



MATLAB IMPLEMENTATION OF THE SPRING- MASS HARMONIC OSCILLATOR AND ITS RELEVANCE TO SOLVING REAL-WORLD PROBLEMS

Mani Bhushan Thoopal¹, S.Somashekar², Dr.K.T.Vasudevan³

¹*Mechanical, University Technolgy Sydney, (Australia)*

²*Mechanical, T.John Institute of Technolgy, Visveraya Technolical University Belgaum, (India)*

³*Physics, T.John Institute of Technolgy, Visveraya Technolical University Belgaum, (India)*

ABSTRACT

This report will focus on the spring mass system, with a comparative usage of MATLAB and numerical methods like Euler's method, the Euler Cromer method and the Runge-Kutta method to find and solve the equations of motion from the mathematical models produced. The report will initially analyse a Single DOF system, then go on to discuss the various types of damping, namely the underdamped, critically damped and overdamped conditions. After introducing a forcing term, plot of the displacement response vs frequency ratios at $w/w_n = 1.5, \sqrt{2}, 0.9$ and 1.0. A derivation of the analytical solution to a mathematical model of a 2DOF system with two masses, three springs and three dampeners will then be shown. This will be entailed by a MATLAB implementation of a similar system but without damping. The various displacement vs frequency ratio and phase angle vs frequency ration graphs will then be analysed and compared with the analytical solution derived earlier. An overview of the of the mathematics involved in modelling Multiple Degrees Of Freedom Systems will then be provided, with the focus solely on the mathematical aspect. The report will then conclude with some real world examples of systems that demonstrate the need for vibration understanding, analysis and control.

Keywords: *Dampers, Matlab, Numerical Methods*

I INTRODUCTION

Numerical Methods

Many a time, when models of physical systems are required for analysis, they can't be described by simple equations with simple trivial solutions. Differential equations are vital tools which can be used to describe a system in equations [8]. These equations however don't always have specific formulas that can be substituted in to find a solution. In fact, most differential equations don't even have explicit solutions. Before computers with large data processing abilities were available, scientists would often seek to find simpler mathematical models for complex systems, thus allowing them to arrive at solutions explicitly. This was always a work around, rather than a complete solution, as mathematical models generally make assumptions, thus making them applicable under certain circumstances.

With the arrival of the ‘computer era’, a whole new method of solving complex differential equations was brought forward. Rather than trying to find simple models, modern computers chug through large amounts of raw data, accepting input values of each equation, performing the required calculations and then outputting the result. As the speed and efficiency of processors increased, the amount of raw data that could be analysed in small period of time rapidly increased. Now, we have reached a point in computing technology where in calculations can be performed for an equation and resulting outputs can be tabulated into tables and visual graphs. This process can occur thousands of times over a small period of time, and the resultant trend can be analysed. As these trends are represented visually via the means of simulations and complex graphics usage, scientists and engineers have a powerful tool to analyse even extremely complex systems in a small amount of time. One such example is the Euler Method.

Numerical Methods

One Degree of Freedom Maths

Underdamped:

For this example, we’ ll use the following equation

$$y'' + 0.2y' + 1.01y = 0$$

This has roots $-0.1+i$ and $-0.1-i$. Thus the solution is complex and is of the form –

$$y(t) = e^{(-0.1+i)t} = e^{-0.1t} e^{it}$$

If we take the real and imaginary parts of the above solution we obtain the general solution –

$$k_1 e^{-0.1t} \cos(t) + k_2 e^{-0.1t} \sin(t)$$

This solution contains two decaying exponentials and oscillating cosine and sine terms. Thus we can predict that the mass will oscillate about the point of rest with an exponentially decreasing amplitude, eventually coming to rest. The following MATLAB graph demonstrates the displacement response of an under damped system (in this case a harmonic oscillator).

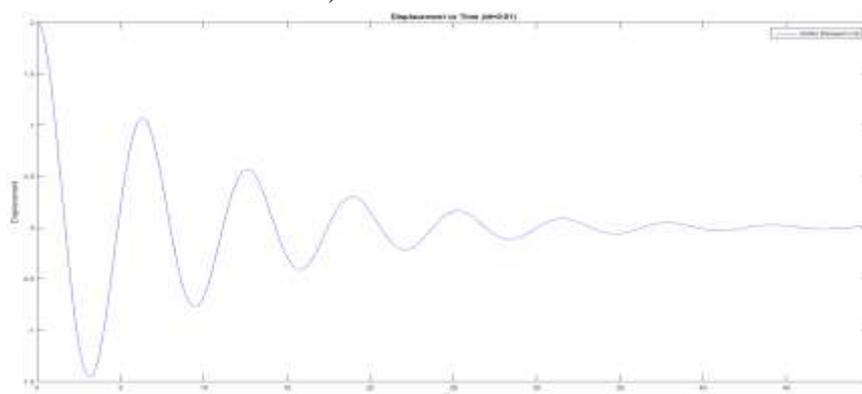


Figure1: Graph for displacement vs time for an under damped mass-spring system

Critically Damped

A system is said to be critically damped when the damping coefficient is greater than zero, and is equal to the



critical damping coefficient of the system. In this case, the characteristic equation will produce only one root. The following example illustrates a critically damped harmonic oscillator.

The equation used in this instance is –

$$y'' + 2y' + y = 0$$

This factors into –

$$(s + 1)^2 = 0$$

Thus

we have

$$k_1 e^{-t} + k_2 t e^{-t} \tag{25}$$

the real

root $s = -1$. The general solution is therefore –

We know that $k_1 e^{-t} \rightarrow 0$ as $t \rightarrow \infty$. However, in the expression $k_2 t e^{-t}$, we have a ‘t’ term which goes to infinity multiplied by a term which goes to zero. We can rewrite the second expression as – $\frac{t}{e^t}$

Now we have an expression where both the numerator and denominator go to infinity. At this point we can use l’Hopital’s rule which says that the limit of a function whose numerator and denominator both go to infinity is the same as that of its derivative;

$$\frac{d}{dt}(t) = 1 \text{ and } \frac{d}{dt} e^t = e^t$$

Thus we have –

$$\frac{1}{e^t} \rightarrow 0$$

Now we know that the second expression in (25) also goes to zero. Thus we’ll expect the amplitude of the mass to decrease relatively quickly down to the rest position. The following MATLAB graph demonstrates the displacement response of a critically damped system.

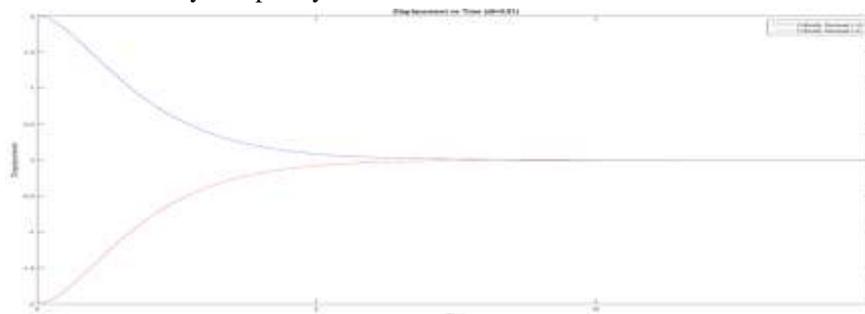


Figure 2 - Graph for displacement vs time for an underdamped mass-spring system

Overdamped: $(b > 0 \text{ \& } b > bc)$

An overdamped system can be achieved when the spring constant is equal to 2 and the damping constant is 3, for example, forming the following differential equation:

$$y'' + 3y' + 2y = 0$$

This gives the following differential equation:

$$s^2 + 3s + 2 = 0$$

This results in the roots $s_1 = -1$ and $s_2 = -2$. Hence the general solution becomes:

$$y(t) = k_1 e^{-t} + k_2 e^{-2t}$$

The following graph comes from MATLAB. The code used describes an unforced mass-spring system. The code for all MATLAB graphs can be found in the appendix section of this report.

There are no units for the 'displacement' and 'time' variables in the following graph as the units are arbitrary.

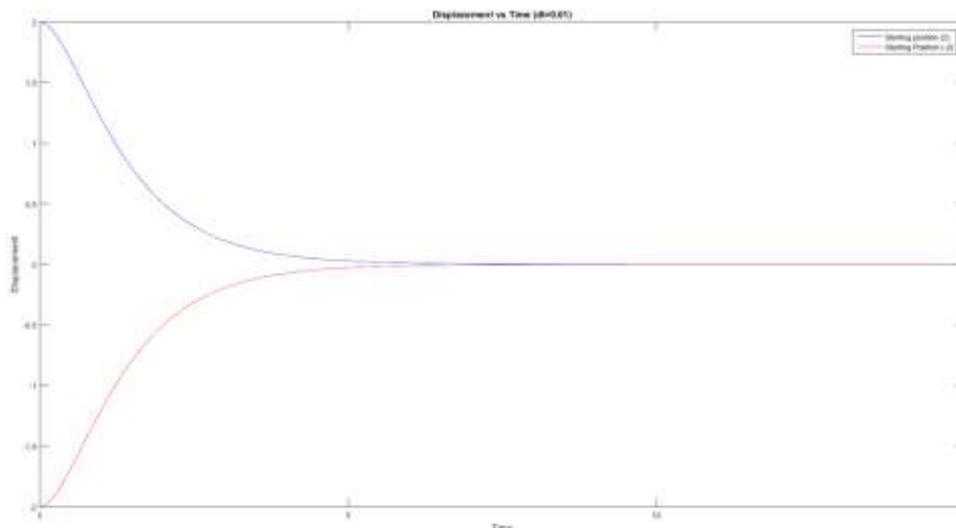


Figure 3 - Graph of Displacement Vs Time for an overdamped 1DOF spring-mass system

Forced Oscillator:

Let the following differential equation be used as an example:

$$y'' + 3y' + 2y = \cos(t)$$

The characteristic equation of the homogenous form of this equation is –

$$s^2 + 3s + 2 = 0$$

$$(s + 2)(s + 1) = 0$$

Thus the roots are $s = -2, -1$

If we solve for the homogeneous version of this equation we obtain the following solution –

$$y(t) = k_1 e^{-2t} + k_2 e^{-t}$$

Thus the unforced oscillator will tend to zero. In order to find a solution to the forced oscillator, we need to guess a possible solution. $A \sin(t)$ and $B \cos(t)$ can't be solutions because the y' term gives a cosine term in the case of $A \sin(t)$ and vice versa in the case of $B \cos(t)$. Guessing a combination of the two, however, works: $A \sin(t) + B \cos(t)$.

Plugging in this into equation (27) gives the solution $A = 3/10$ and $B = 1/10$. Thus any solution of this equation is of the form –

$$y(t) = k_1 e^{-2t} + k_2 e^{-t} + \frac{3}{10} \sin(t) + \frac{1}{10} \cos(t)$$

Since the first two terms will always tend to zero regardless of what k_2 is, the solution will always end up as –

$$y(t) = \frac{3}{10} \sin(t) + \frac{1}{10} \cos(t)$$

This is the steady-state solution and has a period of 2π , which is the frequency of the forcing term. The following graph from MATLAB illustrates this behaviour. Four separate starting points are taken and the system always ends up oscillating at the same frequency as the input force, regardless of the starting position.

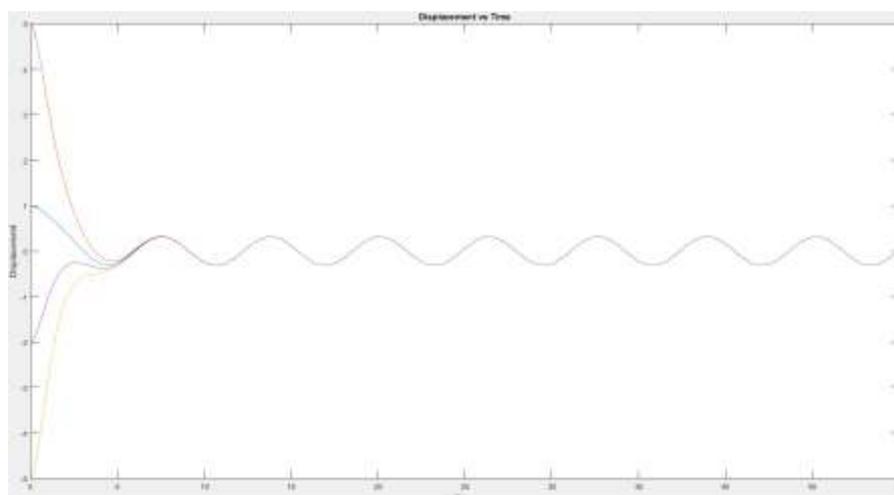


Figure 4 - Displacement vs Time for the force, damped harmonic oscillator

The different starting points in the above graph are – 1 unit above zero (blue line), 2 units below zero (purple line), 5 units below zero (yellow line) and 5 unites above zero (orange line).

Forced Oscillator without damping

Let the following differential equation be used as an example:

$$y'' + y = \cos(\omega t)$$

The homogeneous equation then becomes:

$$y'' + y = 0$$

Thus the general solution to the homogeneous equation is:

$$k_1 \cos(t) + k_2 \sin(t)$$

The solution to this equation does not reduce to rest, and to obtain the full non-homogeneous solution we make a similar guess as before; $A \sin(t)$ and $B \cos(t)$. Putting this into the differential equation we get:

$$B(1 - \omega^2) \cos(\omega t) + A(1 + \omega^2) \sin(\omega t) = y'' + y = \cos(\omega t)$$

This means that $B = \frac{1}{1 - \omega^2}$ and $A = 0$. Thus the particular solution to the differential equation:

$$\frac{1}{1 - \omega^2} \cos(\omega t) + k_1 \cos(t) + k_2 \sin(t)$$

Now we'll look at what happens to the system when the forcing term's frequency is set at different values.

At $\omega = 1.5$

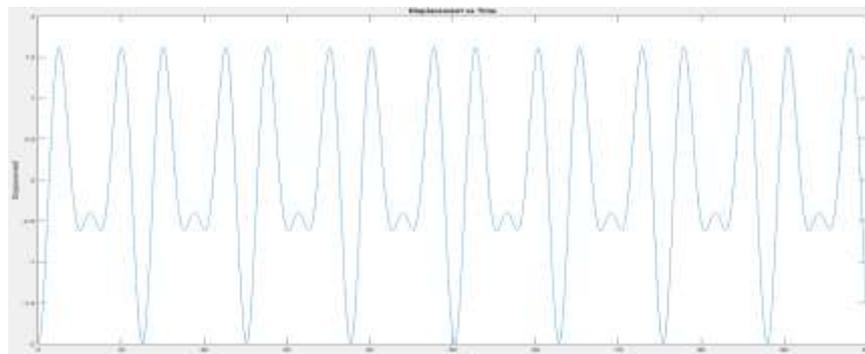


Figure 5 - Displacement vs Time for undamped, forced harmonic oscillator with $\omega = 1.5$

The response seen is visibly periodic, however it doesn't display a smooth sine response as seen in the earlier response.

With the forcing frequency as an irrational number, the above graph shows that the motion of the mass is no longer periodic. At $\omega = 0.9$

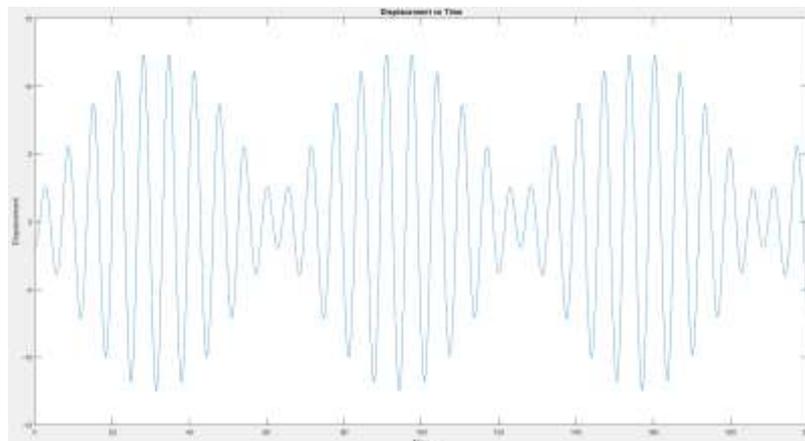


Figure 7 - Displacement vs Time of undamped, forced harmonic oscillator with $\omega = 0.9$

The response seen above is a good example of the 'beats' phenomenon. 'Beats' are produced when the frequency of the external input is extremely close to, but not equal to the natural frequency of the system.

Finally, when $\omega = 1$:

$$y'' + y = \cos(t)$$

In this new case, a solution in the form of: $A\cos(t) + B\sin(t)$ cannot be used, as this is a solution for a homogenous equation. As this is a non-homogenous equation. Hence a new solution must be assumed, and is in the form of:

$$At\cos(t) + Bt\sin(t)$$

Substituting this into the above differential equation then gives:

$$y'' + y = -2A\sin(t) + 2B\cos(t)$$

Thus:

$$-2A\sin(t) + 2B\cos(t) = \cos(t)$$

Solving the above equation yields $A = 0$ and $B = \frac{1}{2}$, making the general solution

$$\frac{1}{2}t\sin(t) + k_1 \cos(t) + k_2 \sin(t)$$

Assuming initial conditions $y(0) = 0$ and $y'(0) = 0$, the solution which satisfies these conditions is:

$$y(t) = \frac{1}{2}t\sin(t)$$

It can clearly be seen that the displacement $y(t)$ is directly proportional to the time t . Hence, the amplitude of the displacement will proportionately increase with time, and can be seen in the figure MATLAB output below. This phenomenon is known as **resonance**. Hence, the phenomenon of increase in the amplitude of oscillation when a system is exposed to a periodic force, whose frequency is equal to the undamped natural frequency of the system is called resonance.

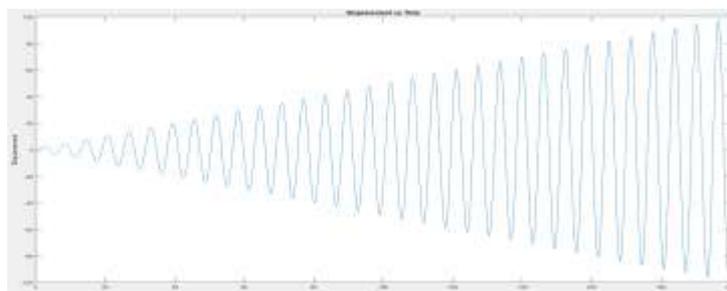


Figure 8 - Displacement vs Time for undamped, forced harmonic oscillator with $\omega = 1$

Resonance is generally considered to be harmful, and can have devastating effects on engineering structures when triggered. This is discussed in a later section, which talks about the collapse of the Tacoma Narrows Bridge in 1940. It can however be useful in a system. An example where resonance is essential is in wireless electricity transfer using inductive coupling. The primary and secondary coils must have a certain number of turns, with the right frequency of current in order for them to resonate with each other.

II MATLAB IMPLEMENTATION OF 2DOF SYSTEM

In order to use MATLAB to compute the response to a 2DOF system, a slightly simpler variant of the system described earlier will be used. Instead of the system having three springs and three dampeners, a system having two masses, two springs and two dampeners will be used instead. The system diagram is given below:

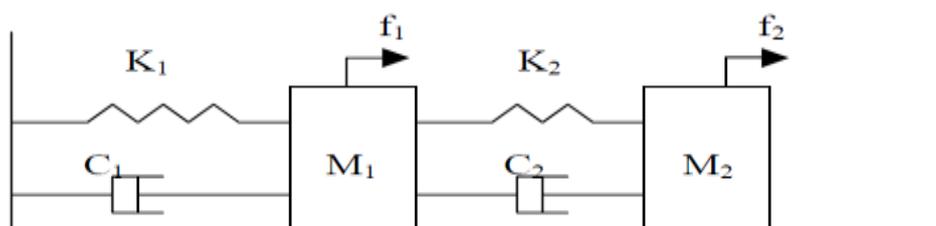


Figure 9 – 2DOF System Diagram

The specific matrices used to produce the graph below are as follows:

$$\text{Mass matrix} - \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \text{ Where } M_1 = 2 \text{ and } M_3 = 3$$

$$\text{Stiffness matrix} - \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \text{ Where } K_1 = 2 \text{ and } K_2 = 1$$

$$\text{Damping matrix} - \begin{bmatrix} 0.003 & -0.001 \\ -0.001 & 0.001 \end{bmatrix} \text{ Where } C_1 = 0.003 \text{ and } C_2 = 0.001$$

$$\text{Force matrix} - \begin{bmatrix} 2 \\ 1 \end{bmatrix} \text{ Where } f_1 = 2 \text{ and } f_2 = 1$$

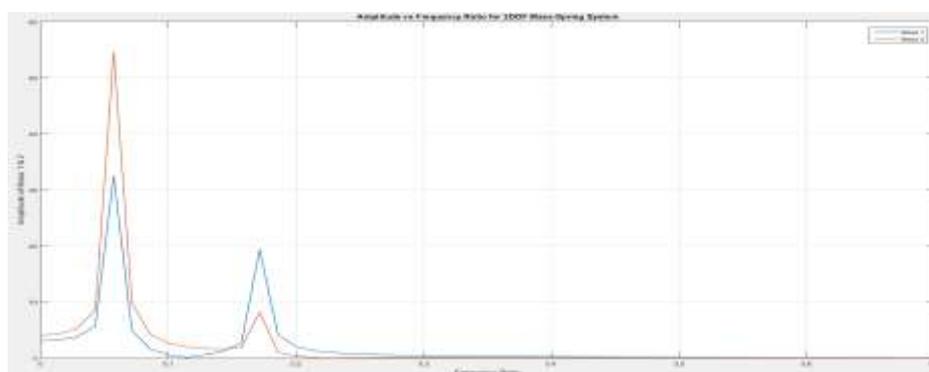


Figure 10 - Amplitudes of two masses vs Frequency ratio

The above graph represents the displacement response of the 2DOF system with the variance of the frequency ratio w/w_n using the input parameters given earlier. The two observable peaks seen on the graph represent the two vibrational modes of the system. Each mode of vibration occurs at a particular frequency, at which the plot shows a steep rise. The first mode occurs at a lower frequency, and the displacement in this mode for both masses are at their largest. As f_1 has twice the amplitude of f_2 , mass 1 also has a displacement proportionally higher than mass 2. We know that the lower the damping, the lesser the resistance to motion, and hence greater the displacements. In the ideal situation that was attempted to be plotted, there should have been no damping. MATLAB however encounters a problem when faced with a zero matrix. This is because taking the inverse of a zero matrix leads to a singularity error. Hence a value close to zero had to be entered for the damping matrix instead. Another interesting phenomenon observable is the presence of the 'anti resonant' frequency. When the frequency ratio is roughly 0.1, it can be seen that the displacement of mass 2 completely drops to zero, while the displacement of mass one approaches it. An extremely useful application of this anti resonant frequency would be to in vibration control. It can be seen in the above graph that if the frequency ratio is increased past 0.1 the amplitude starts to rise again. In systems facing vibration issues, for example a car suspension system resonating with a certain engine frequency, adding a spring or changing the value of the existing spring would change the value of the natural frequency of the system. Hence, instead of the resonating frequency being constantly hit, it

could be tuned such that the anti-resonating frequency is hit constantly instead. This would drastically reduce the vibrations caused due to resonance.

III CONCLUSION AND FUTURE OUTLOOK

This report was focused on the spring mass system, with a comparative usage of MATLAB and numerical methods like Euler's method, the Euler Cromer method and the Runge-Kutta method to find and solve the equations of motion of the various systems discussed. The report initially analysed a Single DOF system, describing the effects and implications of both forced and unforced systems. It then went on to discuss the various types of damping, namely the underdamped, critically damped and overdamped conditions.

A forcing term was then introduced to the system, and the system's response was plotted into several graphs using MATLAB. These graphs illustrated the effects of the forced input at different frequency ratios. Instances where the frequency ratio, w/w_n was equal to 1.5, $\sqrt{2}$, 0.9 and 1.0. Each case produced an interesting non trivial output. When the frequency ratio was $\sqrt{2}$ an interesting response of jagged waves were seen in stark contrast to the smooth sine waves normally observed. Another interesting response was noted when the frequency ratio was equal to one, with the phenomenon of resonance being observed. The system responses of the damped 1DOF system from an analytical perspective and through MATLAB were then compared, and it was seen that the steady state response was always in phase with the forced input.

A simple 2DOF system with two masses, three springs and three dampeners then introduced, and the underlying mathematical derivation of the steady state response was deliberated upon. A MATLAB implementation of an undamped 2DOF system was undertaken, with a focus on the description of the displacement of masses vs frequency and phase of masses vs frequency responses. It was seen that when the frequency ratio was raised in steps of 0.1, the displacements would always drop down to zero (or extremely close to it) before steeply rising again. It was also seen that the phases would vary by 180 degrees each time the frequency ratios were increased by 0.1.

Through this report, a firm belief that an understanding of wave dynamics and vibration control is critical in the proper functioning of engineering structures. Resonance can prove disastrous, but can also be harnessed and used in applications like wireless electricity transfer and radio wave enhancements.

References

- [1]Wikipedia, 4/02/2015, "*Symplectic Integrator*", viewed 10/04/2015,
http://en.wikipedia.org/wiki/Symplectic_integrator
- [2]Astrophysics at North Carolina State University, "*Introduction to Scientific Computing*", North Carolina State University, viewed 10/04/2015,
<http://astro.physics.ncsu.edu/~cekolb/urca/lessons/14.html>
- [3]Introduction to Dynamics and Vibrations, "*5.5 Introduction to vibration of systems with many degrees of freedom*", School of Engineering - Brown University, viewed 13/04/2015,
http://www.brown.edu/Departments/Engineering/Courses/En4/Notes/vibrations_mdof/vibrations_mdof.htm



- [4]Howard. P, 2009, “*Solving ODE in MATLAB*”, viewed 10/04/2015
- [5]Introduction to Aerospace Structures (ASEN 3112), “*MDOF Dynamic Systems*”, Department of Aerospace Engineering Sciences – University of Colorado at Boulder, 2014, Lecture 19, p.1-24, viewed 15/05/2015
- [6]Wikipedia, 27/05/2015, “*Runge-Kutta Methods*”, viewed 28/05/2015,
http://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods
- [7]Harihara P., Childs D.W., “*Solving Problems in Dynamics and Vibrations Using MATLAB*”, Dept of Mechanical Engineering – Texas A&M University – College Station, p.56-59, viewed 12/05/2015