EFFICIENT COMPUTATION OF FREQUENCY RESPONSE OF MULTI-DEGREE OF FREEDOM NON-LINEAR VIBRATIONAL SYSTEM

A Thesis in
Mechanical Engineering
by
Arjun Pradeep Kumar

© 2017 Arjun Pradeep Kumar

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Masters of Science

August 2017
The thesis of Arjun Pradeep Kumar was reviewed and approved* by the following:

ALOK SINHA
Professor of Mechanical Engineering
Thesis Advisor

QIAN WANG
Professor of Mechanical Engineering

KAREN THOLE
Professor of Mechanical Engineering
Head of the Department of Mechanical and Nuclear Engineering Program

*Signatures are on file in the Graduate School
ABSTRACT

Frequency sweep problems occur in several applications of structural dynamics, acoustics and structural acoustics. In general, the evaluation of a frequency response function involves finding solution to a large-scale system of coupled equations defining a vast system. Hence finding solutions to frequency response functions for a large range of frequencies is computationally exhaustive. However, the established method of interpolation techniques can be implemented to reduce the cost of computation. So far, several techniques of interpolation techniques have been successfully implemented in systems involving large-scale coupled linear equations. This thesis proposes implementation of Padé’s interpolation technique in a large-scale nonlinear system. More specifically, this thesis focuses on the additional computational efforts required in finding solution to frequency sweep problems of large nonlinear systems when compared with large linear systems. The accuracy and computational efficiency of the mentioned approach are demonstrated with solutions to frequency sweep problems for a single and two degrees of freedom of nonlinear systems. Further this thesis has discussed methods to approach solution of frequency response problem of a multi-degree of freedom nonlinear system using finite element method.

KEY WORDS: frequency sweep; interpolation; Padé’s interpolation technique; non-linear systems; structural acoustics; structural dynamics; finite element method.
# TABLE OF CONTENTS

List of Figures ............................................................................................................. v

Acknowledgements ....................................................................................................... vii

Chapter 1 Introduction ............................................................................................... 1

Chapter 2 Interpolation technique .......................................................................... 4
  2.1 Single-point Padé interpolation technique ....................................................... 5
  2.2 Multi-point Padé interpolation technique ....................................................... 7

Chapter 3 Analysis of a single degree of freedom non-linear vibrational system ...... 10
  3.1 Model and governing equation of motion ....................................................... 10
  3.2 Analytical solution .......................................................................................... 12
  3.3 Solution by Padé’s interpolation method ....................................................... 14
  3.4 Comparison of results .................................................................................... 18

Chapter 4 General approach to analysis of a multi-degree of freedom non-linear vibrational system ............................................................................. 25
  4.1 Model and governing equation of motion ....................................................... 25
  4.2 Analytical solution .......................................................................................... 26
  4.3 Solution by Padé’s interpolation method ....................................................... 29

Chapter 5 Analysis of a two degrees of freedom non-linear vibrational system .......... 35
  5.1 Model and governing equation of motion ....................................................... 35
  5.2 Analytical solution .......................................................................................... 36
  5.3 Solution by Padé’s interpolation method ....................................................... 40
  5.4 Comparison of Results .................................................................................... 41

Chapter 6 Finite element analysis of non-linear vibrational system ......................... 47
  6.1 Finite element analysis of a beam structure with attached frictionally damped spring ......................................................................................... 49
  6.2 Finite element analysis of a sample beam structure ...................................... 52

Chapter 7 Conclusion ............................................................................................... 61

Appendix A MATLAB program code for chapter-3 ................................................. 63
Appendix B Evaluation of Nonlinear Term ............................................................... 71
Appendix C MATLAB program code for chapter-5 ................................................ 74
Appendix D MATLAB program code for chapter-6 ................................................ 87
REFERENCES ............................................................................................................... 97
LIST OF FIGURES

Figure 3-1. Equivalent single degree of freedom system of blade to damper model. Source [3]........................................................................................................................................10

Figure 3-2. Nondimensionalized single degree of freedom system of blade to damper model with positive damping coefficient. Source [3]..................................................................................11

Figure 3-3. Comparison of values of amplitude function $A(\omega)$ and Padé’s approximant amplitude function $A_{\text{Pade}}(\omega)$ ........................................................................................................20

Figure 3-4. Error between analytical solution of amplitude function $A(\omega)$ and Padé’s approximant amplitude function $A_{\text{Pade}}(\omega)$ values ............................................................................21

Figure 3-5. Comparison of values of amplitude function $A(\omega)$ and Padé’s approximant amplitude function $A_{\text{Pade}}(\omega)$ with increased sample spaces ............................................................................22

Figure 3-6. Comparison of values of amplitude function $A(\omega)$ and Padé’s approximant amplitude function $A_{\text{Pade}}(\omega)$ of second sample space ............................................................................23

Figure 3-7. Error between analytical solution of amplitude function $A(\omega)$ and Padé’s approximant amplitude function $A_{\text{Pade}}(\omega)$ values ............................................................................24

Figure 5-1. Two degrees of freedom system of blade to damper model .........................................................................................................................35

Figure 5-2. Comparison of values of amplitude function $A_{1}(\omega)$ and Padé’s approximant amplitude function $A_{1,\text{Pade}}(\omega)$ ..................................................................................................43

Figure 5-3. Comparison of values of amplitude function $A_{2}(\omega)$ and Padé’s approximant amplitude function $A_{2,\text{Pade}}(\omega)$ ..................................................................................................43

Figure 5-4. Error between amplitude function $A_{1}(\omega)$ and Padé’s approximant amplitude function $A_{1,\text{Pade}}(\omega)$ ..................................................................................................45

Figure 5-5. Error between amplitude function $A_{2}(\omega)$ and Padé’s approximant amplitude function $A_{2,\text{Pade}}(\omega)$ ..................................................................................................45

Figure 6-1. Beam element of uniform cross-sectional area .................................................................................................................................49

Figure 6-2. The cantilever beam structure of interest ........................................................................................................................................52
Figure 6-3. The shape of the lateral displacement of the cantilever beam structure for different excitation frequencies varying from 200 rad/s to 245 rad/s. ..........................56

Figure 6-4. a) Comparison of analytical and Padé’s approximant solution to amplitude function \( A_1(\omega) \), b) Error between the analytical and Padé’s approximant solution values. ........................................................................................................................................56

Figure 6-5. a) Comparison of analytical and Padé’s approximant solution to amplitude function \( A_2(\omega) \), b) Error between the analytical and Padé’s approximant solution values. ........................................................................................................................................57

Figure 6-6. a) Comparison of analytical and Padé’s approximant solution to amplitude function \( A_3(\omega) \), b) Error between the analytical and Padé’s approximant solution values. ........................................................................................................................................57

Figure 6-7. a) Comparison of analytical and Padé’s approximant solution to amplitude function \( A_4(\omega) \), b) Error between the analytical and Padé’s approximant solution values. ........................................................................................................................................58

Figure 6-8. a) Comparison of analytical and Padé’s approximant solution to amplitude function \( A_5(\omega) \), b) Error between the analytical and Padé’s approximant solution values. ........................................................................................................................................58

Figure 6-9. a) Comparison of analytical and Padé’s approximant solution to amplitude function \( A_6(\omega) \), b) Error between the analytical and Padé’s approximant solution values. ........................................................................................................................................59

Figure 6-10. a) Comparison of analytical and Padé’s approximant solution to amplitude function \( A_7(\omega) \), b) Error between the analytical and Padé’s approximant solution values. ........................................................................................................................................59

Figure 6-11. a) Comparison of analytical and Padé’s approximant solution to amplitude function \( A_8(\omega) \), b) Error between the analytical and Padé’s approximant solution values. ........................................................................................................................................60

Figure B-1. Frictionally damped spring with spring constant \( K_d \). ..............................................71

Figure B-2. Frictionally damped spring force vs displacement \( A_i \). ..............................................71
ACKNOWLEDGEMENTS

I would like to extend my whole-hearted gratitude to Dr. Alok Sinha for his continuous guidance and support throughout the course of this thesis. Working with Dr. Alok Sinha was such an enriching experience and his unflinching commitment towards top-quality research and his continuous motivation has greatly inspired me to deliver this thesis with great zeal and dedication. During the course of this thesis, I have had several insightful discussions with him that gave me an in-depth understanding into really rigorous concepts spanning across multiple domains. I would like to thank him for his timely inputs in terms of knowledge and resources that he extended me which assisted greatly towards the successful completion of this thesis.

I would also like to thank my friends and family who have stood by me in times of distress, who have constantly motivated me over the last four years to pursue my dreams successfully. I am greatly indebted to them and but for their support, a lot of my dreams would have remained distant.
Chapter 1

Introduction

The frequency sweep problems appear in various computational engineering fields such as structural dynamics, fluid dynamics, acoustics and vibro-acoustics which involve formulations in the frequency domain. The frequency response functions are to be evaluated over a frequency band in such cases. According to [1], in most cases of structural dynamics, acoustics and structural acoustics the problem of interest takes the form as given by equation (1.1).

\[
Z(\omega)u(\omega) = f(\omega)
\]

where \( Z(\omega) = (K - \omega^2 M + i\omega C) \) is the impedance matrix with \( K, M \) and \( C \) denoting the stiffness, mass and damping matrices of the given computational model, and \( f(\omega) \) is the amplitude function of the harmonic external excitation characterized by an angular frequency \( \omega \) which belongs to the angular frequency band of interest \([\omega_L, \omega_H]\). In general, this involves brute force method of finding solutions to the frequency response functions for the entire frequency domain of interest. However, the straightforward solutions to such problems are computationally exhaustive and expensive. Several methods have been developed for alleviating this computational burden by approximating the frequency response functions. Model order reduction is one of the techniques for reducing the quantum of computations involved in finding solutions to problems involving mathematical simulations. The challenges encountered in numerical simulations are mainly the complexity and the large magnitude of computations. Model order reduction is a technique to decrease the computational complexity of such problems, for example, in simulations of large-scale systems by a reduction of the model's dimensions or degrees of freedom to get an approximation of the original model. Among the other methods, from [1] it is
claimed that interpolation methods are perhaps the most successful. Interpolation methods are approximation methods for evaluating the frequency response functions. They work by precomputing such functions and some of their derivatives at certain frequencies of interest, then interpolating the data to predict the responses at other desired frequencies even though with a slightly reduced degree of accuracy. Their main objective is to greatly reduce the cost of frequency sweep analyses in computational fields such as structural dynamics, acoustics, and vibro-acoustics. One of the widely-used interpolation techniques is the Padé’s interpolation technique which according to [2] is based on the fact that the solution to the frequency sweep problems can be reconstructed in the frequency band of interest by making use of the computed solutions and their derivatives at lesser number of frequency points of interest.

So far, the Padé’s interpolation technique has been successfully established for those frequency sweep problems of the form given by equation (1.1) which involve linear governing equations of motion. To the best of my knowledge, the application of Padé’s interpolation method to frequency sweep problems entailing nonlinear governing equations of motion has not been explored or evaluated in great detail, as evaluating the derivatives of solutions of such problems posed challenges due to the massive scale of computational requirements. In contrary, this thesis proposes methods for the application of Padé’s interpolation technique on that class of nonlinear frequency sweep problems which involves response function of predominantly single harmonic output in response to a single harmonic input characterized by an angular frequency. In particular, this thesis focus on implementation of an interpolation technique in nonlinear frequency sweep problem of the form given by equation (1.1) but with the impedance matrix \( Z \) given by the equation (1.2).

\[
Z = \left( K(u) - \omega^2 M + i\omega C(u) \right)
\]  

(1.2)
where the stiffness matrix $K$ and the damping matrix $C$ are functions of the amplitude of the frequency response function $u(\omega)$ of the computational model. The impedance matrix $Z$ as given in equation (1.2) is chosen to be the problem of interest, as it represents a variety of vibrational models of engineering applications, for instance vibrational models involving frictional dampers. To begin with, this thesis discusses the computational efficiency of the Padé’s interpolation method for a single degree of freedom nonlinear vibrational system involving friction damping. This thesis proposes special algorithms to carry out the required computations to evaluate the derivatives of solution at the selected interpolation points in the frequency domain of interest. This thesis also discusses about the accuracy, feasibility and the quantum of supplementary computational efforts required for the successful implementation of the interpolation technique. Further with the establishment of a general approach to a multi-degree of freedom nonlinear vibrational system, the thesis demonstrates the successful implementation of the same on to a two degrees of freedom nonlinear vibrational system. In addition, the extension of the Padé’s interpolation technique to finite element method and its application to large-scale nonlinear vibrational systems have also been discussed and demonstrated on a simple vibrating beam structure involving nonlinear aspects owing to the attached frictionally damped spring.
Chapter 2

Interpolation technique

The study of complex physical systems of scientific interest and industrial value involves large-scale simulations of models that represent the key characteristics, behaviors and functions of the selected physical system or process. This often leads to analysis of large-scale and complex dynamic models that requires extensive computational resources and efforts, limiting the extent and scope of the study. However, with a little compensation on the demands of the accuracy of the study, it would be possible to produce approximate models which are capable of accurately approximating the behavior of the original model. An approximate model with high fidelity can be used as a reliable substitute to the original model which is efficient and less time consuming in nature when compared with the original model.

Interpolation techniques have proved to be successful approximation methods of large-scale dynamic systems. The interpolation involves construction of an approximate model that matches the values of the original model at selected interpolation points in the frequency domain. As the problem of interest is the frequency sweep of large-scale vibrational models, most of the interpolation techniques are rooted in the interpolation of the solution function \( u(\sigma) \), equation (1.1), in the frequency domain \([\omega_L, \omega_R]\). The interpolation techniques such as Taylor series and Padé expansions upon implementation reproduces the exact amplitude function values at the selected interpolation points and produces fairly approximate solutions to amplitude functions at the points belonging to the interpolated regions. According to [1] Padé expansions are
numerically stable over a larger domain of frequency and hence Padé expansions are preferred over Taylor series, particularly when applied to functions containing poles.

### 2.1 Single-point Padé interpolation technique

The single-point Padé interpolation technique develops a rational function which is an approximate to the original solution function by matching the values of the original function values and the values of its derivatives at a selected point in the frequency domain. Let $u(\sigma)$ denote the solution function of the given computational model and $\sigma_0$ denote the selected point in the frequency domain $[\omega_L, \omega_R]$. According to [2], the single-point Padé interpolation technique claims that the value of the original solution function at any point in the neighborhood of $\sigma_0$ which is given by $u(\sigma_0 + \Delta \sigma)$ can be approximated as a rational function as given by equation (2.1).

\[
u(\sigma_0 + \Delta \sigma) \approx \frac{P(\sigma_0 + \Delta \sigma)}{Q(\sigma_0 + \Delta \sigma)}
\]

(2.1)

Where,

\[
P(\sigma_0 + \Delta \sigma) = p_0 + p_1(\sigma_0 + \Delta \sigma) + \ldots + p_L(\sigma_0 + \Delta \sigma)^L
\]

is of degree $L$ 

(2.2)

\[
Q(\sigma_0 + \Delta \sigma) = 1 + q_1(\sigma_0 + \Delta \sigma) + \ldots + q_M(\sigma_0 + \Delta \sigma)^M
\]

is of degree $M$, and is normalized to have a unit constant coefficient.

The equation (2.2) is rewritten as given by equation (2.4).

\[
u(\sigma_0 + \Delta \sigma)Q(\sigma_0 + \Delta \sigma) = P(\sigma_0 + \Delta \sigma)
\]

(2.4)
Substitution of equations (2.2) and (2.3) into equation (2.4) and successful differentiation of equation (2.4) with respect to \((\sigma_0 + \Delta \sigma)\) leads to a series of equations which takes the form as given by equation (2.5).

\[
\sum_{r=0}^{k} \binom{k}{r} u^{(k-r)}(\sigma_0 + \Delta \sigma)^{l-r} q_r - \sum_{l=k}^{L} \frac{l!}{(l-k)!} p_l (\sigma_0 + \Delta \sigma)^{l-k} = 0
\]

(2.5)

where \(\binom{k}{r} = \frac{(k)!}{(r)!(k-r)!}\) with \((k)\) denoting the factorial of \(k\) and where \(u^{[v]}\) denotes the \(v^{th}\) derivative of the function \(u(\sigma)\) with respect to \(\sigma\) at \((\sigma_0 + \Delta \sigma)\),

\(k = 0,1,...,L+M\) with,

\(q_r = 0\) if \(r > M\) and \(p_k = 0\) if \(k > L\).

It is observed that for the particular case of \(\Delta \sigma = 0\), the equation (2.5) simplifies to equation (2.6) which confirms that the function values and its derivatives be matched with the values of the rational function at \((\sigma_0)\).

\[
\sum_{r=0}^{k} \binom{k}{r} u^{(k-r)}(\sigma_0)^{l-r} q_r - k! p_k = -u(\sigma_0)^{[k]}
\]

(2.6)

Equation (2.5) can be written in matrix form as given by equation (2.7).

\[
[A] \{x\} = \{b\}
\]

(2.7)

Where \([A] \in C^{(L+M+1) \times (L+M+1)}\), \(\{x\} = \left[ p_0 \ldots p_L q_1 \ldots q_M \right]^T\)

(2.8)

The coefficients of the polynomial functions \(P\) and \(Q\) can be evaluated by inverting the matrix \([A]\) and multiplying it with the vector \(\{b\}\) from equation (2.7). According to [2] the developed one-point \(\left[ L \atop M \right]\) rational function as given in equation (2.1) can deliver an approximate solution
to the function $u(\sigma)$ faster than the straightforward frequency sweep which involves repeated solution of the given problem at every interested point in the frequency domain $[\omega_L, \omega_R]$.

### 2.2 Multi-point Padé interpolation technique

According to [2], the single-point Padé interpolation technique can be extended to multi-point Padé interpolation technique. This is done by coarse sampling of the frequency domain of interest into a set of frequency points which may or may not be equally spaced. The solution to the function $u(\sigma_j)$ and its $N_d$ derivatives with respect to $\sigma$ are evaluated at each frequency point of interest $\sigma_j$ at first, where

$$N_d = \left\lceil \frac{L+M+1}{n} \right\rceil - 1$$  \hspace{1cm} (2.9)

$\lceil \cdot \rceil$ denotes the ceiling function and $n$ denotes the total number of frequency points of interest. The single-point Padé equation (2.5) when applied to each frequency point of interest $\sigma_j$ takes the form as given by equation (2.10).

$$\sum_{r=0}^{k} \left( \sum_{l=r}^{M} \frac{M!}{(l-r)!} q_l \left( \sigma_0 + \Delta \sigma_j \right)^{l-r} \right) - \sum_{l=k}^{L} \frac{L!}{(l-k)!} p_l \left( \sigma_0 + \Delta \sigma_j \right)^{l-k} = 0$$  \hspace{1cm} (2.10)

Where $u_j^{[v]}$ denotes the $v^{th}$ derivative of the function $u(\sigma)$ with respect to $\sigma$ at $(\sigma_0 + \Delta \sigma_j)$, with $\Delta \sigma_j = \sigma_j - \sigma_0$ and $k = 0, 1, \ldots, N_d$. The associated matrix of equations for each value of $\sigma_j$ has been given by equation (2.11).

$$[A_j][x] = [b_j]$$  \hspace{1cm} (2.11)

Where $[A_j] \in C^{(N_d)\times(L+M+1)}$, $[b_j] \in C^{(N_d)\times1}$, $j = 1, 2, \ldots, n$
It is claimed that the coefficients of polynomials $P$ and $Q$ that define the rational function given in equation (2.1) can be determined by solving the system of equations given in equation (2.12).

$$\begin{bmatrix}
A_1 \\
. \\
. \\
A_j \\
. \\
. \\
A_n
\end{bmatrix} \begin{bmatrix}
b_1 \\
. \\
. \\
b_j \\
. \\
. \\
b_n
\end{bmatrix} = \sum_{q=0}^{L} \binom{(b-1)}{(a-1-q)} (h_j)^{(b-1)-(a-1-q)}$$  

(2.12)

The equation (2.12) is symbolically solved for the purpose of developing algorithms for computation. The elements of the matrices $[A]$ and $[b]$ are symbolically evaluated at $\sigma_j$ and are given by equations (2.13), (2.14) and (2.15).

$$[A]_{ab} = -\sum_{b=0}^{L} \frac{(b-1)!}{((b-1)-(a-1))!} \binom{(b-1)}{(a-1-q)} (h_j)^{(b-1)-(a-1-q)}$$  

(2.13)

for $a = 1, \ldots, N_d + 1$ and $b = 1, \ldots, L + 1$,

and,

$$[A]_{ab} = \sum_{r=0}^{a-1} \frac{(a-1)!}{((a-1)-(r))!} \binom{(a-1-(r))}{(b-(L+1)-(r))!} \binom{(b-(L+1)-(r))}{(b-(L+1)-(r))!} (h_j)^{(b-(L+1)-(r))}$$  

(2.14)

for $a = 1, \ldots, N_d + 1$ and $b = L + 2, \ldots, L + M + 1$, while $(b-(L+1)) \geq r$.

$$b_a = -(u_j)^{(a-1)}$$  

(2.15)

Where,

$u_j^{(k)}$ is the value of $k^{th}$ differential of the solution function $u(\sigma)$ at $\sigma_j$,

$h_j^{(k)}$ is the value of $k^{th}$ power of $h_j = \sigma_j - \sigma_0$.

The Padé approximant function $u_{\text{pade}}(\sigma)$ of the solution function is claimed to be the rational function and is given by equation (2.16).
\[ u_{\text{pade}}(\sigma) = \frac{P(\sigma)}{Q(\sigma)} \]  

(2.16)

Hence the function \( u_{\text{pade}}(\sigma) \) is claimed to produce an approximate solution to the function \( u(\sigma) \) in the frequency domain of interest \([\omega_L, \omega_R]\).
Chapter 3

Analysis of a single degree of freedom non-linear vibrational system

This chapter is focused on the analysis of the motion of a single degree of freedom non-linear vibrational system. The procedure to find the solution to the vibrational motion of the model, and the feasibility along with the essential supplementary computations to implement an interpolation technique to match the analytical solution have been discussed in this chapter. A reference model [3] is considered to evaluate the feasibility of the application of interpolation technique.

3.1 Model and governing equation of motion

![Diagram of equivalent single degree of freedom system of blade to damper model](image)

Figure 3-1. Equivalent single degree of freedom system of blade to damper model. Source [3].
Figure 3-2. Nondimensionalized single degree of freedom system of blade to damper model with positive damping coefficient. Source [3].

The model depicted in Fig 3-1, represents a single mode of a blade with a massless, flexible, blade-to-ground friction damper. The nondimensionalized model has also been developed in [3] and has been shown in Fig 3-2. The governing equation of motion for the model illustrated in Fig 3-2 has been given by equation (3.1).

\[
\frac{d^2x}{dt^2} + 2\xi \frac{dx}{dt} + x = \varepsilon y + \varepsilon f_0 \cos(\omega t)
\]  

(3.1)

The non-linear characteristic of the model has been incorporated in the expression \( y(x) \). According to [3], the system is linear until the slip occurs. The slip occurs at the friction damper when the amplitude of response of the mass exceeds value one. When the slip occurs, the system becomes piecewise linear and the value of \( y \) can be written as a function of \( x \) as discussed in section 3.2.
3.2 Analytical solution

As the steady-state response has been assumed to be approximately harmonic, the solution has been taken as given by equation (3.2).

\[ x = A \cos(\omega t + \phi) \]  

(3.2)

Since the response is assumed to be harmonic, the non-linear term \( \varepsilon y \) can be expanded using Fourier series. According to [3] only the first terms are considered which are given by equation (3.3).

\[ \varepsilon y = a \cos(\omega t + \phi) + b \sin(\omega t + \phi) \]  

(3.3)

Where,

\[ a = \frac{\varepsilon A}{\pi} \left( \pi - \theta_c + \frac{\sin(2\theta_c)}{2} \right) \]  

(3.4)

\[ b = \frac{4 \varepsilon}{\pi} \left( 1 - \frac{1}{A} \right) \]  

(3.5)

\[ \theta_c = \cos^{-1} \left( 1 - \frac{2}{A} \right) \]  

(3.6)

The non-linear term \( \varepsilon y \) can be rewritten as:

\[ \varepsilon y = a \frac{x}{A} + b \left( \frac{\omega A}{-1} \right) \frac{dx}{dt} \]  

(3.7)

Substituting equation (3.7) into equation (3.1) we get:

\[ \frac{d^2 x}{dt^2} + \left[ 2 \xi + \frac{b}{\omega A} \right] \frac{dx}{dt} + \left[ 1 - \frac{a}{A} \right] x = \varepsilon f_0 \cos(\omega t) \]  

(3.8)

As claimed earlier, the system has a linear response before slip occurs or in other words when the value of \( y = 0 \). The system is solved for its response by substituting equation (3.2) into equation
(3.1) and comparing the coefficients of $\cos(\omega t)$ and $\sin(\omega t)$ which led to equations (3.9) and (3.10).

\[(1 - \omega^2)(A\cos \phi) - (2\xi \omega)(A \sin \phi) = e f_0\]  \hspace{1cm} (3.9)

\[(1 - \omega^2)(A \sin \phi) + (2\xi \omega)(A \cos \phi) = 0\]  \hspace{1cm} (3.10)

Further equations (3.9) and (3.10) are solved to find values for $A \cos \phi$ and $A \sin \phi$. The value of $A$ is obtained from equation (3.11).

\[A = \sqrt{(A \cos \phi)^2 + (A \sin \phi)^2}\]  \hspace{1cm} (3.11)

If the value of $A \leq 1$, then the response is realized to be linear and the value for $A$ is taken as obtained from equation (3.11). On the other hand, if the value of $A > 1$ then it is realized that slip has occurred and the response should have included the non-linear aspects. The value of $A$ is calculated by substituting equation (3.2) into equation (3.8) and comparing the coefficients of $\cos(\omega t)$ and $\sin(\omega t)$ which led to equations (3.12) and (3.13).

\[(K - \omega^2)(A \cos \phi) - (\omega C)(A \sin \phi) = e f_0\]  \hspace{1cm} (3.12)

\[(K - \omega^2)(A \sin \phi) + (\omega C)(A \cos \phi) = 0\]  \hspace{1cm} (3.13)

Where,

\[K = 1 - \frac{a}{A}\]  \hspace{1cm} (3.14)

\[C = 2\xi + \frac{b}{\omega A}\]  \hspace{1cm} (3.15)

Further equations (3.12) and (3.13) are solved to find values for $A \cos \phi$ and $A \sin \phi$. The revised value of $A$ is obtained from equation (3.11).
3.3 Solution by Padé’s interpolation method

From chapter-2 it is understood that \( N_d \) is the number of successive derivatives of the solution function which in this case is the amplitude function \( A(\omega) \), with respect to \( \omega \) which are required to determine the Padé’s approximant. The method to find the derivatives has been discussed in this section. From reference [1], it can be claimed that for a linear vibration model the differentials of \( (A \cos \phi) \) and \( (A \sin \phi) \) could be obtained by recursively differentiating the equations (3.9) and (3.10) with respect to \( \omega \), which led to equations (3.16) and (3.17) respectively, while the amplitude function \( A(\omega) \) is a function of \( \omega \).

\[
\sum_{r=0}^{n} C_r \left( \frac{d^r (A \cos \phi)}{d\omega^r} \right) \left( \frac{d^{n-r} (1 - \omega^2)}{d\omega^{n-r}} \right) - \sum_{r=0}^{n} C_r \left( \frac{d^r (A \sin \phi)}{d\omega^r} \right) \left( \frac{d^{n-r} (2\xi \omega)}{d\omega^{n-r}} \right) = \left( \frac{d^n (\in f_0)}{d\omega^n} \right) \]
(3.16)

\[
\sum_{r=0}^{n} C_r \left( \frac{d^r (A \sin \phi)}{d\omega^r} \right) \left( \frac{d^{n-r} (1 - \omega^2)}{d\omega^{n-r}} \right) + \sum_{r=0}^{n} C_r \left( \frac{d^r (A \cos \phi)}{d\omega^r} \right) \left( \frac{d^{n-r} (2\xi \omega)}{d\omega^{n-r}} \right) = 0 \]
(3.17)

On the other hand, for the non-linear region of the vibration model the differentials of \( (A \cos \phi) \) and \( (A \sin \phi) \) could be obtained by recursively differentiating the equations (3.12) and (3.13) with respect to \( \omega \), which led to equations (3.18) and (3.19) respectively.

\[
\sum_{r=0}^{n} C_r \left( \frac{d^r (A \cos \phi)}{d\omega^r} \right) \left( \frac{d^{n-r} (K - \omega^2)}{d\omega^{n-r}} \right) - \sum_{r=0}^{n} C_r \left( \frac{d^r (A \sin \phi)}{d\omega^r} \right) \left( \frac{d^{n-r} (\omega C)}{d\omega^{n-r}} \right) = \left( \frac{d^n (\in f_0)}{d\omega^n} \right) \]
(3.18)

\[
\sum_{r=0}^{n} C_r \left( \frac{d^r (A \sin \phi)}{d\omega^r} \right) \left( \frac{d^{n-r} (K - \omega^2)}{d\omega^{n-r}} \right) + \sum_{r=0}^{n} C_r \left( \frac{d^r (A \cos \phi)}{d\omega^r} \right) \left( \frac{d^{n-r} (\omega C)}{d\omega^{n-r}} \right) = 0 \]
(3.19)
It is observed that the nonlinear stiffness expression $K(A)$ is a function of the amplitude and the amplitude function $A(\omega)$ is a function of $\omega$. Hence the derivative of the nonlinear stiffness expression $K(A)$ is obtained by performing the following procedure.

The first derivative of the stiffness expression $K(A)$ is given by the equation (3.20).

$$\left( \frac{dK(A)}{d\omega} \right) = \left( \frac{\partial K(A)}{\partial A} \right) \left( \frac{dA}{d\omega} \right)$$

The $n^{th}$ derivative of the stiffness expression $K(A)$ is given by the equation (3.21).

$$\left( \frac{d^n K(A)}{d\omega^n} \right) = \left( \frac{d^{n-1} \left( \frac{\partial K(A)}{\partial A} \right) \left( \frac{dA}{d\omega} \right)}{d\omega^{n-1}} \right) = \sum_{r=0}^{n-1} (-1)^r C_n^r \left( \frac{d^r \left( \frac{\partial K(A)}{\partial A} \right)}{d\omega^r} \right) \left( \frac{d^{n-r} A}{d\omega^{n-r}} \right)$$

$$= (-1)^{n-1} C_n^0 \left( \frac{\partial K(A)}{\partial A} \right) \left( \frac{d^n A}{d\omega^n} \right) + \sum_{r=1}^{n-1} (-1)^r C_n^r \left( \frac{d^r \left( \frac{\partial K(A)}{\partial A} \right)}{d\omega^r} \right) \left( \frac{d^{n-r} A}{d\omega^{n-r}} \right)$$

Where $\left( \frac{d^r \left( \frac{\partial K(A)}{\partial A} \right)}{d\omega^r} \right)$ is obtained by recursively evaluating the equation (3.22) for $j = 1, 2, \ldots, r$.

$$\left( \frac{d^j \left( \frac{\partial K(A)}{\partial A} \right)}{d\omega^j} \right) = \left( \frac{d}{d\omega} \right)^{j-1} \left( \frac{\partial}{\partial A} \right) \left( \frac{dA}{d\omega} \right)$$

(3.22)

Similarly, it is observed that the nonlinear damping expression $C(A, \omega)$ is a function of the amplitude $A$ and the excitation frequency $\omega$, while the amplitude function $A(\omega)$ is a function of
$$\omega$$. Hence the derivative of the nonlinear damping expression \( C(A, \omega) \) is obtained by performing the following procedure.

The first derivative of the damping expression \( C(A, \omega) \) is given by the equation (3.23).

$$\left( \frac{dC(A, \omega)}{d\omega} \right) = \left( \frac{\partial C(A, \omega)}{\partial A} \right) \left( \frac{dA}{d\omega} \right) + \left( \frac{\partial C(A, \omega)}{\partial \omega} \right)$$  \hspace{1cm} (3.23)

The \( n^{th} \) derivative of the damping expression \( C(A, \omega) \) is given by the equation (3.24).

$$\left( \frac{d^n C(A, \omega)}{d\omega^n} \right) = \left[ \frac{d^{n-1} \left( \frac{\partial C(A, \omega)}{\partial A} \right) \left( \frac{dA}{d\omega} \right) + \left( \frac{\partial C(A, \omega)}{\partial \omega} \right) }{d\omega^{n-1}} \right]$$

$$= \sum_{r=0}^{n-1} C_r \left( \frac{d^r \left( \frac{\partial C(A, \omega)}{\partial A} \right)}{d\omega^r} \right) \left( \frac{d^{n-r} A}{d\omega^{n-r}} \right) + \left( \frac{\partial C(A, \omega)}{\partial \omega} \right)$$ \hspace{1cm} (3.24)

$$= C_0 \left( \frac{\partial C(A, \omega)}{\partial A} \right) \left( \frac{d^n A}{d\omega^n} \right) + \sum_{r=1}^{n-1} C_r \left( \frac{d^r \left( \frac{\partial C(A, \omega)}{\partial A} \right)}{d\omega^r} \right) \left( \frac{d^{n-r} A}{d\omega^{n-r}} \right) + \left( \frac{\partial C(A, \omega)}{\partial \omega} \right)$$

Where \( \left( \frac{d^r \left( \frac{\partial C(A, \omega)}{\partial A} \right)}{d\omega^r} \right) \) is obtained by recursively evaluating the equation (3.25) for \( j = 1, 2, \ldots, r \).

$$\left( \frac{d^j \left( \frac{\partial C(A, \omega)}{\partial A} \right)}{d\omega^j} \right) = \left( \frac{d^{j-1} \left( \frac{\partial C(A, \omega)}{\partial A} \right)}{d\omega^{j-1}} \right) \left( \frac{dA}{d\omega} \right) + \left( \frac{d^{j-1} \left( \frac{\partial C(A, \omega)}{\partial \omega} \right)}{d\omega^{j-1}} \right) \left( \frac{d\omega}{d\omega} \right)$$ \hspace{1cm} (3.25)
And where \( \left( \frac{d^j \partial C(A, \omega)}{\partial \omega} \right) \) is obtained by recursively evaluating the equation (3.26) for \( j = 1, 2, \ldots, r \).

\[
\left( \frac{d^j \partial C(A, \omega)}{\partial \omega} \right) = \left( \frac{d^{j-1} \partial C(A, \omega)}{\partial \omega^{j-1}} \right) \left( \frac{d A}{d \omega} \right) + \left( \frac{d^{j-1} \partial C(A, \omega)}{\partial \omega^{j-1}} \right) \left( \frac{d A}{d \omega} \right)
\]

(3.26)

It was claimed that \( \left( \frac{d^n A}{d \omega^n} \right) \) can be written as given by equation (3.27) by differentiating equation (3.11) \( n \) number of times with respect to \( \omega \).

\[
\frac{d^n A}{d \omega^n} = \frac{d^n \sqrt{(A \cos \phi)^2 + (A \sin \phi)^2}}{d \omega^n}
\]

(3.27)

Equations (3.18) and (3.19) are solved for \( \left( \frac{d^n (A \cos \phi)}{d \omega^n} \right) \) and \( \left( \frac{d^n (A \sin \phi)}{d \omega^n} \right) \) by replacing equations (3.21), (3.24) and (3.27) for \( \left( \frac{d^n K(A)}{d \omega^n} \right) \), \( \left( \frac{d^n C(A, \omega)}{d \omega^n} \right) \) and \( \left( \frac{d^n A}{d \omega^n} \right) \), respectively and thereby substituting the values of \( \left( \frac{d^r (A \cos \phi)}{d \omega^r} \right) \) and \( \left( \frac{d^r (A \sin \phi)}{d \omega^r} \right) \) for \( r = 0, 1, \ldots, (n - 1) \).

Having obtained the values for \( \left( \frac{d^n (A \cos \phi)}{d \omega^n} \right) \) and \( \left( \frac{d^n (A \sin \phi)}{d \omega^n} \right) \) for every case of linear and nonlinear region of interest, equation (3.27) is substituted for the values of \( \left( \frac{d^r (A \cos \phi)}{d \omega^r} \right) \) and
\[
\left(\frac{d'(A\sin \phi)}{d\omega'}\right) \text{ for } r = 0,1,...,n \text{ to obtain the value of } \left(\frac{d^n A}{d\omega^n}\right) \text{ for every case of linear and nonlinear region of interest. This procedure is repeated for } n = 1,2,...,N_d \text{ to obtain the values of } N_d \text{ number of successive derivatives of amplitude function } A(\omega) \text{ with respect to } \omega, \text{ that are required to determine the Padé’s approximant.}
\]

By replacing the solution parameter \( \sigma \) and the solution function \( u(\sigma) \) with the excitation frequency \( \omega \) and the amplitude function \( A(\omega) \), respectively over a frequency domain \([\omega_L, \omega_R]\) with defined interested coarse points of frequency \( \omega_j \in [\omega_L, \omega_R] \) for \( j = 1,2,...,n \) in the procedure followed in section 2.2 of chapter 2, it is claimed that the Padé’s approximant function developed \( u_{pade}(\sigma) \) in section 2.2 serves as an approximate solution function to the amplitude function \( A(\omega) \), denoted by \( A_{pade}(\omega) \).

### 3.4 Comparison of results

The values for the parameters of the problem of interest as taken from reference [3] are as given below.

\( \varepsilon = 0.1 \)
\( f_0 = 1 \)
\( \xi = 0.001 \)
\( m = 1 \)

The values for the parameters for the evaluation of the Padé’s approximant are as given below.

\( L = 29 \)
\( M = 30 \)
\( n = 30 \)
A MATLAB code has been developed to evaluate the analytical solution and the Padé’s approximant solution to the amplitude function \( A(\omega) \) which has been attached in Appendix-A. The results obtained from the computation have been compared and contrasted in Fig 3-3.

The ‘True values (Training)’ represent the exact values of the amplitude function \( A(\omega) \) at the selected interpolation points \( \omega_j \) in the domain \( [\omega_L, \omega_R] \) for \( j = 1,2,\ldots,n \) where \( \omega_L = 0.93 \) rad/s and \( \omega_R = 1.0 \) rad/s. The ‘True values (Test)’ represent the exact values of the amplitude function \( A(\omega) \) at the trisected points of each interval \( [\omega_i, \omega_{i+1}] \) in the domain \( [\omega_L, \omega_R] \) for \( i = 1,2,\ldots,n \). The ‘True values (Training)’ and their derivatives have been used for the interpolation to give the Padé’s approximant amplitude function \( A_{\text{pade}}(\omega) \). The ‘Padé approximation values (Training)’ represent the values of the Padé’s approximant amplitude function \( A_{\text{pade}}(\omega) \) at the same points as the domain of the ‘True values (Training)’. The ‘Padé approximation values (Test)’ represent the values of the Padé’s approximant amplitude function \( A_{\text{pade}}(\omega) \) at the same points as the domain of the ‘True values (Test)’. 
Figure 3-3. Comparison of values of amplitude function $A(\omega)$ and Padé’s approximant amplitude function $A_{\text{Pade}}(\omega)$.

The error between the analytical solution of amplitude function $A(\omega)$ and Padé’s approximant amplitude function $A_{\text{Pade}}(\omega)$ values has been calculated as given by equation (3.28) for the Test points of frequency and are depicted in Fig 3-4.

\[ error = A(\omega) - A_{\text{Pade}}(\omega) \]  

(3.28)
Figure 3-4. Error between analytical solution of amplitude function $A(\omega)$ and Padé’s approximant amplitude function $A_{\text{Pade}}(\omega)$ values.

It is observed that the values of the Padé’s approximant amplitude function $A_{\text{Pade}}(\omega)$ matches well with that of the analytical solution of the amplitude function $A(\omega)$ for the ‘Test’ points of frequency. However, a comparatively higher value of $error \approx -0.05$ is observed within the frequency region $[0.95,0.96]$ as there seemed to be a sudden increase in the amplitude function $A(\omega)$ values in the same frequency region which would have caused severe variations in the derivative values. A more accurate approximant amplitude function $A_{\text{Pade}}(\omega)$ is obtained by increasing the number of selected interpolation points in the frequency region of interest.

The accuracy of the results can further be improved by dividing the entire frequency region into more number of sample spaces, which would lead to the evaluation of a new Padé’s approximant for each sample space. As increasing the number of sample spaces improved
accuracy, the entire frequency domain of interest is expanded as given $[\omega_L, \omega_R]=[0.00,1.25]$. Since the non-linear aspects of the problem which leads to higher variations in the derivative values of the amplitude function, is concentrated at and around the damped natural frequency whose value is given by the equation (3.29), the sample spaces have been chosen as,

Sample space – 1: $[0.00,0.95] \text{ rad/s}$ with $L=9, M=10, n=10$.

Sample space – 2: $[0.95,0.965] \text{ rad/s}$ with $L=29, M=30, n=20$.

Sample space – 3: $[0.965,1.25] \text{ rad/s}$ with $L=29, M=30, n=30$.

$$\omega_d = \omega_n \sqrt{1 - \xi^2}$$

(3.29)

Figure 3-5. Comparison of values of amplitude function $A(\omega)$ and Padé’s approximant amplitude function $A_{\text{Padé}}(\omega)$ with increased sample spaces.

The analytical solution and the Padé’s approximant solution to the amplitude function $A(\omega)$ with the new sample spaces have been compared in Fig 3-5. The comparison between the
analytical solution and the Padé’s approximant solution of the amplitude function $A(\omega)$ corresponding to the second sample space has been isolated and distinctly depicted in Fig 3-6.

![Nondimensionalized Amplitude function value](image)

**Figure 3-6.** Comparison of values of amplitude function $A(\omega)$ and Padé’s approximant amplitude function $A_{pade}(\omega)$ of second sample space.

The values of error between the analytical solution and the Padé’s approximant solution of the amplitude function $A(\omega)$ of the ‘Test’ points of frequency have been depicted in Fig 3-7. It is observed that the maximum value of $error = -0.00065$ has decreased by about 98%. In addition, the mean value of the error function has come down drastically. On the whole, the Padé’s approximant amplitude function $A_{pade}(\omega)$ has proved to be a really good estimation of the analytical amplitude function $A(\omega)$. 
Figure 3-7. Error between analytical solution of amplitude function $A(\omega)$ and Padé’s approximant amplitude function $A_{\text{Padé}}(\omega)$ values.
Chapter 4

General approach to analysis of a multi-degree of freedom non-linear vibrational system

This chapter is focused on the analysis of the motion of a multi degree of freedom non-linear vibrational system with impedance matrix $Z$ of the form given by equation (1.2). The procedure to find the solution to the vibrational motion of the model, and the feasibility along with the essential supplementary computations to implement an interpolation technique to match the analytical solution have been discussed in this chapter. In addition, this chapter has focused on the development of algorithms that are less computationally exhaustive in nature, to find out the derivatives of the amplitude function values at interested excitation frequencies for a multi-degree of freedom non-linear vibrational system.

4.1 Model and governing equation of motion

The equations of motion of any linear multi-degree of freedom vibrational system can be written in the general form as given in equation (4.1).

$$
[M] \frac{d^2 \{ \bar{X} \}}{dt^2} + [C] \frac{d \{ \bar{X} \}}{dt} + [K] \{ \bar{X} \} = \{ F \}
$$

(4.1)

Where

$\{ \bar{X} \}$ is the vector of motion in all degrees of freedom of the vibrational model

$[M]$ is the mass matrix of the vibrational model

$[C]$ is the damping matrix of the vibrational model
$[K]$ is the stiffness matrix of the vibrational model

$\{F\}$ is the force vector of the vibrational model

### 4.2 Analytical solution

Since $\{F\}$ is assumed to be harmonic in nature it is claimed that the steady-state response of each degree of freedom of motion is harmonic and hence the solution is taken as given in equation (4.2).

$$x_i = A_i \cos(\omega t + \phi_i)$$  \hspace{1cm} (4.2)

Where,

$x_i$ is the $i^{th}$ element of $\{X\}$ vector

$A_i$ is the assumed amplitude of motion of $i^{th}$ degree of freedom

$\phi_i$ is the assumed phase angle of motion of $i^{th}$ degree of freedom

As the total degrees of freedom of the vibrational system is assumed to be $n$, the equation (4.1) upon substitution of equation (4.2) is written in the form as depicted in equation (4.3).

$$\begin{bmatrix}
\begin{bmatrix}
A_i \cos(\omega t + \phi_i)
\end{bmatrix} & \begin{bmatrix}
A_i \sin(\omega t + \phi_i)
\end{bmatrix}
\end{bmatrix} - \omega^2 \begin{bmatrix}
\ddots
\end{bmatrix} + \begin{bmatrix}
\ddots
\end{bmatrix} + \begin{bmatrix}
\begin{bmatrix}
A_i \cos(\omega t + \phi_i)
\end{bmatrix} & \begin{bmatrix}
A_i \sin(\omega t + \phi_i)
\end{bmatrix}
\end{bmatrix} = \{F\}$$  \hspace{1cm} (4.3)

Where

$$\{F\} = \begin{bmatrix}
F_1 \cos(\omega t)
\vdots
F_n \cos(\omega t)
\end{bmatrix}$$  \hspace{1cm} (4.4)
The equation (4.3) is expanded and compared for the terms of \( \cos(\omega t) \) and \( \sin(\omega t) \) which led to equations (4.5) and (4.6) respectively.

\[
\begin{bmatrix}
-\omega^2[M] + [K] & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
A_n \cos(\phi_n) & A_n \sin(\phi_n) & \cdot \\
\end{bmatrix}
\begin{bmatrix}
A_1 \cos(\phi_1) \\
\cdot \\
A_n \cos(\phi_n) \\
\end{bmatrix}
+ \begin{bmatrix}
-\omega[C] \\
\cdot \\
C_n \\
\end{bmatrix}
\begin{bmatrix}
A_1 \sin(\phi_1) \\
\cdot \\
A_n \sin(\phi_n) \\
\end{bmatrix}
= \begin{bmatrix}
F_1 \\
\cdot \\
F_n \\
\end{bmatrix}
\] (4.5)

\[
\begin{bmatrix}
-\omega^2[M] + [K] & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
-\omega[C] & \cdot & \cdot \\
\end{bmatrix}
\begin{bmatrix}
A_1 \cos(\phi_1) \\
\cdot \\
A_n \cos(\phi_n) \\
\end{bmatrix}
+ \begin{bmatrix}
-\omega[C] \\
\cdot \\
C_n \\
\end{bmatrix}
\begin{bmatrix}
A_1 \sin(\phi_1) \\
\cdot \\
A_n \sin(\phi_n) \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
\cdot \\
0 \\
\end{bmatrix}
\] (4.6)

Equations (4.5) and (4.6) are combined into equation (4.7).

\[
\begin{bmatrix}
-\omega^2[M] + [K] & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
-\omega[C] & \cdot & \cdot \\
\end{bmatrix}
\begin{bmatrix}
A_1 \cos(\phi_1) \\
\cdot \\
A_n \cos(\phi_n) \\
\end{bmatrix}
+ \begin{bmatrix}
-\omega[C] \\
\cdot \\
C_n \\
\end{bmatrix}
\begin{bmatrix}
A_1 \sin(\phi_1) \\
\cdot \\
A_n \sin(\phi_n) \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
\cdot \\
0 \\
\end{bmatrix}
\] (4.7)

Where,

\[
[R]_{2n \times 2n} \{A\}_{2n \times 1} = \{Q\}_{2n \times 1}
\] (4.8)

\[
\{A\}_{n \times 1} =
\begin{bmatrix}
A_1 \cos(\phi_1) \\
\cdot \\
A_n \cos(\phi_n) \\
A_1 \sin(\phi_1) \\
\cdot \\
A_n \sin(\phi_n) \\
\end{bmatrix}_{n \times 1}
\] (4.9)
As claimed earlier, the system has a linear response before slip occurs or in other words when the value of \( y = 0 \). The system is solved for its response in the linear region by solving equation (4.7) which is given by equation (4.11).

\[
\{A\}^*_{2n_2} = [R]^{-1} \{Q\}^*_{2n_2}
\]  

(4.11)

The values of \( \{A\}^*_{2n_2} \) are utilized to find the values of magnitude of the amplitude function \( \{A_m\}_{n_1} \) where,

\[
\{A_m\}_{n_1} = \begin{bmatrix}
    A_1 \\
    \cdot \\
    \cdot \\
    A_n
\end{bmatrix}_{n_1}
\]

(4.12)

And each value of \( A_i, i = 1, 2, \ldots, n \) of equation (4.12) is calculated from equation (4.13).

\[
A_i = \sqrt{(A_i \cos \phi_i)^2 + (A_i \sin \phi_i)^2}
\]

(4.13)

However, if the value of the amplitude function \( A_i \) of the linear response at \( i^{th} \) degree of freedom where the frictionally damped spring of stiffness value \( K_d \) is connected, exceeded the limiting value of friction then it is realized that a slip has occurred and the vibrational motion of \( i^{th} \) degree of freedom has entered nonlinear region. As discussed in section 3.2 of Chapter 3, the
nonlinear aspects were incorporated into the equations of motion of the vibrational model by adding terms \( \frac{b}{\omega A_i} \) and \( -\frac{a}{A_i} \) to the damping and stiffness coefficients of equation of motion of \( i^{th} \) degree of freedom respectively, where \( a \) and \( b \) have been calculated as discussed in Appendix-B and are given by equations (4.14) and (4.15). The solution to the equation (4.7) is evaluated by solving for its \( 2n \) coupled equations simultaneously.

\[
a = \frac{K_d}{\pi}
\left(A_\pi - A_\theta + (A_i - 2A_{i0})\sin(\theta_d)\right) \tag{4.14}
\]

\[
b = \frac{4K_d A_{i0}}{\pi}
\left(1 - \frac{A_{i0}}{A_i}\right) \tag{4.15}
\]

\[
\theta_d = \cos^{-1}
\left(1 - \frac{2A_{i0}}{A_i}\right) \tag{4.16}
\]

\( A_{i0} \) is the limiting value of \( A_i \) when slip occurs.

The revised values of \( \{A_m\}_{nx1} \) are evaluated from the newly evaluated set of values of \( \{A\}_{2nx1} \).

### 4.3 Solution by Padé’s interpolation method

From chapter-2 it is understood that \( N_d \) is the number of successive derivatives of amplitude function \( A_i(\omega) \) with respect to \( \omega \) which are required to determine the Padé’s approximant. The method to find the differentials for a multi degree of freedom system have been discussed in this section.

To find the value of \( j^{th} \) derivative of amplitude function \( A_i(\omega) \) with respect to \( \omega \), the equation (4.7) is differentiated \( j \) times as given in equation (4.17).
$$[[R][A]]^{[j]} = [[Q]]^{[j]}$$  \hspace{1cm} (4.17)

Where $[ ]^{[j]}$ denotes $j^{th}$ differential of $[ ]$ with respect to $\omega$.

Equation (4.17) is further expanded into equation (4.18) with the help of chain rule of derivation.

$$\sum_{r=0}^{j} i C_r [R]^{[r]} \{A\}^{[j-r]} = \{Q\}^{[j]}$$  \hspace{1cm} (4.18)

Where $\{ \}^{[j]}$ denotes $j^{th}$ differential of $\{ \}$ with respect to $\omega$.

For the linear region, the value of $\{A\}^{[j]}$ is evaluated as depicted in equation (4.19) which is obtained by rearranging terms of equation (4.18).

$$\{A\}^{[j]} = [R]^3 \left[ \{Q\}^{[j]} - \sum_{r=1}^{j} i C_r [R]^{[r]} \{A\}^{[j-r]} \right]$$  \hspace{1cm} (4.19)

The value of $\{A\}^{[n]}$ is calculated by recursively evaluating the equation (4.19) for $j = 1,2,\ldots,n$ such that at each $j^{th}$ evaluation the right hand side of equation (4.19) is calculated from the values evaluated at $(j-1)^{th}$ evaluation, except for the evaluation at $j = 1$, where the value of $\{A\}^{[0]}$ is taken as the value of $\{A\}$.

Since $[R]$ is a function of the excitation frequency $\omega$ and the amplitude function $A(\omega)$ in the nonlinear region, equation (4.18) is evaluated as discussed below. At first equation (4.18) is expanded to isolate the first and last terms of the left-hand side as given in equation (4.20).

$$i C_0 [R][A]^{[j]} + \sum_{r=1}^{j} i C_r [R]^{[r]} \{A\}^{[j-r]} + i C_j [R]^{[j]} \{A\} = \{Q\}^{[j]}$$  \hspace{1cm} (4.20)

The purpose of this expansion is to isolate the coefficient of $\{A\}^{[j]}$. 

As \([R]\) is claimed earlier to be a function of the excitation frequency \(\omega\) and the amplitude function \(A(\omega)\), \([R]^{(j)}\) is expanded as given in equation (4.21).

\[
[R]^{(j)} = \left[ [R]^{(j)} \right]^{j-1} = \left[ \frac{\partial [R]}{\partial \{A\}} \right] [A]^{(j)} + \left[ \frac{\partial [R]}{\partial \omega} \right]^{j-1}
\]

\[
= \sum_{i=0}^{i-1} C_{i} \left[ \frac{\partial [R]}{\partial \{A\}} \right]^{[i]} [A]^{(j-i)} + \left[ \frac{\partial [R]}{\partial \omega} \right]^{j-1}
\]

\[
= \left[ C_{0} \frac{\partial [R]}{\partial \{A\}} [A]^{(j)} + \sum_{i=1}^{i-1} C_{i} \left[ \frac{\partial [R]}{\partial \{A\}} \right]^{[i]} [A]^{(j-i)} + \left[ \frac{\partial [R]}{\partial \omega} \right]^{j-1} \right]
\]  \(4.21\)

where,

\[
\frac{\partial [R]}{\partial \{A\}} = \left[ \frac{\partial [R]}{\partial A_{1}}, \ldots, \frac{\partial [R]}{\partial A_{2n}} \right]
\]  \(4.22\)

\[
[A] = \begin{bmatrix}
[A_{1} \cos(\phi_{1})]^{*} \text{Id}(n) \\
\vdots \\
[A_{n} \cos(\phi_{n})]^{*} \text{Id}(n) \\
[A_{1} \sin(\phi_{1})]^{*} \text{Id}(n) \\
\vdots \\
[A_{n} \sin(\phi_{n})]^{*} \text{Id}(n)
\end{bmatrix}_{n \times n}
\]

where \(\text{Id}(u)\) represents the identity matrix of size \(uxu\) and \(a.*[A]\) represents element-wise scalar multiplication of \(a\) with the matrix \([A]\).  \(4.23\)

Equation (4.20) upon substitution with equation (4.21) is written as given by equation (4.24).

\[
\sum_{i=0}^{i-1} C_{i} [R]^{[i]} + \sum_{i=1}^{i-1} C_{i} \left[ \frac{\partial [R]}{\partial \{A\}} \right]^{[i]} [A]^{(j-i)} + \left[ \frac{\partial [R]}{\partial \omega} \right]^{j-1}
\]

\[
= C_{0} \left[ \frac{\partial [R]}{\partial \{A\}} \right] [A]^{(j)} + \sum_{i=1}^{i-1} C_{i} \left[ \frac{\partial [R]}{\partial \{A\}} \right]^{[i]} [A]^{(j-i)} + \left[ \frac{\partial [R]}{\partial \omega} \right]^{j-1} \{A\} = \{Q\}^{(j)}
\]  \(4.24\)

Upon further simplification, equation (4.24) is written as given by equation (4.25).
\[ iC_0 [R] [A]^{[j]} + iC_j \sum_{r=1}^{j-1} [C_r [R]^{[j-r]} A]^{[j-r]} + \sum_{r=1}^{j-1} [C_r [R]^{[j-r]} A]^{[j-r]} \]

\[ + C_j \sum_{r=1}^{j-1} C_r [\frac{\partial [R]}{\partial [A]}]^{[r]} [A]^{[j-r]} + \left[ \frac{\partial [R]}{\partial \omega} \right]^{[j]} [A] = \{Q\}^{[j]} \]

(4.25)

The second term of the left-hand side of the equation (4.25) is written as given by equation (4.26) through a series of simplification as shown below.

\[ \frac{\partial [R]}{\partial [A]} [\frac{\partial [R]}{\partial [A]}]^{[j]} + \left[ \frac{\partial [R]}{\partial \omega} \right]^{[j]} [A] = \{Q\}^{[j]} \]

(4.26)

where,
\[ [T] = \left[ \frac{\partial [R]}{\partial A_1} \ldots \frac{\partial [R]}{\partial A_n} \right] \]  

Equation (4.28) is obtained by substituting equation (4.27) into equation (4.25).

\[ \{A\}^{[j]} = [R] + [T]^{-1} \left[ \{Q\}^{[j]} - \sum_{r=1}^{j-1} j C_r [R]^{[r]} \{A\}^{[j-r]} \right] 
- i C_j \left[ \sum_{r=1}^{j-1} j C_r \left[ \frac{\partial [R]}{\partial \omega} \right]^{[r]} [\vec{A}]^{[j-r]} + \left[ \frac{\partial [R]}{\partial \omega} \right]^{[j-1]} \right] \{A\} \]  

The value of \( \{A\}^{[k]} \) for the non-linear region is calculated by recursively evaluating the equation (4.28) for \( j = 1, 2, \ldots, k \) such that at each \( j^{th} \) evaluation the right hand side of equation (4.28) is calculated from the values evaluated at \( (j-1)^{th} \) evaluation, except for the evaluation at \( j = 1 \), where the value of \( \{A\}^{[0]} \) is taken as the value of \( \{A\} \).

The value of \( \{A_m\}^{[k]} \) is evaluated by calculating the value of each element \( \{A_m\}^{[k]} \) for \( i = 1, 2, \ldots, n \) which in turn is calculated by differentiating equation (4.13) \( n \) times with respect to \( \omega \) and substituting the values of \( [A_i \cos \phi_i]^{[r]} \) and \( [A_i \sin \phi_i]^{[r]} \) for \( r = 0, 1, \ldots, k \). This procedure is repeated for \( k = 1, 2, \ldots, N_d \) to obtain the values of \( N_d \) number of successive derivatives of the amplitude function \( A_i(\omega) \) with respect to \( \omega \), that are required to determine the Padé’s approximant.

As discussed in section 3.3 of chapter-3, by replacing the solution parameter \( \sigma \) and the solution function \( u(\sigma) \) with the excitation frequency \( \omega \) and each amplitude function \( A_i(\omega) \), respectively over a frequency domain \([\omega_L, \omega_R]\) with defined interested coarse points of frequency
\( \omega_j \in [\omega_L, \omega_R] \) for \( j = 1, 2, ..., n \) in the procedure followed in section 2.2 of chapter-2, it is claimed that the Padé’s approximant function developed \( u_{\text{pade}}(\sigma) \) in section 2.2 serves as an approximate solution function to the corresponding amplitude function \( A_j(\omega) \), denoted by \( A_{j,\text{pade}}(\omega) \). Thereby, the Padé’s approximant function \( A_{j,\text{pade}}(\omega) \) can be used as a surrogate to the corresponding amplitude function \( A_j(\omega) \).
Chapter 5

Analysis of a two degrees of freedom non-linear vibrational system

This chapter is focused on the analysis of the motion of a two degree of freedom non-linear vibrational system with impedance matrix $Z$ of the form given by equation (1.2). The procedure that was discussed in chapter-4 has been employed to find the solution to the two degrees of freedom vibrational motion of the model, and the feasibility along with the essential supplementary computations to implement an interpolation technique to match the analytical solution.

5.1 Model and governing equation of motion

According to [4] the model depicted in Fig 5.1, represents two modes of a blade with a massless, flexible, blade-to-ground friction damper. The governing equations of motion for the model illustrated in Fig 5.1 are given by equations (5.1) and (5.2).

Figure 5-1. Two degrees of freedom system of blade to damper model.
The equations of motion of the depicted two degrees of freedom spring damper vibrational system are as given in equations (5.1) and (5.2).

\[
m_1 \frac{d^2 x_1}{dt^2} + c \frac{dx_1}{dt} + k_1 x_1 = k_1 x_2 + Q \cos(\omega t) \quad (5.1)
\]

\[
m_2 \frac{d^2 x_2}{dt^2} + (k_1 + k_2 + k_d) x_2 = k_1 x_1 + k_d y \quad (5.2)
\]

5.2 Analytical solution

It is assumed that the frictionally damped spring starts slipping at the limiting value of \( x_2 = x_{20} \). In addition, the system is linear in nature before the slip occurred and the value of \( y = 0 \) when the system is in the linear region. In the linear regions of response, the equations (5.1) and (5.2) are combined and written as given in equation (5.3).

\[
[M] \frac{d^2 \{\bar{x}\}}{dt^2} + [C_L] \frac{d \{\bar{x}\}}{dt} + [K_L] \{\bar{x}\} = \{F\} \quad (5.3)
\]

Where

\[
\{\bar{x}\} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ is the vector of motion in two degrees of freedom} \quad (5.4)
\]

\[
[M] = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \text{ is the mass matrix} \quad (5.5)
\]

\[
[C_L] = \begin{bmatrix} c & 0 \\ 0 & 0 \end{bmatrix} \text{ is the linear damping matrix} \quad (5.6)
\]

\[
[K_L] = \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 + k_2 + k_d \end{bmatrix} \text{ is the linear stiffness matrix} \quad (5.7)
\]
\( \{F\} = \begin{bmatrix} Q \cos(\omega t) \\ 0 \end{bmatrix} \) is the force vector \( (5.8) \)

Equation (5.3) represents the equations of motion of the vibrational system in the linear region. Since \( \{F\} \) has been assumed to be harmonic in nature it is assumed that the steady-state response of each degree of freedom of motion is harmonic and the solution is taken to be as given in equation (5.9).

\[
x_i = A_i \cos(\omega t + \phi_i), \text{ for } i = 1,2. \quad (5.9)
\]

where,

\( x_i \) is the \( i^{th} \) element of \( \{ x \} \) vector,

\( A_i \) is the assumed amplitude of motion of \( i^{th} \) degree of freedom,

\( \phi_i \) is the assumed phase angle of motion of \( i^{th} \) degree of freedom.

The equation (5.3) is substituted with equation (5.9) and is compared for the terms of \( \cos(\omega t) \) and \( \sin(\omega t) \) to get equations (5.10) and (5.11) respectively.

\[
\begin{bmatrix} -\omega^2 [M] + [K_L] \end{bmatrix} \begin{bmatrix} A_1 \cos(\phi_1) \\ A_2 \cos(\phi_2) \end{bmatrix} + \begin{bmatrix} -\omega [C_L] \end{bmatrix} \begin{bmatrix} A_1 \sin(\phi_1) \\ A_2 \sin(\phi_2) \end{bmatrix} = \begin{bmatrix} Q \\ 0 \end{bmatrix} \quad (5.10)
\]

\[
\begin{bmatrix} -\omega^2 [M] + [K_L] \end{bmatrix} \begin{bmatrix} A_1 \sin(\phi_1) \\ A_2 \sin(\phi_2) \end{bmatrix} + \begin{bmatrix} -\omega [C_L] \end{bmatrix} \begin{bmatrix} A_1 \cos(\phi_1) \\ A_2 \cos(\phi_2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (5.11)
\]

Equations (5.10) and (5.11) are combined into as given by equation (5.12).

\[
[R_L] \{ A \} = \{ Q \} \quad (5.12)
\]

where,

\[
[R_L] = \begin{bmatrix} -\omega^2 [M] + [K_L] & -\omega [C_L] \\ -\omega [C_L] & -\omega^2 [M] + [K_L] \end{bmatrix} \quad (5.13)
\]
\[
\{A\} = \begin{bmatrix}
A_1 \cos(\phi_1) \\
A_2 \cos(\phi_2) \\
A_1 \sin(\phi_1) \\
A_2 \sin(\phi_2)
\end{bmatrix}
\]
(5.14)

\[
\{Q\} = \begin{bmatrix}
Q_1 \\
0 \\
0 \\
0
\end{bmatrix}
\]
(5.15)

The system of equations is solved for response in the linear region by solving equation (5.12) which is given by equation (5.16).

\[
\{A\} = [R]^{-1} \{Q\}
\]
(5.16)

The values of \(\{A\}\) are utilized to find the values of magnitude of the amplitude function \(\{A_m\}\) where,

\[
\{A_m\} = \begin{bmatrix}
A_1 \\
A_2
\end{bmatrix}
\]
(5.17)

and the values of \(A_i, i = 1, 2\) of \(\{A_m\}\) are calculated from equation (5.18).

\[
A_i = \sqrt{(A_i \cos \phi_i)^2 + (A_i \sin \phi_i)^2}
\]
(5.18)

However, if the value of the amplitude function \(A_2\) of the linear response at 2\(^{nd}\) degree of freedom where the frictionally damped spring of stiffness value \(k_d\) is connected, exceeded the limiting value of friction then it is realized that a slip has occurred and the vibrational motion of 2\(^{nd}\) degree of freedom has entered nonlinear region. As discussed in section 3.2 of Chapter 3, the nonlinear aspects are incorporated into the equations of motion of the vibrational model by
adding terms $\frac{b}{\omega A_i}$ and $-\frac{a}{A_i}$ to the damping and stiffness coefficients of equation of motion of 2\textsuperscript{nd} degree of freedom respectively, where

$$a = \frac{k_d}{\pi} \left( A_2 \pi - A_2 \theta_d + (A_2 - 2x_{20}) \sin(\theta_d) \right)$$  \hfill (5.19)

$$b = \frac{4k_d x_{20}}{\pi} \left( 1 - \frac{x_{20}}{A_2} \right)$$  \hfill (5.20)

$$\theta_d = \cos^{-1} \left( 1 - \frac{2x_{20}}{A_2} \right)$$  \hfill (5.21)

$x_{20}$ is the limiting value of $A_2$ when slip occurs.

This led to the equations of motion of the vibrational system in the non-linear region and are combined and written as given by equation (5.22).

$$\begin{bmatrix} M \end{bmatrix} \frac{d^2 \{ \ddot{X} \}}{dt^2} + \begin{bmatrix} C_{NL} \end{bmatrix} \frac{d \{ \dot{X} \}}{dt} + \begin{bmatrix} K_{NL} \end{bmatrix} \{ X \} = \{ F \}$$  \hfill (5.23)

where,

$$\begin{bmatrix} C_{NL} \end{bmatrix} = \begin{bmatrix} c & 0 \\ 0 & -\frac{b}{\omega A_2} \end{bmatrix}$$  \hfill (5.24)

$$\begin{bmatrix} K_{NL} \end{bmatrix} = \begin{bmatrix} k_i & -k_i \\ -k_i & k_i + k_2 + k_4 - \frac{a}{A_2} \end{bmatrix}$$  \hfill (5.25)

The equation (5.23) is expanded by substituting equation (5.9) and is compared for the terms of $\cos(\omega t)$ and $\sin(\omega t)$ to get equation (5.26).

$$\begin{bmatrix} R_{NL} \end{bmatrix} \{ A \} = \{ Q \}$$  \hfill (5.26)

Where $$\begin{bmatrix} R_{NL} \end{bmatrix} = \begin{bmatrix} -\omega^2 [M] + [K_{NL}] \\ -\omega [C_{NL}] \\ -\omega^2 [M] + [K_{NL}] \end{bmatrix}$$  \hfill (5.27)
The system of equations is solved for its response in the nonlinear region by solving all the 4 coupled equations entailed in equation (5.26) simultaneously. The new set of values of \( \{A_m\} \) are evaluated from the revised set of values of \( \{A\} \) obtained for the nonlinear response of the vibrational system.

### 5.3 Solution by Padé’s interpolation method

From chapter-2 it is understood that \( N_d \) is the number of successive derivatives of amplitude function \( A_i(\omega) \) with respect to \( \omega \) for \( i = 1,2 \) which are required to determine the Padé’s approximant amplitude functions \( A_{i,Pade}(\omega) \), respectively. The method to find the differentials for the two degrees of freedom system have been discussed in this section.

Following the procedure discussed in chapter-4, equation (4.19) is iteratively used to find the value of the derivatives \( \{A_i^{[j]}\} \) for \( i = 1,2 \) and \( j = 1,2,\ldots,N_d \) in the region of linear response of the system while equation (4.28) is iteratively used to find the value of the derivatives \( \{A_i^{[j]}\} \) for \( i = 1,2 \) and \( j = 1,2,\ldots,N_d \) in the region of nonlinear response of the system. The values for \( \{A_m^{[k]}\} \) and the Padé’s approximant amplitude function \( A_{i,Pade}(\omega) \) are evaluated with the values calculated for \( \{A_i^{[j]}\} \), as discussed in section 4.2 of chapter-4.

As discussed earlier in section 3.3 of chapter-3, by replacing the solution parameter \( \sigma \) and the solution function \( u(\sigma) \) with the excitation frequency \( \omega \) and each amplitude function
$A_1(\omega)$ and $A_2(\omega)$, respectively over a frequency domain $[\omega_L, \omega_R]$ with defined interested coarse points of frequency $\omega_j \in [\omega_L, \omega_R]$ for $j = 1, 2, \ldots, n$ in the procedure followed in section 2.2 of chapter-2, it is claimed that the Padé’s approximant function developed $u_{pade}(\sigma)$ in section 2.2 corresponding to each amplitude serves as an approximate solution function to each amplitude function $A_1(\omega)$ and $A_2(\omega)$, denoted by $A_{i,pade}(\omega)$ and $A_{2,pade}(\omega)$, respectively.

5.4 Comparison of Results

According to [4], the values for the parameters of the problem of interest are taken as given below. The values of $k_d$ and $c$ were chosen in such a manner that the nonlinearity of the problem is reduced to a practical engineering application.

$m_1 = 15kg$
$m_2 = 250kg$
$k_1 = 23000N/m$
$k_2 = 10000N/m$
$k_d = 5000N/m$
c = 78.3156N.s/m$
$Q = 100N$
$x_{20} = 0.05m$

The analytical solution and the Padé’s approximant solution to both the amplitude functions $A_1(\omega)$ and $A_2(\omega)$ have been found out. The accuracy of the results was further improved by dividing the entire frequency region into more number of sample spaces, as has been discussed in section 3.4 of Chapter-3. For an interested frequency domain of $[\omega_L, \omega_R] = [0, 45]rad/s$, the sample spaces have been chosen as given below.
Sample space – 1: $[0.0,6.6]$ rad/s with $L = 19, M = 20, n = 10$.

Sample space – 2: $[6.6,8.0]$ rad/s with $L = 44, M = 45, n = 30$.


Sample space – 4: $[13.5,30.0]$ rad/s with $L = 9, M = 10, n = 10$.

Sample space – 5: $[30.0,45.0]$ rad/s with $L = 29, M = 30, n = 20$.

A MATLAB code has been developed to evaluate the analytical solution and the Padé’s approximant solution to the amplitude functions $A_1(\omega)$ and $A_2(\omega)$ which has been attached in Appendix-C. The results obtained from the computation have been compared and contrasted below. The analytical solution and the Padé’s approximant solution to the amplitude functions $A_1(\omega)$ and $A_2(\omega)$ have been compared in Fig 5-2 and Fig 5-3 respectively.
Figure 5-2. Comparison of values of amplitude function $A_1(\omega)$ and Padé’s approximant amplitude function $A_{1,\text{Pade}}(\omega)$.

Figure 5-3. Comparison of values of amplitude function $A_2(\omega)$ and Padé’s approximant amplitude function $A_{2,\text{Pade}}(\omega)$.
The ‘True values (Training)’ represent the exact values of the amplitude function $A_i(\omega)$ for $i=1,2$ at the selected interpolation points $\omega_j$ in the domain $[\omega_L, \omega_R]$ for $j=1,2,...,n$ where $\omega_L = 0 \text{ rad/s}$ and $\omega_R = 45 \text{ rad/s}$. The ‘True values (Test)’ represent the values of the amplitude function $A_i(\omega)$ at the trisected points of each interval $[\omega_j, \omega_{j+1}]$ in the domain $[\omega_L, \omega_R]$ for $j=1,2,...,n$. The ‘True values (Training)’ and their derivatives have been used for the interpolation to give the Padé’s approximant amplitude function $A_{i,\text{Pade}}(\omega)$ for $i=1,2$. The ‘Padé approximation values (Training)’ represent the values of the approximant amplitude function $A_{i,\text{Pade}}(\omega)$ at the same points as the domain of the ‘True values (Training)’. The ‘Padé approximation values (Test)’ represent the values of the approximant amplitude function $A_{i,\text{Pade}}(\omega)$ at the same points as the domain of the ‘True values (Test)’.

The error between the analytical and the Padé’s approximant solution to the amplitude functions $A_1(\omega)$ and $A_2(\omega)$ have been calculated individually using equation (3.28), for the ‘Test’ points of frequency and have been depicted in Fig 5-4 and Fig 5-5 respectively.
Figure 5-4. Error between amplitude function $A_1(\omega)$ and Padé’s approximant amplitude function $A_{1,\text{Pade}}(\omega)$.

Figure 5-5. Error between amplitude function $A_2(\omega)$ and Padé’s approximant amplitude function $A_{2,\text{Pade}}(\omega)$. 
It is observed that the values of the Padé’s approximant amplitude functions $A_{i,Pade}(\omega)$ for $i = 1,2$ match well with that of the analytical solution of the amplitude functions $A_i(\omega)$ for $i = 1,2$ respectively, for the ‘Test’ points of frequency, as are evident from the error values from Fig 5-4 and Fig 5-5 respectively. In particular, a high value of error is observed within the frequency regions $[6.0,8.0]$ rad/s and $[11.0,13.0]$ rad/s as there is a sudden increase or decrease of the amplitude function values which causes severe variations in the derivative values within these frequency regions. An even more accurate Padé’s approximant amplitude function can be obtained by increasing the number of selected interpolation points in the frequency region of interest and by dissecting the frequency domain of interest into further multiple sample spaces.
Chapter 6

Finite element analysis of non-linear vibrational system

This chapter is focused on the extension of the analysis of the motion of a multi degree of freedom non-linear vibrational system with impedance matrix $Z$ of the form given by equation (1.2), onto finite element method. The procedure that has been discussed in chapter-4 has been employed to find the solution to the vibrational motion of the finite element model, and the feasibility along with the essential supplementary computations to implement the concept of Padé’s interpolation technique to match the analytical solution.

The finite element method is based on the premise that an approximate solution to any complex engineering problem can be reached by subdividing the structure into discrete number of smaller elements called finite elements. The formulation of the entire problem results in a system of complex algebraic equations while the formulation of these finite elements results in simple algebraic equations. These equations are then assembled into a larger system of equations that models the entire problem with the help of constraints and boundary conditions.

The equation of motion that defines a small element can be represented in the form as given by the equation (6.1).

$$
\left[ m_{el} \right] \frac{d^2 \{ \bar{x} \}}{dt^2} + \left[ c_{el} \right] \frac{d \{ \bar{x} \}}{dt} + \left[ k_{el} \right] \{ \bar{x} \} = \{ \bar{f} \}
$$

Where,

$\{ \bar{x} \}$ is the vector of general coordinates of the elemental vibrational model
\[ m \] is the local mass matrix of the elemental vibrational model

\[ c \] is the local damping matrix of the elemental vibrational model

\[ k \] is the local stiffness matrix of the elemental vibrational model

\[ f \] is the local force vector of the elemental vibrational model

The vibrational motion of each element is described by equation (6.1). The equations of motion of all the elements are assembled into a single equation of motion incorporating all the constraints and boundary equations as given in equation (6.2).

\[
[M_G] \frac{d^2 \{X\}}{dt^2} + [C_G] \frac{d\{X\}}{dt} + [K_G] \{X\} = \{F_G\} \tag{6.2}
\]

where,

\([X] \) is the vector of general coordinates of the whole vibrational model

\([M_G] \) is the global mass matrix of the whole vibrational model

\([C_G] \) is the global damping matrix of the whole vibrational model

\([K_G] \) is the global stiffness matrix of the whole vibrational model

\([F_G] \) is the global force vector of the whole vibrational model

It is observed that the equation (6.2) takes the same form as equation (4.1). The same procedure as discussed in Chapter-4 was followed to find the analytical and Padé’s approximant solution to the amplitude function.
6.1 Finite element analysis of a beam structure with attached frictionally damped spring

For the case of simplicity, the finite element method to analyze a uniform beam structure as given in Fig 6-1, has been discussed in this section although the concept discussed could be extended to any beam structure. In Fig 6-1, \([w_1, w_2, w_3, w_4]\) represents the general coordinates of the beam element, where \(w_1\) and \(w_2\) represent the lateral and angular displacements at node-1, while \(w_3\) and \(w_4\) represent the lateral and angular displacements at node-2.

Figure 6-1. Beam element of uniform cross-sectional area.

According to [5], the local stiffness matrix, the local mass matrix and the local force vector of the beam element have been developed as given in equations (6.3), (6.4) and (6.5) respectively.

\[
\begin{bmatrix}
12 & 6L_{el} & -12 & 6L_{el} \\
6L_{el} & 4L_{el}^2 & -6L_{el} & 2L_{el}^2 \\
-12 & -6L_{el} & 12 & -6L_{el} \\
6L_{el} & 2L_{el}^2 & -6L_{el} & 4L_{el}^2
\end{bmatrix}
\]

(6.3)

\[
\begin{bmatrix}
156 & 22L_{el} & 54 & -13L_{el} \\
22L_{el} & 4L_{el}^2 & 13L_{el} & -3L_{el}^2 \\
54 & 13L_{el} & 156 & -22L_{el} \\
-13L_{el} & -3L_{el}^2 & -22L_{el} & 4L_{el}^2
\end{bmatrix}
\]

(6.4)
\[
\{ \vec{f}_{el} \} = \left( \frac{p_0 L_{el}}{12} \right) \begin{bmatrix} 6 \\ L_{el} \\ 6 \\ -L_{el} \end{bmatrix}
\]  

(6.5)

Where,

\( E \) is the Young’s modulus of the material of the beam structure,

\( I \) is the area moment of inertia about the neutral axis of the beam structure,

\( L_{el} \) is the length of the element considered,

\( \rho \) is the density of the material of the beam structure,

\( A \) is the area of cross-section of the material of the beam structure,

\( p_0 \) is the uniform load acting on the beam element along the length \( L_{el} \).

It is assumed that there is no internal damping within the beam structure and hence the local damping matrix is given by the equation (6.6).

\[
[c_{el}] = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]  

(6.6)

The local stiffness matrices are assembled into a global stiffness matrix as shown in the equation (6.7).

\[
[K_G] = \sum_{i=1}^{n} [L_i] Y [k_{el,i}] L_i + [k_{el}]
\]  

(6.7)

Where,

\[
[L_i] = \begin{bmatrix}
0_{(4 \times i)} & Id_{(4 \times 4)} & 0_{(4 \times n-1)}_{(4 \times n)}
\end{bmatrix}
\]  

(6.8)

where \([Id_{uxu}]\) stands for identity matrix of size \(uxu\),

\([k_{el,i}]\) is the local stiffness matrix of \(i^{th}\) element,
is the external stiffness matrix which represents the external spring attached at the \( j^{th} \) node.

Similarly, the global damping, mass and force matrices are developed as shown in equations (6.10) to (6.12) respectively.

\[
[C_G] = \sum_{i=1}^{n} [L_i]^T [c_{el,i}] [L_i] + [c_{ex}] 
\]

(6.10)

\[
[M_G] = \sum_{i=1}^{n} [L_i]^T [m_{el,i}] [L_i] 
\]

(6.11)

\[
\{F_G\} = \sum_{i=1}^{n} [L_i]^T \{f_{el,i}\} 
\]

(6.12)

Where,

\([c_{el,i}]\) is the local stiffness matrix of \( i^{th} \) element,

\([m_{el,i}]\) is the local stiffness matrix of \( i^{th} \) element,

\([f_{el,i}]\) is the local stiffness matrix of \( i^{th} \) element,

\[
[c_{ex}] = \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ 0 \end{pmatrix}
\]

(6.13)
is the external damping matrix which represents the external damper attached at the $j^{th}$ node.

6.2 Finite element analysis of a sample beam structure

The beam structure of interest is a cantilever beam which consists of 3 elements with the frictionally damped spring attached at the fifth node as shown in Fig 6-2.

![Figure 6-2. The cantilever beam structure of interest.](image)

The motion at nodes 1 and 2 are arrested to incorporate the cantilever boundary conditions. In other words, the first two equations entailed in the equation (6.2) which has been developed for the problem of interest have been truncated. This was achieved by truncating the first two rows and columns of the global stiffness, damping and mass matrices, reducing their size to $(Nu-2)\times(Nu-2)$ where $Nu$ represents the total number of nodes.

From chapter-2 it is understood that $N_d$ is the number of successive derivatives of amplitude function $A_i(\omega)$ with respect to $\omega$ for $i=1,2,...,Nu$ which are required to determine
the Padé’s approximant amplitude function $A_{i,\text{Pade}}(\omega)$. Following the procedure discussed in chapter-4, equation (4.19) is iteratively used to find the value of the derivatives $\{A_i\}^{[j]}$ for $i = 1,2$ and $j = 1,2,\ldots,N_d$ in the region of linear response of the system while equation (4.28) is iteratively used to find the value of the derivatives $\{A_i\}^{[j]}$ for $i = 1,2,\ldots,N_u$ and $j = 1,2,\ldots,N_d$ in the region of nonlinear response of the system. The values for $\{A_n\}^{[k]}$ and the Padé’s approximant amplitude functions for each node $A_{i,\text{Pade}}(\omega)$ for $i = 1,2,\ldots,N_u$ are evaluated with the values calculated for $\{A_i\}^{[j]}$, as discussed in section 4.2 of chapter-4.

As discussed earlier in section 3.3 of chapter-3, by replacing the solution parameter $\sigma$ and the solution function $u(\sigma)$ with the excitation frequency $\omega$ and the amplitude function of each node, respectively over a frequency domain $[\omega_L,\omega_R]$ with defined interested coarse points of frequency $\omega_j \in [\omega_L,\omega_R]$ for $j = 1,2,\ldots,n$ in the procedure followed in section 2.2 of chapter-2, it is claimed that the Padé’s approximant function developed $u_{\text{Pade}}(\sigma)$ in section 2.2 corresponding to each amplitude serves as an approximate solution function to each amplitude function $A_i(\omega)$ for $i = 1,2,\ldots,N_u$, denoted by $A_{i,\text{Pade}}(\omega)$ for $i = 1,2,\ldots,N_u$, respectively.
According to [6], the values for the parameters of the problem of interest were chosen as given below.

\[
\begin{align*}
I &= 7.395e - 7m^4 \\
E &= 140GPa \\
\rho &= 2700kg/m^3 \\
A &= 6.2715e - 4m^2 \\
k_d &= 60000N/m \\
c &= 78.316N.s/m \\
Q &= 1000N \\
A_{\infty} &= 0.09m \\
L_{beam} &= 2m
\end{align*}
\]

The values for the parameters for the evaluation of the Padé’s approximant are as given below.

\[
\begin{align*}
L &= 29 \\
M &= 30 \\
n &= 20
\end{align*}
\]

A MATLAB code has been developed to evaluate the analytical solution and the Padé’s approximant solution to the amplitude functions corresponding to all the eight nodes, which has been attached in Appendix-D. The results obtained from the computation for a frequency domain of \([\omega_L, \omega_R] = [200, 245] rad/s\) concentrated towards the first natural frequency of the beam structure of interest, have been compared and contrasted below. The shape of the lateral displacement of the cantilever beam structure for different excitation frequencies varying from 200 rad/s to 245 rad/s have been depicted in Fig 6-3 in increasing amplitude values at each node. The analytical solution and the Padé’s approximant solution to the amplitude functions of all the eight nodes have been compared in Fig 6-4 to Fig 6-11, respectively.
The ‘True values (Training)’ represent the exact values of the amplitude function $A_i(\omega)$ for $i = 1,2,\ldots,n$ at the selected interpolation points $\omega_j$ in the domain $[\omega_L, \omega_R]$ for $j = 1,2,\ldots,n$ where $\omega_L = 200\text{ rad/s}$ and $\omega_R = 245\text{ rad/s}$. The ‘True values (Test)’ represent the values of the amplitude function $A_i(\omega)$ at the trisected points of each interval $[\omega_j, \omega_{j+1}]$ in the domain $[\omega_L, \omega_R]$ for $j = 1,2,\ldots,n$. The ‘True values (Training)’ and their derivatives have been used for the interpolation to give the Padé’s approximant amplitude function $A_{i,\text{Pade}}(\omega)$ for $i = 1,2,\ldots,n$. The ‘Padé approximation values (Training)’ represent the values of the approximant amplitude function $A_{i,\text{Pade}}(\omega)$ at the same points as the domain of the ‘True values (Training)’. The ‘Padé approximation values (Test)’ represent the values of the approximant amplitude function $A_{i,\text{Pade}}(\omega)$ at the same points as the domain of the ‘True values (Test)’.

The error between the analytical and the Padé’s approximant solution to the amplitude functions of all the eight nodes have been calculated individually using equation (3.28), for the ‘Test’ points of frequency and have been depicted in their corresponding figures.
Figure 6-3. The shape of the lateral displacement of the cantilever beam structure for different excitation frequencies varying from 200 rad/s to 245 rad/s.

Figure 6-4. a) Comparison of analytical and Padé’s approximant solution to amplitude function $A_i(\omega)$. b) Error between the analytical and Padé’s approximant solution values.
Figure 6-5. a) Comparison of analytical and Padé’s approximant solution to amplitude function $A_2(\omega)$, b) Error between the analytical and Padé’s approximant solution values.

As the amplitudes of the first two nodes were set to zero to incorporate the cantilever boundary conditions, hence they were reflected on Fig 6-4 and Fig 6-5, respectively.

Figure 6-6. a) Comparison of analytical and Padé’s approximant solution to amplitude function $A_3(\omega)$, b) Error between the analytical and Padé’s approximant solution values.
Figure 6-7. a) Comparison of analytical and Padé’s approximant solution to amplitude function $A_4(\omega)$, b) Error between the analytical and Padé’s approximant solution values.

Figure 6-8. a) Comparison of analytical and Padé’s approximant solution to amplitude function $A_5(\omega)$, b) Error between the analytical and Padé’s approximant solution values.
Figure 6-9. a) Comparison of analytical and Padé’s approximant solution to amplitude function $A_6(\omega)$, b) Error between the analytical and Padé’s approximant solution values.

Figure 6-10. a) Comparison of analytical and Padé’s approximant solution to amplitude function $A_7(\omega)$, b) Error between the analytical and Padé’s approximant solution values.
Figure 6-11. a) Comparison of analytical and Padé’s approximant solution to amplitude function $A_8(\omega)$, b) Error between the analytical and Padé’s approximant solution values.

It is observed from Fig 6-4 to Fig 6-11 that the Padé’s approximant amplitude function matches well with the analytical solution of the amplitude function as is evident from the error values from the same figures. In spite of the fact that the error values are well within the acceptable range and did belong to the order of $10^{-3}$ when compared with the highest value of the amplitude function, a comparatively higher value of error is observed within the frequency regions $[210,215]$rad/s and $[227.5,232.5]$rad/s in particular as there was a sudden increase or decrease of the amplitude function values which caused severe variations in the derivative values within these frequency regions. This had happened due to the inclusion of nonlinear aspects as the amplitude function $A_8(\omega)$ values have exceeded the limiting value of $A_{m0} = 0.09m$. An even more accurate approximant amplitude function can be obtained by increasing the number of selected interpolation points in the frequency region of interest and by dissecting the frequency domain of interest into further multiple sample spaces.
Chapter 7

Conclusion

It is observed that the application of Padé’s interpolation technique to nonlinear frequency sweep problems with impedance matrix $Z$ of the form as given by equation (1.2) is successful. This is achieved by the fact that the established general approach to developing algorithms to evaluate the derivatives of the steady-state amplitude function to the chosen class of nonlinear frequency sweep problems at the ‘Test’ frequency points has been proved to be efficient and feasible. This is evident from the fact that the error values between the analytical solution and the multi-point Padé’s approximant amplitude solution are well within the acceptable range.

The complexity in finding the values of the derivatives comes with the simultaneous solution of the $2n$ coupled equations entailed in the derivatives of equations (4.5) and (4.6). In the general approach discussed in this thesis to evaluate the derivatives of the steady-state amplitude function, it has been shown in equation (4.28) that not only can the derivatives of amplitude functions be written in the form as given by equation (7.1),

$$DX = E$$  \hspace{1cm} (7.1)

where,

$$D = [R] + [T]$$ from equation (4.28),

$$E = \left\{ Q^{[j]} - \sum_{r=1}^{i-1} j_{C_r} [R]^{[r]} \{ A \}^{[j-r]} \right\}$$ from equation (4.28),

and $X = \{ A \}^{[j]}$ from equation (4.28),
in which the values of the $D$ and $E$ matrices can be found from known parameter values and the values evaluated in the involved preceding steps, but it is also shown that the matrix $D$ has to be inverted only once for finding any number of derivatives at each frequency point of interest. This reinforces the claim that the general approach discussed in this thesis to evaluate the derivatives of the steady-state amplitude function, reduces the required computational efforts to a very large extent thereby, substantiating its feasibility and efficiency.

Hence the developed algorithms to compute the derivatives of solutions to frequency sweep problems can be used to implement the interpolation techniques to reduce the computational efforts and time required to solve for the response of nonlinear vibrational systems. With the light of the subject discussed in this thesis, it is believed that the general approach can be extended to diverse nonlinear frequency sweep problems of different forms and increased complexity.
Appendix A

MATLAB program code for chapter-3

% Program code developed by: Mr. Arjun Pradeep Kumar
% For Master’s thesis at The Pennsylvania State University
% Under the guidance of adviser: Dr. Alok Sinha

% Brief Description:
% The following MATLAB code is developed to perform an efficient computation
% to evaluate the analytical solution and the Padé’s approximant solution to
% the amplitude function of a single degree of freedom vibrational system.

close all
clear all
cle

last_x1 = [];
last_y1 = [];

for sample = 1:1:3
    %% Initialization

    e = 0.1;
    f0 = 1;
    E = 0.001; % zeta
    Mass = 1;

    %% Finding damped natural frequencies
    omgn = 1;
    omgd = sqrt(1 - power(E,2));

    %% Defining range of OMEGA

    if sample == 1
        w0 = 0;
        wf = 0.95;

        % Pade Requirements
        L = 9;
        M = 10;
        nc = 10;
    end
if sample == 2
    w0 = 0.95;
    wf = 0.965;

    % Pade Requirements
    L = 29;
    M = 30;
    nc = 30;
end

if sample == 3
    w0 = 0.965;
    wf = 1.25;

    % Pade Requirements
    L = 29;
    M = 30;
    nc = 30;
end

x_samp(1) = 0.95;
x_samp(2) = 0.965;

limit = 4;

%% Allocating variables
keq = 1;
meq = 1;
ceq = 2*E*sqrt(keq/meq);
Ad = [];
ARd = [];
Ald = [];

AR_true = [];
AI_true = [];
A_true = [];

ii = 1;

%% True Values

nd = (ceil((L+M+1)/nc)-1);
x = w0:(wf-w0)/(3*nc-3):wf;
syms Ar Ai;

for wv = w0:(wf-w0)/(3*nc-3):wf
% Defining PL & QL
PL = (keq-wv.*wv.*meq);
QL = (ceq.*wv);

% Defining R & Q
Rm = [PL -QL;
     -QL -PL];
Qm = [e*0 0];

Sm = vpa(Rm\(Qm));
solAR = Sm(1);
solAI = Sm(2);
solA = vpa(sqrt(power(solAR,2) + power(solAI,2)));

if solA <= 1
    AR_true(1,ii) = vpa(solAR);
    AI_true(1,ii) = vpa(solAI);
    A_true(1,ii) = vpa(solA);
else
    ANL = sqrt(Ar.^2 + Ai.^2);
    thetac = acos(1-2/ANL);
    a = e.*ANL.*(pi - thetac + 0.5*sin(2*thetac))/pi;
    b = (4.*e./pi).*(1 - 1./ANL);

    % Defining P & Q
    PNL = (keq - a./ANL -wv.*wv.*meq);
    QNL = (2*E + b./(wv.*ANL))*wv;

    % Defining Rm & Qm
    Rm = [PNL -QNL;
          -QNL -PNL];
    Am = [Ar; Ai];
    RAm = Rm*Am;
    [solAR, solAI] = solve(RAm == Qm,[Ar,Ai]);
    solA = vpa(sqrt(power(solAR,2) + power(solAI,2)));
    AR_true(1,ii) = vpa(solAR);
    AI_true(1,ii) = vpa(solAI);
    A_true(1,ii) = vpa(solA);
end
ii = ii+1;
end

%% Discrete Points
x_discrete = w0:(wf-w0)/(nc-1):wf;
size_true = size(A_true);
for i = 1:(size_true(2)+2)/3
    Ad(1,i) = A_true(1,3*i-2);
    ARd(1,i) = AR_true(1,3*i-2);
    Ald(1,i) = AI_true(1,3*i-2);
end

i = 1;
g = sym('g',[2 1]);
g2 = sym('g2',[4 2]);
syms ARnlnt AInlnt Pnl(wnl) Qnl(wnl) Anl(wnl) ARnl(wnl) AInl(wnl) AR(w) AI(w) tempARdn tempAIdn tempARd tempAld;

% Defining R Linear
Pnl = (keq - wnl.*wnl.*meq);
Qnl = ceq.*wnl;
Rm = [Pnl -Qnl;
    -Qnl -Pnl];

% Defining R Non-linear
Anlt = sqrt(ARnlt.^2 + AInlt.^2);
Anl = sqrt(ARnl.^2 + AInl.^2);
theta = acos(1 - 2/Anlt);
a = e.*Anlt.*(pi - theta + 0.5*sin(2*theta))/pi;
b = (4.*e./pi).*((1 - 1./Anlt);
Pnl = (keq - a./Anlt - wnl.*wnl.*meq);
Qnl = (2*E + b./(wnl.*Anlt))*wnl;
Rtm = [Pnl -Qnl;
    -Qnl -Pnl];

for wv = w0:(wf-w0)/(nc-1):wf
    if Ad(1,i) <= 1
        % Finding Linear Ad
    for n = 1:nd
        eq_lin = 0;
        for r = 1:1:n
            eq_lin = eq_lin - (factorial(n)./(factorial(r).*factorial(n-r))).*(diff(Rm,wnl,r))*[ARd(n-r+1,i); Ald(n-r+1,i)];
        end
        Rm_inv = vpa(subs(inv(Rm),wnl,wv));
        Am = Rm_inv*(diff(Qm,wnl,n) + eq_lin);
        Amnd = vpa(subs(Am,wnl,wv));
        ARd(n+1,i) = vpa(Amnd(1));
        Ald(n+1,i) = vpa(Amnd(2));
    end
    else

%%% Finding Non-Linear Ad

% Solving for ARd & Ald
\[
g = \text{subs}(g,g,[\text{ARnl};\text{AInlt}]);
\]
\[
\text{Mtm} = [(\text{diff}(\text{Rtm},\text{ARnl},1)*g \text{ diff}(\text{Rtm},\text{AInlt},1))*g];
\]
\[
\text{Mtm}\_\text{dep} = \text{subs}(\text{Mtm},[\text{ARnl};\text{AInlt}],[\text{ARnl};\text{AInlt}]);
\]
\[
\text{Rtm}\_\text{dep} = \text{subs}(\text{Rtm},[\text{ARnl};\text{AInlt}],[\text{ARnl};\text{AInlt}]);
\]
\[
\text{Tm}\_L = \text{subs}(\text{Rtm}\_\text{dep} + \text{Mtm}\_\text{dep},[\text{ARnl};\text{AInlt}],[\text{tempARd};\text{tempAId}]);
\]
\[
\text{Tm}\_L = \text{subs}(\text{Tm}\_L,[\text{tempARd};\text{tempAId}],[\text{ARd}(1,i);\text{Ald}(1,i)]);
\]
\[
\text{Tm}\_L = \text{vpa}(\text{subs}(\text{Tm}\_L,\text{wnl},\text{wv}),12);
\]

\textbf{for} n = 1:nd
\[
\text{eq}\_\text{nlt} = \text{diff}(\text{Qm},\text{wnl},n);
\]
\textbf{for} r = 1:1:n-1
\[
g = \text{subs}(g,g,[\text{diff}(\text{ARnl},\text{wnl},n-r);\text{diff}(\text{AInlt},\text{wnl},n-r)]);
\]
\[
\text{eq}\_\text{nlt} = \text{eq}\_\text{nlt} - (\text{factorial}(n)/\text{factorial}(r)*\text{factorial}(n-r))*\text{diff}(\text{Rtm}\_\text{dep},\text{wnl},r)*g;
\]
\textbf{end}
\[
\text{eq}\_\text{nlt\_extra} = 0;
\]
\[
\text{dRdA} = [\text{diff}(\text{Rtm},\text{ARnl},1) \text{ diff}(\text{Rtm},\text{AInlt},1)];
\]
\[
\text{dRdA}\_\text{dep} = \text{subs}(\text{dRdA},[\text{ARnl};\text{AInlt}],[\text{ARnl};\text{AInlt}]);
\]
\textbf{for} s = 1:1:n-1
\[
g2 = \text{subs}(g2,g2,[\text{diff}(\text{ARnl},\text{wnl},n-s)*\text{eye}(2);\text{diff}(\text{AInlt},\text{wnl},n-s)*\text{eye}(2)]);
\]
\[
\text{eq}\_\text{nlt\_extra} = \text{eq}\_\text{nlt\_extra} + (\text{factorial}(n-1)/\text{factorial}(s)*\text{factorial}(n-1-s))*\text{diff}(\text{dRdA}\_\text{dep},\text{wnl},s)*g2;
\]
\textbf{end}
\[
\text{additional} = \text{diff}(\text{Rtm},\text{wnl},1);
\]
\[
\text{additional}\_\text{dep} = \text{subs}(\text{additional},[\text{ARnl};\text{AInlt}],[\text{ARnl};\text{AInlt}]);
\]
\[
\text{additional}\_\text{dep}\_\text{diff} = \text{diff}(\text{additional}\_\text{dep},\text{wnl},n-1);
\]
\[
\text{eq}\_\text{nlt\_extra} = \text{eq}\_\text{nlt\_extra} + \text{additional}\_\text{dep}\_\text{diff};
\]
\[
g = \text{subs}(g,g,[\text{ARnl};\text{AInlt}]);
\]
\[
\text{eq}\_\text{nlt\_extra} = \text{eq}\_\text{nlt\_extra}*g;
\]
\[
\text{eq}\_\text{nlt} = \text{eq}\_\text{nlt} - \text{eq}\_\text{nlt\_extra};
\]

% Substituting known values
\textbf{for} j = n-1:-1:0
\[
\text{eq}\_\text{nlt} = \text{subs}(\text{eq}\_\text{nlt},[\text{diff}(\text{ARnl},\text{wnl},j);\text{diff}(\text{AInlt},\text{wnl},j)]);\]
\textbf{end}
\[
\text{eq}\_\text{nlt} = \text{subs}(\text{eq}\_\text{nlt},\text{wnl},\text{wv});
\]
\[
\text{Tm}\_\text{R} = \text{vpa}(\text{eq}\_\text{nlt},12);
\]
\[
\text{Stm} = \text{vpa}((\text{Tm}\_\text{L}\backslash \text{Tm}\_\text{R}))
\]
\[
\text{ARd}(n+1,i) = \text{vpa}(\text{Stm}(1));
\]
\[
\text{Ald}(n+1,i) = \text{vpa}(\text{Stm}(2));
\]
\textbf{end}
end

for n = 1:nd
    A = sqrt(power(AR,2) + power(AI,2));
    ndA = diff(A,w,n);
    for j = n:-1:1
        ndA = subs(ndA,[diff(AR(w),w,j),diff(AI(w),w,j)],[tempARd,tempAId]);
        ndA = vpa(subs(ndA,[tempARd,tempAId],[ARd(j+1,i),AId(j+1,i)]));
    end
    ndA = vpa(subs(ndA,[AR(w),AI(w)],[ARd(1,i),AId(1,i)]));
    Ad(n+1,i) = vpa(ndA);
end

i = i+1;
end

kk = size(x);
kk = kk(2);
x_cut = x;
A_true_cut = A_true;

k = 4:3:kk;
x_cut(k) = [];
A_true_cut(k) = [];
kkk = 1:1:kk;
ind = 1:1:(kk+2)/3;
kkk(3*ind-2) = [];

figure(1)
plot(x,A_true,'b-x','LineWidth',0.8,'MarkerIndices',kkk(:),'MarkerSize',10);
hold on;
plot(x_discrete,Ad(1,:),'bd','MarkerSize',10);
hold on;
title('Nondimensionalized Amplitude function value');
xlabel('Nondimensionalized Excitation Frequency');
ylabel('Nondimensionalized amplitude value');

%% Pade Algorithm for Mass-1
clearvars -except Ad ARd Ald L M nd w0 nc x x_discrete A_true A_true_cut x_cut AR_true
omgd x_samp limit last_x1 last_x2 last_y1 last_y2;

A1 = [];
br1 = [];
p1 = [];
q1 = [];

h1 = 0:(wf-w0)/(nc-1):(wf-w0);
h1 = (h1);
\( B1 = \text{sym}(\text{'B1'}, \lceil(L+M+1)/nc \rceil (L+M+1)); \)
\( \text{c1} = \text{sym}(\text{'c1'}, \lceil(L+M+1)/nc \rceil 1); \)
\( \text{for } i = 1: \lceil(L+M+1)/nc \rceil \)
\( \quad \text{c1}(i,1) = \text{subs}(\text{c1}(i,1),0); \)
\( \quad \text{for } j = 1:(L+M+1) \)
\( \quad \quad B1(i,j) = \text{subs}(B1(i,j),0); \)
\( \quad \text{end} \)
\( \text{end} \)
\( \text{for } j = 1 : \text{nc} \)
\( \quad \text{for } a1 = 1 : \text{nd+1} \)
\( \quad \quad \text{for } b1 = a1 : L+1 \)
\( \quad \quad \quad B1(a1,b1) = \text{vpa}(B1(a1,b1) - (\text{factorial}(b1-1).*\text{power}(h1(j,1), (b1-1) - (a1-1)))./\text{factorial}((b1-1) - (a1-1)))); \)
\( \quad \quad \text{end} \)
\( \quad \text{end} \)
\( \quad \text{for } b1 = L+2 : L+M+1 \)
\( \quad \quad \text{for } r = 0 : (a1-1) \)
\( \quad \quad \quad \text{if } ((b1-(L+1)) >=r \&\& (b1-(L+1)) <= M) \)
\( \quad \quad \quad \quad B1(a1,b1) = \text{vpa}(B1(a1,b1) + ((\text{factorial}(a1-1))./\text{factorial}(r).*\text{factorial}((a1-1)-r))).*\text{Ad}((a1-1)-r+1,j).*\text{factorial}(b1-(L+1)).*\text{power}(h1(j,1), b1-(L+1)-r)./\text{factorial}(b1-(L+1)-r)); \)
\( \quad \quad \text{end} \)
\( \quad \text{end} \)
\( \quad c1(a1,1) = \text{vpa}(-\text{Ad}((a1-1)+1,j)); \)
\( \text{end} \)
\( \text{A1} = [A1' B1']; \)
\( \text{br1} = [br1' c1']; \)
\( \text{B1} = \text{zeros(size(B1))}; \)
\( \text{c1} = \text{zeros(size(c1))}; \)
\( \text{end} \)
\( \text{X1} = \text{vpa}(\text{A1}[: \text{br1}]); \)
\( \text{syms } y1 \ P1(y1) \ Q1(y1) \ PA1(y1); \)
\( \text{P1} = 0; \)
\( \text{Q1} = 1; \)
\( \text{for } i = 1:L+1 \)
\( \quad p1(i,1) = X1(i,1); \)
\( \quad \text{P1} = \text{P1} + (p1(i,1)).*\text{power}(y1,i-1); \)
\( \text{end} \)
\( \text{for } i = 1:M \)
\( \quad q1(i,1) = X1(i+(L+1),1); \)
\( \quad \text{Q1} = \text{Q1} + (q1(i,1)).*\text{power}(y1,i); \)
\( \text{end} \)
PA1 = P1./Q1;
pretty(PA1)
PA1_values = vpa(subs(PA1,y1, x - w0));
PA1_values_discrete = vpa(subs(PA1,y1, x_discrete - w0));

kk = size(x);
k = kk(2);
x_cut = x;
PA1_values_cut = PA1_values;

k = 4:3:kk;
x_cut(k) = [];
PA1_values_cut(k) = [];
kkk = 1:1:kk;
ind = 1:1:(kk+2)/3;
kkk(3*ind-2) = [];

figure(1)
plot(x,PA1_values,'r--O','LineWidth',2.5,'MarkerIndices',kk(:),'
MarkerSize',12);
hold on;
plot(x_discrete,PA1_values_discrete,'rs','LineWidth',1.6,'
MarkerSize',17);
hold on;
figure(2)
plot([last_x1 x_cut],[last_y1 -A_true_cut(1,:) + PA1_values_cut],'
Color','r','LineWidth',2);
title({'Nondimensionalized error between analytical and Pade approximant amplitude function
values; at Test values'});
xlabel({'Nondimensionalized Excitation Frequency (rad/sec)'});
ylabel({'Nondimensionalized error; of amplitude function value'});
hold on;
last_x1 = x_cut(end);
last_y1 = -A_true_cut(1,end) + PA1_values_cut(end);
end

figure(1)
set(gca,'FontSize',22);
line([x_samp(1) x_samp(1)],[0 limit],'LineWidth',1);
hold on;
line([x_samp(2) x_samp(2)],[0 limit],'LineWidth',1);
hold on;
legend('True values (Test)','True values (Training)','Pade approximation values (Test)','Pade approximation values (Training)');
Appendix B

Evaluation of Nonlinear Term

Figure B-1. Frictionally damped spring with spring constant $K_d$.

Figure B-2. Frictionally damped spring force vs displacement $A_i$.

The damper force of the frictionally damped spring shown in Fig B-1 is depicted in Fig B-2. The force exerted by the spring is given by the equation (B.1).

\[ F_c = K_d(A_i - y) \]  

(B.1)
where the nonlinear aspects are entailed in the term $K_d y$. It is assumed that the spring starts slipping at the limiting value of $A_i = A_{i0}$. Equations for the lines BC, CD, DE and EB are given by equations (B.2) to (B.5), respectively.

$$F_c = K_d (A_{i0})$$  (B.2)

$$F_c = K_d (A_i) - K_d (A_{m} - A_{i0})$$  (B.3)

$$F_c = -K_d (A_{i0})$$  (B.4)

$$F_c = K_d (A_i) + K_d (A_{m} - A_{i0})$$  (B.5)

Where $A_m$ represents the amplitude of the motion associated with the spring.

A relation between $y$ and $A_i$ was defined by substituting equation (B.1) into equations (B.2) to (B.5). The relation between $y$ and $A_i$ corresponding to the lines BC, CD, DE and EB are given by equations (B.6) to (B.9), respectively.

$$y = A_i - A_{i0}$$  (B.6)

where $(2A_{i0} - A_m) \leq A_i \leq A_m$

$$y = A_m - A_{i0}$$  (B.7)

where $-(2A_{i0} - A_m) \leq A_i \leq A_m$

$$y = A_i + A_{i0}$$  (B.8)

where $-(2A_{i0} - A_m) \geq A_i \geq -A_m$

$$y = -A_m + A_{i0}$$  (B.9)

where $(2A_{i0} - A_m) \geq A_i \geq -A_m$

Since the excitation force was harmonic in nature, the response amplitude function $A_i(\omega)$ was assumed to be harmonic as given by equation (B.10).
\[ A_i = A_m \cos(\theta), \text{ with } \theta = \omega t + \phi \quad (B.10) \]

where \( \omega \) and \( \phi \) represent the excitation frequency and the phase angle, respectively.

The displacement \( y \) could be represented as a function of \( \theta \) with substitution of equation (B.10) in to equations (B.6) to (B.9). Following a similar procedure as discussed in [3] it was found that \( y(\theta) = -y(\theta + \pi) \) and that the period of \( y \) was same that of \( A_i \). The nonlinear term \( K_d y \) was represented as a Fourier series with only the fundamental components as given by equation (B.11).

\[ K_d y = a \cos(\theta) + b \sin(\theta) \quad (B.11) \]

Where,

\[ a = \frac{2}{\pi} \int_0^\pi K_d y \cos(\theta) d\theta \quad (B.12) \]

\[ b = \frac{2}{\pi} \int_0^\pi K_d y \sin(\theta) d\theta \quad (B.13) \]

The integrals in the equations (B.12) and (B.13) have been evaluated and given by equations (B.14) and (B.15), respectively.

\[ a = \frac{K_d}{\pi} \left[ (A_m - 2A_{i0}) \sin(\theta_d) + A_m (\pi - \theta_d) \right] \quad (B.14) \]

\[ b = \frac{4K_d A_{i0}}{\pi} \left[ 1 - \frac{A_{i0}}{A_m} \right] \quad (B.15) \]

where,

\[ \theta_d = \cos^{-1} \left( 1 - \frac{2A_{i0}}{A_m} \right) \quad (B.16) \]
Appendix C

MATLAB program code for chapter-5

% Program code developed by: Mr. Arjun Pradeep Kumar
% For Master’s thesis at The Pennsylvania State University
% Under the guidance of adviser: Dr. Alok Sinha

% Brief Description:
% The following MATLAB code is developed to perform an efficient computation
% to evaluate the analytical solution and the Padé’s approximant solution to
% the amplitude function of a two degrees of freedom vibrational system.

close all
clear all
cle

last1_x1 = []; last1_y1 = [];
last2_x1 = []; last2_y1 = [];

%%% Initialization

for sample = 1:1:5

% System characteristics
m1 = 15;
m2 = 250;
k1 = 23000;
k2 = 10000;
kd = 5000;
E = 0.01;
c = 2*E*sqrt(k1/m1)*100;
Q = 100;
x20 = 0.05;

%%% Finding natural frequencies

Mm = [1 0 0 0;
     0 1 0 0;
     0 0 m1 0;
     0 0 0 m2]

Dm = [0 0 -1 0;
      0 0 0 -1;]
k1 -k1 c 0;
-k1 (k1+k2+kd) 0 0]

[Vm,Em] = eig(Dm,Mm);

omg1 = min(imag(Em(1,1)),imag(Em(3,3)))
omg2 = max(imag(Em(1,1)),imag(Em(3,3)))

%% Defining range of OMEGA

if sample == 1
    w0 = 0;
    wf = 6.6;

    % Pade Requirements
    L = 19;
    M = 20;
    nc = 10;
end

if sample == 2
    w0 = 6.6;
    wf = 8;

    % Pade Requirements
    L = 44;
    M = 45;
    nc = 30;
end

if sample == 3
    w0 = 8;
    wf = 13.5;

    % Pade Requirements
    L = 39;
    M = 40;
    nc = 40;
end

if sample == 4
    w0 = 13.5;
    wf = 30;

    % Pade Requirements
    L = 9;
    M = 10;
    nc = 10;
end
if sample == 5
    w0 = 30;
    wf = 45;

    % Pade Requirements
    L = 29;
    M = 30;
    nc = 20;
end

x_samp(1) = 6.85;
x_samp(2) = 8;
x_samp(3) = 13.5;
x_samp(4) = 30;

limit = 0.08;

%% Allocating variables

A1d = [];
AR1d = [];
AI1d = [];

A2d = [];
AR2d = [];
AI2d = [];

AR1_true = [];
AI1_true = [];
A1_true = [];

AR2_true = [];
AI2_true = [];
A2_true = [];

ii = 1;
x = [];
x_discrete = [];

%% True Values

nd = (ceil((L+M+1)/nc)-1);
x = w0:(wf-w0)/(3*nc-3):wf;
syms Ar1 Ai1 Ar2 Ai2;

for wv = w0:(wf-w0)/(3*nc-3):wf
% Defining PL & QL
P1L = (k1-wv.*wv.*m1);
P2L = (k1);
Q1L = (c.*wv);
Q2L = (k1+k2+kd.*wv.*wv);

% Defining R & Q
Rm = [P1L -P2L -Q1L 0; -Q1L 0 -P1L P2L; -P2L Q2L 0 0; 0 0 P2L -Q2L];
Qm = [Q 0 0 0]';

Sm = vpa(Rm*Qm);
solAR1 = vpa(Sm(1),12);
solAR2 = vpa(Sm(2),12);
solAI1 = vpa(Sm(3),12);
solAI2 = vpa(Sm(4),12);

solA1 = vpa(sqrt(power(solAR1,2) + power(solAI1,2)));
solA2 = vpa(sqrt(power(solAR2,2) + power(solAI2,2)));

if solA2 <= x20
   AR1_true(1,ii) = (solAR1);
   AI1_true(1,ii) = (solAI1);
   A1_true(1,ii) = (solA1);
   AR2_true(1,ii) = (solAR2);
   AI2_true(1,ii) = (solAI2);
   A2_true(1,ii) = (solA2);
else
   omegaNL_check = wv
   ANL2 = sqrt(Ar2.^2 + Ai2.^2);
   thetad = acos(1-2.*x20./ANL2);
   a = (1./pi)*(ANL2*pi - ANL2*thetad + (ANL2-2*x20)*sin(thetad));
   b = (4.*x20./pi).*(1 - x20./ANL2);

% Defining P & Q
P1NL = (k1-wv.*wv.*m1);
P2NL = (k1);
Q1NL = (c.*wv);
Q2NL = (k1+k2+kd.*(1-a./ANL2)-m2.*wv.*wv);
Q3NL = (b.*kd./ANL2);

% Defining Rm & Qm
Rm = [P1NL -P2NL -Q1NL 0; -Q1NL 0 -P1NL P2NL; -P2NL Q2NL 0 -Q3NL];
0 -Q3NL P2NL -Q2NL;

Am = [Ar1; Ar2; Ai1; Ai2];
RAm = Rm*Am;
[solAR1, solAI1, solAR2, solAI2] = solve([RAm(1) == Qm(1), RAm(2) == Qm(2), RAm(3) == Qm(3), RAm(4) == Qm(4)],[Ar1, Ai1, Ar2, Ai2]);

solA1 = vpa(sqrt(power(solAR1, 2) + power(solAI1, 2)));
solA2 = vpa(sqrt(power(solAR2, 2) + power(solAI2, 2)));

AR1_true(1, ii) = (solAR1);
AI1_true(1, ii) = (solAI1);
A1_true(1, ii) = (solA1);

AR2_true(1, ii) = (solAR2);
AI2_true(1, ii) = (solAI2);
A2_true(1, ii) = (solA2);

end

%% Discrete Points

x_discrete = w0:(wf-w0)/(nc-1):wf;
size_true = size(A1_true);
for i = 1:(size_true(2)+2)/3
    A1d(1,i) = A1_true(1, 3*i-2);
    AR1d(1,i) = AR1_true(1, 3*i-2);
    AI1d(1,i) = AI1_true(1, 3*i-2);
    A2d(1,i) = A2_true(1, 3*i-2);
    AR2d(1,i) = AR2_true(1, 3*i-2);
    AI2d(1,i) = AI2_true(1, 3*i-2);
end

i = 1;
g = sym('g',[4 1]);
g2 = sym('g2',[16 4]);
syms AR1nlt AI1nlt AR2nlt AI2nlt P1nl(wnl) P2nl(wnl) Q1nl(wnl) Q2nl(wnl) Q3nl(wnl)
A1nl(wnl) A2nl(wnl) AR1nl(wnl) AI1nl(wnl) AR2nl(wnl) AI2nl(wnl) AR1(w) AI1(w) A1(w)
AR2(w) AI2(w) A2(w) tempAR1dn tempAI1dn tempAR2dn tempAI2dn tempAR1d tempAI1d
tempAR2d tempAI2d;
for wv = w0:(wf-w0)/(nc-1):wf
    if A2d(1,i) <= x20
        % Finding Linear Ad
        % Defining P & Q
        P1nl = (k1-wnl.*wnl.*m1);
        P2nl = (k1);
        Q1nl = c.*wnl;
    end
end
Q2nl = k1 + k2 + kd - m2 * wnl * wnl;

% Defining R & Q
Rm = [P1nl - P2nl - Q1nl 0;
     -Q1nl 0 - P1nl P2nl;
     -P2nl Q2nl 0 0;
     0 0 P2nl - Q2nl];
Rm_inv = subs(inv(Rm), wnl, wv);

for n = 1:nd
    eq_lin = 0;
    for r = 1:n
        eq_lin = eq_lin - ((factorial(n)/(factorial(r)*factorial(n-r)))*
        (diff(Rm, wnl, r))*[AR1d(n-r+1,i); AR2d(n-r+1,i); AI1d(n-r+1,i); AI2d(n-r+1,i)];
    end
    Am = Rm_inv*(diff(Qm, wnl, n) + eq_lin);
    Amnd = vpa(subs(Am, wnl, wv));

    AR1d(n+1,i) = vpa(Amnd(1,12));
    AR2d(n+1,i) = vpa(Amnd(2,12));
    AI1d(n+1,i) = vpa(Amnd(3,12));
    AI2d(n+1,i) = vpa(Amnd(4,12));
end

else

    %%% Finding Non-Linear Ad

    % Solving for ARd & AI

    A1nlt = sqrt(AR1nlt.^2 + AI1nlt.^2);
    A2nlt = sqrt(AR2nlt.^2 + AI2nlt.^2);

    A1nl = sqrt(AR1nl.^2 + AI1nl.^2);
    A2nl = sqrt(AR2nl.^2 + AI2nl.^2);

    % Defining P & Q

    thetad = acos((1 - 2*x20./A2nlt);
    a = (1./pi)*(A2nlt*pi - A2nlt*thetad + (A2nlt-2*x20)*sin(thetad));
    b = (4.*x20./pi).*(1 - x20./A2nlt);
    P1nlt = (k1 - wnl.*wnl.*m1);
    P2nlt = (k1);
    Q1nlt = (c.*wnl);
    Q2nlt = (k1 + k2 + kd*(1 - a./A2nlt)-m2.*wnl.*wnl);
    Q3nlt = (b.*kd./A2nlt);
    Anlt = [AR1nlt; AR2nlt; AI1nlt; AI2nlt];
    Anl = [AR1nl; AR2nl; AI1nl; AI2nl];
% Defining Rtm & Qtm

\[
Rtm = [P_{1nl} - P_{2nl} - Q_{1nl} 0; \\
- Q_{1nl} 0 - P_{1nl} P_{2nl}; \\
- P_{2nl} Q_{2nl} 0 - Q_{3nl}; \\
0 - Q_{3nl} P_{2nl} - Q_{2nl}];
\]

g = sym('g', [4 1]);
g = subs(g, g, [AR{1nl}; AR{2nl}; AI{1nl}; AI{2nl}]);
Mtm = \{\text{diff}(Rtm, AR{1nl}, 1) * g \text{ diff}(Rtm, AR{2nl}, 1) * g \text{ diff}(Rtm, AI{1nl}, 1) * g \text{ diff}(Rtm, AI{2nl}, 1) * g\};

\[
\text{Mtm} = \text{subs}(\text{Mtm}, \{\text{AR}, \text{AR}, \text{AI}, \text{AI}\}, \{\text{AR}, \text{AR}, \text{AI}, \text{AI}\});
\]

\[
\text{Ttm}_{\text{L}} = \text{subs}(\text{Mtm} + \text{Rtm}, \{\text{tempAR}, \text{tempAR}, \text{tempAI}, \text{tempAI}\}, \{\text{tempAR}, \text{tempAR}, \text{tempAI}, \text{tempAI}\});
\]

\[
\text{Ttm}_{\text{L}} = \text{vpa}(\text{subs}(\text{Ttm}_{\text{L}}, \{\text{wnl}; \text{wnl}; \text{wnl}; \text{wnl}\}, 12);
\]

\[
\text{for } n = 1:nd
\]

\[
\text{eq}_{\text{nlt}} = \text{diff}(Qm, \text{wnl}, n);
\]

\[
\text{for } r = 1:1:n-1
\]

\[
g = \text{subs}(g, g, [\text{diff}(\text{AR}, \text{wnl}, n-r) \text{diff}(\text{AR}, \text{wnl}, n-r) \text{diff}(\text{AI}, \text{wnl}, n-r) \text{diff}(\text{AI}, \text{wnl}, n-r)]);
\]

\[
\text{eq}_{\text{nlt}} = \text{eq}_{\text{nlt}} - (\text{factorial}(n)/(\text{factorial}(r) \text{factorial}(n-r))) \text{diff}(\text{Rtm}, \text{wnl}, r) * g;
\]

\[
\text{end}
\]

\[
\text{eq}_{\text{nlt}} = \text{eq}_{\text{nlt}} - \text{eq}_{\text{nlt}}_{\text{extra}}
\]

\[
\text{dRdA} = \text{diff}(\text{Rtm}, \text{AR}{1nl}, 1) \text{diff}(\text{Rtm}, \text{AR}{2nl}, 1) \text{diff}(\text{Rtm}, \text{AI}{1nl}, 1) \text{diff}(\text{Rtm}, \text{AI}{2nl}, 1);
\]

\[
\text{dRdA}_{\text{dep}} = \text{subs}(\text{dRdA}, \{\text{AR}, \text{AR}, \text{AI}, \text{AI}\}, \{\text{AR}, \text{AR}, \text{AI}, \text{AI}\});
\]

\[
\text{for } s = 1:1:n-1
\]

\[
g_{2} = \text{subs}(g_{2}, g_{2}, [\text{diff}(\text{AR}, \text{wnl}, n-s) \text{eye}(4) \text{diff}(\text{AR}, \text{wnl}, n-s) \text{eye}(4) \text{diff}(\text{AI}, \text{wnl}, n-s) \text{eye}(4) \text{diff}(\text{AI}, \text{wnl}, n-s) \text{eye}(4)]);
\]

\[
\text{eq}_{\text{nlt}}_{\text{extra}} = \text{eq}_{\text{nlt}}_{\text{extra}} + (\text{factorial}(n-1)/(\text{factorial}(s) \text{factorial}(n-1-s))) \text{diff}(\text{dRdA}_{\text{dep}}, \text{wnl}, s) * g_{2};
\]

\[
\text{end}
\]

\[
\text{additional} = \text{diff}(\text{Rtm}, \text{wnl}, 1);
\]

\[
\text{additional}_{\text{dep}} = \text{subs}(\text{additional}, \{\text{AR}, \text{AR}, \text{AI}, \text{AI}\}, \{\text{AR}, \text{AR}, \text{AI}, \text{AI}\});
\]

\[
\text{eq}_{\text{nlt}}_{\text{extra}} = \text{eq}_{\text{nlt}}_{\text{extra}} + \text{additional}_{\text{dep}}_{\text{diff}};
\]

\[
g = \text{subs}(g, g, [\text{AR}, \text{AR}, \text{AI}, \text{AI}]);
\]

\[
\text{eq}_{\text{nlt}}_{\text{extra}} = \text{eq}_{\text{nlt}}_{\text{extra}} * g;
\]

\[
\text{eq}_{\text{nlt}} = \text{eq}_{\text{nlt}} - \text{eq}_{\text{nlt}}_{\text{extra}}
\]

% Substituting known values
\[
\text{for } j = n-1:1:0
\]
\[
\text{eq\_nlt} = 
\text{subs(eq\_nlt,[diff(AR1nl,wnl,j),diff(AR2nl,wnl,j),diff(AI1nl,wnl,j),diff(AI2nl,wnl,j)],[tempAR1d, tempAR2d,tempAI1d,tempAI2d]);}
\text{eq\_nlt} = 
\text{subs(eq\_nlt,[tempAR1d,tempAR2d,tempAI1d,tempAI2d],[AR1d(j+1,i),AR2d(j+1,i),AI1d(j+1,i), AI2d(j+1,i))];}
\text{end}
\text{eq\_nlt} = \text{subs(eq\_nlt,wnl,wv); Ttm\_R} = \text{vpa(eq\_nlt,12)}
\]

\[
\text{Ttm} = \text{vpa((Ttm\_L\text{Ttm\_R}),12)}
\]

\[
\text{AR1d(n+1,i)} = \text{vpa(STM(1),12)};
\text{AR2d(n+1,i)} = \text{vpa(STM(2),12)};
\text{AI1d(n+1,i)} = \text{vpa(STM(3),12)};
\text{AI2d(n+1,i)} = \text{vpa(STM(4),12)};
\text{end}
\text{end}
\]

\[
\text{for n = 1:nd}
\text{A1} = \text{sqrt(power(AR1,2) + power(AI1,2));}
\text{A2} = \text{sqrt(power(AR2,2) + power(AI2,2));}
\text{ndA1} = \text{diff(A1,w,n);}
\text{ndA2} = \text{diff(A2,w,n);}
\text{for j = n:-1:1}
\text{ndA1} = \text{subs(ndA1,[diff(AR1(w),w,j),diff(AI1(w),w,j)],[tempAR1d,tempAI1d]);}
\text{ndA1} = \text{subs(ndA1,[tempAR1d,tempAI1d],[AR1d(j+1,i),AI1d(j+1,i))];}
\text{ndA2} = \text{subs(ndA2,[diff(AR2(w),w,j),diff(AI2(w),w,j)],[tempAR2d,tempAI2d]);}
\text{ndA2} = \text{subs(ndA2,[tempAR2d,tempAI2d],[AR2d(j+1,i),AI2d(j+1,i))];}
\text{end}
\text{ndA1} = \text{vpa(subs(ndA1,[AR1(w),AI1(w)],[AR1d(1,i),AI1d(1,i))]);}
\text{ndA2} = \text{vpa(subs(ndA2,[AR2(w),AI2(w)],[AR2d(1,i),AI2d(1,i))]);}
\text{A1d(n+1,i)} = \text{ndA1;}
\text{A2d(n+1,i)} = \text{ndA2;}
\text{end}
\text{i} = \text{i+1;}
\text{end}
\]

\[
\text{kk} = \text{size(x)};
\text{kk} = \text{kk(2)};
\text{x\_cut} = \text{x};
\text{A1\_true\_cut} = \text{A1\_true};
\text{A2\_true\_cut} = \text{A2\_true};\]
k = 4:3:kk;
x_cut(k) = [];
A1_true_cut(k) = [];
A2_true_cut(k) = [];
kkk = 1:1:kk;
ind = 1:1:(kk+2)/3;
kkk(3*ind-2)=[];

figure(1)
plot(x,A1_true,'b-x','LineWidth',0.8,'MarkerIndices',kk(:),'MarkerSize',10);
hold on;
plot(x_discrete,A1d(1,:),'bd','MarkerSize',10);
hold on;
title('Amplitude of MASS-1');
xlabel('Omega (rad/s)');
ylabel('Amplitude (m)');

figure(2)
plot(x,A2_true,'b-x','LineWidth',0.8,'MarkerIndices',kk(:),'MarkerSize',10);
hold on;
plot(x_discrete,A2d(1,:),'bd','MarkerSize',10);
hold on;
title('Amplitude of MASS-2');
xlabel('Omega (rad/s)');
ylabel('Amplitude (m)');

%% Pade Algorithm for Mass-1
clearvars -except A1d AR1d A1d1 A2d AR2d A1d2 A2d2 L M nd w0 wc xc x_discrete A1_true A2_true AR1_true AR2_true x_samp omg1d omg2d limit A1_true_cut A2_true_cut x_cut last1_x1 last1_x2 last1_y1 last1_y2 last2_x1 last2_x2 last2_y1 last2_y2;
A1 = [];
br1 = [];
X1 = [];
h1 = [];
p1 = [];
q1 = [];

h1 = 0:(wf-w0)/(nc-1):(wf-w0);
hs1 = (h1');

B1 = sym('B1','[ceil((L+M+1)/nc) (L+M+1)]');
c1 = sym('c1','[ceil((L+M+1)/nc) 1]');
for i = 1:ceil((L+M+1)/nc)
c1(i,1) = subs(c1(i,1),0);
    for j = 1:(L+M+1)
        B1(i,j) = subs(B1(i,j),0);
    end
end
for j = 1 : nc
    for a1 = 1 : nd+1
        for b1 = a1 : L+1
            B1(a1,b1) = vpa(B1(a1,b1) - (factorial(b1-1).*power(h1(j,1), (b1-1) - (a1-1)))./factorial((b1-1) - (a1-1)));
        end
        for b1 = L+2 : L+M+1
            for r = 0 : (a1-1)
                if ((b1-(L+1)) >= r && (b1-(L+1)) <= M)
                    B1(a1,b1) = vpa((B1(a1,b1) + ((factorial(a1-1))./(factorial(r).*factorial((a1-1)-r))).*A1d((a1-1)-r+1,j).*factorial(b1-(L+1)).*power(h1(j,1), b1-(L+1)-r)./factorial(b1-(L+1)-r)));
                end
            end
        end
    end
    c1(a1,1) = vpa(-A1d((a1-1)+1,j));
end

A1 = [A1' B1']';
br1 = [br1' c1']';
B1 = zeros(size(B1));
c1 = zeros(size(c1));
end

X1 = vpa(A1\br1);
syms y1 P1(y1) Q1(y1) PA1(y1);
P1 = 0;
Q1 = 1;
for i = 1:L+1
    p1(i,1) = X1(i,1);
    P1 = P1 + (p1(i,1)).*power(y1,i-1);
end

for i = 1:M
    q1(i,1) = X1(i+(L+1),1);
    Q1 = Q1 + (q1(i,1)).*power(y1,i);
end

PA1 = P1./Q1;
pretty(PA1)
PA1_values = vpa(subs(PA1,y1, x - w0));
PA1_values_discrete = vpa(subs(PA1,y1, x_discrete - w0));
%% Pade Algorithm for Mass-2

```matlab
A2 = []; br2 = []; X2 = []; h2 = []; p2 = []; q2 = [];

h2 = h1;

B2 = sym('B2',ceil((L+M+1)/nc) (L+M+1));
c2 = sym('c2',ceil((L+M+1)/nc) 1);
for i = 1:ceil((L+M+1)/nc)
    c2(i,1) = subs(c2(i,1),0);
    for j = 1:(L+M+1)
        B2(i,j) = subs(B2(i,j),0);
    end
end
for j = 1 : nc
    for a2 = 1 : nd+1
        for b2 = a2 : L+1
            B2(a2,b2) = vpa(B2(a2,b2) - (factorial(b2-1).*power(h2(j,1), (b2-1) - (a2-1))./factorial((b2-1) - (a2-1))));
        end
    end
end
for b2 = L+2 : L+M+1
    for r = 0 : (a2-1)
        if ((b2-(L+1)) >= r && (b2-(L+1)) <= M)
            B2(a2,b2) = vpa((B2(a2,b2) + ((factorial(a2-1))./factorial(r).*factorial((a2-1)-r))).*A2d((a2-1)-r+1,j).*factorial(b2-(L+1)).*power(h2(j,1), b2-(L+1)-r)./factorial(b2-(L+1)-r));
        end
    end
end
c2(a2,1) = vpa(-(A2d((a2-1)+1,j)));
end
A2 = [A2' B2'];
br2 = [br2' c2'];
B2 = zeros(size(B2));
c2 = zeros(size(c2));
end

X2 = vpa(A2\br2);
syms y2 P2(y2) Q2(y2) PA2(y2);
P2 = 0;
Q2 = 1;
```
for i = 1:L+1
    p2(i,1) = X2(i,1);
    P2 = P2 + (p2(i,1)).*power(y2,i-1);
end

for i = 1:M
    q2(i,1) = X2(i+(L+1),1);
    Q2 = Q2 + (q2(i,1)).*power(y2,i);
end

PA2 = P2./Q2;
pretty(PA2)
PA2_values = vpa(subs(PA2,y2, x - w0));
PA2_values_discrete = vpa(subs(PA2,y2, x discrete - w0));

kk = size(x);
kk = kk(2);
x_cut = x;
PA1_values_cut = PA1_values;
PA2_values_cut = PA2_values;

k = 4:3:kk;
x_cut(k) = [];
PA1_values_cut(k) = [];
PA2_values_cut(k) = [];
kkk = 1:1:kk;
ind = 1:1:(kk+2)/3;
kkk(3*ind-2) = [];

figure(1)
plot(x,PA1_values,'r--O','LineWidth',2.5,'MarkerIndices',kkk(:,),'MarkerSize',12);
hold on;
plot(x discrete,PA1_values_discrete,'rs','LineWidth',1.6,'MarkerSize',17);
hold on;

figure(2)
plot(x,PA2_values,'r--O','LineWidth',2.5,'MarkerIndices',kkk(:,),'MarkerSize',12);
hold on;
plot(x discrete,PA2_values_discrete,'rs','LineWidth',1.6,'MarkerSize',17);
hold on;

figure(3)
plot([last1_x1 x_cut],[last1_y1 -A1_true_cut(1,:) + PA1_values_cut],'r-p');
title({['Error between analytical and Padé approximant amplitude-1 function values';'at Test values']});
xlabel('Excitation Frequency (rad/sec)');
ylabel('Error of Amplitude function value (m)');
hold on;
figure(4)
plot([last2_x1 x_cut],[last2_y1 -A2_true_cut(1,:) + PA2_values_cut],r-p);
title({'Error between analytical and Pade approximant amplitude-2 function values';'at Test values'});
xlabel('Excitation Frequency (rad/sec)');
ylabel('Error of Amplitude function value (m)');
hold on;
last1_x1 = x_cut(end);
last1_y1 = -A1_true_cut(1,end) + PA1_values_cut(end);
last2_x1 = x_cut(end);
last2_y1 = -A1_true_cut(1,end) + PA1_values_cut(end);
end

figure(1)
line([x_samp(1) x_samp(1)],0 limit,LineWidth,1);
hold on;
line([x_samp(2) x_samp(2)],0 limit,LineWidth,1);
hold on;
line([x_samp(3) x_samp(3)],0 limit,LineWidth,1);
hold on;
line([x_samp(4) x_samp(4)],0 limit,LineWidth,1);
hold on;
legend('True values (Test)','True values (Training)','Pade approximation values (Test)','Pade approximation values (Training)');
set(gca,FontSize,20);

figure(2)
line([x_samp(1) x_samp(1)],0 limit,LineWidth,1);
hold on;
line([x_samp(2) x_samp(2)],0 limit,LineWidth,1);
hold on;
line([x_samp(3) x_samp(3)],0 limit,LineWidth,1);
hold on;
line([x_samp(4) x_samp(4)],0 limit,LineWidth,1);
hold on;
legend('True values (Test)','True values (Training)','Pade approximation values (Test)','Pade approximation values (Training)');
set(gca,FontSize,20);

figure(3)
set(gca,FontSize,20);
figure(4)
set(gca,FontSize,20);
Appendix D

MATLAB program code for chapter-6

% Program code developed by: Mr. Arjun Pradeep Kumar
% For Master's thesis at The Pennsylvania State University
% Under the guidance of adviser: Dr. Alok Sinha

% Brief Description:
% The following MATLAB code is developed to perform an efficient computation
% to evaluate the analytical solution and the Padé’s approximant solution to
% the amplitude function of a multi-degree of freedom vibrational system,
% with an application on to finite element method.

close all
clear all
cle

%% INITIALIZATION

Q = 1000;
x20 = 0.09; % Limiting value of Ai
Nel = 3;
kd1 = sym(5000);
sp_n = 2*(Nel+1)-3; % Node at which non-linear spring is attached
dm_n = 2*(Nel+1)-3; % Node at which damper is attached
cv = 78.3156;

%% Defining range of OMEGA

w0 = 200;
wf = 245;

% Pade Requirements
Lp = 29;
Mp = 30;
nc = 20;

nd = ceil((Lp+Mp+1)/nc - 1);
Ltot = 2;
EI = 1.0353e5;
rhoA = 3.3866/2;
Lel = Ltot/Nel; % element length

% uniform planar beam element stiffness and mass matrices
k = EI/Lel^3 * [12 6*Lel -12 6*Lel; 6*Lel 4*Lel^2 -6*Lel 2*Lel^2; ...
-12 -6*Lel 12 -6*Lel; 6*Lel 2*Lel^2 -6*Lel 4*Lel^2];
\[ m = \rho A \cdot L_{el}/420 \cdot \{ 156 \ 22 \cdot L_{el} \ -54 \ -13 \cdot L_{el} \} \]

% Matrix assembly
\[
\text{Nu} = 2^*(\text{Nel}+1); \quad \% \text{total number of coordinates in unconstrained system}
\]
\[
\text{M} = \text{sym} (\text{zeros}(\text{Nu}, \text{Nu}));
\]
\[
\text{C} = \text{M};
\]
\[
\text{K} = \text{M};
\]
\[
\text{F}(1: \text{Nu}, 1) = \text{sym} (\text{zeros}(\text{Nu}, 1));
\]

\textbf{for} \text{n}=1:1:\text{Nel} \quad % \text{assemble global matrices from local ones}
\textbf{end}

\text{K} = \text{K} + k;
\text{M} = \text{M} + m;

\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
As_temp = [As_temp1(w); As_temp2(w); As_temp3(w); As_temp4(w); As_temp5(w); As_temp6(w); As_temp7(w); As_temp8(w); As_temp9(w); As_temp10(w); As_temp11(w); As_temp12(w); As_temp13(w); As_temp14(w); As_temp15(w)];
Ac_temp = Ac_temp(1:Nu);
As_temp = As_temp(1:Nu);
Av_temp = sym('Av_temp', [Nu 1]);
temp_v = sym('temp_v', [1 2*Nu-4]);
SolA = (zeros(1,2*Nu-4));
assume([Ac As Ac_temp As_temp solA], 'real');
for i = 1:1:Nu
    Av(i) = sqrt(power(Ac(i),2)+power(As(i),2));
    Av_temp(i) = sqrt(power(Ac_temp(i),2)+power(As_temp(i),2));
end
thetad = acos(1-2.*x20./Av(sp_n));
a = (1./pi)*(Av(sp_n)*pi - Av(sp_n)*thetad + (Av(sp_n)-2*x20)*sin(thetad));
b = (4.*x20./pi).*(1 - x20./Av(sp_n));
Achg = Ac(3:Nu,1);
Asg = As(3:Nu,1);
Achg_temp = Ac_temp(3:Nu,1);
Asg_temp = As_temp(3:Nu,1);

%% TRUE VALUES
ii = 1;
w_true = w0:(wf-w0)/(3*nc-3):wf;
RmL = [-w.*w.*Mg+Kg -w.*Cg; -w.*Cg w.*w.*Mg-Kg];
KgNL = Kg;
CgNL = Cg;
KgNL(sp_n-2,sp_n-2) = KgNL(sp_n-2,sp_n-2) - kd1.*a./Av(sp_n);
CgNL(sp_n-2,sp_n-2) = CgNL(sp_n-2,sp_n-2) + kd1.*b./Av(sp_n)*w;
RmNL = [-w.*w.*Mg+KgNL -w.*CgNL; -w.*CgNL w.*w.*Mg-KgNL];
for wv = w0:(wf-w0)/(3*nc-3):wf
    % Linear Solution
    RmL_val = vpa(subs(RmL,w,wv));
    Qg = [Fg;
        sym(zeros(size(Fg)))];
    Am = vpa(RmL_val\Qg);
    A_R(1:2) = 0;
    A_I(1:2) = 0;
    A_mag(1:2) = 0;
    for i = 1:1:Nu-2
        A_R(i+2,ii) = Am(i,1);
        A_I(i+2,ii) = Am(Nu-2+i,1);
A_{mag}(i+2,ii) = \text{vpa}(\text{sqrt}((A_{m}(i,1))^2+(A_{m}(N_{u}-2+i,1))^2));

end

if A_{mag}(sp\_n,ii) > x20

%%% Non-Linear solution
R_{mNL\_val} = \text{subs}(R_{mNL},w,wv);
Qg = [F_{g};
\text{sym}([\text{zeros}(\text{size}(F_{g})))];
R_{Am} = R_{mNL\_val}*[A_{cg};A_{sg}];

solA = \text{solve}(R_{Am} == Qg,[A_{cg}',A_{sg}']);
solA = \text{struct2cell}(solA);
solA = \text{vpa}(solA);
for i = 1:1:N_{u}-2
\quad A_{R}(i+2,ii) = solA(i,1,1);
\quad A_{I}(i+2,ii) = solA(N_{u}-2+i,1,1);
\quad A_{mag}(i+2,ii) = \text{vpa}(\text{sqrt}((solA(i,1,1))^2+(solA(N_{u}-2+i,1,1))^2));
end

end

for i = 1:1:(N_{u}/2)
\quad B_{trans}(i,ii) = (A_{mag}(2*i-1,ii));
end

ii = ii + 1;
end

xp = 1:1:(N_{u}/2);
xp = xp';
ii = 1;
for wv = w0:(w_{f}-w0)/(3*nc-3):w_{f}
\quad figure(1)
\quad set(gca,'FontSize',20);
\quad plot(xp,B_{trans}(:,ii));
\quad title('Shape of the lateral displacement of beam');
\quad xlabel('Node number');
\quad ylabel('Lateral displacement (m)');
\quad hold on;
\quad ii = ii + 1;
end

%%% DISCRETE POINTS
w_{discrete} = w0:(w_{f}-w0)/(nc-1):w_{f};
size_{true} = size(A_{mag});
for i = 1:(size_true(2)+2)/3
    ARd(1).set(:,i) = A_R(:,3*i-2);
    AId(1).set(:,i) = A_I(:,3*i-2);
    Ad(1).set(:,i) = A_mag(:,3*i-2);
end

%% DIFFERENTIATION

i = 1;
for j = 2:1:nd+1
    ARd(j).set = ARd(1).set;
    AId(j).set = AId(1).set;
    Ad(j).set = Ad(1).set;
end
for wv = w0:(wf-w0)/(nc-1):wf
    if Ad(1).set(sp_n,i) <= x20
        %%% Linear ARd and AId
        RmL_inv = vpa(subs(inv(RmL),w,wv));
        for n = 1:nd
            eq_lin = 0;
            for r = 1:1:n
                eq_lin = eq_lin - (factorial(n)./(factorial(r).*factorial(n-r))).*(diff(RmL,w,r))*[ARd(n-r+1).set(3:Nu,i); AId(n-r+1).set(3:Nu,i)];
            end
            Am = RmL_inv*(diff(Qg,w,n) + eq_lin);
            Amnd = vpa(subs(Am,w,wv));
            ARd(n+1).set(:,i) = [0;0;Amnd(1:Nu-2)];
            AId(n+1).set(:,i) = [0;0;Amnd(Nu-2+1:end)];
        end
    else
        %%% Non-Linear ARd and AId
        Mtm = [];
        g = subs(g,g,[Acg;Asg]);
        for j = 1:1:Nu-2
            Mtm = [Mtm diff(RmNL,Acg(j),1)*g];
        end
        for j = 1:1:Nu-2
            Mtm = [Mtm diff(RmNL,Asg(j),1)*g];
        end
    end
Mtm_dep = subs(Mtm, [Acg', Asg'], [Acg_temp', Asg_temp']);
RmNL_dep = subs(RmNL, [Acg', Asg'], [Acg_temp', Asg_temp']);
Ttm_L = subs(RmNL_dep + Mtm_dep, [Acg_temp', Asg_temp'], temp_v);
Ttm_L = subs(Ttm_L, temp_v, [ARd(1).set(3:Nu,i)', AId(1).set(3:Nu,i)']);
Ttm_L = vpa(subs(Ttm_L, w, wv), 12);

for n = 1:nd
    eq_nlt = diff(Qg, w, n);
    for r = 1:n-1
        g = subs(g, g, [diff(Acg_temp, w, n-r); diff(Asg_temp, w, n-r)]);
        eq_nlt = eq_nlt - (factorial(n)./(factorial(r).*factorial(n-r))).*(diff(RmNL_dep, w, r))*g;
    end

    eq_nlt_extra = 0;
    dRdA = [];
    for j = 1:1:Nu-2
        dRdA = [dRdA diff(RmNL, Acg(j), 1)];
    end

    for j = 1:1:Nu-2
        dRdA = [dRdA diff(RmNL, Asg(j), 1)];
    end

dRdA_dep = subs(dRdA, [Acg', Asg'], [Acg_temp', Asg_temp']);

for s = 1:1:n-1
    ABar = [];
    for j = 1:1:Nu-2
        ABar = [ABar; diff(Acg_temp(j), w, n-s).*eye(2*Nu-4)];
    end

    for j = 1:1:Nu-2
        ABar = [ABar; diff(Asg_temp(j), w, n-s).*eye(2*Nu-4)];
    end

    g2 = subs(g2, g2, ABar);
    eq_nlt_extra = eq_nlt_extra + (factorial(n-1)./(factorial(s).*factorial(n-1-s))).*(diff(dRdA_dep, w, s))*g2;
end

additional = diff(RmNL, w, 1);
additional_dep = subs(additional, [Acg', Asg'], [Acg_temp', Asg_temp']);
additional_dep_diff = diff(additional_dep, w, n-1);
eq_nlt_extra = eq_nlt_extra + additional_dep_diff;

g = subs(g, g, [Acg_temp; Asg_temp]);
eq_nlt_extra = eq_nlt_extra*g;
eq_nlt = eq_nlt - eq_nlt_extra;

% Substituting known values
for j = n-1:-1:0
    TEMP_m = [];
    for k = 1:Nu-2
        TEMP_m = [TEMP_m, diff(Acg_temp(k),w,j)];
    end

    for k = 1:Nu-2
        TEMP_m = [TEMP_m, diff(Asg_temp(k),w,j)];
    end
    eq_nlt = subs(eq_nlt,TEMP_m,temp_v);
    eq_nlt = subs(eq_nlt,temp_v,[ARd(j+1).set(3:Nu,i)',AId(j+1).set(3:Nu,i)']);
    end
end

Ttm_R = vpa(eq_nlt);
Stm = vpa(Ttm_L\Ttm_R)

ARd(n+1).set(:,i) = vpa([0;0;Stm(1:Nu-2)]);
AId(n+1).set(:,i) = vpa([0;0;Stm(Nu-2+1:end)]);
end

for index = 1:1:Nu
    for n = 1:nd
        ndAi = diff(Ai,w,n);
        for j = n:-1:1
            ndAi = subs(ndAi,[diff(ARi(w),w,j),diff(AIi(w),w,j)],[tempARid,tempAIid]);
            ndAi = subs(ndAi,[tempARid,tempAIid],[ARd(1).set(index,i),AId(1).set(index,i)]);
        end
        ndAi = vpa(subs(ndAi,[ARi(w),AIi(w)],[ARd(1).set(index,i),AId(1).set(index,i)]));
        Ad(n+1).set(index,i) = ndAi;
    end
end

i = i + 1;
end

for index = 1:1:Nu
    kk = size(w_true);
    kk = kk(2);
    w_cut = w_true;
    A_mag_cut = A_mag(index,:);
    k = 4:3:kk;
    w_cut(k) = [];
end
A_mag_cut(k) = [];  
kkk = 1:1:kk;  
ind = 1:1:(kk+2)/3;  
kkk(3*ind-2)=[];

figure(index+1)
subplot(2,1,1)
plot(w_true,A_mag(index,:),'b-x','LineWidth',0.8,'MarkerIndices',kkk(:,1),'MarkerSize',10);  
hold on;
plot(w_discrete,Ad(1).set(index,:),'+','MarkerSize',10);
hold on;
end

clearvars -except Ad ARd Ald Lp Mp w0 wc nd w_discrete xp B_trans Nu A_mag  
A_mag_cut w_cut;

%% Pade Algorithm

for index = 1:1:Nu
    A1 = [];  
br1 = [];  
p1 = [];  
q1 = [];

    h1 = 0:(wf-w0)/(nc-1):(wf-w0);  
h1 = (h1');

    B1 = sym('B1',[ceil((Lp+Mp+1)/nc) (Lp+Mp+1)]);
    c1 = sym('c1',[ceil((Lp+Mp+1)/nc) 1]);
    for i = 1:ceil((Lp+Mp+1)/nc)
        c1(i,1) = subs(c1(i,1),0);
        for j = 1:(Lp+Mp+1)
            B1(i,j) = vpa(B1(i,j) - (factorial(b1-1).*power(h1(j,1), (b1-1)-(a1-1))./factorial((b1-1)-(a1-1))));
        end
    end

    for j = 1 : nc
        for a1 = 1 : nd+1
            for b1 = a1 : Lp+1
                B1(a1,b1) = vpa(B1(a1,b1) -(factorial(b1-1).*power(h1(j,1), (b1-1) - (a1-1))./factorial((b1-1) - (a1-1))));
            end
        end
    end

    for b1 = Lp+2 : Lp+Mp+1
        for r = 0 : (a1-1)
            if ((b1-(Lp+1)) >= r && (b1-(Lp+1)) <= Mp)
\[ B_1(a_1,b_1) = vpa((B_1(a_1,b_1) + ((\text{factorial}(a_1-1))/(\text{factorial}(r)*\text{factorial}((a_1-1)-r)))*\text{Ad}((a_1-1)-r+1).set(index,j)\text{factorial}(b_1-(Lp+1)).*\text{power}(h_1(j,1), b_1-(Lp+1)-r)/\text{factorial}(b_1-(Lp+1)-r))); \]

\[ c_1(a_1,1) = vpa(-\text{Ad}((a_1-1)+1).set(index,j)); \]

\[ A_1 = [A_1' B_1']'; \]
\[ b_{r1} = [b_{r1}' c_1']'; \]
\[ B_1 = \text{zeros(size}(B_1)); \]
\[ c_1 = \text{zeros(size}(c_1)); \]

\[ X_1 = A_1\backslash b_{r1}; \]
\[ \text{syms } y_1 P_1(y_1) Q_1(y_1) PA_1(y_1); \]
\[ P_1 = 0; \]
\[ Q_1 = 1; \]

\[ \text{for } i = 1:Lp+1 \]
\[ p_1(i,1) = X_1(i,1); \]
\[ P_1 = P_1 + (p_1(i,1)).*\text{power}(y_1,i-1); \]
\[ \text{end} \]

\[ \text{for } i = 1:Mp \]
\[ q_1(i,1) = X_1(i+(Lp+1),1); \]
\[ Q_1 = Q_1 + (q_1(i,1)).*\text{power}(y_1,i); \]
\[ \text{end} \]

\[ PA_1 = P_1./Q_1; \]
\[ \text{pretty}(PA_1); \]
\[ PA_1\text{\_values} = vpa(\text{subs}(PA_1,y_1, w\_true - w_0)); \]
\[ PA_1\text{\_values\_discrete} = vpa(\text{subs}(PA_1,y_1, w\_discrete - w_0)); \]

\[ kk = \text{size}(w\_true); \]
\[ kk = kk(2); \]
\[ w\_cut = w\_true; \]
\[ PA_1\text{\_values\_cut} = PA_1\text{\_values}; \]
\[ A\_mag\_cut = A\_mag(index,:); \]

\[ k = 4:3:kk; \]
\[ w\_cut(k) = []; \]
\[ PA_1\text{\_values}\_cut(k) = []; \]
\[ A\_mag\_cut(k) = []; \]
\[ kkk = 1:1:kk; \]
\[ \text{ind} = 1:1:(kk+2)/3; \]
figure(index+1)
set(gca,'FontSize',20);
subplot(2,1,1)
title(['Amplitude - ',num2str(index)]);
xlabel(['Excitation Frequency (rad/s)']);
ylabel(['Amplitude - ',num2str(index), ' (m)']);
plot(w_true,PA1_values,'r--O','LineWidth',2.5,'MarkerIndices',kkk(:),'MarkerSize',12);
hold on;
plot(w_discrete,PA1_values_discrete,'rs','LineWidth',1.6,'MarkerSize',17);
hold on;
set(gca,'FontSize',20);
LG = legend('True values (Test)','True values (Training)','Pade approximation values (Test)','Pade approximation values (Training)');
set(LG,'FontSize',16);

subplot(2,1,2)
plot(w_cut,-A_mag_cut + PA1_values_cut,'r-p');
title(['Error between analytical and Pade approximant amplitude - ',num2str(index),' function values']);
xlabel('Excitation Frequency (rad/sec)');
ylabel('Error of Amplitude function; value (m)');
hold on;
set(gca,'FontSize',20);
end
REFERENCES


5. G. Lesieutre Lecture notes of course ME571 Structural Dynamics at The Pennsylvania State University, Fall 2016.

6. G. Lesieutre Problem-01 of Homework-08 of course ME571 Structural Dynamics at The Pennsylvania State University, Fall 2016.