

**Dynamic Response of Structural Elements
Undergoing Moving Loads and Thermal Strains
Using Finite Elements**

A THESIS

SUBMITTED TO THE FACULTY OF

UNIVERSITY OF MINNESOTA

BY

Anthony Paganelli

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Dr. Kumar K. Tamma, Adviser

December 2014

© Anthony Paganelli 2014

Abstract

Many structures undergo forced vibration due to moving loads: bridges, railway and subway tracks, aircraft carrier decks, etc. Many of these structures are also subjected to various types of thermal loading. Currently, there are limited or no analytical or experimental methods for analyzing the combined effects of the mechanically induced vibrations and thermal loads on complicated structures such as plates and curved beams with moving loads. Instead, it is more preferable to analyze such problems by numerically discretizing the spatial portion of the equations of motion using Finite Elements and the temporal portion with a numerical time stepping algorithm. The preferred time discretization method presented here is the GSSSS framework of algorithms in conjunction with the Finite Element method. This research will focus on: 1.) Developing a procedure for solving the dynamic response of structures undergoing forced vibration due to moving loads, 2.) Applying this procedure to curved beam structures, and 3.) Analyzing effects of the moving loads and thermal loads on the combined dynamic response of curved beams and flat plates. These developments provide a baseline for future research in the areas of combined transient thermo-mechanical problems using the GSSSS family of algorithms.

Contents

Abstract.....	1
List of Tables	5
List of Figures.....	6
1. Chapter 1: Introduction.....	10
2. Chapter 2: Finite Elements for Structural Elements in Space and Time...14	
2.1 Euler-Bernoulli Beam Mass and Stiffness Formulations.....	15
2.1.1 Numerical Examples.....	16
2.2 Timoshenko Beam Mass and Stiffness Formulations.....	17
2.2.1 Numerical Examples.....	21
2.3 Mindlin-Reissner Plate Mass and Stiffness Formulations	22
2.3.1 Numerical Examples.....	27
2.4 Numerical Time Integration.....	27
3. Chapter 3: Dynamic Response of Structural Elements Subject to Moving Loads Using Finite Elements.....	31

3.1	Finite Element Formulation of a Moving Load	32
3.2	Numerical Examples	33
3.3	Conclusions.....	39
4. Chapter 4: Error Analysis of Numerically Non-Dissipative $V_0(1,1,\rho_{3\infty})$		
	Algorithms for a Beam Subject to a Moving Load	40
4.1	Numerical Examples	41
4.2	Conclusions.....	45
5. Chapter 5: Dynamic Response of Curved Beams Subject to Moving Loads		
	Using Finite Elements	46
5.1	Numerical Examples	50
5.1.1	Static Curved Beam Deflection.....	50
5.1.2	Free Vibration of Curved Beams	51
5.1.3	Curved Beams Subject to a Moving Load	52
5.2	Conclusions.....	58
6. Chapter 6: Dynamic Response of Curved Beams Subject to Moving Loads		
	and Thermal Strains Using Finite Elements	59
6.1	Numerical Examples	61
6.1.1	Thermal Expansion of a Ring	61
6.1.2	Thermal Bending of a Cantilever Beam	61
6.1.3	Curved Beams Subject to Moving Loads and Thermal Strains	62

6.2	Conclusions.....	65
7. Chapter 7: Dynamic Response of Plates Subject to Moving Loads and Thermal Strains Using Finite Elements.....66		
7.1	Numerical Examples.....	67
7.1.1	Thermal Bending of a Plate.....	67
7.1.2	Plates Subject to Moving Loads and Thermal Strains.....	68
7.2	Conclusions.....	70
8. Chapter 8: Concluding Remarks.....71		
9. References.....73		
Appendix A – Static Error Convergence Plots.....76		
Appendix B – Time Histories for Selected Curved Beams with Moving Loads and Thermal Strains.....79		
Appendix C – Time Histories for Selected Plates with Moving Loads and Thermal Strains.....88		
Appendix D – Matlab Codes.....95		

List of Tables

Table 1: Comparison of the FE transverse tip deflection, w_{min} , and the first two natural frequencies, ω_1 and ω_2 for various number of elements.	17
Table 2: Comparison of the transverse deflection, w_{min} , at the center of the beam and the first non-dimensional first natural frequency for two cases: (Left) $L/h=4$; (Right) $L/h=10$	22
Table 3: Comparison of the non-dimensional FE transverse deflection, w_{min} , at the center of the plate and the non-dimensional natural frequency, ω_w for various number of elements. .	27
Table 4: Common GSSSS V0 algorithms.....	29
Table 5: Non-dimensional deflections at the center of the beam for various loading cases and boundary conditions.....	51
Table 6: Fundamental frequency for curved beams of various radii.	52
Table 7: Maximum static deflection of the clamped-clamped beam for various radii of curvature.	58
Table 8: Change in radius of various rings subject to a temperature change of (Left) 40°C (Right) - 40°C	61
Table 9: Bending of the tip of a cantilever beam for various temperature changes through the height of the beam.....	62
Table 10: Five thermal loading cases.....	69

List of Figures

Figure 1: Transverse displacement, w , at $x = 7L16$ for three FE formulations: a) Timoshenko beam, b) Euler-Bernoulli beam, and c) Mindlin-Reissner plate as compared to the experimental measurements.....	35
Figure 2: Time histories at $x = 7L16$ calculated using Timoshenko beam theory and $V0(1,1,0)$ for: a) $w(t)$, b) $\theta y(t)$, c) $w(t)$, d) $\theta y(t)$, e) $w(t)$, and f) $\theta y(t)$	36
Figure 3: Time histories at $x = 7L16$ calculated using Euler-Bernoulli beam theory and $V0(1,1,0)$ for: a) $w(t)$, b) $\theta y(t)$, c) $w(t)$, d) $\theta y(t)$, e) $w(t)$, and f) $\theta y(t)$	37
Figure 4: Time histories at $x = 7L16$ and $y = W2$ calculated using Mindlin-Reissner plate theory and $V0(1,1,0)$ for: a) $w(t)$, b) $\theta y(t)$, c) $w(t)$, d) $\theta y(t)$, e) $w(t)$, and f) $\theta y(t)$	38
Figure 5: Comparison of the transverse deflection for Euler-Bernoulli beam theory, Timoshenko beam theory, and experimental measurements at $x = 7L16$	42
Figure 6: The Euler-Bernoulli truncation error, Π , at the midpoint using MPR-MPA and MPR-EPA as a function of the chosen time step for the following degrees of freedom: a) $w(t)$, b) $\theta y(t)$, c) $w(t)$, d) $\theta y(t)$, e) $w(t)$, and f) $\theta y(t)$	43
Figure 7: The Timoshenko truncation error, Π , at the midpoint using MPR-MPA and MPR-EPA as a function of the chosen time step for the following degrees of freedom: a) $w(t)$, b) $\theta y(t)$, c) $w(t)$, d) $\theta y(t)$, e) $w(t)$, and f) $\theta y(t)$	44

Figure 8: The truncation error, Π , as a function of the selected ρ_3 for the (Left) Euler-Bernoulli beam (Right) Timoshenko beam..... 45

Figure 9: Impact of the beam radius and the speed of the moving load on the maximum absolute vertical displacements at the midpoint of the beam according to [20]. 53

Figure 10: Time history of the vertical displacements of the curved beam at L_4 and L_2 with the speed of the load $v = 4ms$ according to [20]. 53

Figure 11: Time histories at $x = L_4, x = L_2, \& x = 3L_4$ for the moving mass moving at $v = 10ms$ for each of the following: a) ut , b) $w(t)$, c) $\theta(t)$, d) $u(t)$ e) $w(t)$, f) $\theta(t)$, g) $u(t)$, h) $w(t)$, and i) $\theta(t)$ 55

Figure 12: Time histories at $x = L_4, x = L_2, \& x = 3L_4$ for the moving mass moving at $v = 100ms$ for each of the following: a) ut , b) $w(t)$, c) $\theta(t)$, d) $u(t)$ e) $w(t)$, f) $\theta(t)$, g) $u(t)$, h) $w(t)$, and i) $\theta(t)$ 56

Figure 13: Effect of the beam radius of curvature and the speed of the moving load, v , on the maximum absolute non-dimensional vertical displacement, D 57

Figure 14: Effect of the beam radius of curvature and the speed of the moving load on the maximum absolute non-dimensional vertical displacement, D , for various thermal strains: a) $TTOP = 25^\circ C, TBOT = 25^\circ C, \& T_\infty = 25^\circ C$, b) $TTOP = 25^\circ C, TBOT = 25^\circ C, \& T_\infty = 30^\circ C$, c) $TTOP = 20^\circ C, TBOT = 25^\circ C, \& T_\infty = 22.5^\circ C$, d) $TTOP = 20^\circ C, TBOT = 25^\circ C, \& T_\infty = 25^\circ C$ 64

Figure 15: Comparison of the FE and exact solution for the thermal bending of the plate for (a) $w_{x,y} = W/2$ and (b) $w_x = L/2, y$ 68

Figure 16: Effect of the speed of the moving load on the maximum absolute non-dimensional vertical displacement, D , for various thermal strains..... 70

Figure 17: Error convergence of the first two natural frequencies as the element length decreases of Euler-Bernoulli beams from Section 2.1.1.	76
Figure 18: Error convergence of the first natural frequency as the element length decreases of two Timoshenko beams from Section 2.1.1.	77
Figure 19: Error convergence for the internal energy, $U = 12\mathbf{K}\{\mathbf{u}\}$ as the average element length decreases for the Mindlin-Reissner plate from Section 2.3.1.	77
Figure 20: Error convergence for the fundamental natural frequency as the average element length decreases for the Mindlin-Reissner plate from Section 2.3.1.	78
Figure 21: Time histories at $x = L4, x = L2, \& x = 3L4$ for the moving mass moving at $v = 10ms$, $TTOP = 25^\circ C, TBOT = 25^\circ C, T_\infty = 25^\circ C, \& R = 20m$ for each of the following: a), ut , b) $w(t)$, c) $\theta(t)$, d) $u(t)$ e) $w(t)$, f) $\theta(t)$, g) $u(t)$, h) $w(t)$, and i) $\theta(t)$	80
Figure 22: Time histories at $x = L4, x = L2, \& x = 3L4$ for the moving mass moving at $v = 10ms$, $TTOP = 25^\circ C, TBOT = 25^\circ C, T_\infty = 30^\circ C, \& R = 20m$ for each of the following: a), ut , b) $w(t)$, c) $\theta(t)$, d) $u(t)$ e) $w(t)$, f) $\theta(t)$, g) $u(t)$, h) $w(t)$, and i) $\theta(t)$	81
Figure 23: Time histories at $x = L4, x = L2, \& x = 3L4$ for the moving mass moving at $v = 10ms$, $TTOP = 20^\circ C, TBOT = 25^\circ C, T_\infty = 22.5^\circ C, \& R = 20m$ for each of the following: a), ut , b) $w(t)$, c) $\theta(t)$, d) $u(t)$ e) $w(t)$, f) $\theta(t)$, g) $u(t)$, h) $w(t)$, and i) $\theta(t)$	82
Figure 24: Time histories at $x = L4, x = L2, \& x = 3L4$ for the moving mass moving at $v = 10ms$, $TTOP = 20^\circ C, TBOT = 25^\circ C, T_\infty = 25^\circ C, \& R = 20m$ for each of the following: a), ut , b) $w(t)$, c) $\theta(t)$, d) $u(t)$ e) $w(t)$, f) $\theta(t)$, g) $u(t)$, h) $w(t)$, and i) $\theta(t)$	83
Figure 25: Time histories at $x = L4, x = L2, \& x = 3L4$ for the moving mass moving at $v = 100ms$, $TTOP = 25^\circ C, TBOT = 25^\circ C, T_\infty = 25^\circ C, \& R = 20m$ for each of the following: a), ut , b) $w(t)$, c) $\theta(t)$, d) $u(t)$ e) $w(t)$, f) $\theta(t)$, g) $u(t)$, h) $w(t)$, and i) $\theta(t)$	84

Figure 26: Time histories at $x = L4, x = L2, \& x = 3L4$ for the moving mass moving at $v = 100ms$, $TTOP = 25^\circ C, TBOT = 25^\circ C, T^\infty = 30^\circ C, \& R = 20m$ for each of the following: a) ut , b) $w(t)$, c) $\theta(t)$, d) $u(t)$ e) $w(t)$, f) $\theta(t)$, g) $u(t)$, h) $w(t)$, and i) $\theta(t)$.
 85

Figure 27: Time histories at $x = L4, x = L2, \& x = 3L4$ for the moving mass moving at $v = 100ms$, $TTOP = 20^\circ C, TBOT = 25^\circ C, T^\infty = 22.5^\circ C, \& R = 20m$ for each of the following: a) ut , b) $w(t)$, c) $\theta(t)$, d) $u(t)$ e) $w(t)$, f) $\theta(t)$, g) $u(t)$, h) $w(t)$, and i) $\theta(t)$.
 86

Figure 28: Time histories at $x = L4, x = L2, \& x = 3L4$ for the moving mass moving at $v = 100ms$, $TTOP = 20^\circ C, TBOT = 25^\circ C, T^\infty = 25^\circ C, \& R = 20m$ for each of the following: a) ut , b) $w(t)$, c) $\theta(t)$, d) $u(t)$ e) $w(t)$, f) $\theta(t)$, g) $u(t)$, h) $w(t)$, and i) $\theta(t)$.
 87

Figure 29: Time histories at $x = L4, x = L2, x = 3L4$, and $y = W2$ for $TTOP = 20^\circ C, TBOT = 20^\circ C$, and $v = 5ms$ for: a) $w(t)$, b) $\theta y(t)$, c) $w(t)$, d) $\theta y(t)$, e) $w(t)$, and f) $\theta y(t)$ 89

Figure 30: Time histories at $x = L4, x = L2, x = 3L4$, and $y = W2$ for $TTOP = 20^\circ C, TBOT = 60^\circ C$, and $v = 5ms$ for: a) $w(t)$, b) $\theta y(t)$, c) $w(t)$, d) $\theta y(t)$, e) $w(t)$, and f) $\theta y(t)$ 90

Figure 31: Time histories at $x = L4, x = L2, x = 3L4$, and $y = W2$ for $TTOP = 60^\circ C, TBOT = 20^\circ C$, and $v = 5ms$ for: a) $w(t)$, b) $\theta y(t)$, c) $w(t)$, d) $\theta y(t)$, e) $w(t)$, and f) $\theta y(t)$ 91

Figure 32: Time histories at $x = L4, x = L2, x = 3L4$, and $y = W2$ for $TTOP = 20^\circ C, TBOT = 20^\circ C$, and $v = 25ms$ for: a) $w(t)$, b) $\theta y(t)$, c) $w(t)$, d) $\theta y(t)$, e) $w(t)$, and f) $\theta y(t)$. . 92

Figure 33: Time histories at $x = L4, x = L2, x = 3L4$, and $y = W2$ for $TTOP = 20^\circ C, TBOT = 60^\circ C$, and $v = 25ms$ for: a) $w(t)$, b) $\theta y(t)$, c) $w(t)$, d) $\theta y(t)$, e) $w(t)$, and f) $\theta y(t)$. . 93

Figure 34: Time histories at $x = L4, x = L2, x = 3L4$, and $y = W2$ for $TTOP = 60^\circ C, TBOT = 20^\circ C$, and $v = 25ms$ for: a) $w(t)$, b) $\theta y(t)$, c) $w(t)$, d) $\theta y(t)$, e) $w(t)$, and f) $\theta y(t)$. . 94

Chapter 1: Introduction

Of particular interest to engineers are the dynamic effects of moving loads on structural elements. Such systems are most commonly applied to, but not limited to, the transient responses of bridges subjected to the vehicles moving along the road deck, trains moving along the tracks, and an aircraft landing on an aircraft carrier, etc. In order for engineers to design the safest possible structures, the worst case loading conditions must be known. Previous attempts to analytically, experimentally, and numerically study structural elements with moving loads are modeled as either straight beams or flat plates. From these studies, it is well known that dynamic loading effects vary greatly when compared to the simple static loading cases.

Analytical solutions are difficult to obtain and show variations in error based on the mathematical model for the moving mass and the magnitude of the load. Jiang [11] shows that two analytical methods, MRRM and MME, show relatively good results for the maximum deflection in a simply supported beam. However, they show inconsistent time histories of the vertical displacement as well as inconsistent error as a function of the velocity of the moving load. Even though these methods appear as a good methodology for solving these engineering problems, they are limited to simple 1D beam models. They cannot be practically applied to a 3D structure like an aircraft carrier deck or bridge.

Experiments can provide accurate and useful information, but they too have limitations. Large scale measurements, such as those by Green and Cebon [8], show good results for displacements at various locations along the span. Unfortunately before these large scale measurements are taken, the structure must be completely constructed. For a designer, this is not practical or economical. Bilello [3] created a small scale experimental method for overcoming the difficulties of high cost and long lead times of full scale tests. The small scale tests also show good results for the time histories of displacement along the beam. It is unknown how good these small scale experiments scale up to full scale as the complexity of the structure increases. They also do not provide a designer the ability to rapidly change the model parameters and recreate the results.

Numerical methods are preferred for solving structural dynamic problems. In the present study, the special discretization is conducted using the Bubnov-Galerkin Finite Element (FE) method . FE has been shown to be effective for solving the forced vibrations of beams due to moving loads. Primarily, most research has focused on straight beams and flat plates [11,15,21,23]. These studies have shown that FE continually shows expected results for simple cases as compared to analytical and experimental methods.

Only one previously published paper has focused on the forced vibration of curved beams by moving loads [20]. This research focused on the application of exact curved beam elements and straight beam elements. The authors came to the conclusion that straight beam elements are sufficient enough for solving these forced vibration problems. The examples shown in this paper show some inconsistencies in the transient solutions. The time histories of various points on the beam experience some form of damping during the free vibration after the force is no longer on the beam even though damping has not been incorporated in the derivation of the FE equations. This

paper also claims that the maximum absolute vertical deflection of the beam goes to zero as the speed of the mass is increased. This phenomenon contradicts all the previously published results for straight beams and flat plates.

This research lays out a method for solving both straight and curved beam problems. FE is used to discretize the Timoshenko beam equations in space and uses the Midpoint-Rule with midpoint acceleration (MPR-MPA) to discretize the equations of motion in time because of its robustness. MPR-MPA is a numerically non-dissipative algorithm [9,17]; hence, no numerical dissipation occurs regardless of the chosen time increment and number of time iterations. Using this algorithm, it can be shown that the physics of the problem is not affected by numerical errors or noise. It is shown that the maximum absolute vertical deflection of curved beams increases as a function of the speed of the moving load, corresponding with the known physics of straight beams with moving loads.

As far as this author knows, no previous research exists to evaluate the effects of combined thermal loading with the forced vibration due to a moving load. It is known that temperature strains can be large and therefore must be included in the design. What is not known is how the magnitude of these thermal effects compare to those of the forced vibration results. A procedure is presented to solve the combined loading case using FE and the MPR-MPA algorithm.

Contributions of this Research

- Define the procedure for using FE and the MPR-MPA algorithm to solve the transient responses of structural elements undergoing moving loads.

- It is well known what effect the speed of a moving mass has on a straight beam.

Investigate the effect the radius of curvature of a beam and the speed of the load has on the maximum deflection of the beam.

- The use of FE and the MPR-MPA algorithm to solve the transient coupled effects of thermal loading and the speed of a moving load on curved beams.
- The use of FE and the MPR-MPA algorithm to solve the transient coupled effects of thermal loading and the speed of a moving load on flat plates.

Chapter 2: Finite Elements for Structural Elements in Space and Time

The Finite Element Method (FE) is a well-known numerical discretization technique used to solve partial differential equations in space and/or time for complicated engineering and mathematics problems. Structural elements fall into this category of problems that are easily solved using FE. Structural elements include beams, frames, and plates among others. Euler-Bernoulli and Timoshenko formulations are used to simplify the linear theory of elasticity to solve beam and frame problems. Likewise, Mindlin-Reissner formulation is used for solving problems involving flat plates with transverse loads. No matter the formulation, FE is widely applied to structures undergoing mechanical and thermal loading.

Numerous algorithms exist in order to solve the discretized FE equations of motion. Two of the most widely known second order, linear multistep (LMS) methods are the Newmark method and the classical version of the Midpoint Rule with end point acceleration (MPR-EPA). Each of these algorithms falls into the General Single Step Single Solve (GSSSS) framework of algorithms [9] along with limitless numbers of other useful and practical optimal algorithms.

The following lays out the steps for discretizing space using FE and time using GSSSS framework algorithms.

2.1 Euler-Bernoulli Beam Mass and Stiffness Formulations

An Euler-Bernoulli (EB) beam is assumed to be a long, slender, homogeneous, straight structural member that bends subject to transverse loading. The governing partial differential equation for a BE beam is defined as

$$\rho A(x) \frac{\partial^2 w(x, t)}{\partial t^2} + EI(x) \frac{\partial^4 w(x, t)}{\partial x^4} = f_{ext}(x, t) \quad (2.1)$$

The discretization of the above PDE begins by creating a two noded elements with two degrees of freedom, $w_i(x, t)$ and $\theta_{yi}(x, t)$, at each node. For an EB beam, shear effects are neglected such that the rotations along the beam are defined as

$$\theta_{yi}(x, t) = -\frac{\partial w_i(x, t)}{\partial x} \quad (2.2)$$

The shape functions for an EB beam are known as the Hermite cubic interpolation functions and are given by

$$\begin{aligned} N_1(x) &= 1 - \frac{3x^2}{L_e^2} + \frac{2x^3}{L_e^3}, & N_2(x) &= -x + \frac{2x^2}{L_e} - \frac{3x^3}{L_e^2} \\ N_3(x) &= \frac{3x^2}{L_e^2} - \frac{2x^3}{L_e^3}, & N_4(x) &= \frac{x^2}{L_e} - \frac{x^3}{L_e^2} \end{aligned} \quad (2.3)$$

The above equations can be rewritten in matrix form as

$$\tilde{w}^h(x, t) = \mathbf{N}_e(x) \mathbf{u}^e(t) \quad (2.4)$$

where the element interpolation function $\mathbf{N}_e(x)$ is defined as

$$\mathbf{N}_e(x) = [N_1(x) \quad N_2(x) \quad N_3(x) \quad N_4(x)] \quad (2.5)$$

and the element displacement vector $\mathbf{u}^e(t)$ is defined as

$$\mathbf{u}^e(t)^T = [w_1(t) \quad \theta_{y1}(t) \quad w_2(t) \quad \theta_{y2}(t)] \quad (2.6)$$

The curvature of the neutral surface of a beam is defined as

$$\kappa_x = -\frac{\partial^2 w^h(x, t)}{\partial x^2} = [B_1(x) \quad B_2(x) \quad B_3(x) \quad B_4(x)] \mathbf{u}^e(t) \quad (2.7)$$

where $B_i(x) = \frac{\partial^2 N_i}{\partial x^2}$, such that

$$\begin{aligned} B_1(x) &= \frac{6}{L_e^2} - \frac{2x}{L_e^3}, & B_2(x) &= -\frac{4}{L_e} + \frac{6x}{L_e^2} \\ B_3(x) &= -\frac{6}{L_e} + \frac{12x}{L_e^3}, & B_4(x) &= -\frac{2}{L_e} + \frac{6x}{L_e^2} \end{aligned} \quad (2.8)$$

The curvature can be rewritten in matrix form as

$$\kappa_x^h(x, t) = \mathbf{B}_e(x) \mathbf{u}^e(t) \quad (2.9)$$

The element stiffness matrix is given as

$$\mathbf{K}^e = \int_{L_e} EI(x) \mathbf{B}_e^T \mathbf{B}_e dx = \frac{EI}{L_e} \begin{bmatrix} 12 & 6L_e & -12 & 6L_e \\ 6L_e & 4L_e^2 & -6L_e & 2L_e^2 \\ -12 & -6L_e & 12 & -6L_e \\ 6L_e & 2L_e^2 & -6L_e & 4L_e^2 \end{bmatrix} \quad (2.10)$$

Assuming that the cross sectional area and density of the element is constant, the consistent element mass matrix $\mathbf{M}^e(x)$ is defined as

$$\mathbf{M}^e = \int_{L_e} \rho A \mathbf{N}_e^T \mathbf{N}_e dx = \frac{\rho A L_e}{420} \begin{bmatrix} 156 & -22L_e & 54 & 13L_e \\ -22L_e & 4L_e^2 & -13L_e & -3L_e^2 \\ 54 & -13L_e & 156 & 22L_e \\ 13L_e & -3L_e^2 & 22L_e & 4L_e^2 \end{bmatrix} \quad (2.11)$$

2.1.1 Numerical Examples

Two numerical examples are used to validate that the stiffness and mass matrices are derived and programmed correctly. The numerical solutions for the static deflection and free vibration of a cantilever beam are compared to the exact, analytical solution. The beam properties are: $L =$

25 in, $A = 0.2 \text{ in}^2$, $EI = 2.5 \times 10^6 \text{ lb} - \text{in}^2$, $\rho = 0.00075 \text{ lb} - \frac{\text{s}^2}{\text{in}^4}$. For the analysis of the static deflection, a force of -250 lb is applied at the tip of the beam. The exact solution for the maximum deflection at the tip of the beam is given by $w_{max} = \frac{FL^3}{3EI}$ [5]. For the analysis of the free vibration of the beam, the first two natural frequencies are given by $\omega_1 = \frac{3.516}{L^2} \sqrt{\frac{EI}{A\rho}}$ and $\omega_2 = 6.2669\omega_1$ [14]. Table 1 shows that the FE solutions approaches the analytical solution of static deflection with just one element because the assumed trial functions approximate the exact analytical solution. The natural frequencies calculated using FE approach the analytical solution as the number of elements increases as expected.

Elements	w_{min} (in)	ω_1 (rad/s)	ω_2 (rad/s)
2	-0.5208333	208.278	1328.344
3	-0.5208333	219.891	1459.749
5	-0.5208333	225.548	1468.698
10	-0.5208333	227.824	1444.642
15	-0.5208333	228.239	1437.990
25	-0.5208333	228.452	1434.380
100	-0.5208333	228.567	1432.525
Exact	-0.5208333	228.575	1432.454

Table 1: Comparison of the FE transverse tip deflection, w_{min} , and the first two natural frequencies, ω_1 and ω_2 for various number of elements.

2.2 Timoshenko Beam Mass and Stiffness Formulations

Timoshenko beam (TB) theory varies from the classical, BE beam theory because transverse shear effects are not neglected in the mathematical model. This allows for Timoshenko beam theory to be used for beams with length to thickness ratios less than 10. The total rotation along the length of the beam is composed of both the slope of the deflection, $\frac{\partial w(x,t)}{\partial x}$, and shear rotation, $\gamma(x,t)$, such that

$$\theta(x, t) = \frac{\partial w(x, t)}{\partial x} + \gamma(x, t) \quad (2.12)$$

Therefore, the bending moment is defined as

$$M(x, t) = EI(x) \frac{\partial \theta_y}{\partial x} \quad (2.13)$$

and the shear force is defined as

$$Q(x, t) = \kappa GA(x, t) \left(\frac{\partial w(x, t)}{\partial x} + \theta_y(x, t) \right) \quad (2.14)$$

where κ is the shear correction factor. For a rectangular cross section, $\kappa = \frac{5}{6}$.

A Timoshenko beam in dynamic equilibrium is governed by two coupled equations because of the two dependent variables, $w(x, t)$ and $\theta(x, t)$. The coupled equations are

$$\rho A(x) \ddot{w}(x, t) - \frac{\partial}{\partial x} \left(\kappa GA(x) \left(\frac{\partial w(x, t)}{\partial x} + \theta_y(x, t) \right) \right) - \bar{w}(x, t) = 0 \quad (2.15)$$

$$\rho I(x) \ddot{\theta}_y(x, t) + \kappa GA(x) \left(\frac{\partial w(x, t)}{\partial x} + \theta_y(x, t) \right) - \frac{\partial}{\partial x} \left(EI(x) \frac{\partial \theta_y(x, t)}{\partial x} \right) = 0 \quad (2.16)$$

with natural boundary conditions defined as

$$M_L = EI(x) \frac{\partial \theta_y}{\partial x} \Big|_{x=L} \quad (2.17)$$

$$Q_L = \kappa GA(x) \left(\frac{\partial w}{\partial x} + \theta_y \right) \Big|_{x=L} \quad (2.18)$$

and essential boundary conditions defined as

$$w|_{x=x_0} = w_0 \quad (2.19)$$

$$\theta_y|_{x=x_0} = \theta_0 \quad (2.20)$$

In order to formulate the FE stiffness and mass matrices for a TB, the first step is to apply the Principle of Virtual Work to the equilibrium equations leading to

$$\int_{\Omega} \left\{ \rho A(x) \ddot{w} - \frac{\partial}{\partial x} \left(\kappa G A(x) \left(\frac{\partial w}{\partial x} + \theta_y \right) \right) - \bar{w}(x) \right\} \delta w d\Omega = 0 \quad (2.21)$$

$$\int_{\Omega} \left\{ \rho I(x) \ddot{\theta}_y + \kappa G A(x) \left(\frac{\partial w}{\partial x} + \theta_y \right) - \frac{\partial}{\partial x} \left(E I(x) \frac{\partial \theta_y}{\partial x} \right) \right\} \delta \theta_y d\Omega = 0 \quad (2.22)$$

These coupled equations are discretized using the variational method. The first step is to create trial functions $\tilde{w}^h(x, t)$ and $\tilde{\theta}_y^h(x, t)$ defined as

$$\tilde{w}^h(x, t) = \sum_{i=1}^n N_i(\xi) w_i(t) = \mathbf{N}_w(\xi) \mathbf{w}^e(t) \quad (2.23)$$

$$\tilde{\theta}_y^h(x, t) = \sum_{i=1}^n N_i(\xi) \theta_{y_i}(t) = \mathbf{N}_{\theta_y}(\xi) \boldsymbol{\theta}_y^e(t) \quad (2.24)$$

where n is the number of nodes per element. A higher order, cubic element is developed here such that each element has 4 equally spaced nodes and shape functions defined as

$$\mathbf{N}_w(\xi) = \mathbf{N}_{\theta_y}(\xi) = [N_1 \quad N_2 \quad N_3 \quad N_4] \quad (2.25)$$

$$N_1 = -\frac{9}{16} (\xi - 1) \left(\xi - \frac{1}{3} \right) \left(\xi + \frac{1}{3} \right) \quad (2.26)$$

$$N_2 = \frac{27}{16} (\xi - 1) (\xi + 1) \left(\xi - \frac{1}{3} \right) \quad (2.27)$$

$$N_3 = -\frac{27}{16} (\xi - 1) (\xi + 1) \left(\xi + \frac{1}{3} \right) \quad (2.28)$$

$$N_4 = \frac{9}{16} (\xi + 1) \left(\xi - \frac{1}{3} \right) \left(\xi + \frac{1}{3} \right) \quad (2.29)$$

$$\mathbf{w}^e(t) = [w_1(t) \quad w_2(t) \quad w_3(t) \quad w_4(t)]^T \quad (2.30)$$

$$\boldsymbol{\theta}_y^e(t) = [\theta_{y1}(t) \quad \theta_{y2}(t) \quad \theta_{y3}(t) \quad \theta_{y4}(t)]^T \quad (2.31)$$

Combining these equations, the two dependent variables are rewritten as

$$\tilde{w}^h(\xi, t) = \mathbf{H}_w(\xi) \mathbf{u}^e(t) \quad (2.32)$$

$$\tilde{\theta}_y^h(\xi, t) = \mathbf{H}_{\theta_y}(\xi) \mathbf{u}^e(t) \quad (2.33)$$

The interpolation functions $\mathbf{H}_w(\xi)$ and $\mathbf{H}_{\theta_y}(\xi)$ are defined as

$$\mathbf{H}_w(\xi) = [N_w \quad 0] \quad (2.34)$$

$$\mathbf{H}_{\theta_y}(\xi) = [0 \quad N_{\theta_y}] \quad (2.35)$$

The nodal variable vector $\mathbf{u}^e(t)$ is defined as

$$\mathbf{u}^e(t) = [\mathbf{w}^e(t) \quad \boldsymbol{\theta}_y^e(t)]^T \quad (2.36)$$

The trial functions can now be rewritten in their final, concise form as

$$\begin{Bmatrix} \tilde{w}^h(\xi, t) \\ \tilde{\theta}_y^h(\xi, t) \end{Bmatrix} = \begin{bmatrix} N_w & 0 \\ 0 & N_{\theta_y} \end{bmatrix} \begin{Bmatrix} w^e(t) \\ \theta_y^e(t) \end{Bmatrix} = \mathbf{H}_e(\xi) \mathbf{u}_e(t) \quad (2.37)$$

where

$$\mathbf{H}_e(\xi) = \begin{bmatrix} \mathbf{H}_w(\xi) \\ \mathbf{H}_{\theta_y}(\xi) \end{bmatrix} \quad (2.38)$$

The bending strain gradient matrix is defined as

$$\mathbf{B}_b(\xi) = \left[\frac{\partial \mathbf{H}_e^{\theta_y}(\xi)}{\partial \xi} \right] \quad (2.39)$$

Therefore, the bending stiffness matrix is defined as

$$\mathbf{K}_b = \int_{\Omega} \mathbf{B}_b^T(\xi) \mathbf{D}_b \mathbf{B}_b(\xi) J(\xi) d\xi \quad (2.40)$$

where $\mathbf{D}_b = EI(x)$ and $J(\xi) = |[J]|$. $[J]$ is the Jacobian matrix for the element.

The shear strain gradient matrix is defined as

$$\mathbf{B}_s(\xi) = \left[\frac{\partial \mathbf{H}_e^w(\xi)}{\partial \xi} + \mathbf{H}_e^{\theta_y}(\xi) \right] \quad (2.41)$$

Therefore, the shear stiffness matrix is defined as

$$\mathbf{K}_s = \int_{\Omega} \mathbf{B}_s^T(\xi) \mathbf{G}_s \mathbf{B}_s(\xi) J(\xi) d\xi \quad (2.42)$$

where $\mathbf{G}_s = \frac{\kappa A(x)E}{2(1+\nu)}$.

The element stiffness matrix is defined as the sum of the bending and shear stiffness matrices,

$$\mathbf{K}^e = \mathbf{K}_b + \mathbf{K}_s \quad (2.43)$$

The element mass matrix is defined as

$$\mathbf{M}^e = \int_{\Omega} \mathbf{H}_e^T(\xi) \boldsymbol{\rho} \mathbf{H}_e(\xi) J(\xi) d\xi \quad (2.44)$$

where $\boldsymbol{\rho}$ is defined as

$$\boldsymbol{\rho} = \begin{bmatrix} \rho A(x) & 0 \\ 0 & \rho I(x) \end{bmatrix} \quad (2.45)$$

For a transversely distributed load, $\bar{\omega}$, across the beam, the element force vector is defined as

$$\mathbf{F}_{ext}^e = \int_{\Omega} \bar{\omega}(\xi) \mathbf{H}_w^T(\xi) J(\xi) d\xi \quad (2.46)$$

For the TB elements, exact integration is not performed. Instead, isoperimetric integration is used to formulate the FE matrices. This is because Selective Reduced Integration (SRI) can now be used to reduce shear locking [2]. Shear locking gives rise to numerical error and causes the elements to become magnitudes stiffer. When Gaussian Integration is used for calculating the shear stiffness matrix, \mathbf{K}_s , one less Gauss point is used. The bending matrix, however, is still integrated using the exact Gauss points. The combination of cubic interpolation functions and SRI help combat the effects of locking.

2.2.1 Numerical Examples

Two models are used to validate that the stiffness and mass matrices are derived and programmed correctly. The numerical solutions for the static deflection and free vibration of a simply supported beam are compared to the exact, analytical solutions. The beam properties are: $L = 1 \text{ m}$, $E = 210 \text{ GPa}$, $\rho = 7800 \frac{\text{kg}}{\text{m}^3}$. Two different heights of beams are analyzed, $\frac{L}{h} = 10$ & $\frac{L}{h} = 4$. For the static deflection, a line pressure of $-100 \frac{\text{kN}}{\text{m}}$ is applied along the length of the beam. The maximum deflection of the beam is given by [18]. For the free vibration of the beam, the exact non-

dimensional fundamental frequency is given by [16]. The non-dimensional fundamental frequency is defined as

$$\bar{\omega}_w = \omega_1 \left(\frac{L^2}{h} \right) \sqrt{\frac{\rho}{E}} \quad (2.47)$$

Table 2 shows that the FE solution approaches the analytical solution of static deflection and fundamental frequency as the number of elements is increased.

Elements	L/h = 4		L/h = 10	
	w_{min} (mm)	$\bar{\omega}_w$	w_{min} (mm)	$\bar{\omega}_w$
1	-0.018895	13.856	-0.651429	34.641
3	-0.021697	2.942	-0.751244	3.221
5	-0.021904	2.713	-0.758552	2.943
10	-0.022019	2.624	-0.762619	2.836
15	-0.022006	2.608	-0.762169	2.817
25	-0.022014	2.600	-0.762457	2.808
100	-0.022019	2.596	-0.762619	2.803
Exact	-0.002202	2.596	-0.762619	2.802

Table 2: Comparison of the transverse deflection, w_{min} , at the center of the beam and the first non-dimensional first natural frequency for two cases: (Left) L/h=4; (Right) L/h=10.

2.3 Mindlin-Reissner Plate Mass and Stiffness Formulations

Mindlin-Reissner (MR) plate theory is the two dimensional equivalence of Timoshenko beam theory. MR plate theory differs from classical plate theory, Kirchoff-Love plate theory, by including the effects of transverse shear deformation through the thickness of the plate. The rotations are defined as

$$\theta_x = - \left(\frac{\partial w_0}{\partial x} - \gamma_{xz} \right) \quad (2.48)$$

$$\theta_y = \frac{\partial w_0}{\partial y} - \gamma_{yz} \quad (2.49)$$

Therefore, the bending moments are defined as

$$\mathbf{m} = \begin{pmatrix} M_{xx} \\ M_{yy} \\ M_{xy} \end{pmatrix} = \mathbf{D}\boldsymbol{\kappa} \quad (2.50)$$

$$\mathbf{D} = \frac{Eh^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \text{ and } \boldsymbol{\kappa} = \begin{pmatrix} \frac{\partial\theta_y}{\partial x} \\ -\frac{\partial\theta_x}{\partial y} \\ \frac{\partial\theta_y}{\partial y} - \frac{\partial\theta_x}{\partial x} \end{pmatrix} \quad (2.51)$$

The shear forces per unit length are defined as

$$\mathbf{Q} = \begin{pmatrix} Q_x \\ Q_y \end{pmatrix} = \begin{pmatrix} \kappa Gh \left(\frac{\partial w}{\partial x} + \theta_y \right) \\ \kappa Gh \left(\frac{\partial w}{\partial y} - \theta_x \right) \end{pmatrix} \quad (2.52)$$

where κ is the shear correction factor. For a rectangular plate with uniform thickness, $\kappa = \frac{\pi}{\sqrt{12}}$

A Reissner-Mindlin plate in dynamic equilibrium is governed by three coupled equations because of the three dependent variables, $w(x, y, t)$, $\theta_x(x, y, t)$ and $\theta_y(x, y, t)$. These equations are

$$\rho h \ddot{w} - \kappa Gh \frac{\partial}{\partial x} \left(\frac{\partial w}{\partial x} + \theta_y \right) - \kappa Gh \frac{\partial}{\partial y} \left(\frac{\partial w}{\partial y} - \theta_x \right) - \bar{w}(x, y) = 0 \quad (2.53)$$

$$\rho I_p \ddot{\theta}_x - \kappa Gh \left(\frac{\partial w}{\partial y} + \theta_x \right) - \frac{1}{2} D (1 - \nu) \left(\frac{\partial^2 \theta_x}{\partial x^2} + \frac{\partial^2 \theta_x}{\partial y^2} \right) + \frac{1}{2} D (1 + \nu) \left(\frac{\partial^2 \theta_x}{\partial y^2} - \frac{\partial^2 \theta_y}{\partial x \partial y} \right) = 0 \quad (2.54)$$

$$\rho I_p \ddot{\theta}_y - \kappa Gh \left(\frac{\partial w}{\partial x} + \theta_y \right) - \frac{1}{2} D (1 - \nu) \left(\frac{\partial^2 \theta_y}{\partial x^2} + \frac{\partial^2 \theta_y}{\partial y^2} \right) + \frac{1}{2} D (1 + \nu) \left(\frac{\partial^2 \theta_y}{\partial x^2} - \frac{\partial^2 \theta_x}{\partial x \partial y} \right) = 0 \quad (2.55)$$

with natural boundary conditions

$$Q_x + \frac{\partial M_{xy}}{\partial y} = \bar{V}_x \quad (2.56)$$

$$Q_y + \frac{\partial M_{xy}}{\partial x} = \bar{V}_y \quad (2.57)$$

$$M_x = \bar{M}_x, \quad M_y = \bar{M}_y, \quad M_{nn} = \bar{M}_{nn} \quad (2.58)$$

and essential boundary conditions

$$w(x, y, t) = \bar{w}, \quad \frac{\partial w}{\partial x} = \bar{\psi}_x, \quad \frac{\partial w}{\partial y} = \bar{\psi}_y \quad (2.59)$$

The FE formulation begins by applying the Principle of Virtual Work to the equilibrium equations leading to

$$\int_{\Omega} \left(\rho h \dot{w} - \kappa G h \frac{\partial}{\partial x} \left(\frac{\partial w}{\partial x} + \theta_y \right) - \kappa G h \frac{\partial}{\partial y} \left(\frac{\partial w}{\partial y} - \theta_x \right) - \bar{w}(x, y) \right) \delta w d\Omega = 0 \quad (2.60)$$

$$\int_{\Omega} \left(\rho I_p \ddot{\theta}_x - \kappa G h \left(\frac{\partial w}{\partial y} + \theta_x \right) - \frac{1}{2} D(1-\nu) \left(\frac{\partial^2 \theta_x}{\partial x^2} + \frac{\partial^2 \theta_x}{\partial y^2} \right) + \frac{1}{2} D(1+\nu) \left(\frac{\partial^2 \theta_x}{\partial y^2} - \frac{\partial^2 \theta_y}{\partial x \partial y} \right) \right) \delta \theta_x d\Omega = 0 \quad (2.61)$$

$$\int_{\Omega} \left(\rho I_p \ddot{\theta}_y - \kappa G h \left(\frac{\partial w}{\partial x} + \theta_y \right) - \frac{1}{2} D(1-\nu) \left(\frac{\partial^2 \theta_y}{\partial x^2} + \frac{\partial^2 \theta_y}{\partial y^2} \right) + \frac{1}{2} D(1+\nu) \left(\frac{\partial^2 \theta_y}{\partial x^2} - \frac{\partial^2 \theta_x}{\partial x \partial y} \right) \right) \delta \theta_y d\Omega = 0 \quad (2.62)$$

These coupled equations are discretized using the variational method. The first step is to create trial functions $\tilde{w}^h(x, y, t)$, $\tilde{\theta}_x^h(x, y, t)$, and $\tilde{\theta}_y^h(x, y, t)$ defined as

$$\tilde{w}^h(x, y, t) = \sum_{i=1}^n N_i(\xi, \eta) w_i(t) = \mathbf{N}_w(\xi, \eta) \mathbf{w}^e(t) \quad (2.63)$$

$$\tilde{\theta}_x^h(x, y, t) = \sum_{i=1}^n N_i(\xi, \eta) w_i(t) = \mathbf{N}_{\theta_x}(\xi, \eta) \boldsymbol{\theta}_x^e(t) \quad (2.64)$$

$$\tilde{\theta}_y^h(x, y, t) = \sum_{i=1}^n N_i(\xi, \eta) w_i(t) = \mathbf{N}_{\theta_y}(\xi, \eta) \boldsymbol{\theta}_y^e(t) \quad (2.65)$$

where n is the number of nodes per element.

For a 4 node, bilinear quadrilateral, $n = 4$, the interpolation functions are

$$\mathbf{N}_w(\xi, \eta) = \mathbf{N}_{\theta_x}(\xi, \eta) = \mathbf{N}_{\theta_y}(\xi, \eta) = \begin{bmatrix} \frac{1}{4}(1-\xi)(1-\eta) & \frac{1}{4}(1+\xi)(1-\eta) & \frac{1}{4}(1+\xi)(1+\eta) & \frac{1}{4}(1-\xi)(1+\eta) \end{bmatrix} \quad (2.66)$$

$$\mathbf{w}^e(t) = [w_1(t) \quad w_2(t) \quad w_3(t) \quad w_4(t)]^T \quad (2.67)$$

$$\boldsymbol{\theta}_x^e(t) = [\theta_{x1}(t) \quad \theta_{x2}(t) \quad \theta_{x3}(t) \quad \theta_{x4}(t)]^T \quad (2.68)$$

$$\boldsymbol{\theta}_y^e(t) = [\theta_{y1}(t) \quad \theta_{y2}(t) \quad \theta_{y3}(t) \quad \theta_{y4}(t)]^T \quad (2.69)$$

Combining these equations, the three dependent variables are rewritten as

$$\tilde{w}^h(\xi, \eta, t) = \mathbf{H}_w(\xi, \eta) \mathbf{u}^e(t) \quad (2.70)$$

$$\tilde{\theta}_x^h(\xi, \eta, t) = \mathbf{H}_{\theta_x}(\xi, \eta) \mathbf{u}^e(t) \quad (2.71)$$

$$\tilde{\theta}_y^h(\xi, \eta, t) = \mathbf{H}_{\theta_y}(\xi, \eta) \mathbf{u}^e(t) \quad (2.72)$$

The interpolation functions, $\mathbf{H}_w(\xi, \eta)$, $\mathbf{H}_{\theta_x}(\xi, \eta)$, and $\mathbf{H}_{\theta_y}(\xi, \eta)$, are defined as

$$\mathbf{H}_w(\xi, \eta) = [N_w \quad 0 \quad 0] \quad (2.73)$$

$$\mathbf{H}_{\theta_x}(\xi, \eta) = [0 \quad N_{\theta_x} \quad 0] \quad (2.74)$$

$$\mathbf{H}_{\theta_y}(\xi, \eta) = [0 \quad 0 \quad N_{\theta_y}] \quad (2.75)$$

The nodal variable vector $\mathbf{u}^e(t)$ is defined as

$$\mathbf{u}^e(t) = [\mathbf{w}_t^e(t) \quad \boldsymbol{\theta}_x^e(t) \quad \boldsymbol{\theta}_y^e(t)]^T \quad (2.76)$$

The trial functions can now be rewritten in their final, concise form as

$$\begin{Bmatrix} \tilde{w}^h(\xi, \eta, t) \\ \tilde{\theta}_x^h(\xi, \eta, t) \\ \tilde{\theta}_y^h(\xi, \eta, t) \end{Bmatrix} = \begin{bmatrix} N_w & 0 & 0 \\ 0 & N_{\theta_x} & 0 \\ 0 & 0 & N_{\theta_y} \end{bmatrix} \begin{Bmatrix} \mathbf{w}_t^e(t) \\ \boldsymbol{\theta}_x^e(t) \\ \boldsymbol{\theta}_y^e(t) \end{Bmatrix} = \mathbf{H}_e(\xi, \eta) \mathbf{u}_e(t) \quad (2.77)$$

where

$$\mathbf{H}_e(\xi) = \begin{bmatrix} \mathbf{H}_w(\xi, \eta) \\ \mathbf{H}_{\theta_x}(\xi, \eta) \\ \mathbf{H}_{\theta_y}(\xi, \eta) \end{bmatrix} \quad (2.78)$$

The bending strain matrix is defined as

$$\mathbf{B}_b(\xi, \eta) = \begin{bmatrix} \frac{\partial \mathbf{H}_e^{\theta_y}(\xi, \eta)}{\partial \xi} \\ -\frac{\partial \mathbf{H}_e^{\theta_x}(\xi, \eta)}{\partial \eta} \\ -\frac{\partial \mathbf{H}_e^{\theta_x}(\xi, \eta)}{\partial \xi} + \frac{\partial \mathbf{H}_e^{\theta_y}(\xi, \eta)}{\partial \eta} \end{bmatrix} \quad (2.79)$$

Therefore, the bending stiffness matrix is defined as

$$\mathbf{K}_b = \int_{\Omega} \mathbf{B}_b^T(\xi, \eta) \mathbf{D}_b \mathbf{B}_b(\xi, \eta) J(\xi, \eta) d\xi d\eta \quad (2.80)$$

where

$$\mathbf{D}_b = \frac{Eh^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \quad (2.81)$$

The shear strain matrix is defined as

$$\mathbf{B}_s(\xi, \eta) = \begin{bmatrix} \frac{\partial \mathbf{H}_e^w(\xi, \eta)}{\partial \xi} + \mathbf{H}_e^{\theta_y}(\xi, \eta) \\ \frac{\partial \mathbf{H}_e^w(\xi, \eta)}{\partial \eta} - \mathbf{H}_e^{\theta_x}(\xi, \eta) \end{bmatrix} \quad (2.82)$$

Therefore, the shear stiffness matrix is defined as

$$\mathbf{K}_s = \int_{\Omega} \mathbf{B}_s^T(\xi, \eta) \mathbf{G}_s \mathbf{B}_s(\xi, \eta) J(\xi, \eta) d\xi d\eta \quad (2.83)$$

where

$$\mathbf{G}_s = \frac{\kappa AE}{2(1+\nu)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.84)$$

The element stiffness matrix is defined as the sum of the bending and shear stiffness matrices,

$$\mathbf{K}^e = \mathbf{K}_b + \mathbf{K}_s \quad (2.85)$$

The element mass matrix is defined as

$$\mathbf{M}^e = \int_{\Omega} \mathbf{H}_e^T(\xi, \eta) \boldsymbol{\rho} \mathbf{H}_e(\xi, \eta) J(\xi, \eta) d\xi d\eta \quad (2.86)$$

where $\boldsymbol{\rho}$ is defined as

$$\boldsymbol{\rho} = \begin{bmatrix} \rho h & 0 & 0 \\ 0 & \rho I & 0 \\ 0 & 0 & \rho I \end{bmatrix} \quad (2.87)$$

Therefore, the consistent mass matrix is defined as

$$\mathbf{M}^e = \begin{bmatrix} \rho h N_w^T N_w & 0 & 0 \\ 0 & \rho I N_{\theta_x}^T N_{\theta_x} & 0 \\ 0 & 0 & \rho I N_{\theta_y}^T N_{\theta_y} \end{bmatrix} \quad (2.88)$$

For a transversely distributed load, $\bar{\omega}$, across the plate, the element force vector is defined as

$$\mathbf{F}_{ext}^e = \int_{\Omega} \bar{\omega}(\xi, \eta) \mathbf{H}_e^T(\xi, \eta) J(\xi, \eta) d\xi d\eta \quad (2.89)$$

Just like with the Timoshenko beam mass and stiffness matrices, Gaussian integration is used instead of exact integration. Again, like with the TB formulation, Gaussian integration is used in order to reduce the effects of shear locking using SRI.

2.3.1 Numerical Examples

Two numerical examples are used to validate that the stiffness and mass matrices are derived and programmed correctly. The numerical solutions for the static deflection and free vibration of a plate are compared to the exact, analytical solution. The plate properties are: $L = W = 1 \text{ mm}$, $E = 10920 \frac{N}{mm}$, $\rho = 1 \frac{kg}{mm^3}$, $\nu = 0.30$, and $h = 0.1 \text{ mm}$. For the static deflection, a pressure of $-100 \frac{kN}{mm^2}$ is applied on the top surface of the beam. The maximum vertical deflection is given in [7]. For the free vibration of the plate, the first natural frequency is given in [6]. Table 3 shows that the maximum deflection and the frequency calculated using FE approach the analytical solution as the number of elements increases as expected.

Elements	\bar{w}_{min}	$\bar{\omega}_w$
4	-0.003517	1.216
16	-0.004188	0.993
36	-0.004228	0.957
64	-0.004240	0.945
100	-0.004246	0.940
144	-0.004249	0.937
400	-0.004253	0.933
Exact	-0.004270	0.930

Table 3: Comparison of the non-dimensional FE transverse deflection, \bar{w}_{min} , at the center of the plate and the non-dimensional natural frequency, $\bar{\omega}_w$ for various number of elements.

2.4 Numerical Time Integration

The previous sections have detailed how to discretize partial differential equations in space giving rise to ODE's in time. In order to discretize these PDEs in time, a time stepping algorithm is needed. Various second order, LMS algorithms exist in literature and commercial software. The

two most commonly known second order algorithms are the Newmark average acceleration method and the Classical Midpoint Rule with end-point acceleration (MPR-EPA) [9]. However, these are not the only implicit, second order accurate, LMS methods. All of the LMS algorithms, including the Newmark method and MPR-EPA, are found within the unified framework of the Generalized Single Step Single Solve (GSSSS) family of algorithms. Within the GSSSS framework, there exist two families of algorithms, U0 and V0. The algorithms in the U0 family exhibit zero-order displacement overshoot. Likewise, the algorithms in the V0 family exhibit zero-order overshoot in velocity.

First consider a second order differential equation of motion as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}(t) \quad (2.90)$$

subject to the initial conditions

$$\dot{\mathbf{u}}(0) = \dot{\mathbf{u}}_0 \quad (2.91)$$

$$\mathbf{u}(0) = \mathbf{u}_0 \quad (2.92)$$

\mathbf{M} is the mass matrix. \mathbf{C} is the damping matrix. \mathbf{K} is the stiffness matrix. \mathbf{f} is the external force vector. The V0 family is defined as

$$\begin{aligned} & (\Lambda_6 W_1 \mathbf{M} + \Lambda_5 W_2 \mathbf{C} \Delta t + \Lambda_3 W_3 \mathbf{K} \Delta t^2) \Delta \ddot{\mathbf{u}} = \\ & -\mathbf{M} \ddot{\mathbf{u}}_n - \mathbf{C}(\dot{\mathbf{u}}_n + \Lambda_4 W_1 \ddot{\mathbf{u}}_n \Delta t) - \mathbf{K}(\mathbf{u}_n + \Lambda_1 W_1 \dot{\mathbf{u}}_n \Delta t + \Lambda_2 W_2 \ddot{\mathbf{u}}_n \Delta t^2) \end{aligned} \quad (2.93)$$

$$+(1 - W_1) \mathbf{f}_n + W_1 \mathbf{f}_{n+1}$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \lambda_1 \dot{\mathbf{u}}_n \Delta t + \lambda_2 \ddot{\mathbf{u}}_n \Delta t^2 + \lambda_3 \Delta \ddot{\mathbf{u}} \Delta t^2 \quad (2.94)$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \lambda_4 \ddot{\mathbf{u}}_n \Delta t + \lambda_5 \Delta \ddot{\mathbf{u}} \Delta t \quad (2.95)$$

$$\ddot{\mathbf{u}}_{n+1} = \ddot{\mathbf{u}}_n + \Delta \ddot{\mathbf{u}} \quad (2.96)$$

where

$$W_1\Lambda_1 = \frac{3 + \rho_{1\infty} + \rho_{2\infty} - \rho_{1\infty}\rho_{2\infty}}{2(1 + \rho_{1\infty})(1 + \rho_{2\infty})}, \quad \lambda_1 = 1 \quad (2.97)$$

$$W_2\Lambda_2 = \frac{1}{(1 + \rho_{1\infty})(1 + \rho_{2\infty})}, \quad \lambda_2 = \frac{1}{2} \quad (2.98)$$

$$W_3\Lambda_3 = \frac{1}{(1 + \rho_{1\infty})(1 + \rho_{2\infty})(1 + \rho_{3\infty})}, \quad \lambda_3 = \frac{1}{2(1 + \rho_{3\infty})} \quad (2.99)$$

$$W_1\Lambda_4 = \frac{3 + \rho_{1\infty} + \rho_{2\infty} - \rho_{1\infty}\rho_{2\infty}}{2(1 + \rho_{1\infty})(1 + \rho_{2\infty})}, \quad \lambda_4 = 1 \quad (2.100)$$

$$W_2\Lambda_5 = \frac{2}{(1 + \rho_{1\infty})(1 + \rho_{2\infty})(1 + \rho_{3\infty})}, \quad \lambda_5 = \frac{1}{1 + \rho_{3\infty}} \quad (2.101)$$

$$W_1\Lambda_6 = \frac{2 + \rho_{1\infty} + \rho_{2\infty} + \rho_{3\infty} - \rho_{1\infty}\rho_{2\infty}\rho_{3\infty}}{(1 + \rho_{1\infty})(1 + \rho_{2\infty})(1 + \rho_{3\infty})} \quad (2.102)$$

$\rho_{1\infty}$, $\rho_{2\infty}$, and $\rho_{3\infty}$ are the first principal root, the second principal root, and the spurious root that must satisfy

$$0 \leq \rho_{3\infty} \leq \rho_{1\infty} \leq \rho_{2\infty} \leq 1 \quad (2.103)$$

Some of the most common algorithms within the V0 family are shown in Table 4.

The Raleigh damping matrix \mathbf{C} is defined by the proportional damping coefficients α and β such that

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K} \quad (2.104)$$

$$\alpha = \frac{2\omega_1\omega_2(\xi_1\omega_2 - \xi_2\omega_1)}{\omega_2^2 - \omega_1^2} \quad \text{and} \quad \beta = \frac{2(\xi_2\omega_2 - \xi_1\omega_1)}{\omega_2^2 - \omega_1^2} \quad (2.105)$$

where ω_1 and ω_2 are the first two natural frequencies, and ξ_1 and ξ_2 are the corresponding damping ratios. The first two natural frequencies are calculated using

$$\mathbf{K}\phi = \lambda\mathbf{M}\phi \quad (2.106)$$

Algorithms Family	Common Name	Conditions
$\mathbf{V0}(\rho_{1\infty}, \mathbf{1}, \rho_{3\infty})$	U0-V0 Optimal	$\rho_{1\infty} = \rho_{3\infty} = \rho_{\infty}; 0 \leq \rho_{\infty} \leq 0$
$\mathbf{V0}(\mathbf{1}, \mathbf{1}, \mathbf{1})$	MPR-EPA	-
$\mathbf{V0}(\mathbf{1}, \mathbf{1}, \mathbf{0})$	MPR-MPA	-

Table 4: Common GSSSS V0 algorithms.

The MPR-MPA algorithm is equivalent to the GSSSS V0(1,1,0) algorithm. The MPR-MPA algorithm is defined by the equation

$$\begin{aligned} & \left(\mathbf{M} + \frac{1}{2}\mathbf{C}\Delta t + \frac{1}{4}\mathbf{K}\Delta t^2\right)\Delta\dot{\mathbf{u}} = \\ & -\mathbf{M}\ddot{\mathbf{u}}_n - \mathbf{C}\left(\dot{\mathbf{u}}_n + \frac{1}{2}\ddot{\mathbf{u}}_n\Delta t\right) - \mathbf{K}\left(\mathbf{u}_n + \frac{1}{2}\dot{\mathbf{u}}_n\Delta t + \frac{1}{4}\ddot{\mathbf{u}}_n\Delta t^2\right) \\ & + \frac{1}{2}(\mathbf{f}_n + \mathbf{f}_{n+1}) \end{aligned} \quad (2.107)$$

with the following updates

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \dot{\mathbf{u}}_n\Delta t + \frac{1}{2}\ddot{\mathbf{u}}_n\Delta t^2 + \frac{1}{2}\Delta\dot{\mathbf{u}}\Delta t^2 \quad (2.108)$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \ddot{\mathbf{u}}_n\Delta t + \frac{1}{2}\Delta\ddot{\mathbf{u}}\Delta t \quad (2.109)$$

$$\ddot{\mathbf{u}}_{n+1} = \ddot{\mathbf{u}}_n + \Delta\ddot{\mathbf{u}} \quad (2.110)$$

All of the examples presented in this research are linear dynamical systems. Because of this, any of the V0(1,1, ρ_3) and the Newmark Method, GSSSS U0(1,1,0), algorithms are suitable for use in solving these problems. The MPR-MPA algorithm is selected out of this group for use in this research. For further research into structural elements with moving loads, MPR-MPA will be preferred. Of the numerically non-dissipative algorithms, MPR-MPA is robust especially for non-linear dynamical systems.

Chapter 3: Dynamic Response of Structural Elements Subject to Moving Loads Using Finite Elements

The dynamic response of moving loads on straight beams and flat plates is not new to science. Scientists and engineers have solved these problems using analytical methods [11], experimental methods [3,8], and numerical methods [11,15,21]. For straight beams and flat plates, these methods have been shown effective for solving the dynamic responses. Of these, none of the published numerical methods have used the MPR-MPA algorithm for solving the transient responses.

Three different FE formulations can be used for solving these problems, Euler-Bernoulli beam formulation, Timoshenko beam formulation, and Mindlin-Reissner plate formulation. Euler-Bernoulli beam formulation is limited to the fact that the ratio of the beam length to height ratio must be sufficiently large because shear effects are neglected in the formulation. Timoshenko beam formulation is valid for any 1D case in which the loading is constant over the width of the beam. Mindlin-Reissner plate formulation is the most encompassing because the load can be located at any point on the width of the beam, i.e. eccentric loading.

3.1 Finite Element Formulation of a Moving Load

The moving load is assumed to be a concentrated mass acting at a single point. Any inertial effects and friction effects are neglected. This point force is caused by the concentrated mass and is given by

$$f_m(x, t) = W_m \delta[x - x_m(t)] \quad (3.1)$$

where W_m is the weight of the moving mass such that $W_m = -mg$. x_m is the location of the mass at given time, t , is a function of the velocity of the mass

$$x_m(t) = v_m t \quad (3.2)$$

At a given time, t , the mass is located at l_m on element, e . The location of the mass relative to the length of the element is defined as

$$\eta(t) = \frac{l_m}{L_e} \quad (3.3)$$

For the BE formulation, the discretized force vector is defined as [14]

$$\widetilde{f}_m(t) = W \left\{ \begin{array}{l} (2\eta + 1)(1 - \eta)^2 \\ L_e(1 - \eta)^2 \\ \eta^2(3 - 2\eta) \\ L_e\eta(1 - \eta) \end{array} \right\} \quad (3.4)$$

For a four noded TB element, the discretized force vector is defined as

$$\widetilde{f}_m(t) = W\mathbf{H}_w(\eta) = W \begin{pmatrix} -\frac{9\eta^3}{2} + 9\eta^2 - \frac{11\eta}{2} + 1 \\ 0 \\ \frac{9\eta(3\eta^2 - 5\eta + 2)}{2} \\ 0 \\ -\frac{9\eta(3\eta^2 - 4\eta + 1)}{2} \\ 0 \\ \frac{\eta(9\eta^2 - 9\eta + 2)}{2} \\ 0 \end{pmatrix} \quad (3.5)$$

For the four node Mindlin-Reissner plate element, the load is assumed to be a line pressure, $p = -\frac{mg}{w}$. The location of the load relative to the length in the x-direction of the element, L_x , is defined as

$$\eta(t) = \frac{l_{xm}}{L_x} \quad (3.6)$$

The discretized force vector is defined as

$$\widetilde{f}_m(t) = pL_y\mathbf{H}_w(\eta) = \frac{pL_y}{2} \begin{pmatrix} 1 - \eta \\ 0 \\ 0 \\ \eta \\ 0 \\ 0 \\ \eta \\ 0 \\ 0 \\ 0 \\ 1 - \eta \\ 0 \\ 0 \end{pmatrix} \quad (3.7)$$

3.2 Numerical Examples

The transverse deflection of each of these three formulations is compared to published experimental results [3]. A simply supported rectangular beam is subject to a moving mass, $m = 0.505kg$,

moving at a constant velocity, $v = 7.59 \frac{km}{h}$. The aluminum beam has a length, $L = 1071.5mm$, width, $w = 105.25mm$, and a height, $h = 6.35mm$. The material properties are Young's modulus and area moment of inertia, $EI = 162.6Nm^2$, Poisson's ratio, $\nu = 0.30$, and density, $\rho = 2764 \frac{kg}{m^3}$. The proportional damping ratios are assumed to be $\xi_1 = \xi_2 = 0.05$. Using FE, the first two natural frequencies are calculated as $\omega_1 = 81 \frac{rad}{s}$ and $\omega_2 = 322 \frac{rad}{s}$.

For the beam formulations, the essential boundary conditions for a simply supported beam are

$$w(t)|_{x=0} = 0 \quad (3.8)$$

$$w(t)|_{x=L} = 0 \quad (3.9)$$

and the natural boundary conditions are

$$M(t)|_{x=0} = 0 \quad (3.10)$$

$$M(t)|_{x=L} = 0 \quad (3.11)$$

For the plate formulation, the essential boundary conditions for hard simply supported edges

$$w(t)|_{x=0} = 0, w(t)|_{x=L} = 0 \quad (3.12)$$

$$\theta_x(t)|_{x=0} = 0, \theta_x(t)|_{x=L} = 0 \quad (3.13)$$

and the natural boundary conditions are

$$M(t)|_{x=0} = 0 \quad (3.14)$$

$$M(t)|_{x=L} = 0 \quad (3.15)$$

The structures are initially at rest, so the initial conditions are

$$\mathbf{u}(0) = \dot{\mathbf{u}}(0) = 0 \quad (3.16)$$

Figure 1 shows the comparison of the transverse deflection at $x = \frac{7L}{16}$ for each of the FE formulations compared to the experimental results. Figure 2-Figure 4 show the time histories

calculated using each of the three formulations for each degree of freedom at $x = \frac{7L}{16}$. Each of the

FE formulations agree closely with the experimental results.

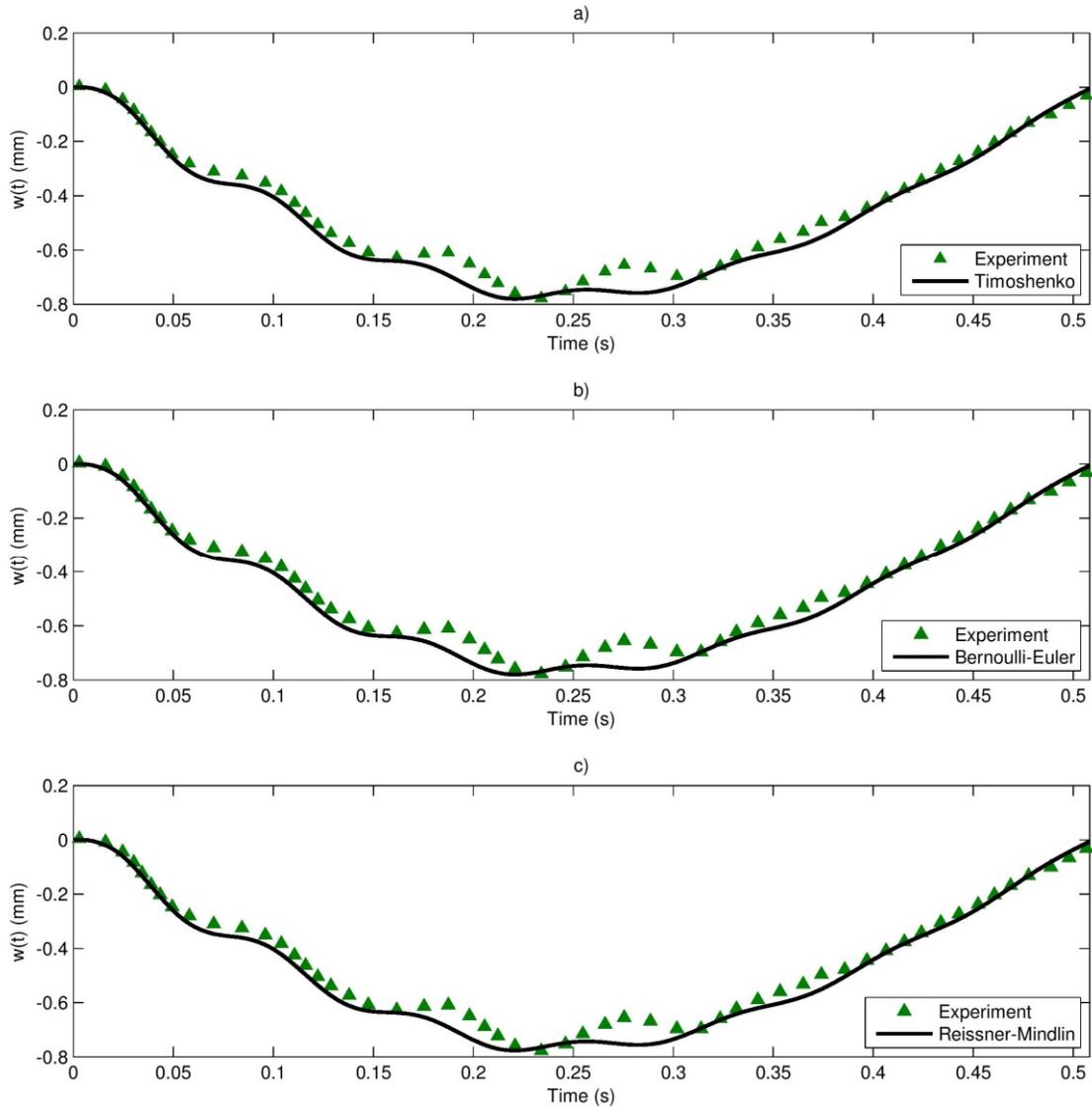


Figure 1: Transverse displacement, w , at $x = \frac{7L}{16}$ for three FE formulations: a) Timoshenko beam, b) Euler-Bernoulli beam, and c) Mindlin-Reissner plate as compared to the experimental measurements.

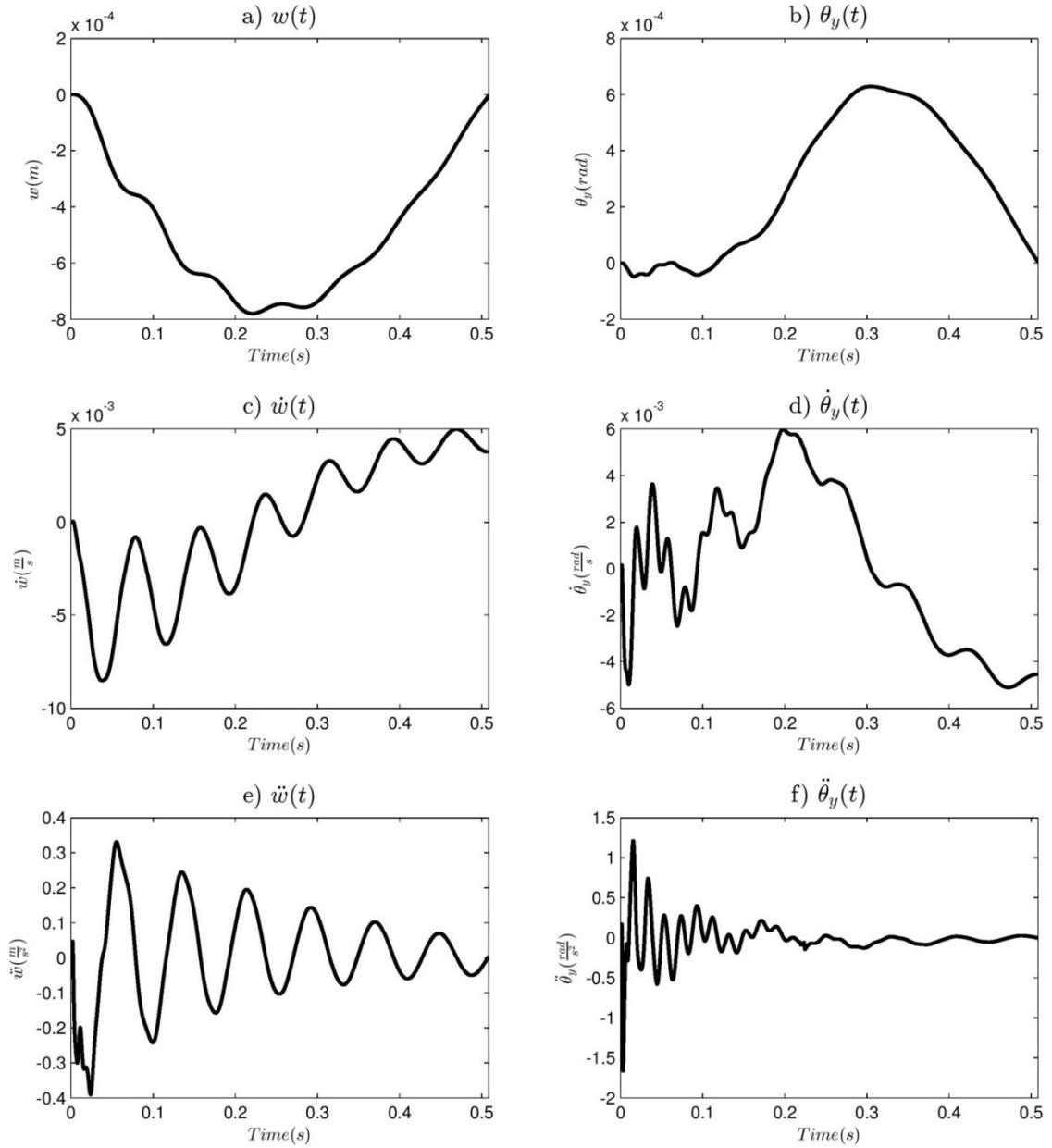


Figure 2: Time histories at $x = \frac{7L}{16}$ calculated using Timoshenko beam theory and V0(1,1,0) for:
a) $w(t)$, b) $\theta_y(t)$, c) $\dot{w}(t)$, d) $\dot{\theta}_y(t)$, e) $\ddot{w}(t)$, and f) $\ddot{\theta}_y(t)$.

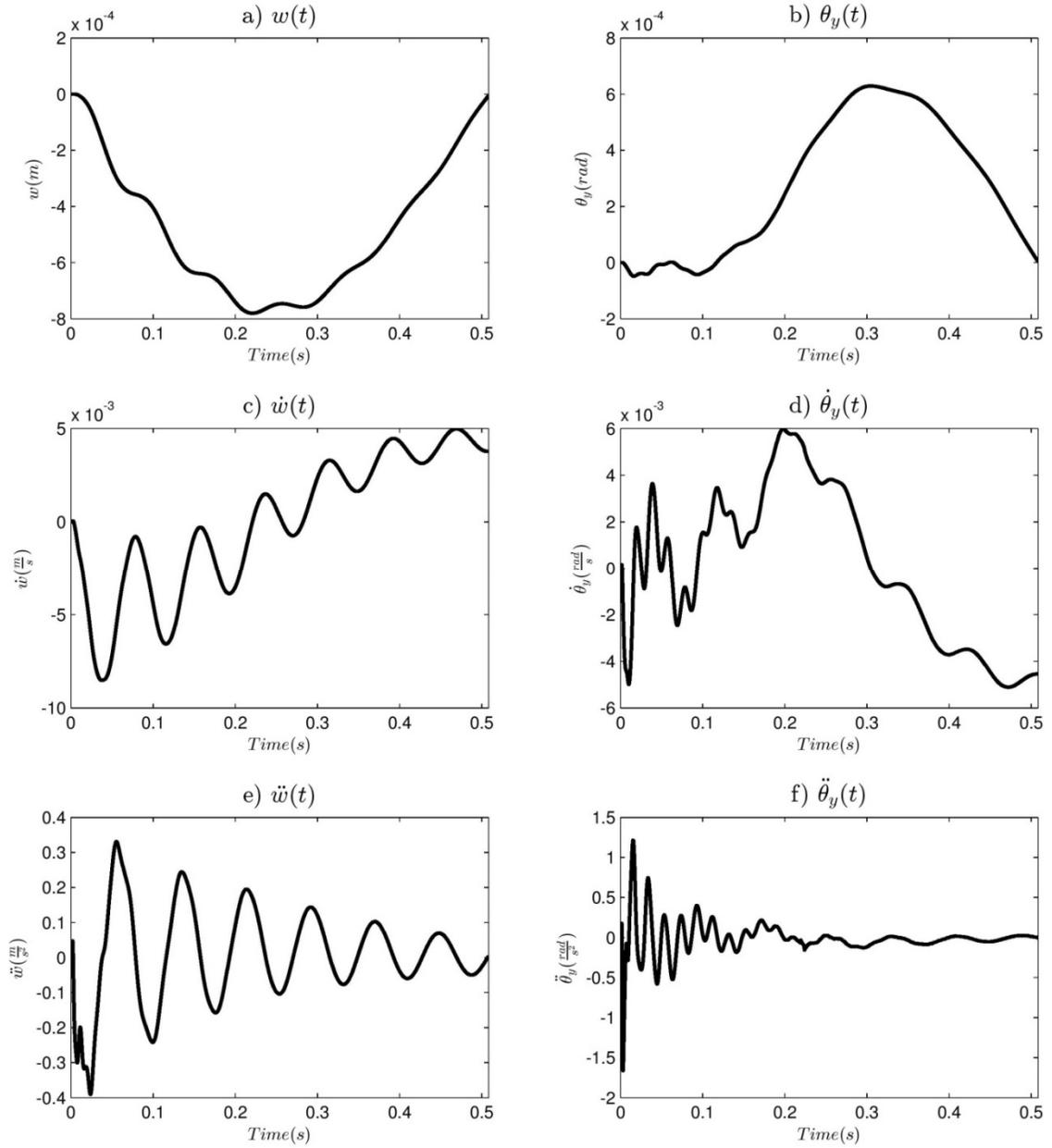


Figure 3: Time histories at $x = \frac{7L}{16}$ calculated using Euler-Bernoulli beam theory and V0(1,1,0) for: a) $w(t)$, b) $\theta_y(t)$, c) $\dot{w}(t)$, d) $\dot{\theta}_y(t)$, e) $\ddot{w}(t)$, and f) $\ddot{\theta}_y(t)$.

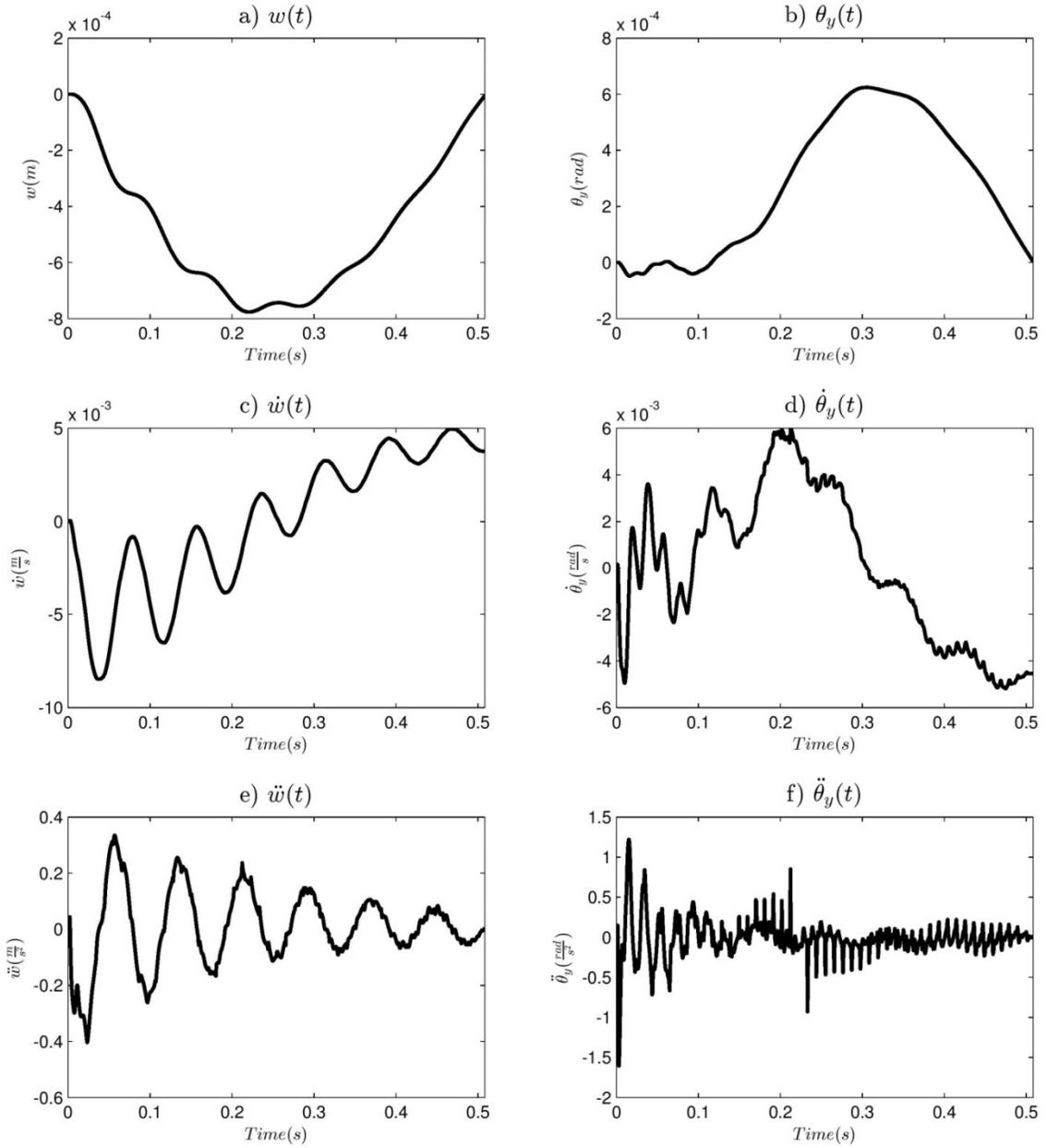


Figure 4: Time histories at $x = \frac{7L}{16}$ and $y = \frac{W}{2}$ calculated using Mindlin-Reissner plate theory and V0(1,1,0) for: a) $w(t)$, b) $\theta_y(t)$, c) $\dot{w}(t)$, d) $\dot{\theta}_y(t)$, e) $\ddot{w}(t)$, and f) $\ddot{\theta}_y(t)$.

3.3 Conclusions

The Euler-Bernoulli beam theory, Timoshenko beam theory, and Mindlin-Reissner plate theory are all suitable for solving problems of flat structural elements undergoing moving loads. The transverse deflection for each agrees with previously published experimental results. For a simply supported beam model, Timoshenko beam theory is the preferred method. Euler-Bernoulli beam theory is only sufficient when the beam is long and slender. Mindlin-Reissner plate theory is preferred over Kirchoff-Love plate theory because of the incorporation of shear effects. There is no severe restriction on the thickness of the beam. However, Mindlin-Reissner plate formulation is more complicated because the stiffness and mass matrices must include the extra dimension. For a beam with a moving load, Timoshenko beam theory is the most suitable choice. It combines the physics of the Mindlin-Reissner plate theory and the ease of implementation of the Euler-Bernoulli beam theory.

Chapter 4: Error Analysis of Numerically Non-Dissipative $V_0(1,1,\rho_{3\infty})$ Algorithms for a Beam Subject to a Moving Load

The previously published analyses of beams with moving loads have used the Wilson θ method [15], the Newmark method [19], among other algorithms. There does not appear to be any rational for why these algorithms were chosen for solving this subset of engineering problems; most probably, it may be due to its historical significance. Within the framework of the GSSSS algorithms exists all second order accurate LMS algorithms including the Wilson θ and the Newmark method. Of particular interest is if one of these algorithms performs better than others.

All algorithms within the GSSSS framework are second-order time accurate, but the numerical dissipation is not equal. For a linear, dynamic system, there is no numerical dissipation for any scheme with $U_0(1,1,\rho_{3\infty})$ and $V_0(1,1,\rho_{3\infty})$ [17]. Focus will be paid to the V_0 family of algorithms.

The performance of an algorithm can be quantified by looking at the truncation error. The truncation error is the cumulative error that is caused by the sum of the error for each time step. The truncation error for displacement at a given time, t , is given as the percent error between the numerical and exact solution as

$$\Pi_w = \frac{|w_{num}(t) - w_{exact}(t)|}{w_{exact}(t)} \quad (4.1)$$

This holds true for all degrees of freedom, $w, \dot{w}, \ddot{w}, \theta, \dot{\theta},$ and $\ddot{\theta}$. For a beam with a moving mass, there exists no exact solution to use in the calculation of the truncation error. Instead, the exact solution is taken as the FE solution with a time step that is many magnitudes smaller than the time step being studied. The errors for the conditions $\rho_{1\infty} = \rho_{2\infty} = 1$ and $\rho_{3\infty} \in [0,1]$ in the V0 families are calculated and subsequently compared. The preferred algorithm exists for a chosen $\rho_{3\infty}$ so that the error is minimized.

The accelerations, \ddot{w} and $\ddot{\theta}$, are not calculated at the same time level as the displacements and velocities. This is important for correctly calculating the truncation error at any particular time. Displacements and velocities are calculated at time level t_{n+1} after iteration n . The accelerations are calculated at time level $t_{n-\phi+1}$ after iteration n [10]. Within the framework of GSSSS, the acceleration at the last time step must be shifted by

$$a_{n-\phi+1} = -\phi a_n + (1 + \phi)a_{n+1} \quad (4.2)$$

$$\phi = W_1(\Lambda_6 - 1) \quad (4.3)$$

4.1 Numerical Examples

The effect of $\rho_{3\infty}$ on the on the error of the dynamic response of a simply supported beam with a moving mass is analyzed. The properties of the 6061 aluminum beam with a rectangular cross section are $L = 1071.5mm, w = 105.25mm, h = 6.35, E = 72.4 GPa, \rho = 2763.6 \frac{kg}{m^3}$ [3]. For

the TB formulation, $\kappa = \frac{5}{6}$. The moving load has a mass of 0.505 kg and a constant speed of $2.108 \frac{\text{m}}{\text{s}}$. The proportional damping ratios are $\xi_1 = \xi_2 = 0.05$. In order to calculate the proportional damping coefficients, α and β , the first two natural frequencies are $\omega_1 = 81 \frac{\text{rad}}{\text{s}}$ and $\omega_2 = 322 \frac{\text{rad}}{\text{s}}$. Figure 5 shows the comparison of the BE, TB, and experimental results for vertical displacement at $x = \frac{7L}{16}$ for V0(1,1,0).

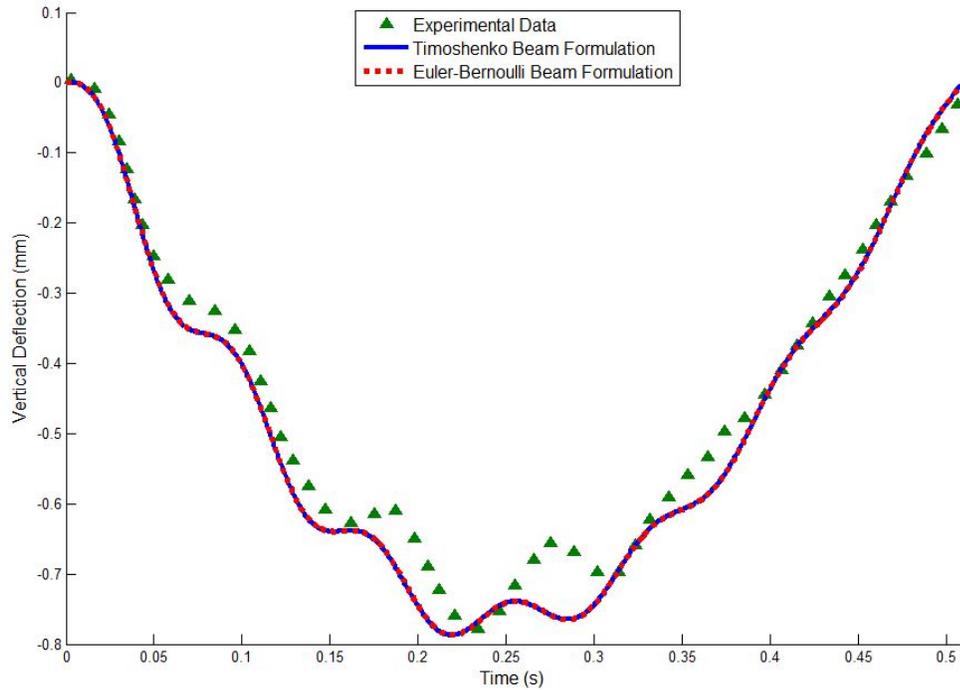


Figure 5: Comparison of the transverse deflection for Euler-Bernoulli beam theory, Timoshenko beam theory, and experimental measurements at $x = \frac{7L}{16}$.

Figure 6 and Figure 7 show the convergence rates of the truncation errors as the time step is decreased at the midpoint of the beam. As expected, the error converges at a second-order rate for each algorithm for each degree of freedom. It is also clear that the magnitude of error changes based on the algorithm chosen. If $\rho_{3\infty}$ is now changed so that it falls between 0 and 1, then the truncation error is calculated and plotted. For this particular problem, $w, \dot{w}, \theta,$ and $\dot{\theta}$ are not

affected by the value of $\rho_{3\infty}$. The error in accelerations, \ddot{w} and $\ddot{\theta}$, can vary greatly based on the value of $\rho_{3\infty}$. Figure 8 shows that the optimal algorithm is V0(1,1,1) for the BE and the TB solutions for the chosen problem and the chosen time at which the error was calculated because this is the location of the least error.

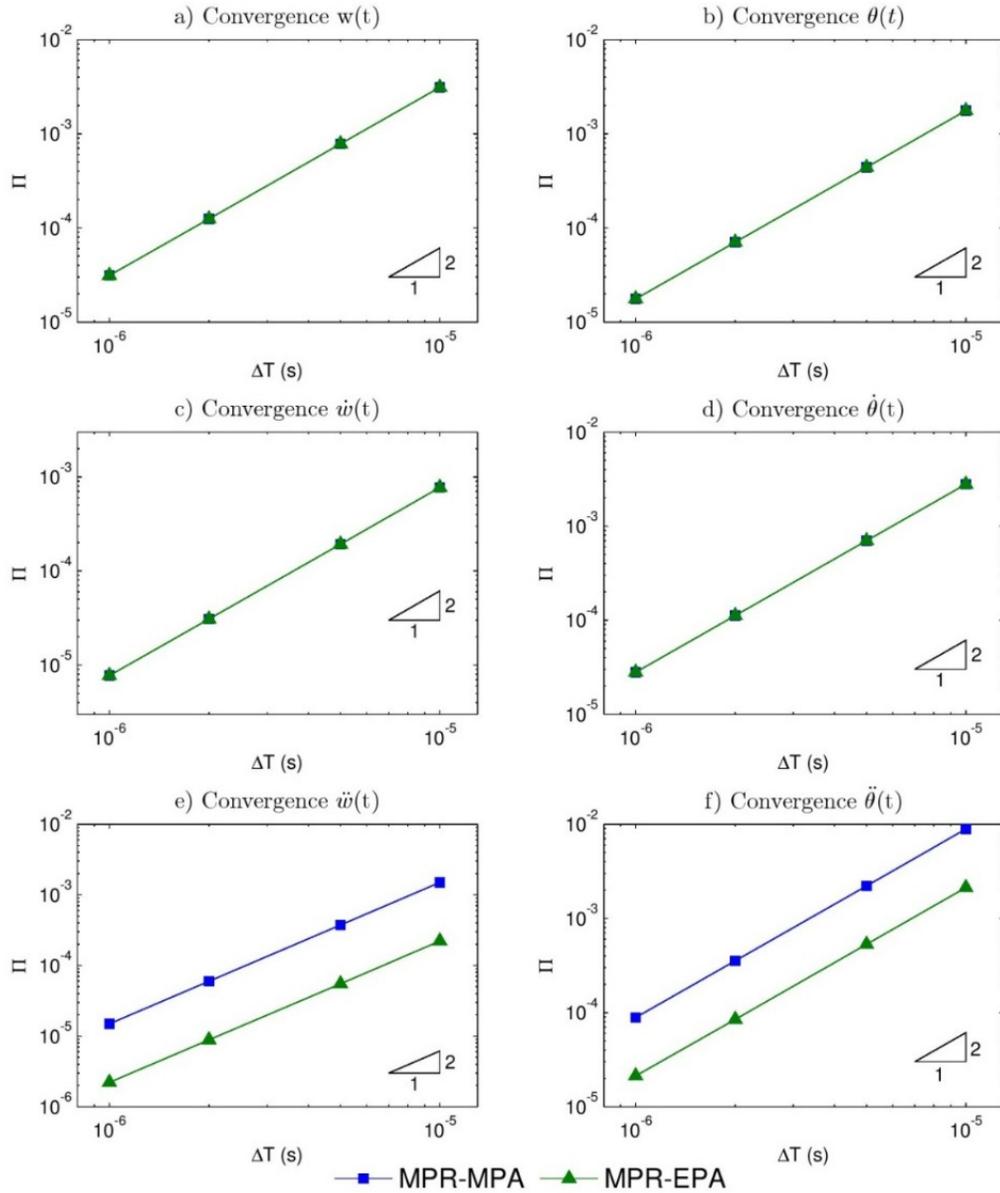


Figure 6: The Euler-Bernoulli truncation error, Π , at the midpoint using MPR-MPA and MPR-EPA as a function of the chosen time step for the following degrees of freedom: a) $w(t)$, b) $\theta_y(t)$, c) $\dot{w}(t)$, d) $\dot{\theta}_y(t)$, e) $\ddot{w}(t)$, and f) $\ddot{\theta}_y(t)$.

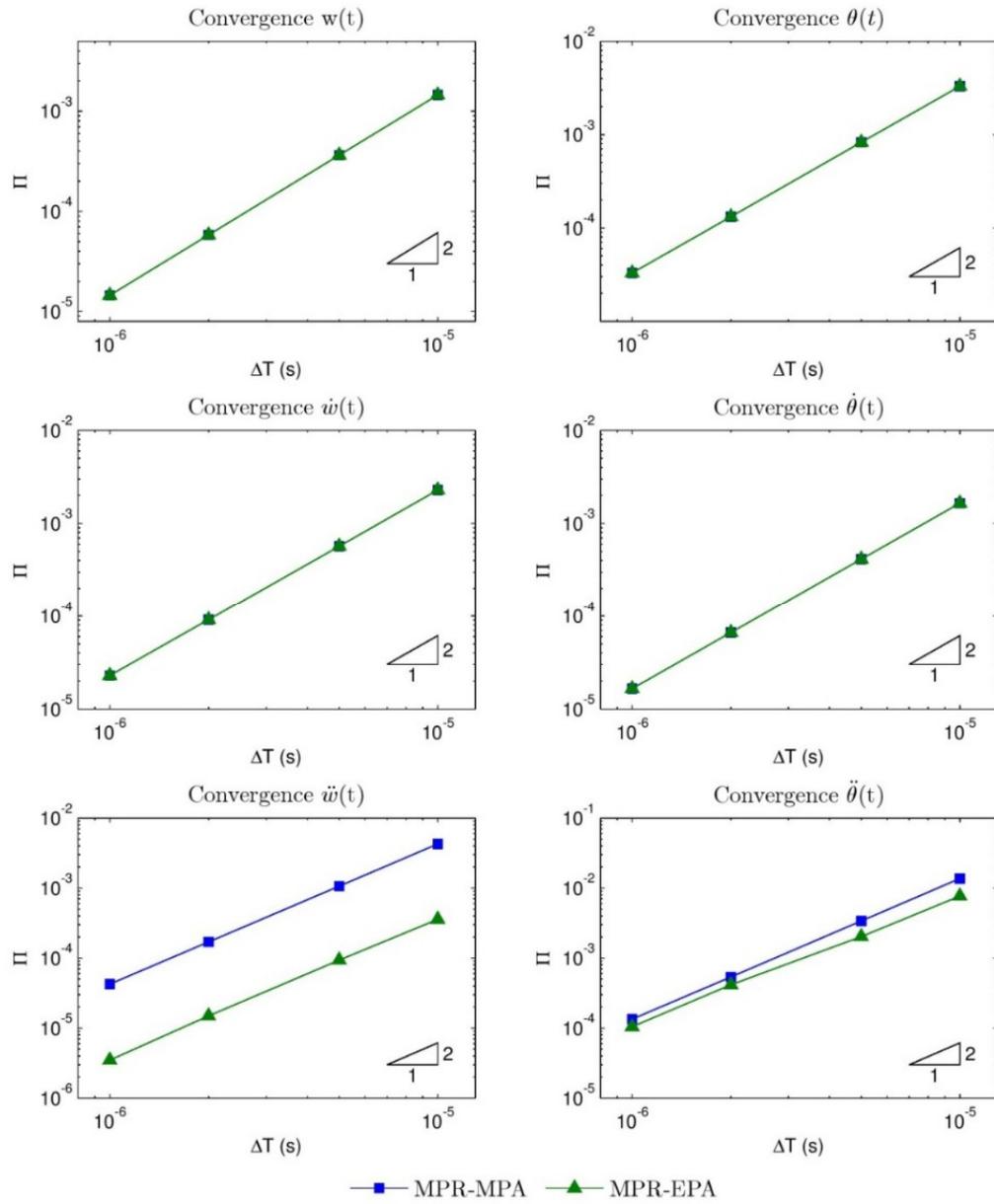


Figure 7: The Timoshenko truncation error, Π , at the midpoint using MPR-MPA and MPR-EPA as a function of the chosen time step for the following degrees of freedom: a) $w(t)$, b) $\theta_y(t)$, c) $\dot{w}(t)$, d) $\dot{\theta}_y(t)$, e) $\ddot{w}(t)$, and f) $\ddot{\theta}_y(t)$.

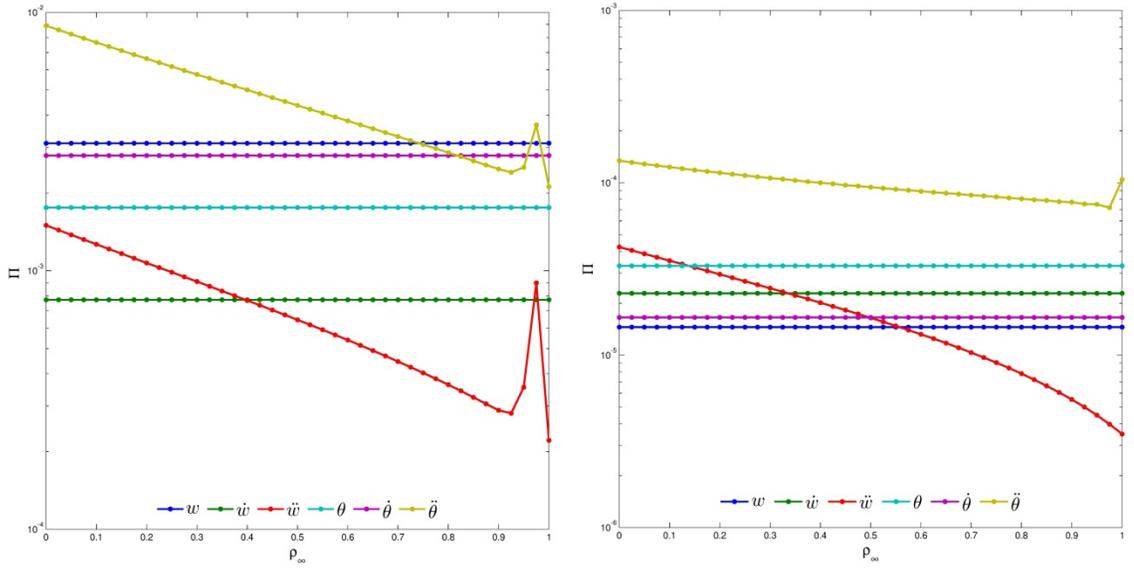


Figure 8: The truncation error, Π , as a function of the selected $\rho_{3\infty}$ for the (Left) Euler-Bernoulli beam (Right) Timoshenko beam.

4.2 Conclusions

All of the algorithms in the V0 family within GSSSS framework are second order time accurate and have zero overshoot in velocity. A small subset, $(1,1,\rho_{3\infty})$ with $\rho_{3\infty} \in [0,1]$, of the V0 algorithms are numerically non-dissipative. The ideal, non-dissipative V0 algorithm is the algorithm with the minimal truncation error, Π is calculated as a function of $\rho_{3\infty}$. This optimal algorithm was tested on a beam with a moving load. For a specified time and position along the beam, the optimal algorithm was found to be V0(1,1,1) for both the Euler-Bernoulli and Timoshenko beam formulations. This is merely an observation for this given problem and the physics of this problem. Further research is needed to understand exactly why V0(1,1,1) is mathematically the most ideal algorithm for this problem.

Chapter 5: Dynamic Response of Curved Beams Subject to Moving Loads Using Finite Elements

As previously mentioned, the study of the dynamic responses of straight beams with a moving load is not new. It is well known that the maximum displacement of the beam is a function of the speed of the moving mass. The previous research has focused on initially straight beams. Engineers and designers will often use slightly curved structural members to increase the overall stiffness. This curvature will have an impact on the maximum deflection of the structure subject to moving loads.

Plane frame elements must be used as opposed to those previously derived in Chapter 2. Plane frame elements have an extra degree of freedom to allow for compression and extension along the length of the element. These elements are assumed to be TB elements. TB beam elements are preferred over BE elements because they are not limited by the ratio of length to height and include shear effects. Plane frame elements have three dependent variables: displacement in the x-direction, $u(t)$, displacement in the y-direction, $w(t)$, and the rotations about the z-axis, $\theta(t)$. The

governing equations for any arbitrarily oriented element as defined by the local element coordinates, (x', y') , now become

$$\rho A \ddot{u}' - \frac{\partial}{\partial x'} \left(EA \left(\frac{\partial u'}{\partial x'} \right) \right) - \bar{b}'(x) = 0 \quad (5.1)$$

$$\rho A \ddot{w}' - \frac{\partial}{\partial x'} \left(\kappa GA \left(\frac{\partial w'}{\partial x'} + \theta \right) \right) - \bar{w}'(x) = 0 \quad (5.2)$$

$$\rho I \ddot{\theta} + \kappa GA \left(\frac{\partial w}{\partial x'} + \theta \right) - \frac{\partial}{\partial x'} \left(EI \frac{\partial \theta}{\partial x'} \right) = 0 \quad (5.3)$$

The variables x' and y' denote the local axis along the element and transverse to the element. As previously done for the beam elements, the Principle of Virtual Work is applied to Equations 5.1-5.3 and subsequently discretized using the variational method. The local trial functions are defined as

$$\tilde{u}'^h(x, y, t) = \sum_{i=1}^n N_i(\xi) u'_i(t) = \mathbf{N}_u(\xi) \mathbf{u}'^e(t) \quad (5.4)$$

$$\tilde{w}'^h(x, y, t) = \sum_{i=1}^n N_i(\xi) w'_i(t) = \mathbf{N}_w(\xi) \mathbf{w}'^e(t) \quad (5.5)$$

$$\tilde{\theta}_y^h(x, y, t) = \sum_{i=1}^n N_i(\xi) \theta_i(t) = \mathbf{N}_{\theta_y}(\xi) \boldsymbol{\theta}^e(t) \quad (5.6)$$

The shape functions are the same as those in Equation 2.25 such that, $\mathbf{N}_u(\xi) = \mathbf{N}_w(\xi) = \mathbf{N}_{\theta}(\xi)$.

The three dependent variables now become

$$\tilde{u}'^h(\xi, t) = \mathbf{H}_u(\xi) \mathbf{u}'^e(t) \quad (5.7)$$

$$\tilde{w}'^h(\xi, t) = \mathbf{H}_w(\xi) \mathbf{u}'^e(t) \quad (5.8)$$

$$\tilde{\theta}^h(\xi, t) = \mathbf{H}_{\theta}(\xi) \mathbf{u}'^e(t) \quad (5.9)$$

The interpolation functions $\mathbf{H}_u(\xi)$, $\mathbf{H}_w(\xi)$ and $\mathbf{H}_{\theta_y}(\xi)$ are defined as

$$\mathbf{H}_u(\xi) = [N_u \quad 0 \quad 0] \quad (5.10)$$

$$\mathbf{H}_w(\xi) = [0 \quad N_w \quad 0] \quad (5.11)$$

$$\mathbf{H}_\theta(\xi) = [0 \quad 0 \quad N_\theta] \quad (5.12)$$

The nodal variable vector $\mathbf{u}^e(t)$ is defined as

$$\mathbf{u}'_e(t) = [\mathbf{u}'^e(t) \quad \mathbf{w}'^e(t) \quad \boldsymbol{\theta}^e(t)]^T \quad (5.13)$$

The trial functions can now be rewritten in their final, concise form as

$$\begin{Bmatrix} \tilde{u}^h(\xi, t) \\ \tilde{w}^h(\xi, t) \\ \tilde{\theta}^h(\xi, t) \end{Bmatrix} = \begin{bmatrix} N_u & 0 & 0 \\ 0 & N_w & 0 \\ 0 & 0 & N_{\theta_y} \end{bmatrix} \begin{Bmatrix} u'^e(t) \\ w'^e(t) \\ \theta^e(t) \end{Bmatrix} = \mathbf{H}_e(\xi) \mathbf{u}'_e(t) \quad (5.14)$$

where

$$\mathbf{H}_e(\xi) = \begin{bmatrix} \mathbf{H}_u(\xi) \\ \mathbf{H}_w(\xi) \\ \mathbf{H}_\theta(\xi) \end{bmatrix} \quad (5.15)$$

The element bending and shear stiffness matrices are the same as those in Equations 2.40 and 2.42.

The local axial strain gradient matrix is defined as

$$\mathbf{B}_a(\xi) = \left[\frac{\partial \mathbf{H}_e^u(\xi)}{\partial \xi} \right] \quad (5.16)$$

The local axial stiffness matrix is defined as

$$\mathbf{K}_a = \int_{\Omega} \mathbf{B}_a^T(\xi) \mathbf{D}_a \mathbf{B}_a(\xi) J(\xi) d\xi \quad (5.17)$$

where $\mathbf{D}_b = EA$ and $J(\xi) = |[J]|$. $[J]$ is the Jacobian matrix for the element.

The local element stiffness matrix is defined as the sum of the axial, bending, and shear stiffness matrices (Equations 2.40 and 2.42),

$$\mathbf{K}'^e = \mathbf{K}_a + \mathbf{K}_b + \mathbf{K}_s \quad (5.18)$$

The local element mass matrix is defined as

$$\mathbf{M}'^e = \int_{\Omega} \mathbf{H}_e^T(\xi) \rho \mathbf{H}_e J(\xi) d\xi \quad (5.19)$$

where ρ is defined as

$$\rho = \begin{bmatrix} \rho A & 0 & 0 \\ 0 & \rho A & 0 \\ 0 & 0 & \rho I \end{bmatrix} \quad (5.20)$$

These are the local element stiffness and mass matrices. For an element arbitrarily oriented in space at an angle of ϕ relative to the x-axis, the local coordinates can be transformed to the global coordinate system by

$$\begin{Bmatrix} u^e(t) \\ w^e(t) \\ \theta_y^e(t) \end{Bmatrix} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u'^e(t) \\ w'^e(t) \\ \theta_y'^e(t) \end{Bmatrix} \quad (5.21)$$

For a planar frame element, the transformation matrix is defined as

$$\mathbf{T} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sin(\phi) & \cos(\phi) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(\phi) & \sin(\phi) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\sin(\phi) & \cos(\phi) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cos(\phi) & \sin(\phi) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.22)$$

The local element matrices are transformed to the global coordinate system by

$$\mathbf{K}^e = \mathbf{T}^T \mathbf{K}'^e \mathbf{T} \quad \text{and} \quad \mathbf{M}^e = \mathbf{T}^T \mathbf{M}'^e \mathbf{T} \quad (5.23)$$

The moving load is assumed to be a concentrated force, $W = -mg$, always in the y-direction. The centrifugal force because of the beam curvature is ignored. For the load on element e at time t , the discretized force vector is defined as

$$\widetilde{\mathbf{f}}_m(t) = W \mathbf{T}^T \begin{Bmatrix} N_u(\eta(t)) \sin(\phi_e) \\ N_w(\eta(t)) \cos(\phi_e) \\ 0 \end{Bmatrix} \quad (5.24)$$

$\eta(t)$ is the location of the load relative to element e as defined as $\eta(t) = \frac{l_m}{L_e}$.

5.1 Numerical Examples

5.1.1 Static Curved Beam Deflection

To validate the stiffness matrix for a curved beam, various static loading cases are analyzed. The rectangular cross-sectioned beam has a span of, $L = 6.928m$, a radius, $R=4m$, a width, $w = 0.4m$, and a height, of $h = 0.6m$. The overall arc length of the beam is $l = 8.378m$. The beam has a Young's modulus, $E = 30GPa$ and Poisson's ratio, $\nu = 0.17$. The boundary conditions at the ends and the loads are varied as shown in Table 5. The results shown in Table 5 show the normalized displacements at the center point and are defined as $\bar{u}_c = \frac{u(x=\frac{L}{2})}{l}$, $\bar{w}_c = \frac{w(x=\frac{L}{2})}{l}$, and $\bar{\theta}_c = \frac{\theta(x=\frac{L}{2})}{l}$.

The results calculated using 100 elements agree closely with the results from [13].

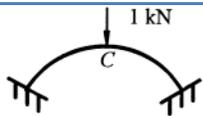
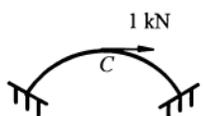
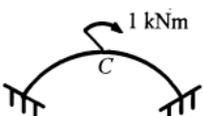
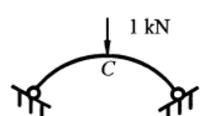
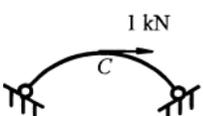
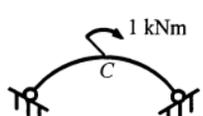
Arch Type [13]	Dimensionless Displacements	Present FE Study (100 Elements)	Litewka & Rakowski [13]	[% Error]
	$\bar{u}_c * 10^6$	0.0000	0.0000	0.00
	$\bar{w}_c * 10^6$	-0.2488	-0.2488	0.00
	$\bar{\theta}_c * 10^6$	0.0000	0.0000	0.00
	$\bar{u}_c * 10^6$	0.1252	0.1252	0.00
	$\bar{w}_c * 10^6$	0.0000	0.0000	0.00
	$\bar{\theta}_c * 10^6$	0.3795	0.3796	0.03
	$\bar{u}_c * 10^6$	-0.0949	-0.0949	0.00
	$\bar{w}_c * 10^6$	0.0000	0.0000	0.00
	$\bar{\theta}_c * 10^6$	1.0822	1.0824	0.02
	$\bar{u}_c * 10^6$	0.0000	0.0000	0.00
	$\bar{w}_c * 10^6$	-0.2809	-0.3047	7.81
	$\bar{\theta}_c * 10^6$	0.0000	0.0000	0.00
	$\bar{u}_c * 10^6$	0.2882	0.2884	0.07
	$\bar{w}_c * 10^6$	0.0000	0.0000	0.00
	$\bar{\theta}_c * 10^6$	0.8061	0.8064	0.04
	$\bar{u}_c * 10^6$	0.2015	0.2016	0.05
	$\bar{w}_c * 10^6$	0.0000	0.0000	0.00
	$\bar{\theta}_c * 10^6$	1.3613	1.3613	0.00

Table 5: Non-dimensional deflections at the center of the beam for various loading cases and boundary conditions.

5.1.2 Free Vibration of Curved Beams

To validate the derivation of the mass matrix, the free vibration of various curved beams is analyzed. The hinged-hinged beam has a radius of curvature, $R = 12in$, cross-section width and height $w = h = 0.25in$, and a central angle, ω . The beam has a Young's modulus, $E = 3.04 \times 10^7 psi$, a Poisson's ratio, $\nu = 0.3$, and a density, $\rho = 0.02736 \frac{slug-ft}{in^4}$. As shown in Table 6, the results for 100 elements and various central angles, ω , agree closely with those by [12].

Fundamental Frequency (rad/s)			
ω (°)	100 FE Elements	Krishnan, et al. [12]	% Error
10	5841.43	5874.30	0.33
20	2827.41	2823.10	0.27
30	2339.37	2345.20	0.03
60	561.99	561.20	0.14
90	230.96	230.40	0.24
120	114.73	116.30	1.35
150	64.35	64.93	0.89
180	37.84	38.24	1.05

Table 6: Fundamental frequency for curved beams of various radii.

5.1.3 Curved Beams Subject to a Moving Load

Previous research has attempted to tackle the problem of moving loads on curved beams [20]. This research has two fundamental flaws. The first is shown in Figure 9. This figure shows that the maximum absolute vertical deflection approaches zero as the speed of the moving load increases. This goes against all previous research for forced vibration of beams undergoing moving loads. The second inconsistency is in the time histories. The time histories of two points on the curved beam show spikes and dissipation after the load has passed across the beam. These appear to be possible numerical errors. There is no mention of damping in the derivation of the FE matrices. This means that the damping comes from numerical dissipation. The time history of the center of the beam, δ_{59} from Figure 10, should be roughly symmetric over the period of 0 to ≈ 12 s. The examples from this research will show valid applications of FE and the MPR-MPA algorithm for

solving the dynamic response of curved beams due to moving loads.

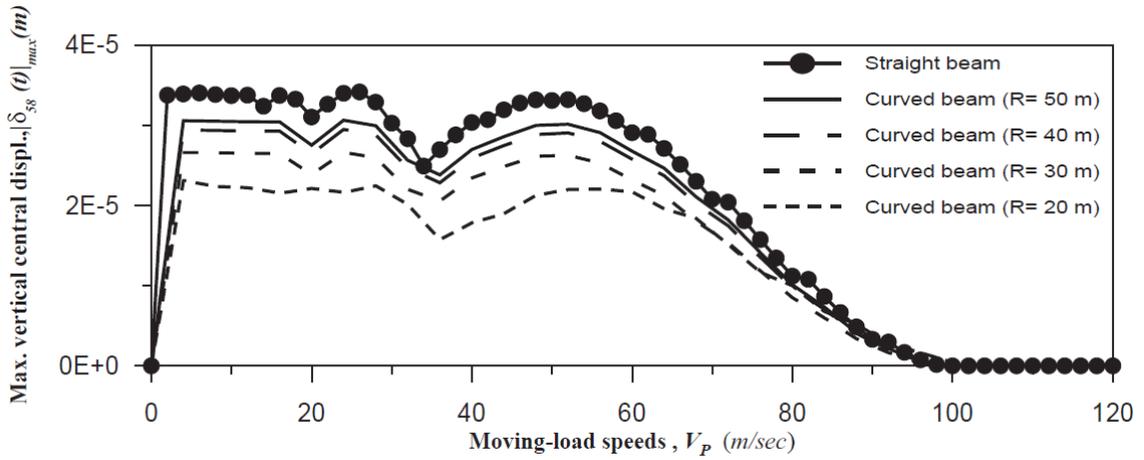


Figure 9: Impact of the beam radius and the speed of the moving load on the maximum absolute vertical displacements at the midpoint of the beam according to [20].

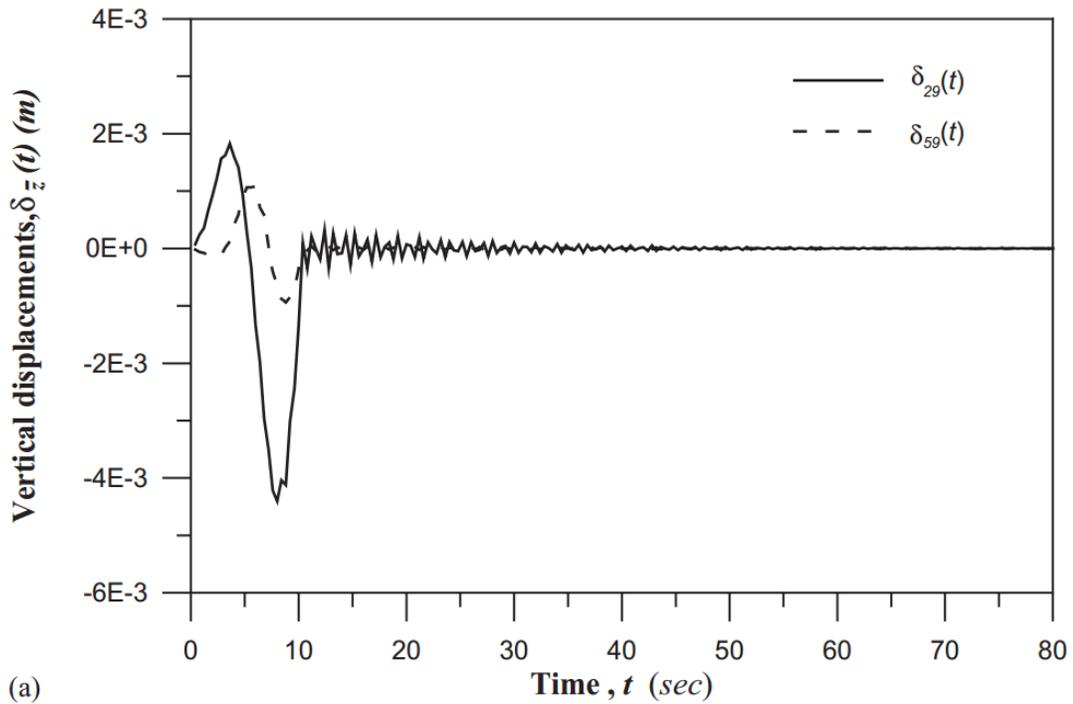


Figure 10: Time history of the vertical displacements of the curved beam at $\frac{L}{4}$ and $\frac{L}{2}$ with the speed of the load $v = 4 \frac{m}{s}$ according to [20].

Various hinged-hinged rectangular curved beams undergo excitation by a concentrated force moving along the beam's length at various constant tangential speeds. These are the same curved

beams as studied by [20]. The beams have a span, $L = 10m$, cross-section width of $w = 1.5m$, and a height $h = 0.8m$. They have a Young's modulus, $E = 120 \text{ GPa}$, Poisson's ration, $\nu = 0.3$, and a density, $\rho = 7200 \frac{\text{kg}}{\text{m}^3}$. The moving force has a magnitude, $f_w = 50kN$. The damping ratios are assumed to be $\xi_1 = \xi_2 = 0.05$. The velocity of the force is varied from $v = 10 \frac{\text{m}}{\text{s}}$ to $v = 200 \frac{\text{m}}{\text{s}}$. Five different radii are analyzed: $R_1 = 20m$, $R_2 = 30m$, $R_3 = 40m$, $R_4 = 50m$, and $R_5 = 9 \times 10^{99}m$. The beam with a radius $R = 9 \times 10^{99}m$ equates to a straight beam with no curvature. Figure 11 and Figure 12 show the time histories of various points along the beam with a radius $R = 20m$ for the cases when the velocity is $v = 10 \frac{\text{m}}{\text{s}}$ and $v = 100 \frac{\text{m}}{\text{s}}$ using the MPR-MPA algorithm.

The curved beam is simply supported, therefore, the essential boundary conditions are

$$u(t)|_{x=0} = 0, \quad u(t)|_{x=L} = 0, \quad w(t)|_{x=0} = 0, \quad \& \quad w(t)|_{x=L} = 0 \quad (5.25)$$

and the natural boundary conditions are

$$M(t)|_{x=0} = 0 \quad \text{and} \quad M(t)|_{x=L} = 0 \quad (5.26)$$

The beam is initially at rest, therefore, the initial conditions are

$$\mathbf{u}(0) = 0 \quad \text{and} \quad \dot{\mathbf{u}}(0) = 0 \quad (5.27)$$

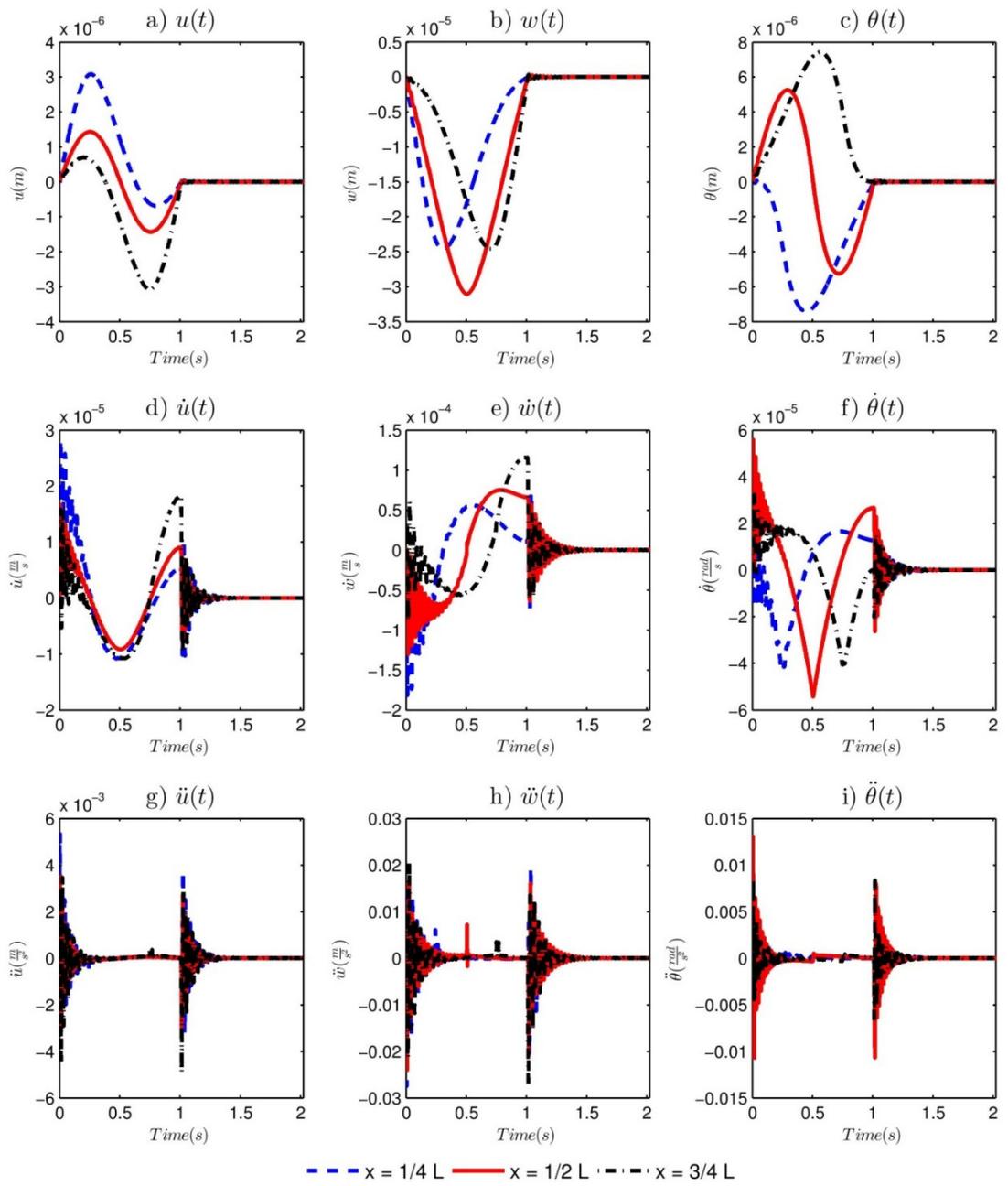


Figure 11: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, & $x = \frac{3L}{4}$ for the moving mass moving at $v = 10 \frac{m}{s}$ for each of the following: a) $u(t)$, b) $w(t)$, c) $\theta(t)$, d) $\dot{u}(t)$ e) $\dot{w}(t)$, f) $\dot{\theta}(t)$, g) $\ddot{u}(t)$, h) $\ddot{w}(t)$, and i) $\ddot{\theta}(t)$.

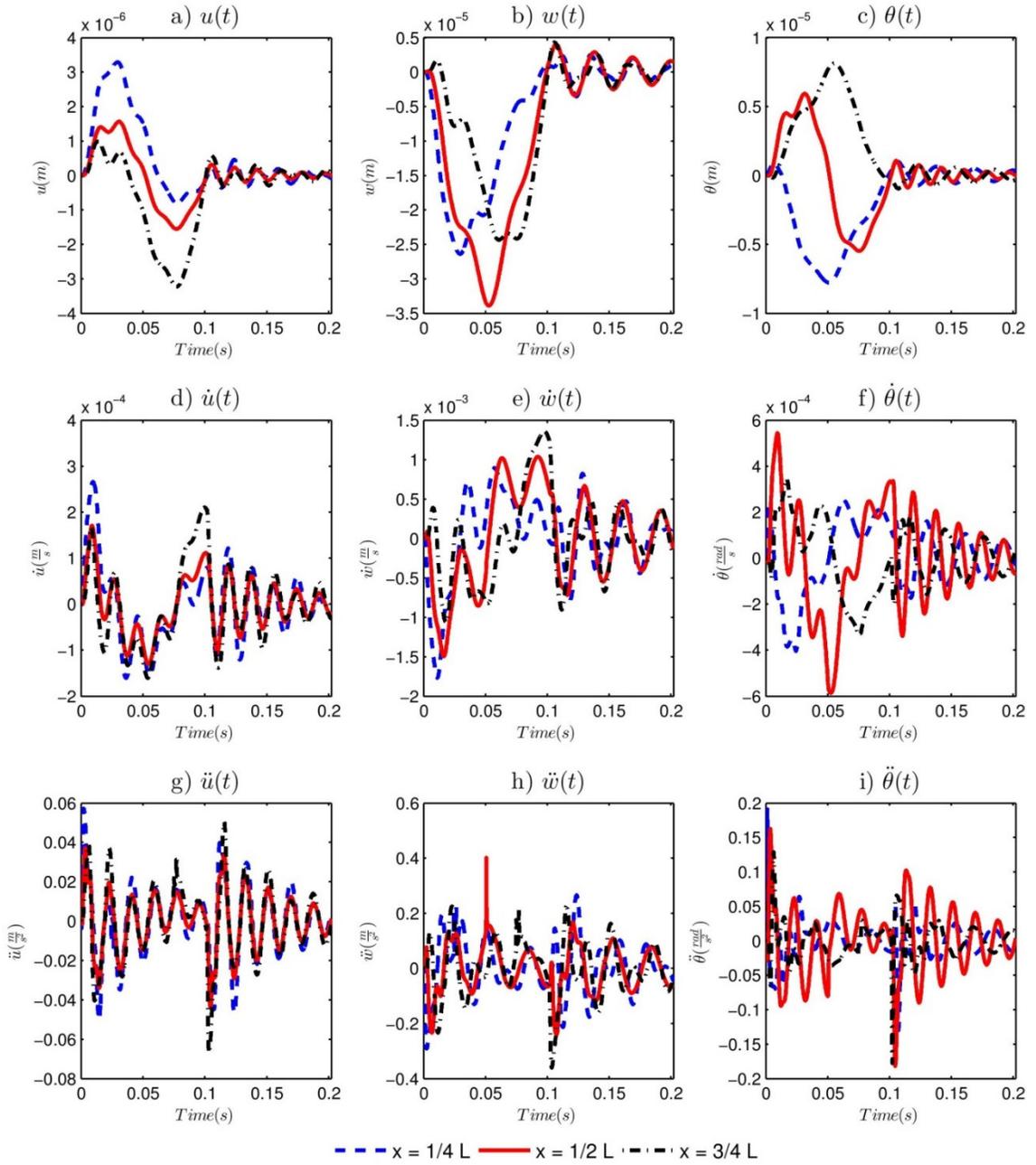


Figure 12: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, & $x = \frac{3L}{4}$ for the moving mass moving at $v = 100 \frac{m}{s}$ for each of the following: a), $u(t)$, b) $w(t)$, c) $\theta(t)$, d) $\dot{u}(t)$ e) $\dot{w}(t)$, f) $\dot{\theta}(t)$, g) $\ddot{u}(t)$, h) $\ddot{w}(t)$, and i) $\ddot{\theta}(t)$.

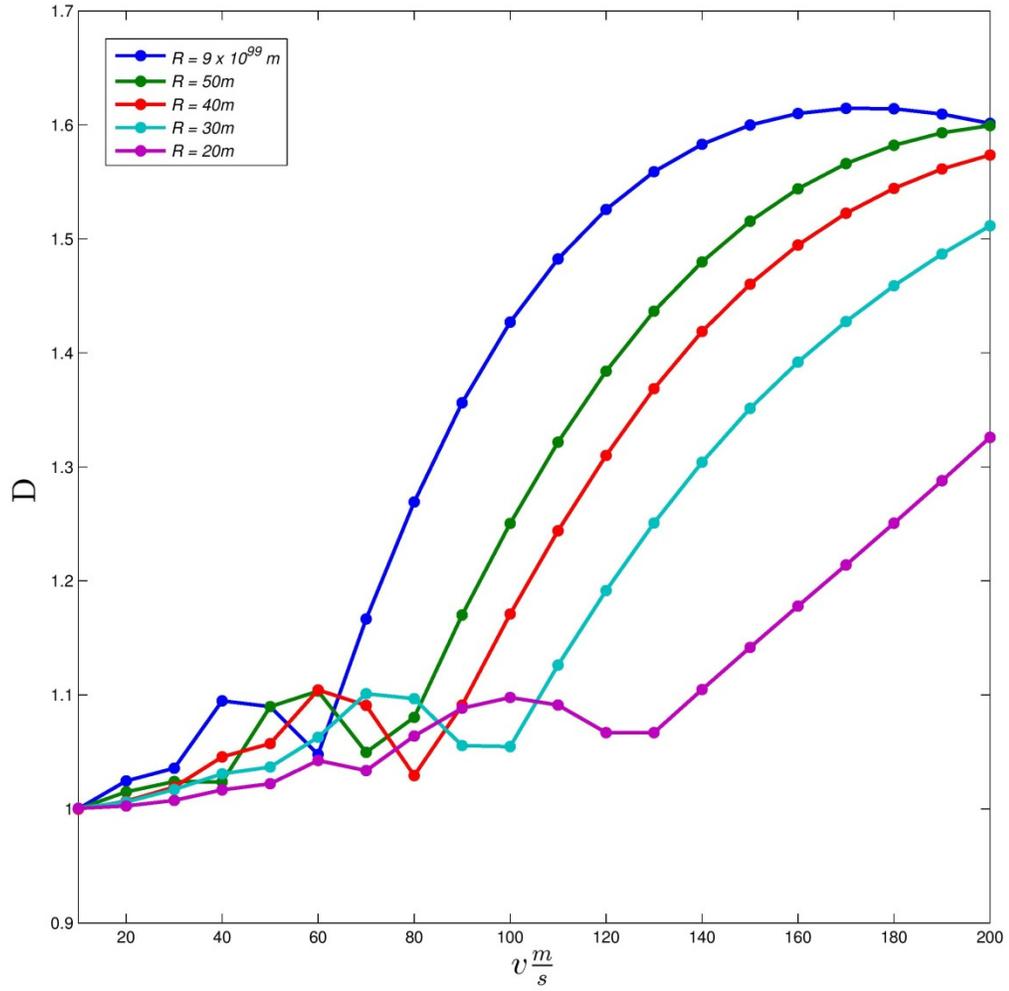


Figure 13: Effect of the beam radius of curvature and the speed of the moving load, v , on the maximum absolute non-dimensional vertical displacement, D .

As the speed of the force is increased, the maximum vertical deflection of the beam increases. Figure 13 shows the maximum non-dimensional vertical deflection, D , for each of the five variations of beams as a function of the speed of the moving force. The non-dimensional vertical deflection is defined as

$$D = \left| \frac{W_{\max \text{ Dynamic}}}{W_{\max \text{ Static}}} \right| \quad (5.28)$$

The maximum vertical static deflection is calculated using FE, and these values are listed in Table 7.

Radius (m)	$w_{max\ static} (m)$
9×10^{99}	1.383×10^{-4}
50	8.641×10^{-5}
40	7.167×10^{-5}
30	5.277×10^{-5}
20	3.109×10^{-5}

Table 7: Maximum static deflection of the clamped-clamped beam for various radii of curvature.

5.2 Conclusions

The dynamic response of structures with moving loads is magnified compared to the static loads. For a relatively slow moving force, this max deflection is nearly equivalent to the maximum static loading case. At much higher speeds, the maximum deflection can be greater than 160% of the maximum static loading case. If the same structures are slightly curved to increase the strength, the effects of the moving load on the vertical deflection can be greatly reduced. By increasing the curvature, a structure can be designed to carry much faster moving loads. This is very important for engineers and designers as moving vehicles, trains, and aircraft are increasing their max speeds. Structures must be designed to handle such loads. FE and MPR-MPA provide useful tools for the analysis of these structures.

Chapter 6: Dynamic Response of Curved Beams Subject to Moving Loads and Thermal Strains Using Finite Elements

As shown in Chapter 5, the radius of curvature and speed of a moving load can greatly affect the dynamic responses of beams. Increasing the radius of curvature of the beam can reduce these effects while increasing the speed of the load magnifies these effects. Such structures can also undergo thermal strains due to temperature variations. To date, no work is available that combines moving loads with thermal strains. These temperature differences can create large strains that will have a large effect on the dynamic response of the beam. Curved beams undergo thermal expansion when the mean temperature of the beam varies from the temperature of its environment. They undergo thermal bending when the temperature varies through the height of the beam.

For the FE discretization the force vector must now include the effects of these thermal stresses.

The force vector is now defined as

$$\tilde{f}(t) = \tilde{f}_w(t) + \tilde{f}_{thermal} = \tilde{f}_w(t) + \tilde{f}_{TE} + \tilde{f}_{TB} \quad (6.1)$$

The external force vector due to the moving external load, $\tilde{f}_w(t)$, remains the same as in Equation 3.5. The thermal force vector is a combination of the thermal expansion, \tilde{f}_{TE} , and the thermal bending moment, \tilde{f}_{TB} . Thermal expansion is governed by the strain equation

$$\delta_T = \alpha L \Delta T \quad (6.2)$$

where α is the thermal expansion coefficient for a specific material, and ΔT is the difference in the mean temperature of the beam and the environment. The discretized local element thermal expansion force is defined as

$$\tilde{f}_{TE}' = \alpha (T_{avg} - T_\infty) \int_{\Omega} \mathbf{B}_a^T(\xi) \mathbf{D}_a J(\xi) d\xi \quad (6.3)$$

T_{avg} is the average temperature through the height of the beam and is considered constant throughout the width and length of the beam. \mathbf{B}_a is defined in Equation 5.16.

Thermal bending in a beam is governed by the equation

$$M_T = E\alpha \int_{-\frac{h}{2}}^{\frac{h}{2}} (T(h) - T_\infty) h dh \quad (6.4)$$

The temperature distribution in the beam is assumed to be linear through the cross-section height of the beam. The discretized local element thermal bending vector now becomes

$$\tilde{f}_{TB}' = \frac{\alpha (T_{BOT} - T_{TOP})}{h} \int_{\Omega} \mathbf{B}_B^T(\xi) \mathbf{D}_B J(\xi) d\xi \quad (6.5)$$

\mathbf{B}_B is defined in Equation 2.39. These are the local element thermal force vectors. They need to be transformed to the global coordinate system using the transformation matrix \mathbf{T} from Equation 5.22

$$\tilde{f}_{TE} = \mathbf{T}' \tilde{f}_{TE}' \quad (6.6)$$

6.1 Numerical Examples

6.1.1 Thermal Expansion of a Ring

In order to validate the derivation of the thermal expansion force vector, the expansion of a ring is analyzed. Rings of various radii are subjected to a 40°C and a -40°C change in temperature from their environment. They have a Young's modulus, $E = 205 \text{ GPa}$, Poisson's ratio $\nu = 0.29$, and thermal expansion coefficient $\alpha = 11.9 \times 10^{-6} \text{ K}^{-1}$. Table 8 shows that the FE results for change in the radius of the center of the beam compare to within machine precision to the exact solution

$$dR = \alpha R \Delta T \quad (6.7)$$

R (m)	$T_{\text{avg}} = 40^{\circ}\text{C}$			$T_{\text{avg}} = -40^{\circ}\text{C}$		
	dR(m) for 100 FE Elements	dR(m) Exact Solution	[% Error]	100 FE Elements	Exact Solution	[% Error]
1	4.76E-04	4.76E-04	3.11E-09	-4.76E-04	-4.76E-04	3.09E-09
2	9.52E-04	9.52E-04	2.24E-09	-9.52E-04	-9.52E-04	2.27E-09
3	1.43E-03	1.43E-03	4.32E-08	-0.00143	-1.43E-03	4.32E-08
4	1.90E-03	1.90E-03	6.40E-08	-0.0019	-1.90E-03	6.40E-08
5	2.38E-03	2.38E-03	2.64E-07	-0.00238	-2.38E-03	2.64E-07
6	2.86E-03	2.86E-03	9.50E-08	-0.00286	-2.86E-03	9.50E-08
7	3.33E-03	3.33E-03	7.74E-08	-0.00333	-3.33E-03	7.74E-08
8	3.81E-03	3.81E-03	5.86E-08	-0.00381	-3.81E-03	5.86E-08
9	4.28E-03	4.28E-03	4.07E-07	-0.00428	-4.28E-03	4.07E-07
10	4.76E-03	4.76E-03	9.12E-07	-0.00476	-4.76E-03	9.12E-07

Table 8: Change in radius of various rings subject to a temperature change of (Left) 40°C (Right) -40°C .

6.1.2 Thermal Bending of a Cantilever Beam

In order to validate the derivation of the thermal bending force vector, the bending of a cantilever beam subject to a temperature change through the height of the beam is analyzed. The square cross-sectioned beam has a length, $L = 1\text{m}$, width $w = 0.1\text{m}$, height $h = 0.05\text{m}$. The properties of the beam are Young's modulus, $E = 205 \text{ GPa}$, Poisson's ratio $\nu = 0.29$, and thermal expansion coefficient $\alpha = 11.9 \times 10^{-6} \text{ K}^{-1}$. The temperature of the bottom surface of the beam is constant,

$T_{BOT} = 0^{\circ}C$. The temperature of the environment is $T_{\infty} = \frac{(T_{TOP} - T_{BOT})}{2}$ so that the thermal expansion effects are negated. Table 9 shows that the FE results compare closely to the exact BE solution [1] for various temperature variations.

$$w_{max} = \frac{\alpha L^2 (T_{BOT} - T_{TOP})}{2h} \quad (6.8)$$

$T_{TOP} (^{\circ}C)$	100 FE Elements	Exact BE Solution	[% Error]
-40	4.76E-03	4.76E-03	2.59E-07
-30	3.57E-03	3.57E-03	2.59E-07
-20	2.38E-03	2.38E-03	2.59E-07
-10	1.19E-03	1.19E-03	2.59E-07
0	0.00E+00	0.00E+00	0.00E+00
10	-1.19E-03	-1.19E-03	2.59E-07
20	-2.38E-03	-2.38E-03	2.59E-07
30	-3.57E-03	-3.57E-03	2.59E-07
40	-4.76E-03	-4.76E-03	2.59E-07

Table 9: Bending of the tip of a cantilever beam for various temperature changes through the height of the beam.

6.1.3 Curved Beams Subject to Moving Loads and Thermal Strains

The same beams from Section 5.1.3 are analyzed, but now thermal strains are applied to the beams. The beams have a span, $L = 10m$, cross-section width of $w = 1.5m$, and a height $h = 0.8m$. They have a Young's modulus, $E = 120 GPa$, Poisson's ration $\nu = 0.3$, a density $\rho = 7200 \frac{kg}{m^3}$, thermal expansion coefficient, $\alpha = 11.9 \times 10^{-6} K^{-1}$. The moving force has a magnitude, $f_w = 50kN$. The damping ratios are assumed to be, $\xi_1 = \xi_2 = 0.05$. The velocity of the force is varied from $v = 10 \frac{m}{s}$ to $v = 200 \frac{m}{s}$. Five different radii are analyzed: $R_1 = 20m$, $R_2 = 30m$, $R_3 = 40m$, $R_4 = 50m$, and $R_5 = 9 \times 10^9 m$. The beam with a radius $R = 9 \times 10^9 m$ equates to a straight beam with no curvature.

Four thermal cases are studied. The first is the baseline study with no thermal strains. The second case is thermal expansion only. The temperature of the top and bottom of the beam are $T_{TOP} = T_{BOT} = 25^{\circ}C$, and the temperature of the environment is $T_{\infty} = 30^{\circ}C$. The third case is thermal bending strains only. The temperature of the top of the beam is $T_{TOP} = 20^{\circ}C$. The temperature of the bottom of the beam is $T_{BOT} = 25^{\circ}C$. The temperature of the environment is equal to the average temperature through the height of the beam $T_{\infty} = T_{avg} = 22.5^{\circ}C$. The fourth case is the combined thermal expansion and thermal bending strain. The temperature of the top of the beam is $T_{TOP} = 20^{\circ}C$. The temperature of the bottom of the beam is $T_{BOT} = 25^{\circ}C$. The temperature of the environment is $T_{\infty} = 25^{\circ}C$.

The non-dimensional maximum vertical deflection, D , is again calculated as a function of the speed of the moving load. The maximum static deflection is again the same as in Table 7. Figure 14 shows the effect of the radius of the beam and the thermal loading effects on the maximum vertical deflection. Appendix B shows the time histories of various points along the beam with a radius $R = 20m$ for each loading case for velocity of the moving load $v = 10 \frac{m}{s}$ and $v = 100 \frac{m}{s}$ using the MPR-MPA algorithm.

The curved beam is simply supported, therefore, the essential boundary conditions are

$$u(t)|_{x=0} = 0, \quad u(t)|_{x=L} = 0, \quad w(t)|_{x=0} = 0, \quad \& \quad w(t)|_{x=L} = 0 \quad (6.9)$$

and the natural boundary conditions are

$$M(t)|_{x=0} = 0 \quad \text{and} \quad M(t)|_{x=L} = 0 \quad (6.10)$$

The beam is initially subject to the thermal expansion and bending, therefore, the initial conditions are

$$\mathbf{u}(0) = \mathbf{K}^{-1} \tilde{\mathbf{f}}_{Thermal} \quad \text{and} \quad \dot{\mathbf{u}}(0) = \mathbf{0} \quad (6.11)$$

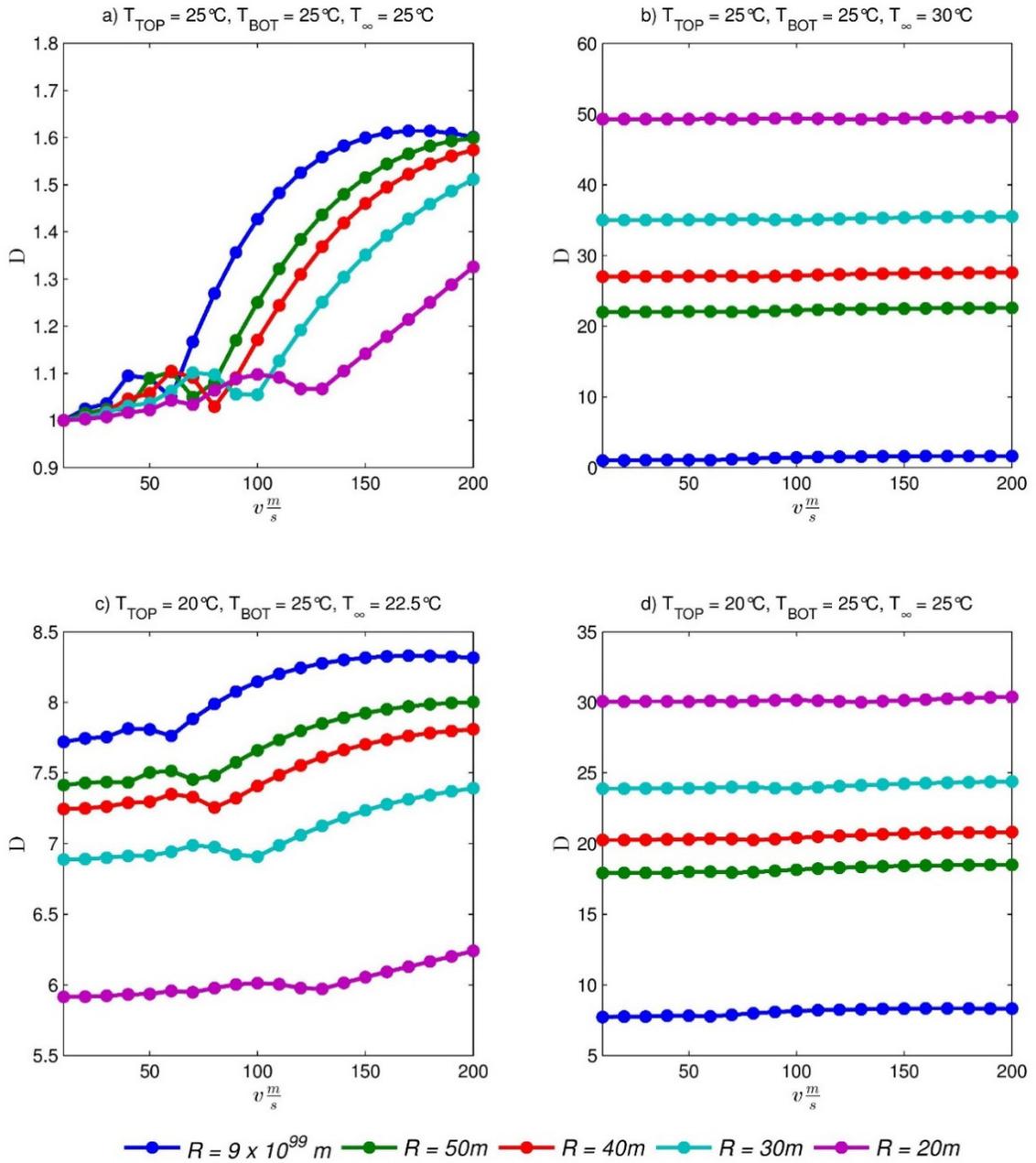


Figure 14: Effect of the beam radius of curvature and the speed of the moving load on the maximum absolute non-dimensional vertical displacement, D , for various thermal strains: a) $T_{TOP} = 25^\circ C, T_{BOT} = 25^\circ C, \& T_\infty = 25^\circ C$, b) $T_{TOP} = 25^\circ C, T_{BOT} = 25^\circ C, \& T_\infty = 30^\circ C$, c) $T_{TOP} = 20^\circ C, T_{BOT} = 25^\circ C, \& T_\infty = 22.5^\circ C$, d) $T_{TOP} = 20^\circ C, T_{BOT} = 25^\circ C, \& T_\infty = 25^\circ C$.

6.2 Conclusions

The dynamic response of structures due to moving loads and thermal strains is magnified compared to the static loads. The effects of thermal expansion and thermal bending can have a much larger effect on the deflection of curved beams. For the given example, the thermal expansion effects are >10x the effects of the moving load. This is especially true when the radius becomes smaller, and the curvature becomes greater. For the case when the radius equaled half the length of the span, a half circle, the thermal expansion deflection is $\approx 40x$ greater than the case with no thermal bending strain. These effects must be well understood by structural engineers and designers in order to build safer and stronger structures. If the perceived temperature changes can vary greatly, the thermal strains will become more dominant than those caused by the moving loads. The optimal approach given by this research is to use the Finite Element method combined with the MPR-MPA algorithm.

Chapter 7: Dynamic Response of Plates Subject to Moving Loads and Thermal Strains Using Finite Elements

As previously shown, the effects of thermal strains can greatly impact the dynamic response of a beam undergoing a moving load. Beams are not the only structural elements that exhibit these effects. Plates are also of particular interest. Examples of such plates include road decks and the decks of aircraft carriers. Similarly to beams, it is known that the magnitude of the response of a plate due to moving loads varies as a function of the speed of the load [21].

Of importance is what effect thermal strains can have on the response of these plates undergoing excitation by moving loads. Temperature variations through the thickness of a plate create thermal bending moments. The thermal bending moment is governed by

$$M_T = E\alpha \int_{-\frac{h}{2}}^{\frac{h}{2}} (T(z) - T_\infty)z dz \quad (7.1)$$

For a linear temperature distribution through the thickness of the plate, the thermal bending moment is

$$M_T = \frac{E\alpha h^2(T_{BOT} - T_{TOP})}{12} \quad (7.2)$$

The discretized force vector for the thermal bending of a plate is defined as

$$\tilde{f}_{TB} = \frac{E\alpha h^2(T_{BOT} - T_{TOP})}{12(1-\nu)} \int_{\Omega} \begin{Bmatrix} 0 \\ \frac{\partial N_{\theta_x}(\xi, \eta)}{\partial \eta} \\ \frac{\partial N_y(\xi, \eta)}{\partial \xi} \end{Bmatrix} J(\xi, \eta) d\xi d\eta \quad (7.3)$$

7.1 Numerical Examples

7.1.1 Thermal Bending of a Plate

In order to validate the derivation of the thermal bending moment vector, the static thermal bending of a plate is analyzed. A hard simply supported plate with a length, $L = 2m$, width, $W = 4m$, and a thickness, $h = 0.1m$ has a temperature, $T_{TOP} = 100^\circ C$, applied to the top and a temperature, $T_{BOT} = 25^\circ C$, applied to the bottom surface. The plate has a Young's modulus, $E = 72 GPa$, Poisson's ratio, $\nu = 0.33$, and a thermal expansion coefficient, $\alpha = 2.3 \times 10^{-7} K^{-1}$. Figure 15 shows the transverse vertical deflection calculated by FE as compared to the exact solution. The exact analytical solution is given by [4] as

$$w(x, y) = \frac{4L^2\alpha\Delta T(1+\nu)}{\pi^3 h} \sum_{m=1,3,5}^{\infty} \frac{1}{m^3} \sin\left(\frac{m\pi x}{L}\right) \left[1 - \frac{\cosh\left(\frac{m\pi y}{L}\right)}{\cosh\left(\frac{m\pi W}{2L}\right)} \right] \quad (7.4)$$

The plate has hard simply supported edges, therefore the essential boundary conditions are

$$w|_{x=0} = w|_{x=L} = w|_{y=0} = w|_{y=W} = 0 \quad (7.5)$$

$$\theta_x|_{x=0} = 0, \theta_x|_{x=L} = 0 \quad (7.6)$$

$$\theta_y|_{y=0} = 0, \theta_y|_{y=W} = 0 \quad (7.7)$$

and the natural boundary conditions are

$$M|_{x=0} = M|_{x=L} = M|_{y=0} = M|_{y=W} = 0 \quad (7.8)$$

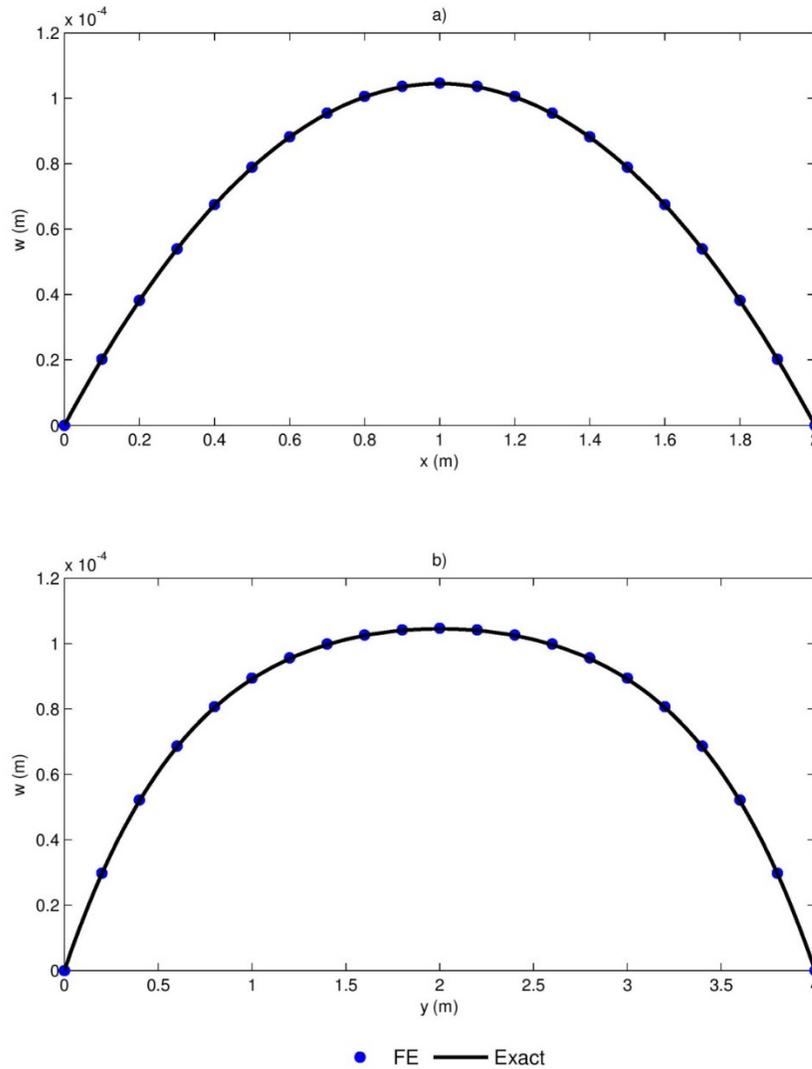


Figure 15: Comparison of the FE and exact solution for the thermal bending of the plate for (a) $w\left(x, y = \frac{W}{2}\right)$ and (b) $w\left(x = \frac{L}{2}, y\right)$.

7.1.2 Plates Subject to Moving Loads and Thermal Strains

The combined effects of a moving load and thermal bending on a hard simply supported plate are analyzed. The square plate has a length and width, $L = W = 2m$, and a thickness, $h = 0.01m$.

The plate has a Young's modulus, $E = 72 \text{ GPa}$, Poisson's ratio, $\nu = 0.33$, density, $\rho = 2700 \frac{\text{kg}}{\text{m}^3}$,

and a thermal expansion coefficient, $\alpha = 2.3 \times 10^{-7} K^{-1}$. The proportional damping ratios are assumed to be $\xi_1 = \xi_2 = 0.05$. Using FE, the first two natural frequencies are found to be $\omega_1 = 78 \frac{rad}{s}$ and $\omega_2 = 197 \frac{rad}{s}$. The moving load is a line pressure across the beam with a magnitude of $p = -500 \frac{N}{m}$ and a length of 0.5m in the y-direction and centered on the plate.

Five thermal cases are studied. The five cases are

Case	T _{TOP} (°C)	T _{BOT} (°C)
1	20	20
2	40	20
3	20	40
4	60	20
5	20	60

Table 10: Five thermal loading cases.

The non-dimensional maximum vertical deflection, D , is again calculated as a function of the speed of the moving load. The maximum static deflection is $|w_{\max static}| = 0.001627m$. Figure 16 shows the effect of the radius of the beam and the thermal loading effects on the maximum vertical deflection. For each loading case the velocity of the moving load is varied between $2.5 \frac{m}{s}$ and $50 \frac{m}{s}$. Each transient solution is calculated using the MPR-MPA algorithm with 1000 time steps.

Therefore, the time step for each initial velocity is $\Delta t = \frac{1}{500v}$.

The plate is hard simply supported on all edges, therefore, the essential boundary conditions are

$$w(t)|_{x=0} = w(t)|_{x=L} = w(t)|_{y=0} = w(t)|_{y=W} = 0 \quad (7.9)$$

$$\theta_x(t)|_{x=0} = \theta_x(t)|_{x=L} = \theta_y(t)|_{y=0} = \theta_y(t)|_{y=W} = 0 \quad (7.10)$$

and the natural boundary conditions are

$$M|_{x=0} = M|_{x=L} = M|_{y=0} = M|_{y=W} = 0 \quad (7.11)$$

The plate is initially subject to the thermal bending, therefore, the initial conditions are

$$\mathbf{u}(0) = \mathbf{K}^{-1} \tilde{f}_{TB} \text{ and } \dot{\mathbf{u}}(0) = 0 \quad (7.12)$$

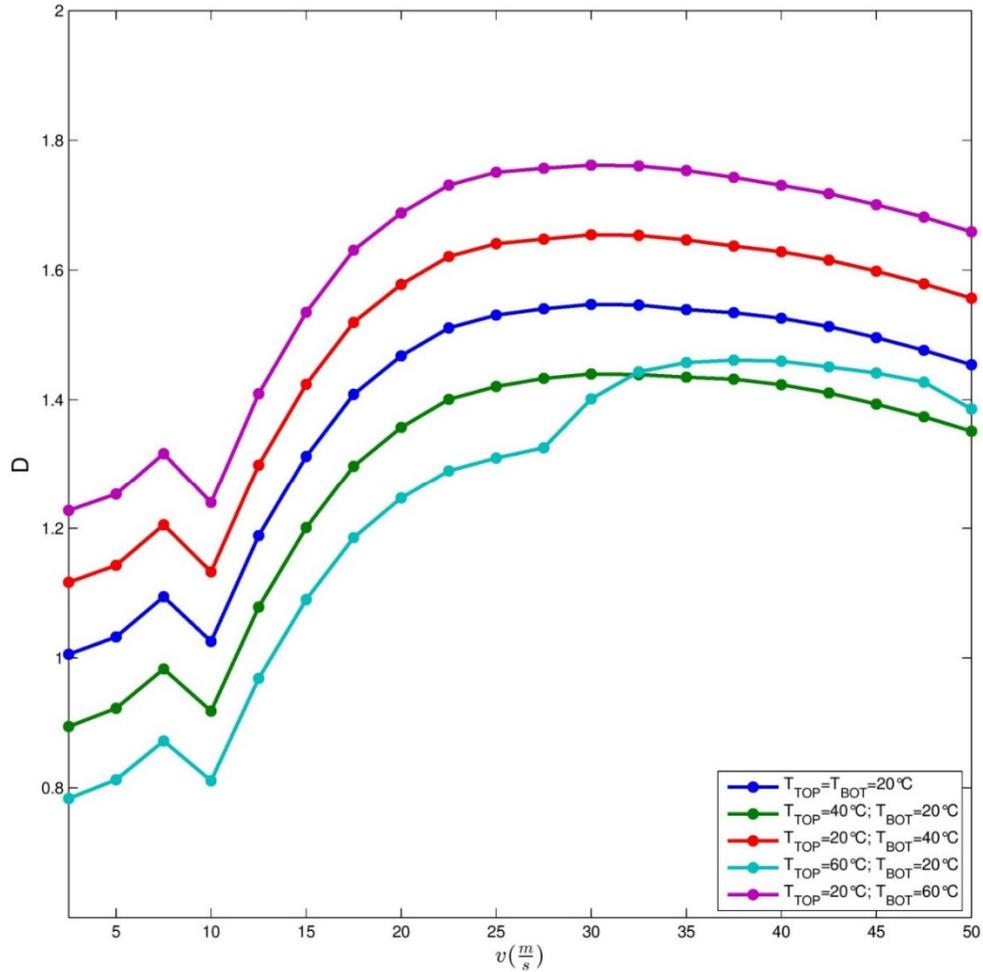


Figure 16: Effect of the speed of the moving load on the maximum absolute non-dimensional vertical displacement, D , for various thermal strains.

7.2 Conclusions

The Finite Element Method and the MPR-MPA time stepping algorithm are effective for studying the effects of dynamic loading and thermal bending on Mindlin-Reissner plates. The thermal bending moment is discretized and superimposed on the beam along with the effects of the moving load. Depending upon the expected temperature differences for a given problem, the thermal bending can have a significant effect on the transverse deflection of the plate. For the given example, the thermal bending increased the transverse deflection by a maximum of 22%.

Chapter 8: Concluding Remarks

Numerical methods are preferred for application involving structures undergoing moving loads. Analytical methods are not capable of handling complicated geometries, material properties, and loads. Experimental methods are not capable of handling multiple iterations of inputs. They can also be costly and time consuming. The preferred method is to use Finite Elements to discretize space and the GSSSS framework to discretize time. Finite Elements allow designers to analyze complicated geometries in space as well as adding any possible mechanical or thermal loads. Within the framework of the GSSSS algorithms, the V0(1,1,0), MPR-MPA, algorithm is preferred because it is robust enough to handle any potential situations, such as nonlinearities.

Three of the most common structures analyzed with Finite Elements are derived using Euler-Bernoulli beam theory, Timoshenko beam theory, and Mindlin-Reissner plate theory. All three can be effectively chosen for these forced vibration problems, but Timoshenko beam elements are the preferred elements. These elements are not conditional to the aspect ratio of the length to height ratio. They are also effective for analyzing in-plane problems such as curved beams. Mindlin-Reissner plate elements are valuable when the loading case is not consistent along the width of the structure., i.e. eccentric loading.

Using Finite Elements, curved beam structures undergoing forced vibration by moving loads can easily be analyzed. It is well known that the magnitude of the dynamic response of a structure undergoing moving loads is a function of the speed of the load. If the structures are curved, the magnitude of the response is also a function of the initial radius of curvature. Whenever allowable, it is preferred to have a structure with as much curvature as possible.

However, this does not hold true if large thermal strains are expected. The effects of thermal expansion are magnified for beams with a large curvature versus those with a small curvature. Thermal bending due to temperature differences through the cross-section of the beam have the inverse effect. The bending loads have a greater impact on curved beams with a smaller curvature. Either way, thermal loading can have an impact magnitudes larger than those of the forced vibration due to moving loads.

These thermal effects are critical to understand for flat plate analyzes as well. Using Finite Elements, it was shown that the magnitude of the dynamic response of plates undergoing moving loads can be greatly affected by thermal loads.

References

- [1] Barron, Randall F., and Brian R. Barron. *Design for Thermal Stresses*. Hoboken, NJ: Wiley, 2012. Print.
- [2] Bathe, Klaus-Jürgen. *Finite Element Procedures*. Englewood Cliffs, NJ: Prentice Hall, 1996. Print.
- [3] Bilello, Cristiano, Lawrence A. Bergman, and Daniel Kuchma. "Experimental Investigation of a Small-Scale Bridge Model under a Moving Mass." *Journal of Structural Engineering* 130.5 (2004): 799-805. Web.
- [4] Boley, Bruno A., Jerome H. Weiner, and W. Prager. "Theory of Thermal Stresses." *Journal of Applied Mechanics* 28.2 (1961): 318. Web.
- [5] Budynas, Richard G., Keith. Nisbett, and Joseph Edward. Shigley. *Shigley's Mechanical Engineering Design*. New York: McGraw-Hill, 2011. Print.
- [6] Dawe, D.j., and O.I. Roufaeil. "Rayleigh-Ritz Vibration Analysis of Mindlin Plates." *Journal of Sound and Vibration* 69.3 (1980): 345-59. Web.
- [7] Ferreira, A. J. M. *MATLAB Codes for Finite Element Analysis Solids and Structures*. Dordrecht: Springer Science & Business Media, 2009. Print.

- [8] Green, M.f., and D. Cebon. "Dynamic Response of Highway Bridges to Heavy Vehicle Loads: Theory and Experimental Validation." *Journal of Sound and Vibration* 170.1 (1994): 51-78. Web.
- [9] Har, Jason, and Kumar K. Tamma. *Advanced Computational Dynamics of Particles, Materials and Structures*. Chichester, West Sussex: Wiley, 2012. Print.
- [10] Hoitink, Andrew J. "Investigations Encompassing the Equations of Motion and Proper Accurate Treatment of Algorithmic Variables: Computational Structural Dynamics and Stiff Systems." Thesis. University of Minnesota, 2009. Web.
- [11] Jiang, J.-Q. "Transient Responses of Timoshenko Beams Subject to a Moving Mass." *Journal of Vibration and Control* 17.13 (2011): 1975-982. Web.
- [12] Krishnan, A., S. Dharmaraj, and Y.j. Suresh. "Free Vibration Studies of Arches." *Journal of Sound and Vibration* 186.5 (1995): 856-63. Web.
- [13] Litewka, Przemysław, and Jerzy Rakowski. "The Exact Thick Arch Finite Element." *Computers & Structures* 68.4 (1998): 369-79. Web.
- [14] Logan, Daryl L. *A First Course in the Finite Element Method*. Pacific Grove, CA: Brooks/Cole, 2002. Print.
- [15] Lou, Ping, Gong-Lian Dai, and Qing-Yuan Zeng. "Dynamic Analysis of a Timoshenko Beam Subjected to Moving Concentrated Forces Using the Finite Element Method." *Shock and Vibration* 14.6 (2007): 459-68. Web.
- [16] Sayyad, A. S. "Comparison of Various Refined Beam Theories for the Bending and Free Vibration Analysis of Thick Beams." *Applied and Computational Mechanics* 5 (2011): 217-30. Web.

- [17] Shimada, Masao. "Integration Framework and Algorithms by Design: Implicit and Explicit Families of Generalized Single Step Single Solve Algorithms by Design in Two- and Single- Field Forms." Thesis. University of Minnesota, 2013. Web.
- [18] Wang, C. M. "Timoshenko Beam-Bending Solutions in Terms of Euler-Bernoulli Solutions." *Journal of Engineering Mechanics* 121.6 (1995): 763-65. Web.
- [19] Wu, Jia-Jang. "Vibration Analyses of a Portal Frame under the Action of a Moving Distributed Mass Using Moving Mass Element." *International Journal for Numerical Methods in Engineering* 62.14 (2005): 2028-052. Web.
- [20] Wu, Jong-Shyong, and Lieh-Kwang Chiang. "Dynamic Analysis of an Arch Due to a Moving Load." *Journal of Sound and Vibration* 269.3-5 (2004): 511-34. Web.
- [21] Wu, Jong-Shyong, Ming-Ling Lee, and Tser-Shyong Lai. "The Dynamic Analysis of a Flat Plate under a Moving Load by the Finite Element Method." *International Journal for Numerical Methods in Engineering* 24.4 (1987): 743-62. Web.
- [22] Xiliang, Chen. "Development of Meshfree Methods for Plates." Thesis. National University of Singapore, 2003. Web.
- [23] Yang, Y.-B., C.-M. Wu, and J.-D. Yau. "Dynamic Response of a Horizontally Curved Beam Subjected To Vertical And Horizontal Moving Loads." *Journal of Sound and Vibration* 242.3 (2001): 519-37. Web.

Appendix A – Static Error Convergence

Plots

Appendix A contains the error convergence plots for the numerical examples from Chapter 2.

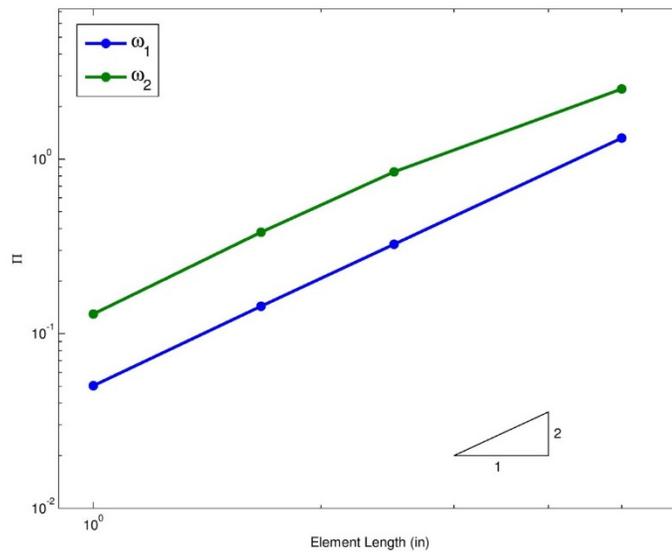


Figure 17: Error convergence of the first two natural frequencies as the element length decreases of Euler-Bernoulli beams from Section 2.1.1

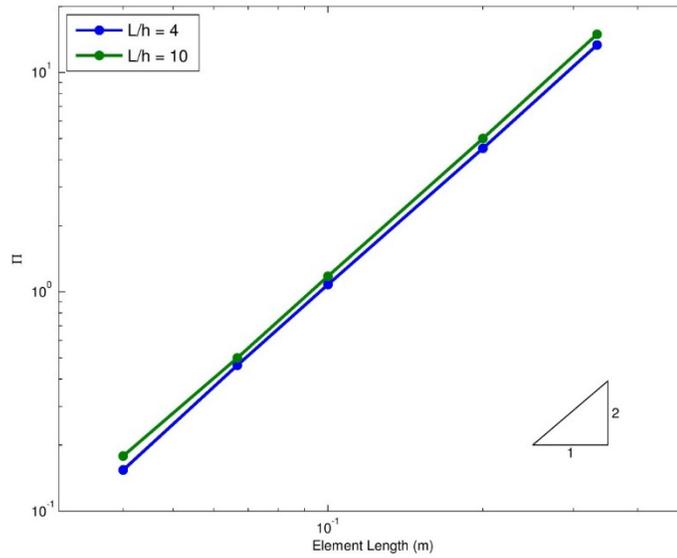


Figure 18: Error convergence of the first natural frequency as the element length decreases of two Timoshenko beams from Section 2.1.1

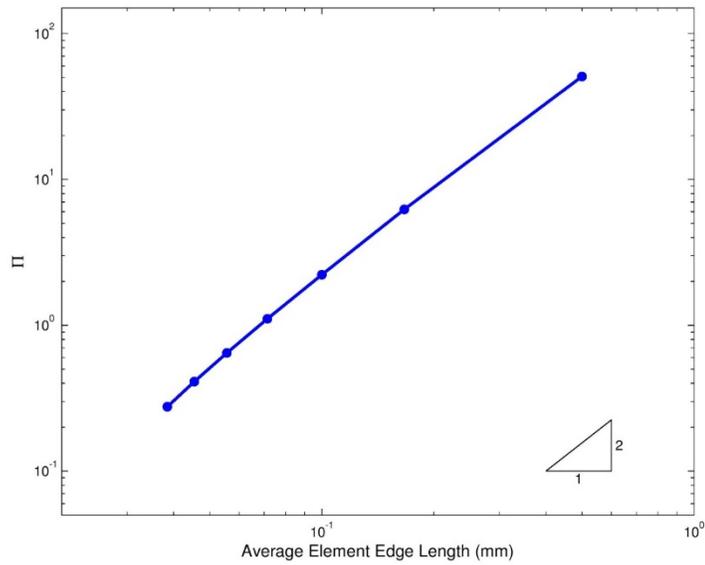


Figure 19: Error convergence for the internal energy, $U = \frac{1}{2} [\mathbf{K}]\{\mathbf{u}\}$ as the average element length decreases for the Mindlin-Reissner plate from Section 2.3.1

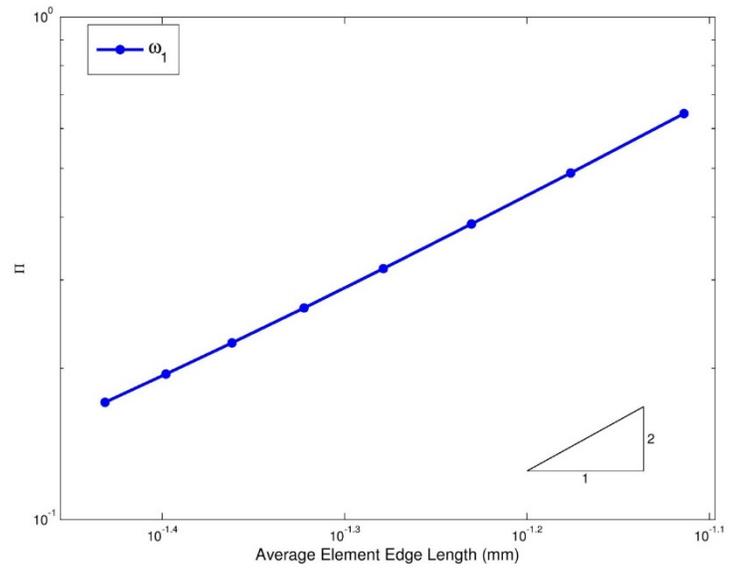


Figure 20: Error convergence for the fundamental natural frequency as the average element length decreases for the Mindlin-Reissner plate from Section 2.3.1

Appendix B – Time Histories for Selected Curved Beams with Moving Loads and Thermal Strains

Appendix B shows the time histories for various examples taken from the numerical examples in Section 6.1.3

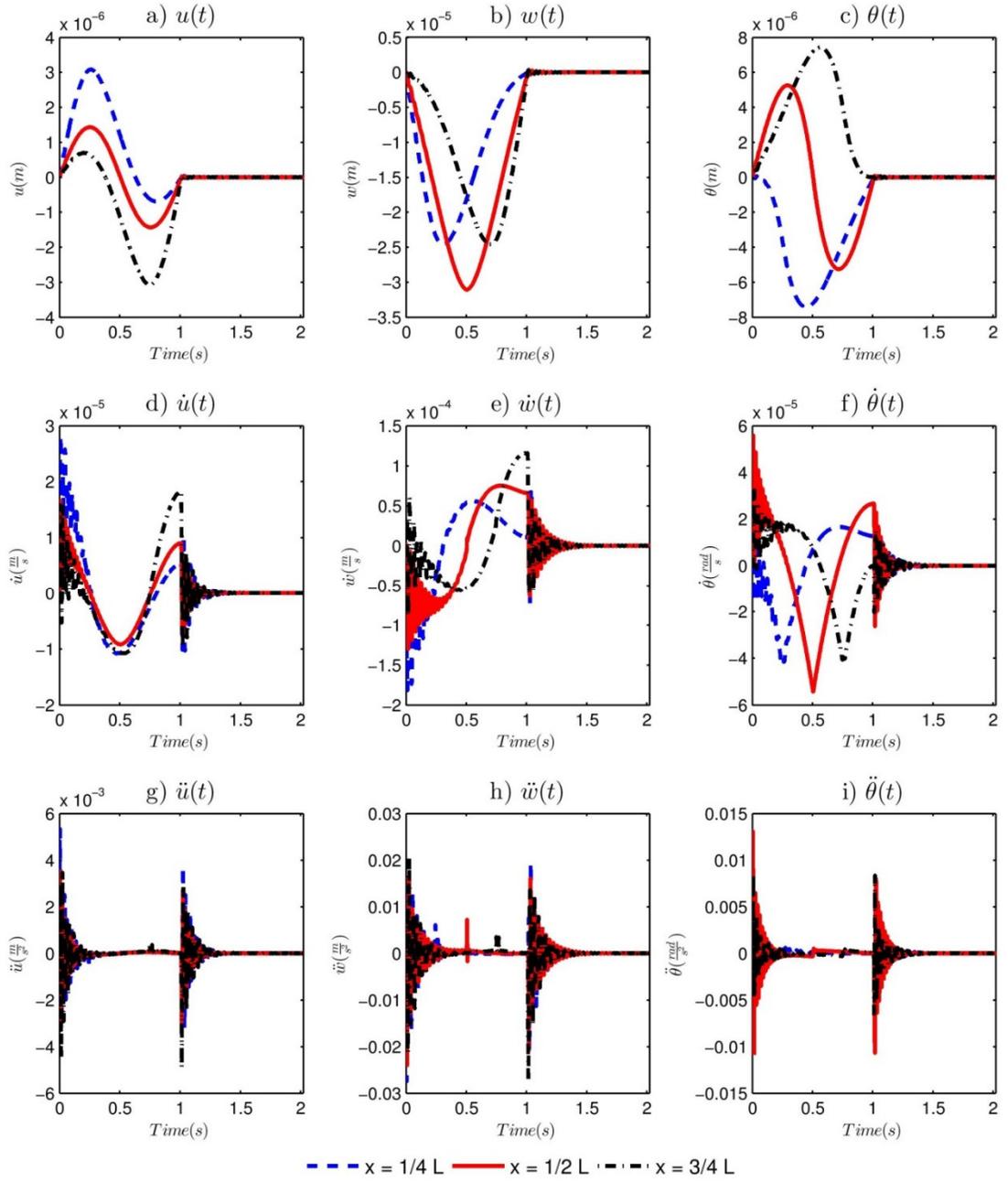


Figure 21: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, & $x = \frac{3L}{4}$ for the moving mass moving at $v = 10 \frac{m}{s}$, $T_{TOP} = 25^\circ C$, $T_{BOT} = 25^\circ C$, $T_\infty = 25^\circ C$, & $R = 20m$ for each of the following: a), $u(t)$, b) $w(t)$, c) $\theta(t)$, d) $\dot{u}(t)$ e) $\dot{w}(t)$, f) $\dot{\theta}(t)$, g) $\ddot{u}(t)$, h) $\ddot{w}(t)$, and i) $\ddot{\theta}(t)$.

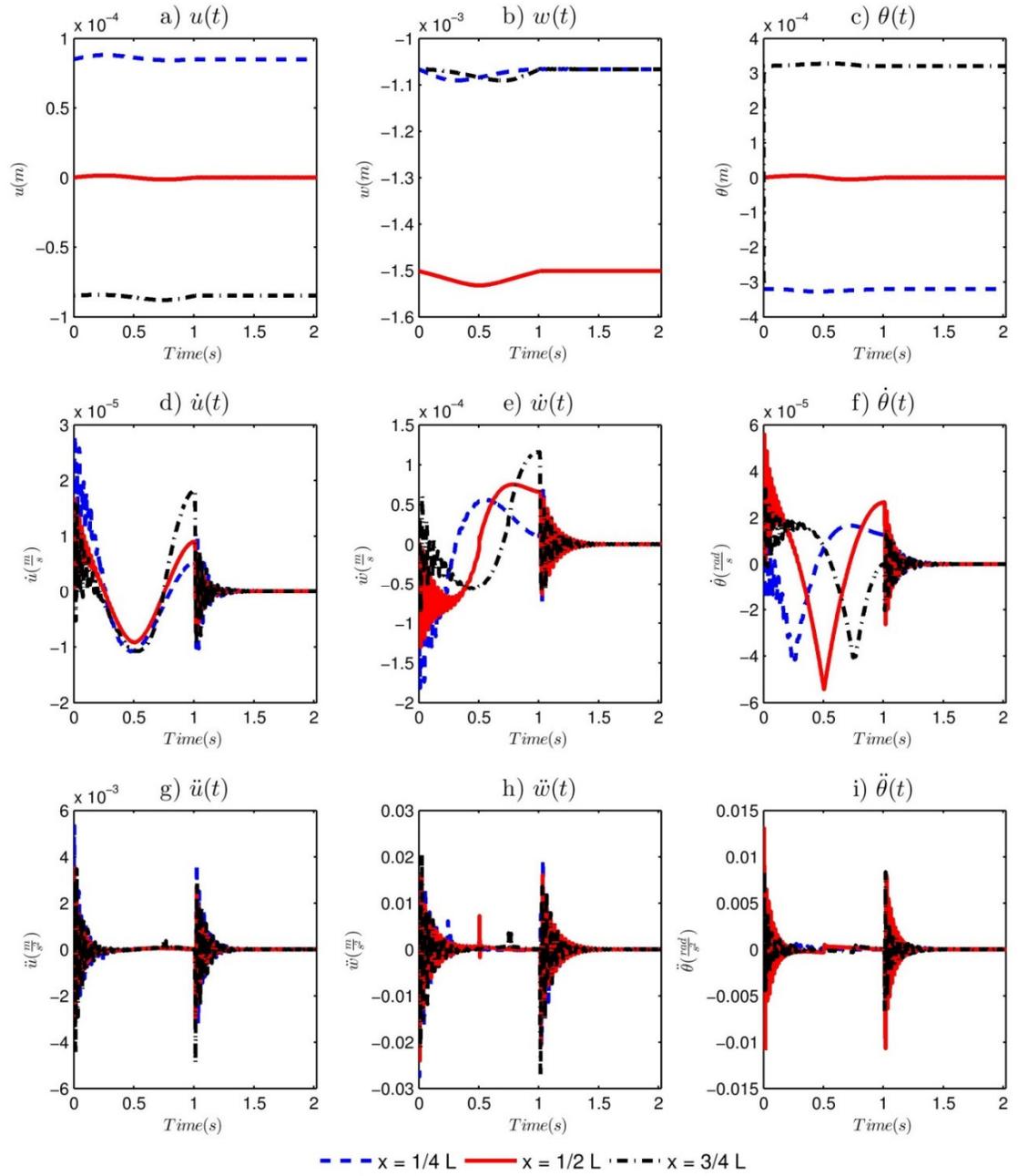


Figure 22: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, & $x = \frac{3L}{4}$ for the moving mass moving at $v = 10 \frac{m}{s}$, $T_{TOP} = 25^\circ C$, $T_{BOT} = 25^\circ C$, $T_\infty = 30^\circ C$, & $R = 20m$ for each of the following: a) $u(t)$, b) $w(t)$, c) $\theta(t)$, d) $\dot{u}(t)$ e) $\dot{w}(t)$, f) $\dot{\theta}(t)$, g) $\ddot{u}(t)$, h) $\ddot{w}(t)$, and i) $\ddot{\theta}(t)$.

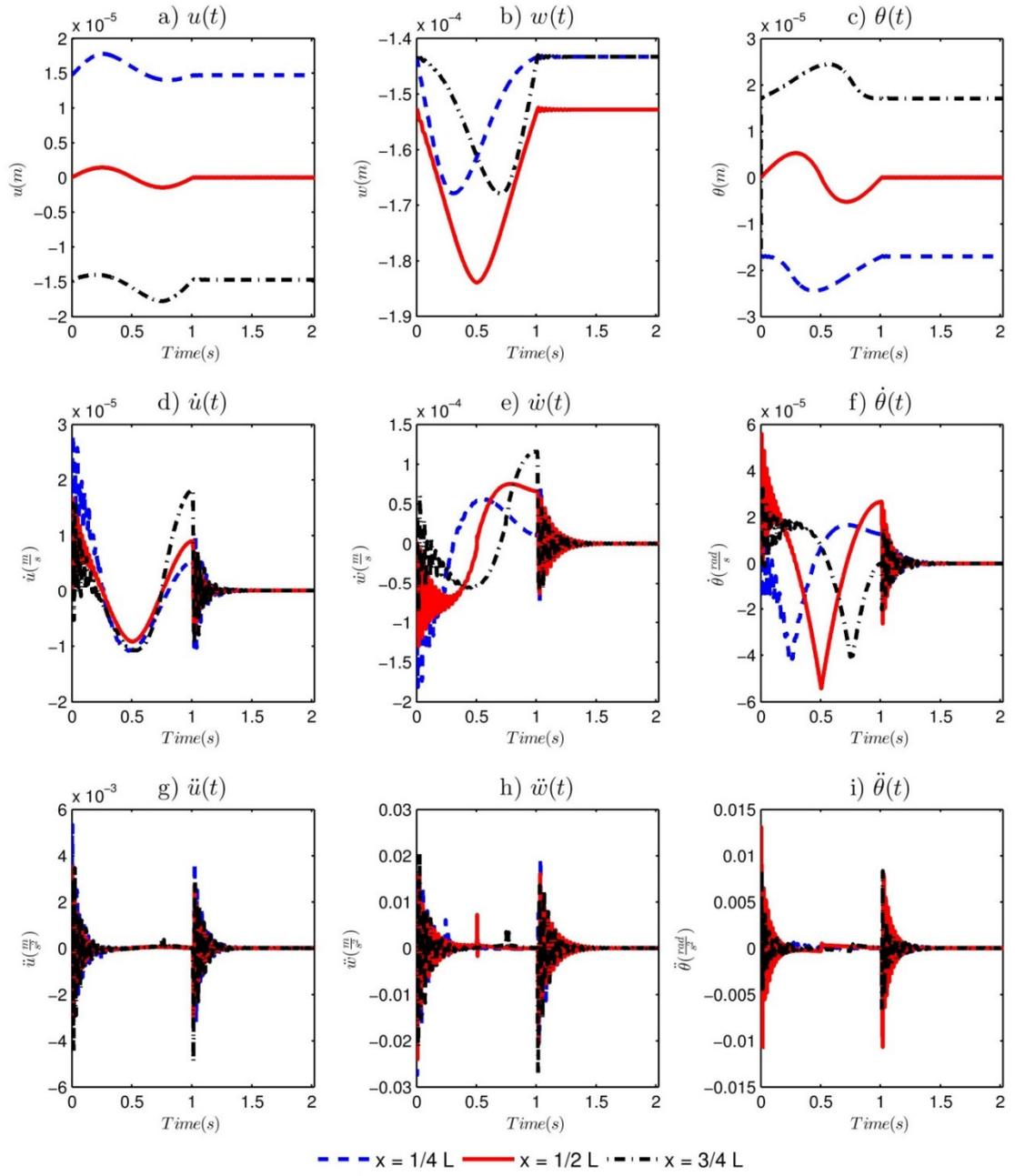


Figure 23: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, & $x = \frac{3L}{4}$ for the moving mass moving at $v = 10 \frac{m}{s}$, $T_{TOP} = 20^\circ C$, $T_{BOT} = 25^\circ C$, $T_\infty = 22.5^\circ C$, & $R = 20m$ for each of the following: a) $u(t)$, b) $w(t)$, c) $\theta(t)$, d) $\dot{u}(t)$ e) $\dot{w}(t)$, f) $\dot{\theta}(t)$, g) $\ddot{u}(t)$, h) $\ddot{w}(t)$, and i) $\ddot{\theta}(t)$.

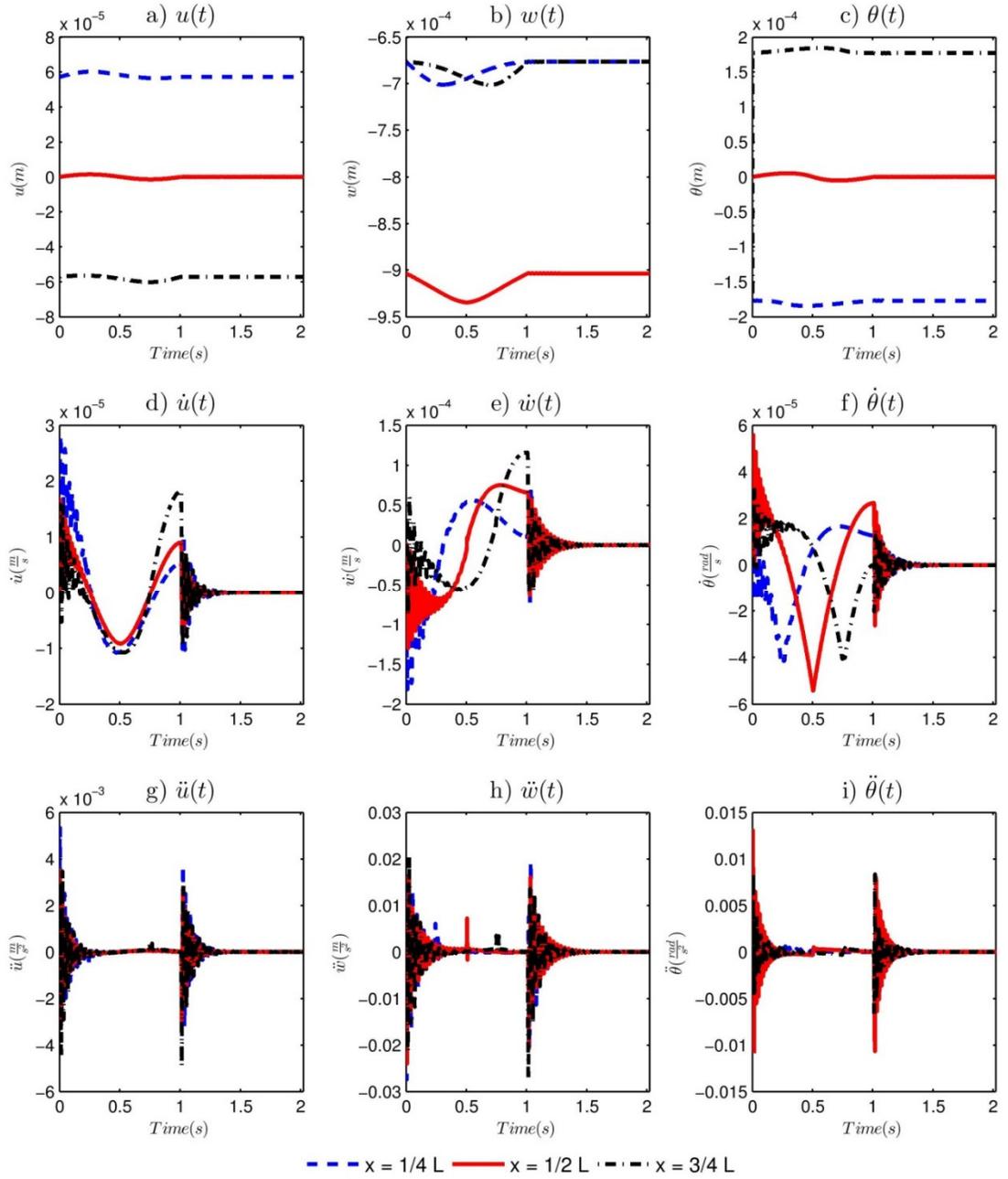


Figure 24: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, & $x = \frac{3L}{4}$ for the moving mass moving at $v = 10 \frac{m}{s}$, $T_{TOP} = 20^\circ C$, $T_{BOT} = 25^\circ C$, $T_\infty = 25^\circ C$, & $R = 20m$ for each of the following: a) $u(t)$, b) $w(t)$, c) $\theta(t)$, d) $\dot{u}(t)$ e) $\dot{w}(t)$, f) $\dot{\theta}(t)$, g) $\ddot{u}(t)$, h) $\ddot{w}(t)$, and i) $\ddot{\theta}(t)$.

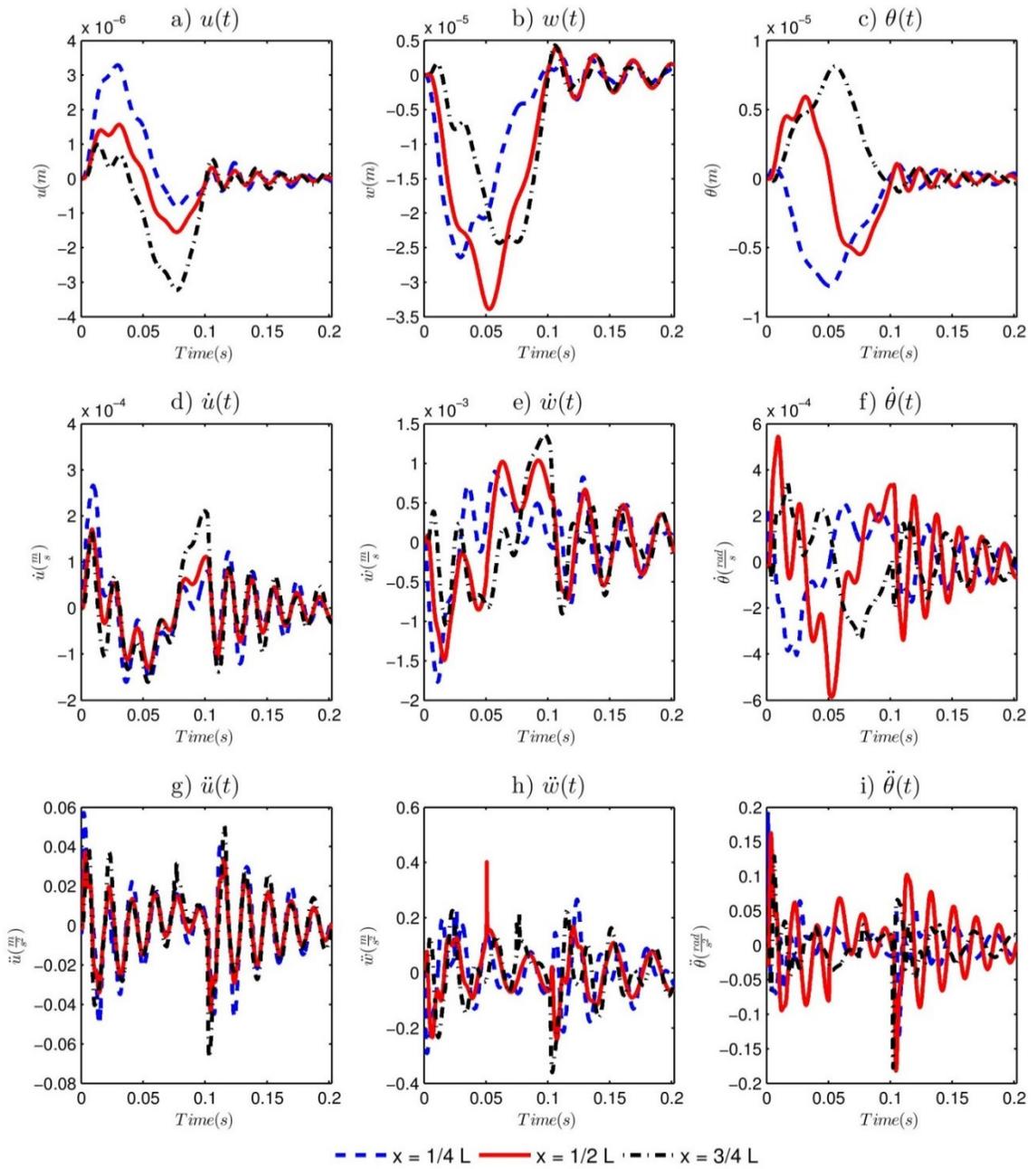


Figure 25: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, & $x = \frac{3L}{4}$ for the moving mass moving at $v = 100 \frac{m}{s}$, $T_{TOP} = 25^\circ C$, $T_{BOT} = 25^\circ C$, $T_\infty = 25^\circ C$, & $R = 20m$ for each of the following: a) $u(t)$, b) $w(t)$, c) $\theta(t)$, d) $\dot{u}(t)$ e) $\dot{w}(t)$, f) $\dot{\theta}(t)$, g) $\ddot{u}(t)$, h) $\ddot{w}(t)$, and i) $\ddot{\theta}(t)$.

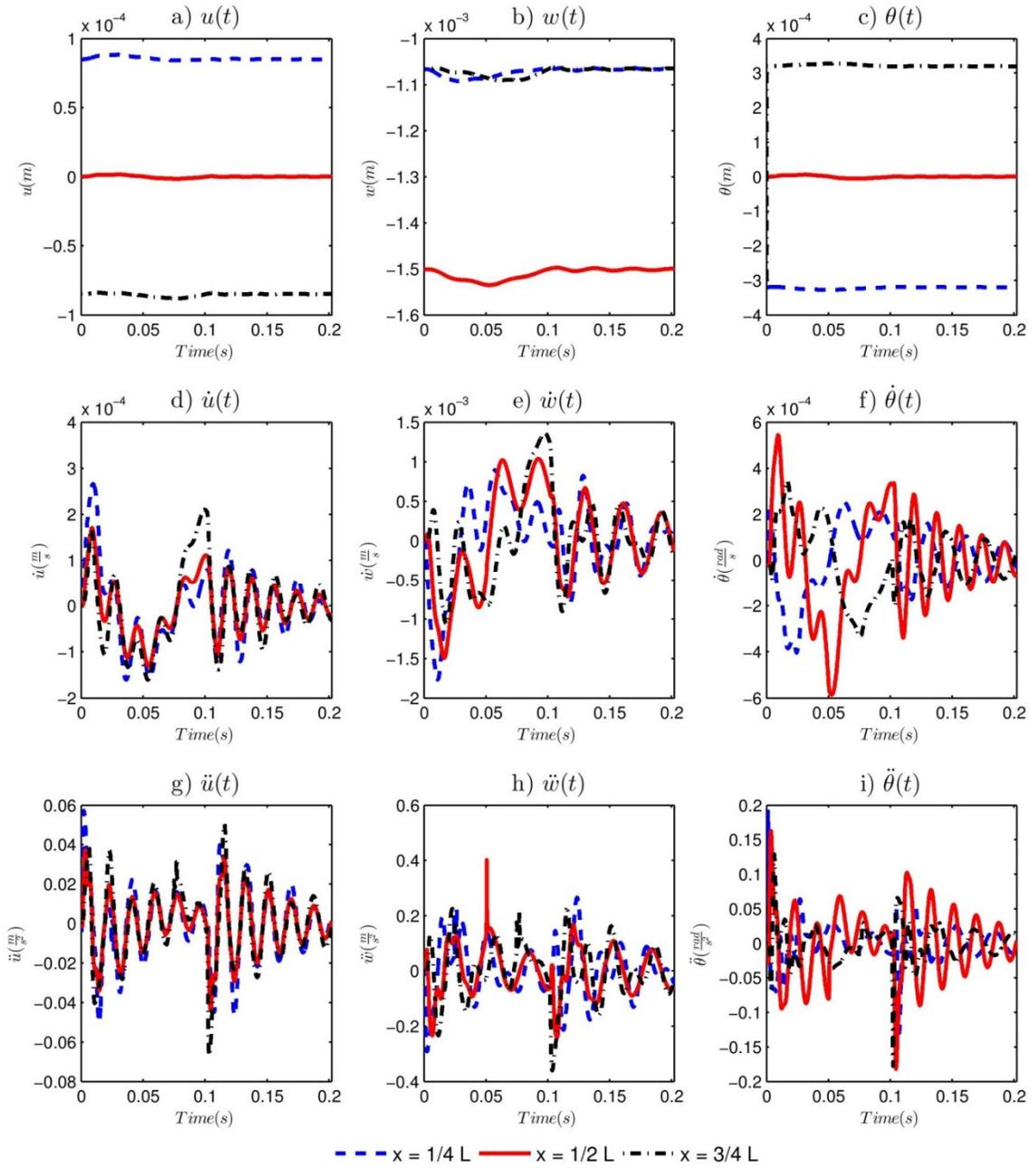


Figure 26: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, & $x = \frac{3L}{4}$ for the moving mass moving at $v = 100 \frac{m}{s}$, $T_{TOP} = 25^\circ C$, $T_{BOT} = 25^\circ C$, $T_\infty = 30^\circ C$, & $R = 20m$ for each of the following: a) $u(t)$, b) $w(t)$, c) $\theta(t)$, d) $\dot{u}(t)$ e) $\dot{w}(t)$, f) $\dot{\theta}(t)$, g) $\ddot{u}(t)$, h) $\ddot{w}(t)$, and i) $\ddot{\theta}(t)$.

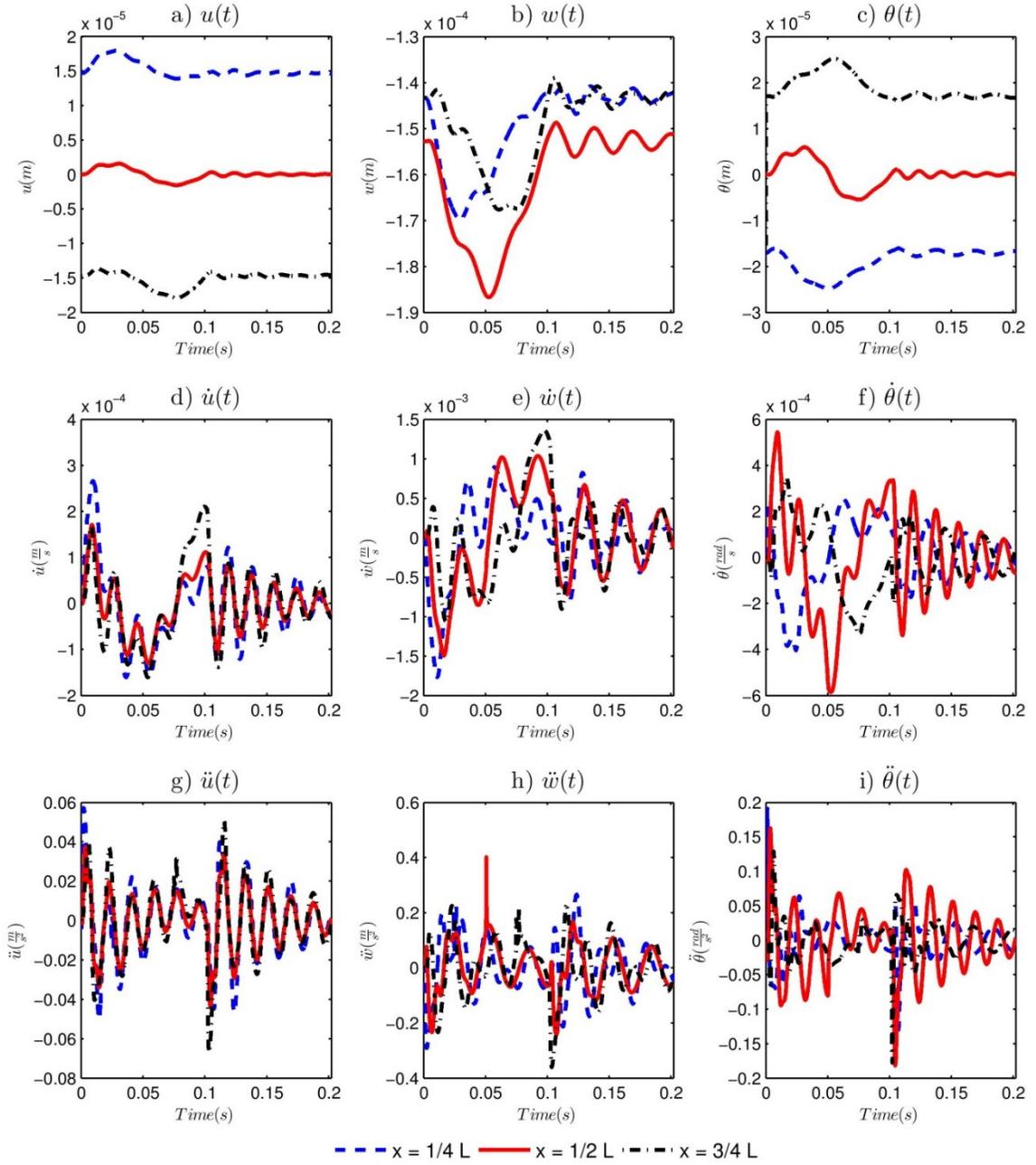


Figure 27: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, & $x = \frac{3L}{4}$ for the moving mass moving at $v = 100 \frac{m}{s}$, $T_{TOP} = 20^\circ C$, $T_{BOT} = 25^\circ C$, $T_\infty = 22.5^\circ C$, & $R = 20m$ for each of the following: a), $u(t)$, b) $w(t)$, c) $\theta(t)$, d) $\dot{u}(t)$ e) $\dot{w}(t)$, f) $\dot{\theta}(t)$, g) $\ddot{u}(t)$, h) $\ddot{w}(t)$, and i) $\ddot{\theta}(t)$.

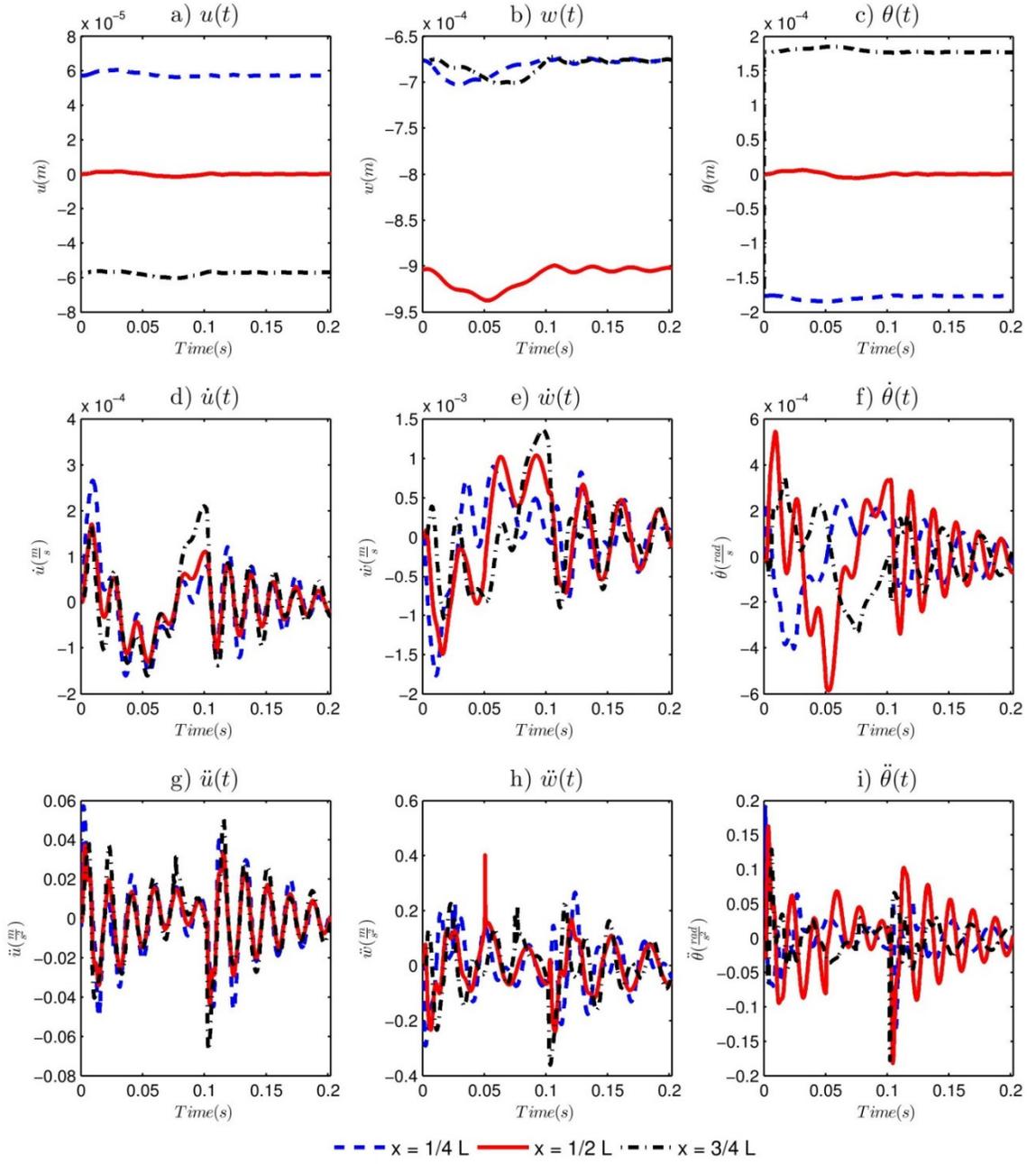


Figure 28: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, & $x = \frac{3L}{4}$ for the moving mass moving at $v = 100 \frac{m}{s}$, $T_{TOP} = 20^\circ C$, $T_{BOT} = 25^\circ C$, $T_\infty = 25^\circ C$, & $R = 20m$ for each of the following: a) $u(t)$, b) $w(t)$, c) $\theta(t)$, d) $\dot{u}(t)$ e) $\dot{w}(t)$, f) $\dot{\theta}(t)$, g) $\ddot{u}(t)$, h) $\ddot{w}(t)$, and i) $\ddot{\theta}(t)$.

Appendix C – Time Histories for Selected Plates with Moving Loads and Thermal Strains

Appendix C shows the time histories for various examples taken from the numerical examples in Section 6.1.3

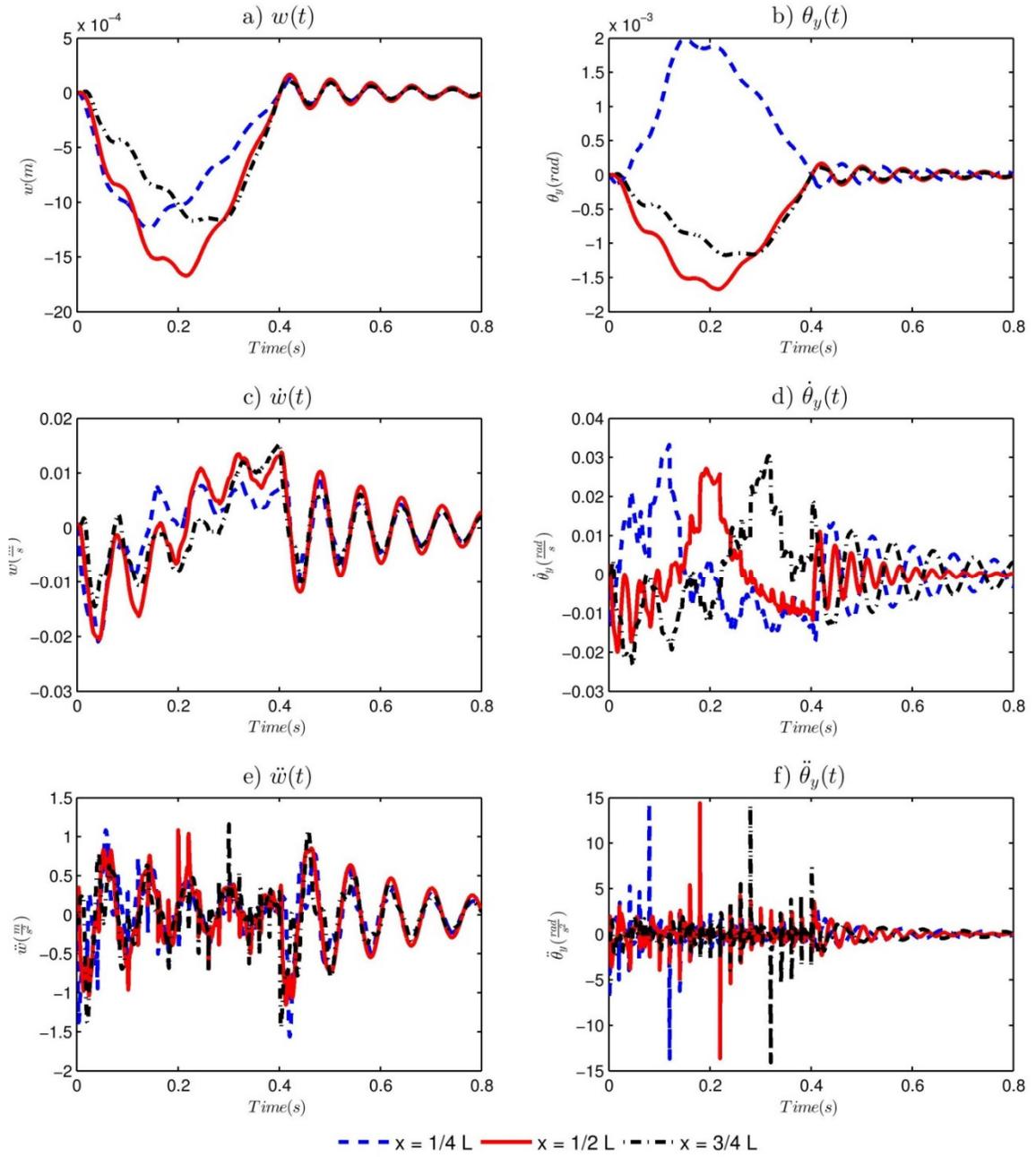


Figure 29: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, $x = \frac{3L}{4}$, and $y = \frac{W}{2}$ for $T_{TOP} = 20^\circ C$, $T_{BOT} = 20^\circ C$, and $v = 5 \frac{m}{s}$ for: a) $w(t)$, b) $\theta_y(t)$, c) $\dot{w}(t)$, d) $\dot{\theta}_y(t)$, e) $\ddot{w}(t)$, and f) $\ddot{\theta}_y(t)$.

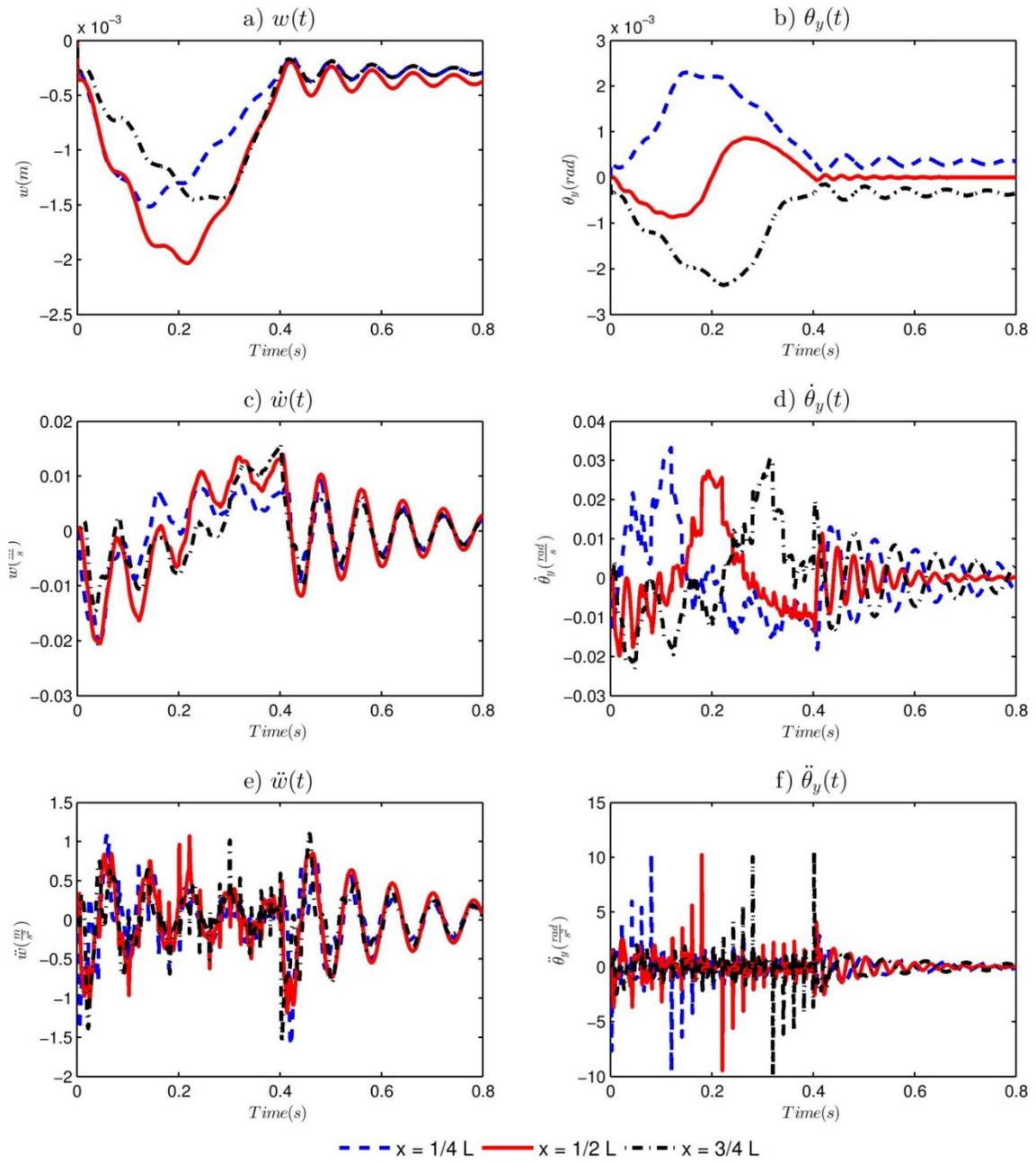


Figure 30: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, $x = \frac{3L}{4}$, and $y = \frac{W}{2}$ for $T_{TOP} = 20^\circ C$, $T_{BOT} = 60^\circ C$, and $v = 5 \frac{m}{s}$ for: a) $w(t)$, b) $\theta_y(t)$, c) $\dot{w}(t)$, d) $\dot{\theta}_y(t)$, e) $\ddot{w}(t)$, and f) $\ddot{\theta}_y(t)$.

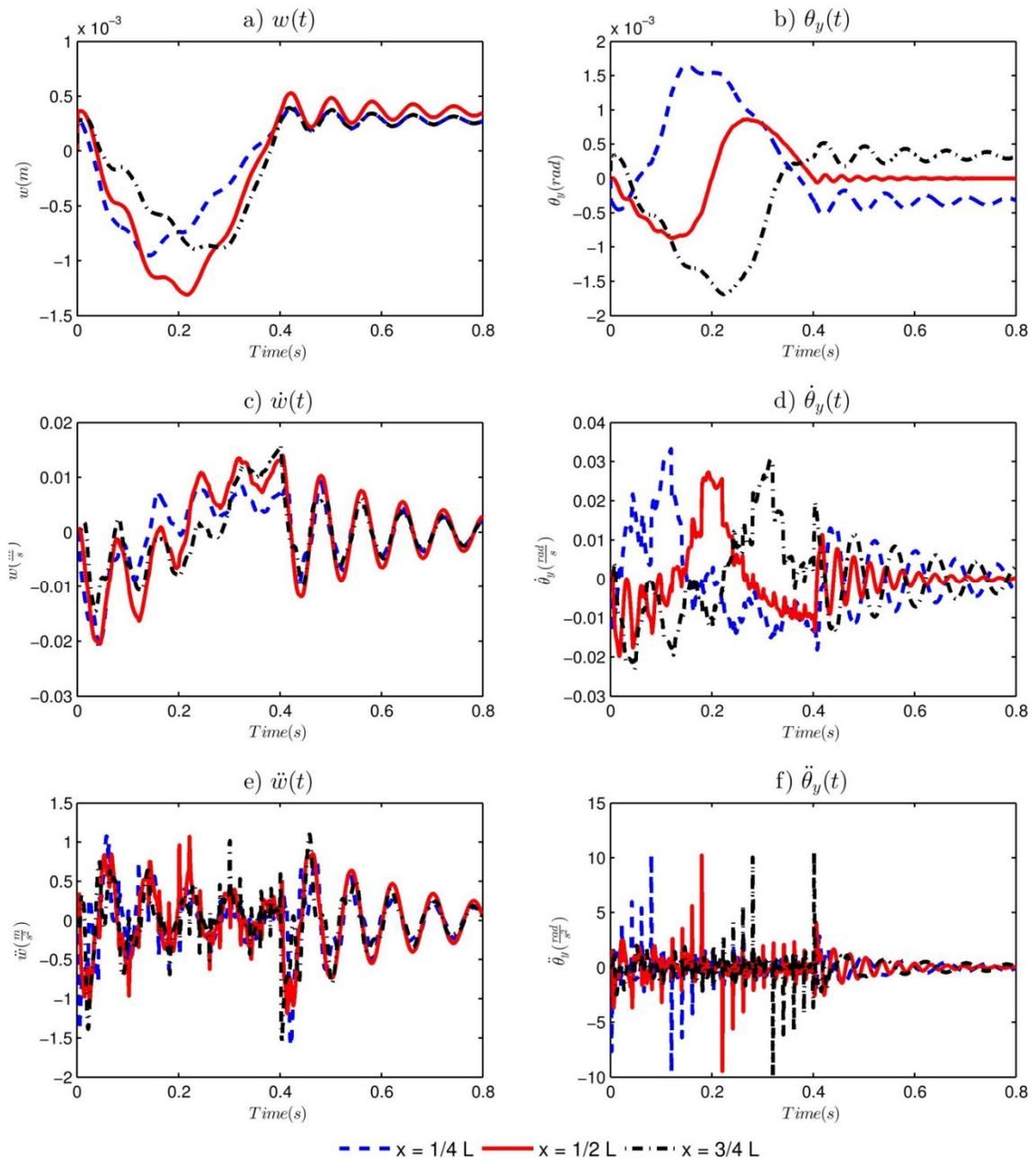


Figure 31: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, $x = \frac{3L}{4}$, and $y = \frac{W}{2}$ for $T_{TOP} = 60^\circ C$, $T_{BOT} = 20^\circ C$, and $v = 5 \frac{m}{s}$ for: a) $w(t)$, b) $\theta_y(t)$, c) $\dot{w}(t)$, d) $\dot{\theta}_y(t)$, e) $\ddot{w}(t)$, and f) $\ddot{\theta}_y(t)$.

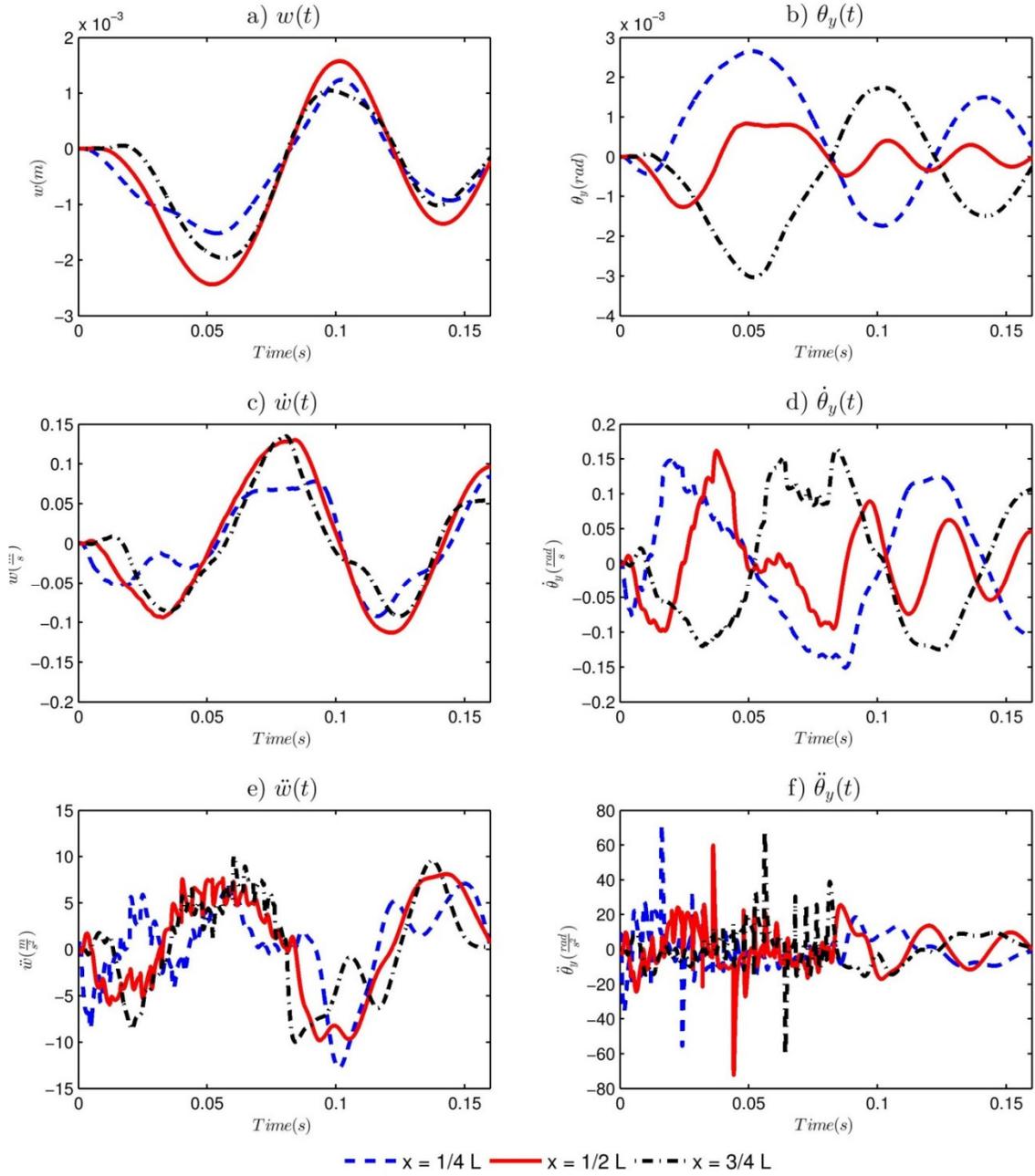


Figure 32: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, $x = \frac{3L}{4}$, and $y = \frac{W}{2}$ for $T_{TOP} = 20^\circ C$, $T_{BOT} = 20^\circ C$, and $v = 25 \frac{m}{s}$ for: a) $w(t)$, b) $\theta_y(t)$, c) $\dot{w}(t)$, d) $\dot{\theta}_y(t)$, e) $\ddot{w}(t)$, and f) $\ddot{\theta}_y(t)$.

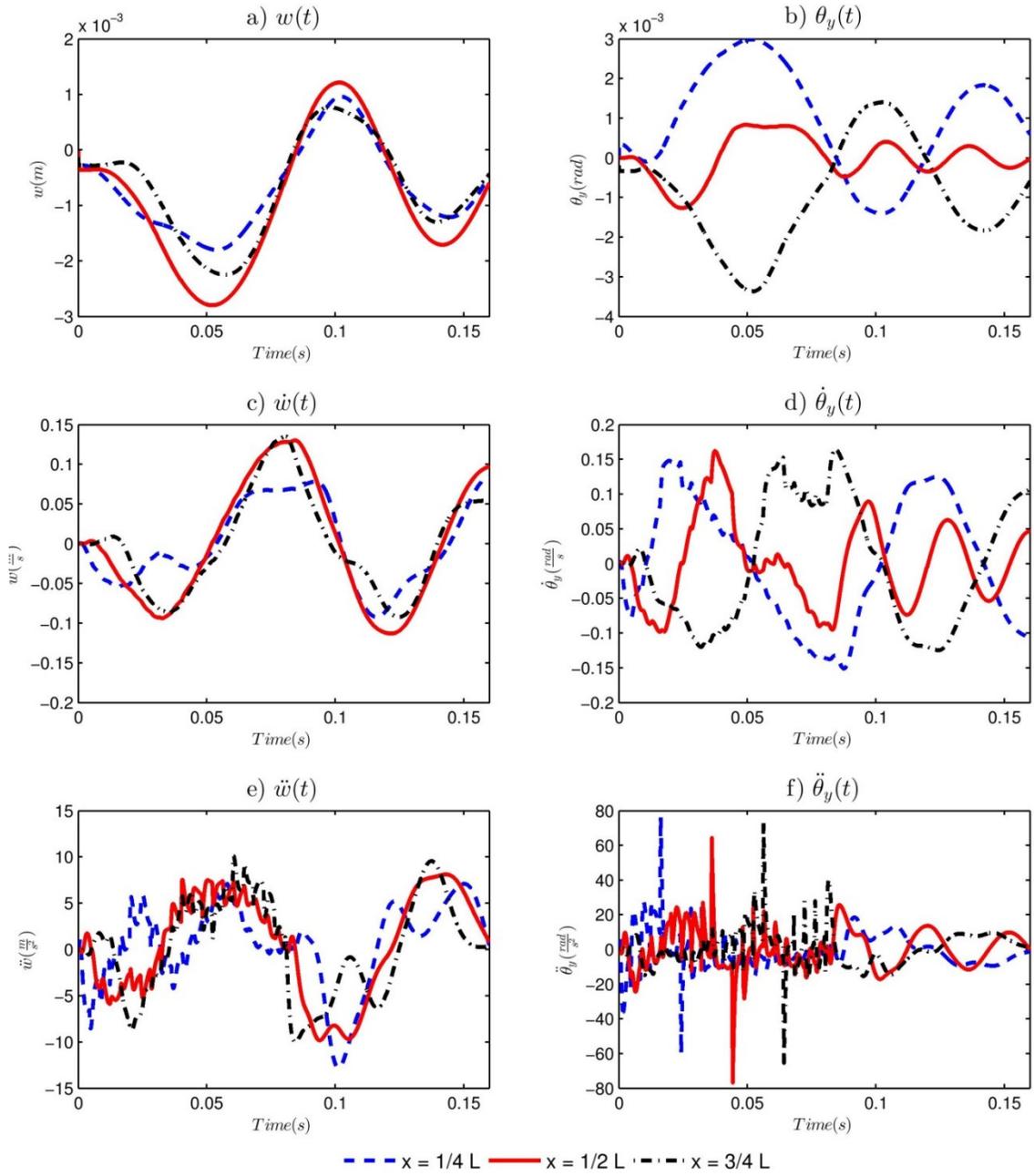


Figure 33: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, $x = \frac{3L}{4}$, and $y = \frac{W}{2}$ for $T_{TOP} = 20^\circ C$, $T_{BOT} = 60^\circ C$, and $v = 25 \frac{m}{s}$ for: a) $w(t)$, b) $\theta_y(t)$, c) $\dot{w}(t)$, d) $\dot{\theta}_y(t)$, e) $\ddot{w}(t)$, and f) $\ddot{\theta}_y(t)$.

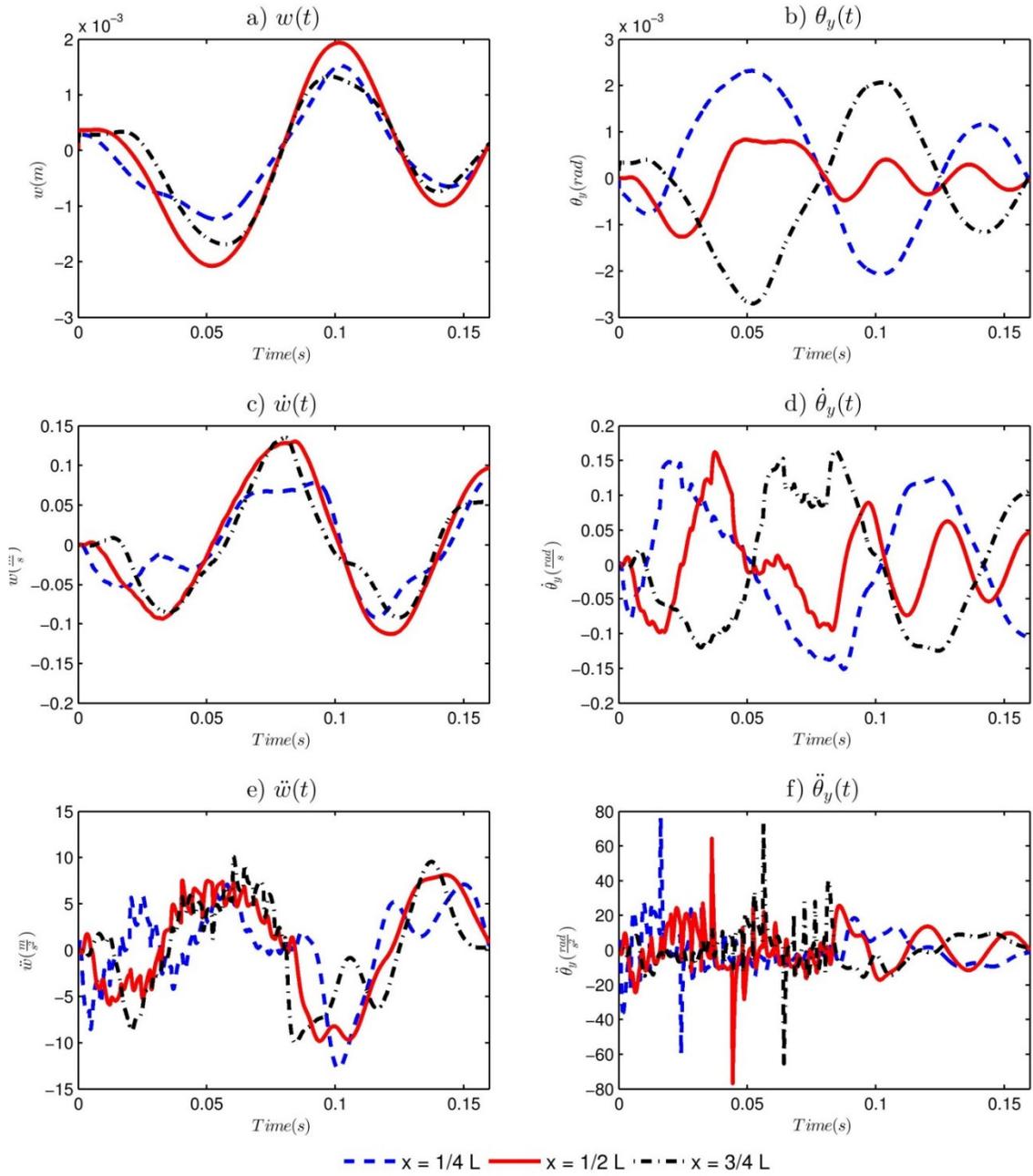


Figure 34: Time histories at $x = \frac{L}{4}$, $x = \frac{L}{2}$, $x = \frac{3L}{4}$, and $y = \frac{W}{2}$ for $T_{TOP} = 60^\circ C$, $T_{BOT} = 20^\circ C$, and $v = 25 \frac{m}{s}$ for: a) $w(t)$, b) $\theta_y(t)$, c) $\dot{w}(t)$, d) $\dot{\theta}_y(t)$, e) $\ddot{w}(t)$, and f) $\ddot{\theta}_y(t)$.

Appendix D – Matlab Codes

```
%Curved_beam_moving_mass_thermal_strain.m

clear
clc
close all

%moving mass
mass=5e4/9.81; %kg
g=-9.81; %m/s^2
vel=100; %m/s
time_steps=1000;

%GSSS-2
P=[1 1 0];

%Centrifugal Force: 1=on 0=off
CF=0;

Nodes=51;
elements=Nodes-1;
nodes=(elements*3)+1;

span=10;
R=20;

arch_angle=2*asin(span/(2*R));
xsection_width=1.5;
xsection_height=0.8;
A=xsection_height*xsection_width;
I=(xsection_width*xsection_height^3/12);
pr=0.3;
E=12e10;
G=E/(2*(1+pr));
k=5/6;
rho=7.2e3;
Alpha=11.9e-6;

T_top=25;
T_bottom=25;
T_avg=0.5*(T_top+T_bottom);
T_inf=25;

%Proportional Damping Coefs
xi1=0.05;
xi2=0.05;
Omega1=157;
Omega2=334;
alpha=2*Omega1*Omega2*(xi1*Omega2-xi2*Omega1)/(Omega2^2-Omega1^2);
beta=2*(xi2*Omega2-xi1*Omega1)/(Omega2^2-Omega1^2);
```

```

%create mesh
element_angle=arch_angle/(Nodes-1);
mesh_angles=element_angle:element_angle:arch_angle;
init_vector=-0.5*span+R*cos(arch_angle/2)*i;
mesh_vectors=init_vector*(-1+exp(-i*mesh_angles));
meshx=[0 real(mesh_vectors)];
meshy=[0 imag(mesh_vectors)];

K=zeros(3*nodes,3*nodes);
M=zeros(3*nodes,3*nodes);
F=zeros(3*nodes,1);
QF=zeros(3*nodes,1);

unit_x=[0 1/3 2/3 1];
unit_y=[0 0 0 0];

total_calcs=elements;
iter_time=0;
h=waitbar(0,'Assembling Stiffness & Mass Matrices...');
for index1=1:elements;
    tic;
    est_time=total_calcs*iter_time/60;
    est_time_min=floor(est_time);
    est_time_sec=(est_time-est_time_min)*60;
    waitbar(index1/elements,h,sprintf('Assembling Stiffness & Mass Matrices... %3.0f:%02.0f
remaining',...
    [est_time_min est_time_sec]))

    K_temp=zeros(3*nodes,3*nodes);
    M_temp=zeros(3*nodes,3*nodes);
    QF_temp=zeros(3*nodes,1);
    element_angle(index1)=atan((meshy(index1+1)-meshy(index1))/(meshx(index1+1)-meshx(index1)));
    element_length(index1)=sqrt((meshy(index1+1)-meshy(index1))^2+(meshx(index1+1)-
meshx(index1))^2);
    s=sin(element_angle(index1));
    c=cos(element_angle(index1));
    T_unit=[c s 0;-s c 0;0 0 1];
    T=blkdiag(T_unit,T_unit,T_unit,T_unit);
    trans=[1 0 meshx(index1);0 1 meshy(index1);0 0 1];
    rot=[cos(element_angle(index1)) -sin(element_angle(index1)) 0;sin(element_angle(index1))
cos(element_angle(index1)) 0;0 0 1];
    scale=[element_length(index1) 0 0;0 0 0;0 0 1];
    element_cords=trans*rot*scale*[unit_x;unit_y;1 1 1 1];
    element_xs=element_cords(1,:);
    element_ys=element_cords(2,:);
    gauss_loc=[sqrt(0.6) sqrt((3+2*sqrt(6/5))/7) sqrt((3-2*sqrt(6/5))/7)];
    gauss_loc1=[-gauss_loc(2) -gauss_loc(3) gauss_loc(3) gauss_loc(2)];
    gauss_loc2=[-gauss_loc(1) 0 gauss_loc(1)];
    gauss_weights1=[(18-sqrt(30))/36 (18+sqrt(30))/36 (18+sqrt(30))/36 (18-sqrt(30))/36];
    gauss_weights2=[5/9 8/9 5/9];
    Bs=zeros(3,12);
    Bb=zeros(3,12);
    Ba=zeros(3,12);

```

```

k_mat=zeros(12,12);
m_mat=zeros(12,12);
qf_vec=zeros(12,1);

M_element=-Alpha*(T_top-T_bottom)*E*I/xsection_height;
F_element=Alpha*(T_avg-T_inf)*A*E;

for index2=1:4;
    xi=gauss_loc1(index2);
    N=[-(9*(xi-1)*(xi-1/3)*(xi+1/3))/16,...
        (27*(xi-1)*(xi+1)*(xi-1/3))/16,...
        -(27*(xi-1)*(xi+1)*(xi+1/3))/16,...
        (9*(xi+1)*(xi-1/3)*(xi+1/3))/16];
    dNdx1=[-(9*(xi-1)*(xi-1/3))/16-(9*(xi-1)*(xi+1/3))/16-(9*(xi-1/3)*(xi+1/3))/16,...
        (27*(xi-1)*(xi+1))/16+(27*(xi-1)*(xi-1/3))/16+(27*(xi+1)*(xi-1/3))/16,...
        -(27*(xi-1)*(xi+1))/16-(27*(xi-1)*(xi+1/3))/16-(27*(xi+1)*(xi+1/3))/16,...
        (9*(xi+1)*(xi-1/3))/16+(9*(xi+1)*(xi+1/3))/16+(9*(xi-1/3)*(xi+1/3))/16];
    detJ=element_length(index1)/2;
    invJ=1/detJ;
    dNdx1=dNdx1*invJ;
    Bb(2,:)=[0 0 dNdx1(1) 0 0 dNdx1(2) 0 0 dNdx1(3) 0 0 dNdx1(4)];
    Ba(1,:)=[dNdx1(1) 0 0 dNdx1(2) 0 0 dNdx1(3) 0 0 dNdx1(4) 0 0];

    k_mat=k_mat+Bb'*Bb*E*I*detJ*gauss_weights1(index2)+Ba'*Ba*E*A*detJ*gauss_weights1(index2);
    m_mat=m_mat+rho*A*[N(1) 0 0 N(2) 0 0 N(3) 0 0 N(4) 0 0]*[N(1) 0 0 N(2) 0 0 N(3) 0 0 N(4) 0
0]*detJ*gauss_weights1(index2);
    m_mat=m_mat+rho*A*[0 N(1) 0 0 N(2) 0 0 N(3) 0 0 N(4) 0]*[0 N(1) 0 0 N(2) 0 0 N(3) 0 0 N(4)
0]*detJ*gauss_weights1(index2);
    m_mat=m_mat+rho*I*[0 0 N(1) 0 0 N(2) 0 0 N(3) 0 0 N(4)]*[0 0 N(1) 0 0 N(2) 0 0 N(3) 0 0
N(4)]*detJ*gauss_weights1(index2);
    qf_vec=qf_vec+M_element.*Bb(2,:)*detJ*gauss_weights1(index2);
    qf_vec=qf_vec+F_element.*[dNdx1(1) 0 0 dNdx1(2) 0 0 dNdx1(3) 0 0 dNdx1(4) 0
0]*detJ*gauss_weights1(index2);
end
for index2=1:3;
    xi=gauss_loc2(index2);
    N=[-(9*(xi-1)*(xi-1/3)*(xi+1/3))/16,...
        (27*(xi-1)*(xi+1)*(xi-1/3))/16,...
        -(27*(xi-1)*(xi+1)*(xi+1/3))/16,...
        (9*(xi+1)*(xi-1/3)*(xi+1/3))/16];
    dNdx1=[-(9*(xi-1)*(xi-1/3))/16-(9*(xi-1)*(xi+1/3))/16-(9*(xi-1/3)*(xi+1/3))/16,...
        (27*(xi-1)*(xi+1))/16+(27*(xi-1)*(xi-1/3))/16+(27*(xi+1)*(xi-1/3))/16,...
        -(27*(xi-1)*(xi+1))/16-(27*(xi-1)*(xi+1/3))/16-(27*(xi+1)*(xi+1/3))/16,...
        (9*(xi+1)*(xi-1/3))/16+(9*(xi+1)*(xi+1/3))/16+(9*(xi-1/3)*(xi+1/3))/16];
    dNdx1=dNdx1*invJ;
    Bs(3,:)=[0 dNdx1(1) -N(1) 0 dNdx1(2) -N(2) 0 dNdx1(3) -N(3) 0 dNdx1(4) -N(4)];
    k_mat=k_mat+Bs'*Bs*k*G*A*detJ*gauss_weights2(index2);
end
a=(index1-1)*9+1;
b=a+length(k_mat)-1;
K_temp(a:b,a:b)=T'*k_mat*T;
M_temp(a:b,a:b)=T'*m_mat*T;
QF_temp(a:b,1)=T'*qf_vec;
K=K+K_temp;

```

```

M=M+M_temp;
QF=QF+QF_temp;
Global_x((index1-1)*3+1:(index1-1)*3+4)=element_xs;
Global_y((index1-1)*3+1:(index1-1)*3+4)=element_ys;

toc_temp=toc;
total_calcs=total_calcs-1;
iter_time=(iter_time*(index1-1)+toc_temp)/index1;
end
close(h)
F=F+QF;
F_temp=F;
F_star=F;

K_star=K;
K_star(1,:)=zeros(nodes*3,1);
K_star(:,1)=zeros(1,nodes*3);
K_star(1,1)=1;
K_star(2,:)=zeros(nodes*3,1);
K_star(:,2)=zeros(1,nodes*3);
K_star(2,2)=1;
K_star(end-2,:)=zeros(nodes*3,1);
K_star(:,end-2)=zeros(1,nodes*3);
K_star(end-2,end-2)=1;
K_star(end-1,:)=zeros(nodes*3,1);
K_star(:,end-1)=zeros(1,nodes*3);
K_star(end-1,end-1)=1;

F_star(1)=0;
F_star(2)=0;
F_star(end-2)=0;
F_star(end-1)=0;

Gx=Global_x';
Gy=Global_y';

C=alpha*M+beta*K;

x_vect=[0 cumsum(element_length)];
L=sum(element_length);
t_f=2*L/vel;
delta_t=t_f/time_steps;
time=0:delta_t:t_f;
mass_position=vel*time;

pmin=P(1);
pmax=P(2);
ps=P(3);
phi=1/(1+ps)-0.5;

Lambda1W1=(3+pmin+pmax-pmin*pmax)/(2*(1+pmin)*(1+pmax));
Lambda2W2=1/((1+pmin)*(1+pmax));
Lambda3W3=1/((1+pmin)*(1+pmax)*(1+ps));
Lambda4W1=(3+pmin+pmax-pmin*pmax)/(2*(1+pmin)*(1+pmax));

```

```

Lambda5W2=2/((1+pmin)*(1+pmax)*(1+ps));
Lambda6W1=(2+pmin+pmax+ps-pmin*pmax*ps)/((1+pmin)*(1+pmax)*(1+ps));
W1=(3+pmin+pmax-pmin*pmax)/(2*(1+pmin)*(1+pmax));
lambda1=1;
lambda2=0.5;
lambda3=1/(2*(1+ps));
lambda4=1;
lambda5=1/(1+ps);

Phi=K_star\F_star;
Phi_dot=zeros(3*nodes,1);
Phi_dbldot=zeros(3*nodes,1);

index1=1;
u(:,index1)=Phi(1:3:end,index1);
v(:,index1)=Phi(2:3:end,index1);
theta(:,index1)=Phi(3:3:end,index1);

u_14(:,index1)=interp1(x_vect',u(1:3:end,index1),0.25*L);
v_14(:,index1)=interp1(x_vect',v(1:3:end,index1),0.25*L);
theta_14(:,index1)=interp1(x_vect',theta(1:3:end,index1),0.25*L);

u_12(:,index1)=interp1(x_vect',u(1:3:end,index1),0.5*L);
v_12(:,index1)=interp1(x_vect',v(1:3:end,index1),0.5*L);
theta_12(:,index1)=interp1(x_vect',theta(1:3:end,index1),0.5*L);

u_34(:,index1)=interp1(x_vect',u(1:3:end,index1),0.75*L);
v_34(:,index1)=interp1(x_vect',v(1:3:end,index1),0.75*L);
theta_34(:,index1)=interp1(x_vect',theta(1:3:end,index1),0.25*L);

Gx(:,index1)=u(:,index1)+Global_x';
Gy(:,index1)=v(:,index1)+Global_y';

wb_index=0;
total_calcs=length(time);
iter_time=0;
h=waitbar(0,'Calculating');
for index1=2:length(time);
    tic;
    est_time=total_calcs*iter_time/60;
    est_time_min=floor(est_time);
    est_time_sec=(est_time-est_time_min)*60;
    waitbar(index1/length(time),h,sprintf('Solving... %3.0f:%02.0f remaining.',...
        [est_time_min est_time_sec]))

    Phi_n=Phi(:,index1-1);
    Phi_dot_n=Phi_dot(:,index1-1);
    Phi_dbldot_n=Phi_dbldot(:,index1-1);

    F_n=F;
    F=zeros(3*nodes,1);

    if mass_position(index1)>max(x_vect)
        mass_position(index1)=max(x_vect);

```

```

end
b=(mass_position(index1)-x_vect<=0)-(mass_position(index1)>x_vect)~-1;
[c loc]=max(b);
element=loc-1;
l=x_vect(element+1)-x_vect(element);
iso_par_loc=(mass_position(index1)-x_vect(element))/(x_vect(element+1)-x_vect(element));
N=[(9*(1-l*iso_par_loc)*(1/3-l*iso_par_loc)*((2*1)/3-l*iso_par_loc))/(2*1^3),(27*1*iso_par_loc*(1-
1*iso_par_loc)*((2*1)/3-l*iso_par_loc))/(2*1^3),-(27*1*iso_par_loc*(1-l*iso_par_loc)*...
(1/3-l*iso_par_loc))/(2*1^3),(9*1*iso_par_loc*(1/3-l*iso_par_loc)*((2*1)/3-l*iso_par_loc))/(2*1^3)];
s=sin(element_angle(element));
c=cos(element_angle(element));
T_unit=[c s 0;-s c 0;0 0 1];
T=blkdiag(T_unit,T_unit,T_unit,T_unit);
fx_element=mass*g*sin(element_angle(element));
fy_element=mass*g*(-CF*vel^2/R+cos(element_angle(element)));
if time(index1)>L/vel
    fx_element=0;
    fy_element=0;
end
F_element=zeros(12,1);
F_element(1:3:end)=fx_element*N;
F_element(2:3:end)=fy_element*N;
F_element=T'*F_element;

a=(element-1)*9+1;
b=a+11;
F(a:b)=F(a:b)+F_element;
F=F+QF;

F_nplus1=F;
F_temp(:,index1)=F;

B1=Lambda6W1*M+Lambda5W2*delta_t*C+Lambda3W3*delta_t^2*K;
B2=-M*Phi_dbl_dot_n-C*(Phi_dot_n+Lambda4W1*delta_t*Phi_dbl_dot_n)...
-K*(Phi_n+Lambda1W1*delta_t*Phi_dot_n+Lambda2W2*delta_t^2*Phi_dbl_dot_n)+...
F_n+W1*(F_nplus1-F_n);

B1(1,:)=zeros(nodes*3,1);
B1(:,1)=zeros(1,nodes*3);
B1(1,1)=1;
B1(2,:)=zeros(nodes*3,1);
B1(:,2)=zeros(1,nodes*3);
B1(2,2)=1;
B1(end-2,:)=zeros(nodes*3,1);
B1(:,end-2)=zeros(1,nodes*3);
B1(end-2,end-2)=1;
B1(end-1,:)=zeros(nodes*3,1);
B1(:,end-1)=zeros(1,nodes*3);
B1(end-1,end-1)=1;
B2(1)=0;
B2(2)=0;
B2(end-2)=0;
B2(end-1)=0;

```

```

Delta_Phi_dbldot=B1\B2;

Phi_dbldot(:,index1)=Phi_dbldot_n+Delta_Phi_dbldot;
Phi_dot(:,index1)=Phi_dot_n+lambda4*Phi_dbldot_n*delta_t+lambda5*Delta_Phi_dbldot*delta_t;

Phi(:,index1)=Phi_n+lambda1*Phi_dot_n*delta_t+lambda2*Phi_dbldot_n*delta_t^2+lambda3*Delta_Phi_dbldot*delta_t^2;

u(:,index1)=Phi(1:3:end,index1);
v(:,index1)=Phi(2:3:end,index1);
theta(:,index1)=Phi(3:3:end,index1);

u_14(:,index1)=interp1(x_vect',u(1:3:end,index1),0.25*L);
v_14(:,index1)=interp1(x_vect',v(1:3:end,index1),0.25*L);
theta_14(:,index1)=interp1(x_vect',theta(1:3:end,index1),0.25*L);

u_12(:,index1)=interp1(x_vect',u(1:3:end,index1),0.5*L);
v_12(:,index1)=interp1(x_vect',v(1:3:end,index1),0.5*L);
theta_12(:,index1)=interp1(x_vect',theta(1:3:end,index1),0.5*L);

u_34(:,index1)=interp1(x_vect',u(1:3:end,index1),0.75*L);
v_34(:,index1)=interp1(x_vect',v(1:3:end,index1),0.75*L);
theta_34(:,index1)=interp1(x_vect',theta(1:3:end,index1),0.75*L);

u_dot(:,index1)=Phi_dot(1:3:end,index1);
v_dot(:,index1)=Phi_dot(2:3:end,index1);
theta_dot(:,index1)=Phi_dot(3:3:end,index1);

u_dot_14(:,index1)=interp1(x_vect',u_dot(1:3:end,index1),0.25*L);
v_dot_14(:,index1)=interp1(x_vect',v_dot(1:3:end,index1),0.25*L);
theta_dot_14(:,index1)=interp1(x_vect',theta_dot(1:3:end,index1),0.25*L);

u_dot_12(:,index1)=interp1(x_vect',u_dot(1:3:end,index1),0.5*L);
v_dot_12(:,index1)=interp1(x_vect',v_dot(1:3:end,index1),0.5*L);
theta_dot_12(:,index1)=interp1(x_vect',theta_dot(1:3:end,index1),0.5*L);

u_dot_34(:,index1)=interp1(x_vect',u_dot(1:3:end,index1),0.75*L);
v_dot_34(:,index1)=interp1(x_vect',v_dot(1:3:end,index1),0.75*L);
theta_dot_34(:,index1)=interp1(x_vect',theta_dot(1:3:end,index1),0.75*L);

u_ddot(:,index1)=Phi_dbldot(1:3:end,index1);
v_ddot(:,index1)=Phi_dbldot(2:3:end,index1);
theta_ddot(:,index1)=Phi_dbldot(3:3:end,index1);

u_ddot_14(:,index1)=interp1(x_vect',u_ddot(1:3:end,index1),0.25*L);
v_ddot_14(:,index1)=interp1(x_vect',v_ddot(1:3:end,index1),0.25*L);
theta_ddot_14(:,index1)=interp1(x_vect',theta_ddot(1:3:end,index1),0.25*L);

u_ddot_12(:,index1)=interp1(x_vect',u_ddot(1:3:end,index1),0.5*L);
v_ddot_12(:,index1)=interp1(x_vect',v_ddot(1:3:end,index1),0.5*L);
theta_ddot_12(:,index1)=interp1(x_vect',theta_ddot(1:3:end,index1),0.5*L);

u_ddot_34(:,index1)=interp1(x_vect',u_ddot(1:3:end,index1),0.75*L);
v_ddot_34(:,index1)=interp1(x_vect',v_ddot(1:3:end,index1),0.75*L);

```

```

theta_ddot_34(:,index1)=interp1(x_vect',theta_ddot(1:3:end,index1),0.75*L);

Gx(:,index1)=u(:,index1)+Global_x';
Gy(:,index1)=v(:,index1)+Global_y';

toc_temp=toc;
total_calcs=total_calcs-1;
iter_time=(iter_time*(index1-1)+toc_temp)/index1;
end
close(h)

X1=time;
YMatrix1=[u_14; u_12; u_34];
YMatrix2=[v_14; v_12; v_34];
YMatrix3=[theta_14; theta_12; theta_34];

YMatrix4=[u_dot_14; u_dot_12; u_dot_34];
YMatrix5=[v_dot_14; v_dot_12; v_dot_34];
YMatrix6=[theta_dot_14; theta_dot_12; theta_dot_34];

YMatrix7=[u_ddot_14; u_ddot_12; u_ddot_34];
YMatrix8=[v_ddot_14; v_ddot_12; v_ddot_34];
YMatrix9=[theta_ddot_14; theta_ddot_12; theta_ddot_34];

% Create figure
figure1 = figure('Color',[1 1 1]);

% Create multiple lines using matrix input to plot
subplot1=subplot(3,3,1);
plot1 = plot(X1,YMatrix1,'LineWidth',2);
xlim([0 t_f]);
set(plot1(1),'LineStyle','--','DisplayName','x = 1/4 L');
set(plot1(2),'Color',[1 0 0],'DisplayName','x = 1/2 L');
set(plot1(3),'LineStyle','-','DisplayName','x = 3/4 L','Color',[0 0 0]);

% Create xlabel
xlabel('$Time (s)$','interpreter','latex');

% Create ylabel
ylabel('$u (m)$','interpreter','latex','FontName','cmr12');

% Create title
title('$u(t)$','interpreter','latex','FontSize',14);

% Create multiple lines using matrix input to plot
subplot(3,3,2)
plot1 = plot(X1,YMatrix2,'LineWidth',2);
xlim([0 t_f]);
set(plot1(1),'LineStyle','--','DisplayName','x = 1/4 L');
set(plot1(2),'Color',[1 0 0],'DisplayName','x = 1/2 L');
set(plot1(3),'LineStyle','-','DisplayName','x = 3/4 L','Color',[0 0 0]);

% Create xlabel
xlabel('$Time (s)$','interpreter','latex');

% Create ylabel
ylabel('$w (m)$','interpreter','latex','FontName','cmr12');

% Create title
title('$w(t)$','interpreter','latex','FontSize',14);

% Create multiple lines using matrix input to plot
subplot(3,3,3)

```

```

plot1 = plot(X1, YMatrix3, 'LineWidth', 2);
xlim([0 t_f]);
set(plot1(1), 'LineStyle', '--', 'DisplayName', 'x = 1/4 L');
set(plot1(2), 'Color', [1 0 0], 'DisplayName', 'x = 1/2 L');
set(plot1(3), 'LineStyle', '-.', 'DisplayName', 'x = 3/4 L', 'Color', [0 0 0]);

% Create xlabel
xlabel('$T_{time}$ (s)', 'interpreter', 'latex');

% Create ylabel
ylabel('$\theta$ (m)', 'interpreter', 'latex', 'FontName', 'cmr12');

% Create title
title('$\theta(t)$', 'interpreter', 'latex', 'FontSize', 14);

% Create multiple lines using matrix input to plot
subplot(3,3,4)
plot1 = plot(X1, YMatrix4, 'LineWidth', 2);
xlim([0 t_f]);
set(plot1(1), 'LineStyle', '--', 'DisplayName', 'x = 1/4 L');
set(plot1(2), 'Color', [1 0 0], 'DisplayName', 'x = 1/2 L');
set(plot1(3), 'LineStyle', '-.', 'DisplayName', 'x = 3/4 L', 'Color', [0 0 0]);

% Create xlabel
xlabel('$T_{time}$ (s)', 'interpreter', 'latex');

% Create ylabel
ylabel('$\dot{u}$ (m)', 'interpreter', 'latex', 'FontName', 'cmr12');

% Create title
title('$\dot{u}(t)$', 'interpreter', 'latex', 'FontSize', 14);

% Create multiple lines using matrix input to plot
subplot(3,3,5)
plot1 = plot(X1, YMatrix5, 'LineWidth', 2);
xlim([0 t_f]);
set(plot1(1), 'LineStyle', '--', 'DisplayName', 'x = 1/4 L');
set(plot1(2), 'Color', [1 0 0], 'DisplayName', 'x = 1/2 L');
set(plot1(3), 'LineStyle', '-.', 'DisplayName', 'x = 3/4 L', 'Color', [0 0 0]);

% Create xlabel
xlabel('$T_{time}$ (s)', 'interpreter', 'latex');

% Create ylabel
ylabel('$\dot{w}$ (m)', 'interpreter', 'latex', 'FontName', 'cmr12');

% Create title
title('$\dot{w}(t)$', 'interpreter', 'latex', 'FontSize', 14);

% Create multiple lines using matrix input to plot
subplot(3,3,6)
plot1 = plot(X1, YMatrix6, 'LineWidth', 2);
xlim([0 t_f]);
set(plot1(1), 'LineStyle', '--', 'DisplayName', 'x = 1/4 L');
set(plot1(2), 'Color', [1 0 0], 'DisplayName', 'x = 1/2 L');
set(plot1(3), 'LineStyle', '-.', 'DisplayName', 'x = 3/4 L', 'Color', [0 0 0]);
% Create xlabel
xlabel('$T_{time}$ (s)', 'interpreter', 'latex');

% Create ylabel
ylabel('$\dot{\theta}$ (m)', 'interpreter', 'latex', 'FontName', 'cmr12');

% Create title
title('$\dot{\theta}(t)$', 'interpreter', 'latex', 'FontSize', 14);

% Create multiple lines using matrix input to plot
subplot(3,3,7)

```

```

plot1 = plot(X1, YMatrix7, 'LineWidth', 2);
xlim([0 t_f]);
set(plot1(1), 'LineStyle', '--', 'DisplayName', 'x = 1/4 L');
set(plot1(2), 'Color', [1 0 0], 'DisplayName', 'x = 1/2 L');
set(plot1(3), 'LineStyle', '-.', 'DisplayName', 'x = 3/4 L', 'Color', [0 0 0]);

% Create xlabel
xlabel('$T$ (s)', 'interpreter', 'latex');

% Create ylabel
ylabel('$\ddot{u}$ (m)', 'interpreter', 'latex', 'FontName', 'cmr12');

% Create title
title('$\ddot{u}(t)$', 'interpreter', 'latex', 'FontSize', 14);

% Create multiple lines using matrix input to plot
subplot(3,3,8)
plot1 = plot(X1, YMatrix8, 'LineWidth', 2);
xlim([0 t_f]);
set(plot1(1), 'LineStyle', '--', 'DisplayName', 'x = 1/4 L');
set(plot1(2), 'Color', [1 0 0], 'DisplayName', 'x = 1/2 L');
set(plot1(3), 'LineStyle', '-.', 'DisplayName', 'x = 3/4 L', 'Color', [0 0 0]);

% Create xlabel
xlabel('$T$ (s)', 'interpreter', 'latex');

% Create ylabel
ylabel('$\ddot{w}$ (m)', 'interpreter', 'latex', 'FontName', 'cmr12');

% Create title
title('$\ddot{w}(t)$', 'interpreter', 'latex', 'FontSize', 14);

% Create multiple lines using matrix input to plot
subplot(3,3,9)
plot1 = plot(X1, YMatrix9, 'LineWidth', 2);
xlim([0 t_f]);
set(plot1(1), 'LineStyle', '--', 'DisplayName', 'x = 1/4 L');
set(plot1(2), 'Color', [1 0 0], 'DisplayName', 'x = 1/2 L');
set(plot1(3), 'LineStyle', '-.', 'DisplayName', 'x = 3/4 L', 'Color', [0 0 0]);

% Create xlabel
xlabel('$T$ (s)', 'interpreter', 'latex');

% Create ylabel
ylabel('$\ddot{\theta}$ (m)', 'interpreter', 'latex', 'FontName', 'cmr12');

% Create title
title('$\ddot{\theta}(t)$', 'interpreter', 'latex', 'FontSize', 14);

legend1 = legend(subplot1, 'show');
set(legend1, 'EdgeColor', [1 1 1], 'Orientation', 'horizontal', 'YColor', [1 1 1], ...
'XColor', [1 1 1], ...
'Position', [0.39685029522514 0.0392584514721921 0.241542465105276 0.0283533260632497], ...
'FontSize', 12);

```

```

%Plate_moving_mass_thermal_strain.m

clc
clear
close all

Length=2;
Width=2;
Height=0.01;
h=Height;
rho=2700;

E=72e9;
pr=0.33;
k=pi/sqrt(12);
I=h^3/12;
Alpha=2.3e-7;

%moving mass
mass=2.5e2/9.81;
g=-9.81;
mass_thickness=0.25*Width;
MT=mass_thickness;
p=mass*g/MT;
vel=5;

T_top=20;
T_bottom=60;

elements_per_edge_x=20;
EPEx=elements_per_edge_x;
elements_per_edge_y=10;
EPEy=elements_per_edge_y;

Timesteps=1000;
t_f=2*Length/vel;
delta_t=t_f/Timesteps;
time=0:delta_t:t_f;

%GSSS-2
P=[1 1 0];

%Proportional Damping Coefs
xi1=0.05;
xi2=0.05;
Omega1=78;
Omega2=197;
alpha=2*Omega1*Omega2*(xi1*Omega2-xi2*Omega1)/(Omega2^2-Omega1^2);
beta=2*(xi2*Omega2-xi1*Omega1)/(Omega2^2-Omega1^2);

sf_order=1;

[mesh,connectivity,no_elements,no_nodes]=plate_mesh(Length,0.5*Width,EPEx,EPEy,sf_order);
[meshx,meshy]=meshgrid(linspace(0,Length,EPEx),linspace(0,0.5*Width,EPEy));

```

```

time_steps=0;
calcs=1;
wb_index=0;
total_calcs=no_elements;
iter_time=0;

K=zeros(3*no_nodes,3*no_nodes);
M=zeros(3*no_nodes,3*no_nodes);
F=zeros(3*no_nodes,length(time));
QF=zeros(3*no_nodes,1);

wb=waitbar(0,'Calculating');
for index1=1:no_elements
tic;
est_time=total_calcs*iter_time/60;
est_time_min=floor(est_time);
est_time_sec=(est_time-est_time_min)*60;
waitbar(calcs/no_elements,wb,sprintf('Assembling Stiffness Matrix. %3.0f:%02.0f remaining.',...
[est_time_min est_time_sec]))
clear K_temp F_temp
K_temp=zeros(3*no_nodes,3*no_nodes);
M_temp=zeros(3*no_nodes,3*no_nodes);
QF_temp=zeros(3*no_nodes,1);
Element_length=max(mesh(connectivity(index1,2:end),2))-min(mesh(connectivity(index1,2:end),2));
Element_width=max(mesh(connectivity(index1,2:end),3))-min(mesh(connectivity(index1,2:end),3));
A=Element_length*Element_width;

M_element=-Alpha*(T_top-T_bottom)*E*h^2/12;

X=mesh(connectivity(index1,2:end),2);
Y=mesh(connectivity(index1,2:end),3);

D=E*h^3/(12*(1-pr^2))*[1 pr 0;pr 1 0;0 0 0.5*(1-pr)];
G=k*E*h/(2*(1+pr))*[1 0;0 1];

gauss_loc=[0 sqrt(1/3)];
gauss_weights=[2 1];

DOF=12;

k_mat=zeros(DOF);
m_mat=zeros(DOF);
m_mat_new=zeros(DOF);
qf_vec=zeros(DOF,1);

for index2=1:4;
clear N dNdx1 dNdx2
Bb=zeros(3,DOF);
Bs=zeros(2,DOF);

gauss_pts=gauss_loc(2).*[-1 1 1 -1;-1 -1 1 1];
xi=gauss_pts(1,index2);
eta=gauss_pts(2,index2);

```

```

N=[((eta - 1)*(xi - 1))/4,...
  -((eta - 1)*(xi + 1))/4,...
  ((eta + 1)*(xi + 1))/4,...
  -((eta + 1)*(xi - 1))/4];

dNdx=[eta/4 - 1/4,...
  1/4 - eta/4,...
  eta/4 + 1/4,...
  - eta/4 - 1/4];

dNdeta=[xi/4 - 1/4,...
  - xi/4 - 1/4,...
  xi/4 + 1/4,...
  1/4 - xi/4];

J=[X Y]*[dNdx;dNdeta]';
detJ=det(J);
invJ=inv(J);

temp=invJ*[dNdx;dNdeta];
dNdx=temp(1,:);
dNdeta=temp(2,:);

Bb(1,9:12)=dNdx;
Bb(2,5:8)=-dNdeta;
Bb(3,9:12)=dNdeta;
Bb(3,5:8)=-dNdx;

k_mat=k_mat+Bb'*D*Bb*detJ*gauss_weights(2)*gauss_weights(2);

qf_vec(5:8)=qf_vec(5:8)+1/(1-pr)*M_element*dNdeta'*detJ*gauss_weights(2)*gauss_weights(2);
qf_vec(9:12)=qf_vec(9:12)-1/(1-pr)*M_element*dNdx'*detJ*gauss_weights(2)*gauss_weights(2);

m_mat_new(1:4,1:4)=N'*N*rho*h*detJ*gauss_weights(2)*gauss_weights(2);
m_mat_new(5:8,5:8)=N'*N*rho*I*detJ*gauss_weights(2)*gauss_weights(2);
m_mat_new(9:12,9:12)=N'*N*rho*I*detJ*gauss_weights(2)*gauss_weights(2);
m_mat=m_mat+m_mat_new;
end
clear index2

for index2=1:1;
  clear N dNdx dNdeta
  Bb=zeros(3,DOF);
  Bs=zeros(2,DOF);

  gauss_pts=gauss_loc(1).*[0;0];
  xi=gauss_pts(1,index2);
  eta=gauss_pts(2,index2);

  N=[((eta - 1)*(xi - 1))/4,...
    -((eta - 1)*(xi + 1))/4,...
    ((eta + 1)*(xi + 1))/4,...
    -((eta + 1)*(xi - 1))/4];

```

```

    -((eta + 1)*(xi - 1))/4];

dNdx=[eta/4 - 1/4,...
      1/4 - eta/4,...
      eta/4 + 1/4,...
      - eta/4 - 1/4];

dNdeta=[xi/4 - 1/4,...
        - xi/4 - 1/4,...
        xi/4 + 1/4,...
        1/4 - xi/4];

J=[X Y]'*[dNdx;dNdeta]';
detJ=det(J);
invJ=inv(J);

temp=invJ*[dNdx;dNdeta];
dNdx=temp(1,:);
dNdeta=temp(2,:);

Bs(1,1:4)=dNdx;
Bs(1,9:12)=N;
Bs(2,1:4)=dNdeta;
Bs(2,5:8)=-N;

k_mat=k_mat+B*s*G*B*s*detJ*gauss_weights(1)*gauss_weights(1);
end

K_mat{index1}=k_mat;
temp3=[];
temp3=[connectivity(index1,2:end) no_nodes+connectivity(index1,2:end)
2*no_nodes+connectivity(index1,2:end)];
K_temp(temp3,temp3)=k_mat;
M_temp(temp3,temp3)=m_mat;
QF_temp(temp3,1)=qf_vec;
K=K+K_temp;
M=M+M_temp;
QF=QF+QF_temp;
toc_temp=toc;
total_calcs=total_calcs-1;
calcs=calcs+1;
iter_time=(iter_time*(calcs-1)+toc_temp)/calcs;
end
close(wb)
C=alpha*M+beta*K;

%find edge nodes
edge_nodes=mesh(find(mesh(:,2))==0 | mesh(:,2)==Length | mesh(:,3)==0.5*Width));
zero_force_nodes=[edge_nodes];

%find "hard" SS edges
clear edge_nodes
edge_nodes=mesh(find(mesh(:,2))==0 | mesh(:,2)==Length));
edge_nodes=edge_nodes+no_nodes;

```

```

zero_force_nodes=[zero_force_nodes; edge_nodes];

clear edge_nodes
edge_nodes=mesh(find(mesh(:,3)==0.5*Width));
edge_nodes=edge_nodes+2*no_nodes;
zero_force_nodes=[zero_force_nodes; edge_nodes];

%find symmetry nodes
clear edge_nodes
edge_nodes=mesh(find(mesh(:,3)==0));
mid_nodes=edge_nodes;
edge_nodes=edge_nodes+no_nodes;
zero_force_nodes=[zero_force_nodes; edge_nodes];

zero_force_nodes=sort(unique(zero_force_nodes));
no_zfn=length(zero_force_nodes);

time_steps=0;
calcs=1;
wb_index=0;
total_calcs=length(time)-1;
iter_time=0;

clear temp
y_pos=linspace(0,MT/2,EPEy+1);
wb=waitbar(0,'Calculating');
index3=0;
for index1=1:(length(time)/2)-1;
    tic;
    est_time=total_calcs*iter_time/60;
    est_time_min=floor(est_time);
    est_time_sec=(est_time-est_time_min)*60;
    waitbar(calcs/length(time),wb,sprintf('Assembling Force Vector. %3.0f:%02.0f remaining.',...
        [est_time_min est_time_sec]))
    fe=[];
    cross=[];
    part=[];
    x_pos=vel*time(index1);
    if x_pos>Length
        x_pos=Length;
    end
    for index2=1:no_elements;
        testx=[];
        testy=[];
        N1=connectivity(index2,2);
        N2=connectivity(index2,3);
        N3=connectivity(index2,4);
        N4=connectivity(index2,5);
        temp=unique([mesh(N1,2),mesh(N2,2),mesh(N3,2),mesh(N4,2)]);
        Ex1=min(temp);
        Ex2=max(temp);
        temp=unique([mesh(N1,3),mesh(N2,3),mesh(N3,3),mesh(N4,3)]);
        Ey1=min(temp);
        Ey2=max(temp);
    end
end

```

```

testx = Ex1<=x_pos & Ex2>x_pos;
testy=Ey2>=y_pos & max(y_pos)>Ey1;
if testx==1 & sum(testy)>1 & mean(testy)~=1
    cross=[cross index2];
elseif testx==1 & mean(testy)==1;
    part=index2;
end
end

for index2=1:length(cross);
    F_temp=zeros(3*no_nodes,1);
    ei=cross(index2);
    N1=connectivity(ei,2);
    N2=connectivity(ei,3);
    N3=connectivity(ei,4);
    N4=connectivity(ei,5);
    temp=unique([mesh(N1,2),mesh(N2,2),mesh(N3,2),mesh(N4,2)]);
    Ex1=min(temp);
    Ex2=max(temp);
    temp=unique([mesh(N1,3),mesh(N2,3),mesh(N3,3),mesh(N4,3)]);
    Ey1=min(temp);
    Ey2=max(temp);
    dx=Ex2-Ex1;
    dy=Ey2-Ey1;
    dA=dx*dy;
    xi=2*(x_pos-Ex1)/dx-1;
    eta=0;
    N=[((eta - 1)*(xi - 1))/4,...
        -((eta - 1)*(xi + 1))/4,...
        ((eta + 1)*(xi + 1))/4,...
        -((eta + 1)*(xi - 1))/4];
    f_vec=p*dy*N';
    temp2=connectivity(ei,2:5);
    F_temp(temp2,:)=f_vec;
    F(:,index1)=F(:,index1)+F_temp;
end
F_temp=zeros(3*no_nodes,1);
N1=connectivity(part,2);
N2=connectivity(part,3);
N3=connectivity(part,4);
N4=connectivity(part,5);
temp=unique([mesh(N1,2),mesh(N2,2),mesh(N3,2),mesh(N4,2)]);
Ex1=min(temp);
Ex2=max(temp);
temp=unique([mesh(N1,3),mesh(N2,3),mesh(N3,3),mesh(N4,3)]);
Ey1=min(temp);
Ey2=max(temp);
dx=Ex2-Ex1;
dy=Ey2-Ey1;
dA=dx*dy;
xi=2*(x_pos-Ex1)/dx-1;
eta=2*(0.5*MT-Ey1)/dy-1;
N=[((eta - 1)*(xi - 1))/4,...
    -((eta - 1)*(xi + 1))/4,...

```

```

        ((eta + 1)*(xi + 1))/4,...
        -((eta + 1)*(xi - 1))/4];
    f_vec=p*(0.5*MT-Ey1)*N';
    temp2=connectivity(part,2:5);
    F_temp(temp2,:)=f_vec;
    F(:,index1)=F(:,index1)+F_temp+QF;
    toc_temp=toc;
    total_calcs=total_calcs-1;
    calcs=calcs+1;
    iter_time=(iter_time*(calcs-1)+toc_temp)/calcs;
    index3=index3+1;
end
for index1=index3+1:length(time)
    F(:,index1)=F(:,index1)+QF;
end
close(wb)

clear F_temp K_temp
F_temp1=F(:,1);
K_temp1=K;

K_temp1(zero_force_nodes,:)=zeros(no_zfn,3*no_nodes);
K_temp1(:,zero_force_nodes)=zeros(3*no_nodes,no_zfn);
for index2=1:no_zfn
    K_temp1(zero_force_nodes(index2),zero_force_nodes(index2))=1;
end
F_temp1(zero_force_nodes,1)=0;

Phi=K_temp1\F_temp1;
Phi_dot=zeros(3*no_nodes,1);
Phi_dbldot=zeros(3*no_nodes,1);
w(:,1)=Phi(1:no_nodes,1);

time_steps=0;
calcs=1;
wb_index=0;
total_calcs=length(time)-1;
iter_time=0;
wb=waitbar(0,'Calculating');
for index1=2:length(time)
    tic;
    est_time=total_calcs*iter_time/60;
    est_time_min=floor(est_time);
    est_time_sec=(est_time-est_time_min)*60;
    waitbar(calcs/length(time),wb,sprintf('Solving. %3.0f:%02.0f remaining.',...
        [est_time_min est_time_sec]))
    pmin=P(1);
    pmax=P(2);
    ps=P(3);
    Lambda1W1=(3+pmin+pmax-pmin*pmax)/(2*(1+pmin)*(1+pmax));
    Lambda2W2=1/((1+pmin)*(1+pmax));
    Lambda3W3=1/((1+pmin)*(1+pmax)*(1+ps));
    Lambda4W1=(3+pmin+pmax-pmin*pmax)/(2*(1+pmin)*(1+pmax));
    Lambda5W2=2/((1+pmin)*(1+pmax)*(1+ps));

```

```

Lambda6W1=(2+pmin+pmax+ps-pmin*pmax*ps)/((1+pmin)*(1+pmax)*(1+ps));
W1=(3+pmin+pmax-pmin*pmax)/(2*(1+pmin)*(1+pmax));
lambda1=1;
lambda2=0.5;
lambda3=1/(2*(1+ps));
lambda4=1;
lambda5=1/(1+ps);
clear B1 B2 Delta_Phi_dbldot Phi_n Phi_dot_n Phi_dbldot_n
Phi_n=Phi(:,index1-1);
Phi_dot_n=Phi_dot(:,index1-1);
Phi_dbldot_n=Phi_dbldot(:,index1-1);
F_n=F(:,index1-1);
F_nplus1=F(:,index1);
B1=Lambda6W1*M+Lambda5W2*delta_t*C+Lambda3W3*delta_t^2*K;
B2=-M*Phi_dbldot_n-C*(Phi_dot_n+Lambda4W1*delta_t*Phi_dbldot_n)...
-K*(Phi_n+Lambda1W1*delta_t*Phi_dot_n+Lambda2W2*delta_t^2*Phi_dbldot_n)+...
F_n+W1*(F_nplus1-F_n);

B1(zero_force_nodes,:)=zeros(no_zfn,3*no_nodes);
B1(:,zero_force_nodes)=zeros(3*no_nodes,no_zfn);
for index2=1:no_zfn
    B1(zero_force_nodes(index2),zero_force_nodes(index2))=1;
end
B2(zero_force_nodes,1)=0;

Delta_Phi_dbldot=B1\B2;

Phi_dbldot(:,index1)=Phi_dbldot_n+Delta_Phi_dbldot;
Phi_dot(:,index1)=Phi_dot_n+lambda4*Phi_dbldot_n*delta_t+lambda5*Delta_Phi_dbldot*delta_t;

Phi(:,index1)=Phi_n+lambda1*Phi_dot_n*delta_t+lambda2*Phi_dbldot_n*delta_t^2+lambda3*Delta_Phi_
dbldot*delta_t^2;

w(:,index1)=Phi(1:no_nodes,index1);
thetax(:,index1)=Phi(no_nodes+1:2*no_nodes,index1);
thetay(:,index1)=Phi(2*no_nodes+1:3*no_nodes,index1);

w_dot(:,index1)=Phi_dot(1:no_nodes,index1);
thetax_dot(:,index1)=Phi_dot(no_nodes+1:2*no_nodes,index1);
thetay_dot(:,index1)=Phi_dot(2*no_nodes+1:3*no_nodes,index1);

w_dbldot(:,index1)=Phi_dbldot(1:no_nodes,index1);
thetax_dbldot(:,index1)=Phi_dbldot(no_nodes+1:2*no_nodes,index1);
thetay_dbldot(:,index1)=Phi_dbldot(2*no_nodes+1:3*no_nodes,index1);

w_14(:,index1)=interp1(mesh(1:EPEX,2),w(1:EPEX,index1),0.25*Length);
thetay_14(:,index1)=interp1(mesh(1:EPEX,2),thetay(1:EPEX,index1),0.25*Length);
w_dot_14(:,index1)=interp1(mesh(1:EPEX,2),w_dot(1:EPEX,index1),0.25*Length);
thetay_dot_14(:,index1)=interp1(mesh(1:EPEX,2),thetay_dot(1:EPEX,index1),0.25*Length);
w_ddot_14(:,index1)=interp1(mesh(1:EPEX,2),w_dbldot(1:EPEX,index1),0.25*Length);
thetay_ddot_14(:,index1)=interp1(mesh(1:EPEX,2),thetay_dbldot(1:EPEX,index1),0.25*Length);

w_12(:,index1)=interp1(mesh(1:EPEX,2),w(1:EPEX,index1),0.5*Length);
thetay_12(:,index1)=interp1(mesh(1:EPEX,2),thetay(1:EPEX,index1),0.5*Length);

```

```

w_dot_12(:,index1)=interp1(mesh(1:EPEX,2),w_dot(1:EPEX,index1),0.5*Length);
thetay_dot_12(:,index1)=interp1(mesh(1:EPEX,2),thetay_dot(1:EPEX,index1),0.5*Length);
w_ddot_12(:,index1)=interp1(mesh(1:EPEX,2),w_dbldot(1:EPEX,index1),0.5*Length);
thetay_ddot_12(:,index1)=interp1(mesh(1:EPEX,2),thetay_dbldot(1:EPEX,index1),0.5*Length);

w_34(:,index1)=interp1(mesh(1:EPEX,2),w(1:EPEX,index1),0.75*Length);
thetay_34(:,index1)=interp1(mesh(1:EPEX,2),thetay(1:EPEX,index1),0.75*Length);
w_dot_34(:,index1)=interp1(mesh(1:EPEX,2),w_dot(1:EPEX,index1),0.75*Length);
thetay_dot_34(:,index1)=interp1(mesh(1:EPEX,2),thetay_dot(1:EPEX,index1),0.75*Length);
w_ddot_34(:,index1)=interp1(mesh(1:EPEX,2),w_dbldot(1:EPEX,index1),0.75*Length);
thetay_ddot_34(:,index1)=interp1(mesh(1:EPEX,2),thetay_dbldot(1:EPEX,index1),0.75*Length);

Energy(index1)=0.5*transpose(Phi(:,index1))*K*Phi(:,index1);

toc_temp=toc;
total_calcs=total_calcs-1;
calcs=calcs+1;
iter_time=(iter_time*(calcs-1)+toc_temp)/calcs;
end
close(wb)

X1=time;
YMatrix1=[w_14; w_12; w_34];
YMatrix2=[thetay_14; thetay_12; thetay_34];

YMatrix3=[w_dot_14; w_dot_12; w_dot_34];
YMatrix4=[thetay_dot_14; thetay_dot_12; thetay_dot_34];

YMatrix5=[w_ddot_14; w_ddot_12; w_ddot_34];
YMatrix6=[thetay_ddot_14; thetay_ddot_12; thetay_ddot_34];

% Create figure
figure1 = figure('Color',[1 1 1]);
set (figure1, 'Units', 'normalized', 'Position', [0,0,0.5,1]);
set(figure1,'PaperSize',[8.5 11]);
set(figure1,'PaperPositionMode','auto');

% Create multiple lines using matrix input to plot
subplot1=subplot(3,2,1);
plot1 = plot(X1,YMatrix1,'LineWidth',2);
xlim([0 t_f]);
set(plot1(1),'LineStyle','--','DisplayName','x = 1/4 L');
set(plot1(2),'Color',[1 0 0],'DisplayName','x = 1/2 L');
set(plot1(3),'LineStyle','-','DisplayName','x = 3/4 L','Color',[0 0 0]);

% Create xlabel
xlabel('$Time (s)$','interpreter','latex');

% Create ylabel
ylabel('$w (m)$','interpreter','latex','FontName','cmr12');

% Create title
title('a) $ w(t)$','interpreter','latex','FontSize',14);

```

```

% Create multiple lines using matrix input to plot
subplot(3,2,2)
plot1 = plot(X1,YMatrix2,'LineWidth',2);
xlim([0 t_f]);
set(plot1(1),'LineStyle','--','DisplayName','x = 1/4 L');
set(plot1(2),'Color',[1 0 0],'DisplayName','x = 1/2 L');
set(plot1(3),'LineStyle','-','DisplayName','x = 3/4 L','Color',[0 0 0]);

% Create xlabel
xlabel('$Time (s)$','interpreter','latex');

% Create ylabel
ylabel('$\theta_y (rad)$','interpreter','latex','FontName','cmr12');

% Create title
title('b) $ \theta_y(t)$','interpreter','latex','FontSize',14);

% Create multiple lines using matrix input to plot
subplot(3,2,3)
plot1 = plot(X1,YMatrix3,'LineWidth',2);
xlim([0 t_f]);
set(plot1(1),'LineStyle','--','DisplayName','x = 1/4 L');
set(plot1(2),'Color',[1 0 0],'DisplayName','x = 1/2 L');
set(plot1(3),'LineStyle','-','DisplayName','x = 3/4 L','Color',[0 0 0]);

% Create xlabel
xlabel('$Time (s)$','interpreter','latex');

% Create ylabel
ylabel('$\dot{w} (\frac{m}{s})$','interpreter','latex','FontName','cmr12');

% Create title
title('c) $ \dot{w}(t)$','interpreter','latex','FontSize',14);

% Create multiple lines using matrix input to plot
subplot(3,2,4)
plot1 = plot(X1,YMatrix4,'LineWidth',2);
xlim([0 t_f]);
set(plot1(1),'LineStyle','--','DisplayName','x = 1/4 L');
set(plot1(2),'Color',[1 0 0],'DisplayName','x = 1/2 L');
set(plot1(3),'LineStyle','-','DisplayName','x = 3/4 L','Color',[0 0 0]);

% Create xlabel
xlabel('$Time (s)$','interpreter','latex');

% Create ylabel
ylabel('$\dot{\theta}_y (\frac{rad}{s})$','interpreter','latex','FontName','cmr12');

% Create title
title('d) $ \dot{\theta}_y(t)$','interpreter','latex','FontSize',14);

% Create multiple lines using matrix input to plot
subplot(3,2,5)

```

```

plot1 = plot(X1,YMatrix5,'LineWidth',2);
xlim([0 t_f]);
set(plot1(1),'LineStyle','--','DisplayName','x = 1/4 L');
set(plot1(2),'Color',[1 0 0],'DisplayName','x = 1/2 L');
set(plot1(3),'LineStyle','-','DisplayName','x = 3/4 L','Color',[0 0 0]);

% Create xlabel
xlabel('$Time (s)$','interpreter','latex');

% Create ylabel
ylabel('$\ddot{w} (\frac{m}{s^2})$','interpreter','latex','FontName','cmr12');

% Create title
title('e) $ \ddot{w}(t)$','interpreter','latex','FontSize',14);

% Create multiple lines using matrix input to plot
subplot(3,2,6)
plot1 = plot(X1,YMatrix6,'LineWidth',2);
xlim([0 t_f]);
set(plot1(1),'LineStyle','--','DisplayName','x = 1/4 L');
set(plot1(2),'Color',[1 0 0],'DisplayName','x = 1/2 L');
set(plot1(3),'LineStyle','-','DisplayName','x = 3/4 L','Color',[0 0 0]);

% Create xlabel
xlabel('$Time (s)$','interpreter','latex');

% Create ylabel
ylabel('$\ddot{\theta}_y (\frac{rad}{s^2})$','interpreter','latex','FontName','cmr12');

% Create title
title('f) $ \ddot{\theta}_y(t)$','interpreter','latex','FontSize',14);

legend1 = legend(subplot1,'show');
set(legend1,'EdgeColor',[1 1 1],'Orientation','horizontal','YColor',[1 1 1],...
'XColor',[1 1 1],...
'Position',[0.39685029522514 0.0392584514721921 0.241542465105276 0.0283533260632497],...
'FontSize',12);

```

```

%plate_mesh.m

function [mesh,connectivity,no_elements,no_nodes] = plate_mesh( Mesh_x_length, Mesh_y_length,
edge_x_elements, edge_y_elements , order)

Lx=Mesh_x_length;
Ly=Mesh_y_length;
Nx=(order*edge_x_elements)+1;
Ny=(order*edge_y_elements)+1;
Ex=edge_x_elements;
Ey=edge_y_elements;
Dx=linspace(0,Lx,Nx);
Dy=linspace(0,Ly,Ny);

nodes_per_element=(order+1)^2;
NPE=nodes_per_element;

no_elements=Ex*Ey;
no_nodes=Nx*Ny;

[xs,ys]=meshgrid(Dx,Dy);
Nodes=transpose(reshape([1:no_nodes],Nx,Ny));
Elements(:,1)=[1:no_elements];
mesh=sortrows([reshape(Nodes,no_nodes,1) reshape(xs,no_nodes,1) reshape(ys,no_nodes,1)]);

element=1;
for index1=1:Ey
    for index2=1:Ex
        connectivity(element,1)=element;
        origin=[Dx(index2*order-(order-1)) Dy(index1*order-(order-1))];
        dx=Dx(index2*order-(order-1)+order)-origin(1);
        dy=Dy(index1*order-(order-1)+order)-origin(2);
        if order==1;
            x_nodes=origin(1)+[0 dx dx 0];
            y_nodes=origin(2)+[0 0 dy dy];
        elseif order==2
            x_nodes=origin(1)+[0 dx dx 0 0.5*dx dx 0.5*dx 0 0.5*dx];
            y_nodes=origin(2)+[0 0 dy dy 0 0.5*dy dy 0.5*dy 0.5*dy];
        end
        for index3=1:NPE;
            x=x_nodes(index3);
            y=y_nodes(index3);
            [x_loc y_loc]=find((abs(xs-x))<1e-7 & (abs(ys-y))<1e-8);
            connectivity(element,index3+1)=Nodes(x_loc,y_loc);
        end
        element=element+1;
    end
end
end
end

```