wn_A(1,1)) - Z_Wn_A(1,1),
Ma1\((Q\(Pa1(2,1)*Pa2(2,1)))*Z_Wn_A(2,2)),...
      -Ma1\((Q\(Pa1(2,1)*Pb1(1,1)))*Z_Wn_B(1,1)), -
Ma1\((Q\(Pa1(2,1)*Pb2(1,1)))*Z_Wn_B(2,2));...
      ...
      Ma2\((Q\(Pa2(2,1)*Pa1(2,1)))*Wa(1,1)^2), Ma2\((Q\(Pa2(2,1)^2))*Wa(2,2)^2) -
Wa(2,2)^2,...
      -Ma2\((Q\(Pa2(2,1)*Pb1(1,1)))*Wb(1,1)^2), -
Ma2\((Q\(Pa2(2,1)*Pb2(1,1)))*Wb(2,2)^2),...
      ...
      Ma2\((Q\(Pa2(2,1)*Pa1(2,1)))*Z_Wn_A(1,1)), Ma2\((Q\(Pa2(2,1)^2))*Z_Wn_A(2,2))
- Z_Wn_A(2,2),...
      -Ma2\((Q\(Pa2(2,1)*Pb1(1,1)))*Z_Wn_B(1,1)), -
Ma2\((Q\(Pa2(2,1)*Pb2(1,1)))*Z_Wn_B(2,2));...
      ...
      -Mb1\((Q\(Pb1(1,1)*Pa1(2,1)))*Wa(1,1)^2), -
Mb1\((Q\(Pb1(1,1)*Pa2(2,1)))*Wa(2,2)^2),...
      Mb1\((Q\(Pb1(1,1)^2))*Wb(1,1)^2) - Wb(1,1)^2,
Mb1\((Q\(Pb1(1,1)*Pb2(1,1)))*Wb(2,2)^2),...
      ...
      -Mb1\((Q\(Pb1(1,1)*Pa1(2,1)))*Z_Wn_A(1,1)), -
Mb1\((Q\(Pb1(1,1)*Pa2(2,1)))*Z_Wn_A(2,2)),...
      Mb1\((Q\(Pb1(1,1)^2))*Z_Wn_B(1,1)) - Z_Wn_B(1,1),
Mb1\((Q\(Pb1(1,1)*Pb2(1,1)))*Z_Wn_B(2,2));...
      ...
      -Mb2\((Q\(Pb2(1,1)*Pa1(2,1)))*Wa(1,1)^2), -
Mb2\((Q\(Pb2(1,1)*Pa2(2,1)))*Wa(2,2)^2),...
      Mb2\((Q\(Pb2(1,1)*Pb1(1,1)))*Wb(1,1)^2), Mb2\((Q\(Pb2(1,1)^2))*Wb(2,2)^2) -
Wb(2,2)^2,...
      ...
      -Mb2\((Q\(Pb2(1,1)*Pa1(2,1)))*Z_Wn_A(1,1)), -
Mb2\((Q\(Pb2(1,1)*Pa2(2,1)))*Z_Wn_A(2,2)),...
      Mb2\((Q\(Pb2(1,1)*Pb1(1,1)))*Z_Wn_B(1,1)), Mb2\((Q\(Pb2(1,1)^2))*Z_Wn_B(2,2))
- Z_Wn_B(2,2)];
  LAM = eig(A);
 % %Trial
% keeep = 5:8
% kep = 1:4
%
% G = A(keeep,keeep)
% L = A(keeep,kep)
% Omega_N = sqrt(L)
% eig(G)
% eig(Omega_N)
% for i = 1:4
%     Zeta =
 % eigenvalues of A are sigma(i) +/- j wd(i)
% sigma = zeta(i) * Wn(i)
% Wd(i) = Wn(i) * sqrt(1-zeta(i))
% two eqns in two unknowns
 sigma = real(LAM);
Wd = imag(LAM);

for i = 1:4
    WD(i,1) = Wd(i*2-1,1);
    SIGMA(i,1) = sigma(i*2-1,1);
end
WD;
SIGMA;

for i = 1:sdof
    Zeta(i,1) = sqrt((((WD(i,1)^2/SIGMA(i,1)^2)+1)^-1));
end
for i = 1:sdof
    WN(i,1) = -SIGMA(i,1)/Zeta(i,1);
end
WN;
Zeta;
Table_Solution_WN_Zeta = [WN Zeta]
 %Check Natural Frequencies
%Building overall C, M, K matrices
C = zeros(sdof,sdof);
```

```
M = zeros(sdof,sdof);
K = zeros(sdof,sdof);
index = [1 , 2];
C(index,index) = C(index,index) + Ca;
M(index,index) = M(index,index) + Ma;
K(index,index) = K(index,index) + Ka;
index = index + 1;
C(index,index) = C(index,index) + Cb;
M(index,index) = M(index,index) + Mb;
K(index,index) = K(index,index) + Kb;
 [Phi,Lam] = eig(K,M);
for i =1:length(M)
    Check_Natfreq(i,1) = sqrt(Lam(i,i));
end
Check_Natfreq;
 % Check Zeta
C = Phi'*C*Phi;
for i = 1:length(M)
    Check_Zeta(i,1) = C(i,i)/(2*Check_Natfreq(i,1));
end
Check_Zeta;

Table_Check_Natfreq_Zeta = [Check_Natfreq Check_Zeta]
```

A-3 COUPLING OF 2 DOF MASS SPRING SYSTEM USING THE ZERO-EIGENVALUE THEOREM

```
nel=2;                    % number of elements
nnel=2;                   % number of nodes per element
ndof=1;                   % number of dofs per node
nnode=(nnel-1)*nel+1;     % total number of nodes in system
sdof=nnode*ndof;          % total system dofs
conek = 1;                %# of connection
num_subs = 2;             %# of substructure
 m1 = [.01,.02];            %mass values for subs1
k1 = [100,125];           %stiffness values for subs1
zetas1 = [.01,.01];     %zeta values for subs1
 m2 = [.01,.02];             %mass values for subs2
k2 = [125,100];           %stiffness values for subs2
zetas2 = [.01,.01];       %zeta values for subs2
K1=zeros(sdof,sdof);
K2=zeros(sdof,sdof);
 %Building of K and M matrices for each substructure;
 index = [0 , 1];
%Substructure A
 for i = 1 : length(k1)
     index = index + 1 ;
     ka = [k1(1,i) , -k1(1,i) ; -k1(1,i) ,  k1(1,i)];
     K1(index,index) = K1(index,index) + ka;
 end
 %Apply boundary conditions for subs A
Keep = 2:sdof;
Ka = K1(Keep, Keep);
Ma = diag(m1);
 %Calculation of Substructure "A"s  mode shapes, natural freq, and
%physical C matrix
 [PhiA,LamA] = eig(Ka,Ma);
Wa = sqrt(LamA);
Check_MassNormA = PhiA'*Ma*PhiA;
CheckKa = PhiA'*Ka*PhiA;
 %Substructure A's C matrix
for i = 1:length(zetas1)
    Z_Wn_A(i,i)  = 2*Wa(i,i)*zetas1(1,i);
end
Ca = Ma*PhiA*Z_Wn_A*PhiA'*Ma;
CheckCa = PhiA'*Ca*PhiA;
 index = [0 , 1];
%Substructure B
for i = 1 : length(k2)
     index = index + 1 ;
     kb = [k2(1,i) , -k2(1,i) ; -k2(1,i) ,  k2(1,i)];
     K2(index,index) = K2(index,index) + kb ;
end
 %Apply boundary conditions for subs B
Keep = 1:sdof-1;
Kb = K2(Keep, Keep);
Mb = diag(m2);
 %Calculation of Substructure "B"s  mode shapes, natural freq, and
%physical C matrix
[PhiB,LamB] = eig(Kb,Mb);
Wb = sqrt(LamB);
Check_MassNormB = PhiB'*Mb*PhiB;
CheckKb = PhiB'*Kb*PhiB;
 %Substructure B's C matrix
for i = 1:length(zetas2)
    Z_Wn_B(i,i)  = 2*Wb(i,i)*zetas2(1,i);
end
Cb = Mb*PhiB*Z_Wn_B*PhiB'*Mb;
CheckCb = PhiB'*Cb*PhiB;
 % Solution
Pa1 = PhiA(:,1);
Pa2 = PhiA(:,2);
Pb1 = PhiB(:,1);
Pb2 = PhiB(:,2);
 Z_Wn_subs = zeros(sdof+1,sdof+1);
W_subs = zeros(sdof+1,sdof+1);
A =[Pa1(2,1) Pa2(2,1) -Pb1(1,1)  -Pb2(1,1)];
A = A'*A;
B = null(A);
```

```
A*B;
 index = [1,2];
Z_Wn_subs(index,index) = Z_Wn_subs(index,index) - Z_Wn_A;
W_subs(index,index) = W_subs(index,index) - LamA;

index = index + 2;
Z_Wn_subs(index,index) = Z_Wn_subs(index,index) - Z_Wn_B;
W_subs(index,index) = W_subs(index,index) - LamB;
 G = B'*Z_Wn_subs*B;
L = B'*W_subs*B;
 E = [zeros(3,3) , eye(3,3);
     L, G];
 LAM = eig(E)
 % eigenvalues of A are sigma(i) +/- j wd(i)
% sigma = zeta(i) * Wn(i)
% Wd(i) = Wn(i) * sqrt(1-zeta(i))
% two eqns in two unknowns
 sigma = real(LAM);
Wd = imag(LAM);
 for i = 1:3
    WD(i,1) = Wd(i*2-1,1);
    SIGMA(i,1) = sigma(i*2-1,1);
end
WD;
SIGMA;
WD = sort(WD);
SIGMA = sort(-SIGMA);
 for i = 1:sdof
    Zeta(i,1) = sqrt((((WD(i,1)^2/SIGMA(i,1)^2)+1)^-1));
end
for i = 1:sdof
    WN(i,1) = SIGMA(i,1)/Zeta(i,1);
end
WN ;
Zeta;
Table_Solution_WN_Zeta = [WN Zeta]
 %Check Natural Frequencies
%Building overall C, M, K matrices
C = zeros(sdof,sdof);
M = zeros(sdof,sdof);
K = zeros(sdof,sdof);
index = [1 , 2];
C(index,index) = C(index,index) + Ca;
M(index,index) = M(index,index) + Ma;
K(index,index) = K(index,index) + Ka;
index = index + 1;
C(index,index) = C(index,index) + Cb;
M(index,index) = M(index,index) + Mb;
K(index,index) = K(index,index) + Kb;
 [Phi,Lam] = eig(K,M);
for i =1:length(M)
    Check_Natfreq(i,1) = sqrt(Lam(i,i));
end
Check_Natfreq;
 % Check Zeta
C = Phi'*C*Phi;
for i = 1:length(M)
    Check_Zeta(i,1) = C(i,i)/(2*Check_Natfreq(i,1));
end
Check_Zeta;

Table_Check_Natfreq_Zeta = [Check_Natfreq Check_Zeta]
%%%%%%%%Why does eig(G)/2Wn = zeta for overall structure%%%%
[eigvecL,eigvalL] = eig(L);
Diag_K = eigvecL'*L*eigvecL;
Omega_N = sort(sqrt(diag(-eigvalL)));
[eigvecG,eigvalG] = eig(G);
ANewG = eigvecL'*G*eigvecL;
NewG = sort(-diag(ANewG));
 % eigNewG = eig(NewG)
% NewGa = eigvecL'*NewG*eigvecL
% G
% eigval_NewG = sort(-eig(NewG))
% eigvalG = sort(-diag(G))
% eigvecL;
 for i = 1:length(Omega_N);
    eigG_over_twoOmega_N(i,1) = NewG(i,1)/(2*Omega_N(i,1));
end
table_Omega_N_and_eigG_over_twoOmega_N = [Omega_N eigG_over_twoOmega_N]
```

A-4 COUPLING OF 2 ELEMENT BEAM STRUCTURE USING THE ZERO-EIGENVALUE THEOREM, WHICH GAVE THE BASIC FOUNDATION FOR THE OPTIMIZATION CODE

```
Keep = 1:4; %to keep all DOF multiply # elements by 2
 %Input for Substructure A
%Substructure A
 e=2;                          %# of elements
npe=2;                        %# nodes/element
dofpn=2;                      %# dof/node
tn = (e+1)*(npe-1);           %# total nodes of system
sysdof= tn*npe;               %total system dof
 Lenght = 10*12;               %Length of beam in Inches
A = 6*6;                      %X-sectional Area 2X2 inches
E = 10e6;                     %Elastic Modulus Aluminum
I = 1/12*2*2^3;               %Moment of Inertia about the z-axis
l = Lenght/e;                 %Element Length
RhoA = .00028497;                     %Density 2702 Kg/m^3
%Damping Ratios
zetasA = [.005,.01,.015,.02];
kk=zeros(sysdof,sysdof);
mm=zeros(sysdof,sysdof);
 for iel=1:e
    index=indexing_subA(iel,npe,dofpn);   %Build an index matrix to be used in
building the overall mass and stiffness matrices
    [k,m]=build_MK_element_subA(E,I,l,A,RhoA);  %Build Element matrix for M and K
    [kk,mm]=assemble_MK_matrix_subA(kk,k,mm,m,index);  %Assembly of M and K
matrices
end
 [Ka,Ma]=apply_BC_subA(kk,mm,sysdof); %Applying boundary condition for the beam
structure
 [PhiA,LamA] = eig(Ka,Ma);  %Calculate natural freq and mode shapes
WnA = sqrt(LamA);
Check_MassNorm = PhiA'*Ma*PhiA; %Check make sure this is equal to [I]
Diag_K = PhiA'*Ka*PhiA; %This should be diagonalized Ka matrix
 %Substructure A's C matrix
 [Z_Wn_A,Ca]=build_Ca(zetasA,WnA,PhiA,Ma); %Build physical damping matrix Ca and
modal damping matrix 2*zetaA*WnA
Check_Ca = PhiA'*Ca*PhiA; %This should be diagonalized Ca matrix
 %Input for Substructure B
%Substructure B
 e=2;                          %# of elements
npe=2;                        %# nodes/element
dofpn=2;                      %# dof/node
tn = (e+1)*(npe-1);           %# total nodes of system
sysdof= tn*npe;               %total system dof
 Lenght = 10*12;               %Length of beam in Inches
A = 6*6;                      %X-sectional Area 2X2 inches
E = 10e6;                     %Elastic Modulus Aluminum
I = 1/12*2*2^3;               %Moment of Inertia about the z-axis
l = Lenght/e;                 %Element Length
RhoB = .00028497;                    %Density 2702 Kg/m^3
%Damping Ratios
zetasB =[.005,.01,.015,.02];
 kk=zeros(sysdof,sysdof);
mm=zeros(sysdof,sysdof);
 for iel=1:e
    index=indexing_subB(iel,npe,dofpn); %Build an index matrix to be used in
building the overall mass and stiffness matrices
    [k,m]=build_MK_element_subB(E,I,l,A,RhoB);  %Build Element matrix for M and K
    [kk,mm]=assemble_MK_matrix_subB(kk,k,mm,m,index); %Assembly of M and K matrices
end
 [Kb,Mb]=apply_BC_subB(kk,mm,sysdof); %Applying boundary condition for the beam
structure
 [PhiB,LamB] = eig(Kb,Mb);  %Calculate natural freq and mode shapes
WnB = sqrt(LamB);
Check_MassNorm = PhiB'*Mb*PhiB; %Check make sure this is equal to [I]
Diag_K = PhiB'*Kb*PhiB;  %This should be diagonalized Ka matrix
 %Substructure B's C matrix
 [Z_Wn_B,Cb]=build_Cb(zetasB,WnB,PhiB,Mb); %Build physical damping matrix Cb and
modal damping matrix 2*zetaB*WnB
Check_Cb = PhiB'*Cb*PhiB; %This should be diagonalized Ca matrix
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%   Solution   %%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
 [L,G,Table_Solution_WN_Zeta] = Calc_WN_Zeta(Keep,PhiA,...
    PhiB,sysdof,Z_Wn_A,Z_Wn_B,LamA,LamB);
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%Check Natural Frequencies%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % Building overall C, M, K matrices and solving for Check_Zeta and
% Check_Natfreq
[K,M,C,Table_Check_Natfreq_Zeta]=build_overall_KMC(Ca,Ma,Ka,Cb,Mb,Kb,sysdof,npe,dof
pn,e);
 %Why does eig(G)/2Wn = zeta for overall structure
 [eigvecL,eigvalL] = eig(L);
Diag_K = eigvecL'*L*eigvecL;
Omega_N = sqrt(diag(-eigvalL));
NewG = eigvecL'*G*eigvecL;
NewG_soln = -diag(NewG);
 for i = 1:length(Omega_N);
    NewG_over_twoOmega_N(i,1) = NewG_soln(i,1)/(2*Omega_N(i,1));
end
 table_Omega_N_and_NewG_over_twoOmega_N = [Omega_N NewG_over_twoOmega_N]
```

A-4-1 Subprogram “indexing_subA”

```
function [index]=indexing_subA(iel,npe,dofpn)
 dof_e = npe*dofpn;
 initial = (iel-1)*(npe-1)*dofpn;
    for i=1:dof_e
      index(i)=initial+i;
   end
```

A-4-2 Subprogram “build_MK_element_subA”

```
function [k,m]=build_MK_element_subA(E,I,l,A,Rho)
  q=E*I/(l^3);
 k=q*[12      6*l   -12       6*l;...
      6*l  4*l^2 -6*l   2*l^2;...
      -12     -6*l    12      -6*l;...
      6*l  2*l^2 -6*l   4*l^2];
    r=Rho*A*l/420;
 m=r*[156      22*l   54       -13*l;...
      22*l  4*l^2  13*l  -3*l^2;...
      54       13*l   156      -22*l;...
      -13*l -3*l^2 -22*l   4*l^2];
```

A-4-3 Subprogram “assemble_MK_matrix_subA”

```
function [kk,mm]=assemble_MK_matrix_subA(kk,k,mm,m,index)
  Q = length(index);
 for i=1:Q
   ii=index(i);
     for j=1:Q
       jj=index(j);
         kk(ii,jj)=kk(ii,jj)+k(i,j);
     end
 end
  for i=1:Q
   ii=index(i);
     for j=1:Q
       jj=index(j);
         mm(ii,jj)=mm(ii,jj)+m(i,j);
     end
 end
```

A-4-4 Subprogram “apply_BC_subA”

```
function [Ka,Ma]=apply_BC_subA(kk,mm,sysdof)
 %Applying Boundary Conditions nodes 1 and 2 are both zero therefore
%only columns and rows 1 and 2 are zeroes
 Keep = 3:sysdof;
Ka = kk(Keep, Keep);
Ma = mm(Keep,Keep);
```

A-4-5 Subprogram “apply_BC_subA”

```
function [Z_Wn_A,Ca]=build_Ca(zetasA,WnA,PhiA,Ma)
 for i = 1:length(zetasA)
    Z_Wn_A(i,i)  = 2*WnA(i,i)*zetasA(1,i);
end
Ca = Ma*PhiA*Z_Wn_A*PhiA'*Ma;
```

A-4-6 Subprogram “Calc_WN_Zeta”

```
function [L,G,Table_Solution_WN_Zeta] =
Calc_WN_Zeta(Keep,PhiA,PhiB,sysdof,Z_Wn_A,Z_Wn_B,LamA,LamB)
 % Uncomment the following only if you want to compare
% more than 1 number of modes to be retained
% WN = zeros(2*(e-1)+length(Ma),length(MODE));
% Zeta = zeros(2*(e-1)+length(Ma),length(MODE));
% for MODE
%Want to use only the first 5 modes
% % Keep = 1:k;
```

```
 Z_Wn_subs = zeros(2*length(Keep),2*length(Keep)); % Block Diagonal Matrix of
2*Zeta*Wn from Substructure A and B
W_subs = zeros(2*length(Keep),2*length(Keep));  %Block Diagonal Matrix of Wn from
Substructure A and B
 % This portion will create an A matrix (Interconnecting Condition Between
% the Components), the constraint
 PhiA;                          %Eigen Vector of Substructure A, Mode shapes
Pa1 = PhiA(sysdof - 3,:);
Pa2 = PhiA(sysdof - 2,:);
PhiB;                          %Eigen Vector of Substructure B, Mode shapes
Pb1 = PhiB(1,:);
Pb2 = PhiB(2,:);
 % A = zeros(2,2*length(Keep));
A = [Pa1(:,Keep),-Pb1(:,Keep);
     Pa2(:,Keep),-Pb2(:,Keep)];

D = A'*A;
N = null(A);                   % Orthogonal Complement of D matrix
I=N'*N;                        % Make sure N'*N = [I]
zero=A*N;                      % Make sure A*N = 0
 %This will build Z_Wn_subs and W_subs Block Diagonals, inputting values from each
substructure
index = [1:length(Keep)];
Z_Wn_subs(index,index) = Z_Wn_subs(index,index) - Z_Wn_A(Keep,Keep);
W_subs(index,index) = W_subs(index,index) - LamA(Keep,Keep);
 index = index + length(Keep);
Z_Wn_subs(index,index) = Z_Wn_subs(index,index) - Z_Wn_B(Keep,Keep);
W_subs(index,index) = W_subs(index,index) - LamB(Keep,Keep);
 %This step will convert my diagonal block matrix into my overall structures
%Modal Damping matrix (2ZetaWn) and Modal matrix of eigenvalues and eigenvectors
G = N'*Z_Wn_subs*N;     %overall structure
L = N'*W_subs*N;        %overall structure
 % E is a matrix of first order differential equations
 E = [zeros(2*length(Keep)-2,2*length(Keep)-2), eye(2*length(Keep)-
2,2*length(Keep)-2);
      L, G];
 LAM = eig(E);
 % eigenvalues of E are sigma(i) +/- j wd(i)
% sigma = zeta(i) * Wn(i)
% Wd(i) = Wn(i) * sqrt(1-zeta(i))
% two eqns and two unknowns
% Portion Provided by Prof Gordis
     sigma = real(LAM);
    Wd = imag(LAM);

    for i = 1:2*length(Keep)-2
    WD(i,1) = Wd(i*2-1,1);
    SIGMA(i,1) = sigma(i*2-1,1);
    end
%     WD = sort(WD);  %undamped natural frequency
%     SIGMA = sort(-SIGMA);
     % replace 1 in the following with k-2 if comparing
    % multiple modes Zeta(i,1) and WN(i,1)
     for i = 1:2*length(Keep)-2
    Zeta(i,1) = sqrt((((WD(i,1)^2/SIGMA(i,1)^2)+1)^-1));
    end
    for i = 1:2*length(Keep)-2
        WN(i,1) = -SIGMA(i,1)/Zeta(i,1);
    end
 % Table of natural frequency and Zeta SOLUTION
 Table_Solution_WN_Zeta = [WN Zeta]
A-4-7  Subprogram “build_overall_KMC”
function
[K,M,C,Table_Check_Natfreq_Zeta]=build_overall_KMC(Ca,Ma,Ka,Cb,Mb,Kb,sysdof,npe,dof
pn,e);
C = zeros(4*e-2,4*e-2);
M = zeros(4*e-2,4*e-2);
K = zeros(4*e-2,4*e-2);
 dof_e = e*2;
initial = 0;
   for i=1:dof_e
      index(i)=initial+i;
   end
 C(index,index) = C(index,index) + Ca;
M(index,index) = M(index,index) + Ma;
K(index,index) = K(index,index) + Ka;
index = index + (2*e-2);
C(index,index) = C(index,index) + Cb;
M(index,index) = M(index,index) + Mb;
```

```
K(index,index) = K(index,index) + Kb;
 [Phi,Lam] = eig(K,M);
 for i =1:length(M)
    Check_Natfreq(i,1) = sqrt(Lam(i,i));
end
Check_Natfreq;
 % Check Zeta
C = Phi'*C*Phi;
for i = 1:length(M)
    Check_Zeta(i,1) = C(i,i)/(2*Check_Natfreq(i,1));
end
Check_Zeta;
 %Table Natural frequencies and Zetas to be compared to the solution
Table_Check_Natfreq_Zeta = [Check_Natfreq Check_Zeta]
```

A-4-7 Subprogram “System”

```
%Second Check for the System
 e=4;                              %# of elements
npe=2;                            %# nodes/element
dofpn=2;                          %# dof/node
tn = (e+1)*(npe-1);               %# total nodes of system
sysdof= tn*npe;                   %total system dof

Lenght = 20*12;                   %Length of beam in Inches
A = 6*6;                          %X-sectional Area 2X2 inches
E = 10e6;                         %Elastic Modulus Aluminum
I = 1/12*2*2^3;                   %Moment of Inertia about the z-axis
l = Lenght/e;                     %Element Length
RhoA = .00028497;                 %Density 2702 Kg/m^3
 kk=zeros(sysdof,sysdof);
mm=zeros(sysdof,sysdof);
 for iel=1:e
    index=indexing_subA(iel,npe,dofpn);   %Build an index matrix to be used in
building the overall mass and stiffness matrices
    [k,m]=build_MK_element_subA(E,I,l,A,RhoA);  %Build Element matrix for M and K
    [kk,mm]=assemble_MK_matrix_subA(kk,k,mm,m,index);  %Assembly of M and K
matrices
end
 Keep = 3:sysdof;
Ka = kk(Keep, Keep);
Ma = mm(Keep,Keep);
 Keep = 1:length(Ka)-2
Kone = Ka(Keep,Keep)
Mone = Ma(Keep,Keep)
```

A-5 DECOUPLING

```
Keep = 1:6; %to keep all DOF multiply # elements by 2
 %Input for Substructure A
%Substructure A
 e=3;                          %# of elements
npe=2;                        %# nodes/element
dofpn=2;                      %# dof/node
tn = (e+1)*(npe-1);           %# total nodes of system
sysdof= tn*npe;               %total system dof
Lenght = 10*12;               %Length of beam in Inches
A = 2*2;                      %X-sectional Area 2X2 inches
E = 10e6;                     %Elastic Modulus Aluminum
I = 1/12*2*2^3;               %Moment of Inertia about the z-axis
l = Lenght/e;                 %Element Length
RhoA = .1;                    %Density 2702 Kg/m^3
 kk=zeros(sysdof,sysdof);
mm=zeros(sysdof,sysdof);
 for iel=1:e
    index=indexing_subA(iel,npe,dofpn);   %Build an index matrix to be used in
building the overall mass and stiffness matrices
    [k,m]=build_MK_element_subA(E,I,l,A,RhoA);  %Build Element matrix for M and K
    [kk,mm]=assemble_MK_matrix_subA(kk,k,mm,m,index);  %Assembly of M and K
matrices
end
 [Ka,Ma]=apply_BC_subA(kk,mm,sysdof); %Applying boundary condition for the beam
structure
 [PhiA,LamA] = eig(Ka,Ma);  %Calculate natural freq and mode shapes
WnA = sqrt(LamA);
Check_MassNorm = PhiA'*Ma*PhiA; %Check make sure this is equal to [I]
Diag_K = PhiA'*Ka*PhiA; %This should be diagonalized Ka matrix
 %%%%%%%%%%%%%%%%%%%%Input for Substructure B%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%Substructure B%%%%%%%%%%%%%%%%%%%%%%%%%%
 e=3;                          %# of elements
npe=2;                        %# nodes/element
dofpn=2;                      %# dof/node
tn = (e+1)*(npe-1);           %# total nodes of system
sysdof= tn*npe;               %total system dof
 Lenght = 10*12;               %Length of beam in Inches
A = 2*2;                      %X-sectional Area 2X2 inches
E = 10e6;                     %Elastic Modulus Aluminum
I = 1/12*2*2^3;               %Moment of Inertia about the z-axis
l = Lenght/e;                 %Element Length
RhoB = .1;                     %Density 2702 Kg/m^3
 kk=zeros(sysdof,sysdof);
mm=zeros(sysdof,sysdof);
 for iel=1:e
    index=indexing_subB(iel,npe,dofpn); %Build an index matrix to be used in
building the overall mass and stiffness matrices
    [k,m]=build_MK_element_subB(E,I,l,A,RhoB);  %Build Element matrix for M and K
    [kk,mm]=assemble_MK_matrix_subB(kk,k,mm,m,index); %Assembly of M and K matrices
end
 [Kb,Mb]=apply_BC_subB(kk,mm,sysdof); %Applying boundary condition for the beam
structure
 [PhiB,LamB] = eig(Kb,Mb);  %Calculate natural freq and mode shapes
WnB = sqrt(LamB);
Check_MassNorm = PhiB'*Mb*PhiB; %Check make sure this is equal to [I]
Diag_K = PhiB'*Kb*PhiB;  %This should be diagonalized Ka matrix
 %Substructure B's C matrix
 % [Z_Wn_B,Cb]=build_Cb(zetasB,WnB,PhiB,Mb); %Build physical damping matrix Cb and
modal damping matrix 2*zetaB*WnB
% Check_Cb = PhiB'*Cb*PhiB; %This should be diagonalized Ca matrix
 % Uncomment the following only if you want to compare
% more than 1 number of modes to be retained
% WN = zeros(2*(e-1)+length(Ma),length(MODE));
% Zeta = zeros(2*(e-1)+length(Ma),length(MODE));
% for MODE
%Want to use only the first 5 modes
% % Keep = 1:k;
 W_subs = zeros(2*length(Keep),2*length(Keep));  %Block Diagonal Matrix of Wn from
Substructure A and B
 % This portion will create an A matrix (Interconnecting Condition Between
% the Components), the constraint
 PhiA;                          %Eigen Vector of Substructure A, Mode shapes
Pa1 = PhiA(sysdof - 3,:);
Pa2 = PhiA(sysdof - 2,:);
```

```
PhiB;                          %Eigen Vector of Substructure B, Mode shapes
Pb1 = PhiB(1,:);
Pb2 = PhiB(2,:);
A = [Pa1(:,Keep),-Pb1(:,Keep);
     Pa2(:,Keep),-Pb2(:,Keep)];
D = A'*A
N = null(D)                  % Orthogonal Complement of D matrix
In=N'*N                      % Make sure N'*N = [I]
zero=D*N                     % Make sure A*N = 0
 %This will build W_subs Block Diagonals, inputting values from each substructure
index = [1:length(Keep)];
W_subs(index,index) = W_subs(index,index) - LamA(Keep,Keep);
index = index + length(Keep);
W_subs(index,index) = W_subs(index,index) - LamB(Keep,Keep);
 % %This step will convert my diagonal block matrix into my system structures
L = N'*W_subs*N;        %system structure
  %%%%%%%%%find natural freq for system%%%%%%%%%%%%
[eigvecL,eigvalL] = eig(L)
WN = sort(diag(sqrt(-eigvalL)))
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%   Solution   %%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %Generate Values for system Zeta
a = .03;
b = .01;
modes = 10;
Zeta = a + (b-a).*rand(modes,1);
 %Build system damping matrix
for i = 1:modes
    TWO_ZETA_WN_SYSTEM(i,i) = 2*Zeta(i,1)*WN(i,1);
end
TWO_ZETA_WN_SYSTEM = eigvecL*TWO_ZETA_WN_SYSTEM*eigvecL'
%Find substructures Modal damping matrices
Two_Zeta_Wn_A_B = N*TWO_ZETA_WN_SYSTEM*N';
WnA = diag(sort(diag(WnA), 'descend'));
WnB = diag(sort(diag(WnA), 'descend'));
 %Ignoring the off diagonals assume proportionally damped substructures
Asubs =zeros(6,6); Bsubs=zeros(6,6)
index = [1:length(Keep)];
Asubs(index,index) = Two_Zeta_Wn_A_B(index,index)
save = [7:length(Keep)*2]
Bsubs(index,index) = Two_Zeta_Wn_A_B(7:12,7:12)
 % Ra = PhiA'*Asubs*PhiA
% Rb = PhiB'*Bsubs*PhiB
for i = 1:length(Keep)
    zetasA(i,i) =  Asubs(i,i)/(2*WnA(i,i))
    zetasB(i,i) =  Bsubs(i,i)/(2*WnB(i,i))
end
 Zeta
ZetasA = diag(zetasA)
ZetasB = diag(zetasB)
 % [PhiA,LamA] = eig(Ka,Ma);  %Calculate natural freq and mode shapes
% WnA = sqrt(LamA);
% [PhiB,LamB] = eig(Kb,Mb);  %Calculate natural freq and mode shapes
% WnB = sqrt(LamB)
 for i = 1:length(zetasA)
    Z_Wn_A(i,i)  = 2*WnA(i,i)*zetasA(i,i);
    Z_Wn_B(i,i)  = 2*WnB(i,i)*zetasB(i,i);
end
  %Check if it is almost close to system damping
 Z_Wn_subs = zeros(2*length(Keep),2*length(Keep));
index = [1:length(Keep)];
Z_Wn_subs(index,index) = Z_Wn_subs(index,index) + Z_Wn_A(Keep,Keep);
index = index + length(Keep);
Z_Wn_subs(index,index) = Z_Wn_subs(index,index) + Z_Wn_B(Keep,Keep);
 G = N'*Z_Wn_subs*N;
sol = diag(G)
 TWO_ZETA_WN_SYSTEM;
check = diag(TWO_ZETA_WN_SYSTEM)
```

The subprograms used for this decoupling program is the same as the one used for the beam coupling.

A-6 FMINCON BEAM

```
global modes e Uppersys Lowersys
 % Lenght = 10*12;              %Length of beam in Inches
% A = 6*6;                     %X-sectional Area 6X6 inches
% E = 10e6;                    %Elastic Modulus Aluminum
% I = 1/12*2*2^3;              %Moment of Inertia about the z-axis
% l = Lenght/e;                %Element Length
%Rho = .00028497;              %Density
 % disp('ZetaA & ZetaB (damping ratios for each substructure)')
% disp('Find bounds for ZetaA & ZetaB that would meet the prescribed constraint for
the System damping')
% e = input('Enter number of elements for each Substructure:')
e = 20
 % disp('Inputing desired damping ratios for each substructure')
% init_damping = input('Enter Initial value for substructure damping typically
between .02 and .2:')
init_damping =.2
 % disp('Upper and Lower bound for each substructure')
% Uppersubs = input('Enter Upper bound for substructure damping:')
% Lowersubs = input('Enter Lower bound for substructure damping:')
Uppersubs = .060
Lowersubs = .045
 % disp('Prescribed bounds for system damping')
% Uppersys = input('Enter Upper bound for sys damping:')
% Lowersys = input('Enter Lower bound for sys damping:')
Uppersys = .045
Lowersys = .040

% select = input( 'To use all modes enter 1.  If not enter 2:') ;
%
% if select == 1;
%     modes = e*2
%
% else
%     disp('Inputing desired modes kept')
%     modes = input('Input # of modes to be Kept:')
% end
modes = 20
 zetas_init = zeros(modes,1);
for i = 1:modes
    zetas_init(i,1) = init_damping;
end
zetasA = zetas_init;
 A=[];
b=[];
Aeq=[];
beq=[];
for i = 1:modes
    lb(i,1) = Lowersubs;
    ub(i,1) = Uppersubs;
end
 [f,fval,exitflag,output] = fmincon(@objfun,zetasA,A,b,Aeq,beq,lb,ub,@cons)
 zetasA = f';
zetasB = f';

%Input for Substructure A
%Substructure A
 Keep = 1:modes;
npe=2;                        %# nodes/element
dofpn=2;                      %# dof/node
tn = (e+1)*(npe-1);           %# total nodes of system
sysdof= tn*npe;               %total system dof
 Lenght = 10*12;               %Length of beam in Inches
A = 6*6;                      %X-sectional Area 2X2 inches
E = 10e6;                     %Elastic Modulus Aluminum
I = 1/12*2*2^3;               %Moment of Inertia about the z-axis
l = Lenght/e;                 %Element Length
RhoA = .00028497;             %Density
kk=zeros(sysdof,sysdof);
mm=zeros(sysdof,sysdof);
 for iel=1:e
    index=indexing_subA(iel,npe,dofpn);   %Build an index matrix to be used in
building the overall mass and stiffness matrices
    [k,m]=build_MK_element_subA(E,I,l,A,RhoA);  %Build Element matrix for M and K
```

```
    [kk,mm]=assemble_MK_matrix_subA(kk,k,mm,m,index);  %Assembly of M and K
matrices
end
 [Ka,Ma]=apply_BC_subA(kk,mm,sysdof); %Applying boundary condition for the beam
structure
 [PhiA,LamA] = eig(Ka,Ma);  %Calculate natural freq and mode shapes
WnA = sqrt(LamA);
Check_MassNorm = PhiA'*Ma*PhiA; %Check make sure this is equal to [I]
Diag_K = PhiA'*Ka*PhiA; %This should be diagonalized Ka matrix

%Substructure A's C matrix
if modes == 2*e
    [Z_Wn_A,Ca]=build_Ca(zetasA,WnA,PhiA,Ma); %Build physical damping matrix Ca and
modal damping matrix 2*zetaA*WnA
    Check_Ca = PhiA'*Ca*PhiA; %This should be diagonalized Ca matrix
else
    [Z_Wn_A]=cons_Z_Wn_A(zetasA,WnA); %Build physical damping matrix Ca and modal
damping matrix 2*zetaA*WnA
end
 %Input for Substructure B
npe=2;                          %# nodes/element
dofpn=2;                        %# dof/node
tn = (e+1)*(npe-1);             %# total nodes of system
sysdof= tn*npe;                 %total system dof

Lenght = 10*12;                 %Length of beam in Inches
A = 6*6;                        %X-sectional Area 2X2 inches
E = 10e6;                       %Elastic Modulus Aluminum
I = 1/12*2*2^3;                 %Moment of Inertia about the z-axis
l = Lenght/e;                   %Element Length
RhoB = .00028497;               %Density
kk=zeros(sysdof,sysdof);
mm=zeros(sysdof,sysdof);
 for iel=1:e
    index=indexing_subB(iel,npe,dofpn); %Build an index matrix to be used in
building the overall mass and stiffness matrices
    [k,m]=build_MK_element_subB(E,I,l,A,RhoB);  %Build Element matrix for M and K
    [kk,mm]=assemble_MK_matrix_subB(kk,k,mm,m,index); %Assembly of M and K matrices
end
 [Kb,Mb]=apply_BC_subB(kk,mm,sysdof); %Applying boundary condition for the beam
structure
 [PhiB,LamB] = eig(Kb,Mb);  %Calculate natural freq and mode shapes
WnB = sqrt(LamB);
Check_MassNorm = PhiB'*Mb*PhiB; %Check make sure this is equal to [I]
Diag_K = PhiB'*Kb*PhiB;  %This should be diagonalized Ka matrix
 %Substructure B's C matrix
if modes == 2*e
    [Z_Wn_B,Cb]=build_Cb(zetasB,WnB,PhiB,Mb); %Build physical damping matrix Cb and
modal damping matrix 2*zetaB*WnB
    Check_Cb = PhiB'*Cb*PhiB; %This should be diagonalized Ca matrix
else
    [Z_Wn_B]=cons_Z_Wn_B(zetasB,WnB); %Build physical damping matrix Cb and modal
damping matrix 2*zetaB*WnB
end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%   Solution   %%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 Z_Wn_subs = zeros(2*length(Keep),2*length(Keep)); % Block Diagonal Matrix of
2*Zeta*Wn from Substructure A and B
W_subs = zeros(2*length(Keep),2*length(Keep));  %Block Diagonal Matrix of Wn from
Substructure A and B
 % This portion will create an A matrix (Interconnecting Condition Between
% the Components), the constraint
 PhiA;                           %Eigen Vector of Substructure A, Mode shapes
Pa1 = PhiA(sysdof - 3,:);
Pa2 = PhiA(sysdof - 2,:);
PhiB;                           %Eigen Vector of Substructure B, Mode shapes
Pb1 = PhiB(1,:);
Pb2 = PhiB(2,:);
 % A = zeros(2,2*length(Keep));
A = [Pa1(:,Keep),-Pb1(:,Keep);
     Pa2(:,Keep),-Pb2(:,Keep)];
 D = A'*A;
N = null(A);                    % Orthogonal Complement of D matrix
I=N'*N;                         % Make sure N'*N = [I]
zero=A*N;                       % Make sure A*N = 0
 %This will build Z_Wn_subs and W_subs Block Diagonals, inputting values from each
substructure
index = [1:length(Keep)];
```

```
Z_Wn_subs(index,index) = Z_Wn_subs(index,index) - Z_Wn_A(Keep,Keep);
W_subs(index,index) = W_subs(index,index) - LamA(Keep,Keep);
 index = index + length(Keep);
Z_Wn_subs(index,index) = Z_Wn_subs(index,index) - Z_Wn_B(Keep,Keep);
W_subs(index,index) = W_subs(index,index) - LamB(Keep,Keep);
 %This step will convert my diagonal block matrix into my overall structures
%Modal Damping matrix (2ZetaWn) and Modal matrix of eigenvalues and eigenvectors
G = N'*Z_Wn_subs*N;     %overall structure
L = N'*W_subs*N;        %overall structure
 %solving using a L and G%%%%% METHOD 3 IS FASTER THAN METHOD 2
 [eigvecL,eigvalL] = eig(L);
Diag_K = eigvecL'*L*eigvecL;
WN = sqrt(diag(-eigvalL));
NewG = eigvecL'*G*eigvecL;
NewG_soln = -diag(NewG);

for i = 1:length(WN);
    Zetasys(i,1) = NewG_soln(i,1)/(2*WN(i,1));
end
Table_Solution = [WN, Zetasys];
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%Check Natural Frequencies%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if modes == 2*e
    % Building overall C, M, K matrices and solving for Check_Zeta and
    % Check_Natfreq
    [K,M,C,Table_Check]=build_overall_KMC(Ca,Ma,Ka,Cb,Mb,Kb,sysdof,npe,dofpn,e);
else
    NOCHECK = 1
end
zetas_initmax = max(zetas_init)
zetas_initmin = min(zetas_init)
zetasA_max = max(f)
zetasA_min = min(f)
Zetasys_max = max(Zetasys)
Zetasys_min = min(Zetasys)
```

A-6-1 Subprogram “Cons”

```
function [c,ceq] = cons(zetasA)
 global modes e Uppersys Lowersys
 zetasA = zetasA';
zetasB = zetasA;
Keep = 1:modes;
 %%%%%%%%%%%%%%%%%%%%Input for Substructure A%%%%%%%%%%%%%%%%
npe=2;                          %# nodes/element
dofpn=2;                        %# dof/node
tn = (e+1)*(npe-1);             %# total nodes of system
sysdof= tn*npe;                 %total system dof
 Lenght = 10*12;                 %Length of beam in Inches
A = 6*6;                        %X-sectional Area 2X2 inches
E = 10e6;                       %Elastic Modulus Aluminum
I = 1/12*2*2^3;                 %Moment of Inertia about the z-axis
l = Lenght/e;                   %Element Length
RhoA = .00028497;               %Density
kk=zeros(sysdof,sysdof);
mm=zeros(sysdof,sysdof);
 for iel=1:e
    index=indexing_subA(iel,npe,dofpn);   %Build an index matrix to be used in
building the overall mass and stiffness matrices
    [k,m]=build_MK_element_subA(E,I,l,A,RhoA);  %Build Element matrix for M and K
    [kk,mm]=assemble_MK_matrix_subA(kk,k,mm,m,index);  %Assembly of M and K
matrices
end
 [Ka,Ma]=apply_BC_subA(kk,mm,sysdof); %Applying boundary condition for the beam
structure
 [PhiA,LamA] = eig(Ka,Ma);  %Calculate natural freq and mode shapes
WnA = sqrt(LamA);
Check_MassNorm = PhiA'*Ma*PhiA; %Check make sure this is equal to [I]
Diag_K = PhiA'*Ka*PhiA; %This should be diagonalized Ka matrix
 %Substructure A's C matrix
 [Z_Wn_A]=cons_Z_Wn_A(zetasA,WnA); %Build physical damping matrix Ca and modal
damping matrix 2*zetaA*WnA
 %%%%%%%%%%%%%%%%%%%%Input for Substructure B%%%%%%%%%%%%%%%%
npe=2;                          %# nodes/element
```

```
dofpn=2;                        %# dof/node
tn = (e+1)*(npe-1);             %# total nodes of system
sysdof= tn*npe;                 %total system dof
 Lenght = 10*12;                 %Length of beam in Inches
A = 6*6;                        %X-sectional Area 2X2 inches
E = 10e6;                       %Elastic Modulus Aluminum
I = 1/12*2*2^3;                 %Moment of Inertia about the z-axis
l = Lenght/e;                   %Element Length
RhoB = .00028497;               %Density
kk=zeros(sysdof,sysdof);
mm=zeros(sysdof,sysdof);
 for iel=1:e
    index=indexing_subB(iel,npe,dofpn); %Build an index matrix to be used in
building the overall mass and stiffness matrices
    [k,m]=build_MK_element_subB(E,I,l,A,RhoB);  %Build Element matrix for M and K
    [kk,mm]=assemble_MK_matrix_subB(kk,k,mm,m,index); %Assembly of M and K matrices
end
 [Kb,Mb]=apply_BC_subB(kk,mm,sysdof); %Applying boundary condition for the beam
structure
 [PhiB,LamB] = eig(Kb,Mb);  %Calculate natural freq and mode shapes
WnB = sqrt(LamB);
Check_MassNorm = PhiB'*Mb*PhiB; %Check make sure this is equal to [I]
Diag_K = PhiB'*Kb*PhiB;  %This should be diagonalized Ka matrix
 %Substructure B's C matrix
 [Z_Wn_B]=cons_Z_Wn_B(zetasB,WnB); %Build physical damping matrix Cb and modal
damping matrix 2*zetaB*WnB
 %%%%%%%%%%   Solution   %%%%%%%%%%%%
Z_Wn_subs = zeros(2*length(Keep),2*length(Keep)); % Block Diagonal Matrix of
2*Zeta*Wn from Substructure A and B
W_subs = zeros(2*length(Keep),2*length(Keep));  %Block Diagonal Matrix of Wn from
Substructure A and B
 % This portion will create an A matrix (Interconnecting Condition Between
% the Components), the constraint
 PhiA;                           %Eigen Vector of Substructure A, Mode shapes
Pa1 = PhiA(sysdof - 3,:);
Pa2 = PhiA(sysdof - 2,:);
PhiB;                           %Eigen Vector of Substructure B, Mode shapes
Pb1 = PhiB(1,:);
Pb2 = PhiB(2,:);
 % A = zeros(2,2*length(Keep));
A = [Pa1(:,Keep),-Pb1(:,Keep);
     Pa2(:,Keep),-Pb2(:,Keep)];
 D = A'*A;
N = null(A);                    % Orthogonal Complement of D matrix
I=N'*N;                         % Make sure N'*N = [I]
zero=A*N;                       % Make sure A*N = 0
 %This will build Z_Wn_subs and W_subs Block Diagonals, inputting values from each
substructure
index = [1:length(Keep)];
Z_Wn_subs(index,index) = Z_Wn_subs(index,index) - Z_Wn_A(Keep,Keep);
W_subs(index,index) = W_subs(index,index) - LamA(Keep,Keep);
 index = index + length(Keep);
Z_Wn_subs(index,index) = Z_Wn_subs(index,index) - Z_Wn_B(Keep,Keep);
W_subs(index,index) = W_subs(index,index) - LamB(Keep,Keep);
 %This step will convert my diagonal block matrix into my overall structures
%Modal Damping matrix (2ZetaWn) and Modal matrix of eigenvalues and eigenvectors
G = N'*Z_Wn_subs*N;     %overall structure
L = N'*W_subs*N;        %overall structure
 %%%%solving using a L and G%%%%% METHOD 3 FASTER THAN METHOD 2
 [eigvecL,eigvalL] = eig(L);
Diag_K = eigvecL'*L*eigvecL;
WN = sqrt(diag(-eigvalL));
NewG = eigvecL'*G*eigvecL;
NewG_soln = -diag(NewG);
 for i = 1:length(WN);
    Zetasys(i,1) = NewG_soln(i,1)/(2*WN(i,1));
end
 lb_sys = Lowersys;
ub_sys = Uppersys;
 Zetaq = Zetasys;
Zetar = Zetasys;
c = zeros(length(Zetasys)*2,1);
for i = 1:length(Zetasys)*2;
    if i < length(Zetasys)+1;
        c(i,1) = Zetaq(i,1) - ub_sys;
    else
        c(i,1) = -Zetar(length(Zetasys),1) + lb_sys;
    end
end
```

```
ceq=[];
```

A-6-2 Subprogram “objfun”

```
function [f,fval,exitflag,output] = objfun(x)
global modes
 f = 0;
for i = 1:modes
    f = f - x(i);
end
```

A-6-3 Subprogram “cons_Z_Wn_A”

```
function [Z_Wn_A]=build_Ca(zetasA,WnA)
 for i = 1:length(zetasA)
    Z_Wn_A(i,i)  = 2*WnA(i,i)*zetasA(1,i);
end
```

The subprograms used for this FMINCON_BEAM program is the same as the one used for the beam coupling.

A-7 COMAPARING THE LOADWORK IN CALCULATING THE EIGENVALUES OF MATRIX SIZES N, 2N, 4N, 8N AND 16N

```
%Code Written By Prof Gordis
clear
clc
 n = 50;
 a = rand(n,n);
a = (a + a')/2;     %  create symmetic a
tic;[phi,lam] = eig(a); t1 = toc;
 n = 2 * n;
a = rand(n,n);
a = (a + a')/2;     %  create symmetic a
tic;[phi,lam] = eig(a); t2 = toc;
 n = 2 * n;
a = rand(n,n);
a = (a + a')/2;     %  create symmetic a
tic;[phi,lam] = eig(a); t3 = toc;
 n = 2 * n;
a = rand(n,n);
a = (a + a')/2;     %  create symmetic a
tic;[phi,lam] = eig(a); t4 = toc;
 n = 2 * n;
a = rand(n,n);
a = (a + a')/2;     %  create symmetic a
tic;[phi,lam] = eig(a); t5 = toc;
 tt = [t1 t2 t3 t4 t5]
tt = tt/tt(5);


nn = [n^3 (2*n)^3 (4*n)^3   (8*n)^3 (16*n)^3]
nn = nn/nn(5);
 [tt;nn]
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Howard Anton and Chris Rorres, Elementary Linear Algebra. New Jersey: John Wiley & Sons, 2005.
2. Klaus-Jurgen Bathe, Finite Element Procedures. New Jersey: Prentice Hall, 1996. Dowell, E. H., "Free Vibrations of a Linear Structure with Arbitrary Support Conditions," Journal of Applied Mechanics. (38), Transactions ASME. 93(1971), 595-600.
3. E.H. Dowell, "Free Vibrations of an Arbitrary Structure in Terms of Component modes," Journal of Applied Mechanics. (39), Transactions ASME. 94(1972), 727-731.
4. J.H. Gordis, "Integral Equation Formulation for Transient Structural Synthesis," AIAA Journal. 33(1995), 320-324
5. S.J. Fuh, and S.Y. Chen, "Constraints of the Structural Modal Synthesis," AIAA Journal. 24(1986), 1045-1047.
6. Abdul J. Jerri, Introduction to Integral Equations with Applications. 2nd ed. New York: John Wiley & Sons INC, 1999.
7. J.W. Kamman, and R L Huston, "Dynamics of Constrained Multibody Systems," Journal of Applied Mechanics, Transactions ASME. 51(1984), 899-903.
8. L.R. Klein and E.H. Dowell, "Analysis of Modal Damping by Component Modes Method Using Lagrange Multipliers," Journal of Applied Mechanics, Transaction ASME. 41(1974), 527-528.
9. J.H. Kuang and Y.G. Tsuei, "A More General Method of Substructure Mode Synthesis for Dynamic Analysis," AIAA Journal. 23(1985), 618-623.
10. K. Kubomura, "Substructure Model Synthesis for Damped Structures," AIAA 23rd Aerospace Sciences Meeting. (1985), 1-9.

11. Young Kwon and Hyochoon Bang, The Finite Element Method Using MATLAB. Florida: CRC Press, 2000.

12. Leonard Meirovitch, Computational Methods in Structural Dynamics. The Netherlands: Sijthoff & Noordhoff, 1980.

13. W.H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Numerical Recipes in FORTRAN. Cambridge: Cambridge University Press, 1992.

14. Gilbert Strang, Linear Algebra and Its Applications. New York: Academic Press Inc, 1976.

15. F.E. Udwadia and Ravi Kumar, "Iterative Methods For Non-Classically Damped Dynamic System," Earthquake Engineering And Structural Dynamics. 23(1994), 137-152.

16. W.C. Walton and E.C. Steeves, "A New Matrix Theorem and its Application for Establishing Independent Coordinates for Complex Dynamical Systems with Constraints," NASA Technical Report. NASA TR R-326 (1969).

17. K. Xu and Takeru Igusa, "Dynamic Characteristics of Non-Classically Damped Structures," Earthquake Engineering and Structural Dynamics. 20 (1991), 1127-1144.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor Joshua H. Gordis, Code ME/Go
   Naval Postgraduate School
   Monterey, California

4. LT Wendel D. Penetrante
   Chula Vista, California