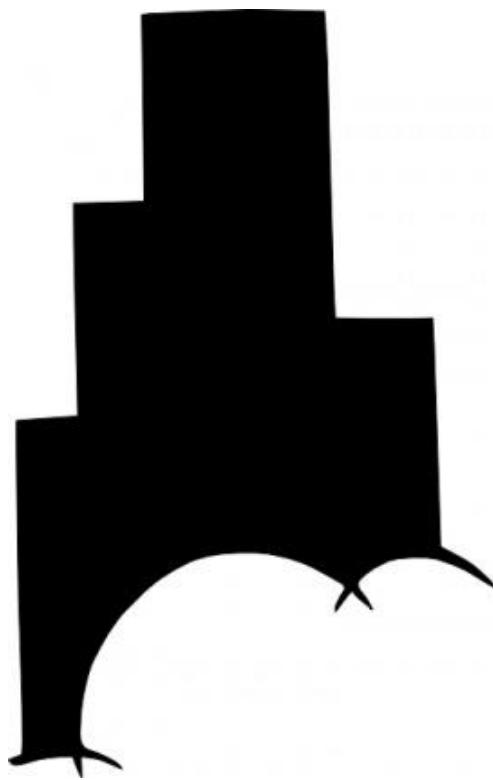


# STRUCTURAL DYNAMICS PORTFOLIO



CEE 536 - STRUCTURAL DYNAMICS

MICHAEL JUSTICE

May. 4, 15

## Table of Contents

The Single Degree of Freedom System (The Oscillator, Homework 1).....	4
Summary .....	4
The Derivation of the SDOF Oscillator with Free Vibration .....	4
Coulomb Friction Damping .....	6
Determining the Damping Constant .....	8
The Damped SDOF (Homework 2) .....	10
Summary .....	10
The Derivation of the Damped SDOF Forced Oscillator System (Constant and Sinusoidal).....	10
The Central Difference Method (Homework 3) .....	15
Summary .....	15
The Derivation of the Exact Solution to the Difference Equation (Analytical Solution) .....	15
The Time Stepping Algorithm (Numerical Solution) .....	16
Comparison Between the Numerical, Classical and Analytical Solution .....	16
Newmark's Method (Homework 4) .....	19
Summary .....	19
The Derivation of the Exact Solution to the Numerical Solution (Analytical Solution) .....	19
The Time Stepping Algorithm (Numerical Solution) .....	20
Comparison Between the Numerical, Classical and Analytical Solution .....	21
Newton's Method with Nonlinear Restoring Forces (Homework 5) .....	23
Summary .....	23
Newton's Method .....	23
Implementing Newton's Method .....	23
Elasto-Plastic SDOF (Homework 6) .....	28
Summary .....	28
The Elasto-Plastic Response.....	28
The NDOF System (Homework 7) .....	31
Summary .....	31
The Equations of Motion .....	31
The Building and Bridge EOM .....	31

Approximating the Time-Dependent Variables .....	33
The Dynamic Response .....	33
Damping of the NDOF System (Homework 8) .....	40
Summary .....	40
The Implementation of Applied Damping .....	40
The Effects of Applied Damping.....	41
Elasto-Plastic Response of the NDOF System (Homework 9) .....	45
Summary .....	45
The Restoring Force .....	45
Elasto-Plasticity and Nonlinearity Implementation .....	46
The Elasto-Plastic Response.....	47
Earthquake for the NDOF System (Homework 10).....	50
Summary .....	50
The Earthquake and EOM .....	50
The Earthquake Response.....	51
The Static and Dynamic Truss (Homework 11 and 13) .....	55
Summary .....	55
A Breakdown of the Truss Program .....	55
Going from Linear to Nonlinear, .....	56
Implementing Elasto-Plasticity .....	57
Observing the Response .....	58
The Dynamics.....	61
The Geometry of the Camelback Truss Bridge .....	62
The Motion.....	67
The Ritz Vectors (Homework 12) .....	80
Summary .....	80
The Ritz Vector.....	80
Gram-Schmidt .....	81
Subspace Iteration .....	81
Ritz and Eigenvector Comparison .....	82
Axial Wave Propagation (Homework 15).....	84

Summary .....	84
The Discretization Requirements.....	84
The Boundary Conditions.....	85
Mode Shapes and Resonance .....	86
Flexural Wave Propagation (Homework 16).....	91
Summary .....	91
The Discretization Requirements.....	91
The Boundary Conditions.....	92
Dispersion with Time .....	94
Mode Shapes and Resonance .....	94

## The Single Degree of Freedom System (The Oscillator, Homework 1)

### Summary

In this homework, we observed the response of a single degree of freedom (SDOF) oscillator with an applied friction force. We also discovered the necessary damping constant of a certain set of oscillation data to match the response of a given damped system.

### The Derivation of the SDOF Oscillator with Free Vibration

From the free body diagram (FBD), we can write:

$$F = ma = m\ddot{u}$$

From FBD,

$$f - ku = m\ddot{u}$$

And solving for f, we obtain the equation of motion (EOM) for this system:

$$m\ddot{u} + ku = f(t) \rightarrow \text{find } u(t)$$

However, this system has no external force (free vibration) and thus  $f(t) = 0$ .

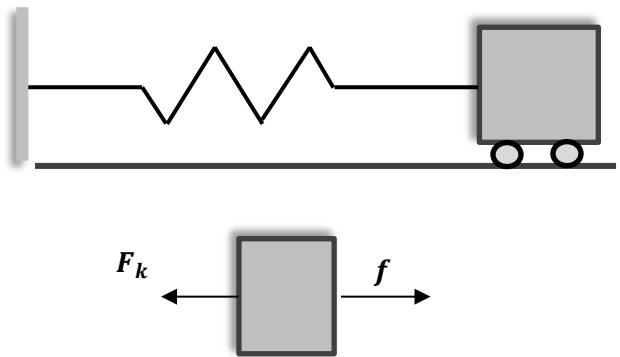


Figure 1

Moving forward, we can assume the generalized solution to the displacement function  $u(t)$  as,

$$u(t) = Ae^{\lambda t}$$

And substituting the square of the natural frequency,

$$\omega^2 = \frac{k}{m},$$

$$\ddot{u} + \omega^2 u = 0$$

Taking the derivative of  $u(t)$ ,

$$\dot{u}(t) = \lambda A e^{\lambda t}$$

$$\ddot{u}(t) = \lambda^2 A e^{\lambda t}$$

And subbing in the derivatives into the EOM,

$$\lambda^2 A e^{\lambda t} + \omega^2 A e^{\lambda t} = 0$$

$$(\lambda^2 + \omega^2) A e^{\lambda t} = 0$$

$A = 0$  would be the trivial solution. We want the non-trivial (more interesting).

$$(\lambda^2 = -\omega^2)$$

$$\lambda = \pm i\omega$$

$$u(t) = A_1 e^{i\omega t} + A_2 e^{-i\omega t}$$

From Euler's Identity,

$$u(t) = B_1 \cos \omega t + B_2 \sin \omega t$$

Applying initial conditions,

$$u(0) = u_0 \text{ initial displacement}$$

$$\dot{u}(0) = v_0 \text{ initial velocity}$$

So,

$$u(0) = B_1$$

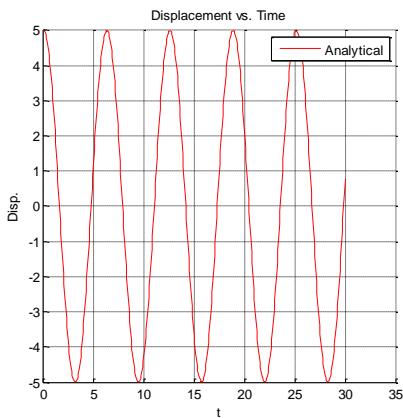
$$\dot{u}(0) = B_2 \omega = v_0$$

$$B_2 = v_0 / \omega$$

And finally,

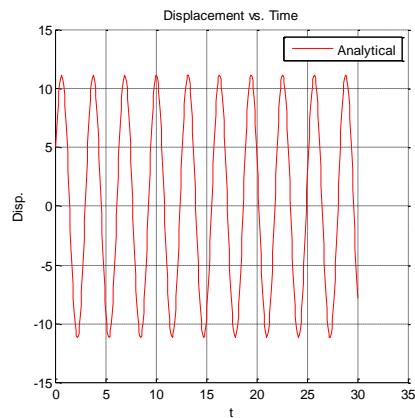
$$u(t) = u_0 \cos \omega t + \frac{v_0}{\omega} \sin \omega t$$

We can visualize the displacement of the system by plotting the displacement over a time interval using Matlab.



$$\omega = 1, u_0 = 5, v_0 = 0$$

Figure 2



$$\omega = 2, u_0 = 5, v_0 = 20$$

Figure 3

The figures above outline the motion of the system over a time interval of 30 units, both with different natural frequencies and initial velocities. As the natural frequency increases from 1 to 2, the frequency of the system increases, and the period of the system becomes shorter. When the velocity is adjusted from 0 to 20, the amplitude of the system increases dramatically (more than two times as much!). This makes perfect sense, as the velocity is in the numerator of the displacement function, and thus has an “increasing” effect on the system. The most obvious observation one can make of this system is that the system never slows down, never ceases to oscillate. This is of course not realistic of a real world system, as energy of a system can either increase or dissipate (or both).

To create a more realistic system, we can incorporate energy loss due to friction, and this cause damping. This system is called Coulomb Friction Damping, and it is entirely dependent on the friction forces that arise between the mass and the characteristics of the ground it slides on.

### Coulomb Friction Damping

Using the same procedure, we can develop the classical solution to the Coulomb Friction Damping model.

From the FBD to the right, we can develop the EOM, and the eventual displacement equation.

$$\begin{aligned} F &= ma = m\ddot{u} \\ f - ku - F_f &= 0 \quad (+) \\ f - ku + F_f &= 0 \quad (-) \end{aligned}$$

Assuming no forcing function,

$$\begin{aligned} ku + F_f &= m\ddot{u} \quad (+) \\ ku - F_f &= m\ddot{u} \quad (-) \end{aligned}$$

Taking only one direction and assuming the sign change remains consistent (it is linear, after all)...

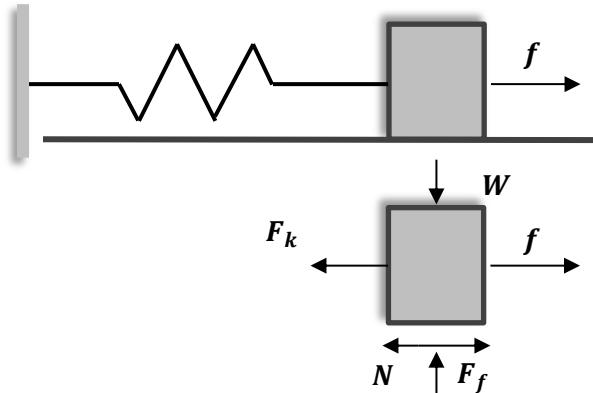


Figure 4

$$\begin{aligned} m\ddot{u} - ku - F_f &= 0 \\ \ddot{u} - \frac{ku}{m} - \frac{F_f}{m} &= 0 \\ \ddot{u} - \omega^2 u - \frac{\omega^2 F_f}{k} &= 0 \end{aligned}$$

Initial conditions,

$$u(0) = u_0$$

And no initial acceleration,

$$\begin{aligned} -\omega^2 u_0 - \frac{\omega^2 F_f}{k} &= 0 \\ u_0 &= -\frac{F_f}{k} \end{aligned}$$

Rewriting the equation of motion,

$$\ddot{u} - \omega^2 u - \frac{\omega^2 F_f}{k} = 0$$

The first two terms are just the oscillating motion (the third is constant). And so, from Euler's Identity,

$$\begin{aligned} u(t) &= B_1 \cos \omega t + B_2 \sin \omega t - \frac{F_f}{k} \quad (+) \\ u(t) &= B_1 \cos \omega t + B_2 \sin \omega t + \frac{F_f}{k} \quad (-) \end{aligned}$$

Boundary conditions,

$$u(0) = u_0 = B_1 - \frac{F_f}{k}$$

$$B_1 = u_0 + \frac{F_f}{k}$$

Substituting B1 and noting that zero initial velocity yields B2 = 0,

$$u(t) = (u_0 + \frac{F_f}{k}) \cos \omega t - \frac{F_f}{k}$$

Direction changes at time 't', when the slope of the function is zero i.e. is oscillating. Thus, the time at  $u(t)' = 0$  is the time when the box is moving to the right.

$$t_n = \frac{n\pi}{2}$$

Using this condition, the displacement function can be used for any time 't' at a whole number 'n'. Utilizing boundary conditions upon the previous conditions, a pattern emerges.

$$u(t) = \left( u_0 + (1 + 2 * n) \frac{F_f}{k} \right) \cos \omega t + (\text{sgn}) \frac{F_f}{k}$$

Where (sgn) designates the sign i.e. the direction of the moving box. 'n' is used as a cyclic parameter for the oscillating portion of the displacement function.

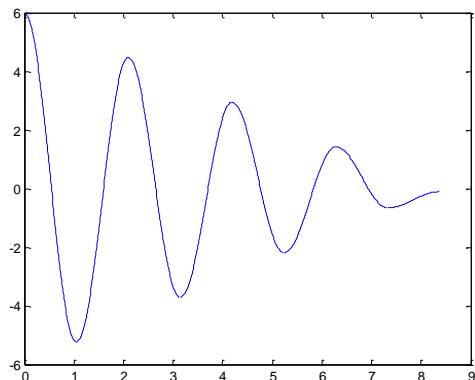


Figure 5

$$u_0 = 6, \mu = 0.35, g = 9.81, k = 50, \omega = 3$$

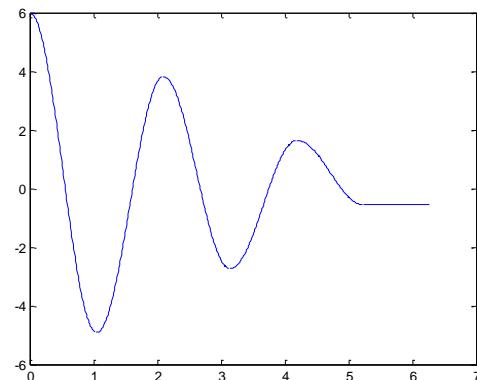


Figure 6

$$u_0 = 6, \mu = 0.50, g = 9.81, k = 50, \omega = 3$$

We have already observed what happens to our system when we change the frequency or the velocity, but what about a change in the coefficient of friction? We assume an increase will slow our model down, and thus "dampen out" faster. With an approximate 43% increase in the friction constant, the model stopped oscillating about 3 seconds sooner! This makes perfect intuitive sense, as more energy was dissipated through the sliding motion due to the higher friction coefficient.

### Determining the Damping Constant

From the model to the right, we can write the EOM as,

$$m\ddot{u} + c\dot{u} + ku = f(t)$$

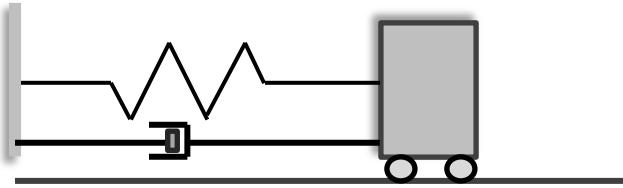


Figure 7

And dividing by 'm' (the mass) with the company of some creative algebra,

$$\begin{aligned}\ddot{u} + \frac{c\dot{u}}{m} + \frac{ku}{m} &= \frac{f(t)}{m} = 0 \\ \ddot{u} + 2\xi\omega\dot{u} + \omega^2u &= 0\end{aligned}$$

Using the decay equation once more,

$$(\lambda^2 + 2\xi\omega\lambda + \omega^2) * Ae^{\lambda t} = 0$$

The nontrivial solution suggests that what's inside the parenthesis is what has to be zero to satisfy the equation. Solving the quadratic yields:

$$\lambda_{1,2} = -\xi\omega \pm i\omega\sqrt{1-\xi^2}$$

Subbing back into  $u(t)$ ,

$$u(t) = A_1e^{-\xi\omega+i\omega\sqrt{1-\xi^2}t} + A_2e^{-\xi\omega-i\omega\sqrt{1-\xi^2}t}$$

Making  $w_D = \omega\sqrt{1-\xi^2}$

$$u(t) = e^{-\xi\omega t}[A_1e^{iw_D t} + A_2e^{-iw_D t}] = e^{-\xi\omega t}[B_1\cos w_D t + B_2\sin w_D t]$$

Applying initial conditions i.e.  $u(0) = u_0$  and  $v(0) = v_0$ , and consequently solving for the unknown constants,

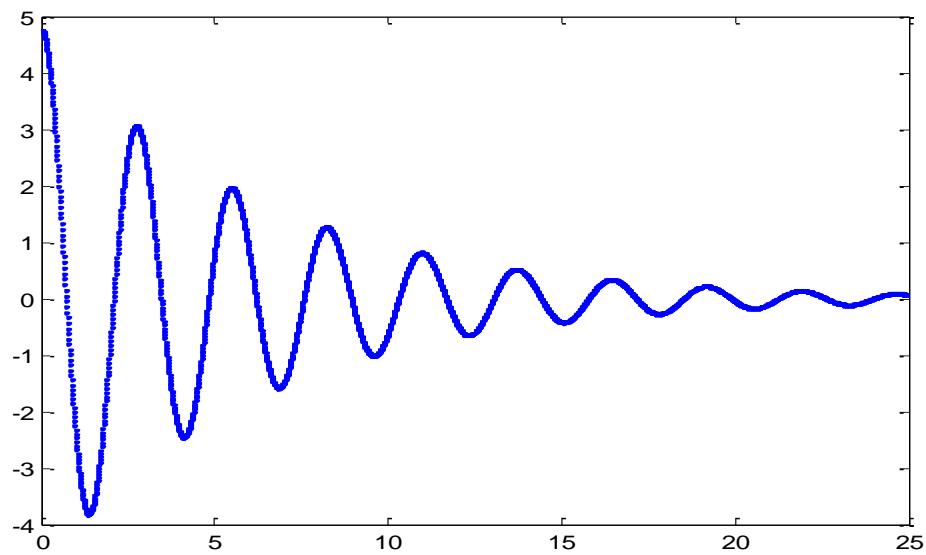
$$u(0) = u_0 = B_1$$

$$B_2 = \frac{v_0 + \xi\omega u_0}{w_D}$$

Plugging the constants back into the  $u(t)$  equation,

$$u(t) = e^{-\xi\omega t}[u_0\cos w_D t + \frac{v_0 + \xi\omega u_0}{w_D}\sin w_D t]$$

Using Matlab, the damping model can be visualized. Changing the parameters, a close approximation of the original damping constant can be made.

**Figure 8**

$$\omega = 2.3, \zeta = 0.07, v_0 = 0, u_0 = 4.76$$

From the above parameters, the damping constant is  $\omega_D \approx 2.3$  (which is also equal to the natural frequency of the system).

## The Damped SDOF (Homework 2)

### Summary

Homework 2 focuses on the damped SDOF system with an impulse loading. Therefore, this system is acted upon by an external force, and is not in free vibration. The response spectrum of the system is key in understanding the core characteristics of the damped SDOF system under impulsive loading.

### The Derivation of the Damped SDOF Forced Oscillator System (Constant and Sinusoidal)

We can still employ the EOM from the damped SDOF system, however we must account for the external force being applied.

$$\ddot{u} + 2\xi\omega\dot{u} + \omega^2u = \frac{f(t)}{m}$$

When we introduce a forcing function, we can write the displacement equation in terms of simply the sum of the homogeneous part (the oscillating part) and the particular (the applied forcing function).

$$u(t) = u_h(t) + u_p(t)$$

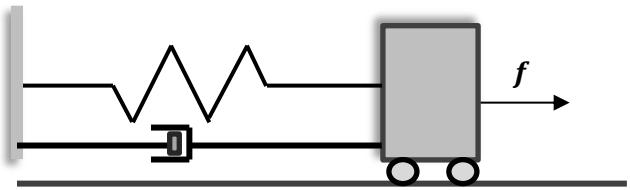


Figure 1

The homogenous displacement equation is already known, as this is simply just free vibration of the system. Starting with the more simpler solution (the constant load),

$$\ddot{u} + 2\xi\omega\dot{u} + \omega^2u = P_0 = \frac{f_0}{m}$$

Making the algebraic substitution  $u_p = \frac{P_0}{\omega^2}$

$$\ddot{u}_p + 2\xi\omega\dot{u}_p + \omega^2u_p = P_0$$

And since  $u_p = \frac{P_0}{\omega^2} = \frac{f_0}{k}$

$$u(t) = e^{-\xi\omega t}[B_1 \cos w_D t + B_2 \sin w_D t] + \frac{P_0}{\omega^2}$$

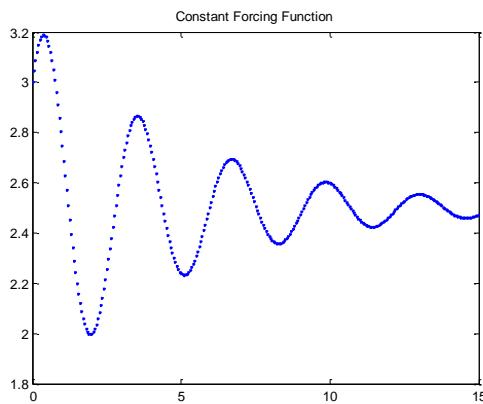
Taking derivative and applying initial conditions, we find that:

$$B_1 = u_0 - \frac{P_0}{\omega^2} = u_0 - u_s$$

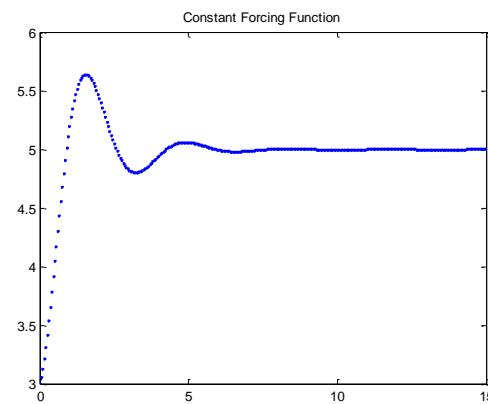
$$B_2 = \frac{v_0 + \xi\omega(u_0 - u_s)}{w_D}$$

Plugging in the constants,

$$u(t) = e^{-\xi\omega t} \left[ (u_0 - u_s) \cos w_D t + \left( \frac{v_0 + \xi\omega(u_0 - u_s)}{w_D} \right) \sin w_D t \right] + u_s$$

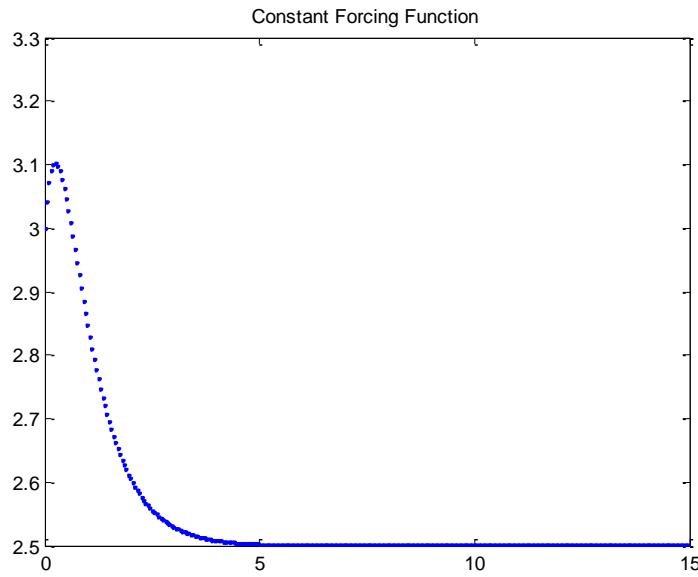


**Figure 2**  
 $\omega = 2, \xi = 0.1, u_0 = 3, v_0 = 1, P_0 = 10$



**Figure 3**  
 $\omega = 2, \xi = 0.35, u_0 = 3, v_0 = 1, P_0 = 20$

The figures above demonstrate the damping nature of the system. Note that a change in  $\xi$  (the damping ratio) directly influences the time it takes the system stop oscillating. Changing the constant force simply changes the amplitude of the system. Decay of the system's displacement can be observed by changing the damping ratio to be greater than 1. When  $\xi > 1$ , the damping ratio becomes imaginary, and since it is a function of the oscillating motion (sine and cosine), the result is exponential decay due to Euler's Identity. The graph for exponential decay is shown below.



**Figure 4**  
 $\xi = 1.1$

We can also determine the classical solution with a sinusoidal forcing function, as shown below, starting with the EOM.

$$\ddot{u} + 2\xi\omega\dot{u} + \omega^2 u = P_0 \sin(\Omega t)$$

The homogeneous equation is the same as before, but the particular equation is of the typical oscillating motion.

$$u_p(t) = A_1 \cos(\Omega t) + A_2 \sin(\Omega t)$$

And the equation of motion is the same,

$$\ddot{u}_p + 2\xi\omega\dot{u}_p + \omega^2 u_p = P_0 \sin(\Omega t)$$

So, taking the derivative of the particular equation and substituting,

$$\begin{aligned} -\Omega^2[A_1 \cos(\Omega t) + A_2 \sin(\Omega t)] + 2\xi\omega\Omega[-A_1 \sin(\Omega t) + A_2 \cos(\Omega t)] + \omega^2[A_1 \cos(\Omega t) + A_2 \sin(\Omega t)] \\ = P_0 \sin(\Omega t) \end{aligned}$$

Taking out the cosines and the sines and making sure to equate to zero,

$$[-\Omega^2 A_1 + 2\xi\omega\Omega A_2 + \omega^2 A_1] \cos(\Omega t) + [-\Omega A_2 - 2\xi\omega\Omega A_1 + \omega^2 A_2 - P_0] \sin(\Omega t) = 0$$

Pulling out the constants and placing in matrix form, dividing by  $\omega^2$  and defining  $\beta = \Omega/\omega$ ,

$$\begin{bmatrix} 1 - \beta^2 & 2\xi\beta \\ -2\xi\beta & 1 - \beta^2 \end{bmatrix} \begin{Bmatrix} A_1 \\ A_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ P_0 \end{Bmatrix}$$

Taking the inverse to solve for the constants, and plugging back into the particular equation,

$$u_p = \frac{-2\xi\beta P_0}{D} \cos(\Omega t) + \frac{(1 - \beta^2)P_0}{D} \sin(\Omega t)$$

Now the total displacement equations becomes,

$$u(t) = e^{-\xi\omega t} [B_1 \cos w_D t + B_2 \sin w_D t] + \frac{-2\xi\beta P_0}{D} \cos(\Omega t) + \frac{(1 - \beta^2)P_0}{D} \sin(\Omega t)$$

Solving for the “B” constants by taking initial conditions ...

$$\begin{aligned} B_1 &= u_0 + \frac{2\xi\beta P_0}{D} = u_0 + u_s \\ B_2 &= v_0 + \xi\omega(u_0 + u_s) - \frac{(1 - \beta^2)P_0\Omega}{D} \end{aligned}$$

Plugging back into our trustworthy displacement equation,

$$\begin{aligned} u(t) &= e^{-\xi\omega t} \left[ (u_0 + u_s) \cos w_D t + \left( v_0 + \xi\omega(u_0 + u_s) - \frac{(1 - \beta^2)P_0\Omega}{D} \right) \sin w_D t \right] + \frac{-2\xi\beta P_0}{D} \cos(\Omega t) \\ &\quad + \frac{(1 - \beta^2)P_0}{D} \sin(\Omega t) \end{aligned}$$

Note that the graphs below were created using the same parameters for the constant forcing function.

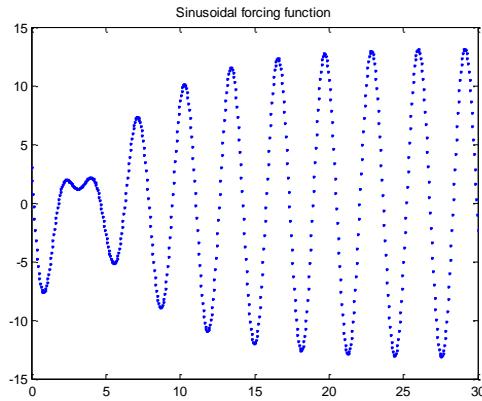


Figure 5

$$\xi = 0.1, \Omega = 2$$

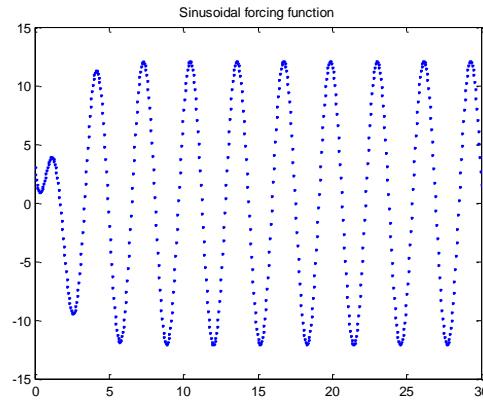


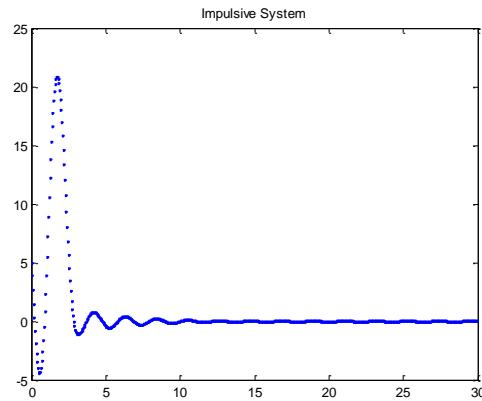
Figure 6

$$\xi = 0.35, \Omega = 2$$

Interesting result! The system appears to gain a consistent amplitude, but continues to oscillate indefinitely. With an increase in the damping ratio however, the system does not stop oscillating, but rather *continues to oscillate with a constant amplitude*. This is an interesting result in that the damping ratio no longer acts as if to cease the motion, but rather to “smooth” it out!

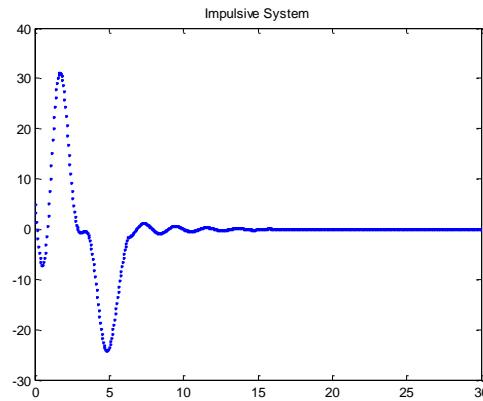
We now have the tools to model the impulsive load. In order for the first lobe to reach zero, the time at the point ( $t_0$ ) must be equal to  $\pi/\Omega$ . This would ensure that the sinusoidal forcing function is zero at the first lobe. After the initial lobe, the load is removed and the system is expected to dampen out to zero displacement. Below is a piecewise function that describes the action of the impulsive load.

$$f(t) = \begin{cases} P_0 \sin \Omega t, & t < 0 < t_0 \text{ where } t_0 = \pi/\Omega \\ 0, & t_0 < t \end{cases}$$



$$\omega = 3, \xi = 0.1, \Omega = 1, u_0 = 5, v_0 = 0, P_0 = 10$$

Figure 7

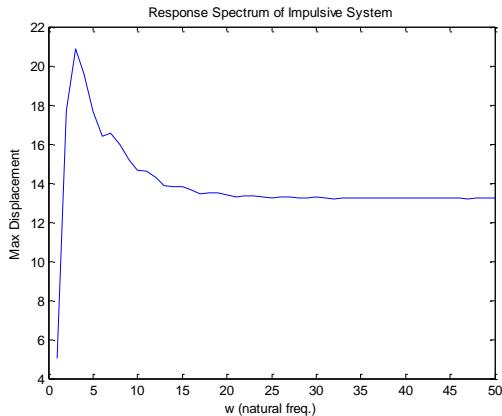


$$\omega = 3, \xi = 0.1, \Omega = 1, u_0 = 5, v_0 = 0, P_0 = 15$$

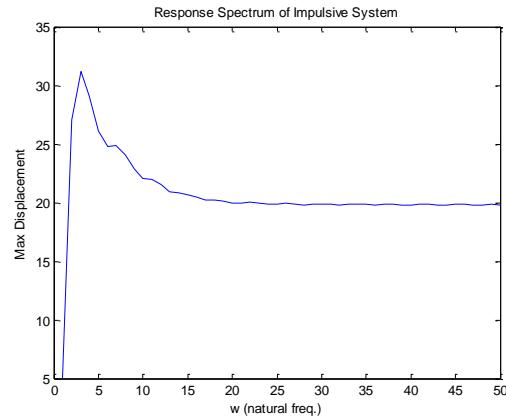
Figure 8

As expected, the model starts to dampen at the time  $t_0$ , which for this system, is exactly equal to pi ( $\Omega = 1$ ). We can change both the amplitude and duration of the impulsive part of the system, and the interesting result is that the system still dampens out at about the same rate. This result contrasts the obvious in that the system should feel the effect of the larger force and consequently take longer to reach static equilibrium.

We can also observe the behavior of this system by creating a response spectrum of the maximum displacement against the natural frequency of the system.



Same as above



Same as above

Figure 9

Figure 10

The most obvious note to make is that the largest maximum displacement happens at the lower natural frequencies, and thus displacement does not change linearly with the natural frequency of the system. Changing the amplitude and duration of the impulsive loading, we find that these changes have little effect on the response – the maximum displacement (though now larger) still occurs at lower natural frequencies. We can conclude by stating that the duration of the impulsive loading simply does not alter the response. Therefore, the area under the load-time curve is not an important organizing value for the response, as changing the duration (the area) has no effect on the response spectrum. This is an important distinction to make, in that the maximum displacement at a particular natural frequency cannot be measured or guessed upon simply by knowing the applied force over time.

## The Central Difference Method (Homework 3)

### Summary

In this homework, we observed and analyzed the central difference method (CDM) for solving the damped vibration problem by finding the exact analytical solution to the discrete equations, as well as a numerical solution. The exact analytical analysis was compared with the numerical one, as well as the classical solution.

### *The Derivation of the Exact Solution to the Difference Equation (Analytical Solution)*

To start, we first need to develop the characteristic equation for the damped vibration system. We begin with variable substitutions for the EOM from the central difference approximations for the first and second derivatives. The derivative approximations are as shown below:

$$v_n = \frac{u_{n+1} - u_{n-1}}{2h} \text{ and } a_n = \frac{u_{n+1} - 2u_n + u_{n-1}}{h^2}$$

We can rewrite the EOM for this system as,

$$\begin{aligned} \ddot{u} + 2\xi\omega\dot{u} + \omega^2u &= 0 \\ a_n + 2\xi\omega v_n + \omega^2u_n &= 0 \end{aligned}$$

Making the appropriate substitutions from the derivative approximations and some algebra, we have the following:

$$u_{n+1} + u_n(h^2\omega^2 - 2) + u_{n-1} + h\omega\xi(u_{n+1} - u_{n-1}) = 0$$

Assuming the solution of the difference equation is in the exponential form,  $u_n = Az^n$ , we can substitute the solution into the above equation to obtain the characteristic equation.

$$Az^{n-1}[z^2 + (h^2\omega^2 - 2)z + 1 + h\omega\xi(z^2 - 1)] = 0$$

The characteristic equation is thus,

$$z^2 + (h^2\omega^2 - 2)z + 1 + h\omega\xi(z^2 - 1) = 0$$

Note that we must now solve for the values of  $z$ , as these are the only values consistent with the difference equation. Solving the quadratic equation and using the imaginary identity, we now have the two roots of the characteristic equation.

$$z = a + ib$$

$$\text{Where } a = \frac{2-h^2\omega^2}{2(h\omega\xi+1)} \text{ and } b = \frac{h\omega\sqrt{1-\frac{1}{4}h^2\omega^2-\xi^2}}{1+h\omega\xi}$$

We can also determine the angle  $\varphi$  (which is essentially the frequency of the system) by taking the inverse tangent of  $a$  and  $b$ . Since the characteristic equation had two roots, the general solution must be the sum of two terms.

$$u_n = A(z_1)^n + B(z_2)^n = A(a + ib)^n + B(a + ib)^n$$

Noting the initial conditions of the system and employing trig identities, the general solution becomes:

$$u_n = r^n[(A + B)\cos n\varphi + i(A - B)\sin n\varphi]$$

From initial conditions, we know that,

$$u_0 = A + B$$

Using algebraic manipulation, the discrete equations and the EOM, we can develop an expression for  $i(A - B)$ , which is shown below.

$$i(A - B) = \left( \frac{2hv_0 + u_0B}{1 + D} - u_0a \right) \left( \frac{1}{b} \right)$$

Where  $B = \frac{2-h^2\omega^2}{1-h\omega\xi}$  and  $D = \frac{1+h\omega\xi}{1-h\omega\xi}$

And thus, the exact solution to the discrete equations is thus,

$$u_n = r^n \left[ u_0 \cos n\varphi + \left( \frac{2hv_0 + u_0B}{1 + D} - u_0a \right) \left( \frac{1}{b} \right) \sin n\varphi \right]$$

Where  $n = t/h$

### *The Time Stepping Algorithm (Numerical Solution)*

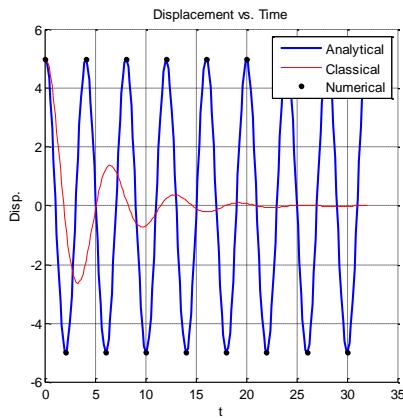
We know that the first step in the algorithm is the prescribed initial displacement. However, we also need the displacement before that (so to speak) so as to calculate the new position. Essentially, three displacements are needed for the CDM numerical solution at every time step: the one before, the current one, and the one after that. The position before the initial position and the new position equations are shown below.

$$x_{oldold} = \frac{2hv_0 - u_0 \frac{B}{D}}{-1(1 + \frac{1}{D})} \text{ and } x_{new} = \frac{x_{oldold}(h\omega\xi - 1) + x_{old}(2 - h^2\omega^2)}{(1 + h\omega\xi)}$$

Note that the new position was simply determined by using the EOM combined with the difference equation for acceleration. The new position is then placed into the position before that, and the old position is placed into the “oldold” position.

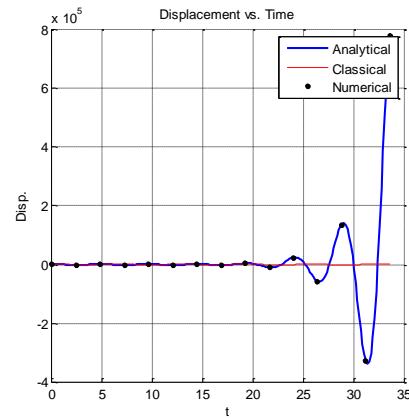
### *Comparison Between the Numerical, Classical and Analytical Solution*

We can compare the three different solutions to the damped SDOF oscillator using Matlab. Note that the stability limit for the analytical solution to the damped oscillator remains  $h < \frac{2}{\omega}$ , as anything greater would equal a negative value in the square root term. The result of instability of the system is shown below.



$$\omega = 1, \xi = 0.2, u_0 = 5, v_0 = 0, h = 2$$

Figure 1

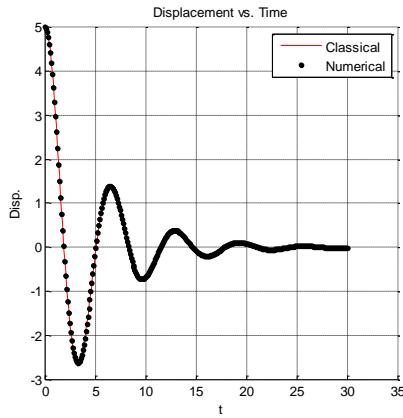


$$\omega = 1, \xi = 0.2, u_0 = 5, v_0 = 0, h = 2.4$$

Figure 2

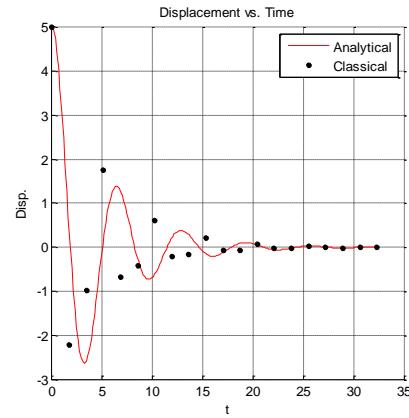
In Figure 1, we see that the analytical and numerical solution oscillate freely without damping, since the square root term was imaginary to a very small degree. If we increase the time step very well past the limit, we note exponential growth, as shown in Figure 2. The red line is the classical solution, which should of course see no effect on stability.

As shown above, the time step is an important parameter in the exactness of the solution. A time step too large can lead to inaccuracy in the motion of the system, and too little can be considered “overkill”. The following graphs of the displacement of the model demonstrate the importance of the time step.



$$\omega = 1, \xi = 0.2, u_0 = 5, v_0 = 0, h = 0.05$$

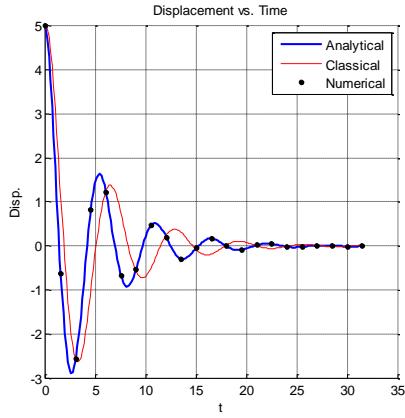
Figure 3



$$\omega = 1, \xi = 0.2, u_0 = 5, v_0 = 0, h = 1.7$$

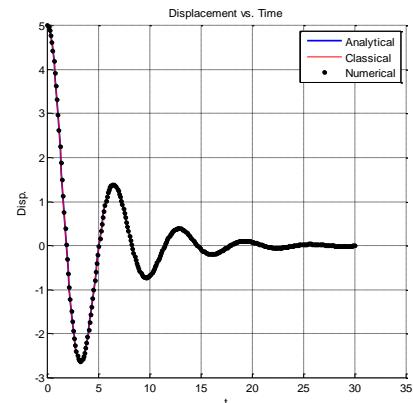
Figure 4

In Figure 4, the numerical solution (black dots) does not fully match up with the classical solution, and appears scattered. Since the numerical solution is merely an approximation of the classical solution, the more points that are computed results in a better approximation (see Figure 3). The numerical and analytical solutions are both derived from the same discrete equations, and thus should produce the same result. Therefore, it just so happens that the analytical solution fails to fall directly in line with the classical solution when a larger time step is computed.



$$\omega = 1, \xi = 0.2, u_0 = 5, v_0 = 0, h = 1.5$$

Figure 5



$$\omega = 1, \xi = 0.2, u_0 = 5, v_0 = 0, h = 0.05$$

Figure 6

Note that the blue line and black dots do not overlap the red line. Increasing the amount of time step computations (decreasing the time step) results in a more accurate solution and better approximation of the classical solution (see Figure 6). Note that the line in Figure 6 should be a purple color, but has reverted to red due to the resizing of the picture.

## Newmark's Method (Homework 4)

### Summary

The undamped vibration problem in this homework was solved using Newmark's Method to approximate the solution and also to provide an exact analytical solution. The performance of the integration parameters were explored, including the particular integration parameters that provide numerical damping and growth. The damped vibration problem was also solved numerically using Newmark's Method.

### *The Derivation of the Exact Solution to the Numerical Solution (Analytical Solution)*

In a similar fashion to the exact solution of the central difference equations, we can produce an exact solution to the undamped vibration problem using the Newmark equations. The Newmark discrete equations are shown below.

$$\begin{aligned} a_{n+1} + \omega^2 u_{n+1} &= 0 \\ v_{n+1} - v_n - h[\gamma a_n + (1 - \gamma)a_{n+1}] &= 0 \\ u_{n+1} - u_n - hv_n - h^2 \left[ \beta a_n + \left( \frac{1}{2} - \beta \right) a_{n+1} \right] &= 0 \end{aligned}$$

First, we can substitute the acceleration equations from the EOM (undamped vibration) at the current and next time step ( $n$  and  $n + 1$ ) into the third equation. The result, to wit

$$D_1 u_{n+1} + D_2 u_n + hv_n = 0$$

Where  $D_1 = \omega^2 h^2 \left( \beta - \frac{1}{2} \right) - 1$  and  $D_2 = 1 - \omega^2 h^2 \beta$

If we are to again assume that the solution to the difference equation is in the exponential form, we must re-write the derived equation above in terms of solely the position at each step. Taking the difference at two adjacent time steps, we have the preliminary difference equation

$$u_n [D_2 - D_1] + D_1 u_{n+1} + h[v_n - v_{n-1}] - D_2 u_{n-1} = 0$$

Calling the third term "C" and substituting the velocity term at the current step for the Newmark discrete equation,

$$C = h^2 [\gamma a_{n-1} + (1 - \gamma)a_n]$$

Substituting again the acceleration equations from the EOM, we have

$$C = -\omega^2 h^2 [\gamma u_{n-1} + (1 - \gamma)u_n]$$

Subbing in "C" into the pre-difference equation and simplifying,

$$\begin{aligned} u_n [D_2 - D_1] + D_1 u_{n+1} + C - D_2 u_{n-1} &= 0 \\ D_1 u_{n+1} + [D_2 - D_1 - \omega^2 h^2 (1 - \gamma)] u_n - [D_2 + \omega^2 h^2 \gamma] u_{n-1} &= 0 \end{aligned}$$

With the difference equation known, we can fit the solution to be of the exponential form, as shown below.

$$D_1 z^2 + [D_2 - D_1 - \omega^2 h^2(1 - \gamma)]z - [D_2 + \omega^2 h^2 \gamma] = 0$$

Solving the difference equation (and performing a lot of algebra), we have the standard form (characteristic equation) of the solution.

$$z = a \pm ib$$

$$\text{Where } a = \frac{\omega^2 h^2 (2\beta - \gamma + \frac{1}{2}) - 2}{2D_1} \text{ and } b = \frac{\omega h \sqrt{\frac{3}{4}\gamma\omega^2 h^2 - \frac{1}{16}\omega^2 h^2 - \frac{1}{4}\omega^2 h^2 \gamma^2 - \omega^2 h^2 \beta + 1}}{D_1}$$

Since the of the characteristic equation is complex (imaginary root) we can put the equation in the form

$$z^n = r^n (\cos n\varphi + i \sin n\varphi) = (a + ib)^n$$

$$\text{Where } r = \sqrt{a^2 + b^2} \text{ and } \varphi = \cos^{-1}(\frac{a}{r})$$

Subbing in the acceleration equations from the EOM at the current and next time step, and subbing in the discrete equation for velocity at the next time step, we have the displacement equation for the next time step.

$$u_1 = \frac{u_0 + hv_0 - \omega^2 h^2 \beta u_0}{1 + \omega^2 h^2 (\frac{1}{2} - \beta)}$$

Form our solution to the difference equation, we can write the displacement initial displacement and displacement at the next time as

$$u_0 = A + B$$

$$u_1 = u_0 + i(A - B)b$$

Since  $u_1$  is known, we can write the imaginary part as

$$i(A - B) = \frac{u_1 - u_0 a}{b}$$

Therefore, the analytical solution to the undamped vibration problem with Newmark's discrete equations is thus

$$u_n = r^n [u_0 \cos n\varphi + \frac{u_1 - u_0 a}{b} \sin n\varphi]$$

Where  $n = t/h$

### *The Time Stepping Algorithm (Numerical Solution)*

To solve the problem numerically, we can use both the equation of motion and the Newmark Discrete equations to approximate the acceleration, velocity and displacement at the successive time steps. The initial velocity and position are pre-defined, and the intitial acceleration is computed from solving the EOM.

$$x_{old} = u_0$$

$$v_{old} = v_0$$

$$a_{old} = -w^2 x_{old}$$

In order to compute the next velocity and position, we need the new acceleration. Using the EOM and the discrete equations,

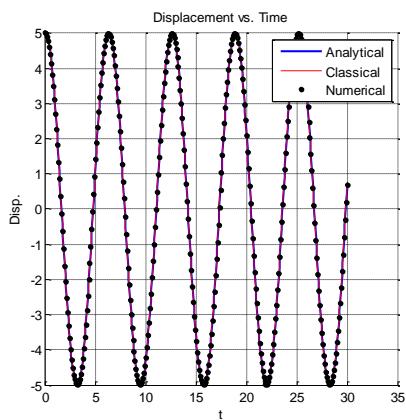
$$a_{new} = -w^2 \frac{b_n}{D_1}$$

Where  $b_n = -(x_{old} + h^2 \beta a_{old} + h v_{old})$

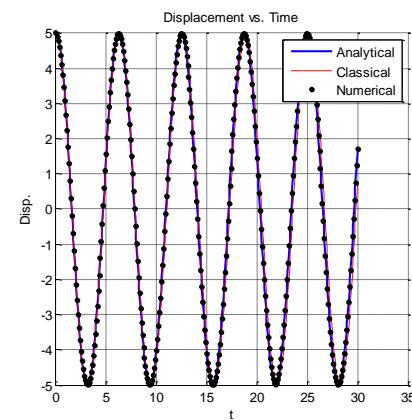
The “new” velocity and position are then calculated from the discrete equations.

### *Comparison Between the Numerical, Classical and Analytical Solution*

With the numerical and analytical solution presented above, we can compare the two with the classical solution, as well as explore a range of values for the integration parameters. Below is a three-way comparison of the numerical, classical and analytical solution to the un-damped vibration problem. Each graph corresponds to a change in the integration parameter  $\beta$ , with the rest of the parameters remaining constant.



$\beta = 0.3, \gamma = 0.5$



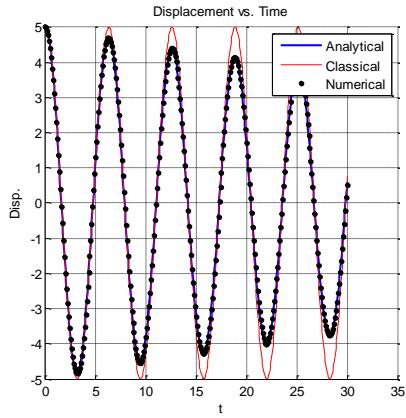
$\beta = 1.7, \gamma = 0.5$

Figure 1

Figure 2

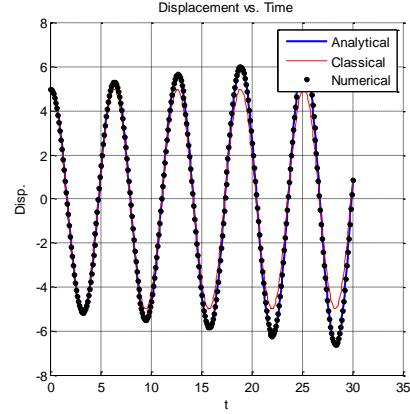
Note the inaccuracy of both the numerical and analytical solution with an increase in the  $\beta$  value! The solution is not quite following the path of the classical solution. This leads to an important discovery in that we really do not know the accuracy of a solution if we cannot solve it classically to check the result. We must infer with our own engineering judgement if our approximation is “good enough” and what isn’t! From our experience with vibration problems, we can conclude that smaller values of  $\beta$  are good, where higher values are not. We can use this same judgment of  $\beta$  for other vibration problems that we will solve using Newmark’s Method.

The other integration parameter,  $\gamma$ , acts a bit differently. It actually has the effect of forcing the system to either grow exponentially, or decay. It is this result that makes the values of  $\gamma$  so interesting for approximating these types of problems, as demonstrated below.



$$\beta = 0.3, \gamma = 0.3$$

Figure 3



$$\beta = 0.3, \gamma = 0.7$$

Figure 4

With  $\gamma = 0.3$ , the numerical solution approximates the solution as exhibiting damping characteristics, whereas  $\gamma = 0.7$  exhibits growth. The sweet spot is  $\gamma = 0.5$ , as shown in Figure 1. At this value, the approximation converges to the actual, classical solution of the un-damped vibration problem. We can make the conclusion that the  $\beta$  value must be close in magnitude to its counterpart,  $\gamma$ , in order to generate a well representation of the vibration problem. Newmark's Method is a terrific approximation of the classical solution to this particular vibration problem, granted the integration parameter values are set appropriately. Newmark's Method will be used hence forth in the analysis of "n"-degrees of freedom (NDOF) systems where the classical solution will be too difficult to determine.

## ***Newton's Method with Nonlinear Restoring Forces (Homework 5)***

### ***Summary***

Homework 5 introduces the principles of solving nonlinear equations through Newton's Method, and the continuation of approximating differential equations using the discrete equations of Newmark's Method in conjunction with the equations of motion of the system. With these two methods combined, the response of a SDOF system with a nonlinear restoring force was approximated. The nonlinear response was compared with the linear elastic response of the previous SDOF systems.

### ***Newton's Method***

Approximating nonlinear equations can be done easily and effectively using Newton's Method. The EOM in this case is nonlinear, we are now accounting for a nonlinear restoring force in the spring. In previous systems, the spring was treated to be completely elastic. In this case, the response is purely nonlinear since the restoring force in the spring changes nonlinearly with the displacement. In essence, Newton's Method takes a nonlinear equation and transforms it into a linear one through the use of the Taylor Series Expansion. The first two terms of the Taylor Series are used as the linear substitute of the real nonlinear equation. The sum of the first two terms is set equal to zero and then solved. It is an iterative process and requires an initial guess to get started, and for problems like this one, can converge rapidly (depending of course on your level of tolerance and error). The first two terms of the Taylor Series can be rearranged, as shown below.

$$x = x_0 - \frac{g(x_0)}{g'(x_0)}$$

Where  $x$  is the new approximation,

$x_0$  is the old approximation (or initial guess at time  $t = 0$ )

$g(x_0)$  is the residual function evaluated at  $x_0$  and

$g'(x_0)$  is the gradient function evaluated at  $x_0$ .

The new approximation  $x$  will be calculated within a “while loop” until either maximum iterations have occurred or the norm of the residual (error) is close to the set tolerance .

### ***Implementing Newton's Method***

The restoring force for a stiffening and softening model can look something like the equation shown below.

$$r(u) = ku(1 + bu^2) \leftarrow \text{Stiffening Model}$$

$$r(u) = \frac{ku}{\sqrt{1 + bu^2}} \leftarrow \text{Softening Model}$$

The restoring force in the spring either increases or decreases nonlinearly with a change in the displacement. With these equations, we can complete the residual and gradient functions for each model.

Since we cannot approximate the new acceleration at the current time step using Newmark alone, we will use Newton's Method *specifically to calculate the new acceleration*. Since the new acceleration is what is needed out of the nonlinear EOM, we must take the derivative of the residual with respect to the acceleration ( $\frac{dg}{da}$ ). The result is the gradient function. The residual and gradient of either model is shown below for convenience.

$$g = ma + r(u) - F$$

$$A = \frac{dg}{da} = m + \frac{dr}{du} \frac{du}{da}$$

Where  $F$  is the sinusoidal forcing function.

The derivative of the restoring force with respect to the displacement is simply,

$$\frac{dr}{du} = k(1 + 3bu^2) \leftarrow \text{Stiffening Model}$$

$$\frac{dr}{du} = \left( kbu^2 \left( -\left( (1 + bu^2)^{-\frac{3}{2}} \right) \right) + k(1 + bu^2)^{-\frac{1}{2}} \right) \leftarrow \text{Softening Model}$$

And the derivative of the displacement with respect to acceleration is simply the constant  $\eta$  (from the Newmark discrete equation for position). Thus, the gradient is

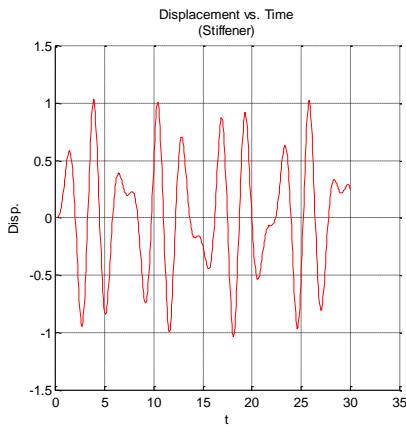
$$A = \frac{dg}{da} = m + k(1 + 3bu^2)\eta \leftarrow \text{Stiffening Model}$$

$$A = \frac{dg}{da} = m + \left( kbu^2 \left( -\left( (1 + bu^2)^{-\frac{3}{2}} \right) \right) + k(1 + bu^2)^{-\frac{1}{2}} \right) \eta \leftarrow \text{Softening Model}$$

Implementing these functions into the Newton “while loop” will generate the best approximation of the acceleration of the mass at each time step.

### Comparing Nonlinearity with the Linear Elastic Response

We know that the linear elastic response of the vibration problem is simply a linear change in the internal force of the spring with the associated displacement. Analyzing the stiffener model, we observe that as the spring displaces, the internal force increases *nonlinearly*. Observe the plots below of the displacement-time and force-displacement curves.



$$u_0 = 0, v_0 = 0, \omega = 2$$

Figure 1

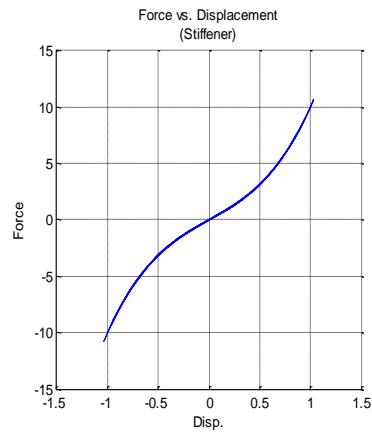
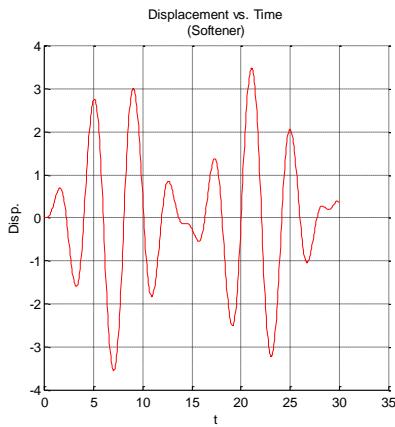


Figure 2

We can see that the nonlinear relationship of the internal force and displacement has a great effect on the response of the system. An obvious note to make is that the internal force increases nonlinearly (almost exponentially) with an increase in the displacement. This response manifests itself in the displacement-time curve. The oscillations begin normally, as any linear undamped vibration system would, but then the nonlinearity of the system shows itself – the displacement decreases rapidly due to the almost exponential increase in internal force. We can categorize this phenomenon as “stiffening”. The repeating oscillations are simply due to the spring going between positive and negative displacements. If the spring were to stretch in one direction infinitely, we would expect the displacement-time curve to essentially dampen-out. Therefore, this model is really the one we want in buildings or bridges, but does not really exist. Materials do not behave in this way. Rather, materials lose stiffness from excessive stretching or compression, and consequently lose the ability to retain internal forces that it otherwise could initially. The more appropriate and accurate way to model a system is to implore a ‘softening’ effect. This phenomenon happens when we motivate the internal force to decrease with an increase in the displacement, or to “soften” the system. The plots are of the “softening” system are shown below.



$$u_0 = 0, v_0 = 0, \Omega = 2$$

Figure 3

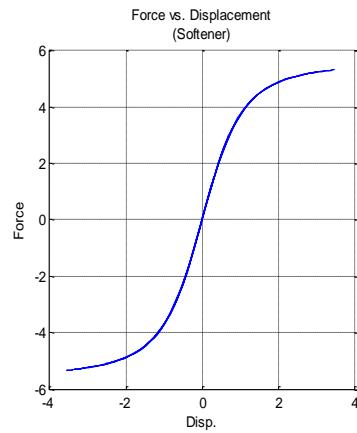
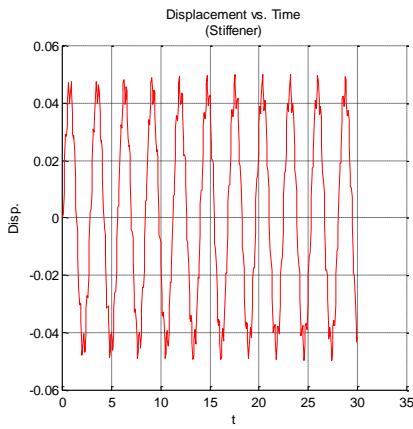


Figure 4

We observe exactly what we were expecting! The displacement-time curve demonstrates that the spring displacement increases. We must note again for clarity that the spring displacement appears to decrease, but that is due to the spring going between positive and negative displacements. For each event, we can note that the spring displacement increases due to the decrease in the internal force. From Figure 4, we can note that the force does indeed “level-off” after some displacement. This effect paints more of the picture of how steel or most other materials actually behave under displacement (or strain) – the internal force increases somewhat linearly with an increase in strain, but after some amount of strain, begins to “level-off”. Essentially, the same load applied continues to stretch the member, and if applied for too long, the member fails. We will observe this phenomenon of when a member fails in the next homework – elasto-plasticity.

Note: It is worth mentioning that the applied force is very much a function of the nonlinear response for this system. An increase in the driving frequency ( $\Omega$ ) of the system will generate a more elastic response due to the now extremely oscillatory behavior of the system. In essence, the applied force is showing more of a presence in the calculations than the effect of the nonlinear internal force on the system due to the magnitude of the applied force. We can adjust the amplitude of the forcing function to also manipulate the nonlinear response. See Figures 5 and 6 below for reference.



$$u_0 = 0, v_0 = 0, \Omega = 20$$

Figure 5

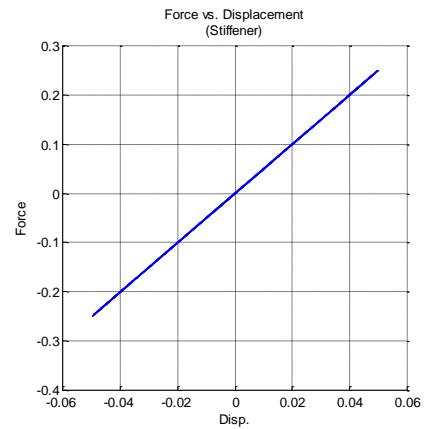
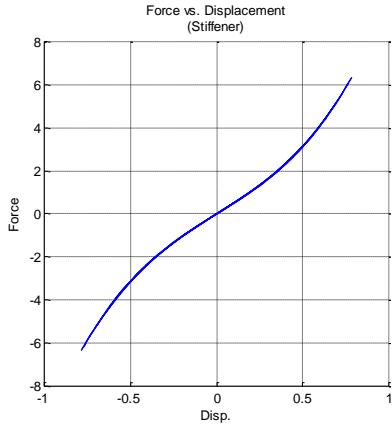


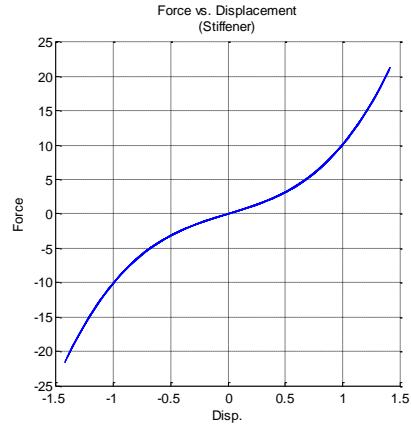
Figure 6

Increasing the driving frequency creates an almost perfectly elastic response. We can observe this effect in both plots. We can also change the amplitude constants of the sinusoidal forcing function to observe the resulting ramifications on the nonlinear system.



$$u_0 = 0, v_0 = 0, \Omega = 2, F_a = 1$$

Figure 7



$$u_0 = 0, v_0 = 0, \Omega = 2, F_a = 5$$

Figure 8

Changing the amplitude of the sinusoidal forcing function has a direct impact on the magnitude of the internal force in the spring. The force in the spring increases with an increase in the amplitude of the forcing function.

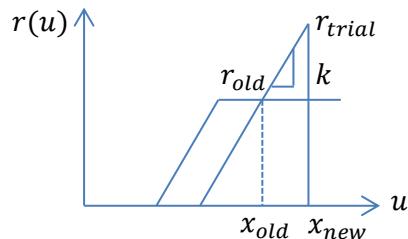
## Elasto-Plastic SDOF (Homework 6)

### Summary

This homework explores the elasto-plastic response of the un-damped vibration problem. The nonlinearity of the system is compared with the linear elastic response of previous systems. A sinusoidal applied force is used to generate motion of the system. This elasto-plastic system serves as a gateway to understanding and implementing NDOF systems with elasto-plastic members.

### The Elasto-Plastic Response

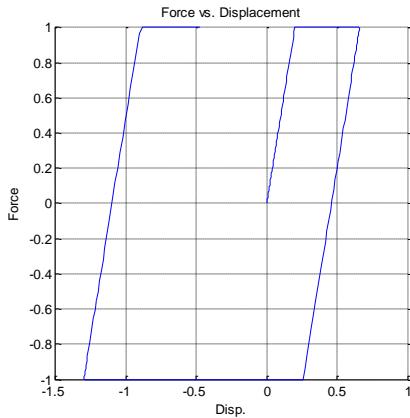
Using the same EOM from Homework 5, we can construct the much needed residual and gradient functions for use in the Newton's loop to generate the acceleration at the next time step. In order to generate the elasto-plastic response in the system, we must first establish the maximum allowable internal force within the spring. We call this the "yielding" force – anything greater than this force and the material (spring, in this case) begins to yield or deform. In this scenario, we want to observe the plastic deformation in the spring. The plastic deformation is essentially the strain in the spring during which it yields. In order to generate the graph needed to observe this response, we have to implement some checks on the internal force. Since the displacement can be either positive or negative, so can the internal force, and thus we must check the absolute value of the internal force against the prescribed yielding force ( $\sigma_m$ ).



From the elasto-plastic plot shown above, we can calculate  $r_{trial}$  by summing the internal force at the current step  $r_{old}$  and the internal force that is comprised of  $x_{old}$  and  $x_{new}$ . This idea is expressed in the following equation.

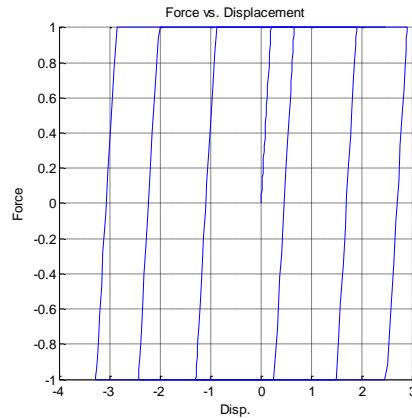
$$r_{trial} = r_{old} + k(x_{new} - x_{old})$$

With the anticipated internal force calculated, it can now be checked to verify if it exceeds or deceeds the yielding value. Using the absolute value of the trial internal force and the sign of the internal force, we can determine the appropriate internal force at the next time step. The plots of the elasto-plastic response can be found below for two different time ranges.



$$\sigma_0 = 1, u_0 = 0, v_0 = 0, \Omega = 2, t_f = 3.5$$

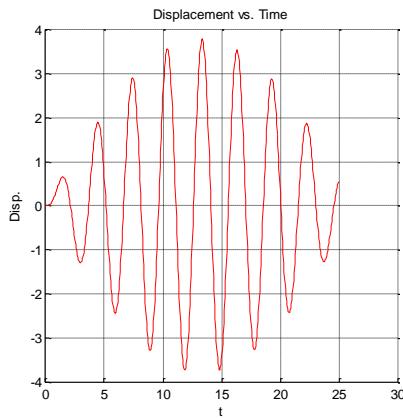
Figure 1



$$\sigma_0 = 1, u_0 = 0, v_0 = 0, \Omega = 2, t_f = 10$$

Figure 2

From Figure 1, we see the elasto-plastic response of the system. The internal force increases linearly with the displacement, then levels out at the yielding force. The internal force then decreases as the spring begins to compress, and the internal force again levels out. We can observe that the plastic deformation (or plastic strain) of this system is approximately 0.45 (from Figure 1). Increasing the time frame, we see that the response continues, though it is now difficult to detect the last data point. Increasing the time frame, we note an interesting result in the displacement-time graph.



$$\sigma_0 = 1, u_0 = 0, v_0 = 0, \Omega = 2, t_f = 25, h = 0.01 \quad \sigma_0 = 1, u_0 = 0, v_0 = 0, \Omega = 2, t_f = 25, h = 0.01$$

Figure 3

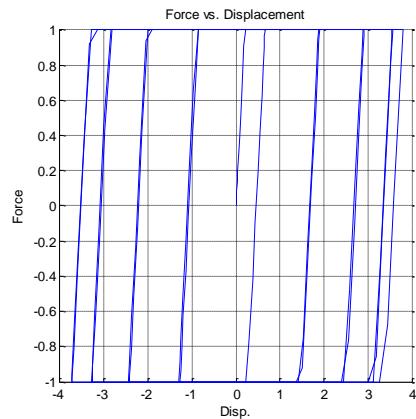
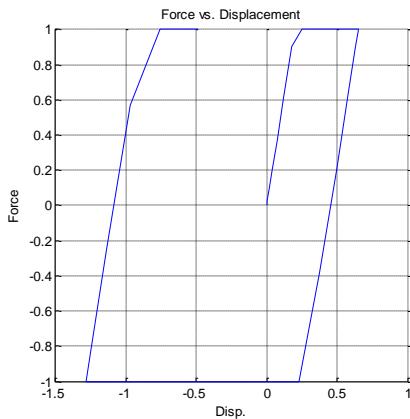


Figure 4

The displacement of the spring increases *and then* decreases during the allotted time frame. This is due to the sinusoidal forcing function of the system. We can note the differences in the result of the response if we change the time step.



$$\sigma_0 = 1, u_0 = 0, v_0 = 0, \Omega = 2, t_f = 25, h = 0.1 \quad \sigma_0 = 1, u_0 = 0, v_0 = 0, \Omega = 2, t_f = 25, h = 1.0$$

Figure 5

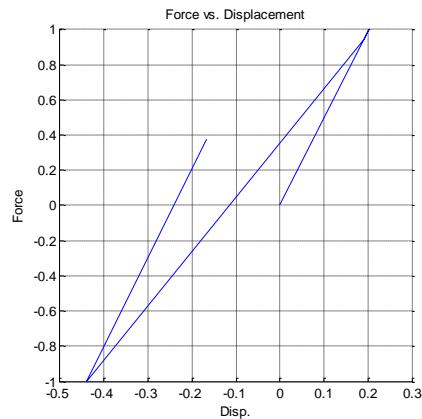


Figure 6

What a dramatic difference! With a step of 0.1, it is somewhat apparent that the result is a bit skewed – the approximation of the internal force is slightly off since  $r_{trial}$  is a function of the new and old displacement. The evidence of inaccurate approximation is discerning in Figure 6 – it does not even resemble elasto-plastic! This study again reinforces the need for small time steps in an approximating method like Newmark.

## The NDOF System (Homework 7)

### Summary

In this assignment, the task was to create a multi-degree of freedom (ndof) system with an undamped, dynamic response. The slight change in the stiffness matrix 'K' was observed so the boundary conditions between the 'shear building' and 'bridge' could be implemented to produce a dynamic response for both types of structures. Newmark's Method was utilized to approximate the acceleration, velocity and displacement at every time step.

### The Equations of Motion

The governing equations that determine the response of a dynamic system are the equations of motion (EOM). The EOM is Newton's Second Law, and is shown below for the ndof system.

$$M\ddot{u} + Ku = f$$

Where  $M$  is the mass matrix (masses on the diagonal),

$\ddot{u}$  is the double derivative of the displacement (acceleration vector of the masses),

$u$  is the displacement vector of the masses, and

$f$  is the applied force vector for the masses.

We can derive the stiffness matrix for both the 'building' system and the 'bridge' system. It is in this matrix quantity that the difference in the boundary conditions between the two systems shows up.

### The Building and Bridge EOM

As mentioned above, the difference between the two systems lies solely in the stiffness matrix. The mass matrix is simply a diagonal matrix containing the masses (degrees of freedom or 'bodies') along the diagonal. We consider this to be a "clumped" mass state, because we are neglecting the deformation of the bodies. Though this is not acceptable if the deformation of masses should be taken into consideration, it is only so that we can study and document the dynamic response is when we wish to simply make the collection of masses constant. Consider a 3 'story' building (or three degrees of freedom) system, with the geometry of the building on the left (Figure 1) and the FBD's of each mass to the right (Figure 2)

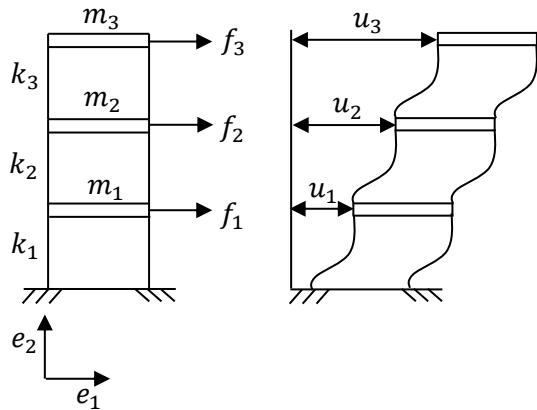


Figure 1

$$\begin{aligned}
 f_{3e_1} &\rightarrow k_3(u_3 - u_2)(-e_1) \\
 f_{2e_1} &\rightarrow k_3(u_3 - u_2)(+e_1) \\
 f_{2e_1} &\rightarrow k_2(u_2 - u_1)(-e_1) \\
 f_{1e_1} &\rightarrow k_2(u_2 - u_1)(+e_1) \\
 f_{1e_1} &\rightarrow k_1(u_1 - 0)(-e_1)
 \end{aligned}$$

Figure 2

By noting that the sum of the forces in Figure 2 must be equal to the product of the mass of the respected element and the acceleration, and rearranging the equations, we have,

$$\begin{aligned}
 m_1\ddot{u}_1 + (k_1 + k_2)u_1 - k_2u_2 &= f_1 \\
 m_2\ddot{u}_2 - k_2u_1 + (k_2 + k_3)u_2 - k_3u_3 &= f_2 \\
 m_3\ddot{u}_3 - k_3u_2 + k_3u_3 &= f_3
 \end{aligned}$$

Putting the three equations into matrix form,

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{Bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_3 \end{Bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix}$$

And thus the EOM becomes,

$$M\ddot{u} + Ku = f$$

We can generalize the stiffness matrix for ‘n’ degrees of freedom, which is shown in the associated Matlab code. For the diagonals, it is simply the sum of the successive story’s stiffness, with the last entry being just the stiffness of the last story (since there is not another after it). The nature of the stiffness matrix is that it is symmetric, so the negative off-diagonal entries are easy to place.

The same procedure is applied for the EOM of the bridge. With a slight difference in the bridge free-body diagrams (the upper mass is fixed, so an additional element is needed), the stiffness matrix and EOM for the bridge is thus,

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{Bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_3 \end{Bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 + k_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix}$$

Using the EOM with the varying stiffness matrix for the building and bridge system, we can observe the dynamic response of either system with an applied load  $f$ .

### Approximating the Time-Dependent Variables

The use of Newmark's Method will again be used here for the ndof system to approximate the acceleration, velocity and position. To start, the old (or initial) acceleration can be calculated by rearranging the EOM with the initial conditions of the system.

$$a_{old} = M^{-1}(f - Kx_{old})$$

Once the initial acceleration is computed, the Newmark time stepping algorithm begins. The sinusoidal forcing function is updated once every time step, along with the new acceleration (based off the Newmark equations and the EOM, and then the new velocity and position are calculated. The process repeats for the prescribed amount of time steps. The equations listed below are in order of execution.

$$a_{new} = (M + \eta K) \backslash (F - Kb_n)$$

Where  $\eta$  is the function  $h^2(\frac{1}{2} - \beta)$  from the Newmark equation for the new displacement ( $x_{new}$ ),

$b_n$  is the function  $x_{old} + hv_{old} + h^2\beta a_{old}$ , which is just the other half of the Newmark equation mentioned above, and

$F$  is the sinusoidal forcing function,  $F = F_{o1} + F_{o2}\sin(F_{wo}t)$

As can be seen here, the new acceleration is simply approximated based off the displacement, velocity and acceleration of the previous time step. The Newmark equation is neatly and subtly substituted into the EOM. The Newmark equations are listed below (again) for reference.

$$\begin{aligned} v_{n+1} &= v_n + h[\gamma a_n + (1 - \gamma)a_{n+1}] \\ u_{n+1} &= u_n + hv_n + h^2 \left[ \beta a_n + \left( \frac{1}{2} - \beta \right) a_{n+1} \right] \end{aligned}$$

### The Dynamic Response

With the time-stepping algorithm discussed, it is now time to explore the response of the structure. However, before the analysis begins, we must make note on a key addition to the program. The masses, stiffness of the elements and the applied force were all modeled to represent a somewhat authentic response of a building being attacked by wind forces, and the dynamic response of such an event. In detail, the model places lower masses towards the base of the tower, and smaller at the top. The stiffness is also made inversely proportional to the height of the building. The “wind” force, on the other hand, increases with height of the building, as this is natural due to more geometrical space and opening at higher elevations. Below is a plot of the wind force model (trajectory and displacement vs. time), alongside a plot of another building without the features of the model under the same conditions. Note that the blue lines is the trajectory of the top mass, and the red lines are the masses below, in order.

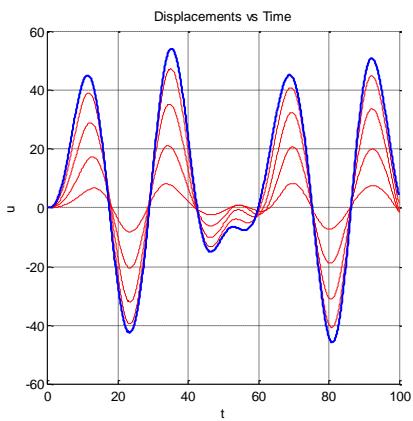


Figure 1

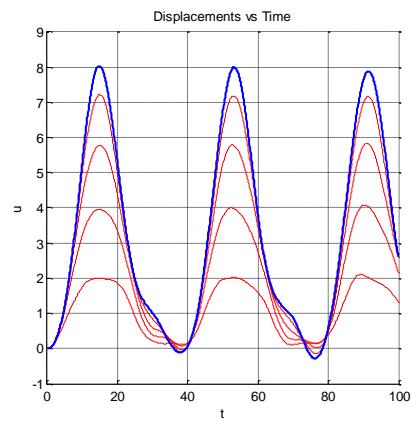


Figure 2

Some quick things to note here are that the model building sways much more than the non-model building, not just in terms of distance, but frequency. The model building also generates so much energy that it sways back *past* the initial starting point and travels in the opposite direction (see the displacement-time graph). The reason for this behavior cannot be readily tracked here, but instead be dissected.

Below are plots of the building's response with just the decreasing mass portion of the model. The forcing function is sinusoidal and equal in magnitude at every story.

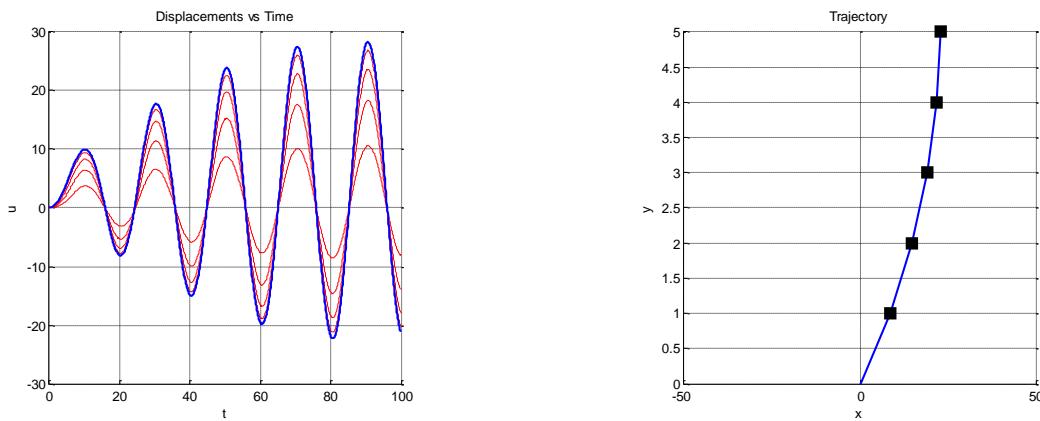


Figure 3

Interesting result! The displacement increases with time, but so does the *frequency*. The increase in displacement is easy to explain – the smaller the mass, the faster its acceleration will be under a constant loading. This is Newton's Second Law,  $F = ma$ . With a decrease in mass and constant force, the acceleration has to increase. But how can we explain the change in frequency? Is it just a phenomenon that has to do with mass, or is there something else that affects the system's overall frequency? Below is the response of the same building, but with the stiffness decreasing (sans the rest of the model).

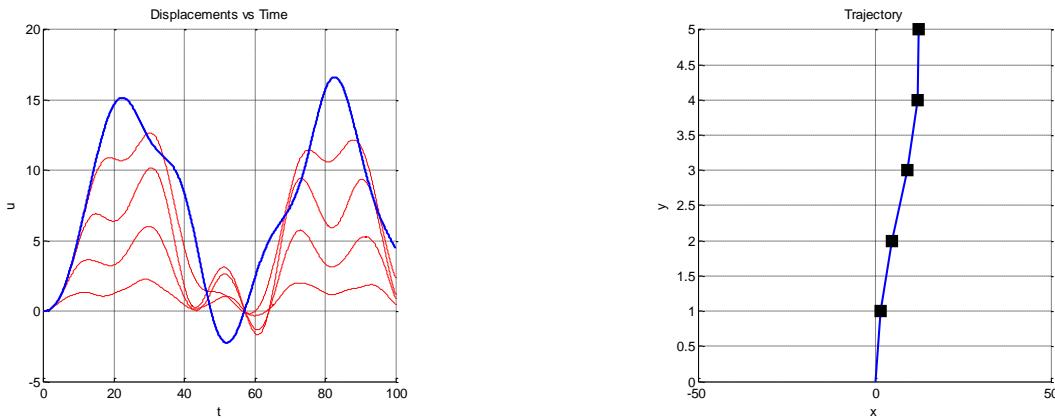


Figure 4

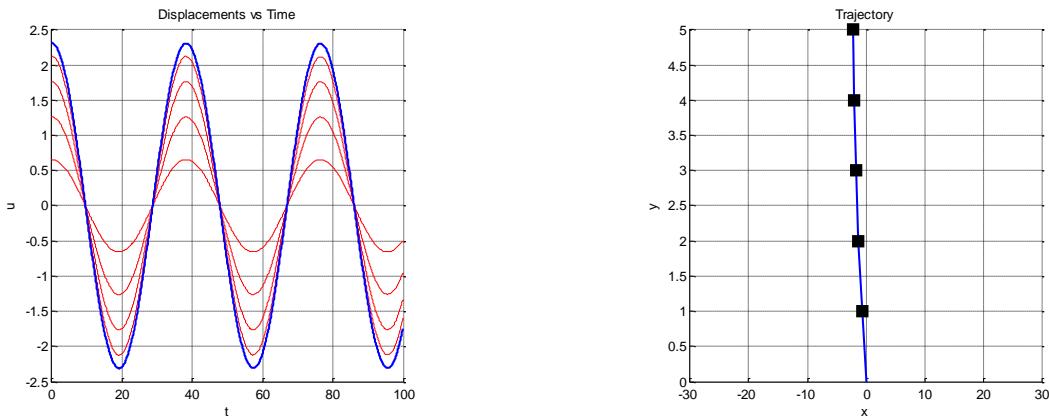
Changing the stiffness most definitely had an effect on the frequency of the system. The frequency was lowered drastically, and thus we can make the claim that stiffness plays a part in changing the frequency, as does the mass. But unlike decreasing the mass to increase the frequency, decreasing the stiffness *decreased* the frequency. From this discovery, we can conclude that the two are inversely related to the frequency, and in the equation form, we have,

$$\omega^2 = \frac{k}{m}$$

Where  $\omega^2$  is the square of the natural frequency of the system.

This result is well documented in undamped, free vibration response, and even though we had to displace the structure by a forcing function, we can still note the relationship the stiffness and mass share in the overall frequency of the structure.

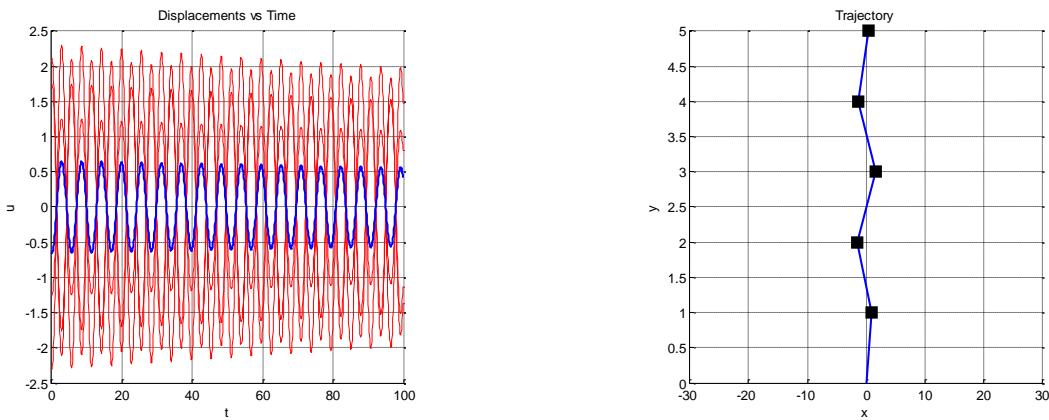
The natural frequency of the system, as defined above, can be found directly. But it is not just one frequency for any system, but the number of free degrees of freedom of the system. In the case of the building above, 5 masses (or stories) were present, and each of these were free to move in 1D space (do not get confused by the trajectory of the plot – it isn't really 2D). Thus, there are 5 free degrees of freedom in this system, and therefore 5 natural frequencies. A better description of this is that there are 5 frequencies to which the system can frequent under the undamped, free vibration condition. If this condition only holds true theoretically for free vibration, then the only way to witness these frequencies is to *displace* the structure in such a way that it does. This is where eigenvectors come from, and they stem from the eigenvalue problem (EVP). The EVP is essentially a nontrivial solution to the EOM of undamped, free vibration, and what comes out of the EVP are the eigenvalues (natural frequencies) and eigenvectors (displacement vectors). Thus, if we displace the structure at a particular eigenvector, we will witness the eigenvalue, or square of the frequency. Since it is undamped, the frequency is constant over time. What is apparent from this realization is that the system resembles consistent displacement *shapes*. We call these shapes modes (or mode shapes). If we solve the EVP and displace the system at the first eigenvector, we get the following mode shape (Figure 5).



**Figure 5**

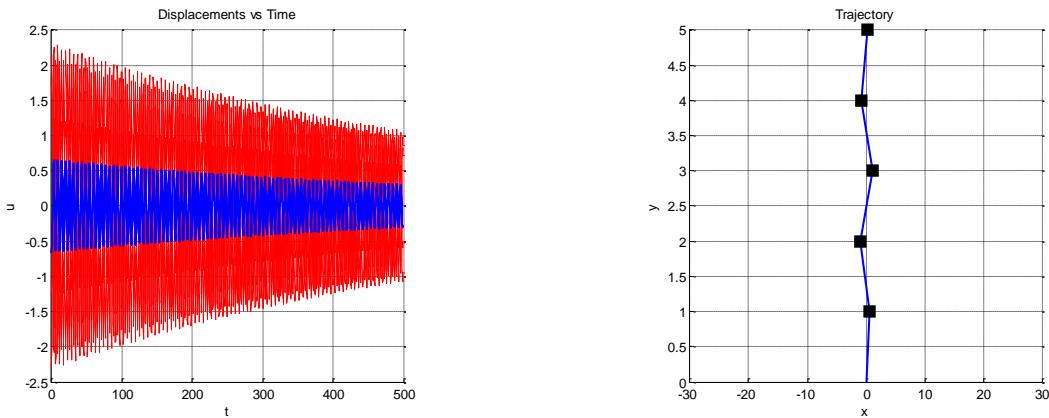
Note the perfect symmetry of the displacement-time graph, and the response seen in the trajectory. The building will continue to operate at this natural frequency.

If we examine the 5<sup>th</sup> mode shape (Figure 6),



**Figure 6**

The mode shape itself is an interesting result. Each mass moves about the other, generating a sort of zig-zag shape. The middle masses displace more than the top mass, which is interesting. However, there are two main takeaways here: the first is about the natural frequency – the frequency *increases* with higher mode shapes. The second takeaway (that ties into the first) is that the structure actually self-dampens at the higher mods! This is a more fascinating result! Below is the same system, ran for a longer time period.



**Figure 7**

Note the decrease in amplitude as time goes on. We must assume here that the structure dampens out at a *faster rate at higher modes*, since the energy is leaving the system faster because of the corresponding larger natural frequencies of each mass. This disproves the previously mentioned observation that lower modes vibrate at a constant rate – this is not true! The lower modes simply self-dampen at a lower rate.

One last thing to note here about the dynamic response of the ndof system is this: what happens when you *force* the system to vibrate at a natural frequency? Before now, we initially displaced the system to view the mode shape and witness the respected frequency, but what if wind forces (or even an earthquake) could force a system to, at least at one point in time, vibrate at a natural frequency? The result, unfortunately, can be catastrophic. But for this reason, the beast of what we call *resonance* cannot be ignored.

Resonance is when a system is forced to vibrate at its natural frequency, and the ensuing result is that the structure continues to displace more, and more. This can lead to eventual failure of the structure, and thus should always be arrested. Though we cannot avoid resonance from ever happening, we can arrest the motion by applying damping (more on this later...). The following plots (Figure 8) demonstrate resonance in the first mode shape.

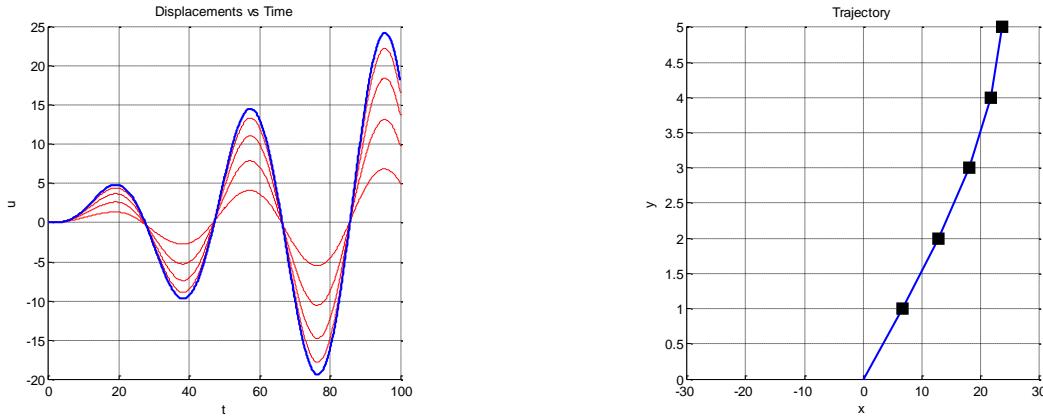


Figure 8

And there it is! The amplitude of the displacement-time curve continues to increase over time. If this behavior is not arrested, it will cause failure within the structure if it were to stretch past yielding. However, the rate of resonance decreases within the higher modes, since the self-dampening rate increases. It is this “battle” between the two phenomena is what is most interesting – the resonance beats out the self-dampening, and the fight can be traced in the displacement-time graph below. Note the structure is at the 5<sup>th</sup> mode.

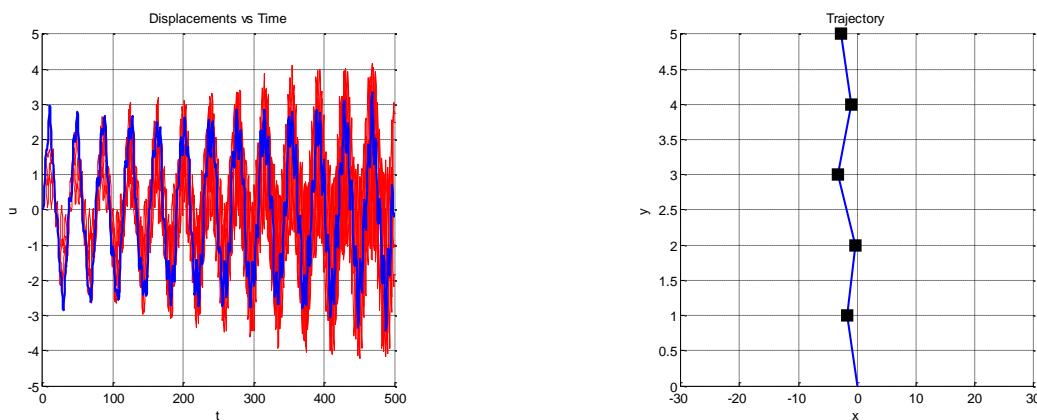


Figure 9

The dynamic qualities and characteristics of the bridge are exactly the same as in the building, though it is worth including in the report some of mode shapes that the bridge can undertake. Below is the first mode shape (Figure 10), which is similar to the response of the bridge.

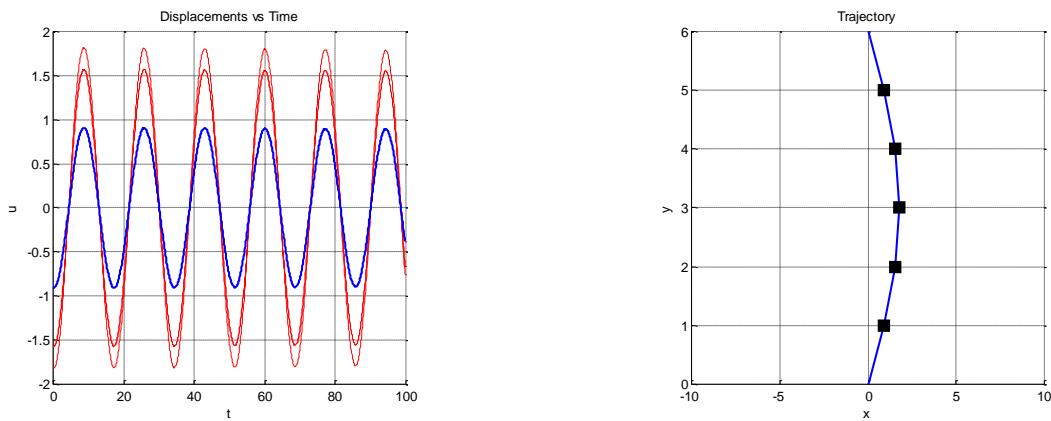


Figure 10

The 5<sup>th</sup> mode is shown below (Figure 11). Note the evident self-dampening of the system.

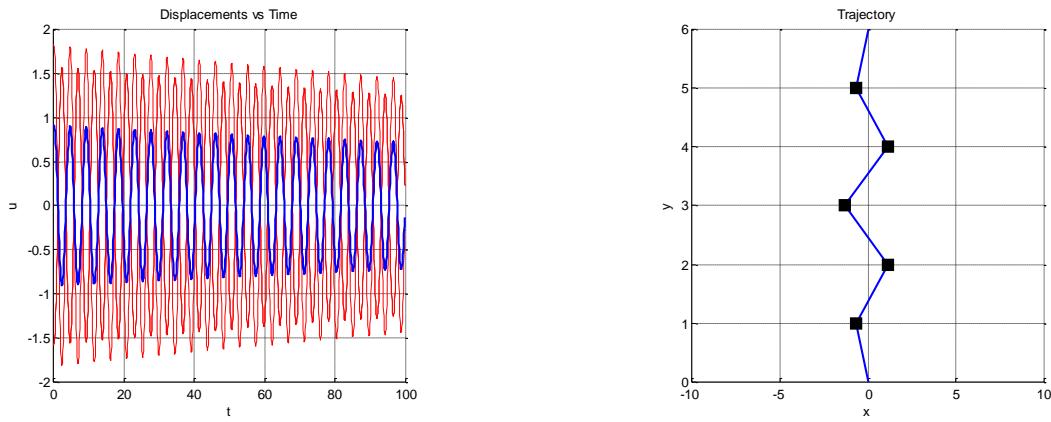


Figure 11

Further dynamic responses of the ndof system, including applied damping and elasto-plasticity, will be discussed in the next few homeworks.

## Damping of the NDOF System (Homework 8)

### Summary

Applied damping was implemented into the ndof code, and the dynamic response of both the damped bridge and building system were explored.

### The Implementation of Applied Damping

In order to implement damping into the system, we must first detail the EOM for the ndof system. Much like in the previous homeworks, damping is dependent on the velocity of the system, and thus the damping matrix  $C$  is attached to the velocity vector  $\dot{u}$  (first derivative of displacement).

$$M\ddot{u} + C\dot{u} + Ku = f$$

Where  $C$  is the damping matrix, and is defined as  $MV\mathbf{C}V^T M$ ,

$V$  is the eigenvectors of the system, and

$\mathbf{C}$  is a diagonal matrix that contains the damping scalars and natural frequencies of the system.

The special matrix  $\mathbf{C}$  is detailed as shown below,

$$\mathbf{C} = \begin{bmatrix} 2\xi_1\omega_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 2\xi_n\omega_n \end{bmatrix}$$

Where  $\xi$  is the damping scalar at each degree of freedom, and

$\omega$  is the natural frequency at each degree of freedom.

The damping matrix  $C$  can then be determined by matrix multiplying by the mass and eigenvector matrix, as shown below.

$$C = \begin{bmatrix} m_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & m_n \end{bmatrix} \begin{bmatrix} \vdots & \vdots & \vdots \\ V_1 & \cdots & V_n \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} 2\xi_1\omega_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 2\xi_n\omega_n \end{bmatrix} \begin{bmatrix} \vdots & \vdots & \vdots \\ V_1 & \cdots & V_n \\ \vdots & \vdots & \vdots \end{bmatrix}^T \begin{bmatrix} m_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & m_n \end{bmatrix}^T$$

Which results in an ‘n’x’n’ matrix. The damping matrix can then be implemented into the EOM.

### Approximating the Time-Dependent Variables (with damping)

The initial acceleration is solved much the same, but damping matrix must be included, as shown below.

$$a_{old} = M^{-1}(f - Kx_{old} - Cv_{old})$$

Once the initial acceleration is computed, the Newmark time stepping algorithm begins. The acceleration at the next time step is thus,

$$a_{new} = (M + \eta K + \zeta C) \backslash (F - (Kb_n + Cc_n))$$

Where  $\eta$  is the function  $h^2(\frac{1}{2} - \beta)$  from the Newmark equation for the new displacement ( $x_{new}$ ),

$b_n$  is the function  $x_{old} + hv_{old} + h^2\beta a_{old}$ , which is just the other portion of the Newmark equation for displacement,

$\zeta$  is the function  $h(1 - \gamma)$  from the Newmark equation for the new velocity ( $v_{new}$ ),

$c_n$  is the function  $v_{old} + h\gamma a_{old}$ , which is just the other portion of the Newmark equation for velocity, and finally,

$F$  is the sinusoidal forcing function,  $F = F_{o1} + F_{o2}\sin(F_{wo}t)$

The Newmark equations are listed below (again) for reference.

$$v_{n+1} = v_n + h[\gamma a_n + (1 - \gamma)a_{n+1}]$$

$$u_{n+1} = u_n + hv_n + h^2 \left[ \beta a_n + \left( \frac{1}{2} - \beta \right) a_{n+1} \right]$$

### The Effects of Applied Damping

We can begin the analysis by using the same system properties of the system in the previous homework. With 5 degrees of freedom, the model response of the building is shown below (Figure 1). Note that damping was scaled so that the most damping takes place at the top of the structure (since the wind forces increase with elevation).

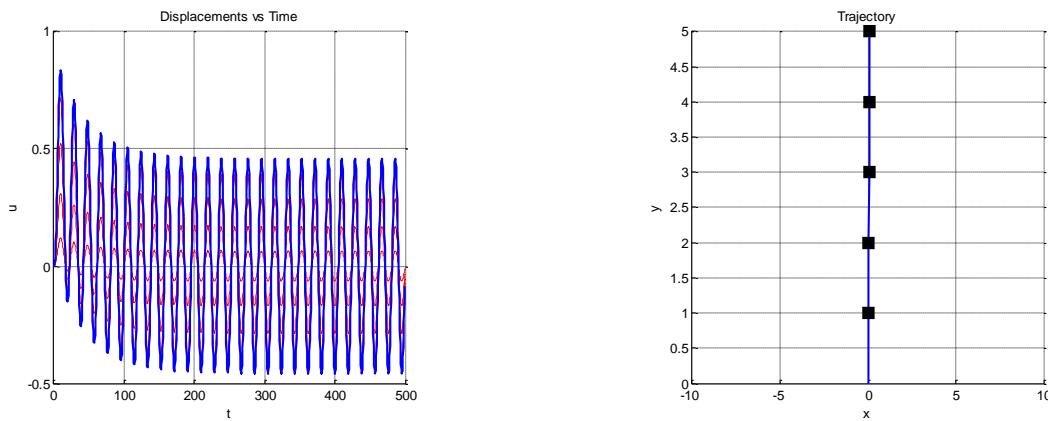


Figure 1

It is obvious to note here that the system does indeed dampen initially, but then after some time levels off and the system continues to vibrate under the given loading. This is expected, as the force applied is continuous over time. We can increase the damping scale factor even further, so that we can note that the structure levels off and vibrates continuously at a lower displacement amplitude (Figure 2).

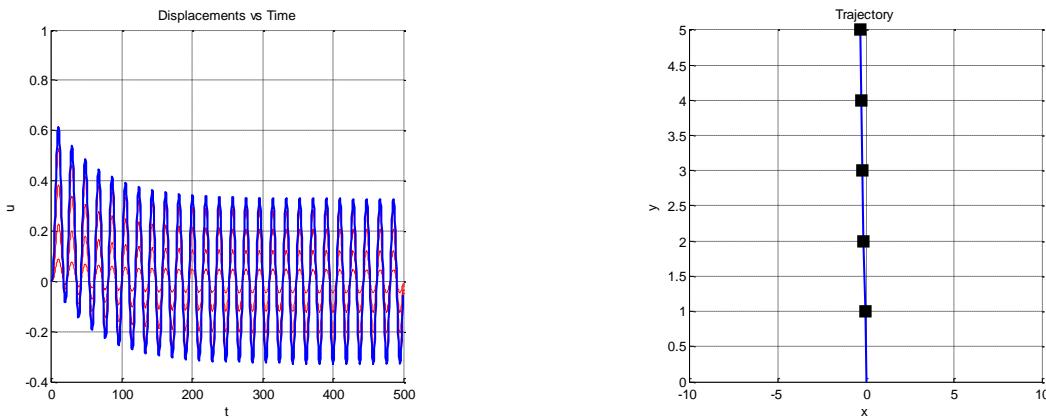


Figure 2

We can note that the displacement amplitude has been lowered. The best way to visualize the effects of damping however, is in the mode shapes of the system. Sans the model feature, the system is displaced under the first mode shape (Figure 3).

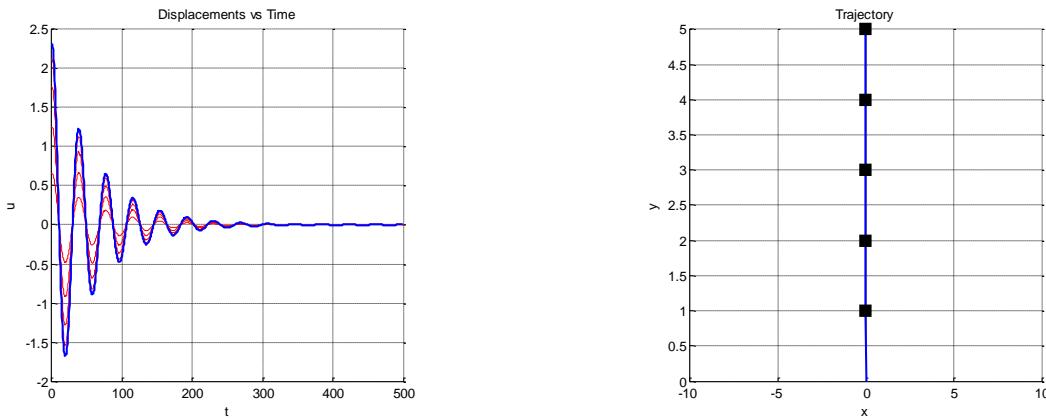


Figure 3

The effects of damping are obvious here. The motion of the system almost completely damps out relatively quickly, at approximately the 300<sup>th</sup> time point. Since we learned that dynamic systems self-dampen at a faster rate at higher modes, we can postulate that the added effects of applied damping will dampen the system more quickly in the 5<sup>th</sup> mode than the 1<sup>st</sup>. Figure 4 below is of the response of the system at the 5<sup>th</sup> mode.

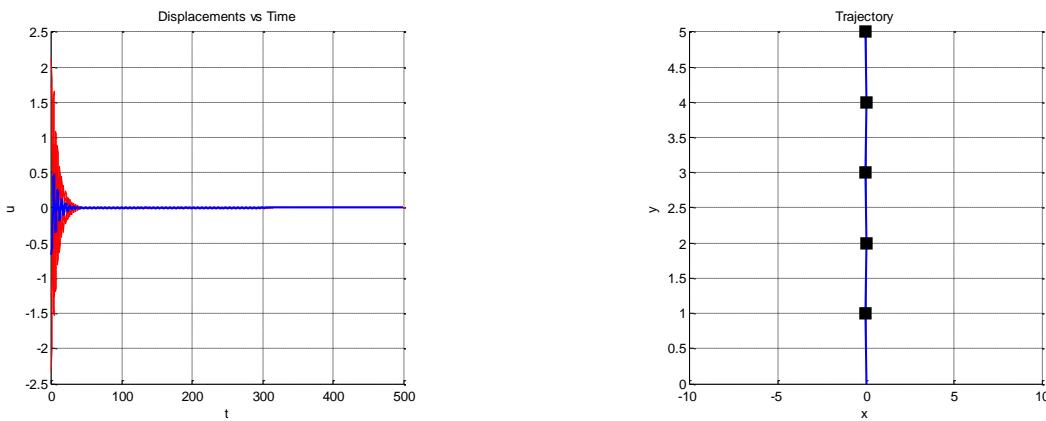


Figure 4

As expected, the system damps out quickly at approximately the 50<sup>th</sup> time point. This further proves the theory that higher modes self-dampen, and as mentioned previously, this must be due to the fact that the higher modes operate at a higher frequency and thus dissipate energy at a faster rate. Furthermore, we can deduce that because the damping rate in general increases with higher natural frequencies, an increase in stiffness/reduction in mass of the system will also increase the damping rate. And thus, we can further postulate that the bridge system (of the same properties of the building) will dampen out faster than the building because of the difference in the stiffness matrix (see Homework 7 for a breakdown of the matrices).

The below two figures (5 and 6) detail the damping of the 1<sup>st</sup> and 5<sup>th</sup> mode, respectively, and showcase the phenomenon that self-damping has on the system.

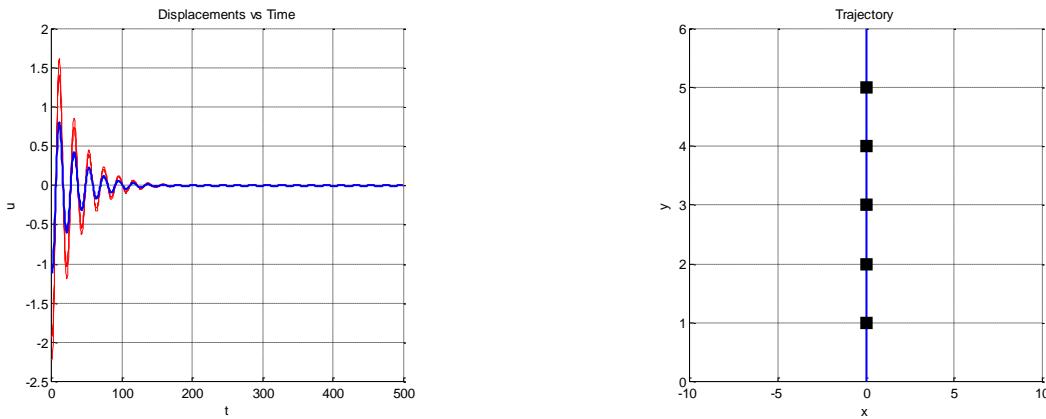


Figure 5

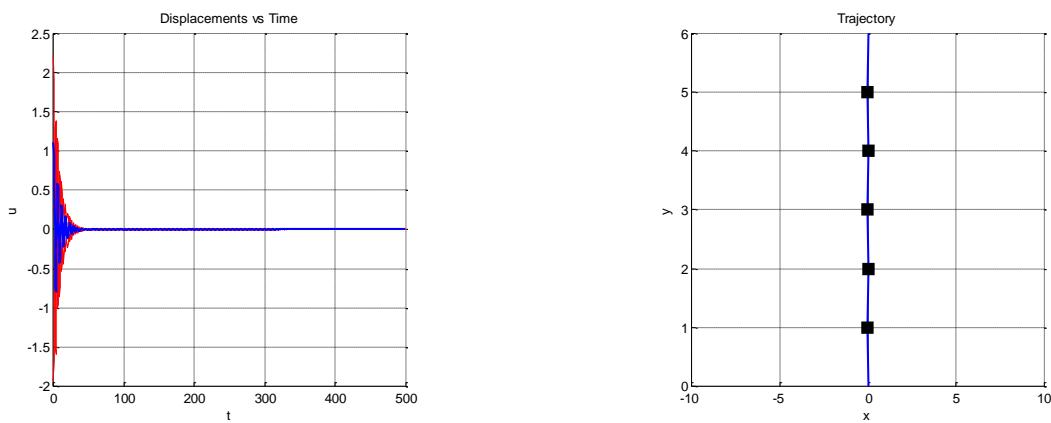


Figure 6

As postulated, damping does happen at a faster rate for the bridge! This further reinforces the effect that natural frequencies have on the rate of damping.

## Elasto-Plastic Response of the NDOF System (Homework 9)

### Summary

In this assignment, elasto-plasticity was added to the ndof system in the form of non-linearity of restoring forces within the elements. Newton's Method was used to solve for the nonlinear EOM and Newmark's method was again used to approximate the time-dependent variables at the time step.

### The Restoring Force

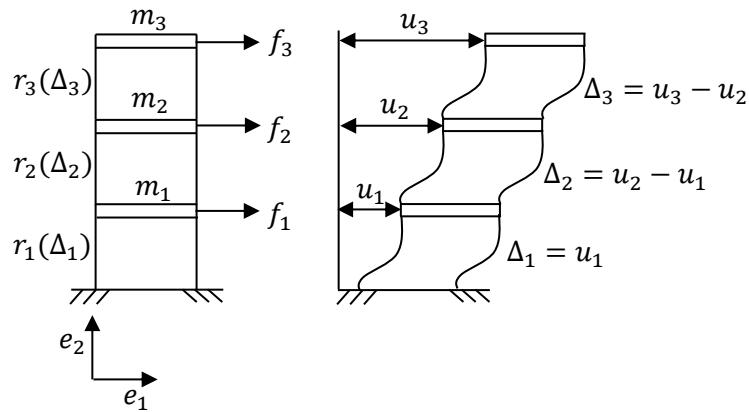
In order to provide an elasto-plastic response of the system, the restoring force within each element (spring) had to be become nonlinear. Though not in the sense that the restoring force in the spring is nonlinear by algebraic means, but rather that the spring can take on two states of stiffness. In the elastic sense, we wish to have the force in the spring always as such,

$$R = ku$$

Where  $R$  is the restoring force in the spring.

This is linear by nature, but the most important analysis a structural engineer can perform is one that describes the forces within the structure in order to make sure it is safe. Thus, we need to keep track of the force in the spring and if it ever is greater than what is allowed (yield force), we deem this as a structural safety issue. In the model sense, we need to make the stiffness zero in an element that is past yielding, so that we can observe the effect that the failed element has on the entirety of the structure. Elasto-plasticity is simply an elementary model of how a real-life structure would behave in the case of a failed structural member.

In order to fully implement elasto-plasticity in the ndof code, we must also keep track of the displacements between the masses. Take the model below (Figure 1) for a 3 degree-of-freedom system.



**Figure 1**

Similar to Figure 2 in Homework 7, we can draw the FBD's of the masses, summing the forces and applying Newton's Second, we have the following EOM,

$$M\ddot{u} + C\dot{u} + R(u) = f$$

Where  $R(u) = \begin{Bmatrix} r_1 - r_2 \\ r_2 - r_3 \\ r_3 \end{Bmatrix}$  and  $\Delta = \begin{Bmatrix} u_1 \\ u_2 - u_1 \\ u_3 - u_2 \end{Bmatrix}$

Note that the restoring force for each degree of freedom is  $r_n = k_n \Delta_n$ . By calculating the restoring force for each element, we can now devise a method to check if the force exceeds the yield force, and if so, set the stiffness equal to zero and the restoring force equal to the yield value. We must also make sure to record the plastic strain of the elements throughout the time stepping process.

### *Elasto-Plasticity and Nonlinearity Implementation*

In order to render elasto-plasticity, we must devise a system to check for yielding in the spring and if it does yield, we must record the plastic strain. The plastic strain is recorded, as shown below.

$$u_{p,i} = \Delta_i - \frac{N_o}{k_i}$$

Where  $\Delta_i$  is the displacement between adjacent masses,

$N_o$  is the yield force, and

$k_i$  is the stiffness of the mass.

This procedure is implemented first with the initial conditions of the system, and then within the time-stepping algorithm. In order to handle the nonlinear EOM, Newton's Method was used to approximate the acceleration at every time step. The residual and gradient equations for use in the Newton loop are shown below.

$$\begin{aligned} g &= Ma_{new} + Cv_{new} + R - F \\ A &= M + C \frac{dv}{da} + \frac{dR}{d\Delta} \left( \frac{d\Delta}{du} \right) \left( \frac{du}{da} \right) \end{aligned}$$

We can note that the derivative of the velocity with respect to the acceleration at the next time step is simply,

$$\frac{d}{a_{new}}(v_{n+1}) = \frac{d}{a_{new}}(v_n + h[\gamma a_n + (1 - \gamma)a_{n+1}]) = h(1 - \gamma) = \zeta$$

Similarly, the derivative of the displacement with respect to the acceleration is thus,

$$\frac{d}{a_{new}}u_{n+1} = \frac{d}{a_{new}}\left(u_n + hv_n + h^2 \left[ \beta a_n + \left(\frac{1}{2} - \beta\right) a_{n+1} \right] \right) = h^2 \left( \frac{1}{2} - \beta \right) = \eta$$

The more complex derivative is  $\frac{dR}{d\Delta} \left( \frac{d\Delta}{du} \right)$ , which is actually fairly simple if we note that first, for a 3 DOF system,

$$\frac{d\Delta}{du} = \begin{bmatrix} \frac{d\Delta_1}{u_1} & \frac{d\Delta_1}{u_2} & \frac{d\Delta_1}{u_3} \\ \frac{d\Delta_2}{u_1} & \frac{d\Delta_2}{u_2} & \frac{d\Delta_2}{u_3} \\ \frac{d\Delta_3}{u_1} & \frac{d\Delta_3}{u_2} & \frac{d\Delta_3}{u_3} \end{bmatrix}$$

And for the derivative of the restoring force with respect to the displacement between masses,

$$\frac{dR}{d\Delta} = \begin{bmatrix} \frac{dR_1}{\Delta_1} & \frac{dR_1}{\Delta_2} & \frac{dR_1}{\Delta_3} \\ \frac{dR_2}{\Delta_1} & \frac{dR_2}{\Delta_2} & \frac{dR_2}{\Delta_3} \\ \frac{dR_3}{\Delta_1} & \frac{dR_3}{\Delta_2} & \frac{dR_3}{\Delta_3} \end{bmatrix}$$

And thus, the tangent stiffness matrix is,

$$T = \frac{dR}{d\Delta} \frac{d\Delta}{du}$$

For the linear case (when stiffness doesn't change), the tangent stiffness matrix is simply  $K$ , the typical stiffness matrix. Now, the complete gradient equation is thus,

$$A = M + C\zeta + T\eta$$

Of course, the acceleration is then approximated by the following,

$$a_{new} = a_{new} + A \backslash g$$

### *The Elasto-Plastic Response*

For the first analysis, we will apply resonance to the structure so that the elements yield after some time. The same ndof structure properties are used for this analysis as in the previous ndof homeworks. With resonance in the first mode, we can view the yielding of the members (red) in the trajectory plot, as well as the force-displacement plot to note the nonlinearity of the system.

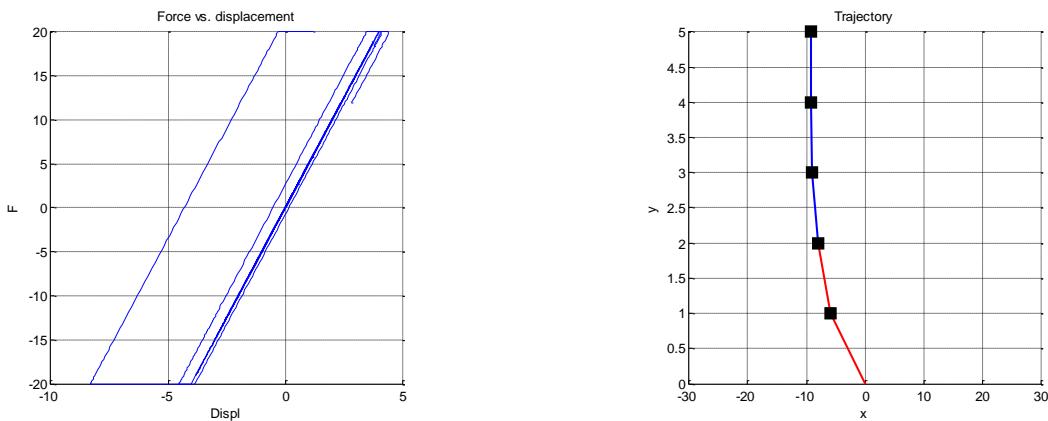


Figure 2

As can be seen in the trajectory plot, the lower two elements yielded. This is further reinforced in the force-displacement plot. Note that the restoring force levels off to show that it reached the yield state. We can add damping to the structure to settle the displacement (and thus lower the restoring force) in order to prevent yielding. The below plots are of the structure with a small amount of damping provided at all degrees of freedom.

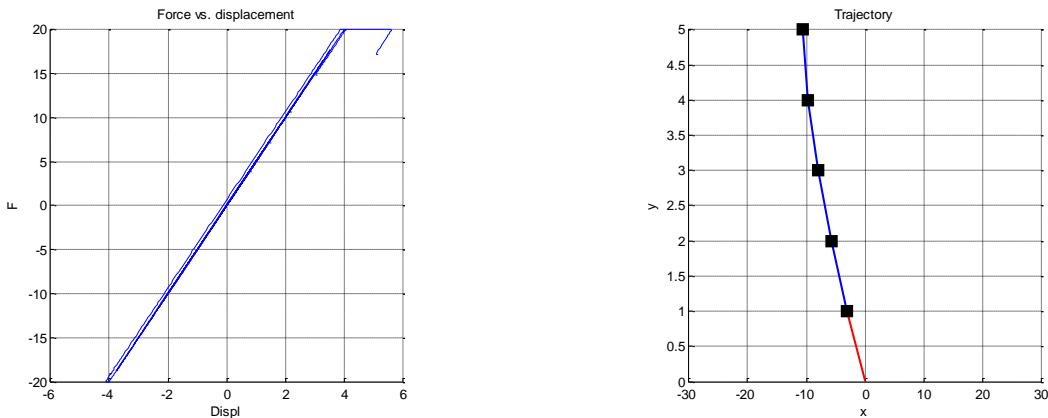


Figure 3

Note the stark change in the response of the structure! Only one element yielded, and the result can be verified in the force-displacement plot. We can further verify the elasto-plastic state by noting the frequency of the system. Yielding members should decrease the tangent stiffness matrix of the system, which should in turn decrease the overall frequency of the system.

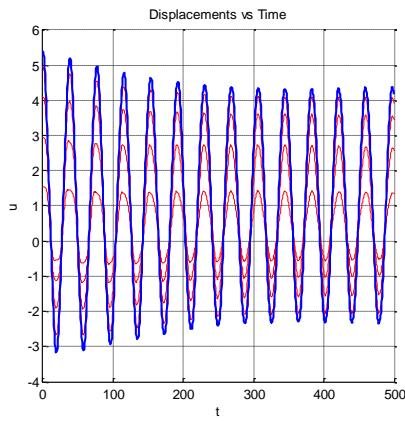


Figure 4

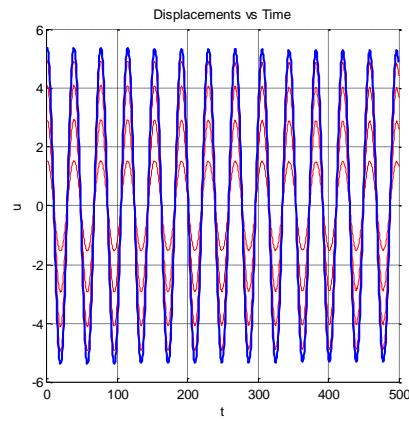


Figure 5

We can note here that there is slight decrease in the frequency of the system between Figures 4 and 5. Figure 4 obviously depicts the motion of the yielded structure, as the amplitude decreased as the elements yielded. Another analysis we can perform is on which elements yield first in the system. We must note, however, that the force in an element depends only on the stretch in that same element, and not on the other elements. Applying resonance to the structure once again, but this time for 25 degrees-of-freedom, note the plots below (Figure 6)

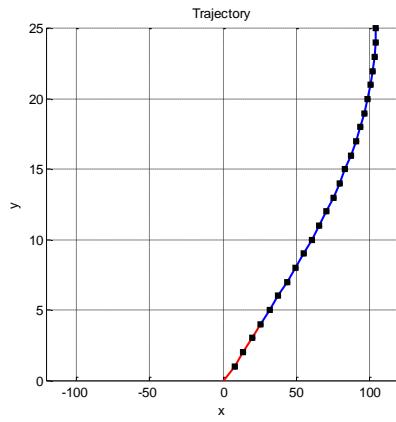
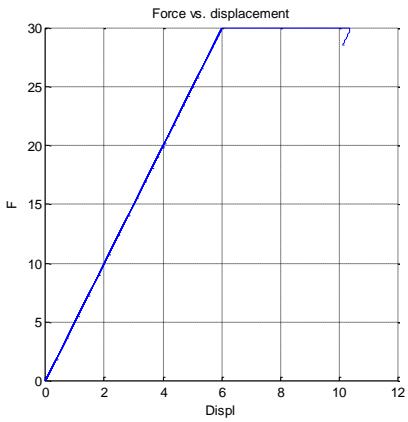


Figure 6

The elements again yielded first at the base, and then propagated upwards. The reason for this is that the base elements are being stretched the most, and thus increasing the axial force within. Since the stiffness is proportional to the axial force, lowering the stiffness in the bottom members should provide for safety against yielding. The downfall to decreasing the stiffness of the structure is that the frequency will decrease and thus the self-dampening of the system will not be as effective. In fact, if the stiffness is decreased so much that the self-damping rate is lower than the rate of resonance, yielding of the bottom members could happen regardless. It is in this example that a structural engineer should take care in when and which elements of a structure should be stiffened.

## Earthquake for the NDOF System (Homework 10)

### Summary

The ground displacement response (earthquake) of a structure at the supports points was examined in this homework. The complete dynamic response (damping, elasto-plasticity), was analyzed and documented.

### The Earthquake and EOM

Ground displacement at the structures supports points is essentially an earthquake, that of which can be modeled into the ndof system by creating a 3 degree-of-freedom model and summing the forces of the FBD's that result. Below is the typical building geometry that was used in the previous ndof homeworks.

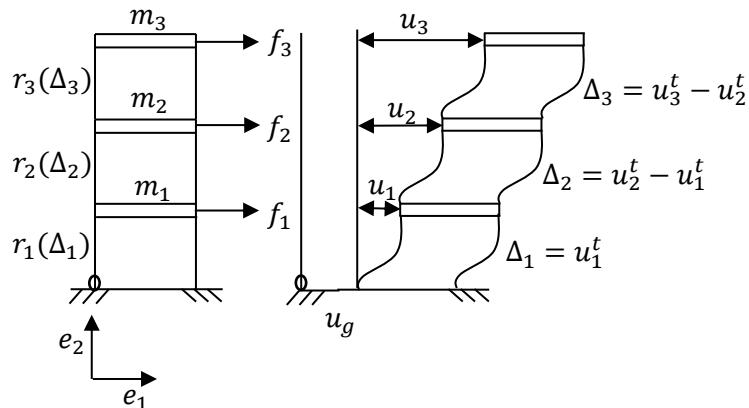


Figure 1

Figure 2

The distinction here and the other displacement shapes is obviously the ground displacement  $u_g$  that we now account. Note that,

$$u_3^t = u_g + u_3$$

$$u_2^t = u_g + u_2$$

$$u_1^t = u_g + u_1$$

And thus, if we some the forces and equate to Newton's Second, we have,

$$m\ddot{u}_3^t = -r_3(\Delta_3)$$

$$m\ddot{u}_2^t = r_3(\Delta_3) - r_2(\Delta_2)$$

$$m\ddot{u}_1^t = r_2(\Delta_2) - r_1(\Delta_1)$$

And thus, if we account for the complete dynamic response, we have the EOM. Note that the mass matrix that is multiplied by the ground acceleration appeared on the left side originally.

$$M\ddot{u} + C\dot{u} + R(u) = f - M1\ddot{u}_g$$

It is now as simple as applying the EOM to the ndof system, and with a few minor tweaks to the program, we can compute the dynamic response of an earthquake.

### *Implementing the Earthquake EOM*

In order to properly implement the earthquake motion into the ndof code, an artificial earthquake response had to be decided upon before the ground displacement terms could be added to the EOM. This response has the form (per Dr. K. D. Hjelmstad),

$$E_q = E_{00}te^{-E_{01}t}[(E_{11}\sin(E_{12}t) + E_{21}\sin(E_{22}t) + E_{31}\sin(E_{32}t) + E_{41}\sin(E_{42}t)]$$

Where  $E_q$  is the initial ground acceleration at the support point.

The constants,  $E_{00}$ ,  $E_{01}$ , etc are simply earthquake parameters that affect the magnitude of the ground acceleration. From this, we can solve for the initial acceleration of the structure as,

$$a_{old} = M \backslash (f - (M1E_q + R + C_{vold}))$$

Since the ground acceleration is of course dependent on time, we must include the  $E_q$  equation in the time-stepping algorithm. It is in this step that must make sure that we set the velocity of the earthquake equal to zero after the set duration of the earthquake. This will ensure that the ground acceleration will be zero after the duration of the earthquake (and thus there will be no ground displacement as there is no velocity). Since the EOM is of course still nonlinear (elasto-plasticity), we can note the change in residual and gradient,

$$g = Ma_{new} + Cv_{new} + R - (F + M1E_q)$$

$$A = M + C\zeta + T\eta$$

Since we are taking the derivative with respect to the system's acceleration, the terms attached to the ground acceleration do not appear in the gradient as they are treated as constants in this instance.

Using Newmark, the new velocity and position of the support point are calculated the exact same way as for the velocity and position of the degrees of freedom of the system.

### *The Earthquake Response*

We can first analyze the trust model condition of the structure, this time being displaced horizontally by an earthquake. Note that 10 degrees-of-freedom were used in this structure, to better note the compounding effects ground displacement has on the entirety of the structure. Below are two plots, one of which is the ground acceleration per time, and the other trajectory of the system.

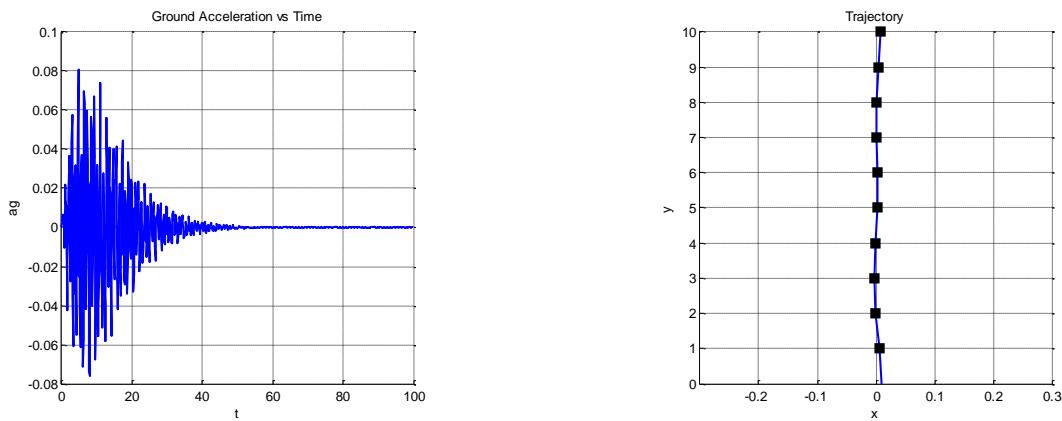


Figure 1

We can note immediately the violent nature of the ground acceleration by observing the above plot. The ground acceleration soon decreases to zero at the end of the earthquake duration. The trajectory plot is also worth noting, as it can be seen that the support point is shifted over, causing waves to propagate throughout the structure. We can better observe wave propagation from ground displacement by turning the model feature off and solely displacing the support point.

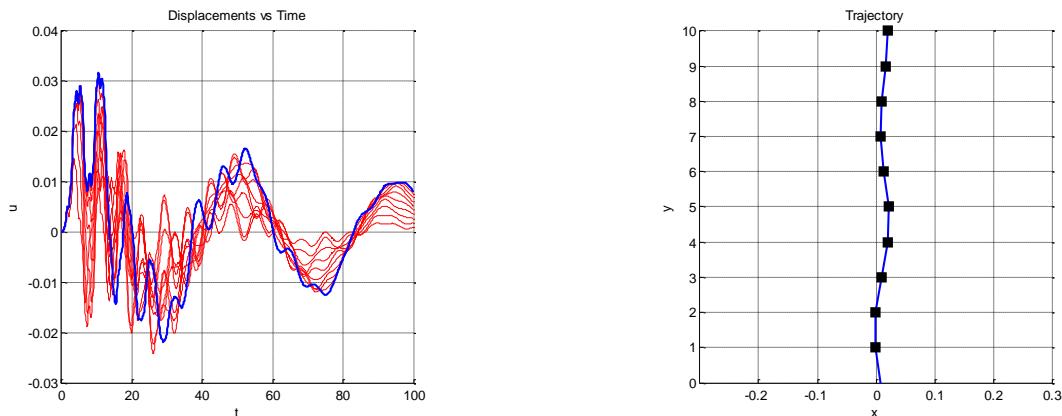


Figure 2

Note the displace-time graph above. The displacement of the masses is evidently dramatic, but settles down and begins to become uniform after some time. This *is the nature* of the response of the structure displaced by an earthquake! The beginning is violent, with severe unorthodox displacing of the masses, but once the earthquake starts to slow, the acceleration of the masses begins to slow down. But the earthquake can cause catastrophic results on the structure if it causes the structure to yield. Take the below example of the previous structure, but with a reduced yield scale.

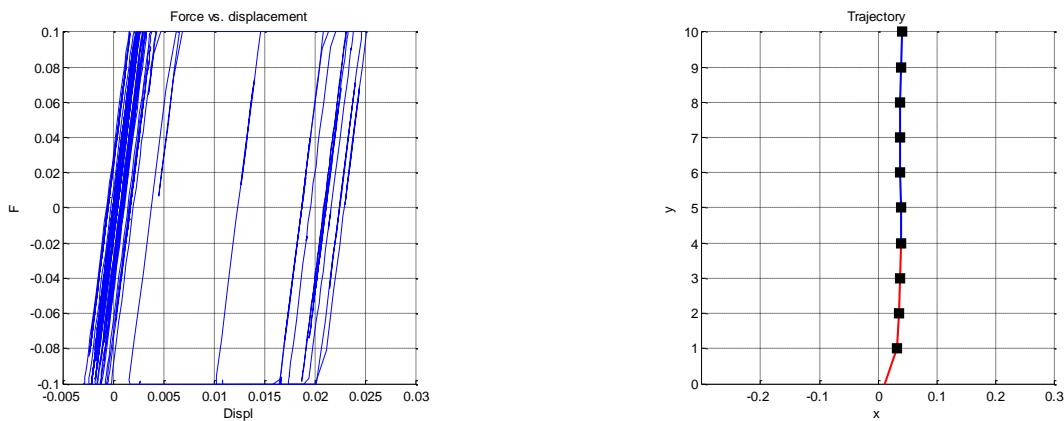


Figure 3

As we can see from the trajectory plot, the lower element yielded and caused permanent deformation of the structure. This can lead to catastrophic results since the lower element is a severe support member for the entirety of the structure. We can note the multiple yielded members from the force-displacement plot.

We simulate the result of built-up resonance within the structure due to the earthquake by applying a small forcing function with a frequency equal to one of the mode shapes. Below are the plots for this system under the first mode.

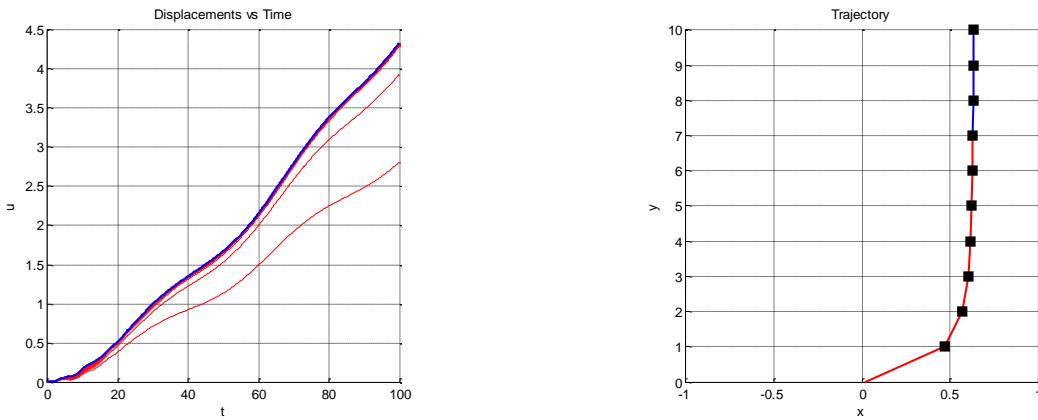


Figure 4

The result of built-up resonance is actually quite obvious. Because of the yielding in the bottom member due to the ground displacement, the compounding effects of resonance further propagate the structure to displace even more (displacement-time plot). Considering that the base element has yielded, it will continue to permanently deform in this model, but the reality is that this member would eventually fail and cause collapse of the structure. This type of failure is called shear failure, and is prevalent in failure of structures due to ground displacement. To prevent this failure, some buildings (called “smart” buildings) are implemented with applied dampers, typically at the top of the structure. We can simulate the response of a “smart” building by damping each mass proportionally using the model feature (most damping at the top).

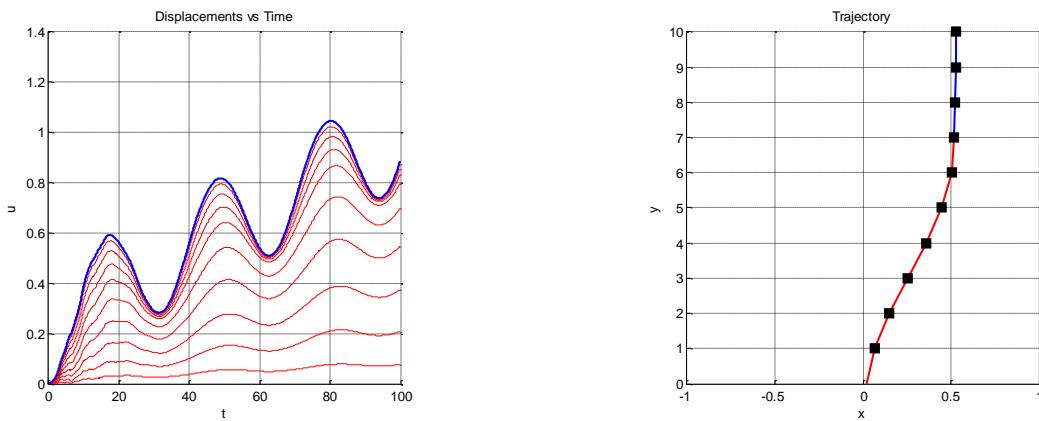


Figure 5

Note the yield shape of the structure now with applied damping. The bottom support element is not as severely displaced as before, and the shape is surely not as catastrophic. We can note the uniformity in the displacement-time plot of the masses to evoke that applied damping is in fact resolving the system. We have to note though however, that resonance will still continue to displace the structure because of the yielded elements. It is because of this observation that we must make sure that no elements yield during the phase of resonance. If we dampen the system further,

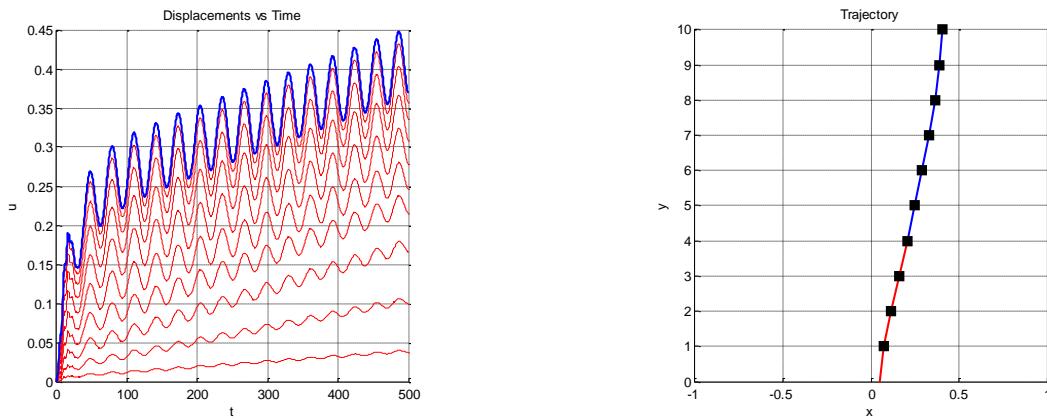


Figure 6

Two things to note here: 1) less of the elements yielded, and 2) the overall displacement of the masses is lower and not as severe in the previous system. From this, we can deduce that because less elements yielded (and therefore the stiffness of the structure was not reduced as much), the displacement of the masses decreased. Careful design consideration can be made in the stiffness of the elements and damping of the structure so as to reduce the violent effects of resonance due to earthquakes (and resonance in general).

## The Static and Dynamic Truss (Homework 11 and 13)

### Summary

In this assignment, the modification of the linear, static, elastic truss program was explored. The truss program was modified so that non-linearity and elasto-plastic response of the truss would be obtained, and for preparation of eventual conversion from static to dynamic.

### A Breakdown of the Truss Program

The truss program provided contains multiple script files that perform various algorithms to establish the appropriate response of a static, linear truss under imposed loading. The ‘main’ script performs the necessary input “calling” from other scripts – it is the basis for this program as all scripts (in one way or another) pass through main. In order to prep the program for a nonlinear analysis (for the dynamic response), a newton loop will be established to solve for the displacement of the free degrees-of-freedom nodes. This loop, though takes place within the script main, is a function of the assembly of the tangent stiffness and residual matrix. It is because of this “go-around” that we must first explain in some detail what each script does, and how each contributes to the overall response of the truss.

The geometry of the truss is set using the ‘inputs’ script. The truss can be created in both 2D and 3D (numdim), and can be easily set within the inputs script. The number of nodes (numnod), number of elements and number of material sets can all be prescribed in the inputs script, as well as the obvious nodal coordinates and connectivity of the elements to the respected nodes. Boundary conditions and the forces on the nodes can also be applied here, where a “1” dictates a fixed degree-of-freedom (DOF) in the respected coordinate and a “0” for a free DOF. The nodal force array is simply the force applied (negative or positive) on any node. Since this is a truss program, bending is not considered as all element-to-element nodes are considered free DOF’s (and thus zero moment). It is also assumed that all forces applied to the structure (wind, ice, etc.) can be broken down as nodal forces, and thus elements are only subjected to compression/tension.

In order to properly assemble the stiffness matrix (and eventual residual), the ‘bound’ script is executed, which creates an array (‘id’) containing sequential numbers relative to the boundary conditions of the problem. For instance, free DOF’s are represented here as positive integers (jpos), where fixed DOF’s are negative (jneg), and are tabulated in sequential order. The use of the size of jpos is used extensively as this details the numbers of free DOF’s. Both arrays ‘id’ and ‘jpos’ are used continuously throughout the program, especially in the assembly routine to produce the stiffness matrix.

Within ‘assemble’, the stiffness matrix is created. We must first create the pointer array (‘ii’) which essentially creates a vector array (dimension of the element stiffness matrix) for each element within the localize routine. The nodal coordinates of each element (i-node and j-node) are also collected here in the localize routine. From there, these arrays are passed back into the assembly routine. Before the global stiffness matrix can be calculated, the element stiffness (and eventual element residual) for each element must first be determined in the ‘elem’ script. The elem script calculates the unit vectors between all prescribed nodes from the inputs script. From there, the element stiffness is easily calculated by,

$$k_e = \frac{EA}{L_e} n_e n_e^T$$

Where  $\frac{EA}{L_e}$  is the axial stiffness derived from the empirical stress-strain equation, and

$n_e$  is the unit vector in the direction of the element.

Note that in the ‘inputs’ script, we can adjust the value of  $EA$  to linearly proportion the stiffness in the elements. The element stiffness is then placed into the “full” element stiffness (a square matrix of twice the number of dimensions). The element stiffness matrix then gets passed back into the assemble routine, and the stiffness matrix is calculated using the pointer array (‘ii’). We must note here that what we truly want out of the stiffness matrix (for a truss analysis, that is) is the free DOF’s (or jpos). This is because the fixed nodes are not displacing (hence fixed) and thus are not needed in the calculation of the displacements and reactions. This is where the ‘solve’ script comes in! Note that in the non-linear truss program, solve is not needed as the virtual work method is used, along with the subsequent residual matrix. In solve, the stiffness matrix is partitioned into segments of jpos and ndof-jpos. Essentially, the stiffness matrix is partitioned between what is a free DOF and what isn’t. The displacement for each jpos is calculated, along with the reactions forces associated at the nodes. Alas, the vector ‘u’ and ‘r’ are assembled, which are the displacements and reaction forces of the entire nodal system, respectively. It is here that I should note that the prescribed displacement of the fixed nodes is simply the applied force at that fixed nodes. The fixed nodes will only displace as much as the applied force.

Exiting the solve script, the last step is to compute the overall stresses in the elements due to the displacements. Once again using the pointer assembly array, the displacement for each i-node and j-node of each element is determined, and then the strain in the element is calculated by taking the difference of the displaced i and j-nodes, normalized by the length of the element and then multiplied by the vector pointing in the element direction. The stress is simply the product of the strain and the associated  $EA$  material constant for the element.

### *Going from Linear to Nonlinear,*

If a basic understanding of the truss code can be made, the alterations to provide nonlinearity are not cumbersome. The need for the solve routine is no more, as the residual matrix (sum of the nodal forces) is calculated within the assemble routine and the global displacement vector is created based on the approximations of the jpos displacements in the Newton loop. The first change of the linear code begins in the elem script. In order to produce the element residuals (and eventual global residual matrix), the Lagrangian strain, element force, new element stiffness and “special E” must be calculated. Special E is simply the sum of the unit element vector and difference of the displacement of the i-node and j-node of the element, normalized by the element length. The equation is shown below.

$$E = n_e + \left(\frac{1}{h}\right) u_e$$

Where  $u_e$  is as detailed above and

$h$  is the length of the element.

The Lagrangian strain is calculated as,

$$E_e = \frac{n_e^T}{h} u_e + \frac{1}{2h} u_e^T u_e$$

Note that the first term of the Lagrangian strain equation is the linear portion. The second term is provided for nonlinearity, but if strains are small enough, this term becomes zero. The element force can be calculated as,

$$N_e = (EA)E_E$$

And finally, the residual in each element (internal resistance) is,

$$g_e = EN_e$$

The element stiffness can be calculated, as shown below. Note that the element stiffness matrix will then be used in the assembly of the tangent stiffness matrix (stiffness matrix at each new, approximated displacement) by using the pointer assembly to correct for each element's nodal displacements. This same procedure is also used for the global residual matrix.

$$k_g = \frac{\mathbf{E}(EA)\mathbf{E}^T}{h} + \frac{N_e \mathbf{I}}{h}$$

Where  $\mathbf{I}$  is the identity matrix of the number of dimensions of the system.

Passing back both the element residual and element stiffness, the tangent stiffness matrix and global residual matrix can be calculated. Once again, only the jpos are of interest, as the fixed node displacements do not change. The last change remains in the main script, as the Newton while loop must be implemented to determine the new jpos displacement. Utilizing the same procedure outlined in Homework 5, the new displacement of the jpos nodes can be calculated iteratively. The new residual is determined from setting the equation of motion (EOM) equal to zero,

$$g = g - F$$

Where  $g$  is the residual vector partitioned with only the jpos residuals, and

$F$  is the applied nodal force vector partitioned with only the jpos applied nodes.

Therefore, the approximation of the displacement is thus,

$$x_{new} = x_{new} - A \setminus g$$

It is worthy to note that the global displacement vector that passes through the assemble routine is a concatenation of the approximated jpos displacement  $x_{new}$  and the prescribed displacements (loads at the fixed nodes). Once Newton is done iterating,  $x_{new}$  is updated into the global displacement vector and passed onto the stress script to calculate the stress in the displaced elements.

### *Implementing Elasto-Plasticity*

In order to implement the elasto-plastic response of the truss, several key elements within the scripts had to be implemented. Using the same procedure as first seen in Homework 6, the force in each element was checked against the prescribed yield value (initiated in the inputs script), and set equal to that yield value if found greater than it before. Furthermore, to simulate “failure” of the material, the element stiffness was set to zero when the element was found to yield. The change in element stiffness and force are then noted in the creation of the tangent stiffness matrix and residual, and then the new displacement is calculated. The global displacement vector is then updated like usual with the new jpos displacements, and then the stresses in the displaced elements are calculated in the stress script.

### Observing the Response

With the modifications made to the truss program to include nonlinearity and an elasto-plastic constitutive model, we can now observe the response of the truss. Below is a simple 2D illustration of a “Box” truss (or variation of the Kingpost truss design) with/without prescribed loading on the nodes.

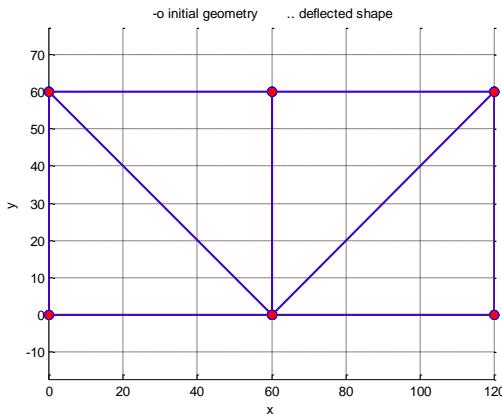


Figure 1

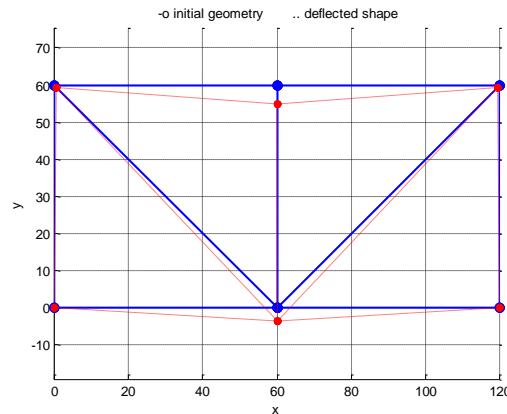


Figure 2

Note the deflected shape of the elements and middle nodes. A downward loading was placed on the top middle node in order to generate the displacement shown. The top left and right nodes also displaced somewhat, but the lower two did not, as those were fixed in both dimensions. We can note that there are 8 free degrees of freedom within this truss (and thus 8 mode shapes, but more on that later...), and so the displacement vector that is approximated within the Newton loop is for 8 displacements (2 for each free node). As mentioned previously, a prescribed loading applied in the direction of a fixed degree of freedom will simply displace that node in proportion to the magnitude of the load. Figures 3 and 4 showcase this result.

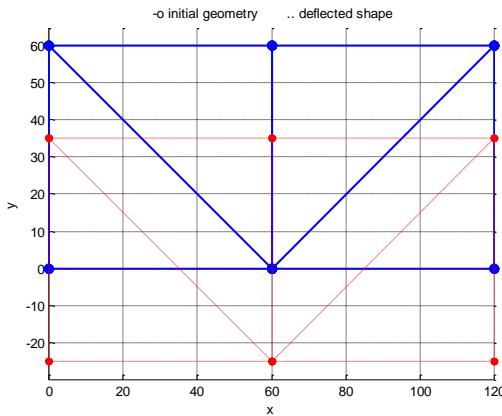


Figure 3

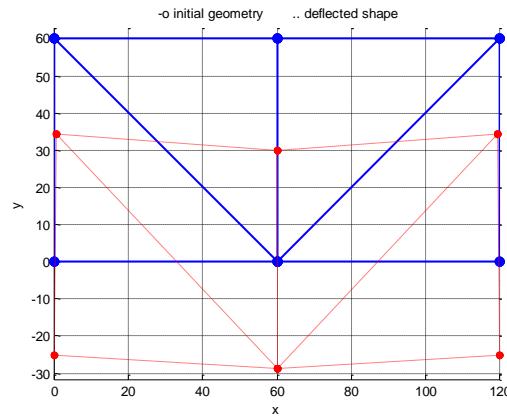


Figure 4

Note that the shape of the deflected truss remains identical between Figures 2 and 4. Changing the initial axial stiffness (or EA) constant , we can note the difference in the displacement of the truss. Using the same vertical loading at the top middle node (sans the fixed nodal loads) and increasing the axial stiffness of all members,

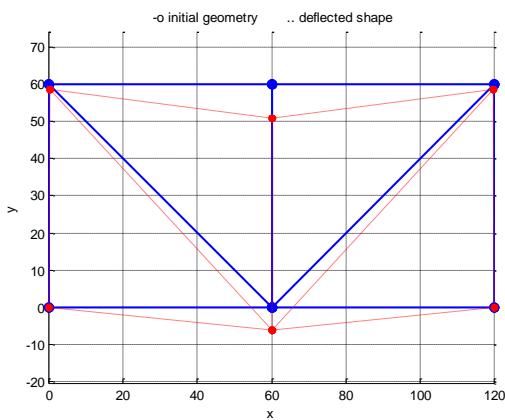
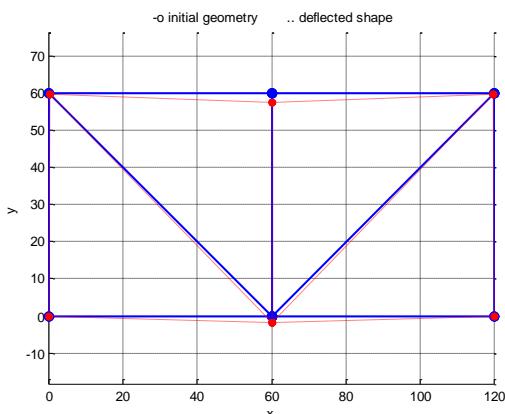


Figure 6

The stiffness in the web members was greatly reduced relative to the chords, and the result is dramatic. Increasing the stiffness of the web members decreases the overall displacement.

The truss does not displace as severely as in Figure 2. The stiffness of the members counteracts the displacement. We can note here that in a general mechanics analysis, the *strength* of a material is not the same as the *stiffness*. However, in the analysis of axial bars, the strength ( $E$ ) is directly proportional to the axial stiffness. We can experiment with the stiffness of the system by proportioning the stiffness among the top and bottom chord, as well as the web members. In a real-world truss, the top and bottom chords would be designed with a similar stiffness, where the web members would be less stiffened to provide additional buckling restraint and to take only some of the compressive/tensile forces.

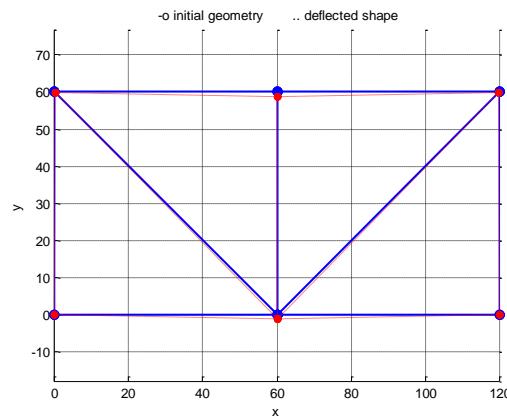
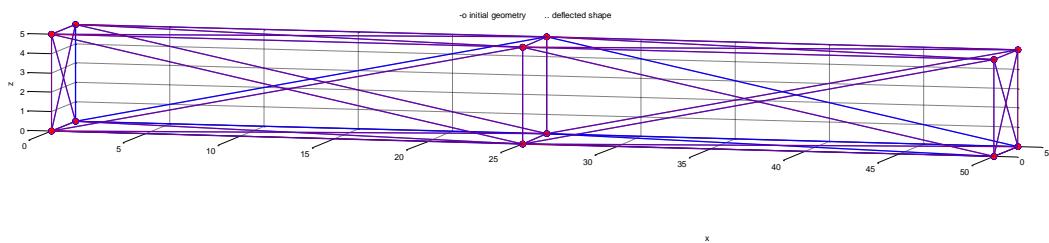


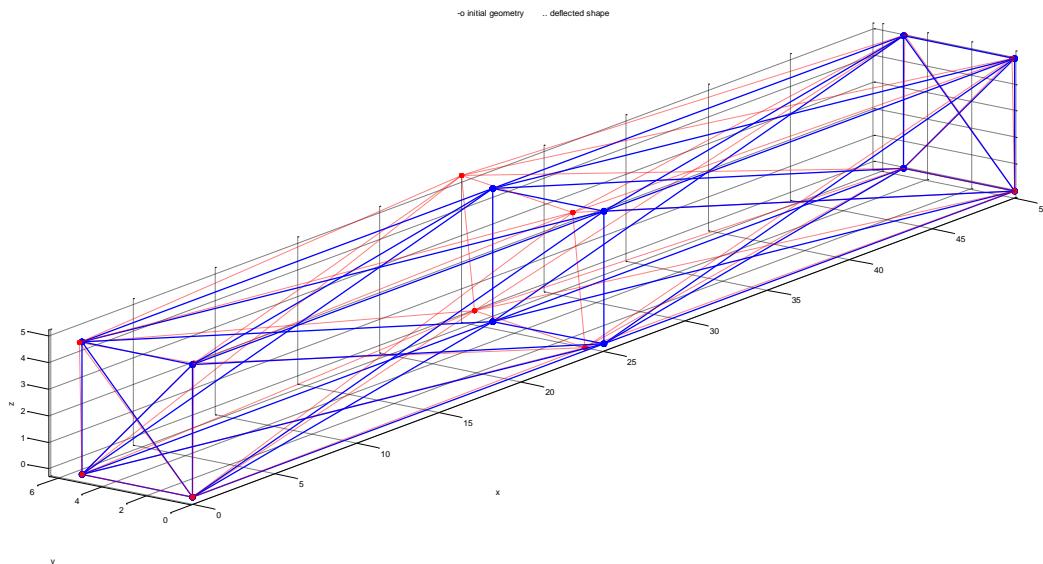
Figure 7

We can further explore the response of the truss by creating a 3D version. The truss is shown below (Figure 8).



**Figure 8**

The most important rule of thumb in designing a truss (and what can quickly become complicated) is the issue of stability. The previous 2D static truss was simple – one face, two supports, a handful of elements. With typical geometric bracing of the elements, the stiffness matrix is invertible, as there is a non-trivial solution to the equation of motion. However, in the case of 3D, the process of designing a truss to provide an invertible stiffness matrix (and thus solve for the displacements) is a bit more cumbersome. It is in the *connection* of the two faces where the most trouble lies. In order to ensure that the stiffness matrix is invertible, adequate lateral bracing (both upper and lower) had to be included. The result is the above 3D geometry of a simple Kingpost truss bridge. Applying loads in the positive y-direction against the face at datum,

**Figure 9**

We can note the displaced geometry of the structure. Perhaps the most efficient way to determine if a truss is unstable is to calculate the eigenvalues of the system. In order to have a non-trivial solution to the equation of motion (in this case, just equilibrium of the applied loads and internal resistance), the eigenvalues must not be zero. Thus, a simple check on the eigenvalue problem (EVP) will indeed disclose any stability issues with the system. However in the static code, there is no acceleration that can be attributed to the mass of the system, and thus there is no EVP to do. A check on the stability of the system for a static rendering is thus only attainable by the singularity of the stiffness matrix.

### *The Dynamics*

The static truss program is now prepped to become dynamic. By implementing Newton's second law,  $F = ma$ , the truss will behave with a dynamic response, and from there we can apply damping, resonance, and sinusoidal forcing functions, among other things. It is the essence in the application of Newton's Second that we are able to make what was once static, now a function of time. From the previous homeworks and now here again as well, the EOM is as follows,

$$Ma + Cv + g = F$$

Where  $M$  is the mass matrix (element mass)

$a$  is the acceleration vector that contains the acceleration in each dimensional direction of the free nodes

$C$  is the damping matrix (see Homework 8)

$v$  is the velocity vector similar to  $a$ , and

$F$  is the sinusoidal forcing function with respect to time.

Similar to how dynamics was implemented in the ‘ndof’ system (Homework 7-10), the residual and gradient (or tangent stiffness matrix with the EOM) are implemented within the Newton while loop. An approximation for the acceleration is determined using Newton’s Method, and then the displacement and

velocity are approximated by using the time-stepping algorithm Newmark's Method (see Homework). The residual and gradient for Newton's Method are shown below.

$$g = Ma + Cv + g - F$$

$$A = M + C\zeta + A\eta$$

And thus the approximated acceleration is,

$$a = a - A \setminus g$$

In order to first make the static code dynamic, the mass matrix had to be created, which is essentially the mass of every element. From this, the EVP was able to be computed and the eigenvalues were checked for stability. The stiffness matrix used in the calculation of the EVP was the stiffness matrix with initial global displacement vector, or the “initial condition” of the system. The use of the eigenvalues was not just for stability, but also to explore the effects of resonance on the structure. The global displacement vector was passed back one more time into assemble to retrieve the residual vector, so as to calculate the initial acceleration, or  $a_{old}$ .

$$a_{old} = M \setminus (F - (g + Cv))$$

From there, the Newmark parameters are computed at every time step and the Newton loop is iterated, and the time-stepping algorithm repeats over the prescribed set time. The most important step for visualizing the magnitude of the stresses within the elements is to update the global displacement vector back into the stress script. This small but essential step in the dynamic code is needed to generate the stress color range for plotting.

### *The Geometry of the Camelback Truss Bridge*

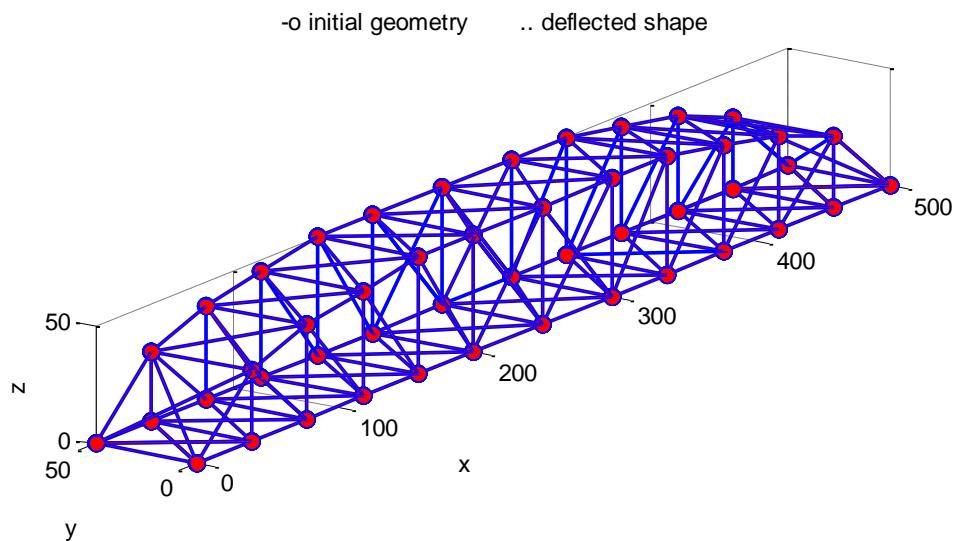
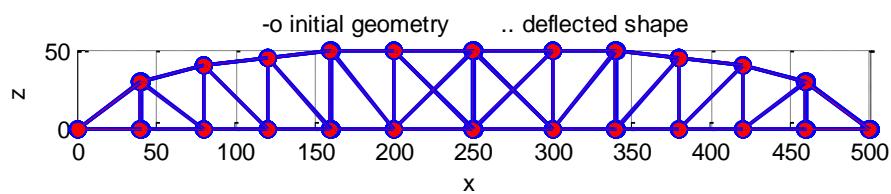
Using AutoCAD, a preliminary truss design was drawn so as to get a sense of the sizing of the truss bridge, and was also used as an excellent form of visual aid to prevent instability of the truss early on in the design process. Excel, in conjunction with AutoCAD, was used to “map-out” the coordinates of the nodes and element connectivity. This was extremely useful in coordinating which elements belonged to which subgroup of material properties. The boundary conditions and nodal forces were also incorporated systematically in Excel. The truss bridge Excel file was then called into Matlab using the ‘.xls Read’ function. Below is Table 1, which contains geometrical information on the Camelback truss bridge.

**Table 1**

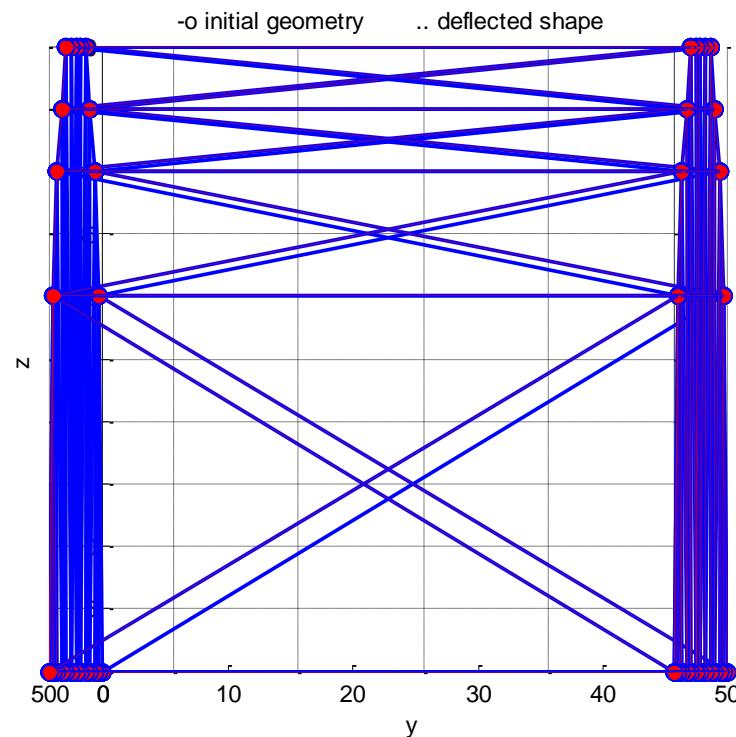
Geometry	
Total Length	500
Total Width	50
Max. Height	50
Connectivity	
Number of Nodes	48
Number of Elements	176
Number of Fixed Nodes	4
Number of Free Degrees of Freedom	132
Material Sets	
Top Chord	1
Bottom Chord	2
Web Members	3
Struts	4
Upper Lateral Bracing	5
Floor Beams	6
Lower Lateral Bracing	7

The labeling of the material sets with the type of truss member is ideal in that there is direct control over which segments of the truss can be manipulated. We can also use this to design a more authentic truss bridge. The chosen length, width and height of the bridge were chosen arbitrarily, but were decided upon in accordance to proper sizing of a typical truss bridge. The fixed nodes were chosen to be placed at the termination ends of the bridge to simulate a fixed support or abutment. The number of free degrees of freedom (jpos) is thus 132, which represents the variability of 44 free nodes moving about in 3 dimensions.

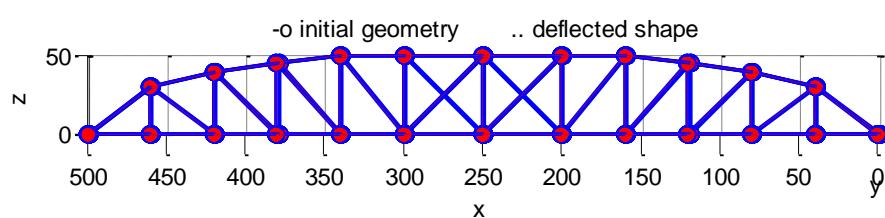
Different vantage points of the Camelback truss bridge are shown below in Figures 1- 16. Note that the bridge is currently in the static condition.

**Horizon View (Figure 10)****Profile View (Datum, Figure 11)**

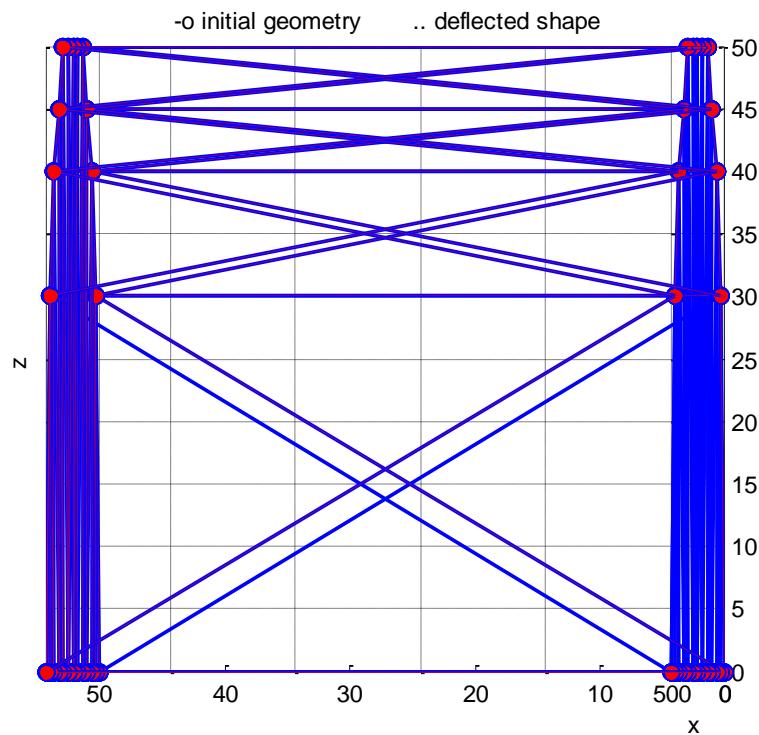
Cross Section View (Front, Figure 12)



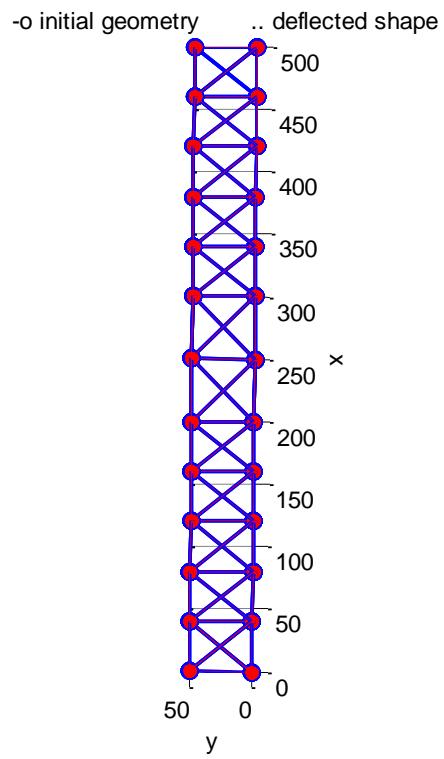
Profile View (Adjacent, Figure 13)



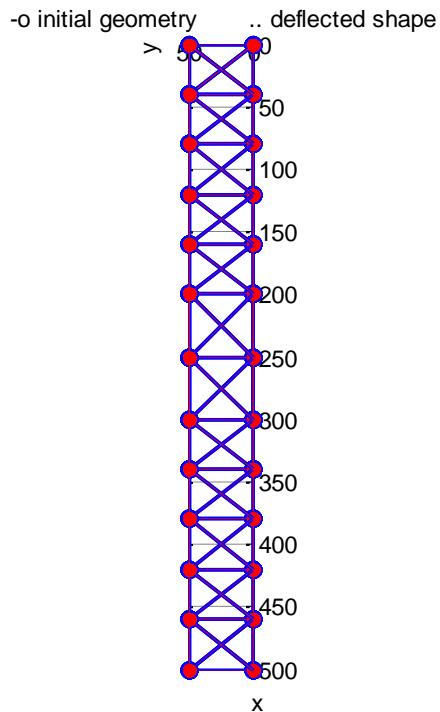
Cross Section View (Rear, Figure 14)



Plan View (Top, Figure 15)



Plan View (Bottom, Figure 16)



From these figures, it is obvious that the generation of the nodal coordinates and element connectivity produced a symmetric and appropriately sized truss bridge geometry. An interesting note to make here is that bracing along the faces of the bridge were not needed at every section, but rather solely at the two center sections. The truss became unstable when there was no lower lateral bracing nearest the ‘supports’.

### *The Motion*

To start, a sinusoidal forcing function will be applied to the truss bridge. The sinusoidal forcing function is of the form,

$$F = F_o \sin(F_{wo} t)$$

Where  $F_o$  is a constant force,

$F_{wo}$  is the forcing frequency, and

$t$  is the time.

By placing a forcing function on the center nodes at length 200, 250 and 300, the response of the bridge can be thought of as gusts of wind attacking the datum side. Note that a forcing function value of ‘1’ was placed in the y-direction in the excel file, and then amplified in the code. Note that the motion is color coded to represent stress levels within the elements. These colors are rendered using the “jet” color pack in Matlab.

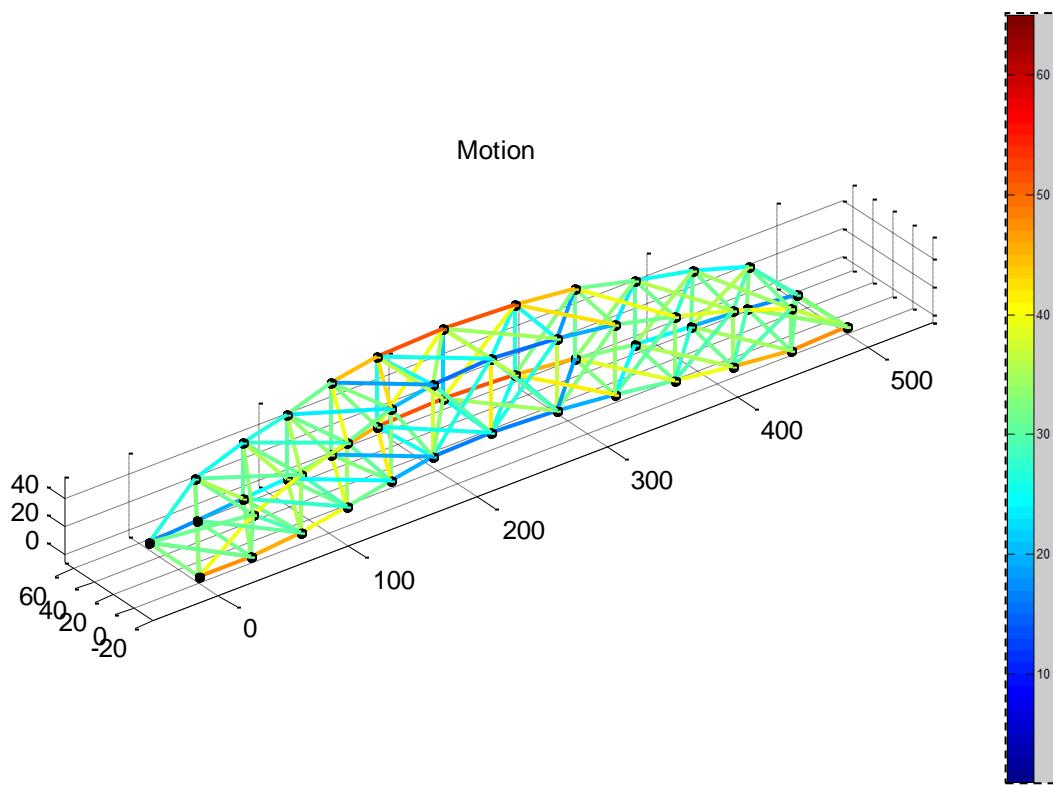


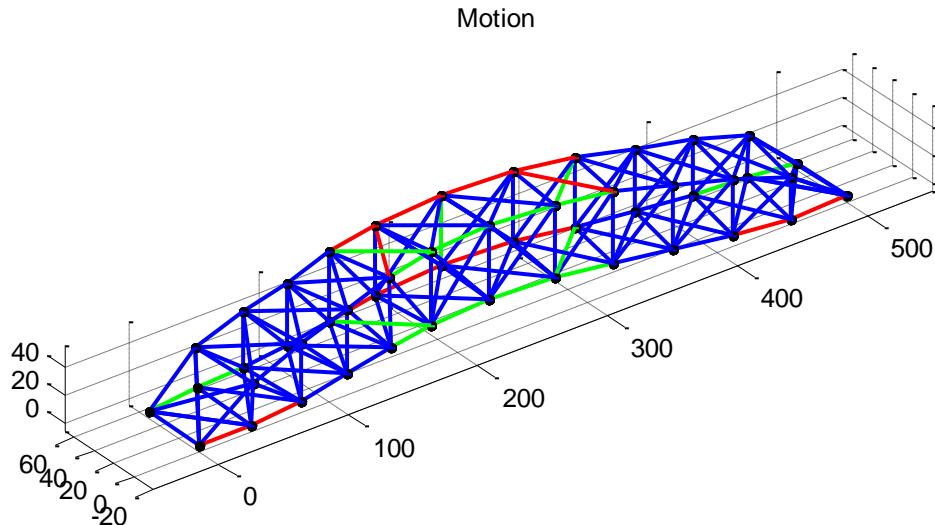
Figure 17

With the applied force in the y-direction, we can note that the elements that undergo the most stress in tension (red) are the two horizontal members of the top and bottom chord on the adjacent face. This makes perfect and intuitive sense, since the force is applied on the nodes directly opposite the members on the datum side. The elements being most stressed in compression (dark blue), are the elements exactly opposite the red. We note here the importance of symmetry of the truss by comparing with the color scheme. It can be seen that the stress colors are symmetric among all elements. This is a good indication that the truss design is symmetric and the program is providing an adequate rendering of the internal forces in the elements.

It is also interesting to note how the stresses develop throughout the framework. The least stressed elements are the upper and lower lateral bracing, as well as some of web members. We also must note that there is a high level stress on the bottom chords near the support. This is proof that the applied forces do in fact travel throughout the entirety of the structure, even at the supports. The reactions at the supports are of course needed to eventually provide equilibrium in the truss.

Another analysis of truss design (besides stress coloring in elements) is whether a member is in tension or compression. It can be noted here using the stress color scheme, but it is more effective to highlight only the two states of tension and compression to better visualize the dynamics of the truss.

Below is Figure 18, which details the compressive/tensile state of each element. Note that for a member to be in tension/compression, the elemental stress must be within a particular range that is a function of the yield stress. Blue represents an element that does not fall within either range, red in tension and green in compression.



**Figure 18**

As expected, the adjacent top and bottom chords near the center are in tension, where the opposite members are in compression. Note that this does not mean the forces exhibited here are equal and opposite, but rather upon careful examination of the motion does the adjacent face reach a tension zone faster than the datum face reaches compression. We can again note here that the bottom chord members nearest the supports are heavily stressed.

Placing nodal forces along the top center nodes, we can exhibit what happens to the bridge under external loading, which could be due to the possibility of newly installed utilities (such as large light fixtures), pedestrians (from walkways) or even cell phone antennae. The response is shown in Figures 19 and 20, both in the stress state and compression/tension.

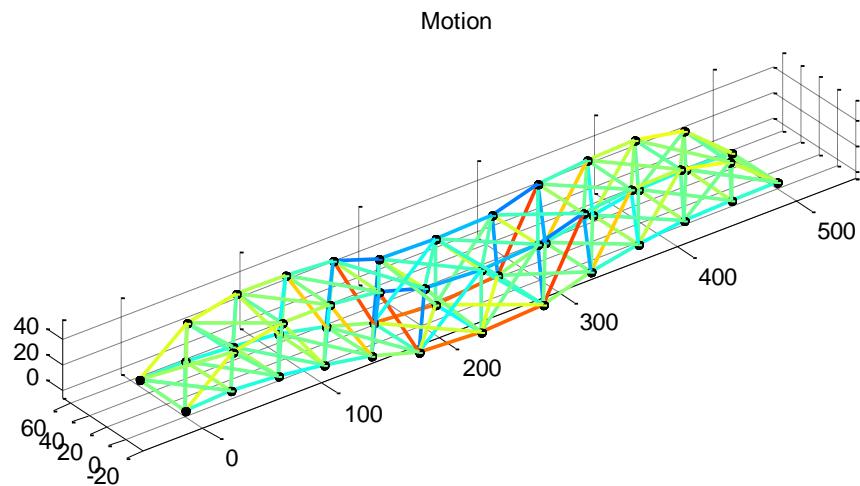


Figure 19

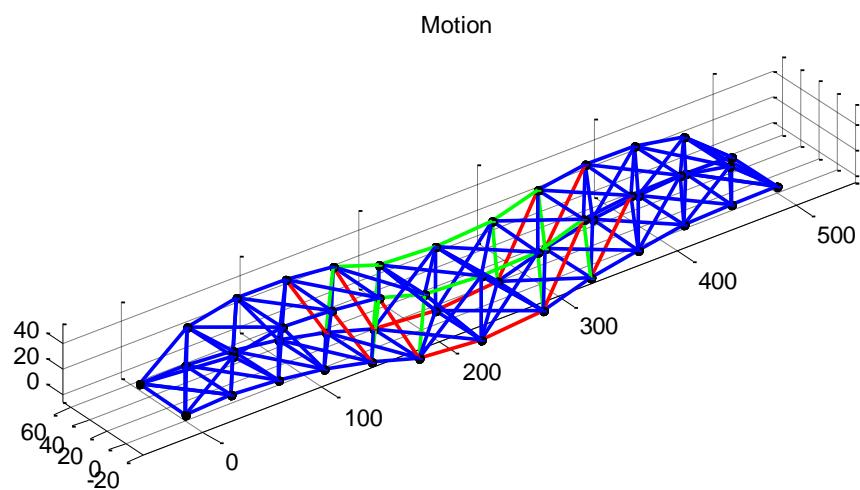


Figure 20

The response is what is expected – the upper chords are the most stressed in compression, and the lower chords in tension. The response is obviously exaggerated, but for academic purposes this is a reasonable result. The last experiment with a pure sinusoidal forcing function is to employ a dual-wind force that attacks both faces of the truss. The result is plotted below (Figures 21 and 22).

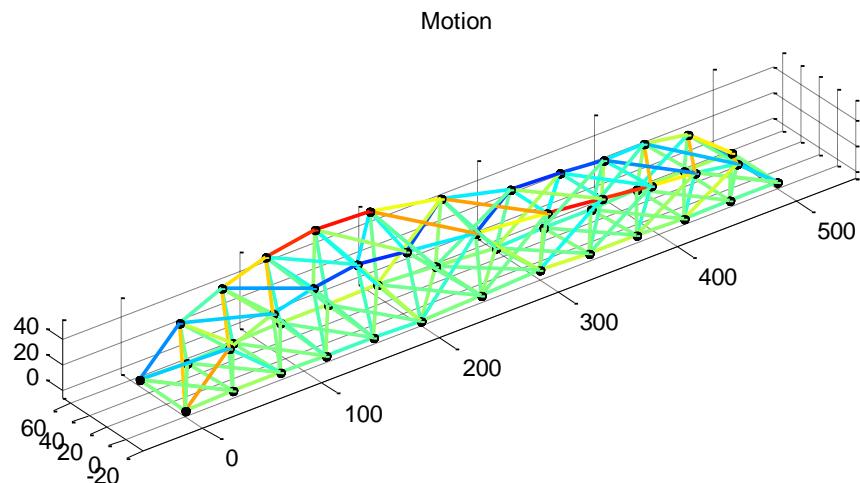


Figure 21

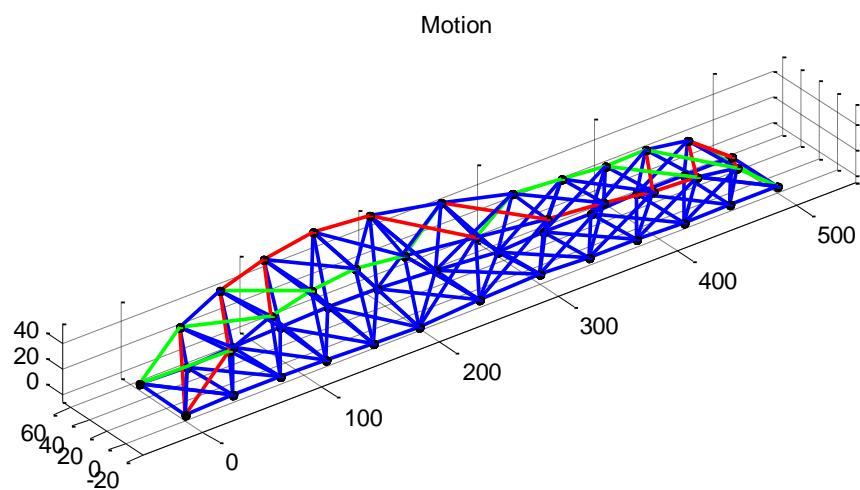


Figure 22

The response is quite interesting! Because of the counteractive forces on each face, the bridge twists to accommodate for the applied forces. If supports were provided at the center of the bridge (fixed nodes on the bottom), the response would be similar to the one seen in Figures 17 and 18. But because the bridge is free to move throughout the entirety of the span, the bridge twists and consequently, severe stresses are built-up throughout almost the entirety of the top chords.

We can explore the effects of damping by forcing the structure and applying a relatively small amount of damping to the entire structure. The below structure (Movie 1) is forced with the duel-wind applied force scenario as mentioned previously. The Figures (23 and 24) below illustrate the deformation of the truss after repeated cycles of the forcing function.



DampingDuelWind.mp4

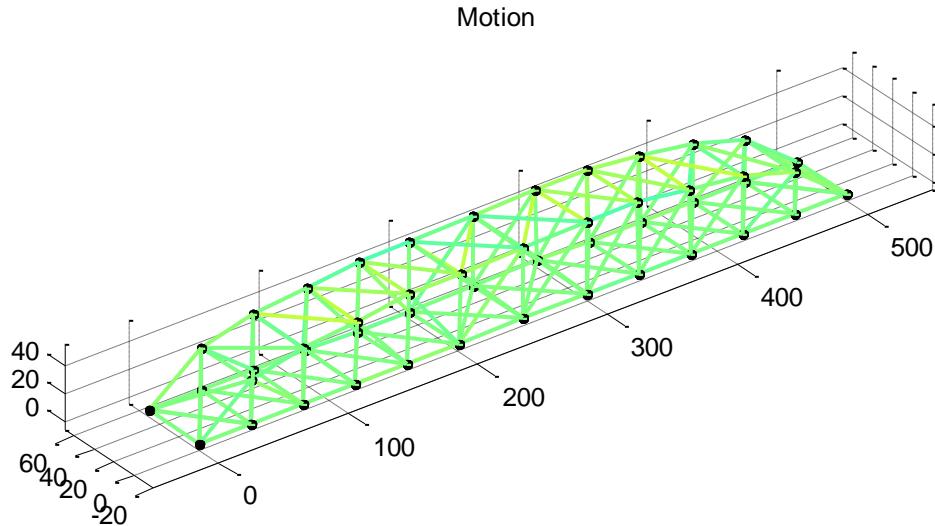


Figure 23

We can note immediately that the bridge does not reflect the severe twisting that the previous bridge had without damping. Viewing the entire movie, we can note that the response eventually settles back to equilibrium. Since damping is essentially supposed to “damp” out the motion of the structure, this is a good result.

Perhaps one of the most interesting phenomenon of dynamic analysis is the mode shape. Since there are 132 free degrees of freedom for this bridge, there are consequently 132 different mode shapes. With the ease of Matlab, we can compute these mode shapes (albeit a few as there are so many) and study the response. Mode shapes were chosen arbitrarily and documented in the figures (and movies!) below. Note that the purpose of studying the mode shapes is to visualize how the bridge could displace under free vibration. Three of the relatively higher modes (above 50) were chosen to be documented as the natural frequency for these shapes are higher and thus easier to visualize.



Mode100.mp4



Mode50.mp4



Mode60.mp4

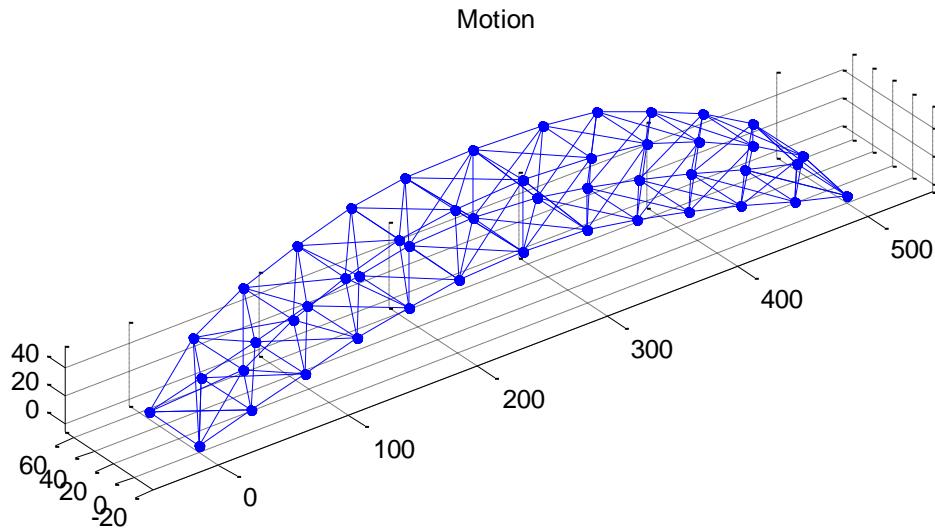


Figure 24

The above figure illustrates the bridge displaced at the 50<sup>th</sup> mode shape. The bridge arches itself at its most upright position, and then bends down into a concave position, and repeats. It is essentially acting like a 1D simply supported beam, bending up and down. It's quite amazing to see something as complex as the above bridge, vibrating in such a way that resembles the most simplistic of systems.

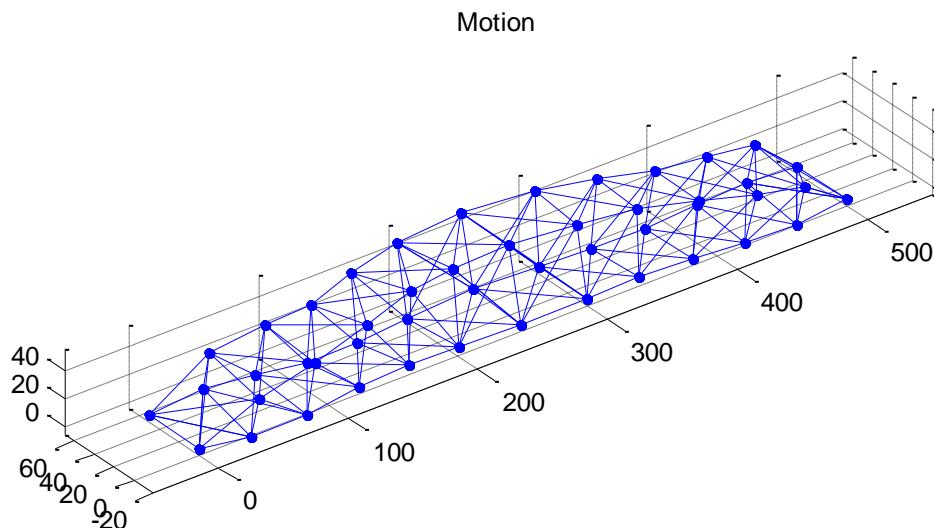


Figure 25

The 60<sup>th</sup> mode shape is quite interesting! This mode shape excites the bridge to vibrate like a typical simply supported beam in its own third mode shape – a sinusoidal wave completing 1.5 periods. Again, this mode shape resembles the very basic of systems.

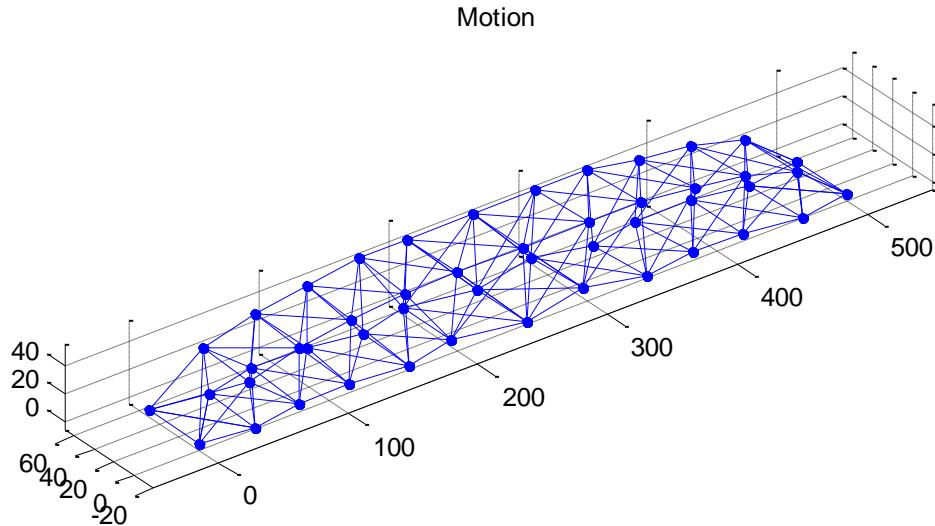
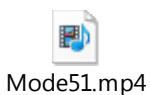


Figure 26

The 100<sup>th</sup> mode shape is not all too interesting as the first two, but does prove that higher modes frequent faster than lower ones. In this mode, every bottom section (lower lateral bracing) rotates back and forth across the x-y plane, and generates some very weird response among the rest of the members.

Resonance can also be observed in every mode shape by forcing the structure to vibrate under its own natural frequency with respect to the corresponding mode shape. To start, we can observe the motion of the 51<sup>st</sup> mode shape under resonance. The 51<sup>st</sup> mode excites the bridge to vibrate by swaying laterally across the x-y plane (Figure 27). The movie of the bridge under free vibration is shown below, as well as under resonance.



Mode51.mp4



Res51.mp4

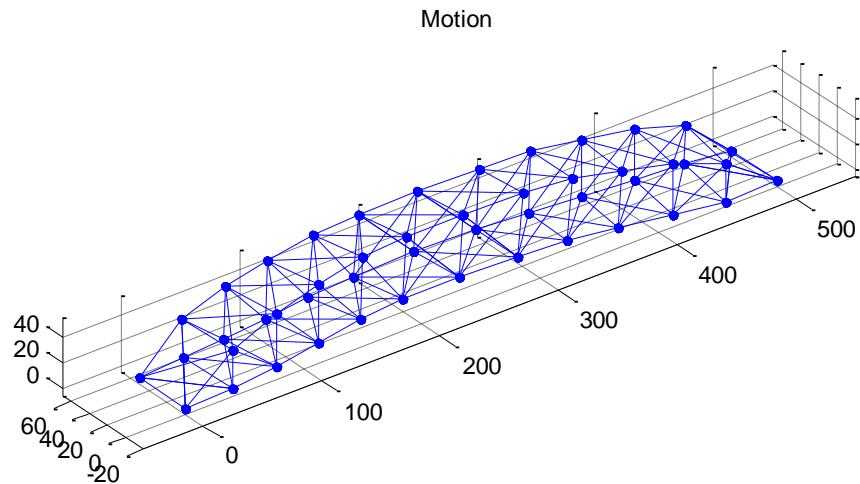


Figure 27

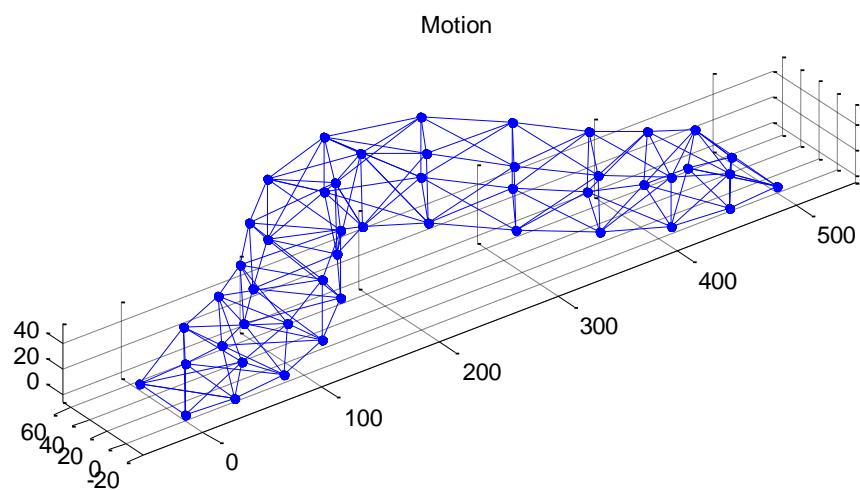


Figure 28

As time progresses, the structure begins to resonate within the mode shape, and eventually goes beyond realistic measures. Resonance is obviously a catastrophic phenomenon, and can be avoided by damping structures to withstand the event, which is most commonly caused by earthquakes. Earthquakes were not observed in the truss analysis code, but the effects would have been interesting to observe and document.

## The Ritz Vectors (Homework 12)

### Summary

Ritz vectors were calculated to estimate the eigenvectors of a static ‘ndof’ system and compared with the associated eigenvectors for accuracy. Subspace iterations were also performed on the Ritz vectors to better approximate the eigenvectors.

### The Ritz Vector

The Ritz vector modal analysis is based off an initial loading displacement of the system, such as ground displacement, and thus is the preferred method in determining the eigenvectors of system with a prescribed initial horizontal displacement. However, in order to understand the effectiveness of the Ritz vectors, we must first start with a static ‘ndof’ system. The first Ritz vector of the system can be determined by accounting for the initial load displacement, as detailed in the steps below.

The static equation from the EOM for the complete dynamic response is as such,

$$M\ddot{u} + C\dot{u} + ku = -M1\ddot{u}_g$$

The Ritz vectors appear in the EOM by substituting the time-dependent displacement vectors for the Ritz modal analysis,

$$u(t) = \sum_{i=1}^M \Psi_i a_i(t)$$

Where  $M$  (in the operator) is the number of Ritz vectors.

We can then pre-multiply by the transpose of the Ritz vector  $\Psi^T$  to the entire EOM. The result is thus,

$$\Psi^T M \Psi \ddot{u} + \Psi^T C \Psi \dot{u} + \Psi^T k \Psi u = \Psi^T (-M1 \ddot{u}_g)$$

We can solve for the first Ritz vector (just dependent on the load) by noting that neither the sign nor the ground acceleration matter, but solely just the shape of the Ritz vector.

$$\Psi_1^T k \Psi_1 = \Psi_1^T (M1)$$

Eliminating the transpose of the Ritz Vector and taking the inverse of the stiffness matrix,

$$\Psi_1 = K^{-1} M1$$

We can then generate the general algorithm for a static system for the Ritz vectors as,

$$K\Psi_{n+1} = M\Psi_n$$

We must note here however, that the Ritz vectors very soon appear to look very similar to the ones before it, as such that,

$$\Psi_n \cong \Psi_{n-1}$$

This is obviously not what we want from the modal analysis, as this result essentially dictates that there are repeating eigenvectors within the system. To account for this, we must use Gram-Schmidt (GS), which is a method that orthonormalizes (orthogonalizes and normalizes) linearly independent vectors

within a subspace. It is in this method of orthonormalising that we avoid the scenario of repeated eigenvectors.

### *Gram-Schmidt*

Essentially, the idea is to link what went wrong between the first and second Ritz vectors, and solve the equation successfully that links the two. For instance, we can find the second vector as,

$$\Psi_2 = K^{-1}M\Psi_1$$

And noting that there is some coefficient between the first two vectors that creates a difference between the two, we can improve the second Ritz by,

$$\widetilde{\Psi}_2 = \Psi_2 - \alpha\Psi_1$$

Where, by setting the improved second Ritz vector orthogonal to the first, we can determine the coefficient  $\alpha$  to be,

$$\alpha = \frac{\Psi_1^T M \Psi_2}{\Psi_1^T M \Psi_1}$$

And now normalizing so that,

$$\widetilde{\Psi}_2^T M \widetilde{\Psi}_2 = 1$$

$$\widetilde{\Psi}_2 = \frac{\widetilde{\Psi}_2}{\sqrt{\widetilde{\Psi}_2^T M \widetilde{\Psi}_2}}$$

We can generalize this procedure into an algorithm, as such,

$$\widetilde{\Psi}_i = \Psi_i - \sum_{j=1}^{i-1} \frac{\widetilde{\Psi}_j^T M \Psi_i}{\widetilde{\Psi}_j^T M \widetilde{\Psi}_j} \widetilde{\Psi}_j$$

Where  $i$  is iterated from 1 to the number of Ritz vectors taken, and

$j$  is iterated from 1 to  $i - 1$ .

These vectors are good approximations of the lower eigenvectors (the first few), but at higher eigenvectors (later ones), the respected Ritz vectors become less accurate. It is because of this that we need to perform subspace iterations on the Ritz vectors.

### *Subspace Iteration*

In order to refine the Ritz vectors, which is essentially reducing the mass and stiffness matrix in order to determine reduced eigenvectors, and use those as the basis of the refined Ritz vectors. The refined Ritz vectors are then normalized within the subspace iteration. To reduce the mass and stiffness matrix, the improved Ritz vectors from GS are used to garner the Ritz vector within the subspace iteration, as shown below

$$Q = K^{-1}M\Psi$$

The reduced mass and stiffness matrix are thus,

$$M_r = Q^T M Q$$

$$K_r = Q^T M Q$$

The eigenvalue problem is then solved using the reduced matrices, and then the reduced eigenvectors are multiplied with  $Q$  to obtain the refined Ritz vector  $\Psi_r$ .

### Ritz and Eigenvector Comparison

Implementing the Ritz code into the ‘ndof’ code, we can run an analysis on the approximation of the eigenvectors using the Ritz vectors for the undamped, free vibrational mass and stiffness matrix. With 25 degrees of freedom and an equal number of Ritz vectors, Figure 1 below shows the first eigenvector and associated Ritz equivalent. The plot in Figure 2 is of the once refined Ritz vector.

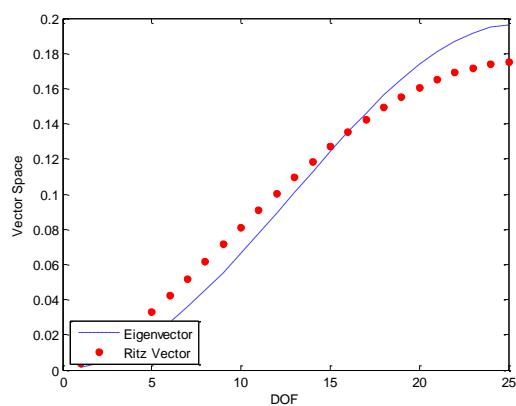


Figure 1

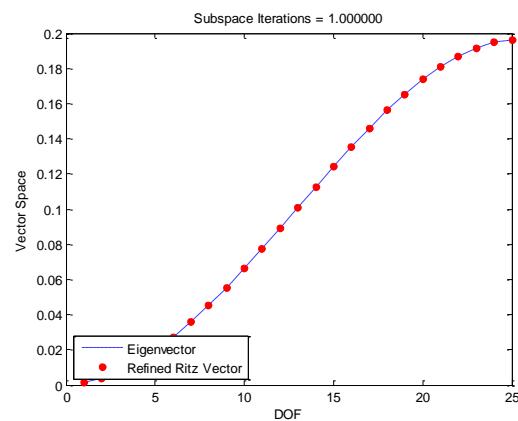


Figure 2

The unrefined Ritz vector does not match up well with the first eigenvectors of the system. With one refinement, the approximation is exact. We can continue to pull higher order Ritz vectors and plot them against the respected eigenvector. Figures 3 and 4 are of the second vector, with one iteration.

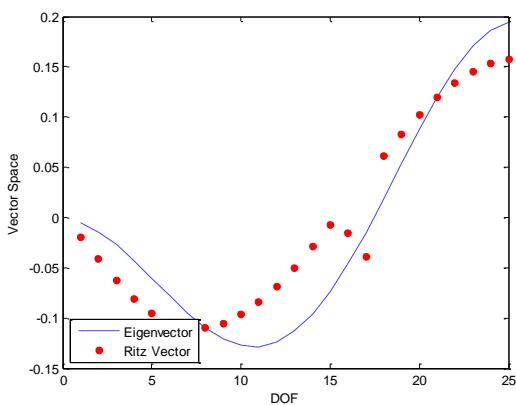


Figure 3

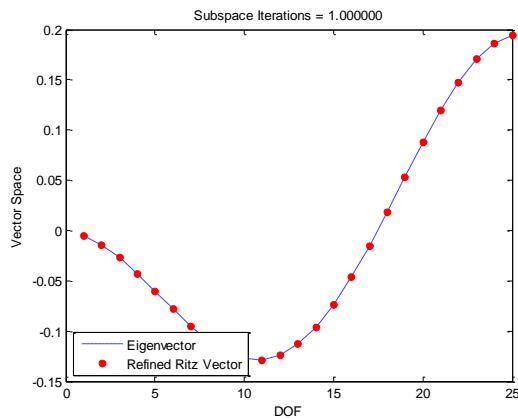


Figure 4

If we look into the higher order Ritz vectors, we note that the initial approximation is nowhere near the respected eigenvector. Increasing the amount of refinements gives a better approximation, but the accuracy is still something to be desired. Figures 5 and 6 illustrate this result at the 20<sup>th</sup> vector.

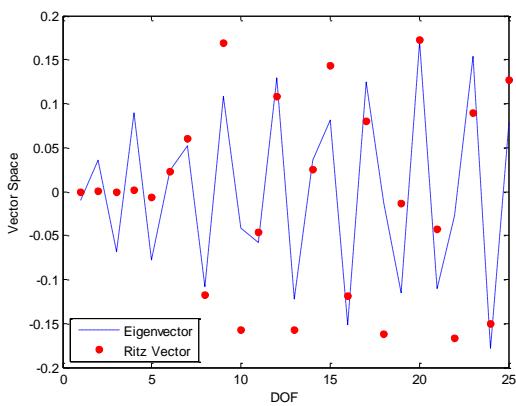


Figure 5

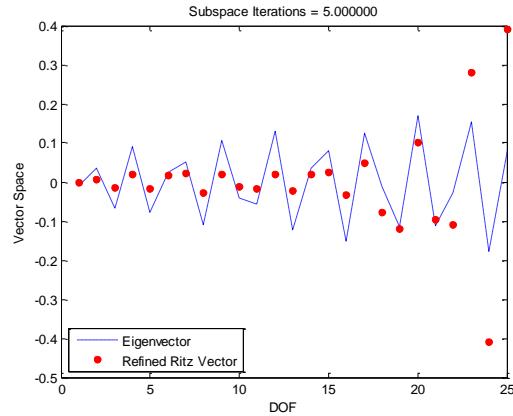


Figure 6

If we increase the iterations beyond 5, Matlab displays a warning that imaginary numbers are being generated within the subspace iteration algorithm. It seems that this result is consistent with higher order Ritz vectors. Though the approximation of the lower order eigenvectors is accurate, the higher order eigenvectors are difficult to impersonate using the Ritz vectors.

## Axial Wave Propagation (Homework 15)

### Summary

In this exploratory homework assignment, axial wave propagation in a bar was observed. The boundary conditions of the bar (fixed and free ends) were also varied to explore the effects this has on the propagation of the wave. The effect of resonance was also observed, as well as damping.

### The Discretization Requirements

In order to properly observe axial wave propagation in a bar (such as the axial bars in trusses), we must first note that a proper number of variability, or degrees-of-freedom, are necessary to evoke a proper response. Below are two plots of the axial bar, both fixed at each end, with varying amount of degrees-of-freedom (number of elements that compose the entirety of the bar). Note that the response of this system was generated by simply prescribing an initial displacement.

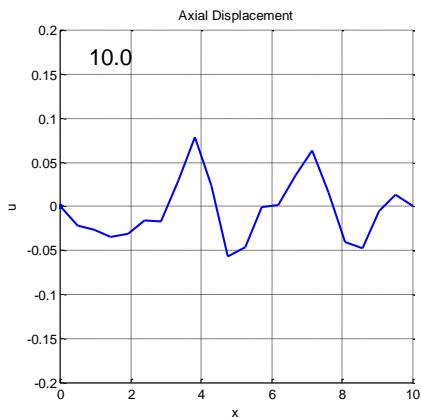


Figure 1

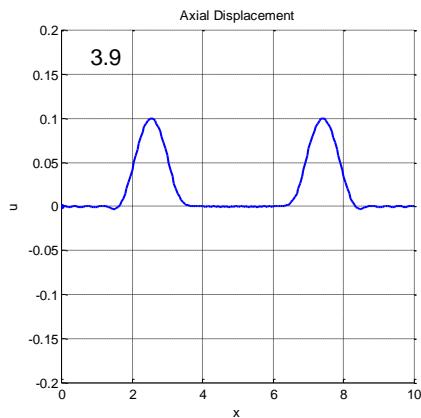


Figure 2

We can immediately note the inconsistency of the wave propagation in the bar. Figure 1 was generated using 20 elements, where Figure 2 was generated with ten times as much (200 elements). For an accurate depiction of the wave, it must look sinusoidal. Perhaps it is better to note the displacement-time graphs of the individual elements for accuracy and symmetry in the system. Below are Figures 3 and 4 of the two systems with different number of elements, each of which depict the displacement of the elements over time.

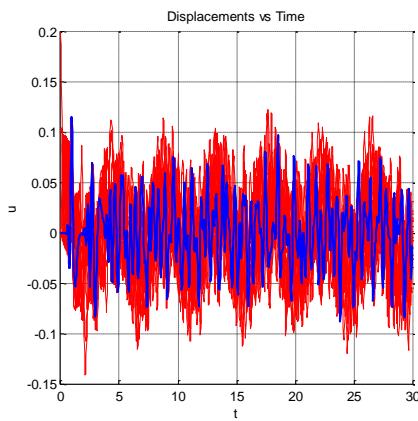


Figure 3

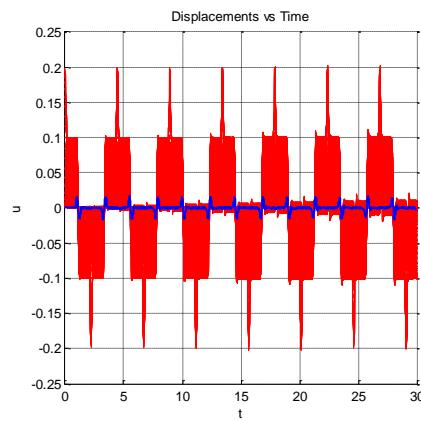


Figure 4

The contrast is stark! As seen in Figure 3, the displacement of the individual elements is not consistent with the others. This is obviously not the case in an actual axial bar! Figure 4 depicts the response that is expected – the elements displace symmetrically, rather resembling more of a whole system than a system made of multiple, individual elements of individual displacements.

We can increase the number of discretization requirements (number of elements) past 200, but the result appears identical. The below two plots are of the displacement of each element per time. Figure 5 is with 250 elements and Figure 6 with 300.

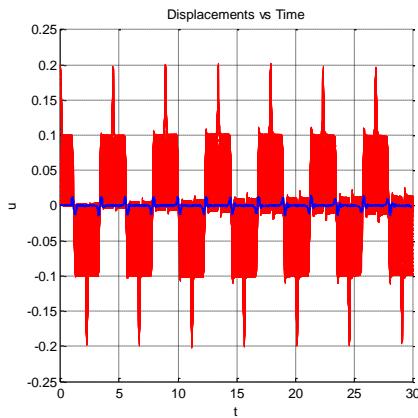


Figure 5

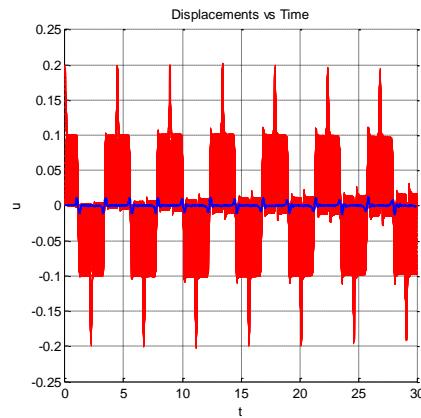


Figure 6

The result is identical, and thus for the remainder of this discussion, we will use 200 elements as more than this significantly decreases the computation time.

### *The Boundary Conditions*

We can observe how the propagation of waves throughout an axial bar vary with changes in the boundary conditions. Fixing both sides of the axial bar (no displacement on either end), the response is shown below Figure 7. Figure 8 is of the far right end free. Note that the wave response is due to an initial displacement.

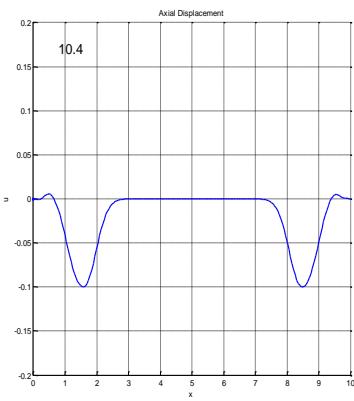


Figure 7

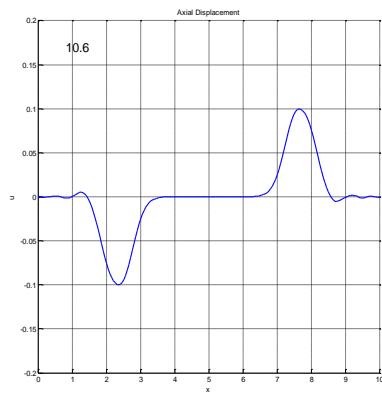


Figure 8

At approximately 10.5 seconds into the response, we can note that the fixed end of Figure 7 (right side) flipped the sign of the wave, as opposed to the free end of the axial bar in Figure 8. We can deduce that a fixed end condition (does not displace) flips the sign of the wave as it comes and goes. This makes sense, as the wave must travel back the opposite direction since it has to pass through the support, and for this bar to be in equilibrium, the returning wave must be equal and opposite. For the free end, the wave doesn't have to return back oppositely, since there is no place for the wave to go to illicit such behavior. Another important distinction to make here is what this means when the two waves clash. When they are opposite, the wave, for a split second, has zero amplitude, and then both regain their original amplitudes and continue on. This makes perfect sense, since the two waves, if both equal and opposite, must reach a state of zero amplitude if they are ever going to cross paths. The exact is true if they are the same sign and clash – they add to create twice the amplitude of either, and then continue on with their original amplitudes.

### Mode Shapes and Resonance

We can invoke a pure mode response by displacing the axial bar exactly proportional to the eigenvectors of the system. For the axial bar fixed at both ends, we can observe the first mode of the bar (note that there are 200 modes for this system, but we only need to analyze the first two to understand them all).

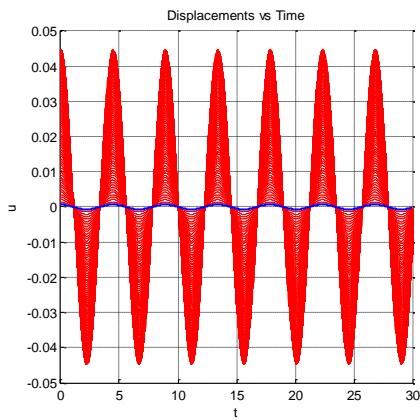


Figure 9

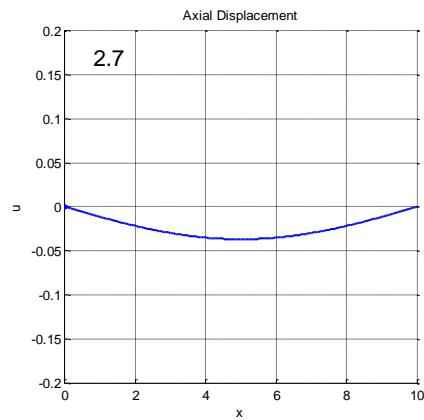


Figure 10

The shape of the mode can be easily seen in Figure 10, and we can note that the mode persists as shown in Figure 9. If we displace the bar proportional to the second mode, we can notice the following response.

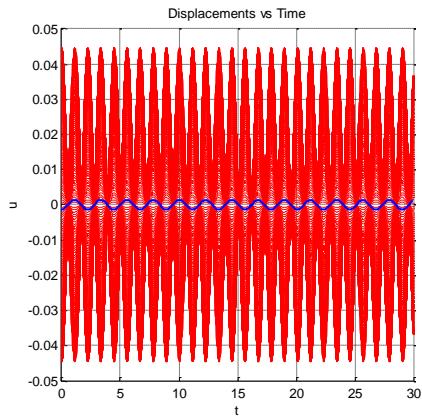


Figure 11

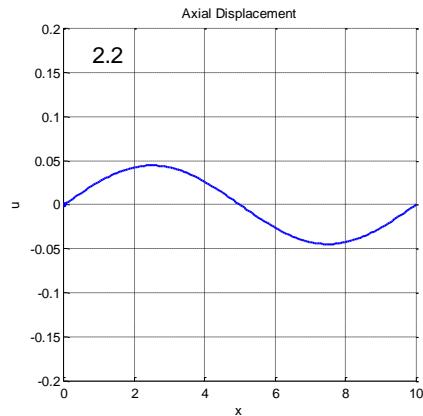


Figure 12

As expected, the frequency of the elements increased, as this system is operating at a higher mode than the previous. Note the wave propagation in the second mode is now a complete sinusoid, as opposed to only half a sinusoid in Figure 10. This pattern is evident in the higher modes. Take mode 50, for example.

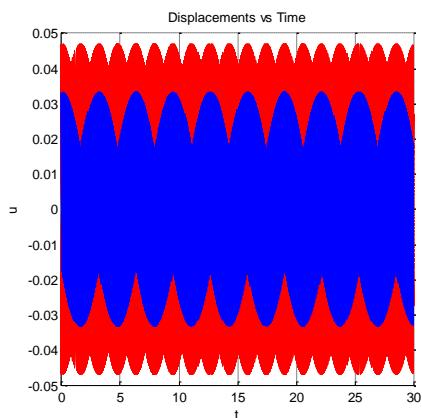


Figure 13

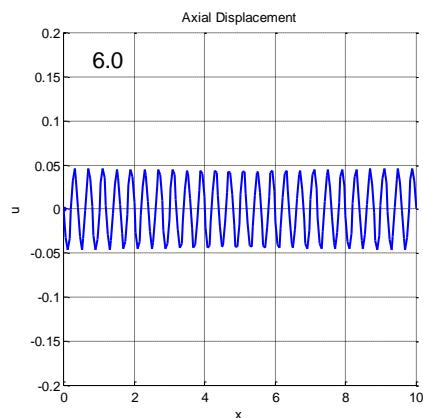


Figure 14

If we count the peak-to-peak (period) we can note the number is 25, half the mode shape that the system is under, as expected. Also, we can again note that the mode persists over time (Figure 13).

We can also observe resonance in this system. By forcing the axial bar with a driving frequency equal to that of an eigenvalue, the waves in the axial bar will propagate in a response of the respected mode shape, with the amplitude of the waves increasing over time. Note resonance at the first mode, below

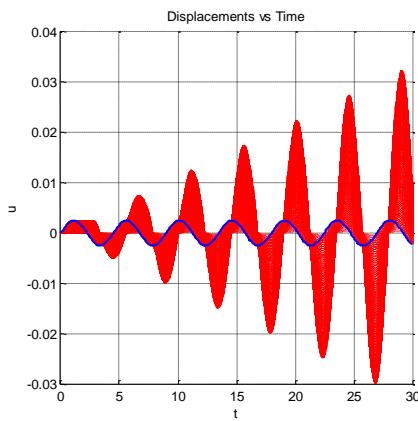


Figure 15

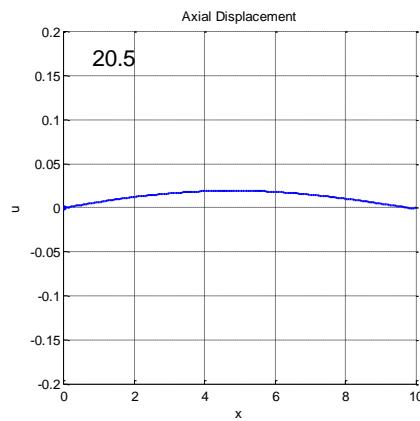


Figure 16

From Figure 15, we can note that the amplitude of the waves increases over time. This is obviously a dangerous phenomenon, since the magnitude of the waves within the axial bar are increasing. The shape (Figure 16), as expected, is of the first mode. Operating the system at higher modes, we can note that resonance still occurs within the axial bar, but since the natural frequency of the bar is higher, the self-dampening rate of the axial bar increases (see Homework 7 for a more detailed description of this phenomenon), and thus the amplitudes of the propagating waves decreases. Below is the system operating at the 50<sup>th</sup> mode with resonance.

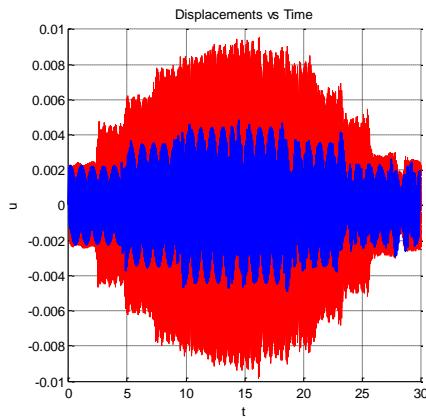


Figure 17

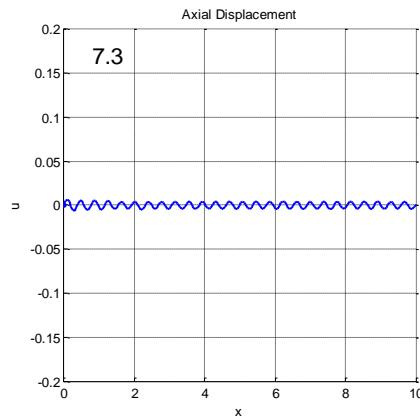


Figure 18

Note that in Figure 17, we see the decrease in amplitude. This is not the end result however, as the waves in the bar will again take the higher amplitudes as resonance goes on indefinitely. In the Figure 18, we can expect the amplitude of the waves to be approximately 0.009 at the 15 time mark, and then decrease after that.

If we apply a sinusoidal forcing function to the axial bar at a driving frequency other than the natural frequencies of the system, we obtain the response shown below. Note that the forcing function is applied at the right end.

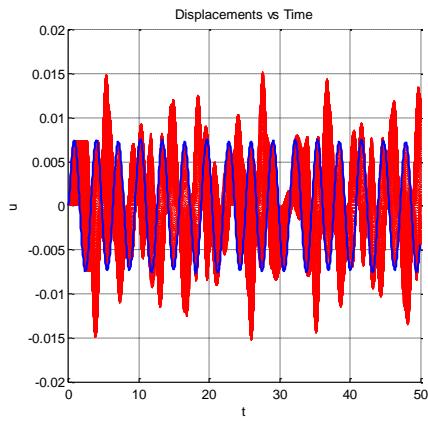


Figure 19

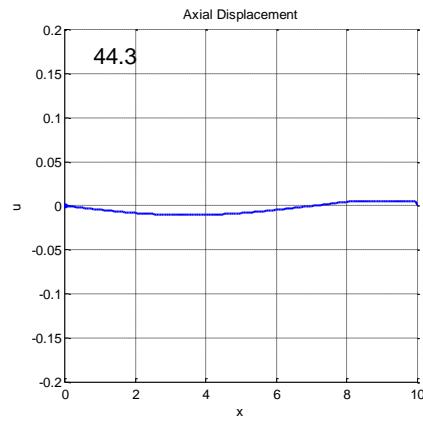


Figure 20

The amazing result of applying a constant sinusoidal forcing function is that one can visualize the waves coming in contact with the boundary conditions and behaving accordingly! For instance, the wave travels to the free ends and then bounces back *equal and opposite*. We can visualize this response further with the metaphor of battle ropes (exercise ropes). As one applies an up and down motion on one end of the rope, the waves propagate and reflect back equal and opposite from the fixed-end connection of the rope. If we are strong enough, we can resist the motion coming back, and thus we ourselves are a “fixed end”, contributing to the equal and opposite behavior. If we are not strong enough, we may simply lose grip of the rope, and thus be acting as a free end, with the waves in the rope traveling back the exact way they came (except we can expect some energy loss in our metaphorical system, and thus the wave will not return at the same magnitude it came with initially). We can use our perfect energy conservation system to model what happens when the right end is free.

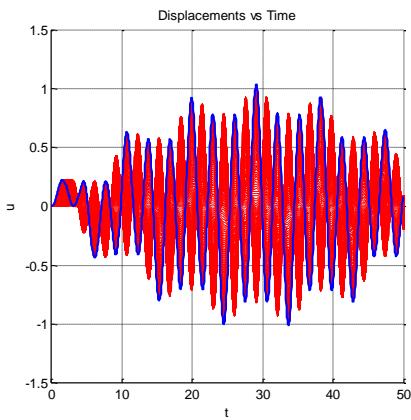


Figure 21

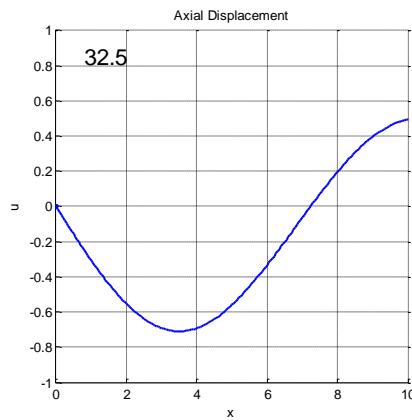


Figure 22

As expected, the model moves wildly since there is no support to counteract the waves propagating in the bar. The waves begin to what appears at first resonate for some time, and then dampen out as time goes on. This is not what is going on here. It is simply just the forcing function increasing and decreasing over time. Except in this system, we can see the effects of that forcing function much more clearly. This system is essentially a “diving board”. As the diver stands at the edge and bends his/her knees, the board feels the effect, and when the diver leaps off, the waves propagate wildly in the board. It is only after

some time after the diver has already jumped, that the board dissipates all the energy into the support, and equilibrium is satisfied. We can think of the energy dissipating in our system too, but it is really just the forcing function decreasing. But the same concepts apply.

## Flexural Wave Propagation (Homework 16)

### Summary

Flexural wave propagation is observed within a simple beam. The boundary conditions of the beam were changed to observe the propagation of stress waves. Resonance was also observed within the beam, along with pure mode propagation. Damping was also applied to the beam to record the effect on flexural wave propagation.

### The Discretization Requirements

In order to fully and accurately understand flexural wave propagation (FWP), enough elements (or degrees of freedom) must be used. Doing this requires that enough approximations be made on the response of FWP. From the previous study on axial wave propagation, the same result on the discretization requirements is found. This is just the nature of the computations. Below are plots of the beam with both ends fixed, with a forcing function applied.

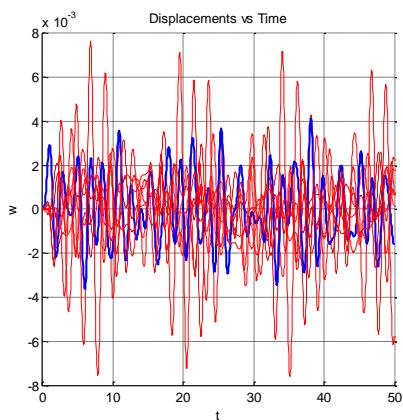


Figure 1

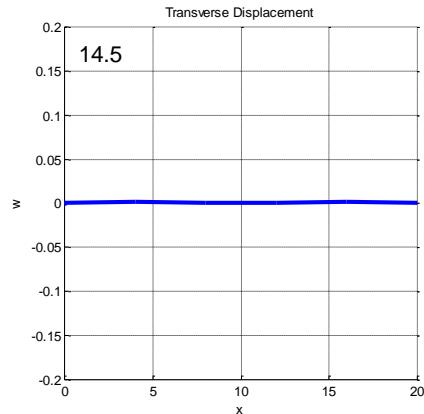


Figure 2

Since both ends are fixed and we are applying a forcing function at the center of the beam, we should expect the beam to deflect in a symmetric up-and-down motion at the center. The result is difficult to discern, though we can note that the amplitude of the displacement-time graph does increase and decrease for each element, which is a good indication. We can plot more elements, as shown below (Figures 3 and 4).

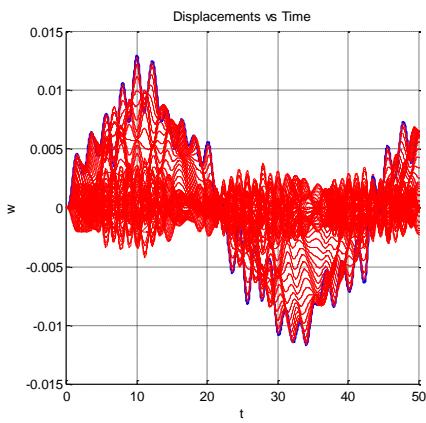


Figure 3

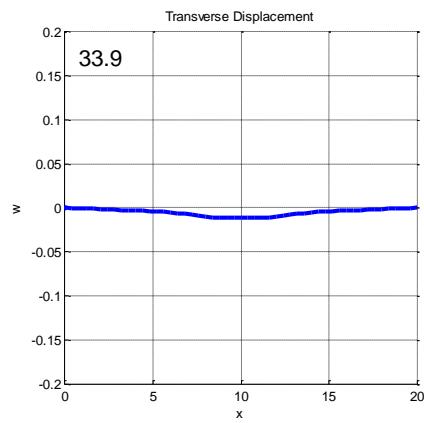


Figure 4

With 50 elements, the result is a bit interesting. What we were really wishing to see in the displacement-time graph is shown in Figure 3, where the *as a whole* move up and down. The now 50 element beam starts to behave more like a system than the individual elements. The nice bending shape is characterized in Figure 4. If we again increase the discretization requirement, this time 100, we can note that there is not much of a difference (sans computation time).

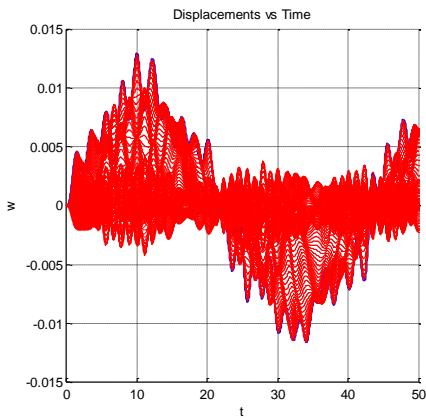


Figure 5

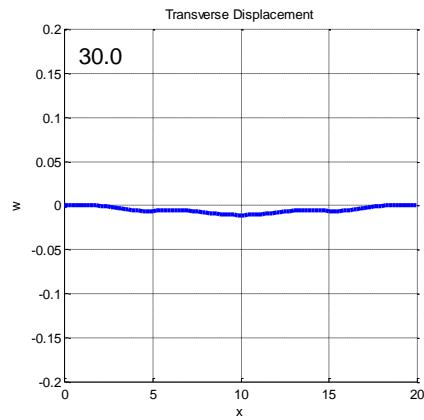


Figure 6

The displacement-time graph remains almost identical to the previous one, though there is more consistency in the joint efforts of the elements to provide a better response. The computation for this amount of elements is relatively quick, and the response as shown, is accurate enough for analysis.

### *The Boundary Conditions*

The boundary conditions have a great effects on the FWP of a beam. Noting the fixed-fixed condition of the beam, we have the followings plots.

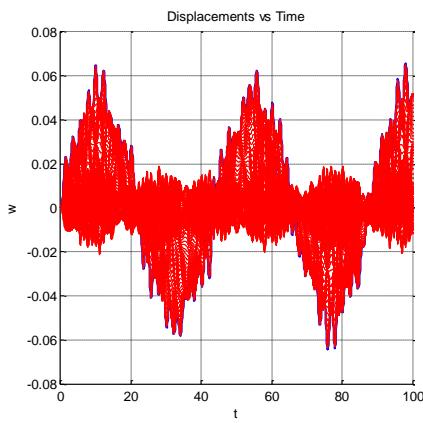


Figure 7

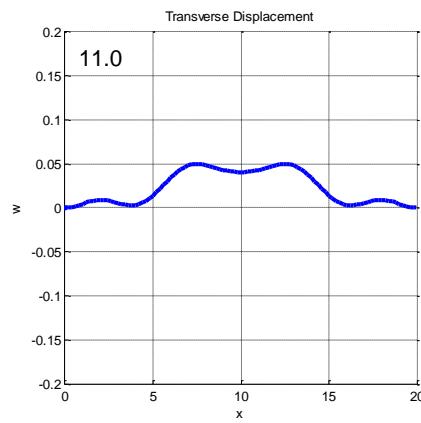


Figure 8

We can note immediately in the fixed-fixed condition that the waves do in fact flip sign when they come in contact with a fixed end. It may not be evident in the transverse displacement plot, but the motion can be depicted in the displacement-time graph. Note that the displacement in the members goes from being in the positive range to the negative range, and continues to repeat in this fashion. The fact that it is symmetrical is key – each fixed end forces the waves to travel back equal and opposite. For the free end, the FWP should basically be of that of a cantilever or “diving board”. The waves should propagate and cause relatively large displacements in the elements.

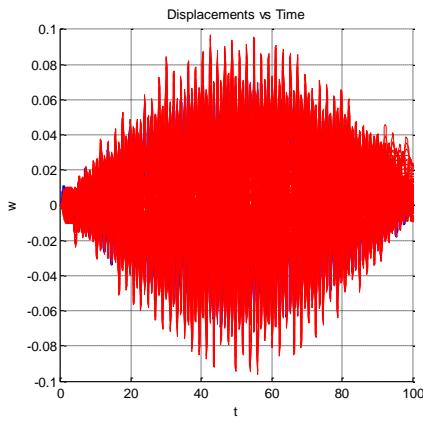


Figure 9

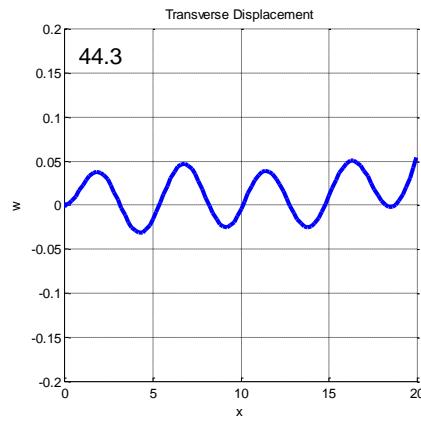


Figure 10

Interesting result! As expected and much similar to the axial bar, the magnitude of the element displacement increase for some time, and then begins to decrease around the 50 time mark (half the overall time). This makes intuitive sense since the end is free – the waves cannot go through anything, so they have to turn around and come back with the same sign and magnitude. This propagation leads to, coupled with the forcing function, increasing displacement. Since the forcing function is sinusoidal, it decreases after some time, and as a result, the beam's displacement decreases.

### Dispersion with Time

Unlike the axial bar, FWP in the beam is actually time dependent, and not simply a function of the square of the elastic modulus divided by the density. In essence, the flexural waves disperse with time, where the axial bar waves were constant. This phenomenon is apparent in the Figure 7, where we can see that the elements displace largely at some points on the graph (10 and 30 time marks), but then for some time the displacement is relatively small. The fact is, the waves are not always going from one large displacement at one time mark, and then on to the next. The flexural waves are *dispersing* throughout the beam. Consider the two plots below of the same beam at two different points in time.

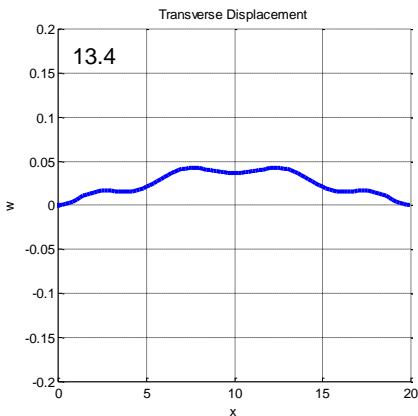


Figure 11

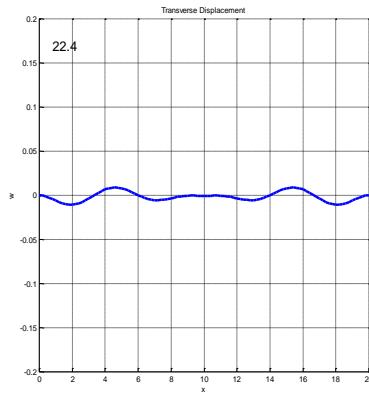


Figure 12

Note the shape of the deflection! This is what the displacement-time graph in Figure 7 is communicating – the waves persist over time, and are simply not like the axial bars that persist and propagate at a constant rate.

### Mode Shapes and Resonance

Finally, we can analyze the effects that resonance has on the FWP and gain an understanding of the modes of the beam. Below is a plot of the beam operating at the first mode under resonance.

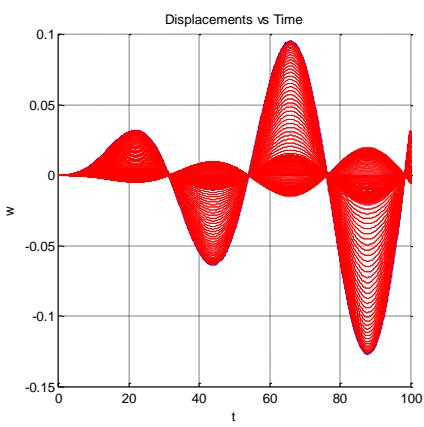


Figure 13

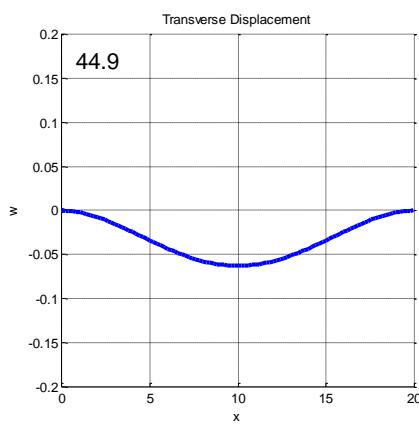


Figure 14

Note the shape of the first mode and the increase in displacement as time goes on. The result is almost identical to the wave propagation in an axial bar. The response is a pure mode that gradually increases in displacement. We can examine the second mode of vibration with resonance, as Figures 15 and 16 illustrate.

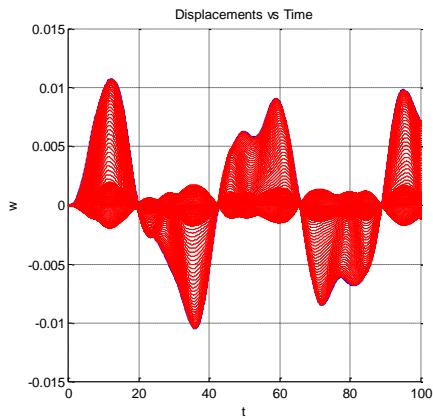


Figure 15

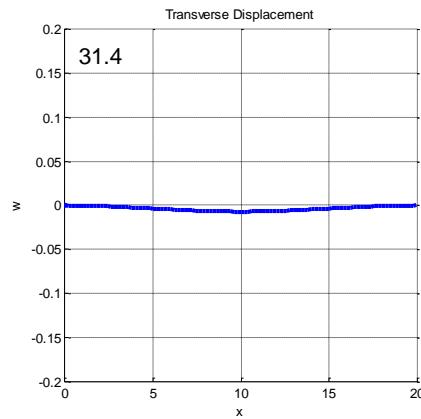


Figure 16

Interesting result! The expectation was to observe a complete, 1periodic sinusoidal motion of the beam. Instead, the result is something between the 1<sup>st</sup> and 2<sup>nd</sup> modes. Perhaps this is due to the nature of the beam in that FWP is a dispersion, and rather does not persist like the axial bar's wave propagation. The third mode clears things up, as shown below.

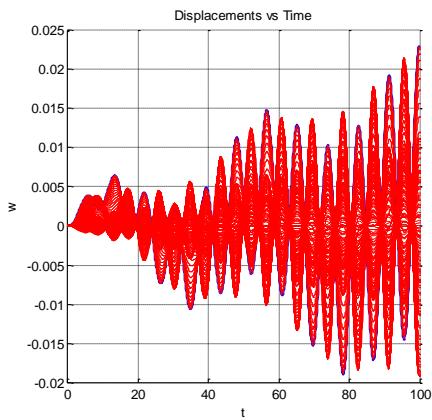


Figure 17

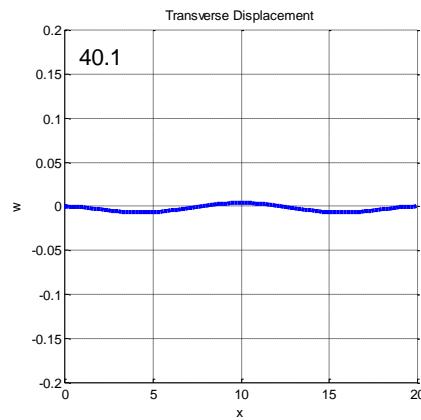


Figure 18

The obvious effects of resonance are seen here in Figure 17. The beam displaces at the typical third mode shape of the system, and is persistent to carry the pure motion of the mode over time. Will the 4<sup>th</sup> mode do the same as the 2<sup>nd</sup> mode? The below two plots capture the response of such a system.

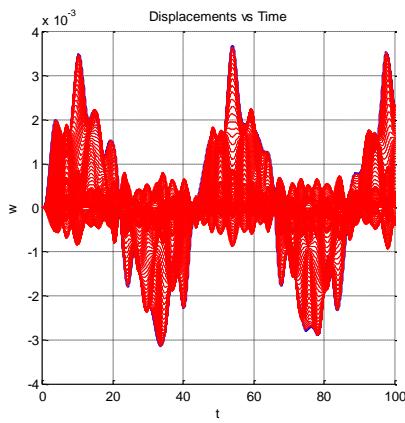


Figure 19

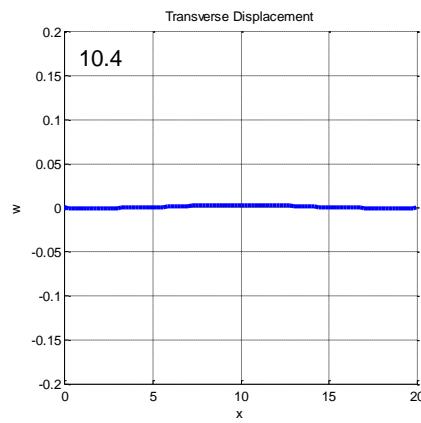


Figure 20

The same results happens in the 4<sup>th</sup> mode compared to the 2<sup>nd</sup>! There must be a pattern here in that every *even* mode or rather, every mode that has a whole number of sine periods, behaves in such a way. It appears that under a whole periodic motion, resonance is cancelled out, and the beam hardly displaces at all (note the magnitude).

We can view what would happen to a beam with a free end. The below two figures capture the response of the fixed-free system.

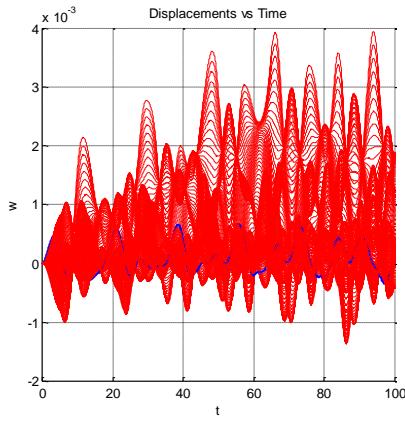


Figure 21

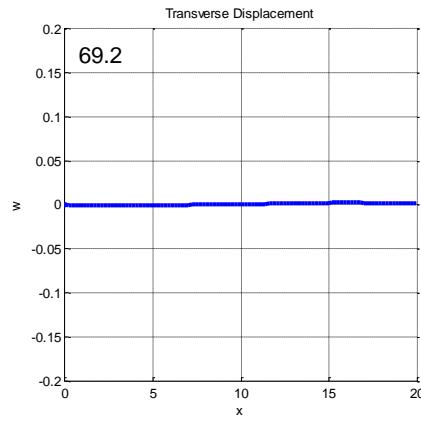


Figure 22

The same response occurs – resonance is sort of existent within the system, yet the beam hardly displaces. If we run the analysis for 150 time marks, we have,

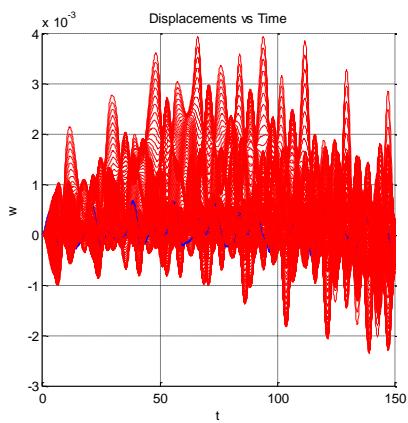


Figure 23

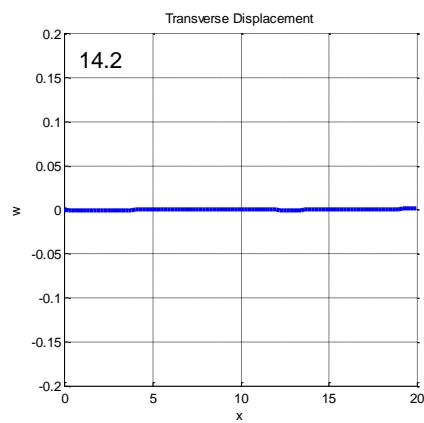


Figure 24

The beam begins to almost dampen out, it appears, but then displaces in the opposite direction. This is an interesting phenomenon for flexural wave propagation under mode shapes with whole order periods.