

**PURDUE UNIVERSITY  
GRADUATE SCHOOL  
Thesis/Dissertation Acceptance**

This is to certify that the thesis/dissertation prepared

By Gregory Bunting

Entitled

PARALLEL REAL-TIME HYBRID SIMULATION OF STRUCTURES USING MULTI-SCALE MODELS

For the degree of Doctor of Philosophy



Is approved by the final examining committee:

Arun Prakash

Chair

Shirley J. Dyke

Pablo D. Zavattieri

Samuel P. Midkiff

Timothy F. Walsh

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Arun Prakash

Approved by: Dulcy M. Abraham

Head of the Departmental Graduate Program

7/28/2016

Date



PARALLEL REAL-TIME HYBRID SIMULATION OF  
STRUCTURES USING MULTI-SCALE MODELS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Gregory B. Bunting

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2016

Purdue University

West Lafayette, Indiana

ProQuest Number: 10170601

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10170601

Published by ProQuest LLC (2016). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code  
Microform Edition © ProQuest LLC.

ProQuest LLC.  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106 - 1346

## ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Arun Prakash for his direction and support throughout this entire processes. My committee members, Dr. Shirley Dyke, Dr. Timothy Walsh, Dr. Pablo Zavattieri, and Dr. Samuel Midkiff for their advice and suggestions on my research. Special thanks to the other students supporting me throughout my studies, including but not limited to Quang Nguyen, Reiko Akiyama, Payton Lindsay, Matt Raymer, Xiaowo Wang, Hui Liu, Amin Magherah, Gaby Ou, and Pablo Laight. I would also like to thank the SD team for their technical knowledge and support, including Garth Reese, Scott Miller, David Day, Nate Roehrig, and Clark Dohrmann. My master's thesis advisor at Clemson University Dr. Brad Putman for introducing me to research. My friends and family for the emotional support, which kept me going during the toughest times.

This material is based in part upon work supported by the National Science Foundation under Grant Numbers CNS-1136073 and CNS-1136075. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Support for the development of the CIRST instrument was made possible through the NSF MRI Program under Grant CNS-1028668 and the School of Mechanical Engineering at Purdue.

## TABLE OF CONTENTS

	Page
TABLE OF CONTENTS . . . . .	iii
LIST OF TABLES . . . . .	vi
LIST OF FIGURES . . . . .	vii
ABSTRACT . . . . .	xii
1 INTRODUCTION . . . . .	1
1.1 Objectives . . . . .	4
1.1.1 Characterize MTS Error and Performance . . . . .	5
1.1.2 Temporal multi-scale models in RTHS . . . . .	6
1.1.3 Parallel nonlinear spatial multi-scale models in RTHS . . . . .	6
1.1.4 Perfectly Matched Layers with an Ellipsoidal Boundary . . . . .	7
2 CHARACTERIZING ERRORS AND EVALUATING PERFORMANCE OF TRANSIENT SIMULATIONS USING MULTI-TIME-STEP INTEGRATION . . . . .	9
2.1 Introduction . . . . .	9
2.2 Overview of time integration methods . . . . .	12
2.3 Multi-time-step methods . . . . .	15
2.3.1 Governing equations for the multi-time-step method . . . . .	16
2.3.2 Solution procedure . . . . .	18
2.3.3 Computational cost of the multi-time-step method . . . . .	20
2.3.4 Exact and reference solutions for computing errors . . . . .	22
2.3.5 Error measures . . . . .	23
2.4 Characterization of Errors . . . . .	23
2.4.1 MTS decomposition for a truss problem . . . . .	24
2.4.2 MTS decomposition for a frame problem . . . . .	29
2.4.3 MTS decomposition for wave propagation in a solid domain . . . . .	31
2.5 Space of Possible MTS Decompositions . . . . .	33
2.5.1 Optimal MTS decomposition . . . . .	36
2.6 Numerical Examples . . . . .	37
2.6.1 Frame Problem . . . . .	37
2.6.2 Lamb Problem . . . . .	37
2.6.3 Discussion and insights . . . . .	39
2.7 Conclusions . . . . .	41

3	ASYNCHRONOUS MULTI-TIME-STEP COUPLING OF NUMERICAL AND PHYSICAL MODELS FOR HIGH-FIDELITY REAL-TIME HYBRID SIMULATION . . . . .	43
3.1	Introduction . . . . .	44
3.2	Real Time Hybrid Simulation (RTHS) . . . . .	46
3.3	Asynchronous multi-time-step (MTS) coupling of numerical and physical models . . . . .	49
3.3.1	Conventional RTHS . . . . .	49
3.3.2	MTS method for numerical substructure . . . . .	51
3.3.3	Asynchronous predictor for MTS coupling of numerical and physical models . . . . .	55
3.3.4	Computational cost of MTS+RTHS . . . . .	56
3.4	Implementation of asynchronous MTS coupling on CyberMech - a novel RTHS platform . . . . .	57
3.4.1	Taskset . . . . .	58
3.4.2	Shared Memory Model . . . . .	58
3.5	Results . . . . .	58
3.5.1	Virtual Real-time hybrid simulation of a nine-story frame . . . . .	59
3.5.2	Error Distribution with Shear Model . . . . .	64
3.5.3	Error Distribution with No Predictor . . . . .	65
3.5.4	Analysis of selected predictor schemes . . . . .	68
3.6	Discussion . . . . .	71
3.7	Conclusions . . . . .	73
4	HIERARHICAL MULTI-SCALE MODELS FOR PARALLEL REAL-TIME HYBRID SIMULATION . . . . .	74
4.1	Introduction . . . . .	74
4.2	Cybermech Computational Platform . . . . .	78
4.3	Formulation of Multi-Scale Models for RTHS . . . . .	79
4.3.1	Nonlinear Explicit MTS . . . . .	79
4.3.2	Equations of Motion . . . . .	80
4.3.3	Coupling Matrices . . . . .	82
4.3.4	Beam-Continuum Coupling . . . . .	83
4.3.5	J2 Plasticity . . . . .	86
4.4	Verification . . . . .	87
4.4.1	Nonlinear Verification . . . . .	88
4.5	Experimental and Numerical Models . . . . .	89
4.5.1	Physical Structure . . . . .	89
4.5.2	Shear Model . . . . .	90
4.5.3	Simple Beam Model . . . . .	92
4.5.4	Detailed Beam Model . . . . .	92
4.5.5	Linear Continuum Model . . . . .	93

	Page	
4.5.6	Linear Beam-Continuum Model . . . . .	94
4.5.7	Nonlinear Beam-Continuum Model . . . . .	95
4.5.8	Hybrid Models . . . . .	95
4.6	Experimental Validation . . . . .	95
4.6.1	Impact Hammer Test . . . . .	96
4.6.2	Nonlinear Cyclic Static Test . . . . .	96
4.6.3	Five Story Shake Table Tests . . . . .	99
4.6.4	Real Time Hybrid Experiment . . . . .	104
4.7	Results . . . . .	104
4.7.1	Pure Numerical Results . . . . .	104
4.7.2	RTHS Results . . . . .	104
4.7.3	Nonlinear Behavior . . . . .	106
4.8	Discussion . . . . .	108
4.9	Conclusions . . . . .	110
5	PARALLEL ELLIPSOIDAL PERFECTLY MATCHED LAYERS FOR ACOUSTIC HELMHOLTZ PROBLEMS ON EXTERIOR DOMAINS . .	111
5.1	Introduction . . . . .	112
5.2	Theory . . . . .	113
5.2.1	Relations Between the PML Formulations . . . . .	124
5.3	LossFunction . . . . .	126
5.4	Implementation . . . . .	127
5.5	Verification . . . . .	128
5.5.1	Oriented Waveguide . . . . .	129
5.5.2	Offset Sphere . . . . .	129
5.5.3	Ellipse . . . . .	130
5.6	Solver . . . . .	131
5.6.1	Preconditioner . . . . .	131
5.7	Results . . . . .	132
5.7.1	Oriented Waveguide . . . . .	133
5.7.2	Offset Sphere . . . . .	133
5.7.3	Ellipsoid . . . . .	134
5.7.4	Comparison of Infinite Elements to PML . . . . .	135
5.8	Conclusions . . . . .	137
6	SUMMARY AND CONCLUSIONS . . . . .	139
	LIST OF REFERENCES . . . . .	141
	VITA . . . . .	151

## LIST OF TABLES

Table	Page
3.1 List of MTS+RTHS Prediction Schemes . . . . .	55
4.1 Static Verification of Beam-Continuum Coupling . . . . .	88
4.2 Static Verification of Nonlinear Q4 Elements . . . . .	89

## LIST OF FIGURES

Figure	Page
1.1 Structural Collapse during 1964 Niigata Earthquake [1] . . . . .	1
1.2 Multi-scale RTHS for interacting soil-structure systems . . . . .	4
2.1 Structure subject to impact/blast loading . . . . .	10
2.2 Representation of subdomain time steps for the MTS method . . . . .	17
2.3 Actual vs. predicted run times based on a basic cost model for the MTS method . . . . .	21
2.4 A sample MTS decomposition of a truss problem into two subdomains with different time steps. Point A lies in Subdomain A, Point B lies in Subdomain B, and Point I lies on the coupling interface. . . . .	24
2.5 Global time history response of nodes A, B, and I for the truss problem. The location of these points is shown in Figure 2.4. . . . .	26
2.6 Close up view of the time histories at nodes A, B, and I of the truss. The location of these points is shown in Figure 2.4. . . . .	27
2.7 Distribution of average local errors in nodal velocity at each degree of freedom for the truss problem. Average global error $\epsilon$ for each case is also mentioned. . . . .	28
2.8 Nine-story frame structure from ( [59]). Subdomain A is represented with light lines, and Subdomain B represented with dark lines. Pinned boundary conditions in the basement are represented by triangles, and lateral boundary conditions at the ground level are represented by white circles. . . . .	30
2.9 Distribution of average local error in nodal velocity for the frame problem.	30
2.10 Wave propagation in a plane-strain solid domain . . . . .	31
2.11 Distribution of average local error in nodal displacement for the Lamb Problem . . . . .	32
2.12 Average global errors and computational cost for different MTS decompositions of the Truss Problem . . . . .	35
2.13 Optimal decomposition of the truss problem. . . . .	36

Figure	Page
2.14 Average global errors and computational cost for different MTS decompositions of the frame problem with one sinusoidal load . . . . .	38
2.15 Average global errors and computational cost for different MTS decompositions of the frame problem with two sinusoidal loads . . . . .	38
2.16 Optimal Decompositions for the nine-story frame structure with one and two load points. Subdomain A (large time step $\Delta T$ ) is shown with light lines, and Subdomain B (small time step $\Delta t$ ) is shown with dark lines.	39
2.17 Average global errors and computational cost for different MTS decompositions of the Lamb wave propagation problem . . . . .	40
2.18 Optimal Decomposition for Lamb Problem. Subdomain A (large time step $\Delta T$ ) is shown with light coloring, and Subdomain B (small time step $\Delta t$ ) is shown with dark coloring. . . . .	40
3.1 An example of a Real-Time Hybrid Simulation Experiment . . . . .	45
3.2 Multi-time-step Decomposition of a Numerical Model . . . . .	46
3.3 Multi Time Step Decomposition in RTHS . . . . .	48
3.4 RTHS Control Flow Schematic . . . . .	49
3.5 RTHS order of computations and physical execution . . . . .	50
3.6 MTS Information Flow Schematic . . . . .	51
3.7 RTHS Control Flow Schematic with no prediction scheme . . . . .	54
3.8 Idealized RTHS Control Flow Schematic with prediction scheme . . . .	56
3.9 Nine-story frame structure from [59]. Pinned boundary conditions in the basement are represented by triangles, and lateral boundary conditions at the ground level are represented by white circles. . . . .	59
3.10 Comparison of Roof Displacement for the 184-DOF FEA model vs the 9-DOF Shear Model . . . . .	61
3.11 Control System Included for Realistic Computation Times . . . . .	62
3.12 virtual Real-Time Hybrid Simulation (vRTHS) . . . . .	62
3.13 Effect of Bouc-Wen Damper. The maximum roof displacement is reduced by 9.93 % . . . . .	63
3.14 Measured MR damper force for FEA model vs Shear Model (passive-on)	66
3.15 Damper Displacement for FEA model vs Shear Model . . . . .	66
3.16 Error with No Predictor (Scheme # 1) . . . . .	67

Figure	Page
3.17 Peak measured MR Damper force with differing time step ratios . . . . .	68
3.18 Predictor Error with $m = 4$ . . . . .	69
3.19 Measured MR Damper Force with $m = 4$ . . . . .	69
3.20 Predictor Error with $m = 16$ . . . . .	70
3.21 Measured MR Damper Force with $m = 16$ . . . . .	71
3.22 Predictor Error with $m = 64$ . . . . .	71
3.23 Measured MR Damper Force with $m = 64$ . . . . .	72
4.1 Real-Time Hybrid Simulation Experiment . . . . .	75
4.2 RTHS Control Flow Schematic . . . . .	76
4.3 System level RTHS with multi-scale models . . . . .	77
4.4 System Level RTHS with Multi-Scale Models . . . . .	79
4.5 Nonlinear Coupling Algorithm for MTS with Beam-Continuum Coupled Models . . . . .	81
4.6 Schematic of Beam-Continuum Coupling . . . . .	84
4.7 Beam-Continuum Coupling Deformed Shape . . . . .	84
4.8 Beam Continuum Deformed Shape and Stresses . . . . .	87
4.9 Quasi Static Verification Beam Bending . . . . .	88
4.10 Dynamic Verification Beam Bending . . . . .	89
4.11 Five Story Frame . . . . .	90
4.12 Elevation View of Single Story, reproduced from [105] . . . . .	91
4.13 Stress Strain Curve from Uni-axial Tension Test (reproduced from [106])	92
4.14 Experimental Models . . . . .	93
4.15 Beam Continuum Model Column-Floor Detail . . . . .	94
4.16 Natural Frequencies of Numerical Models . . . . .	96
4.17 One Story Cyclic Static Test . . . . .	97
4.18 One Story Cyclic Static Test - Linear Range . . . . .	97
4.19 One Story Cyclic Static Test . . . . .	98
4.20 One Story Cyclic Static Test . . . . .	98

Figure	Page
4.21 One Story Cyclic Static Test . . . . .	99
4.22 Shake Table Test A Results . . . . .	100
4.23 Shake Table Test B Results . . . . .	100
4.24 Shake Table Test C Results . . . . .	101
4.25 Shake Table Test D Results . . . . .	101
4.26 Scaled Experiment Comparison (Roof Displacement) . . . . .	102
4.27 Scaled Experiment Comparison (Roof Displacement) . . . . .	102
4.28 Scaled Experiment Comparison (Roof Displacement) . . . . .	103
4.29 Scaled Experiment Comparison (First Floor Displacement) . . . . .	103
4.30 RTHS Setup for Five Story Frame . . . . .	105
4.31 RTHS Control Flow Schematic . . . . .	106
4.32 vRTHS Control Flow Schematic . . . . .	106
4.33 Roof Displacement of Each Model (Numerical) . . . . .	107
4.34 First Floor Displacement (Numerical) . . . . .	107
4.35 Interaction Force (Numerical) . . . . .	108
4.36 Roof Displacement of Each Model (RTHS) . . . . .	108
4.37 First Floor Displacement (RTHS) . . . . .	109
4.38 Interaction Force (RTHS) . . . . .	109
4.39 Roof Displacement of Each Model (RTHS) . . . . .	110
5.1 Domains $\Omega_i$ and $\Omega_e$ and interface $\Gamma$ for the exterior acoustic problem. . . . .	114
5.2 2D Slice of a 3D Acoustic Mesh with PML Ellipsoidal Boundary . . . . .	128
5.3 Numerical Solution for Oriented Waveguide (150 Hz) . . . . .	129
5.4 Numerical Solution for Offset Sphere (50 Hz) . . . . .	130
5.5 Numerical Solution for Ellipsoid (100 Hz) . . . . .	130
5.6 Parameter Studies for Oriented Waveguide (100 Hz) . . . . .	134
5.7 Parameter Studies for OffsetSphere (50 Hz) . . . . .	135
5.8 Parameter Studies for Ellipsoid (100 Hz) . . . . .	136
5.9 Comparison Between IE and PML (100 Hz) . . . . .	136

Figure	Page
5.10 Comparison Between IE and PML (100 Hz) . . . . .	137
5.11 Comparison Between IE and PML (100 Hz) . . . . .	137

## ABSTRACT

Bunting, Gregory B. PhD, Purdue University, August 2016. Parallel Real-Time Hybrid Simulation of Structures Using Multi-Scale Models . Major Professor: Arun Prakash.

Real Time Hybrid Simulation (RTHS) is used to study the behavior of structures by partitioning the structure into two substructures, one that is constructed physically and the other that models the remainder of the structure numerically. These substructures are coupled physically using transfer devices such as hydraulic actuators and the simulation is performed in real time (usually using a time-step corresponding to 1024-Hz). The constraints imposed by the real-time nature of these simulations have historically precluded all but small and relatively simple numerical models that can be run deterministically in a fraction of a second. In this study we advance state of the art RTHS through the use of realistic multi-level finite element models that model the behavior of the structure accurately while still meeting the constraints of real-time computation.

The multi-time-step (MTS) time integration method enables one to use a refined model with a small time-step in the immediate vicinity of the physical substructure to match its fast time-scale, while a relatively coarser model with a large time-step is used for other parts of the structure. A systematic approach for traversing the space of possible MTS decompositions and characterizing the nature of how solution errors and computational costs vary for different decompositions is presented. Based on this approach, optimal decompositions that maximize the benefit of the MTS methods are identified. It is also shown that MTS methods can be used effectively to lower computational cost while maintaining accuracy of the solution by distributing errors evenly across the problem domain.

The MTS time integration method is adapted to RTHS, called MTS+RTHS. Specifically, a predictor is used for the MTS coupling to reduce errors incurred by applying asynchronous updates to the physical substructure. An in-depth study is conducted investigating a series of choices of the predictor, and the effect of predictor choice on the error in both the physical and numerical substructures. It is demonstrated that MTS+RTHS is not only viable, but essential when using high-fidelity models. It is shown that the synchronization error for a problem with high time step ratios is the same order of magnitude as experimental error for a benchmark problem.

Parallel multi-scale RTHS is used to allow high fidelity numerical models in RTHS. Large numerical models that are multi scale in both *space* and *time* use a finer spatial discretization are used in RTHS. The coupling of linear beam elements with nonlinear continuum elements is used to create models that capture nonlinear behavior at structural joints, but are still capable of being solved in real time. The **Cybermech** platform is used to conduct these parallel experiments, and demonstrate the benefits of high fidelity models.

Perfectly Matched Layers (PML) is used as an absorbing boundary for the Helmholtz problem. A parallel, ellipsoidal formulation for PML is presented for acoustic elements, minimizing the need for large numbers of elements in the exterior of the domain. Such a tool could be used a RTHS experiment involving soil structure interaction, reducing the size and number of elements in the soil domain. A comparison between the PML formulation and ellipsoidal infinite elements is conducted, focusing on iteration counts of the parallel linear solver. Perfectly Matched layers are shown to outperform infinite elements on large parallel problems, where the poor matrix conditioning of higher order infinite elements results in increased computational cost.

## 1. INTRODUCTION

In 1994, the Northridge earthquake in the San Fernando Valley region of Los Angeles, California killed 57 people and caused estimated property damage of more than \$20 billion. This earthquake and others like it have driven research to understand why and when earthquakes occur, and what can be done to reduce casualties and damage. Numerous changes have been made to structural design codes, but it is often unclear how these changes perform. Earthquakes come in a variety of types, sizes, and locations, and years can pass between major earthquakes even at the most earthquake susceptible locations. Older buildings that were designed before changes in the code further muddy the water. Figure 1.1 shows a building collapse from the 1964 Niigata earthquake.



Figure 1.1.: Structural Collapse during 1964 Niigata Earthquake [1]

The 1994 Northridge earthquake is one of many events where large structures were subjected to extreme dynamic loading. Other events include earthquakes, tsunamis, hurricanes, and tornadoes. Some of these events have resulted in large scale property damage and building collapse, while others have not. Even within a single event, some

buildings have significantly more damage than others. Civil engineering structures are designed over a range of time periods, built from different material types, and designed to match local building styles and design codes. Additionally, factors such as type of earthquake, soil properties, and foundation design also affect the structure. Each large scale structure is unique, so it difficult to understand why one building collapsed but another did not.

Research in Structural Engineering usually involves either physical testing or numerical modeling of structures / components under a set of load and boundary conditions to evaluate their performance in the real-world. These tests are done at various scales, ranging from small-scale tests on material samples, structural parts and connections up to full-scale tests on complete structures. Naturally, the results of these tests and simulations are better when the structure / component is modeled as closely as possible to its state in the full structure.

As physical testing of large structures such as buildings and bridges at full scale is prohibitively expensive, researchers have developed a hybrid simulation approach that augments physical specimens of structural parts with numerical models of the remainder of the structure. This helps to simulate the response of the small structural component within the system-level dynamics of the entire structure. An example of a hybrid simulation would be a single story physical shear frame structure, with the remainder of the structure simulated numerically.

Dynamically loaded structures exhibit nonlinear, rate-dependent behavior, so it is important to run hybrid tests at real time to capture and understand the rate-dependent response. RTHS is a hybrid test where the response of the numerical model is computed at run time, and must be completed within each time step of the physical system [2]. The focus of these tests is to capture the structural dynamics and failure modes of these structures, so the testing frequency must be sufficiently high to capture the response of the natural frequencies of interest. Frequencies of up to 1024Hz are commonly used and that limits the size of the numerical model that can be solved in real-time. Typically the size and complexity of numerical models used in RTHS

are limited by the computational resources available to solve these models in a single time-step. Limiting the size of the numerical model can restrict the applicability of the results from a hybrid simulation [3]. There has been extensive research conducted [4–7] for developing numerical models that are able to run within the real time constraints and still be able to replicate the physical system of interest adequately. Castaneda *et al.* [8] supplied a RT tool for building computational models. There is a need for higher fidelity models in RTHS that currently exists in the literature. These models are generally too large to run within the real-time constraints.

A multi-scale model is one that can capture behavior of a structure at multiple scales. In this work, concurrent multi-scale models are utilized. In large parts of the model, beam elements are used to describe the behavior, and in critical regions, such as connections undergoing high stresses or strains, refined nonlinear continuum elements are used. The continuum elements are able to represent the behavior better, but come with an additional computational cost. This cost is non-trivial, so it is important balance the need for accuracy with the computational cost.

Figure 1.2 shows a system-level RTHS experiment that uses multi-time-step (MTS) methods and multi-scale techniques to capture the behavior of a structure. In practice, it may not be possible to run such a RTHS in real time. A large numerical model, detailed connections, soil-structure interaction, and the surrounding soil domain must all be computed in real-time to perform a successful test. One approach is to reduce the scope of the model by eliminating the soil and foundation, linearizing the nonlinear behavior, and simplifying the building model to one degree of freedom on each floor. In this work, we present an alternative approach, where different multi-scale techniques are used to reduce computations in some regions of the model while maintaining a high fidelity test. A method for truncating an acoustic domain, called Perfectly Matched Layers (PML), is also presented, though not implemented in vRTHS and RTHS.

In practice, the entire RTHS experiment is simulated on a computer before the actual RTHS experiment is run. This is called virtual Real Time Hybrid Simulation (vRTHS). Virtual RTHS includes the effect of control, compensation, delay, and

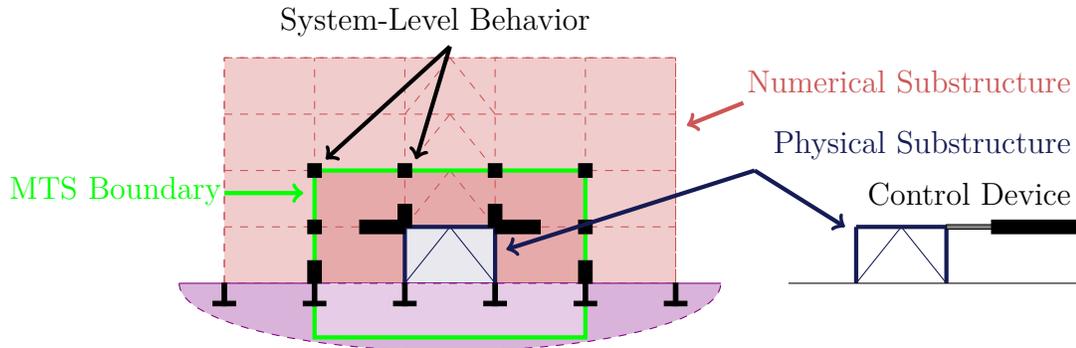


Figure 1.2.: Multi-scale RTHS for interacting soil-structure systems

experimental noise, but these are simulated with models that represent the dynamics of the RTHS hardware. vRTHS is useful for testing new numerical methods and models before coupling with a physical experiment, to minimize the physical danger to both people and equipment. Each experiment in this work is performed as a vRTHS, with the extension to RTHS planned once the models and methods are finalized. To achieve these goals, the **Cybermech** [9] [10] platform was developed to allow complex multi-rate parallel C++ code to run with a real-time scheduler, replacing the existing Matlab/xPC/Simulink programming environment for RTHS.

## 1.1 Objectives

The types of numerical models that model a building realistically are too large to be run within real time parameters. Multi-scale techniques are necessary to conduct an RTHS experiment that accurately captures the important characteristics of the structure while still meeting real time constraints. The overall goal of this research is to advance the state-of-the-art in RTHS by using multi-time-step (MTS) methods and multi-scale techniques to capture the behavior of a structure. This work focuses on computational methods that allow high fidelity models to be run in real time, using multi-scale approaches that refine the model at specific spatial locations determined to be of high interest. Extensive analysis of these multi-scale models is conducted,

and the benefits of such models are demonstrated. The following specific objectives are needed to achieve this goal:

1. Characterize MTS error and cost
2. Temporal multi-scale models in RTHS
3. Parallel nonlinear spatial multi-scale models in RTHS
4. Perfectly Matched Layers with an ellipsoidal boundary

### 1.1.1 Characterize MTS Error and Performance

Multi-time-step (MTS) methods are a type of temporal discretization where different subdomains within the problem domain are solved with different temporal discretizations. While some problems can be solved efficiently at a single time step, there are several problems that are well suited for multi-time-step methods. These are problems that exhibit multi-scale behavior in time. In particular, wave propagation problems show this behavior, as they tend to create reflections and transients in sub regions of the problem domain.

Inherent problem properties, such as geometry, loading, and boundary conditions also affect the quality of the solution. For some problems, the problem characteristics can lead to regions with high error. In the past such regions, or features, have been dealt with by refining the mesh in the area around the feature. However, location specific temporal discretization is not possible with uniform time-stepping schemes (UTS). Using MTS, the temporal discretization can be set for each element or subdomain. Choosing an appropriate time-step for each can eliminate these peaks in local error and consequently reduce global error in the problem.

MTS integration schemes have been developed by Belytschko *et al.* [11], Smolinski [12], Daniel [13], Gravouil and Combescure [14] [15], and Prakash and Hjelmstad [16] [17]. Each of these works provides an integration scheme for solving subdomains with different time-step, but does not discuss how to best decompose a problem into

subdomains. An approach for finding optimal decompositions for MTS problems is developed. A method for decomposing problems and selecting a time-step ratio in a way that minimizes local and global error while meeting computational cost requirements is established.

### 1.1.2 Temporal multi-scale models in RTHS

There is a clear benefit in applying Multi-Time-Step to Real Time Hybrid Simulation, called MTS+RTHS, as it decreases the total computational cost of the model, while a portion of the model is still solved at the same rate as the physical substructure. When the MTS decomposition is selected such that all numerical-physical interaction points are solved at the small time step, the control system and physical substructure can run at the small time step. However, MTS requires an asynchronous update, where information from the current time step is used to update the numerical solution at previous times. To use Multi-Time-Step with Real Time Hybrid Simulation (MTS+RTHS), an approach is developed to predict these updates, and apply the predicted terms to the physical substructure at the appropriate time.

### 1.1.3 Parallel nonlinear spatial multi-scale models in RTHS

Often, nonlinear behavior occurs only in a small region, or several small regions, of the structure. If the model is well understood, nonlinear elements are only necessary in particular locations, which can be selected *a priori*. For high-fidelity models, it is essential to capture this nonlinear behavior, but for RTHS, it is essential to minimize computational cost. A small displacement version of the beam-continuum coupling method developed by Pitandi [18] is used to transition between Euler Bernoulli beam elements and nonlinear continuum elements. OpenMP and openBLAS are used to solve such models in parallel, as a single processor core cannot solve such a problem in real-time. The **Cybermech** platform was modified to support these capabilities.

#### 1.1.4 Perfectly Matched Layers with an Ellipsoidal Boundary

When an earthquake load acts on a building, the response of the soil influences the dynamic response of the building. Additionally, the motion of the structure influences the response of the soil. Soil-structure interaction (SSI) is the two way influence between the structure and the soil during a dynamic loading event. Conventional structural design methods neglect SSI effects [19]. To model SSI in an RTHS, some external boundary must be placed on the soil to absorb outgoing waves. Perfectly Matched Layers (PML) is one such boundary, originally [20] were developed in 1994 for simulating electromagnetic waves on exterior domains. PML provides an elegant and effective means of terminating a finite element mesh without reflections [21–23]. Conceptually, a layer of elements is added to the boundary of the domain of interest for providing absorbing of all incoming energy, independent of incident angle. The PML method can be applied on any exterior boundary surface, including non-convex surfaces, flat surfaces, and surfaces with corners. This is in contrast to infinite elements [24], and many types of absorbing boundary conditions [25], which are restricted to surfaces corresponding to separable coordinate systems. PML is used for a variety of problem types, including unbounded acoustic domains and SSI problems. In the future, PML can be used in an RTHS experiment that captures SSI.

This work presents an implementation of PML in an ellipsoidal coordinate system. Perfectly matched layers are created as a structured mesh on an ellipsoidal boundary surface of an unstructured mesh. In order to facilitate the implementation in a finite element code that is based on Cartesian coordinates, a mapped PML method is presented in which the weak formulation is mapped back to Cartesian coordinates. This allows the PML parameters such as thickness and number of elements to be changed without the need to re-mesh the entire model, which can be computationally intensive in large models. PML is implemented in Sierra SD [26, 27], a parallel structural dynamics code owned by Sandia National Labs. The PML is compared with

the absorbing boundary condition and the infinite element solution for both accuracy and Krylov convergence rates.

Chapter 2 presents a detailed analysis of the costs and benefits associated with various MTS decompositions. Chapter 3 presents the use of models that are multi-scale in time within virtual RTHS, and presents a detailed analysis of predictor schemes for asynchronous updates. Chapter 4 presents the use of models that are multi-scale in space and time within RTHS, and demonstrates the benefits of these methods. Chapter 5 presents an ellipsoidal PML formulation, and a detailed comparison to infinite elements when used in a massively parallel environment. Chapter 6 summarizes the work presented in Chapters 2-5, and the impact to the field of RTHS.

## 2. CHARACTERIZING ERRORS AND EVALUATING PERFORMANCE OF TRANSIENT SIMULATIONS USING MULTI-TIME-STEP INTEGRATION

The multi-time-step method of time integration for problems in structural dynamics allows one to decompose the problem domain into small subdomains and use different time-steps within each subdomain to reduce the computational cost of solving such problems. However, the number of possible decompositions and their associated time-steps for a given model is huge and grows exponentially with the number of elements. Finding an optimal decomposition that minimizes error in the solution while maintaining a bound on the computational cost is challenging. In this work, we utilize existing multi-time-step methods and, for the first time, devise a systematic approach for traversing the space of possible decompositions to characterize the nature of how solution errors and computational costs vary for different decompositions. Through numerical examples for three different types of structures, trusses, frames, and continuum solid bodies, it is shown that the characteristics of these error and cost functions is similar across problem types. Based on these functions, optimal decompositions that maximize the benefits of multi-time-step methods are identified.

### 2.1 Introduction

Structural dynamics is the study of the behavior of structures under time varying loads. To study such behavior, structures are modeled using different theories of mechanics that lead to different types of idealized models. Such idealizations require a series of assumptions about the behavior of the structure and contribute to error in the resulting solution. A common way of studying idealized models of structures is

to discretize them in space using numerical methods such as finite elements and to use a time-stepping scheme to integrate the governing differential equations of motion in time. This approach results in a time history response of the structure subject to different loading scenarios, based on which one may interpret the results in order to gain insight into the dynamic properties of the structure. An important aspect in the numerical modeling of structures is the choice of an appropriate spatial mesh and a corresponding time-step for time integration. Error in the numerical solution can be reduced by using a finer spatial mesh and/or by reducing the time-step [28, 29], albeit at a greater computational cost.

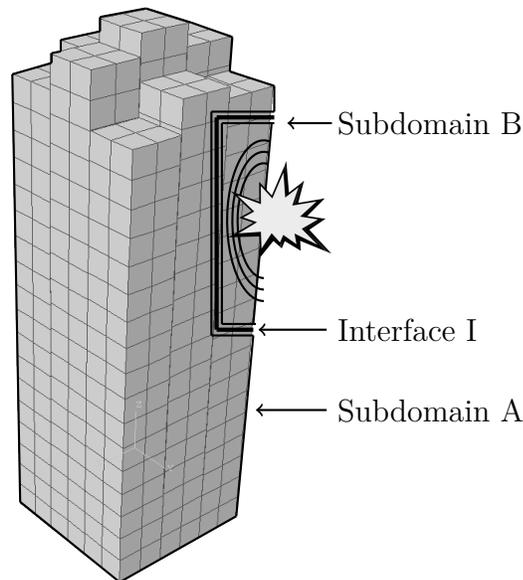


Figure 2.1.: Structure subject to impact/blast loading

Conventional numerical methods for solving problems in structural dynamics employ a uniform time-step (UTS) for the entire structural problem domain. However, this approach is not suitable for problems where the structural response may span across multiple time scales. For instance, as shown in Figure 2.1, a structure subject to an impact/blast load may exhibit steep spatial and temporal gradients and/or damage in the vicinity of the event, whereas the rest of the structure may respond more moderately as the wave dissipates propagating away from the region. For such

problems, one usually employs a finer spatial discretization in parts of the structure that are directly affected by the event, but using a UTS integration for the entire structure is not appropriate because using a small time-step would lead to high computational cost, while even using a moderately large time-step would cause large local errors in the solution.

As an alternative to UTS integration, domain decomposition (DD) methods [30–32] can be used to divide the structural problem domain into smaller subdomains in order to enable the use of different types of models and/or different time-steps in different subdomains. The use of different time-steps in different subdomains is facilitated by multi-time-step (MTS) methods [11–17]. MTS methods, when used with an appropriate decomposition of subdomains and associated time-steps, have been shown to reduce errors and computational cost for problems with multiple temporal scales.

While several techniques have been reported in the literature to allow different spatial and temporal discretizations across the model, there is limited knowledge on how to best divide a structure into subdomains and how to choose the time-step size for each subdomain in the resulting decomposed model. Depending upon the application, the trade-off between computational cost and numerical error can be a significant factor in determining how to construct a spatial and temporal decomposition for a given numerical model. For instance, when studying the effect of different design parameters on the response of a structure under various types of dynamic loads, a structural engineer may need to conduct numerous simulations of the same structure while being constrained by the amount of permissible errors in these simulations. In such situations, one may be able to find a set of decompositions that satisfy a pre-determined error criterion and pick the particular decomposition that is associated with the minimum computational cost to speed up their design process. On the other hand, applications such as active control of tall structures, real-time structural health monitoring, and real-time hybrid simulation, require one to be able to compute the solution of their numerical models within a specified interval of time due to the real-time nature of these applications. Once again, there may be a set of possible

decompositions of a numerical model which meet the real-time constraints and one may choose a decomposition that is associated with the minimum error in order to obtain accurate solutions.

In this work, the variation of solution error and computational cost for different decompositions is characterized. The space of possible decompositions is explored, and a systematic approach for identifying an optimal decomposition that minimizes these performance metrics is presented. Due to the extremely large number of possible decompositions and the complexity of choosing appropriate time-steps for them, this study is restricted to investigating only two-subdomain decompositions and their associated time-steps. Three types of problems, trusses, frames and continuum solid bodies are studied to investigate the effect of choosing different decomposition and time-step combinations on solution error and computational cost. Insights from these problems are used to identify close to optimal decompositions.

## 2.2 Overview of time integration methods

The governing system of ordinary differential equations for structural dynamics of a structure can be written in their semi-discrete form as:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{p} \quad (2.1)$$

where  $\mathbf{M}$ ,  $\mathbf{C}$ , and  $\mathbf{K}$ , denote the mass, damping and stiffness matrices respectively. The vector  $\mathbf{u}$  represents the displacements at all the degrees of freedom in the model  $\Omega$  and  $\mathbf{p}$  represents time varying loads on the structure. Each superimposed dot ( $\dot{\cdot}$ ) represents a time-derivative. Initial and boundary conditions for the problem are specified as:

$$\mathbf{u}(0) = \mathbf{d}_0 \quad \dot{\mathbf{u}}(0) = \mathbf{v}_0 \quad (2.2)$$

$$\mathbf{u}(t) = \bar{\mathbf{d}}(t) \quad (2.3)$$

where  $\mathbf{d}_0$  and  $\mathbf{v}_0$  are the given displacement and velocity respectively, at time  $t = 0$  and  $\bar{\mathbf{d}}(t)$  denotes the displacement time history of a subset of the degrees of freedom  $\Gamma_D$  that have specified boundary conditions imposed on them.

The second order system of ordinary differential equations (2.1), is most commonly solved by time stepping schemes that enforce the equation at discrete instants of time  $t_n$  where  $0 < n < N$  in the time duration of interest, say  $[t_0, t_N]$ . A known state of the system at  $t_n$  is advanced by a time-step  $\Delta t = t_{n+1} - t_n$  to obtain an approximate solution for the state at  $t_{n+1}$ . This process is repeated to advance from one instant of time to the next successively to obtain a time-history of the response from  $t_0$  to  $t_N$  and is called time-stepping.

A number of finite-difference-based time-stepping schemes exist in the literature including the Newmark method [33], Wilson- $\theta$  method [34], the HHT- $\alpha$  method [35], the WBZ- $\alpha$  method [36], and various Runge-Kutta schemes [37]. Denoting the state of the system at time  $t_n$  with  $\mathbf{z}_n = [\mathbf{a}_n, \mathbf{v}_n, \mathbf{d}_n]^T$  where  $\mathbf{a}_n$ ,  $\mathbf{v}_n$ , and  $\mathbf{d}_n$  denote the acceleration, velocity and displacement vectors respectively, the Newmark method [33] can be expressed compactly as:

$$\mathbb{M} \mathbf{z}_{n+1} = \mathbb{P}_{n+1} - \mathbb{N} \mathbf{z}_n \quad (2.4)$$

where  $\mathbb{P}_{n+1} = [\mathbf{p}_{n+1}, \mathbf{0}, \mathbf{0}]^T$  denotes the block external load vector consisting of the discrete load vector  $\mathbf{p}_{n+1}$  for each degree of freedom, and the block matrices  $\mathbb{M}$  and  $\mathbb{N}$  are defined as:

$$\mathbb{M} = \begin{bmatrix} \mathbf{M} & \mathbf{C} & \mathbf{K} \\ -\gamma\Delta t\mathbf{I} & \mathbf{I} & \mathbf{0} \\ -\beta\Delta t^2\mathbf{I} & \mathbf{0} & \mathbf{I} \end{bmatrix} \text{ and } \mathbb{N} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\Delta t(1-\gamma)\mathbf{I} & -\mathbf{I} & \mathbf{0} \\ -\Delta t^2(\frac{1}{2}-\beta)\mathbf{I} & -\Delta t\mathbf{I} & -\mathbf{I} \end{bmatrix} \quad (2.5)$$

where  $\gamma$  and  $\beta$  denote algorithmic parameters that allow the choice between implicit and explicit time integration schemes.  $\eta$  denotes the number of degrees of freedom in the system, and  $\mathbf{I}$  denotes the identity matrix of size  $\eta$ . The state vector  $\mathbf{z}_n$  describes

the entire state of the system at a specific instant of time  $t_n$  and solving the system of equations (2.4) to find  $z_{n+1}$  amounts to advancing this state to  $t_{n+1}$ .

When running a large problem, choosing an appropriate  $\Delta t$  is critical in determining the quality of the solution and the computational cost it takes to obtain it. Choosing a large  $\Delta t$  may result in unacceptably large numerical errors, while a small  $\Delta t$  may result in an unacceptably large computational cost. This trade-off between the computational cost and solution error is also affected by the order of accuracy of the time integration scheme. The Newmark scheme [33] has been shown to be second-order accurate ( $\mathcal{O}(\Delta t^2)$ ) for  $\gamma = \frac{1}{2}$  i.e. halving the time-step reduces the error by a factor of four. However, halving the time-step also doubles the computational cost, and so an optimal  $\Delta t$  must be chosen that meets both criteria. Chopra [29] recommends picking a time-step of  $\Delta t < 0.1T_j$ , where  $T_j$  is the period corresponding to the highest natural frequency of the undamped structure. Finally, the time-step must also be sufficiently small to be able to capture the load history with sufficient resolution.

In addition to computational cost, the choice of  $\Delta t$  also affects the stability of the solution. A time-stepping scheme is said to be stable if the solution remains bounded at all times. For conditionally stable methods, the critical time-step  $\Delta t_{cr}$  provides an upper-bound for  $\Delta t$  is governed by the Courant limit [29]. A basic estimate of the critical time-step (assuming a lumped mass matrix) can be obtained as time it takes the wave to traverse the smallest element:  $\Delta t_{cr} \approx l_e/c$  where the  $l_e$  denotes the smallest dimension of the smallest element in the mesh and the material wave speed  $c$  is given in term of the elastic modulus ( $E$ ) and mass density ( $\rho$ ) as  $c = \sqrt{E/\rho}$ . For consistent mass matrices, the critical time-step limit is even smaller. For problems such as crack propagation, where the crack-tip may be meshed with very small elements, stability requirements can force one to use extremely small time steps.

For some transient problems, the temporal characteristics of the solution change with time and this often requires one to adaptively modify the time-step during the simulation. Bergan and Mollestad [38] examine an algorithm that changes the time-step dynamically. Park and Underwood [39] also investigated the critical time-step

for explicit problems, and how it might change during the simulation for nonlinear problems. Hilber *et al.* [35] show that changing the time-step during computation can lead to changes in algorithmic damping, which itself is dependent on the magnitude of the time-step. Sanchez-Gasca *et al.* [40] discuss an algorithm where the time-step for the entire problem is adapted during integration to capture transients. Even though such methods allow the use different time-steps during different periods of the simulation, they still employ the same uniform time-step for the entire model for time-stepping. For problems containing multiple spatial and temporal scales in their solutions, using a uniform time-step is computationally very inefficient.

### 2.3 Multi-time-step methods

Multi-time-step (MTS) methods utilize domain decomposition (DD) techniques to split a large problem into several smaller subdomains and enable different subdomains to be solved with different time-steps and time integration schemes. This allows one to tailor the time-steps for individual subdomains, not only improving accuracy, but also reducing computational costs. Furthermore, by virtue of DD, the computation can be parallelized for solving very large problems more efficiently.

There are several approaches available in the literature for DD and MTS. Asynchronous variational integrators (AVI) are a family of time integration techniques where elements are solved at different time-steps. Used for explicit integration methods, different time-step can be selected for each element. This removes the requirement that every element in the mesh be solved at the same time-step as the smallest element, reducing total computation time [41]. Fong *et al.* [42] perform a detailed analysis of the stability of the method, and shown that the method can absorb weak instabilities. Kael *et al.* [43] implement a scalable parallel AVI, and show that examples exist where the parallel AVI greatly reduces computational time when compared to uniform time-stepping algorithms.

Belytscho and Mullen [44] presented explicit-implicit mesh partitions where the same time-step is used for both partitions and demonstrated stability for this method. Smolinski and co authors [45] and [12] demonstrated stability for an explicit multi-time-step integration of first and second order differential equations. Farhat and Roux [46] extend this approach to finite element tearing and interconnecting (FETI), that allows subdomains to be solved independently (possibly in parallel) and using Lagrange multipliers to couple the individual solutions. Different types of sub-cycling algorithms that also enable a finer resolution of specific areas of interest in a problem were developed [13], [47], [48]. Felippa *et al.* [49] use domain decomposition to split a large problem into several subdomains, allowing for an implicit algorithm in one part of the system, and an explicit algorithm in another part of the system. Gravouil, Combescure, and others [14, 50, 51] describe a stable coupling algorithm to solve multiple subdomains at different time-steps using the Newmark family of integrators and later extended the method to heterogeneous asynchronous time integrators [52]. Prakash and co-workers [16, 17, 53] developed a MTS method that is unconditionally stable and preserves the energy-norm of the underlying time integration schemes and is computationally very efficient. Park and Felippa [54] perform a detailed analysis on the accuracy of partitioned integrators, and provide a method for estimating the frequency distortion introduced by such partitioned analyses. An brief overview of this MTS method is presented next to facilitate the analysis of computational cost and quantification of errors.

### 2.3.1 Governing equations for the multi-time-step method

Problems in structural dynamics that span multiple temporal scales need to be treated with different time-steps in different subdomains (see Figure 2.1). For the discussion here, the problem is split into two subdomains, A and B that are allowed to run at different time-steps  $\Delta T$  and  $\Delta t$  respectively, where  $\Delta T = m\Delta t$  for some integer time-step ratio  $m$  as shown in Figure 2.2. The two subdomains may also have different

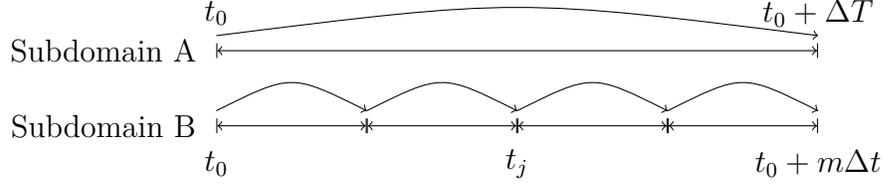


Figure 2.2.: Representation of subdomain time steps for the MTS method

Newmark time-integration parameters  $\gamma$  and  $\beta$  corresponding to different implicit or explicit schemes. Given the state of subdomains A and B at  $t_0$ , multi-time-stepping is used to solve the state at  $t_m = t_0 + m\Delta t$  by advancing subdomain A through one big time-step  $\Delta T$  and advancing subdomain B through  $m$  small time-steps  $\Delta t$ .

The equation of motion for subdomain A is solved using the large time-step  $\Delta T$ :

$$\mathbb{M}^A \mathbb{z}_m^A + \mathbb{C}^A \boldsymbol{\lambda}_m = \mathbb{P}_m^A - \mathbb{N}^A \mathbb{z}_0^A \quad (2.6)$$

where  $\boldsymbol{\lambda}$  denotes the Lagrange multipliers on the interface and the matrix  $\mathbb{C}^A$  represents a mapping between the degrees of freedom on the interface to the degrees of freedom for subdomain A (see [16] for details). The equation of motion for subdomain B is solved  $m$  times using small time-step  $\Delta t$ :

$$\mathbb{M}^B \mathbb{z}_j^B + \mathbb{C}^B \boldsymbol{\lambda}_j = \mathbb{P}_j^B - \mathbb{N}^B \mathbb{z}_{j-1}^B \quad \forall j \in [1, m] \quad (2.7)$$

where the matrix  $\mathbb{C}^B$  represents a similar mapping between the degrees of freedom on the interface to the degrees of freedom for subdomain B. In order to compute Lagrange multipliers  $\boldsymbol{\lambda}_j$  at the intermediate time-steps  $t_j$ , a balance of tractions at the interface is enforced:

$$\mathbb{C}^{AT} [\mathbb{M}^A \mathbb{z}_j^A - \mathbb{P}_j^A + \mathbb{C}^A \boldsymbol{\lambda}_j] = \mathbf{0} \quad \forall j \in [1, m] \quad (2.8)$$

where the states  $\mathbb{z}_j^A$  for subdomain A are obtained by linear interpolation:

$$\mathbb{z}_j^A = \left(1 - \frac{j}{m}\right) \mathbb{z}_0^A + \left(\frac{j}{m}\right) \mathbb{z}_m^A \quad (2.9)$$

Finally, continuity of velocities across the interface between subdomains A and B is enforced at the large time-step:

$$\mathbb{B}^A \mathbb{z}_m^A - \mathbb{B}^B \mathbb{z}_m^B = 0 \quad (2.10)$$

where the block matrices  $\mathbb{B}^A$  and  $\mathbb{B}^B$  represent a reverse mapping from degrees of freedom for subdomains A and B to the interface degrees of freedom and are used to maintain continuity of velocity between the subdomains. It has been shown that continuity of velocities leads to a stable coupled formulation for a range of different coupling parameters rather than the more intuitive continuity of displacements (see [14], [16]), or continuity of accelerations.

### 2.3.2 Solution procedure

On each subdomain, the solution is split into two parts,  $\bar{z}$  and  $z'$  as follows:

$$z^A = \bar{z}^A + z'^A \quad ; \quad z^B = \bar{z}^B + z'^B \quad (2.11)$$

where  $\bar{z}$  that satisfies the uncoupled set of governing equations (in the absence of  $\boldsymbol{\lambda}$ ), and  $z'$  which accounts for the effect of the coupling Lagrange multipliers. Using this approach the solution of the coupled subdomains A and B can be achieved in three steps: (i) the uncoupled subdomain solve (ii) interface solve (iii) subdomain updates. These three steps are outlined below.

#### Uncoupled Subdomain Solution

First, the uncoupled solutions for subdomains A and B are obtained as:

$$\bar{z}_m^A = [\mathbb{M}^A]^{-1} [\mathbb{P}_m^A - \mathbb{N}^A z_0^A] \quad \text{and} \quad (2.12)$$

$$\bar{z}_j^B = [\mathbb{M}^B]^{-1} [\mathbb{P}_j^B - \mathbb{N}^B \bar{z}_{j-1}^B - \mathbb{C}^B \mathbf{s}_j] \quad \forall j \in [1, m] \quad (2.13)$$

where  $\bar{z}_0^B = z_0^B$  and  $\mathbf{s}_j$  represents a predicted value of the coupling Lagrange multipliers given by:

$$\mathbf{s}_j = \left(1 - \frac{j}{m}\right) \boldsymbol{\lambda}_0 \quad (2.14)$$

## Interface Solve

Once the uncoupled solutions have been obtained, the coupling Lagrange multipliers are solved using the equation:

$$\boldsymbol{\lambda}_m = \mathbf{H}^{-1} \mathbf{f} \quad (2.15)$$

where the interface matrix  $\mathbf{H}$  is defined as:

$$\mathbf{H} = \mathbb{B}^A \mathbb{Y}_m^A + \mathbb{B}^B \mathbb{Y}_m^B \quad (2.16)$$

and the right hand side vector  $\mathbf{f}$  is given by the expression:

$$\mathbf{f} = \mathbb{B}^A \bar{\mathbb{Z}}_m^A + \mathbb{B}^B \bar{\mathbb{Z}}_m^B \quad (2.17)$$

In the above equations, the matrices  $\mathbb{Y}_m^A$  and  $\mathbb{Y}_m^B$  are obtained from the following expressions:

$$\mathbb{Y}_m^A = [\mathbb{M}^A]^{-1} \mathbb{C}^{AT} \quad \text{and} \quad (2.18)$$

$$\mathbb{Y}_j^B = [\mathbb{M}^B]^{-1} \left[ \left( \frac{j}{m} \mathbb{C}^{BT} \right) - \mathbb{N}^B \mathbb{Y}_{j-1}^B \right] \quad \forall j \in [1, m] \quad (2.19)$$

respectively, where  $\mathbb{Y}_0^B = 0$ . Note that the  $\mathbb{Y}$  matrices remain constant (for linear problems) for all time-steps and therefore they can be computed prior to beginning the time-stepping loop. Consequently, the interface matrix  $\mathbf{H}$  also remains constant for the duration of the time-stepping and it too can be computed, factorized and stored ahead of time for greater computational efficiency.

## Subdomain Updates

Finally, having computed the coupling Lagrange multipliers, the subdomain solutions can be updated using following contributions:

$$\mathbb{Z}'_m^A = -\mathbb{Y}_m^A \boldsymbol{\lambda}_m \quad (2.20)$$

$$\mathbb{Z}'_j^B = -\mathbb{Y}_j^B \boldsymbol{\lambda}_m \quad \forall j \in [1, m] \quad (2.21)$$

Note that the total solution for each subdomain is obtained using equation (2.11) by simply summing up the contributions from the uncoupled solution in equations (2.12)-(2.13) and the updates from equations (2.20)-(2.21).

### 2.3.3 Computational cost of the multi-time-step method

The computational cost of the multi-time-step method can be estimated by measuring the computational cost of each of the three steps above. The cost associated with each set of operations is related to the size of the subdomains A and B and the size of the interface between them. Let  $\eta_A$  and  $\eta_B$  denote the number of degrees of freedom in Subdomains A and B respectively and  $\eta_I$  be the number of degrees of freedom on the interface.

For the three steps in the MTS solution algorithm, a basic estimate of the computational cost of each step can be obtained as:

1. Uncoupled subdomain solve: Solving against pre-factorized  $\mathbb{M}$  matrices, the cost of obtaining the uncoupled solutions from equations (2.12)-(2.13) will be:  $\mathcal{O}(\eta_A^2) + \mathcal{O}(m\eta_B^2)$ .
2. Interface solve: Again, solving for  $\boldsymbol{\lambda}$  against a pre-factorized  $\mathbf{H}$  matrix in equation (2.15) involves  $\mathcal{O}(\eta_I)^2$  operations.
3. Subdomain updates: Finally, the subdomain updates in equations (2.20)-(2.21) result in a cost of  $\mathcal{O}(\eta_A\eta_I + m\eta_B\eta_I)$ .

For a given problem and a specific hardware platform, the computational cost, in terms of run-time  $R$  can be estimated as:

$$R(\eta_A, \eta_B, \eta_I, m) \geq c_1(\eta_A^2 + m\eta_B^2) + c_2\eta_I^2 + c_3(\eta_A\eta_I + m\eta_B\eta_I) \quad (2.22)$$

where each of the constant coefficients above can be calibrated to a particular computer by solving a set of problems multiple times and recording the actual run-times for each of the three steps above. Note that the cost model above assumes that one is

using implicit methods. However, it can be readily modified to account for cases when one or both subdomains may be solved using explicit time integration methods.

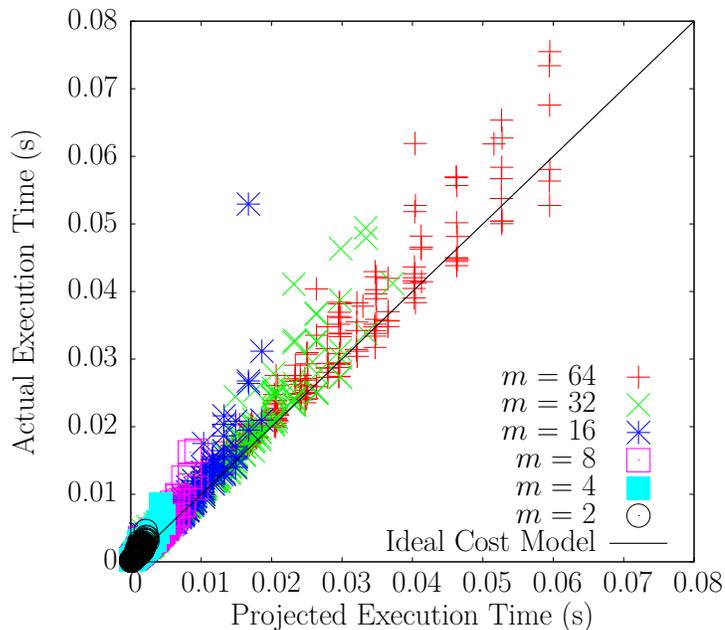


Figure 2.3.: Actual vs. predicted run times based on a basic cost model for the MTS method

A sample of such calibration is shown in Figure 2.3 which compares actual run-times to their predicted values obtained from the equation above for 500 different simulation runs with varying model parameters ( $\eta_A$ ,  $\eta_B$ ,  $\eta_I$ , and  $m$ ). It is observed that for large problems, the cost of the MTS method is dominated by Step 1 (subdomain solves):  $c_1(\eta_A^2) + \mathcal{O}(m\eta_B^2)$  where the value of  $c_1$  is found to be  $2.25 \times 10^{-8}$  for problems with the number degrees of freedom ranging from a few hundred to a few thousand. All the problems were run in MATLAB on a Dell T7600 Precision workstation with dual 8-core 2.4 GHz Xeon processors, 64 GB RAM, and 4.5 TB disk space, running Red Hat Enterprise Linux (RHEL 6).

The appeal of MTS methods is immediately evident from the preceding discussion. In comparison to the cost of solving a problem with the UTS method, which requires  $\mathcal{O}(m\eta)^2$  operations for advancing the solution  $m$  times using a small time-step  $\Delta t$  for the entire problem domain, (where  $\eta = \eta_A + \eta_B - \eta_I$ ), the cost of solving the same

problem with the MTS method for a similar level of accuracy within subdomain B, is  $\mathcal{O}(\eta_A^2) + \mathcal{O}(m\eta_B^2)$ , which in general will be much smaller compared to  $\mathcal{O}(m\eta^2)$ .

### 2.3.4 Exact and reference solutions for computing errors

For most practical problems in structural dynamics, it is usually not possible to obtain exact solutions. However, when solving these problems numerically, it is useful to quantify the error in the solution by comparing it directly to an exact solution, when one is available. For small, linear, MDOF systems, modal analysis can be used to decompose the problem into a set of SDOF systems, and weighted response of each mode can be summed to get the response for the overall system. For larger problems, when modal analysis is not always feasible, a reference solution may be obtained by using a very small time-step for numerical time integration. In this work, both exact and reference solutions (with a time-step of  $\Delta t/20$ ) are obtained for the purpose of measuring errors in different MTS solutions.

For situations where an exact or a reference solution is not available, an error estimator may be used. Most error estimators in the literature are based on comparing the numerical solution at an instant  $t_n$  to a truncated Taylor series approximation [55]. Zienkiewicz and Xie [56] suggest to intuitively estimate average error in the solution at a node as the sum of the instantaneous error estimates, while Wiberg and Li [57] demonstrate that the sum of instantaneous error estimates in the energy norm provides a bound for average error estimates. Choi and Chung [58] state that the error in the energy norm is an adequate criterion on which to judge the validity of a solution. Adaptive methods, where the time-step is adjusted during integration, seek to have an instantaneous error that is uniformly distributed throughout the time history [58].

### 2.3.5 Error measures

Assuming that an exact or reference solution is available, or that a sufficiently accurate error estimator is available, one can define certain measures of error to evaluate the quality of the solution obtained by using different MTS decompositions.

The instantaneous local error is defined by the following expression:

$$\epsilon_n^i = \frac{|x_n^i - x_*^i(t_n)|}{(\text{MAX}_n(x_*^i) - \text{MIN}_n(x_*^i))} \quad (2.23)$$

where  $x_n^i$  and  $x_*^i(t_n)$  are the numerical and the exact solutions at a degree of freedom  $i$  at time  $t_n$  respectively. To normalize this measure, the difference between the numerical and the exact solutions is divided by the range, expressed as the difference between the maximum and minimum of the exact solution over the entire time history.

The average local error (average in time) is defined as:

$$\epsilon^i = \frac{1}{N} \sum_{n=0}^N \epsilon_n^i = \frac{\sum_{n=0}^N |x_n^i - x_*^i(t_n)|}{(\text{MAX}_n(x_*^i) - \text{MIN}_n(x_*^i)) * N} \quad (2.24)$$

where  $N$  is the total number of time-steps in the simulation. Finally, the average global ( $L^\infty$ ) error for a problem is defined as:

$$\epsilon = \text{MAX}_i(\epsilon^i) \quad (2.25)$$

where the maximum is taken over all of the degrees of freedom  $i$  in the model. This error metric was chosen to measure the error more precisely for problems where capturing the peak response immediately after an event (see Figure 2.1) may be more critical rather than its long-term dynamic response.

## 2.4 Characterization of Errors

In this section, numerical simulations using the multi-time-step method are conducted for three types of structures, a truss, a frame, and a continuum solid body, to investigate how different MTS decompositions affect errors and computational cost. For all problems, different MTS decomposition parameters, such as the sizes of subdomains A and B, the time-step ratio  $m = \Delta T / \Delta t$ , and the topology of the

interface are varied to construct and evaluate multiple different MTS solutions against an exact or a reference solution.

#### 2.4.1 MTS decomposition for a truss problem

Truss problems are particularly well-suited for studying errors in different MTS decompositions as they demonstrate the advantages of advanced numerical time integration techniques while maintaining a relatively simple spatial discretization. Figure 2.4 shows the truss chosen for error characterization in this study. The truss is pinned on its entire left edge with a rectangular pulse load of magnitude  $P = 1000$  lbs<sub>f</sub> (4448 N) applied as shown for a duration of 0.1 seconds. The truss properties are chosen so as to model a stiff, light-weight truss with high natural frequencies. Each truss member is assumed to be made of standard W-12x16 (height = 30 centimeters, weight = 233.5 Newtons per linear meter) rolled shape using A992 steel, with Young's modulus  $E = 29,000$  ksi (200 GPa) and density  $\rho = 0.28$  lb<sub>f</sub>/in<sup>3</sup> (7748 kg/m<sup>3</sup>). The model is assumed to have Rayleigh damping with mass proportionality factor  $\alpha = 0$  and stiffness proportionality factor  $\beta = 0.01$ . The total length of the cantilever truss  $L$  is 85 feet (25.9 meters), with a total height  $H$  of 10 feet (3.05 meters).

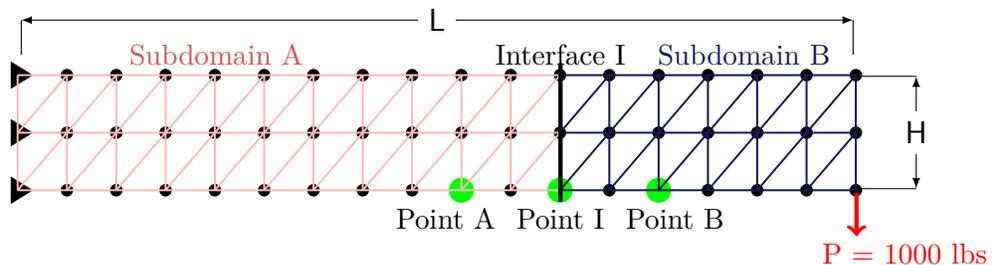


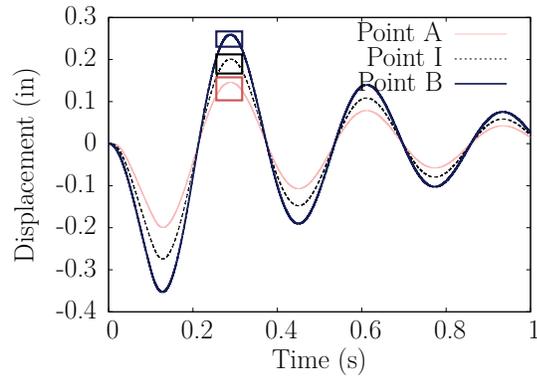
Figure 2.4.: A sample MTS decomposition of a truss problem into two subdomains with different time steps. Point A lies in Subdomain A, Point B lies in Subdomain B, and Point I lies on the coupling interface.

The problem was solved with three different approaches: (i) UTS method with a small time-step  $\Delta t = 2.5 \times 10^{-4}$  seconds, (ii) UTS method with a big time-step  $\Delta T = 10^{-3}$  seconds, and (iii) MTS method with the decomposition shown in Figure 2.4

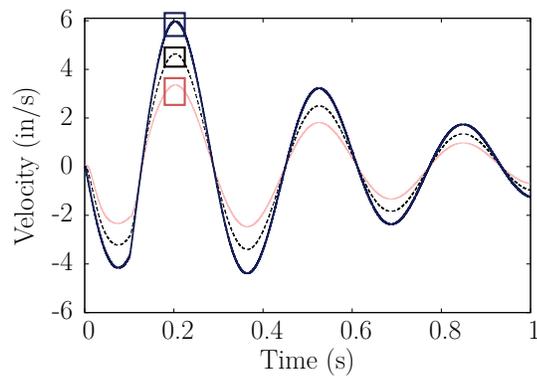
using  $\Delta T$  in subdomain A and  $\Delta t$  in subdomain B. For all these cases, the Newmark implicit integration method with parameters  $\gamma = \frac{1}{2}$  and  $\beta = \frac{1}{4}$  is used. Sample time histories of displacements, velocities and accelerations at points A, B and I as marked in Figure 2.4 are shown in Figure 2.5 for these three approaches. Detailed response within the marked boxes on these plots are shown in Figure 2.6 for each case.

For this problem, the rectangular pulse load  $P$  causes local transients in subdomain B that need to be captured with a small time-step  $\Delta t$ , whereas away from the point of application of this load, the time histories are smoother and can be simulated with a big time-step  $\Delta T$  without incurring big errors. For the coupling node I, Figure 2.6(a) shows that the response of the multi-time-step method, in general, tracks between the responses obtained by the UTS method with a small  $\Delta t$  and the UTS method with a large  $\Delta T$ . Note that consistent with Equation (2.10), the velocities of subdomains A and B match exactly on the interface node at every  $\Delta T$  interval for the MTS method. Similarly, inspecting Figure 2.6(b), for a node within Subdomain B, the multi-time-step solution is very close to the UTS  $\Delta t$  solution. This suggests that, despite the presence of the numerical interface that introduces errors into the solution for Subdomain B, the MTS method is able to capture the response of subdomain B very well. For node A within subdomain A, Figure 2.6(c) shows that the MTS solution is much closer to the solution of the UTS  $\Delta t$  solution, even though the MTS method actually uses the big time step  $\Delta T$  in subdomain A. This suggests that the MTS method not only matches the solution in Subdomain B well, but it also improves the solution for subdomain A in comparison to UTS  $\Delta T$ , thereby reducing both local and global errors throughout the problem domain.

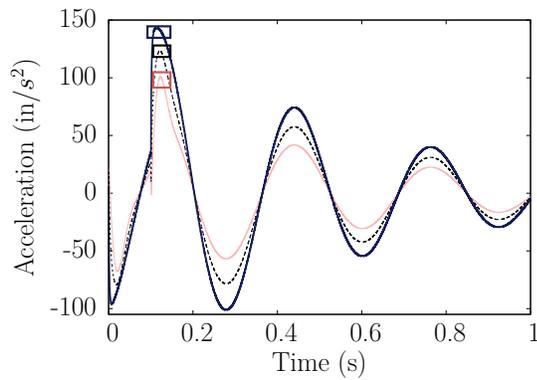
While Figures 2.5 and 2.6 provide valuable information, they show results only at specific degrees of freedom. Based on this information alone, it is difficult to judge how to decompose the truss problem into subdomains A and B, how big these subdomains need to be, what the time-step ratio between them should be and what effect does a particular choice of decomposition have on the computational cost and errors. In order to understand the effect of decomposition choices on errors, one needs to visualize the



(a) Displacement



(b) Velocity



(c) Acceleration

Figure 2.5.: Global time history response of nodes A, B, and I for the truss problem. The location of these points is shown in Figure 2.4.

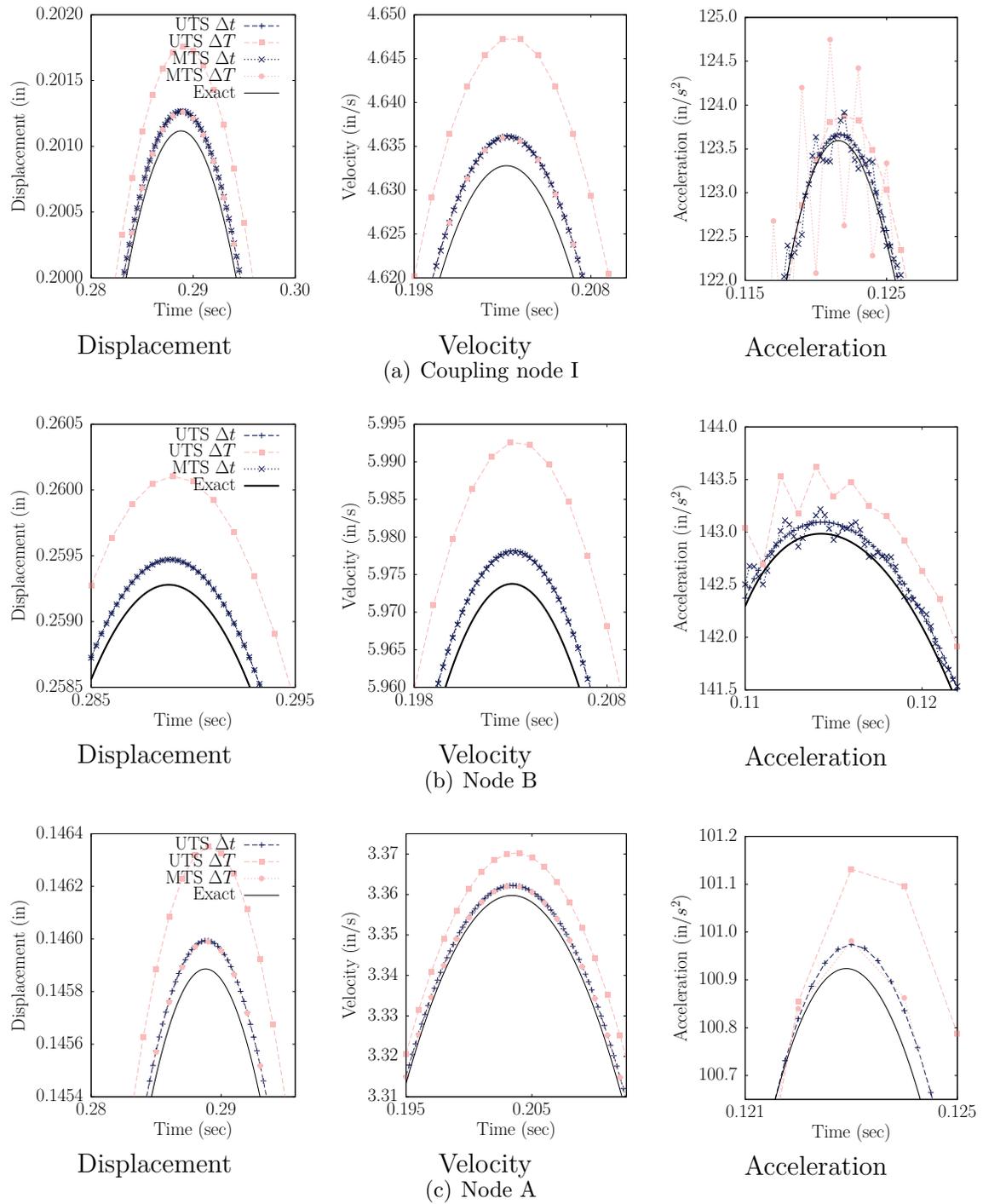


Figure 2.6.: Close up view of the time histories at nodes A, B, and I of the truss. The location of these points is shown in Figure 2.4.

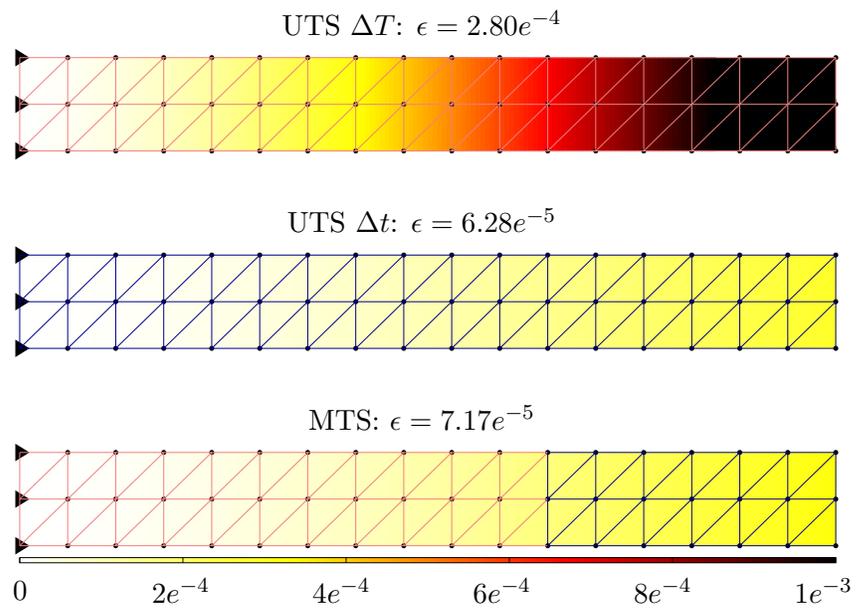


Figure 2.7.: Distribution of average local errors in nodal velocity at each degree of freedom for the truss problem. Average global error  $\epsilon$  for each case is also mentioned.

state of errors over the entire problem domain. For this purpose, error color maps are constructed by computing the average local errors from Equation (2.24) at each degree of freedom in the model, and its distribution over the entire problem domain is plotted as shown in Figure 2.7. Average global errors computed from Equation (2.25) corresponding to each case is also mentioned, clearly showing high errors for the UTS  $\Delta T$  method, whereas the local and global errors for the MTS method are similar to the errors for the UTS  $\Delta t$  method, but at a much lower computational cost (0.257 seconds for the MTS method compared to 0.936 seconds for the UTS  $\Delta t$  method and 0.234 seconds for the UTS  $\Delta T$  method). While only the average local error in nodal velocity (averaged for both  $x$  and  $y$  directions) is shown here, the same procedure can be used to obtain errors in displacements, accelerations, internal forces etc. Plots of error distribution in displacements and accelerations show similar trends as those observed in Figure 2.7.

#### 2.4.2 MTS decomposition for a frame problem

A nine-story frame structure described by [59] is used to repeat a similar analysis as that of the truss structure in the preceding section. The structure is shown in Figure 2.8 and is modeled with one frame element for each beam and column. Properties of steel ( $E = 29,000$  ksi (200 GPa),  $\rho = 0.28$  lb<sub>f</sub>/in<sup>3</sup> (7748 kg/m<sup>3</sup>)) are used for the material. The structure has a total height  $H = 122$  ft (37.2 m) and total length  $L = 150$  ft (45.7 m). Rayleigh damping is used, with mass proportionality factor  $\alpha = 1.2$  and stiffness proportionality factor  $\beta = 0.001$ , which is close to a modal damping of around 5% for each of the first ten modes. A high frequency harmonic loading is applied to the structure as shown in Figure 2.8. The function  $P(t)$  is given by:

$$P(t) = P_0 \times \sin(\omega t) \quad (2.26)$$

where the magnitude of the load  $P_0$  is  $10^{10}$  lbs ( $4.448 \times 10^{10}$  N) and the frequency  $\omega$  is  $1000Hz$ . The frame was decomposed into two parts, subdomain A solved with a large time-step  $\Delta T = 10^{-5}$  seconds and subdomain B solved with a time step of  $\Delta t = 2.5 \times 10^{-6}$ . Time histories and errors were computed as before and the error distribution graphs for this problem are shown in Figure 2.9.

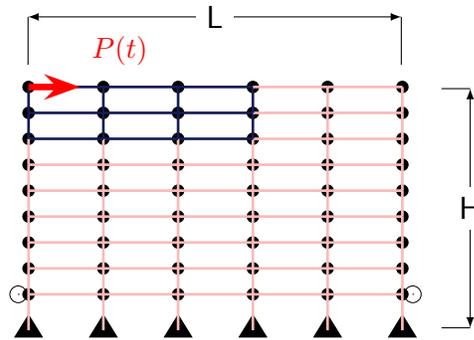


Figure 2.8.: Nine-story frame structure from ([59]). Subdomain A is represented with light lines, and Subdomain B represented with dark lines. Pinned boundary conditions in the basement are represented by triangles, and lateral boundary conditions at the ground level are represented by white circles.

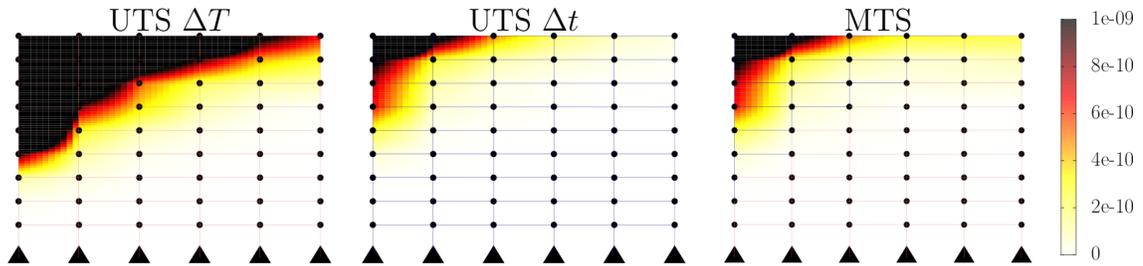


Figure 2.9.: Distribution of average local error in nodal velocity for the frame problem.

Figure 2.9 further corroborates the observations made in the preceding section for characterization of errors in MTS decompositions for truss problems. One may note that, once again, the error is concentrated around the point of loading and that the

UTS  $\Delta T$  error is much bigger than the UTS  $\Delta t$  error and the MTS solution error, both of which are in a similar range.

### 2.4.3 MTS decomposition for wave propagation in a solid domain

In this section, continuum elements are used to model the Lamb problem ([60], [61], [62], and [63]), which is a well studied benchmark problem in continuum mechanics. The problem is shown in Figure 2.10 along with a snapshot of the response after the P-wave has propagated to about half the domain length. The symmetric-half of a rectangular solid domain of dimensions  $2L \times H$  ( $L = H = 2$  m) is fixed on the right and bottom edges, has vertical rollers for the symmetric boundary condition on the left edge, and is unconstrained on the top edge. A downward load of magnitude  $P = 1000$  N is applied as a rectangular pulse starting at  $t = 0$  and ending at  $t = 1 \times 10^{-5}$  s on the top-left corner of the domain. The material properties chosen for this problem are, Young's Modulus  $E = 20$  GPa, Poisson's ratio  $\nu = 0.25$ , and the material density was  $\rho = 2000$  kg/m<sup>3</sup>, resulting in a P-wave speed of 3162 m/s.

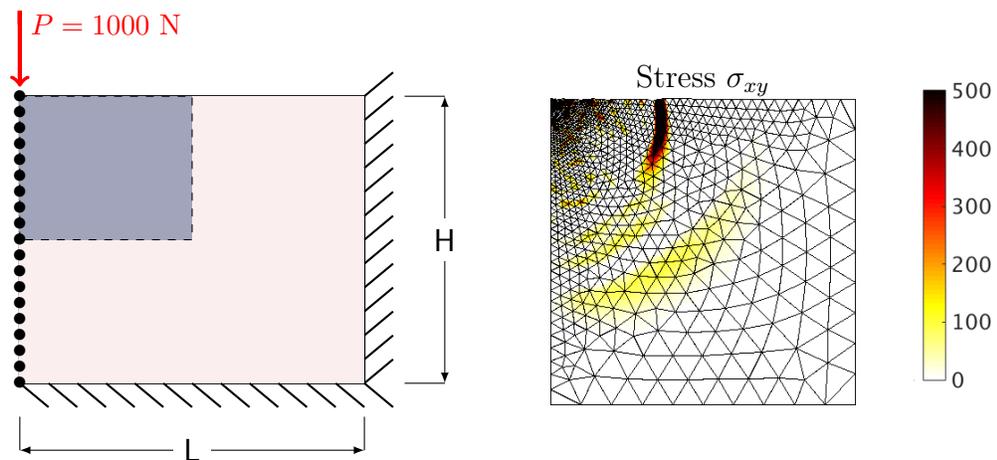


Figure 2.10.: Wave propagation in a plane-strain solid domain

The domain is meshed with 1700 3-noded constant-strain triangular (T3) continuum elements and the mesh is heavily refined near the loading point. For the UTS and MTS

simulations, time-steps of  $\Delta t = 5 \times 10^{-7}$  s and  $\Delta T = 2 \times 10^{-6}$  s are used. For MTS, elements with all three nodes in the top-left quadrant (shaded blue in Figure 2.10) are treated as subdomain B, integrated with  $\Delta t = 5 \times 10^{-7}$  using explicit central difference and the rest of the domain is treated as subdomain A, integrated with  $\Delta T = 2 \times 10^{-6}$  with the Newmark constant average acceleration method. The simulation is run for  $1 \times 10^{-4}$  seconds, which allows the P-wave to propagate a distance of about 0.3 m from the loading point towards the right and bottom edges of the domain, still remaining within the top-left quadrant of the problem domain. Ideally, for such dynamic wave propagation problems, subdomain B should be adaptively repositioned during the simulation to track the wave, a topic for future studies. For the current study, it is sufficient to ensure that the wave remains within subdomain B.

Figure 2.11 shows the distribution of average local errors for the three simulations described above. Errors are higher near the loading point and at the wave front, with the errors in the MTS solution similar in range to the errors in the UTS  $\Delta t$  solution, both of which are much smaller compared to the errors in the UTS in  $\Delta T$  solution. It is clear that this continuum problem also follows similar patterns of error distribution within the problem domain that were observed for the truss and frame problems.

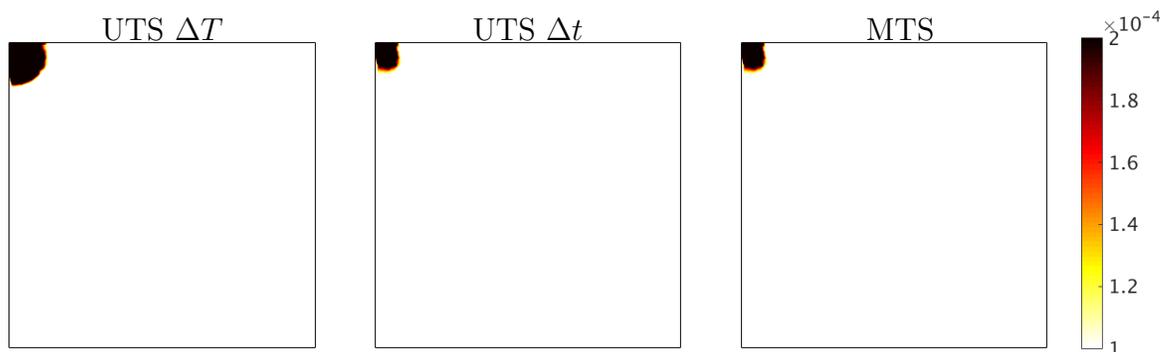


Figure 2.11.: Distribution of average local error in nodal displacement for the Lamb Problem

Having characterized the errors and the computational costs for MTS decompositions, it is now possible to map and explore the space of possible MTS decompositions by comparing their computational costs using the cost model developed in the pre-

vious section to find an almost optimal MTS decomposition for a given problem. A framework for finding such optimal decompositions is presented in the next section.

## 2.5 Space of Possible MTS Decompositions

In this section, the space of possible MTS decompositions is explored in terms of the number of ways a given problem domain can be decomposed into subdomains A and B and the number of possible different time-step combinations that can be used for these subdomains. Note that for a model with  $N$  elements, there are  $2^N$  possible decompositions. For example, for the small truss problem shown in Figure 2.4 with 121 elements, there are approximately  $2.6 \times 10^{36}$  possible decompositions. Further, since the time-steps for each subdomain may be chosen on a continuous scale, as long as they satisfy any stability criteria, there will be infinitely many possible time-step combinations for each of the  $2.6 \times 10^{36}$  decompositions. Clearly, the space of possible decompositions and associated time-steps is huge and even characterizing solution errors and computational costs is a monumental task, let alone finding an optimal decomposition in this space.

While all of the possible decompositions and time-step combinations chosen in the manner described above are valid choices corresponding to a particular numerical model, not all of them are computationally appealing. In order to traverse the space of possible decompositions within a reasonable range of parameters, we constrain the search space by making two key assumptions and characterize how the solution errors and computational costs vary for decompositions within this subset of the space. First, to limit the choices in terms of time-step sizes that can be used, it is assumed that the smallest time-step  $\Delta t$  that one may choose for a given problem would be determined by the highest modal frequency to be captured in the response. Similarly, the largest big time step  $\Delta T$  is constrained by the smallest natural frequency of the structure and this will determine the range of possible time-step ratios  $m$  in the search space. Second, for a particular MTS decomposition, each element is associated with

its average local error (computed from the average local errors of its nodes) and it is postulated that this MTS decomposition will be viable only if every element in subdomain A (with  $\Delta T$ ) is associated with a lower error than the maximum of the errors associated with all of the elements in subdomain B (with  $\Delta t$ ). This assumption ensures that subdomain B, which is likely to be the critical region of interest for most problems, controls the global error for the problem.

With the two assumptions above, one can traverse the space of potentially useful MTS decompositions in the following manner:

1. Solve the problem with the UTS method with time-steps ranging between  $\Delta t$  to  $\Delta T$  in integer multiples of  $\Delta t$  and record the *global* solution errors and computational costs for each case.
2. Solve the problem with different MTS decompositions, initially assigning all the elements to subdomain B and none to subdomain A (which is a valid decomposition, and is the same as the UTS method).
3. LOOP until all the elements are moved from subdomain B to subdomain A
  - Sort all the elements in the model by their average *local* errors.
  - Identify the element with the lowest error in subdomain B and assign it to subdomain A.
  - LOOP over values of  $m$  such that  $m\Delta t \leq \Delta T$  starting with  $m = 2$ 

Solve the problem using the particular MTS decomposition and time-steps  $m\Delta t$  and  $\Delta t$  for subdomains A and B respectively and record the global solution error and computational cost.
4. Plot the variation of global solution errors and computational costs with respect to number of elements in subdomain B and the time-step ratio used.

The steps above allow one to generate plots of average global errors and computational cost as shown in Figure 2.12(a) for the Truss problem described in Figure

2.4. Note that since different MTS decompositions advance the solution by different amounts of time in one big time-step  $\Delta T$ , the computational cost in Figure 2.12(b) has been normalized to express it as the amount of time needed to advance the solution by a small time-step  $\Delta t$ . This is computed by dividing the average run-time per  $\Delta T$  time-step for a given MTS decomposition by the time-step ratio  $m$  between its subdomains.

Starting from the bottom right corner (with all the elements in subdomain B) and at the smallest  $\Delta t$  i.e. time-step ratio  $m = 1$ , the steps above map out the performance of different MTS decompositions by moving from bottom to top and right to left. Figure 2.12 shows the average global errors and computational cost for the truss problem described in the previous section. It is clear that errors are smallest for MTS decompositions with most (or all) elements in subdomain B and run at  $\Delta t$  with a time-step ratio  $m = 1$ . On the other hand computational costs are lowest for decompositions with most (or all) elements in subdomain A and run at  $\Delta T$  with the maximum time step permissible. Here time step ratios up to 32 are explored but one may explore even higher time step ratios if needed.

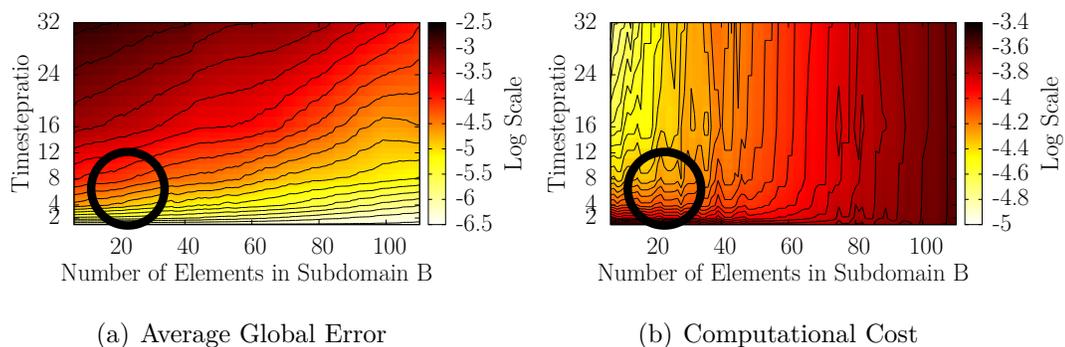


Figure 2.12.: Average global errors and computational cost for different MTS decompositions of the Truss Problem

### 2.5.1 Optimal MTS decomposition

Using the plots of average global errors and computational cost, one can easily determine a close to optimal decomposition for a specific problem and given criteria. For instance, if one needs to find the decomposition with the lowest computational cost for particular error threshold, then one can follow the contour of the specified level of error looking at the corresponding computational costs to find the optimal MTS decomposition. This optimal decomposition clearly defines which elements to choose for subdomains A and B and what time-step ratio to use between them. For instance, if one chooses to restrict global errors to  $\epsilon \leq 1 \times 10^{-5}$ , then following that error contour in Figure 2.12(a), and noting the computational cost of the corresponding point in Figure 2.12(b), it is possible to determine the particular MTS decomposition with the minimum computational cost for the given level of error, as denoted by the red circle on each sub-figure. In this case, the optimal decomposition is shown in Figure 2.13, which has 23 elements in subdomain B and has a computational cost of about  $1 \times 10^{-4}$  s per  $\Delta t$  time-step.

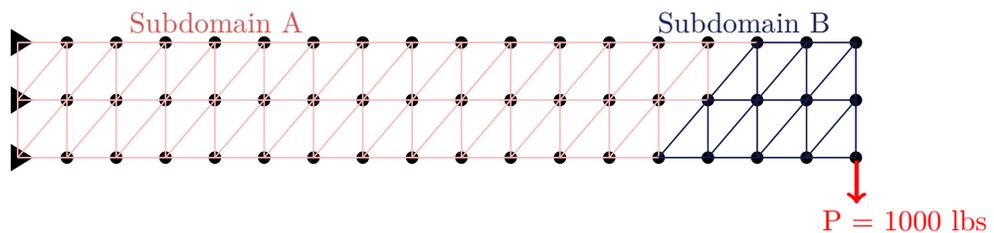


Figure 2.13.: Optimal decomposition of the truss problem.

While it can be very computationally intensive to generate such plots, the insight gained into the behavior of errors and cost by using such plots is invaluable. This insight can be used to design a method for finding optimal MTS decompositions without actually constructing these plots for the entire subset of the space of possible MTS decompositions. A possible way to find an optimal MTS decomposition is to start with all elements in subdomain B (UTS  $\Delta t$  method) and to use a gradient-based

or heuristics-based optimization algorithm to iteratively improve the decomposition during the simulation.

## 2.6 Numerical Examples

In this section, additional examples of how global solution errors and computational costs vary for different MTS decompositions are presented for the Frame problem (see Figure 2.8) and the Lamb wave propagation problem (see Figure 2.10).

### 2.6.1 Frame Problem

The nine-story frame problem (from Figure 2.8) is explored to investigate how errors and computational cost change with changing MTS decompositions. Figure 2.14 shows the global error and computational cost corresponding to the nine-story frame with the sinusoidal load as described in Figure 2.8, solved with implicit-implicit MTS decompositions. As a separate test problem, a second sinusoidal load is added to the nine-story frame on the right-top node, and the space of possible decompositions is regenerated. Note that elements are added to Subdomain A in a different orders, when loading conditions change. However, the overall trends in the space of MTS decompositions remain similar further validating this approach for finding optimal decompositions. Figure 2.15 shows that even for different load combinations, the behavior of errors and cost follows similar trends. Optimal decompositions for the two cases that restrict global solution error to  $\epsilon < 1 \times 10^{-5}$  are also denoted by red-circles and shown in Figure 2.16.

### 2.6.2 Lamb Problem

Finally, the Lamb problem (from Figure 2.10) is explored. Figure 2.17 shows the space of decompositions available for this problem. Note that the contour lines for the Lamb problem are considerably smoother than that of the truss and frame problems

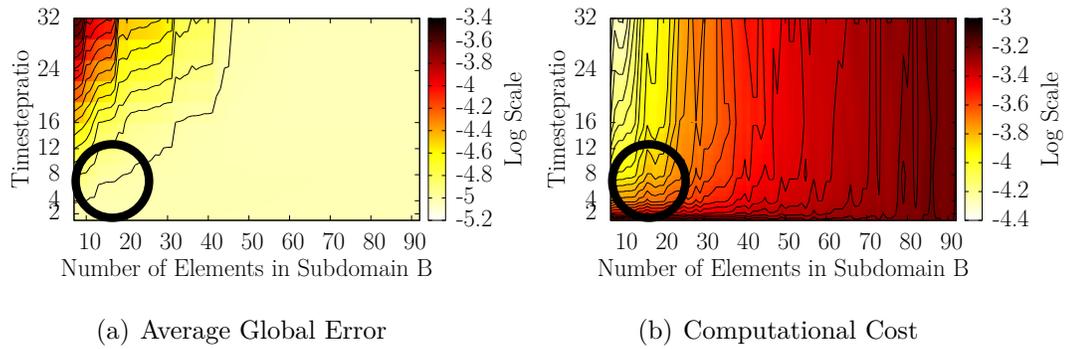


Figure 2.14.: Average global errors and computational cost for different MTS decompositions of the frame problem with one sinusoidal load

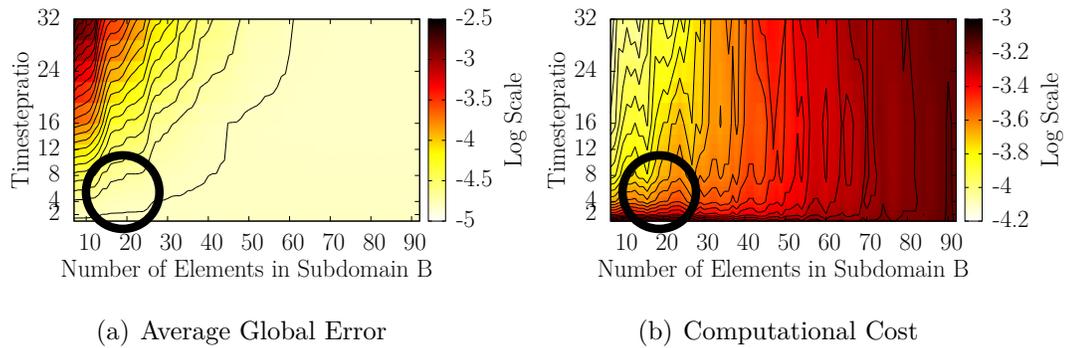


Figure 2.15.: Average global errors and computational cost for different MTS decompositions of the frame problem with two sinusoidal loads

since there are a large number of possible decompositions and, unlike the truss and frame problems, moving one element between subdomains A and B is less likely to cause significant changes to the computational cost or error in the problem.

An optimal decomposition for the Lamb problem that restricts global solution error to  $\epsilon < 1 \times 10^{-4.2}$  is denoted by the red-circles in Figure 2.17 and also depicted in Figure 2.18. Even though the maximum time-step ratio is limited to 12 for this problem, the contour plots of error and cost indicate that it may be possible to achieve even higher time-step ratios and lower computational cost for the same global error.

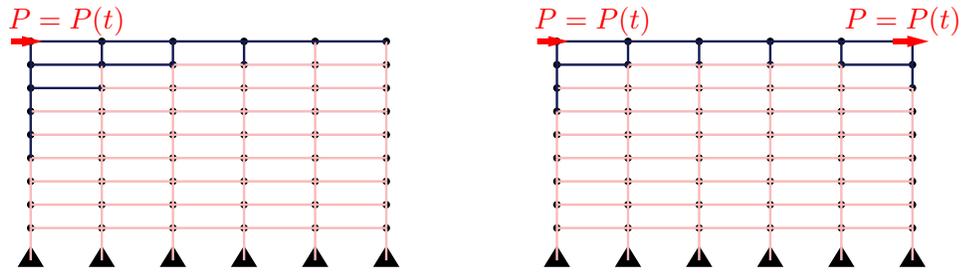


Figure 2.16.: Optimal Decompositions for the nine-story frame structure with one and two load points. Subdomain A (large time step  $\Delta T$ ) is shown with light lines, and Subdomain B (small time step  $\Delta t$ ) is shown with dark lines.

The optimal decomposition for this problem concentrates the subdomain B in a small region around the load point and allows the use of big time-steps for a majority of the domain. This behavior is expected and explained by the fact that the maximum errors in the solution occur during the application of the sharp rectangular pulse load. Once the wave starts propagating, it disperses and the wave front blunts leading to a less steeper wave and consequently lesser local errors. Thus even though some elements of subdomain A near the interface with subdomain B may encounter the wave, the error associated with them even for  $\Delta T$  is smaller than the error associated with the elements near the load point being solved at  $\Delta t$ .

### 2.6.3 Discussion and insights

It is interesting to compare the MTS decompositions of the truss, frame and continuum solid problems from the previous section on 'Characterizing Errors' to the optimal decompositions obtained in the results of this section. It can be observed that, while the specific MTS decompositions from the previous section seemed intuitive and also helped reduce computational cost in comparison to UTS  $\Delta t$  solution and improve

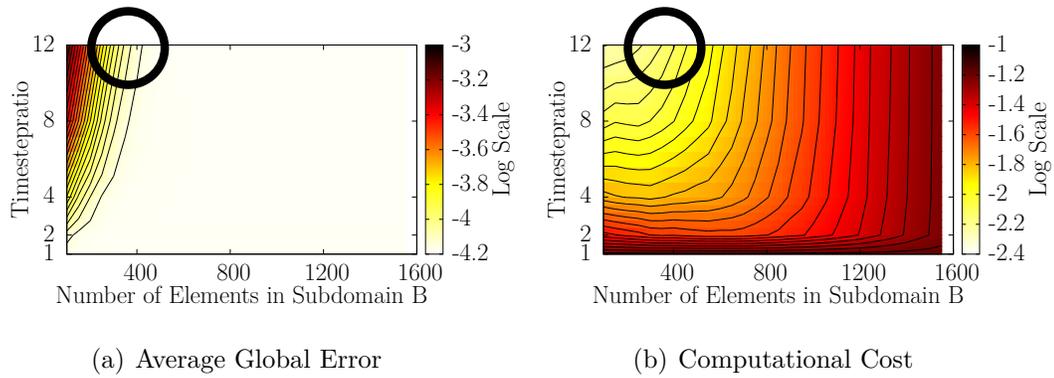


Figure 2.17.: Average global errors and computational cost for different MTS decompositions of the Lamb wave propagation problem

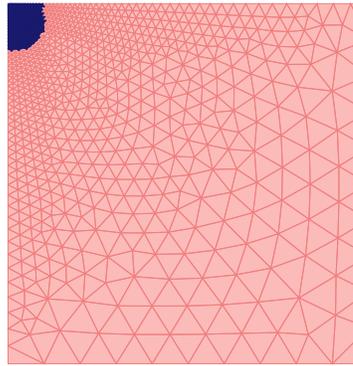


Figure 2.18.: Optimal Decomposition for Lamb Problem. Subdomain A (large time step  $\Delta T$ ) is shown with light coloring, and Subdomain B (small time step  $\Delta t$ ) is shown with dark coloring.

accuracy in comparison to UTS  $\Delta T$  solution, the optimal decompositions obtained here are better.

Further, from the topology of the contour plots of error and cost, it can also be observed that there are locations with steep gradients (where contour lines are closely spaced), indicating that the trade-off between cost and error does not vary uniformly and can change very suddenly for appropriately chosen decompositions. This fact suggests that, even if an informed user may pick an intuitively appealing

MTS decomposition for a specific problem, the likelihood of that decomposition being close to optimal is remote and there may be significant additional gains in accuracy and cost achievable by finding and using an optimal decomposition.

Overall, in all the problems studied, the space of MTS decompositions has a similar topology and exhibits similar patterns in terms of the variation of computational costs and errors. The variation is smoother for larger continuum solid problems than smaller discrete truss and frame problems but the trends are similar. This behavior is typical of problems containing one or two regions of interest with a temporally distinct feature that needs to be captured. With these insights it is possible to devise a method to find optimal decompositions without actually traversing the space of possible decompositions and to potentially modify these optimal decompositions during the simulation for problems with rapidly changing spatio-temporal characteristics. These will be topics for future research.

## 2.7 Conclusions

For problems that exhibit multiple scales of response in their dynamic behavior, the multi-time-step (MTS) method provides great computational efficiency without loss of accuracy. In this study, three different types of problems, a truss, a frame and a continuum solid body, are solved to demonstrate how local and global errors vary with different problem characteristics and solution methods. It has also been demonstrated that solutions obtained by using uniform time step (UTS) methods, irrespective of their time-steps tend to concentrate errors in specific regions of the problem, typically those with high temporal gradients. A novel aspect of this work is showing that MTS methods can be used effectively to lower computational cost while maintaining accuracy of the solution by distributing the errors more evenly across the problem domain. Another new insight gained from this study is that MTS not only helps in reducing the errors in regions of high interest or fast-rate dynamics, but also helps to reduce errors in regions where large time steps are used.

A challenging problem associated with the MTS approach is determining the decomposition that leads to an optimal solution in terms of computational costs and errors. In this work, for the first time, a systematic approach to traverse a subset of the infinitely vast space of possible decompositions has been developed to demonstrate how the computational cost and error depend upon the choice of the MTS decomposition. This approach allows one to characterize the errors and costs of a range of computationally viable decompositions and then easily identify an optimal decomposition that satisfies given error or cost criteria. The methods presented here can be used on different models and methods, though the contents of the plots themselves might change. Specifically, different problem sizes or multi-scale behavior may produce substantial differences in the error and cost curves. As a future extension of this work, it may be possible to develop a method to find an optimal MTS decomposition for a given problem during the simulation itself, without having to explore the subspace of decompositions as described above. While that is not the focus of this study, the insight gained from characterizing errors and costs for different MTS decompositions in identifying an optimal decomposition is crucial for designing such a method.

### 3. ASYNCHRONOUS MULTI-TIME-STEP COUPLING OF NUMERICAL AND PHYSICAL MODELS FOR HIGH-FIDELITY REAL-TIME HYBRID SIMULATION

Real-Time Hybrid Simulations (RTHS) in structural engineering are coupled numerical-physical experiments, where a structure is divided into numerical and physical substructures interacts in real-time. Numerical substructures are generally chosen as the subsections of the structure that are relatively well understood but too large to test in a lab, while physical substructures are less predictable, and can be tested in the lab. Real-time coupling of numerical and physical substructures allows evaluation of the structural dynamics of the full system, utilizing the advantages of both numerical and physical testing.

To preserve nonlinear rate-dependent behavior and dynamic properties in the physical substructure, the numerical substructure is evaluated in real-time. Conducting the coupled experiment necessitates that the numerical substructure also be solved in real-time, limiting the size of the numerical substructure. The goal of this work is to adapt a multi-time-step integration technique with asynchronous updates to allow for execution of larger numerical models in real-time hybrid simulation. A predictor is added to the multi-time-step integration scheme to minimize synchronization errors that arise in the real-time hybrid simulation implementation. The approach is demonstrated through a virtual RTHS experiment where the numerical substructure is a nine-story building and the physical substructure is a 200-kN magneto-rheological damper.

### 3.1 Introduction

Structural dynamics is studied using either physical testing or numerical modeling of structures and components under a set of load and boundary conditions to simulate and evaluate their performance in the real world. Testing can be conducted at various scales, ranging from small-scale tests on material samples, structural components and connections up to full-scale tests on complete structures. The results of these tests and simulations improve when the structural component is exposed to conditions that, as closely as possible, represent those when in the full structure.

Physical testing of large structures such as buildings and bridges at full scale is prohibitively expensive. Researchers have developed a hybrid simulation approach that augments physical specimens of structural components with numerical models of the remainder of the structure to simulate the response of the physical structural component within the system-level dynamics of the entire structure [64]. Figure 3.1 shows such an experiment. Here a damping device on the first floor of a building is modeled physically, while the remainder of the structure is modeled numerically. An actuation device is used to impose the proper numerical boundary on the physical substructure, and measure the response for the numerical substructure. For this example, a magneto-rheological (MR) damper was placed on the first floor of the building. During a dynamic loading condition, such as an earthquake, the damper exhibits nonlinear, rate-dependent behavior. Thus, the physical experiment must be run at real time to capture and understand the rate-dependent response, while the remainder of the structure is modeled numerically.

In Real-Time Hybrid Simulation (RTHS), the response of the numerical model is computed at run time, and must be completed before the physical system advances one numerical time-step [65]. Interaction between the numerical and physical substructures eliminates the possibility of computing all of the load steps before the test begins. These tests are intended to capture the structural dynamics and failure modes of the entire structure, so the testing frequency must be sufficiently large (and time step

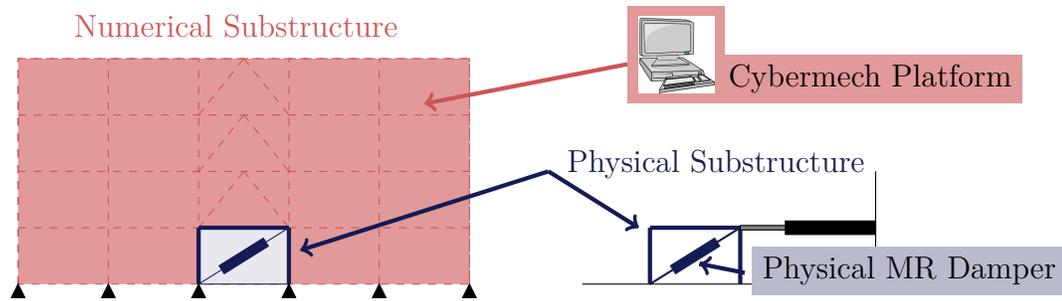


Figure 3.1.: An example of a Real-Time Hybrid Simulation Experiment

sufficiently small) to capture the response of the natural frequencies of interest and control the testing devices in a stable manner. Frequencies of up to 1024Hz (time step  $\Delta T = 0.977$  ms) are commonly used, limiting the size of the numerical model that can be solved in real-time. Typically the size and complexity of the numerical models used in RTHS are limited by the ability of the computer to solve these models within each time-step. Unfortunately, limiting the size of the numerical model can restrict the fidelity of the simulation [3].

A numerical technique that can be used to reduce computational cost in a numerical model, and thereby increase fidelity of RTHS, is multi-time-stepping (MTS). In MTS, the numerical domain is decomposed into sub-models which are allowed to run at different time scales [17]. Multi-time-stepping has been used to focus a numerical analysis on regions of interest structurally, such as connections or flaws in a structure. Solving only these critical regions at small time-steps, and the rest of the structural model with relatively large time-steps, reduces the computation cost of the analysis while still maintaining accuracy.

There are several methods for multi-time-stepping available in the literature. Gravouil and Combescure [66] show that various methods of the Newmark family can be coupled with different time steps in each subdomain. MTS integration schemes have been developed by Belytschko *et al.* [11], Smolinski [12], Daniel [13], Gravouil and Combescure [14, 15], and Prakash and Hjelmstad [16, 17]. Each of these papers provides an integration scheme for solving subdomains with different time-step. Bursi

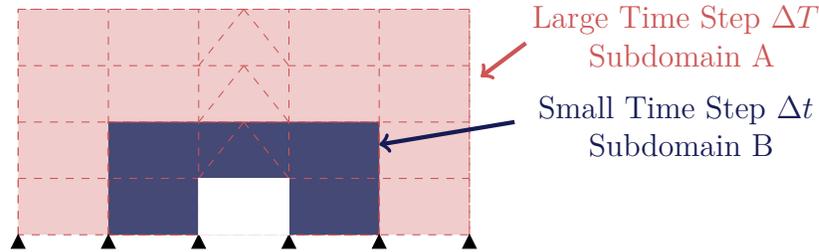


Figure 3.2.: Multi-time-step Decomposition of a Numerical Model

and co-workers [67, 68] use a multi-rate approach for a 2-DOF pseudo dynamics tests where the numerical substructure is solved at one rate, and the physical substructure is solved at another, and show a detailed convergence analysis. Bursi's approach solves the entire numerical substructure at the same time step, but allows it to be slower than the physical substructure.

This work focuses on applying MTS to RTHS, which we call (MTS+RTHS). Specifically, using a predictor for the MTS coupling to reduce the errors incurred by applying asynchronous updates in the physical portion of the model. An in-depth study is conducted investigating a series of choices of the predictor, and the effect of predictor choice on the error in both the physical and numerical substructures. It is demonstrated that MTS+RTHS is not only viable, but also essential when using high-fidelity models in RTHS.

### 3.2 Real Time Hybrid Simulation (RTHS)

RTHS has been demonstrated successfully in a number of studies, for example [65, 69–71]. The goal of RTHS is to provide the most realistic conditions possible, and thus one must consider the influence of both the physical test procedures and computational limitations. To examine the fidelity of RTHS, one must thus consider possible sources of error to minimize those. Errors in RTHS can come from various sources, including the numerical model, the numerical integration method, the testing control and compensation system, and noise in the sensors [71]. These errors can

propagate through a RTHS because the response of each time step is computed from the previous time step, and can lead to instability of the system [72]. Numerical errors can be reduced by increasing the complexity of the numerical substructure, but this will incur a cost of computational time. The available computational time must be shared between all aspects of the test to minimize the total error of the system and prevent instability.

The finite element method is the preferred method for the numerical modeling of physical systems. In the field of RTHS, however, using full-scale finite element models has been challenging because of the large associated computational costs [3]. Many numerical models in past RTHS involve single-degree-of-freedom (SDOF) models of structures [73], simple shear models [74], or other reduced-order models so that the real-time constraints can be met and transfer system controllers can be designed that are stable when implemented on the physical system [75, 76]. Large finite element models that fail to meet real-time constraints can contribute to instabilities in the system if not properly dealt with.

Literature in the field of RTHS covers a variety of approaches for meeting the real-time constraints. Some researchers choose to focus on compensation and control schemes, and do not address the use of large numerical models in their experiments [77–82]. Others focus on the numerical integration algorithms, but still do not include numerical substructures with more than 100 degrees of freedom [83–85]. Some researchers do use large numerical models in RTHS. Maghareh *et al.* [4] provide a detailed analysis of how increasing the number of degrees of freedom in the model can reduce error in an RTHS experiment. Chen and Ricles [86] use modal decomposition, and focus on the low frequency modes of the system. Alternatively, multi-rate approaches can be used, which solve the numerical model at a larger time-step, and interpolates and/or extrapolates displacement signals to send to the control loop. Nakashima and Masayoshi [87] explore this technique, but with a model limited to 10-12 degrees of freedom (DOFs). Other variations of these multi-rate approaches include [5, 75, 88–90]. Alternatively, Saouma and Haussman [3, 7] implement a parallel

finite element code, and increase the number of processors and available computing power to solve models as large as 400 DOFs. Maghareh *et al.* [75] developed an adaptive multi-rate interface to enable rate-transition between two models executed at different computational rates. While available computing resources have increased according to Moore’s law with time, the challenge of increasing the size of the numerical models still remains. Real-time constraints limit the size of the numerical system, and numerical techniques that decrease computation become more valuable as available computing power increases.

Multi-time-stepping provides unique advantages when used in conjunction with RTHS. In addition to the direct computational cost savings provided by the technique, MTS can be used to relax the timing constraints. Specifically, the regions of the numerical substructure that interact with the physical substructure directly, such as the boundary regions, can be solved at the same rate as the physical substructure, and the remainder of the numerical substructure can be solved at a lower rate of execution. This approach allows for more complex numerical substructures to be modeled with high fidelity while still meeting real-time constraints at the boundary points. Figure 3.3 shows an experiment that utilizes both RTHS and MTS, where everything within the fast ( $\Delta t$ ) subdomain is solved at the same rate as the physical substructure.

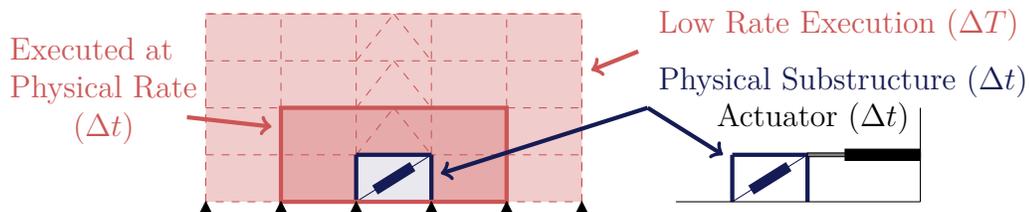


Figure 3.3.: Multi Time Step Decomposition in RTHS

As with any integration scheme or experimental technique, it is important to understand how error propagates through the system. This is particularly important in RTHS as large errors during the execution of the experiment can produce unexpected results, with the possibility of damaging the test specimen or testing equipment. This work investigates the potential for using MTS+RTHS, looking directly at the cost and

the error associated with RTHS modeling choices and MTS decomposition decisions. Additionally, this work addresses algorithmic changes to multi-time-step methods which are needed to maintain computational stability and reduce error when used with RTHS.

### 3.3 Asynchronous multi-time-step (MTS) coupling of numerical and physical models

#### 3.3.1 Conventional RTHS

RTHS is performed in a real-time computing environment, which allows the computer to devote all of its computational power to the problem at hand. A commonly used real-time environment is Matlab's xPC [91], that runs within the Matlab Simulink toolbox. Another environment is the Cybermech [10] platform, which allows support for parallel computations.

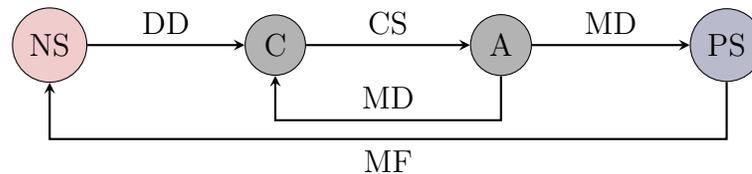


Figure 3.4.: RTHS Control Flow Schematic

A typical RTHS control loop is shown in Figure 3.4. Here *NS* and *PS* denote the numerical and physical substructures respectively, *C* denotes the compensation and control loop, and *A* denotes the actuator. *DD* and *MD* denote the desired and measured displacements, respectively, *CS* represents the command signal sent to the actuator, and *MF* denotes the measured force from the physical substructure. Figure 3.5 shows an idealized version of how the system advances through time, omitting the compensation and control steps. Here *EOM* represents solving the equations of motion for the numerical substructure once.

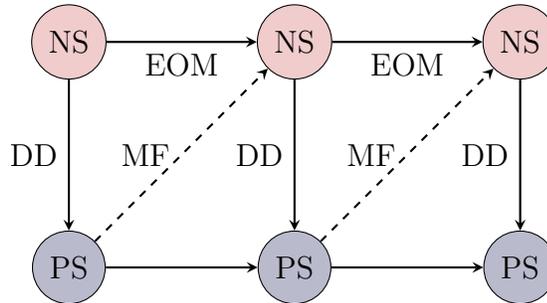


Figure 3.5.: RTHS order of computations and physical execution

The equations of motion for the numerical substructure can be expressed compactly as:

$$\mathbb{M} z_{n+1} = \mathbb{P}_{n+1}^P + \mathbb{P}_{n+1}^N - \mathbb{N} z_n \quad (3.1)$$

where  $z_n$  denote the states of the system (displacement, velocity, and acceleration at each numerical DOF) at time  $t_n$ ,  $\mathbb{P}_{n+1}^N$  denotes the numerical block external load vector consisting of the discrete load vector at time  $t_{n+1}$ , and  $\mathbb{P}_{n+1}^P$  denotes the block external load vector coming directly from the physical substructure. The block matrices  $\mathbb{M}$  and  $\mathbb{N}$  contain the finite element matrices of the structure and the Newmark integration parameters. The full formulation for this notation is available in [16, 92]. These equations can be advanced one step forward in time from  $t_n$  to  $t_{n+1}$  by solving Equation 3.1 once.

The physical substructure is advanced forward in time by sending the command signal to an actuator, which drives the virtual physical portion of the experiment. The desired displacement is enforced at each control point on the physical substructure, and the force in the actuator to generate this displacement is measured and applied to the numerical substructure [69]. Typically an actuator is controlled by a voltage signal, which controls a valve opening to a source of pressurized hydraulic oil. The actual (or measured) displacement is compared to the desired displacement, and a feedback loop is used to correct any differences [93]. Compensation and control techniques convert the desired displacement of the actuator to a command voltage signal. Outer loop

actuator controllers (compensators) are often included to deal with actuator dynamics and enforce the boundary conditions [2, 69, 74, 77, 94].

### 3.3.2 MTS method for numerical substructure

#### MTS Equations of Motion

Usually it is unnecessary to solve the entire numerical substructure at the smallest time step ( $\Delta t$ ) associated with the problem. Specifically, the numerical domain  $N$ , can be decomposed into two subdomains  $N^A$  and  $N^B$ , that are allowed to run at time-steps  $\Delta T$  and  $\Delta t$  respectively, where  $\Delta T = m\Delta t$  for some integer time-step ratio  $m$  as shown in Figure 3.6. The two subdomains may also have different Newmark time-integration parameters  $\gamma$  and  $\beta$  corresponding to different implicit or explicit schemes.

Given the state of subdomains A and B at  $t_0$ , multi-time-stepping is used to solve the state at  $t_m = t_0 + m\Delta t$  by advancing subdomain A through one big time-step  $\Delta T$  and advancing subdomain B through  $m$  small time-steps  $\Delta t$ . Lagrange multipliers,  $\lambda$  are then used to compute coupling forces between the subdomains, ensuring that the governing equations are satisfied. Figure 3.6 shows the flow of information for MTS.

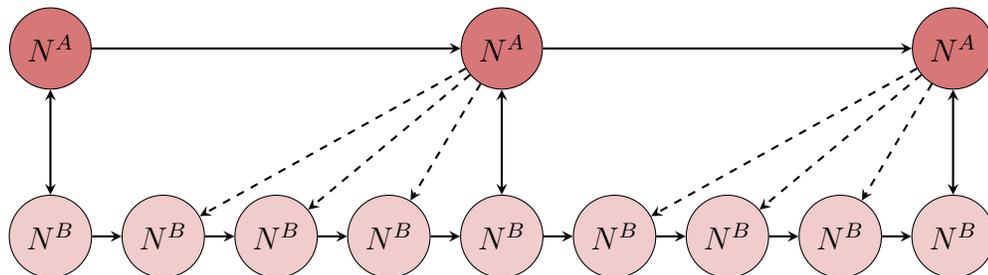


Figure 3.6.: MTS Information Flow Schematic

The equation of motion for subdomain A is solved using the large time-step  $\Delta T$ :

$$\mathbb{M}^A \mathbb{z}_m^A + \mathbb{C}^A \lambda_m = \mathbb{P}_m^A - \mathbb{N}^A \mathbb{z}_0^A \quad (3.2)$$

where  $\boldsymbol{\lambda}$  denotes the Lagrange multipliers on the interface and the matrix  $\mathbb{C}^A$  represents a mapping between the degrees of freedom on the interface to the degrees of freedom for subdomain A (see [16] for details). The equation of motion for subdomain B is solved  $m$  times using small time-step  $\Delta t$ :

$$\mathbb{M}^B \mathbb{z}_j^B + \mathbb{C}^B \boldsymbol{\lambda}_j = \mathbb{p}_j^B - \mathbb{N}^B \mathbb{z}_{j-1}^B \quad \forall j \in [1, m] \quad (3.3)$$

where the matrix  $\mathbb{C}^B$  represents a similar mapping between the degrees of freedom on the interface to the degrees of freedom for subdomain B. The load vector  $\mathbb{p}_j^B$  is composed of both the numerical  $\mathbb{p}_j^{B[N]}$  and physical  $\mathbb{p}_j^{B[P]}$  contributions. In order to compute Lagrange multipliers  $\boldsymbol{\lambda}_j$  at the intermediate time-steps  $t_j$ , a balance of tractions at the interface is enforced:

$$\mathbb{C}^{AT} [\mathbb{M}^A \mathbb{z}_j^A - \mathbb{p}_j^A + \mathbb{C}^A \boldsymbol{\lambda}_j] = \mathbf{0} \quad \forall j \in [1, m] \quad (3.4)$$

### Computational Solution Steps

Here, an overview of the solution procedure presented in [92] is given. On each subdomain, the solution is split into two parts,  $\bar{\mathbb{z}}$  and  $\mathbb{z}'$  as follows:

$$\mathbb{z}^A = \bar{\mathbb{z}}^A + \mathbb{z}'^A \quad ; \quad \mathbb{z}^B = \bar{\mathbb{z}}^B + \mathbb{z}'^B \quad (3.5)$$

where  $\bar{\mathbb{z}}$  that satisfies the uncoupled set of governing equations (in the absence of  $\boldsymbol{\lambda}$ ), and  $\mathbb{z}'$  which accounts for the effect of the coupling Lagrange multipliers. Using this approach the solution of the coupled subdomains A and B must be achieved in three steps: (i) uncoupled subdomain solve, (ii) interface solve, (iii) subdomain updates. These three steps are outlined below.

**Uncoupled Subdomain Solution** First, the uncoupled solutions for subdomains A and B are obtained as:

$$\bar{z}_m^A = [\mathbf{M}^A]^{-1} [\mathbf{P}_m^A - \mathbf{N}^A z_0^A] \quad \text{and} \quad (3.6)$$

$$\bar{z}_j^B = [\mathbf{M}^B]^{-1} [\mathbf{P}_j^B - \mathbf{N}^B \bar{z}_{j-1}^B - \mathbf{C}^B \mathbf{s}_j] \quad \forall j \in [1, m] \quad (3.7)$$

where  $\bar{z}_0^B = z_0^B$  and  $\mathbf{s}_j$  represents a predicted value of the coupling Lagrange multipliers.

$$\mathbf{s}_j = (1 - j/m)\boldsymbol{\lambda}_0 \quad (3.8)$$

**Interface Solve** Once the uncoupled solutions have been obtained, the coupling Lagrange multipliers are solved using the equation:

$$\boldsymbol{\lambda}_m = \mathbf{H}^{-1} \mathbf{f} \quad (3.9)$$

where the interface matrix  $\mathbf{H}$  is defined as:

$$\mathbf{H} = \mathbb{B}^A \mathbb{Y}_m^A + \mathbb{B}^B \mathbb{Y}_m^B \quad (3.10)$$

and the right hand side vector  $\mathbf{f}$  is given by the expression:

$$\mathbf{f} = \mathbb{B}^A \bar{z}_m^A + \mathbb{B}^B \bar{z}_m^B \quad (3.11)$$

In the above equations, the matrices  $\mathbb{Y}_m^A$  and  $\mathbb{Y}_m^B$  are obtained from the following expressions:

$$\mathbb{Y}_m^A = [\mathbf{M}^A]^{-1} \mathbf{C}^{AT} \quad \text{and} \quad (3.12)$$

$$\mathbb{Y}_j^B = [\mathbf{M}^B]^{-1} \left[ \left( \frac{j}{m} \mathbf{C}^{BT} \right) - \mathbf{N}^B \mathbb{Y}_{j-1}^B \right] \quad \forall j \in [1, m] \quad (3.13)$$

respectively, where  $\mathbb{Y}_0^B = 0$ . Note that the  $\mathbb{Y}$  matrices remain constant (for linear problems) for all time-steps and therefore they can be computed prior to beginning the time-stepping loop. Consequently, the interface matrix  $\mathbf{H}$  also remains constant for the duration of the time-stepping and it too can be computed, factorized and stored ahead of time for greater computational efficiency.

**Subdomain Updates** Finally, having computed the coupling Lagrange multipliers, the subdomain solutions can be updated using following contributions:

$$\mathbb{z}'_m{}^A = -\mathbb{Y}_m^A \boldsymbol{\lambda}_m \quad (3.14)$$

$$\mathbb{z}'_j{}^B = -\mathbb{Y}_j^B \boldsymbol{\lambda}_m \quad \forall j \in [1, m] \quad (3.15)$$

Figure 3.7 shows a schematic of the MTS+RTHS scheme, and highlights an inconsistency in the method. Arrows that go from right to left are going backwards in time, and cannot be implemented in MTS+RTHS. While updating the past solutions for subdomain B is not a concern in a pure numerical simulation, this presents a problem in MTS+RTHS, as the physical substructure has already progressed to the next time step, and cannot be updated retroactively. Without the correct value of the coupling terms, solving the *free* form of Subdomain B will introduce error into the physical substructure. This work introduces a predictor for  $\lambda_0$  to minimize the error caused by *synchronization* updates. The predictor is only applied to the desired displacement, maintaining all of the stability and convergence properties of the MTS method, while still producing an accurate desired displacement for the actuator.

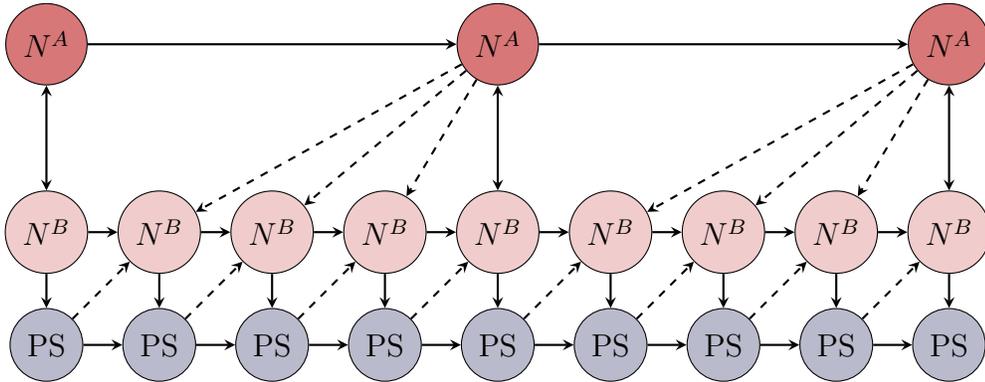


Figure 3.7.: RTHS Control Flow Schematic with no prediction scheme

### 3.3.3 Asynchronous predictor for MTS coupling of numerical and physical models

To overcome the inconsistency of MTS+RTHS, a general form of the predictor used for  $\lambda_m$  is:

$$\lambda_{n+m}^P = \alpha_0 \lambda_n - \alpha_1 \Delta T \dot{\lambda}_n + \alpha_2 \Delta T^2 \ddot{\lambda}_n \quad (3.16)$$

where  $\lambda_n$  is known at the current step,  $\dot{\lambda}_n$  is approximated as

$$\dot{\lambda}_n = \frac{\lambda_n - \lambda_{n-m}}{\Delta T} \quad (3.17)$$

and  $\ddot{\lambda}_n$  is approximated as

$$\ddot{\lambda}_n = \frac{\dot{\lambda}_n - \dot{\lambda}_{n-m}}{\Delta T}. \quad (3.18)$$

Here  $\lambda_n$  represents a zeroth order,  $\dot{\lambda}_n$  represents a first order, and  $\ddot{\lambda}_n$  represents a second order approximation of the coupling forces. Four prediction schemes are explored in Section 3.5.4 of this work. The prediction schemes differ in the choices of  $\alpha_0$ ,  $\alpha_1$ , and  $\alpha_2$ , and are shown in Table 3.1.

Table 3.1: List of MTS+RTHS Prediction Schemes

Scheme	$\alpha_0$	$\alpha_1$	$\alpha_2$
No Prediction (Scheme #1)	0	0	0
Zeroth Order Prediction (Scheme #2)	1	0	0
First Order Prediction (Scheme #3)	1	0.5	0
Second Order Prediction (Scheme #4)	1	0.5	0.25

An idealized version, of the implementation of MTS+RTHS with predictors is shown in Figure 3.8. Compensation and control is implemented between the numerical and physical substructures, though not shown in Figure 3.8. Note that no corrector is used, as the numerical substructures are still synchronized at each large time step, or at every  $m$  small time steps. The predictor is only used when communicating with

the physical experiment. Each update is applied as predicted, so information now always flows forward in time.

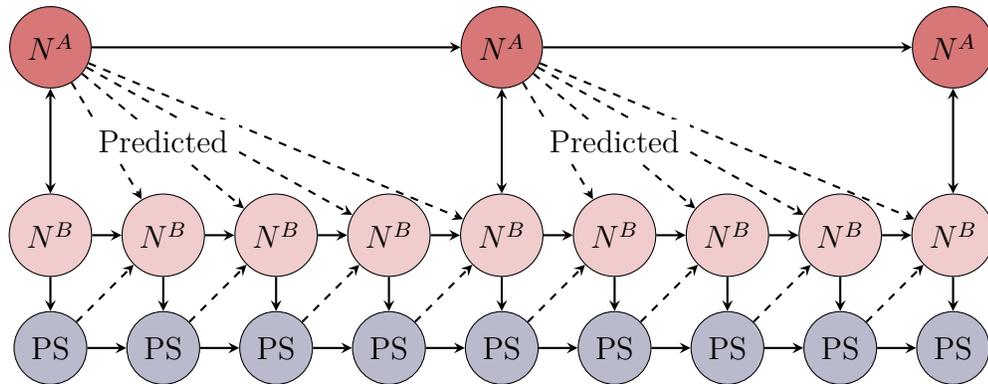


Figure 3.8.: Idealized RTHS Control Flow Schematic with prediction scheme

### 3.3.4 Computational cost of MTS+RTHS

The design and implementation of a successful RTHS experiment requires accurate knowledge of how the model will perform, both physical and numerically. Computational time is shared between numerical computations and control algorithms, and numerical solve time affects the time delay during the test. A large numerical model that cannot be solved within the real-time limits is not helpful. It is important to understand the computational limitations of the testing apparatus, and to design adequately scaled numerical models to match the capabilities of the test platform.

A cost model for the numerical substructure allows the engineer to correlate numerical model size, and time step ratio to real computational time, which can then be assessed from a control/time delay standpoint. Using this information, an engineer can subdivide the numerical model appropriately, and select an ideal time-step ratio, balancing computational time against error in the numerical integration.

The computational cost of the multi-time-step method was developed in [92]. For the three steps in the MTS solution algorithm, a basic estimate of the computational cost of each step can be obtained as:

1. Uncoupled subdomain solve:  $\mathcal{O}(\eta_A^2) + \mathcal{O}(m\eta_B^2)$ .
2. Interface solve are:  $\mathcal{O}(\eta_I)^2$  operations.
3. Applying the subdomain updates are:  $\mathcal{O}(\eta_A\eta_I + m\eta_B\eta_I)$ .

where  $(\eta_A)$  denotes the number of DOFs in subdomain A,  $\eta_B$  denotes the number of DOFs in subdomain B, and  $\eta_I$  denotes the number of DOFs on the MTS interface. Note that the most critical components of the cost is the size of each subdomain, *i.e.* the number of degrees of freedom in both subdomain A and subdomain B.

For this section, we will assume that a RTHS experiment can be run at 1024-Hz (0.977 sec) on a particular computer with at most  $D$  degrees of freedom. For parallel machines with  $N$  cores, each computational core has the capability to solve  $D/N$  degrees of freedom. If a larger uniform time-step is selected  $\Delta T = m * 0.977$  sec, each core can solve alternatively solve  $m * D/N$  DOFs. Thus, for MTS+RTHS, if the numerical substructure degrees of freedom are decomposed into  $D^A + D^B$ , then  $m * D^A + D^B$  DOFs can still be computed in 0.977 sec, effectively increasing the size of the model by about  $(m - 1) * D^A$  DOFs. While this is a simplification of the computational cost of the system, Bunting *et al.* [92] show a detailed analysis of the computational cost for the MTS method, and demonstrates the cost savings provided by the method. The higher the value of  $m$  and the higher the number of degrees of freedom in A, the larger is the size of the finite element model that can be in a RTHS experiment. It is still important to select a large time step  $\Delta T$  (and thereby  $m$ ) that solves subdomain A to within an acceptable amount of error.

### 3.4 Implementation of asynchronous MTS coupling on CyberMech - a novel RTHS platform

CyberMech is a RTHS software platform based on a federated scheduling model that allows parallelization to meet real-time constraints [9, 10]. Parallel execution

allows larger, more realistic numerical models, as well as advanced computational techniques to meet real-time constraints.

### 3.4.1 Taskset

This experiment executes two parallel tasks concurrently. The *slow* task is executed every  $m$  time steps, and has  $\Delta T$  seconds to complete. The *fast* task is executed every time step, and has  $\Delta t$  seconds to complete. These tasks are separate Linux processes, and data is shared between them with a shared memory model.

### 3.4.2 Shared Memory Model

The shared memory model creates an array in memory that is accessible by either task via Linux's shared memory inter-process communication. To prevent race conditions, only one task is allowed to read or write to the shared memory space at any given time. Two such shared memory spaces are employed. The first facilitates passing data from the *slow* process to the *fast* process. For this shared memory space, the *slow* task is only allowed to write to the memory, and the *fast* process is only allowed to read from the memory, and only after the *slow* process has finished updating. The second facilitates passing data from the *fast* process to the *slow* process. For the second shared memory space, the *fast* task is only allowed to write to the memory, and the *slow* task is only allowed to read from the memory, and only after the *fast* process has finished updating.

## 3.5 Results

In this section, a virtual RTHS experiment is performed on a nine-story frame structure, described in [59]. The performance of two numerical models are evaluated, a 184-DOF finite element model and a 9-DOF shear model, under an earthquake

loading condition. A 200 kN magneto-rheological (MR) damper is placed on the first floor of the building.

### 3.5.1 Virtual Real-time hybrid simulation of a nine-story frame

#### Numerical Substructure

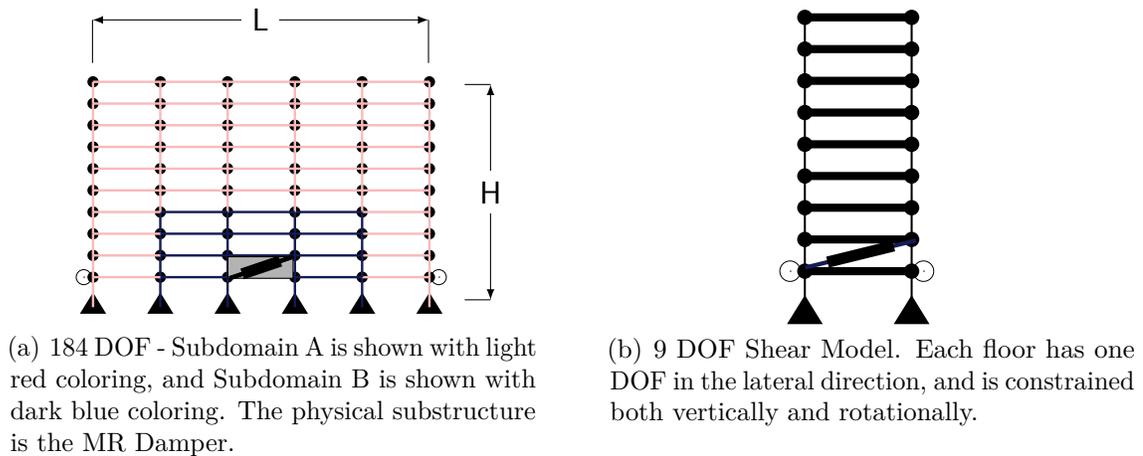


Figure 3.9.: Nine-story frame structure from [59]. Pinned boundary conditions in the basement are represented by triangles, and lateral boundary conditions at the ground level are represented by white circles.

The reference structure is a nine story frame with a 200 kN MR damper on the first floor, shown in Figure 3.9. The numerical substructure is the nine-story frame structure described in [59]. The structure is modeled with one frame element for each beam and column. Properties of steel ( $E = 29,000$  ksi,  $\rho = 0.28$  lb<sub>f</sub>/in<sup>3</sup>) are used for the material. The structure has a total height  $H = 122$  ft and total length  $L = 150$  ft. Rayleigh damping is used [29], with mass proportionality factor  $\alpha = 1.2$  and stiffness proportionality factor  $\beta = 0.001$ . These values provide damping of approximately modal damping of 5% for each of the first ten modes. Inertial loading of the 1940 El Centro earthquake record at a magnitude of 10% is applied to the structure.

As shown in Figure 3.9(a), the 184 DOF frame is decomposed into two parts. Subdomain B, which communicates with the physical substructure, is solved at 1024-Hz. Subdomain A is solved at a range of frequencies: 1024-Hz ( $m = 1$ ), 512-Hz ( $m = 2$ ), 256-Hz ( $m = 4$ ), 128-Hz ( $m = 8$ ), 64-Hz ( $m = 16$ ), 32-Hz ( $m = 32$ ), and 16-Hz ( $m = 64$ ). The error distribution for each frequency is presented.

As an alternative, Figure 3.9(b) shows a shear model of the frame structure. The model is a reduced version of the 184-DOF model, and only allows for translation in the lateral direction at each floor. The model was constructed using the same column elements as the 184-DOF model, but limiting the degrees of freedom to only the translational DOFs in the lateral direction. Both the basement and ground floor of the model were completely constrained as was the rotations and vertical displacements at each floor in the model. In the 184-DOF model, only the outer columns of the first floor are constrained in the lateral direction. Raleigh damping on the mass was used, with  $\alpha = 1.2$  and  $\beta = 0.001$ . This ensured that both models, which had the same total mass, also had the same total damping. The  $\beta$  value was selected to ensure that the rotational DOFs on the 184-DOF model were damped. Figure 3.10 shows the roof displacements of each model for the first ten seconds of the simulation. While both models capture the overall behavior of the structure, there are clearly differences between the two responses.

The mass matrix of the shear model is given as:

$$M = \begin{bmatrix} 1.01e6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9.89e5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9.89e5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.89e5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9.89e5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9.89e5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9.89e5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.89e5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.07e6 \end{bmatrix} \text{ kg} \quad (3.19)$$

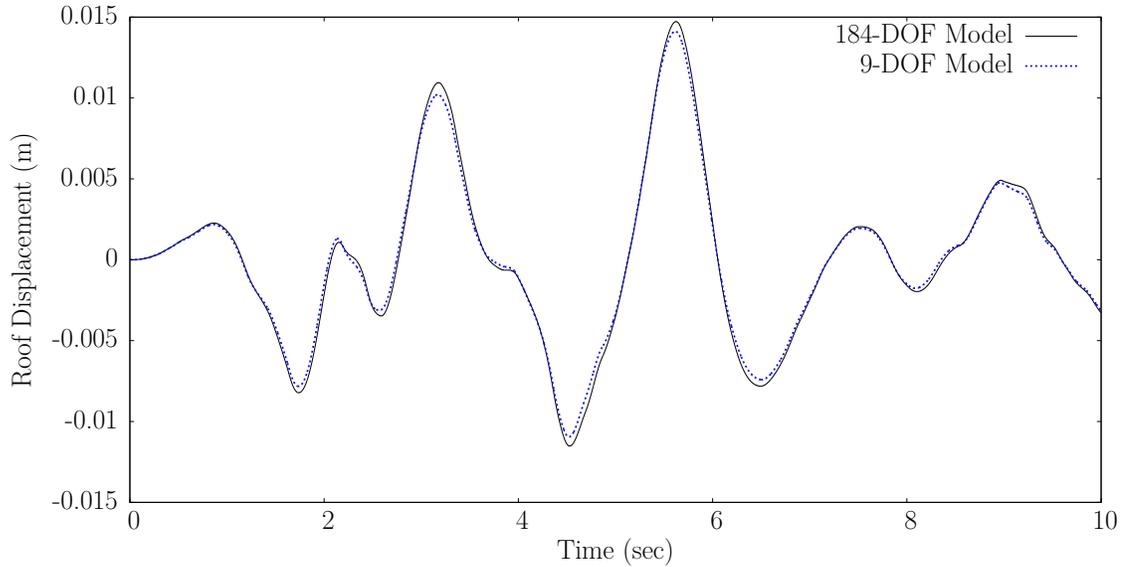


Figure 3.10.: Comparison of Roof Displacement for the 184-DOF FEA model vs the 9-DOF Shear Model

And the stiffness matrix is given as:

$$K = \begin{bmatrix} 5.19e8 & -3.70e8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3.70e8 & 7.07e8 & -3.38e8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3.38e8 & 6.53e8 & -3.16e8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3.156e8 & 5.71e8 & -2.56e8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2.56e8 & 4.87e8 & -2.32e8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2.32e8 & 4.12e8 & -1.80e8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.80e8 & 3.49e8 & -1.69e8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1.69e8 & 3.41e8 & -1.72e8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.72e8 & 1.72e8 & 0 \end{bmatrix} N/m \quad (3.20)$$

## Control System

To realistically account for the computational time required, a controller is included in the vRTHS. Figure 3.11 shows the control schematic for the experiment. The control loop is solved numerically on the Cybermech platform using discrete state space integration. The resulting displacement and velocity are sent to the damper model using a National Instruments NI-6259 data acquisition card. The damper model is implemented on a separate computer as the virtual physical substructure.

## Control System

To realistically account for the computational time required, a controller is included in the vRTHS. Figure 3.11 shows the control schematic for the experiment. The control loop is solved numerically on the Cybermech platform using discrete state space integration. The resulting displacement and velocity are sent to the damper model using a National Instruments NI-6259 data acquisition card. The damper model is implemented on a separate computer as the virtual physical substructure.

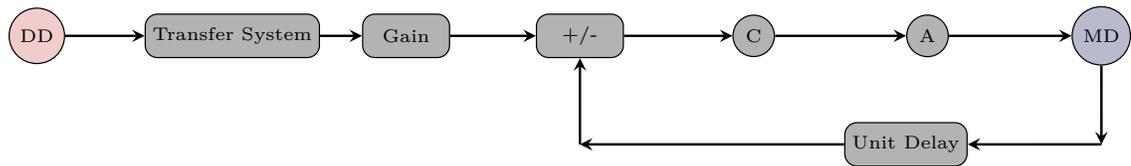


Figure 3.11.: Control System Included for Realistic Computation Times

## Physical Substructure

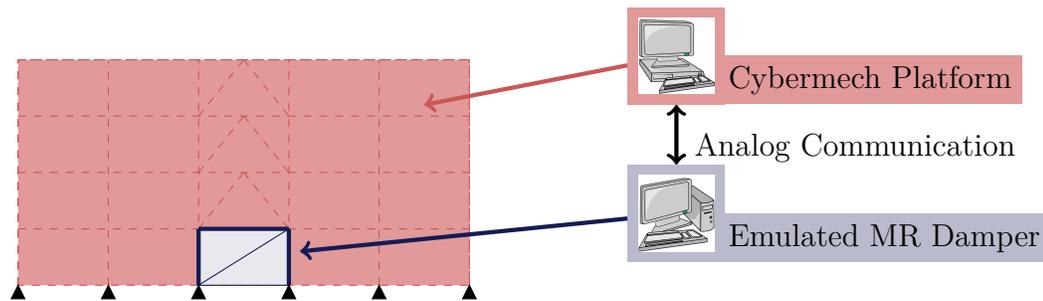


Figure 3.12.: virtual Real-Time Hybrid Simulation (vRTHS)

Figure 3.12 shows the setup for a ‘virtual’ RTHS experiment (vRTHS), where the physical substructure is modeled on a separate computational platform from the numerical substructure. The Cybermech platform sends and receives signals as if it was communicating with a physical substructure in an actual RTHS although no actuator is present as the transfer system. Analog data transfer is used, which introduces some

experimental noise. The virtual physical substructure is a 200 kN magneto-rheological (MR) damper emulated with a Bouc-Wen model [95] using xPC target and placed on the first story of the structure, as shown in Figure 3.9. The MR damper is in a passive-on mode, with a constant DC voltage signal of 2V. The displacement applied to the damper is the numerical difference in displacement between the 1st and 2nd floors of the structure. The force from the damper is then returned to the *Cybermech* computer and applied to the corresponding nodes on the numerical substructure.

For comparison of solutions from MTS+RTHS, a reference solution is generated using the 184-DOF model and a time step ratio of one, with the physical-numerical communication *after* synchronization, which is numerically equivalent to using a uniform time-step method. The structure is excited with the 1940 El Centro earthquake history, with an earthquake magnitude of 10%. Figure 3.13 shows the response of the roof both with and without the damper for the first ten seconds of the simulation. The damper has an effect of reducing the peak roof displacement from 2.217 cm to 1.997 cm, a reduction of 9.93%. The error in the solution is given as

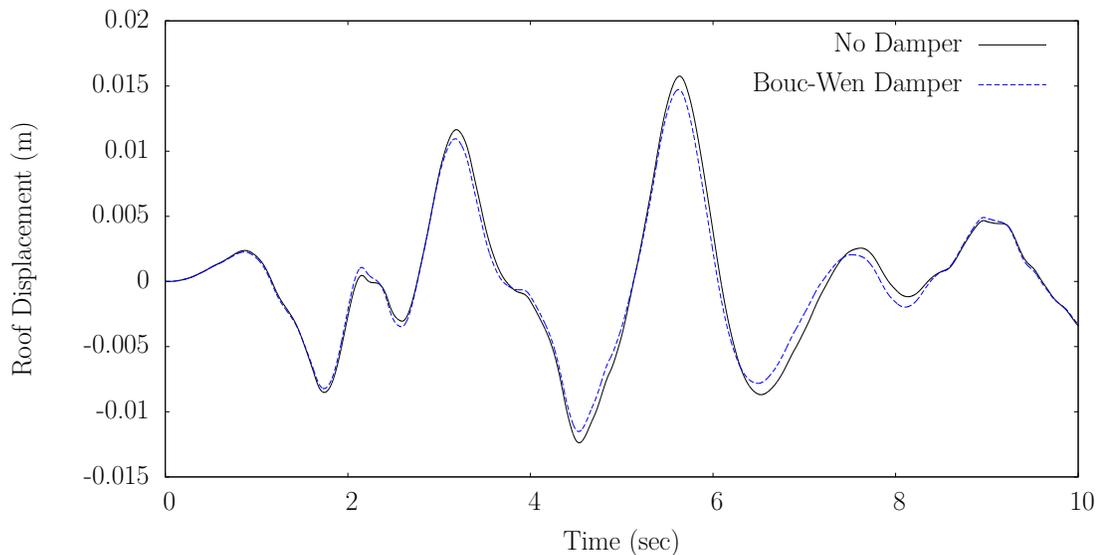


Figure 3.13.: Effect of Bouc-Wen Damper. The maximum roof displacement is reduced by 9.93 %

$$\epsilon = \sqrt{\frac{1}{N} \sum_{n=0}^N \frac{1}{I} \sum_{i=0}^I \left( \frac{x_n^i - x_*^i(t_n)}{MAX(x_*)} \right)^2} \quad (3.21)$$

where  $N$  is the number of small time steps taken, and  $I$  is the number of degrees of freedom in the system.

### 3.5.2 Error Distribution with Shear Model

While this model has substantially a lower computational cost than the 184-DOF shear model, it fails to capture rotations, vertical response, and the behavior of the structure within each floor of the system. In this work, four main sources of error are evaluated: modeling error, time discretization error, experimental error, and synchronization error. Maghareh *et al.* [4] gives a detailed analysis for the modeling error in this structure, and finds that reducing the degrees of freedom in the model from the 184-DOF model (presented here) to a 9-DOF shear model added about 4 percent error in the displacement.

Modeling error is generated by the series of assumptions made when the structure is modeled as a numerical substructure. Some of the modeling error can be alleviated by increasing fidelity, and consequently the size of the numerical model. Figure 3.14 and 3.15 compare the damper force and damper displacement respectively for 184-DOF model and the 9-DOF shear model. The damper force error in the shear model, using the 184-DOF model as a reference, is 16.32%. The displacement error in the shear model, using the 184-DOF model as a reference, is 3.73%. Maghareh *et al* [4] gives a detailed analysis for the modeling error in this structure, and finds that reducing the degrees of freedom in the model from the 184-DOF model (presented here) to a 9-DOF shear model added about 4% error in the displacement, which is consistent with the findings presented here.

Time discretization error is the error generated by the choice of  $\Delta T$ , choosing a larger time step will result in higher error in numerical integration for any integration technique. The time discretization error is computed by solving the different MTS

cases in a pure numerical system, without the MR damper, and comparing to a reference solution ( $m = 1$ ).

Experimental error refers to the random errors resulting from the experimental setup. This is dominated by analog data transfer between the numerical model and the physical substructure, which adds some communication noise. Experimental error can be defined as the expected differences between different experimental runs. To quantify this error here, a reference solution is generated by running each of the cases ( $m = 1, 2, 4, 8, 16, 32, 64$ ) ten times, and taking the mean response at each degree of freedom and time-step over the ten runs. Experimental error is then quantified as the average RMS error between each of those ten runs and the reference solution.

Finally, synchronization error is introduced by the asynchronous updates in the MTS+RTHS algorithm, specifically due to missing the synchronization step between subdomains A and B in the MTS method. For the case when the time step ratio is equal to one, the synchronization error is zero if the updates are applied before communication with the physical control system. This case is used as a reference solution. The synchronization error for each choice of time step ratio is then obtained by computing the RMS error between the average solution and the reference solution, and subtracting the known time discretization error.

### 3.5.3 Error Distribution with No Predictor

The total error is determined by summing the time discretization error, experimental error, and synchronization error. The error associated with each choice of time step ratio  $m$  in the finite element model is shown in Figure 3.16 (a), and the error in the MR damper force is shown in Figure 3.16 (b). The contribution of each type of error to the total error is shown as a normalized percentage in Figure 3.16 (c) and 3.16 (d).

Figure 3.17 shows a portion of the time history response from  $t = 7s$  to  $t = 7.5s$  of the MR damper force at a selected number of time step ratios. Here, the inconsistencies

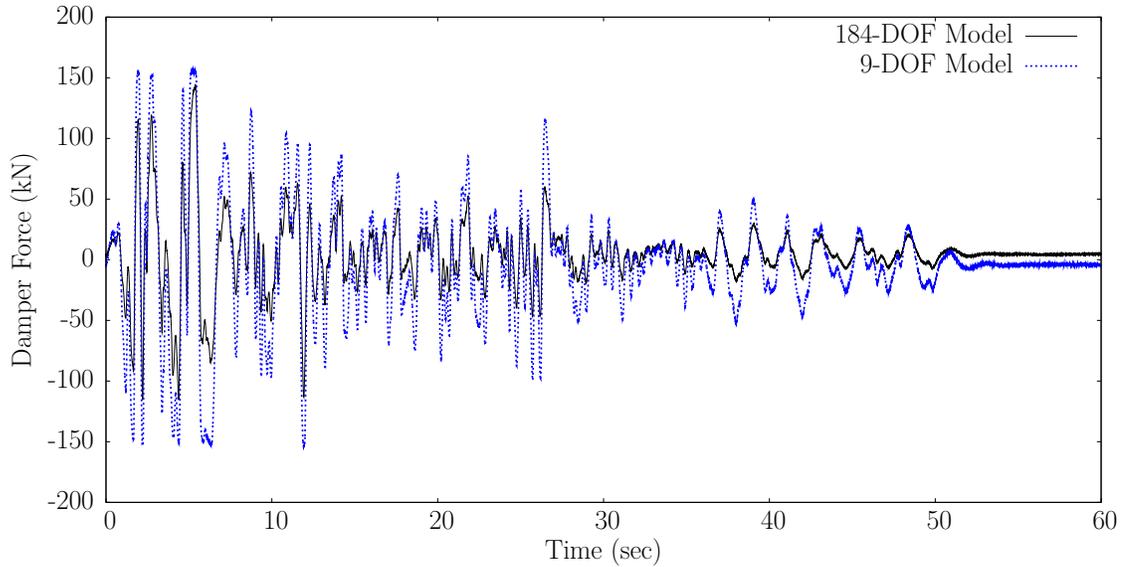


Figure 3.14.: Measured MR damper force for FEA model vs Shear Model (passive-on)

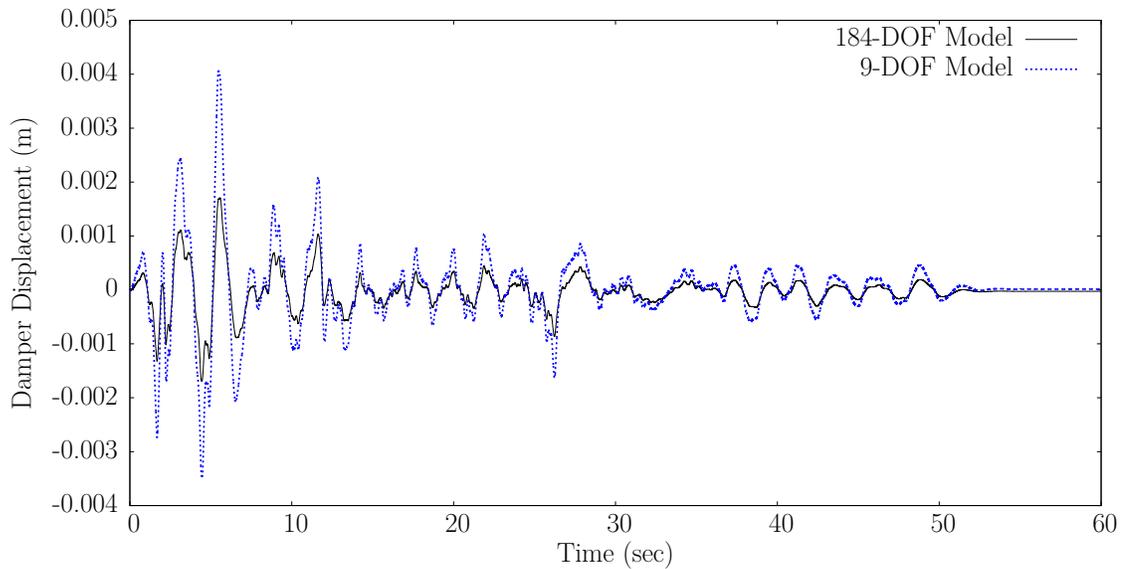


Figure 3.15.: Damper Displacement for FEA model vs Shear Model

due to the MTS method are clear. The damper force takes a sudden jump once every large time step, or once every  $m$  small time steps. Note that this jump is present but not visible for  $m = 4$ , but is very visible for larger choices of  $m$ . In addition to the numerical inconsistencies caused by this jump, note that large changes in desired

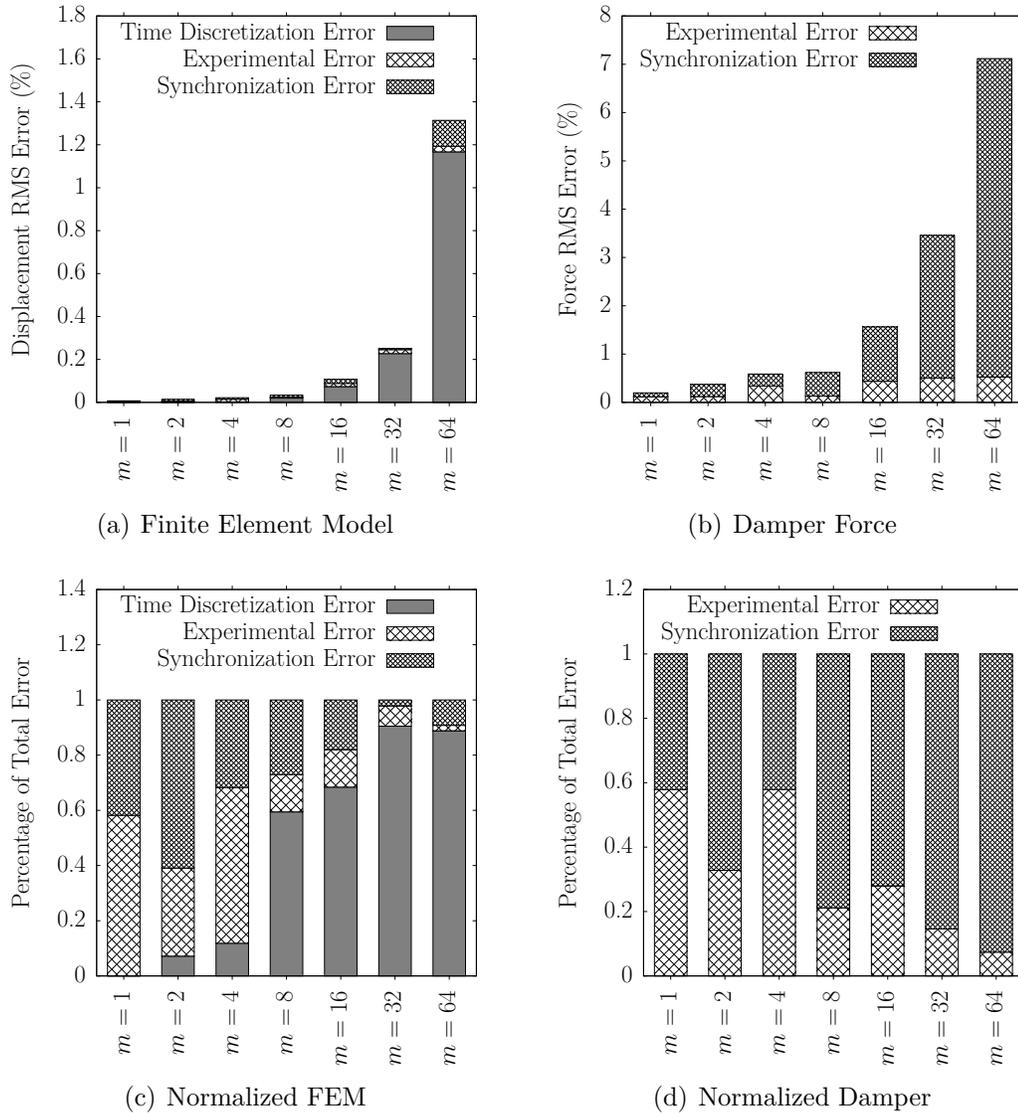


Figure 3.16.: Error with No Predictor (Scheme # 1)

displacement can result in poor performance of the control system. Additionally, these jumps can potentially introduce chattering or instabilities in the physical system, as well as damage physical components. The goal of each predictor scheme is to minimize this jump, and subsequently synchronization error throughout the entire finite element model.

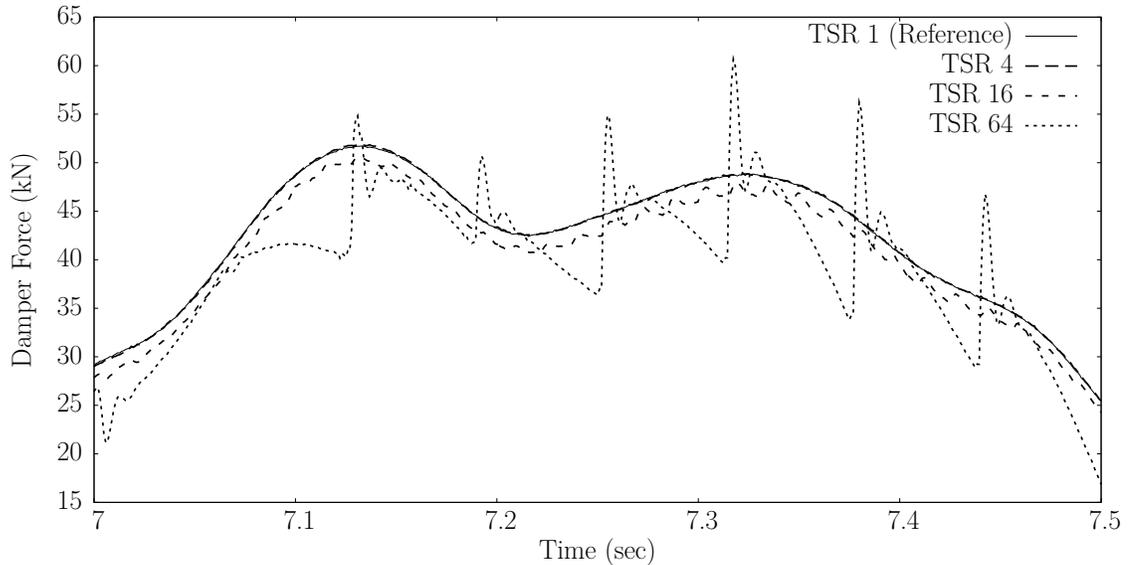
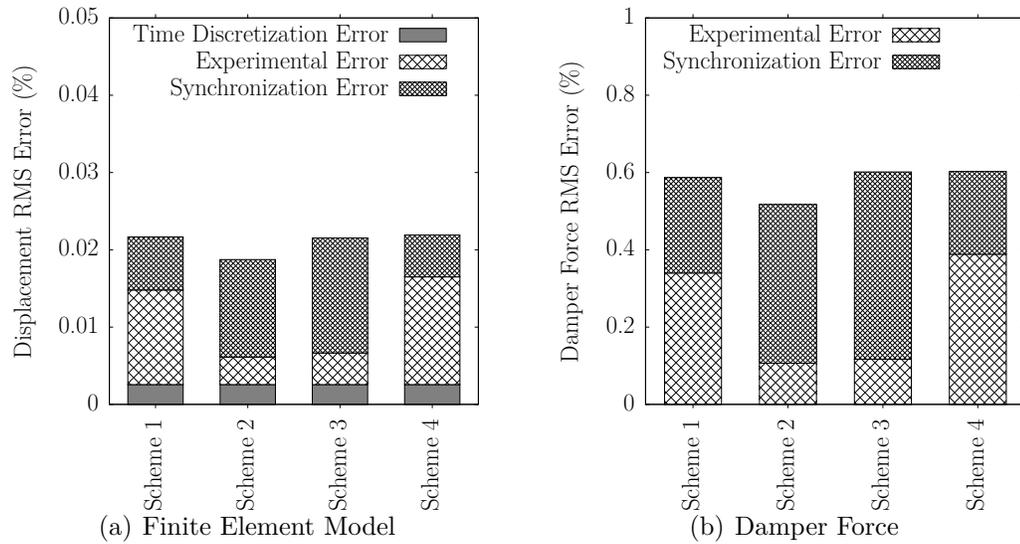
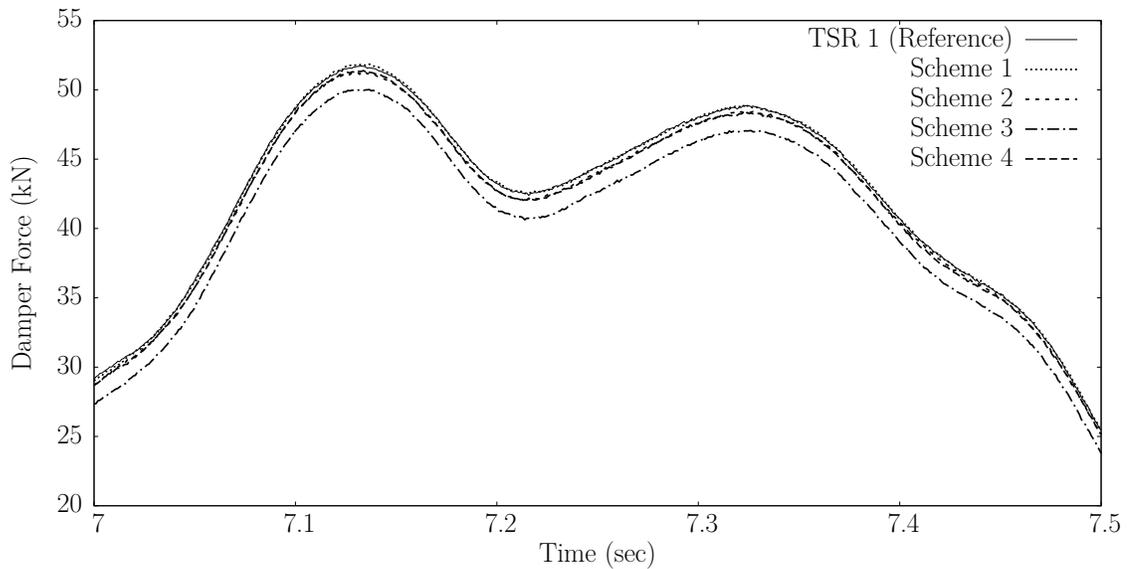


Figure 3.17.: Peak measured MR Damper force with differing time step ratios

#### 3.5.4 Analysis of selected predictor schemes

Recall the list of predictor schemes chosen in Table 3.1, and that Scheme # 1 represents no prediction for reference. Figure 3.18 shows the time discretization, experimental, and synchronization error associated with each prediction scheme when  $m = 4$ . Even for this small value of  $m$ , the prediction schemes are able to reduce the synchronization error in the system. It is interesting to note that for Scheme # 4, the poor prediction results introduced instabilities (Figure 3.17) which increase both the variability between experiments and the experimental error. It is hypothesized, but not explored here that these jumps result from the piecewise continuous nature of the earthquake time record, where large jumps in loading can negate the benefits of higher order prediction schemes that rely on smoothness in the higher order derivatives of the solution. While the true earthquake loading is continuous and smooth, the recorded history at a discrete number of points shows sudden changes in loading values. Figure 3.19 shows the MR damper force with each prediction scheme for  $m = 4$ .

Figure 3.20 shows the time discretization, experimental, and synchronization error associated with each prediction scheme when  $m = 16$ . It is shown here that the

Figure 3.18.: Predictor Error with  $m = 4$ Figure 3.19.: Measured MR Damper Force with  $m = 4$ 

synchronization error in the physical substructure is of the order of the experimental error, and in the finite element model, the synchronization error is dwarfed by the time discretization error. It is interesting to note that reducing the synchronization error

also reduced chattering in the system (caused by discontinuities in desired damper displacement), which decreased the average experimental error.

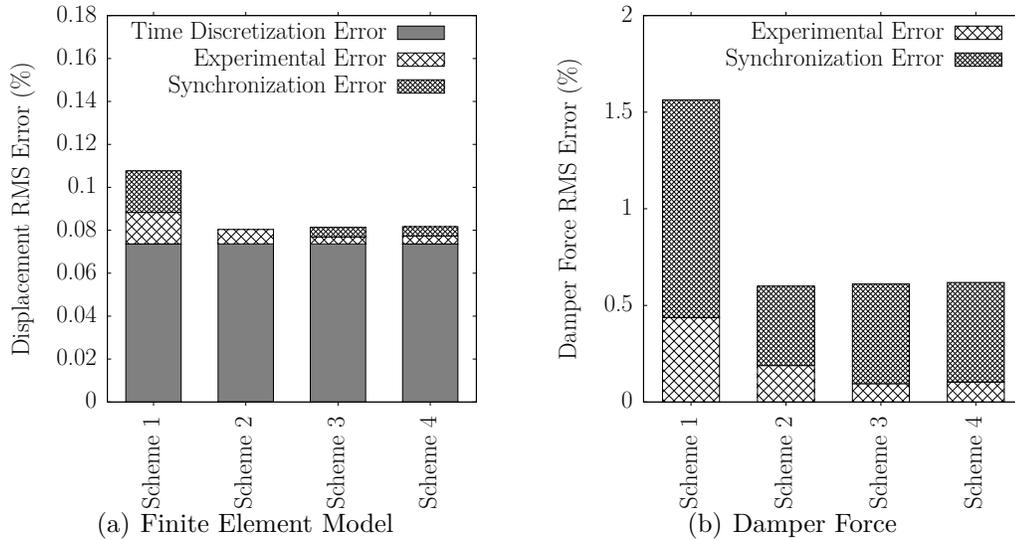


Figure 3.20.: Predictor Error with  $m = 16$

Figure 3.21 shows the MR damper force with each prediction scheme. Note that vRTHS using MTS with Schemes #1 and #4 results in an offset with the reference force, but with schemes #2 and #3, MTS is able to match the reference force very well.

Figure 3.22 shows the error distribution with  $m = 64$ . For this case, the time discretization error is an order of magnitude larger than the other types of error. The synchronization is shown to be larger than the experimental error for this choice of  $m$ . This outcome is not surprising, as Figure 3.23 shows that the predictors have largely failed to match the reference damper force. Though the magnitude of the discontinuities is greatly decreased, the physical substructure is clearly not receiving a smooth signal, and the algorithm is introducing higher order dynamics into the system.

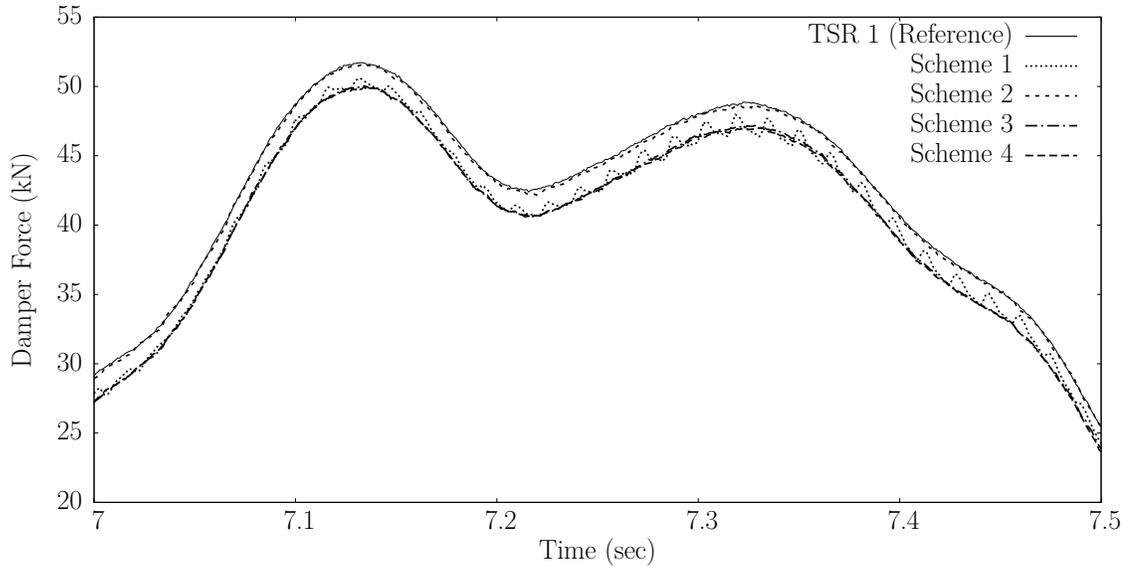


Figure 3.21.: Measured MR Damper Force with  $m = 16$

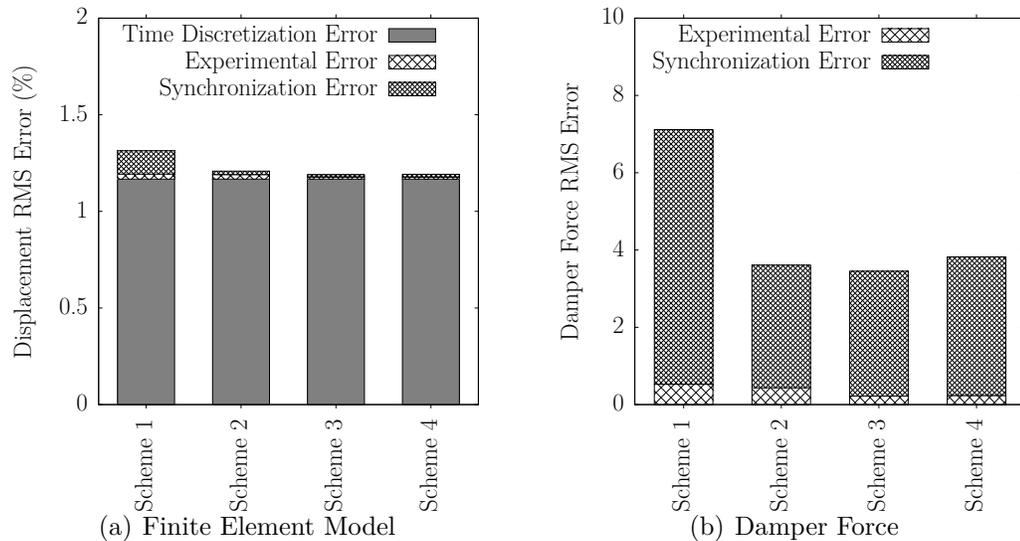


Figure 3.22.: Predictor Error with  $m = 64$

### 3.6 Discussion

In this work, it is shown that the 9-DOF shear model results in a large error relative to the 184-DOF frame model. For high fidelity RTHS, even larger models may be needed, and MTS+RTHS can be used to run these large models within real time

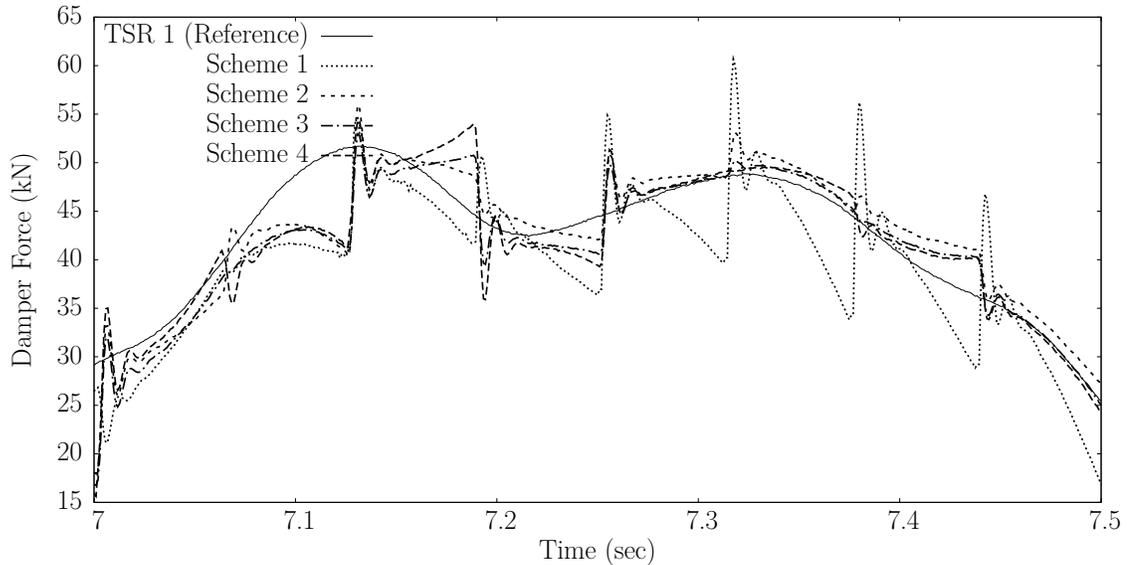


Figure 3.23.: Measured MR Damper Force with  $m = 64$

constraints. The choices made in MTS+RTHS parameters clearly have a significant impact on the behavior of both the numerical substructure (the finite element model), and the simulated physical substructure (the MR damper). It is observed that, while the *asynchronous updates* do introduce some synchronization error into the RTHS system, it is relatively small when compared to the error generated simply by the choice of  $\Delta T$  (shown in Figure 3.18, 3.20, 3.22).

The reduction in error is much lower for the  $m = 4$  case. This result is attributed to the low error in the system. The synchronization error is on the same order of magnitude as the sensor noise, being under 1% total RMS error in the physical substructure and under 0.1% total RMS error in the numerical substructure.

For the case of  $m = 16$ , the synchronization error is nearly eliminated, and is an order of magnitude smaller than the experimental error. This demonstrates that we can use large time step ratios to solve larger models of numerical models using MTS+RTHS.

The time step ratio choice of  $m = 64$  pushes the limits of the method, but the synchronization error still comprises less than 20% of the total error in the finite

element model. Applying the predictor reduced that percentage down to about 10%. However, the 4% error in the MR damper force, along with the additional dynamics added by the MTS method, indicate that  $m = 64$  is too large a choice for this problem. The 1940 El-Centro earthquake acceleration history used in this experiment was only sampled every 0.02, resulting in a piecewise continuous loading that changes values very suddenly. A choice of  $\Delta T$  of 16-Hz (0.0625 sec) is already exceeds this sampling frequency, and may be too large for the prediction schemes.

With these observations, it is clear that MTS+RTHS can be used to solve large numerical substructures coupled with physical substructures to improve the fidelity of RTHS. However, it is important to understand errors created by the choice of  $\Delta T$  and  $\Delta t$ , and to make sure that discontinuities and chattering is not introduced by the coupling method.

### 3.7 Conclusions

MTS+RTHS can be used to solve large numerical substructures while still meeting real-time constraints to improve the fidelity of RTHS. In this study, the MTS synchronization error is studied, and several predictors are presented to mitigate this error. The effects of each predictor at several time step ratios are provided and discussed for a benchmark nine-story frame structure. Furthermore, it is shown that the synchronization error for time step ratios as high as  $m = 16$  is of the same order of magnitude as experimental error for the benchmark problem. This work presents an MTS+RTHS method that enables large finite element models to be used in RTHS, greatly increasing the current RTHS capabilities.

## 4. HIERARCHICAL MULTI-SCALE MODELS FOR PARALLEL REAL-TIME HYBRID SIMULATION

Real-Time Hybrid Simulations (RTHS) are experiments where physical and numerical substructures are coupled in real-time. RTHS is used when a physical model is too large or too expensive to construct, and a numerical model is unable to accurately capture the dynamics of the system. Historically, these simulations have been limited to small, linear numerical models, as the computational limitations of RTHS have prevented the use of larger models. In addition to the steady increase in available computing power, new computational algorithms have emerged that allow for larger numerical models. In this work, a hierarchical structure of numerical models is constructed, ranging from small linear models to large nonlinear models. Large nonlinear numerical models that utilize multi-time-step (MTS) and beam-continuum coupling are used in an RTHS experiment of a five-story scaled model building, and the results are compared to both simple linear models and full physical tests. The use of these models in RTHS is analyzed, and the results demonstrate that such multi-scale models can be effectively used to improve the fidelity of RTHS.

### 4.1 Introduction

In the field of structural analysis, engineering structures are modeled and tested to see how they might perform under various conditions, such as impact or earthquake loading. These models can be physical, typically built at a small scale and tested in an experimental laboratory. These models can also be numerical, modeled in a finite element program, and tested by solving the equations of motion with respect to given loading conditions. Both types of experiments have limitations: full-scale physical experiments are typically prohibitively expensive, small-scale physical experiments

often fail to capture the response of the structure, and numerical experiments may fail to capture non-linearities in the structure. Real-Time Hybrid Simulations (RTHS) are coupled experiments, where a portion of the structure is modeled numerically, and a portion of the experiment is performed physically [65]. RTHS are performed when a substructure of the building is well understood and can be modeled numerically, but a realistic model for another portion of the building is not available, and must be simulated experimentally to get reliable results. An actuation device, often powered by pneumatic pressure, is used to control the interaction between the numerical and physical substructures. Figure 4.1 shows such a real-time hybrid simulation.

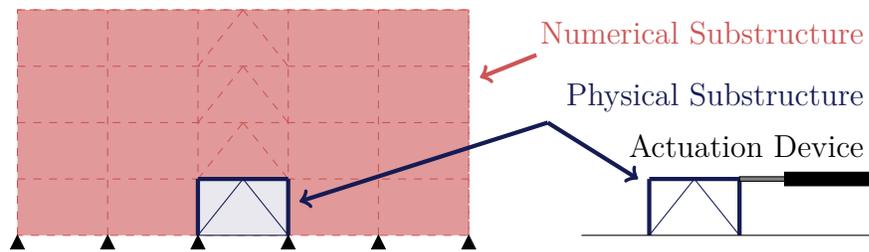


Figure 4.1.: Real-Time Hybrid Simulation Experiment

Many successful Real Time Hybrid Simulations are available in the literature [3, 7, 75, 89, 90]. Figure 4.2 shows the typical data flow between the numerical substructure(s) (NS) and physical substructure(s) (PS) of a real-time hybrid simulation. The numerical substructure (NS) sends a Desired Displacement (DD) to the controller. The controller, with some knowledge of the actuation (A) device and physical substructure (PS), sends a command signal (CS) to the actuator. The measured displacement (MD) is sent to the controller, and applied to the physical substructure. The measured force (MF) is returned to the numerical substructure.

A common type of RTHS experiment is displacement driven, where a hydraulic actuator applies a displacement or series of displacements on the physical substructure. These displacements are applied via actuator devices, which can be powered either electronically or through a hydraulic system. The control of such devices is an active area of research, but typically open or closed loop controllers are used to drive the

devices [93]. Outer loop actuator controllers are often included to enforce the numerical boundary conditions on the physical substructure(s) [2, 69, 74, 77, 94].

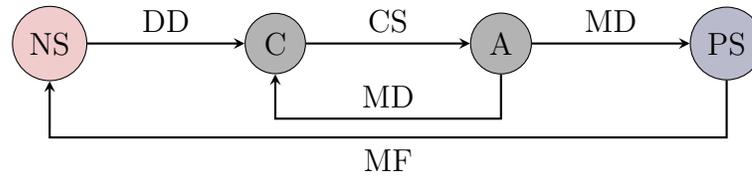


Figure 4.2.: RTHS Control Flow Schematic

The dynamic nature of these experiments mandates that the numerical substructures be solved in *real-time*: to accurately capture the dynamics of the entire system, the numerical substructure must advance at the same rate of execution as the physical substructure. Typically, due to the dynamics of the actuation device, real-time is generally defined as 1024-Hz or faster. The computational cost and real-time constraints limit the size of potential numerical substructures. While most purely numerical simulations include non-linearities, the field of RTHS has historically been limited to small, linear numerical models [77–82, 96]. Some experiments are limited to one degree of freedom (DOF) per building, while others use linear or nonlinear finite element analysis [8, 83–85, 96].

While linear finite element analysis can be used to model structures, these models fail to capture building response in the non-linear range. If structures enter their nonlinear regions, as often happens during earthquakes, nonlinear material models are needed to compute their response. In the non-linear range, the stiffness of the material is dependent on both current and prior displacements. One such material model is J2 plasticity, originally proposed by Von Mises [97]. Other types of non-linearity include large deformations, where small angle assumptions may no longer hold. There are some examples of nonlinear models in RTHS [8], where a specific non-linearities are included to capture a specific type of behavior. However, RTHS experiments utilizing large nonlinear finite elements models are largely absent from the literature.

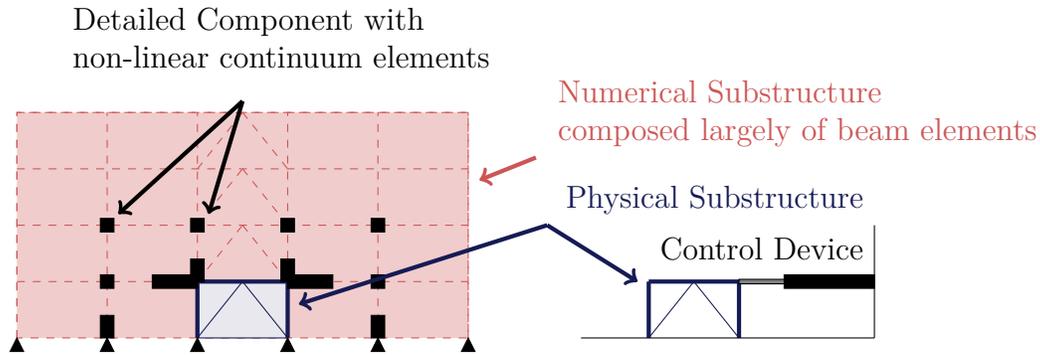


Figure 4.3.: System level RTHS with multi-scale models

In some cases, it is not possible to model the entire structure with a refined mesh or a complex nonlinear model, but it may be possible to model parts of the structure. If areas of the structure likely to go nonlinear can be determined *a-priori*, multi-scale models that capture the nonlinear behavior in nonlinear regions can be used, but use inexpensive models in the linear regions of the structure. Models with that span multiple scales of space and/or time are known as *multi-scale* models. Models that are multi-scale in *time* use techniques such as multi-time-stepping (MTS) to solve a region of the model at a smaller time-step. Models that are multi-scale in *space* may use a finer spatial discretization and complex multi-scale material models to capture nonlinear behavior in select regions of the model. Multi-scale models have been proposed to improve the fidelity of RTHS. The coupling of linear beam elements with nonlinear continuum elements is an example of a model that is multi-scale in space. When nonlinear continuum elements are solved at a finer time-scale to capture multi-scale behavior, the model is both multi-scale in space and multi-scale in time. Conducting parallel RTHS with such models, as shown in Figure 4.3 is the objective of this work..

## 4.2 Cybermech Computational Platform

Recent work by Ferry and colleagues [9, 10] has led to the development of **Cybermech**, a real-time platform based on federated scheduling [98] designed specifically to run large numerical models for RTHS. The work presented in Chapter 3 has led to the development of real-time hybrid simulation with multi-time-step analysis (RTHS+MTS), allowing portions of the numerical model to be solved at different time steps. These developments, along with the availability of faster processors and large parallel computers, have increased the potential size of numerical substructures that can be run in RTHS. In this work, the Cybermech platform is used to run RTHS with hierarchical models, which are multi-scale in both time and space. In this work, the **Cybermech** platform is extended using a patched version of openBLAS [99] (a parallel basic linear algebra library) that forces openBLAS to respect the processor affinities specified in the **Cybermech** platform, as the out-of-the-box OpenBLAS library tends to use every CPU on the host machine. In addition, modifications to the shared memory object that allows passing of data between separate executables running concurrently on **Cybermech** was also implemented to enable parallel RTHS of nonlinear structures.

The Cybermech platform is a computational platform that allows for full, nonlinear finite element models to be solved in real-time [9, 10]. Alternative computational platforms for RTHS include Matlab's xPC target [91], SpeedGoat xPC [100], and dSPACE [101]. However, none of these allow for real-time parallelism. One advantage of the CyberMech platform is that it allows for fully customizable C/C++ code to be run in parallel using real-time priorities, ensuring that real-time constraints can be met. It is this customizable C++ capability that allows for multi-scale RTHS.

The platform is able to run two parallel tasks concurrently, where data is transferred between them via Linux's shared memory model. For a MTS problem, which is multi-scale in time, the *slow* task is computed each coarse time-step  $\Delta T$ , and the *fast* task

is computed each small time-step  $\Delta t$ . The small time step is executed  $m = \frac{\Delta T}{\Delta t}$  times for each coarse time step.

### 4.3 Formulation of Multi-Scale Models for RTHS

#### 4.3.1 Nonlinear Explicit MTS

Multi-time-step integration methods are a family of numerical integration algorithms that allow one to solve dynamic structural analysis problems at different rates in time [14, 16, 17, 92, 102, 103]. Prakash and coauthors [17] derive a nonlinear version of the MTS coupling method. A brief overview of this method is given, focusing on the changes to the algorithm necessary for adapting MTS and multi-scale models to RTHS as shown in Figure 4.4. In this work, multi-scale models are built using beam and continuum coupled elements.

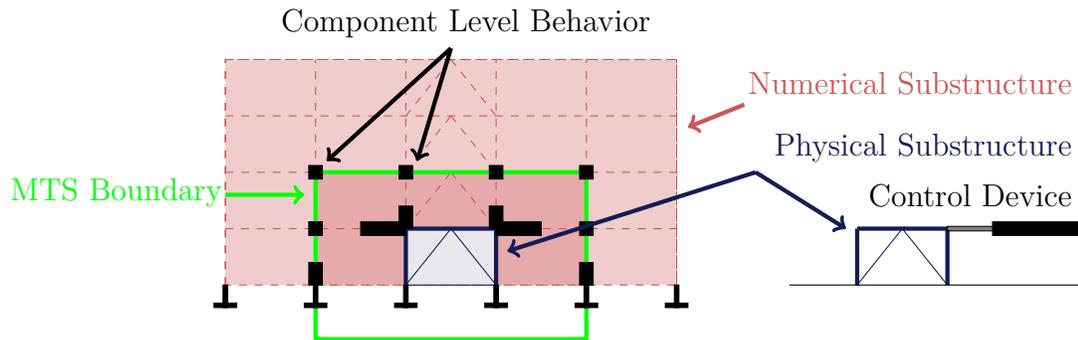


Figure 4.4.: System Level RTHS with Multi-Scale Models

For MTS, a finite element mesh is split into two or more parts. Here, Subdomain A represents the region of the structure solved at the large time step  $\Delta T$ , and Subdomain B represents the region of the structure solved at the small time step  $\Delta t$ . Traditionally, the coupling between two domains creates a constraint matrix that enforces continuity of velocity between two subdomains at each corresponding degree of freedom. Here, an alternative method is constructed, allowing beam nodes (with two translational degrees of freedom and a rotational degree of freedom) to be coupled with continuum

nodes (with only the two translational degrees of freedom). Constraints are added between the rotational degree of freedom on the beam node to the translational degrees of freedom on the continuum nodes, which ensure consistency of the cross-sectional displacements of the beam and continuum elements at their interface. Here the traditional MTS method is extended to encompass beam-continuum coupling.

#### 4.3.2 Equations of Motion

On Subdomain A, a linear, implicit algorithm is used on beam elements. This Subdomain can be advanced by solving its equation of motion

$$\mathbf{M}^A \ddot{\mathbf{u}}^A + \mathbf{D}^A \dot{\mathbf{u}}^A + \mathbf{K}^A \mathbf{u}^A + \mathbf{C}^{A^T} \boldsymbol{\lambda} = \mathbf{f}^A \quad (4.1)$$

using traditional implicit Newmark parameters. On Subdomain B, the internal force vector is denoted as  $\mathbf{f}_{int}^B$ . The equation of motion

$$\mathbf{M}^B \ddot{\mathbf{u}}^B + \mathbf{D}^B \dot{\mathbf{u}}^B + \mathbf{f}_{int}^B + \mathbf{C}^{B^T} \boldsymbol{\lambda} = \mathbf{f}_{ext}^B \quad (4.2)$$

is solved using traditional explicit Newmark parameters. The internal force  $\mathbf{f}_{int}$  may be a nonlinear function of displacement, so an implicit algorithm requires a Newton loop to converge on the nonlinear solution. The real time nature of these experiments, as discussed in Chapter 3 precludes such indefinite algorithmic loops over time, as the physical experiment is moving continuously forward through time. At each large time step, continuity is enforced between the two regions, using linear constraint matrices.

$$\mathbf{C}^A \dot{\mathbf{u}}^A + \mathbf{C}^B \dot{\mathbf{u}}^B = 0 \quad (4.3)$$

Figure 4.5 shows the pseudo-code for the nonlinear solution.

```

for  $t = 0 ; t < tF ; t = t + \Delta t$  do
  while resid > tol do
    Solve Linear Subdomain A -  $d_{lin} = K_{el}^{-1} f_{ext}$ 
    Solve Subdomain B
    for  $n = 0 ; n < m ; n ++$  do
      Obtain  $F_{int}^B$ 
       $resid_{iEl} = [F_{ext} - F_{int}]$ 
    end for
    Couple Subdomains A & B
    Compute Updates ( $\Delta u^A, \Delta u^B, \Delta \lambda$ , etc.)
    Update Solution
  end while
  Exchange with Physical Substructure
end for

```

Figure 4.5.: Nonlinear Coupling Algorithm for MTS with Beam-Continuum Coupled Models

### 4.3.3 Coupling Matrices

The traditional method to build the coupling matrices  $\mathbf{C}^A$  and  $\mathbf{C}^B$  is given in [16], and is adapted here, to highlight the differences between the traditional formulation and the beam-coupling formulation. For multi-time-step methods that couple two continuum regions, the coupling matrices at each node are given as:

$$\mathbf{C}_{node}^A = \begin{array}{c|cc} & X & Y \\ \hline \Lambda_1 & 1 & 0 \\ \Lambda_2 & 0 & 1 \end{array} \quad (4.4)$$

and

$$\mathbf{C}_{node}^B = \begin{array}{c|cc} & X & Y \\ \hline \Lambda_1 & -1 & 0 \\ \Lambda_2 & 0 & -1 \end{array} \quad (4.5)$$

For multi-time-step methods that couple two beam regions, the coupling matrices at each node are given as:

$$\mathbf{C}_{node}^A = \begin{array}{c|ccc} & X & Y & \theta \\ \hline \Lambda_1 & 1 & 0 & 0 \\ \Lambda_2 & 0 & 1 & 0 \\ \Lambda_3 & 0 & 0 & 1 \end{array} \quad (4.6)$$

and

$$\mathbf{C}_{node}^B = \begin{array}{c|ccc} & X & Y & \theta \\ \hline \Lambda_1 & -1 & 0 & 0 \\ \Lambda_2 & 0 & -1 & 0 \\ \Lambda_3 & 0 & 0 & -1 \end{array} \quad (4.7)$$

Each local matrix is assembled into a global coupling matrix for each subdomain, which contains all of the constraints. These constraints are typically enforced on velocities, however it is also possible to enforce them on displacements or accelerations.

#### 4.3.4 Beam-Continuum Coupling

Pitandi [18] developed a beam-continuum coupling model. Lagrange multipliers are used to enforce geometric constraints between beam and continuum regions of a large structural model. Figure 4.6 shows a sample mesh that demonstrates beam-continuum coupling. Figure 4.7 shows the deformed shape of the beam-continuum sample mesh. For beam-continuum coupling, a single beam node is connected to several continuum finite element nodes. The kinematic hypothesis of beam theory is enforced at each point in the continuum mesh. According to the beam kinematic hypothesis, the location of point  $P$  in the continuum model after deformation can be written as

$$x_p = X_b + u_{uw_b} + \Lambda p \quad (4.8)$$

where  $x_p$  is the location of point  $P$  in the deformed configuration,  $X_b$  denotes the location of the beam centroid in the undeformed configuration, and  $u_{uw_b}$  denotes the displacement vector of the beam node containing both the horizontal and vertical components,  $p$  is the position vector of point  $P$  with respect to the centroid of the cross-section, and  $\Lambda$  is the rotation matrix representing the rigid-body rotation of the beam cross-section about its centroid. For planar (2D) problems, the rotation matrix  $\Lambda$  can be simply expressed as

$$\Lambda = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (4.9)$$

The position of a continuum node with respect to a beam node can be described with the variables  $L$  and  $\phi$ . The magnitude  $L$  represents the absolute distance from the beam node to the coupling node, and  $\phi$  represents the counter-clockwise angle from the x-axis to the continuum node, as shown in Figure 4.6. The variable  $\theta$  represents the beam rotational DOF. If small angle theory is used for the rotational degree of freedom at the beam node,  $\sin(\theta) = \theta$ , the result system of linear constraints can be used in place of the traditional coupling matrices for the MTS method. For a large

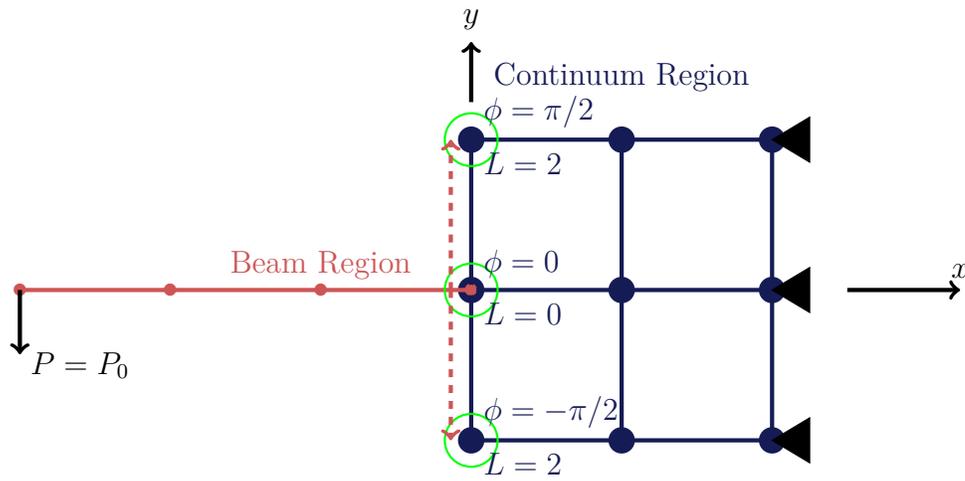


Figure 4.6.: Schematic of Beam-Continuum Coupling

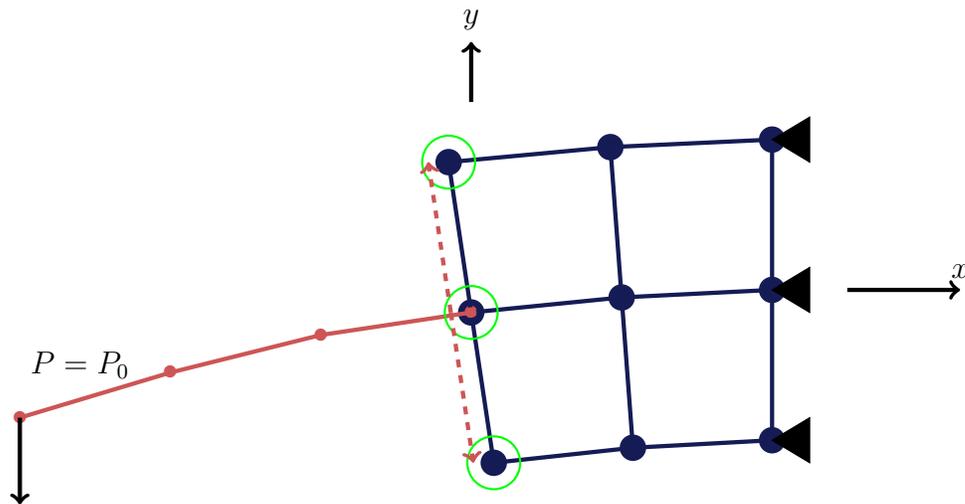


Figure 4.7.: Beam-Continuum Coupling Deformed Shape

displacement nonlinear implementation, the value of  $\phi$  is updated as the continuum nodes move.

The displacement at a node with  $\phi = 0$  caused by a rotation  $\theta$  are:

$$\text{X Direction: } L - L \cos \theta \implies 0 \quad (4.10)$$

$$\text{Y Direction: } L \sin \theta \implies L\theta \quad (4.11)$$

For other values of  $\phi$ , the rotation matrix  $\Lambda$  is applied,

$$\begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} 0 \\ L\theta \end{bmatrix} = \begin{bmatrix} -L\theta \sin \phi \\ L\theta \cos \phi \end{bmatrix} \quad (4.12)$$

resulting in the linear constraint equations for each node:

$$0 = x_{Beam} - L \sin(\theta) \sin(\phi) - x_{Continuum} \quad (4.13)$$

$$0 = y_{Beam} + L \sin(\theta) \cos(\phi) - y_{Continuum} \quad (4.14)$$

These equations rigidly maintain the initial angle of the continuum elements. Note that the continuum nodes along the interface are now also rigidly constrained to each other. A line of continuum elements will remain a line, consistent with a beam cross section. If the beam node and continuum node share the exact coordinates, the distance  $L$  goes to zero, and the contribution from the rotational DOF drops out.

Adding the lateral and vertical constraints on the beam node, the local coupling matrix for a single constraint for the beam subdomain can be derived as:

$$\mathbf{C}_{beam-node}^A = \begin{array}{c|ccc} & X & Y & \theta \\ \hline \Lambda_1 & 1 & 0 & L \sin(\phi) \\ \Lambda_2 & 0 & 1 & -L \cos(\phi) \end{array} \quad (4.15)$$

while the local coupling matrix for a single constraint for the continuum subdomain can be derived as:

$$\mathbf{C}_{\text{continuum-node}}^B = \begin{array}{c|cc} & X & Y \\ \hline \Lambda_1 & -1 & 0 \\ \Lambda_2 & 0 & -1 \end{array} \quad (4.16)$$

These local coupling matrices are assembled into regional coupling matrices, which contain all active DOFs in each region. One beam node can be coupled to multiple continuum nodes. For the linear case, these constraint matrices do not change the overall structure of the MTS algorithm. The only implementation difference for beam-continuum coupling is the sparsity and values of the coupling matrices.

#### 4.3.5 J2 Plasticity

In this work, a basic  $J_2$ -plasticity model with isotropic and kinematic hardening is employed [97]. Von Mises suggested that yielding occurred when  $J_2$  attains a critical value, where  $J_2$  can be written in terms of the principal stresses  $\sigma_1, \sigma_2, \sigma_3$ :

$$J_2 = \frac{1}{6} [(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2] \quad (4.17)$$

The material is treated as elastic when the  $J_2$  is within the yield surface  $J_2 < k^2$ , and is yielding when  $J_2 = k^2$ , where  $R = \sqrt{2k}$  describes the radius of the yield surface, and  $k$  is a measurable material parameter [104].

#### 4.4 Verification

##### Verification of Beam-Continuum Coupling

The cantilever beam verification problem for the beam continuum coupling is shown in Figure 4.8. Figure 4.8 shows the deformed shape and stress for the beam-continuum mesh.

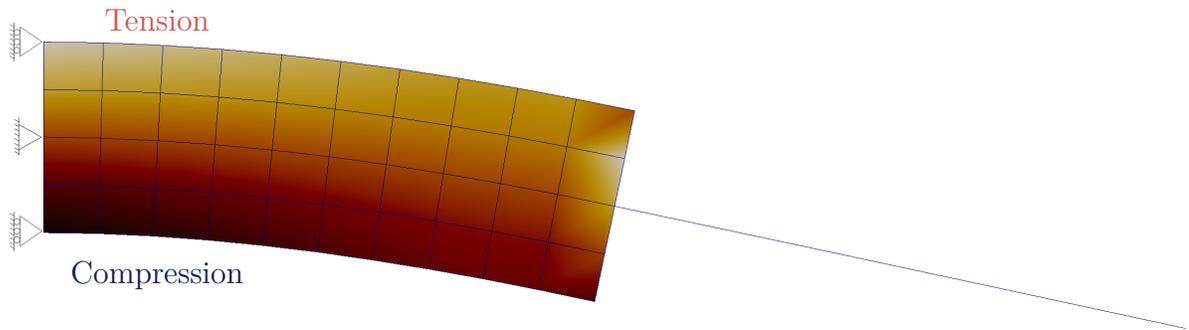


Figure 4.8.: Beam Continuum Deformed Shape and Stresses

The beam is constrained in the lateral direction on the left boundary, with a single vertical constraint at the middle node. At the far right of the structure, a vertical load is applied. The beam had a length of 3 m, a height of 0.5 m, and a depth of 1 m. For the beam elements, the cross section area was 0.5 m<sup>2</sup>, and the moment of inertia about the x-axis was  $I = \frac{dh^3}{12} = 0.01041666m^4$ . For both the continuum elements and the beam elements, Young's Modulus  $E = 2.05 * 10^8$ , and the density  $\rho = 7.8$ . For the continuum elements, Poisson's ratio  $\nu = 0.0$ . A load of  $P = -1.0 * 10^4$  N was applied at the end of the cantilever. The static solution to the cantilever beam problem is given as:

$$\delta = \frac{PL^3}{3EI} = -0.0421463. \quad (4.18)$$

The total mass of the beam is given as

$$m = Lhd\rho. \quad (4.19)$$

where  $L$ ,  $h$ , and  $d$  are the length, height, and depth of the beam respectively. The dynamic response of the beam is shown for a period of 0.1 *sec* in Figure 4.10. There is no damping on the beam for the verification problem. The analytical and finite element static solutions are shown for each case in Table 4.1.

Table 4.1: Static Verification of Beam-Continuum Coupling

	Analytical	Beam	Continuum	Beam-Continuum
Vertical Displacement	-0.0421463	-0.0421463	-0.0409868	-0.040898
Error	0	$2.2 * 10^{-11}$	0.0275	0.0296

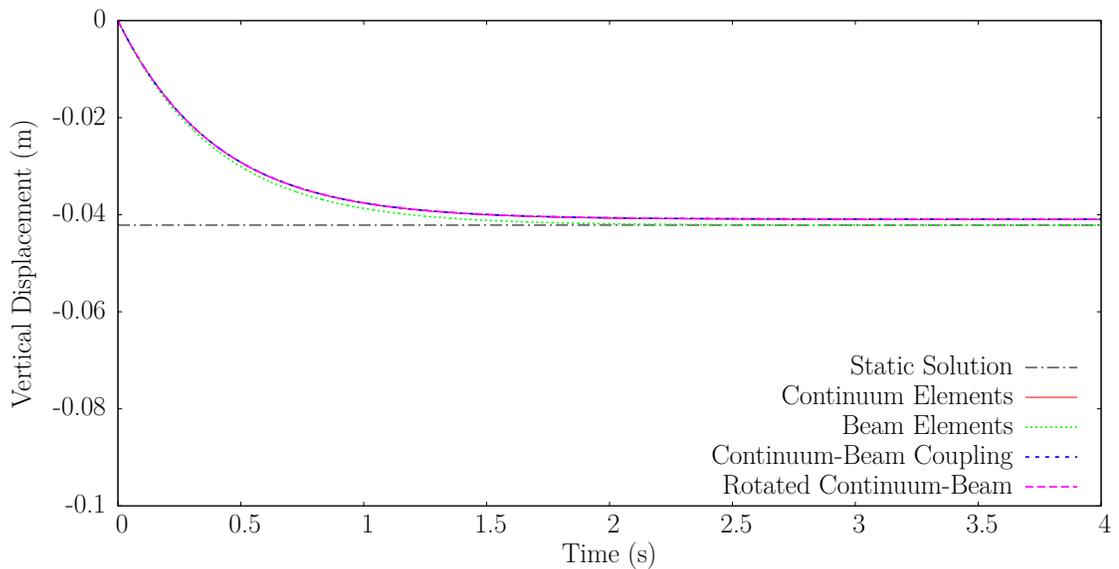


Figure 4.9.: Quasi Static Verification Beam Bending

#### 4.4.1 Nonlinear Verification

The nonlinear Q4 elements were verified on same cantilever beam, composed entirely of Q4 elements. The elastic material parameters consisted of a Young's modulus of  $2e11N/m^2$ , and a Poisson's ratio of 0.0, and a density of  $7800kg/m^3$ . The plastic material parameters consist en of a yield stress of 400 MPa and a plastic

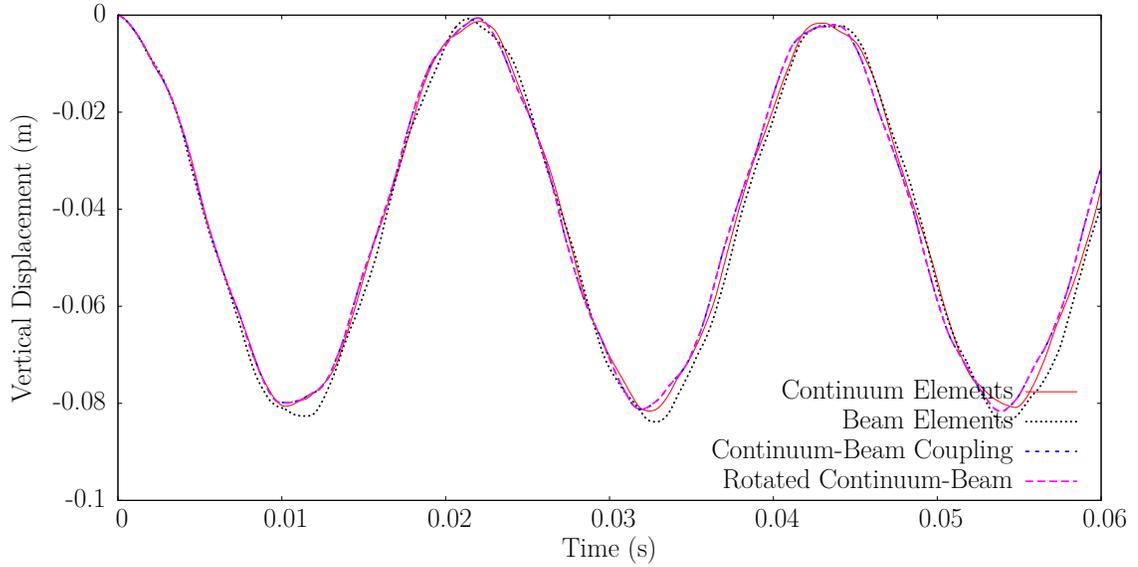


Figure 4.10.: Dynamic Verification Beam Bending

modulus of  $2e10 \text{ N}/m^2$ . The displacement at center of the beam of the cantilevered tip was measured.

Table 4.2 shows the nonlinear verification results.

Table 4.2: Static Verification of Nonlinear Q4 Elements

	ABAQUS	NLQ4
Vertical Displacement	-0.03608	-0.034902

## 4.5 Experimental and Numerical Models

### 4.5.1 Physical Structure

The reference structure is a five story building shown in Figure 4.11. A detailed drawing of a single story is shown in Figure 4.12. The experimental identification of the model is presented in [105, 106]. Each story as a center-to-center column height of 7 in, and is connected by a rigid floor plate. The effective column height, not

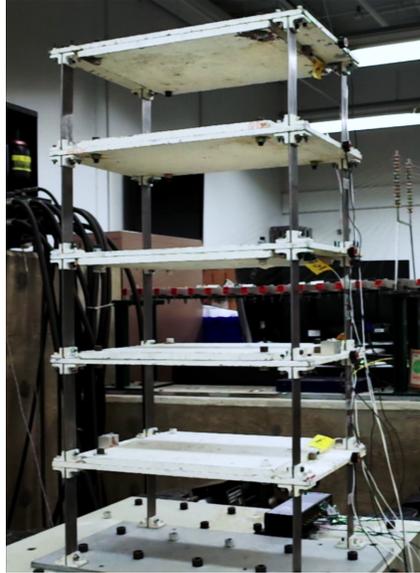


Figure 4.11.: Five Story Frame

constrained by the connection, is 6.5 in. Each column has an average thickness of  $t = 0.125$  in. Each column has an average width of  $w = 1$  in. For beam element models, the cross sectional area of each column  $A = wt = 0.125 \text{ in}^2$ . The moment of inertia,  $I$ , about the weak axis is computed as  $I = wt^3/12 = 1.7342 * 10^{-4} \text{ in}^4$ . Young's Modulus of the columns was measured to be  $E = 192.3 \text{ GPa}$ . A Poisson's ratio of  $\nu = 0.287$  was assumed. A density of  $\rho = 0.284 \frac{\text{lbs}}{\text{in}^3}$  is assumed. Each floor has a total mass of 56.22 lbs. Rayleigh damping is used with a mass contribution of  $\alpha = 1.2$  and a stiffness contribution of  $\beta = 0.01$ . Figure 4.13 shows the stress strain curve of the material obtained from a uni-axial tension test.

#### 4.5.2 Shear Model

The shear model contains 5 degrees of freedom, allowing only lateral movement on each floor. With only 5 degrees of freedom, this is the computationally smallest model to solve, and can be solved well within the real-time limit. There are a number of real time hybrid experiments in the literature that only use shear models as their experimental models. The shear model is shown in Figure 4.14(a).



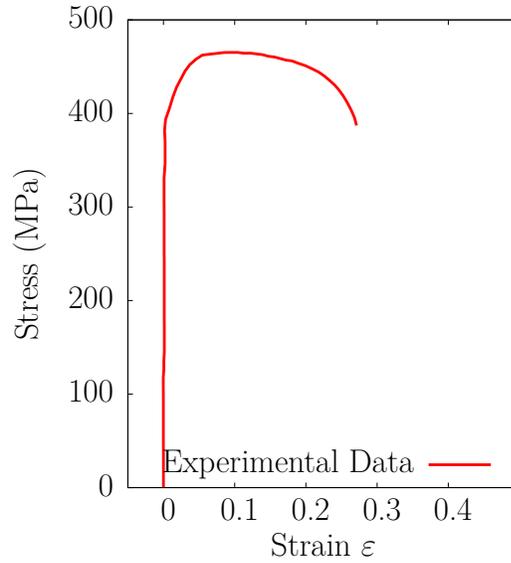


Figure 4.13.: Stress Strain Curve from Uni-axial Tension Test (reproduced from [106])

#### 4.5.3 Simple Beam Model

The simple beam model uses one beam element per column. The two columns are connected by a rigid floor beam element. The model is composed of 12 nodes connected by 25 beam elements. The model contains 30 degrees of freedom: lateral, vertical, and rotational degrees of freedom at each node, minus the four boundary degrees of freedom. The simple beam model is shown in Figure 4.14(b).

#### 4.5.4 Detailed Beam Model

The detailed beam model is made up of multiple Euler-Bernoulli elements with lumped mass. Beam elements with higher cross sectional areas and moments of inertia are used near the connections, in the 0.5 adjacent to the rigid floor. The model is composed of 32 nodes, connected by 45 beam elements. The detailed beam model is shown in Figure 4.14(c).

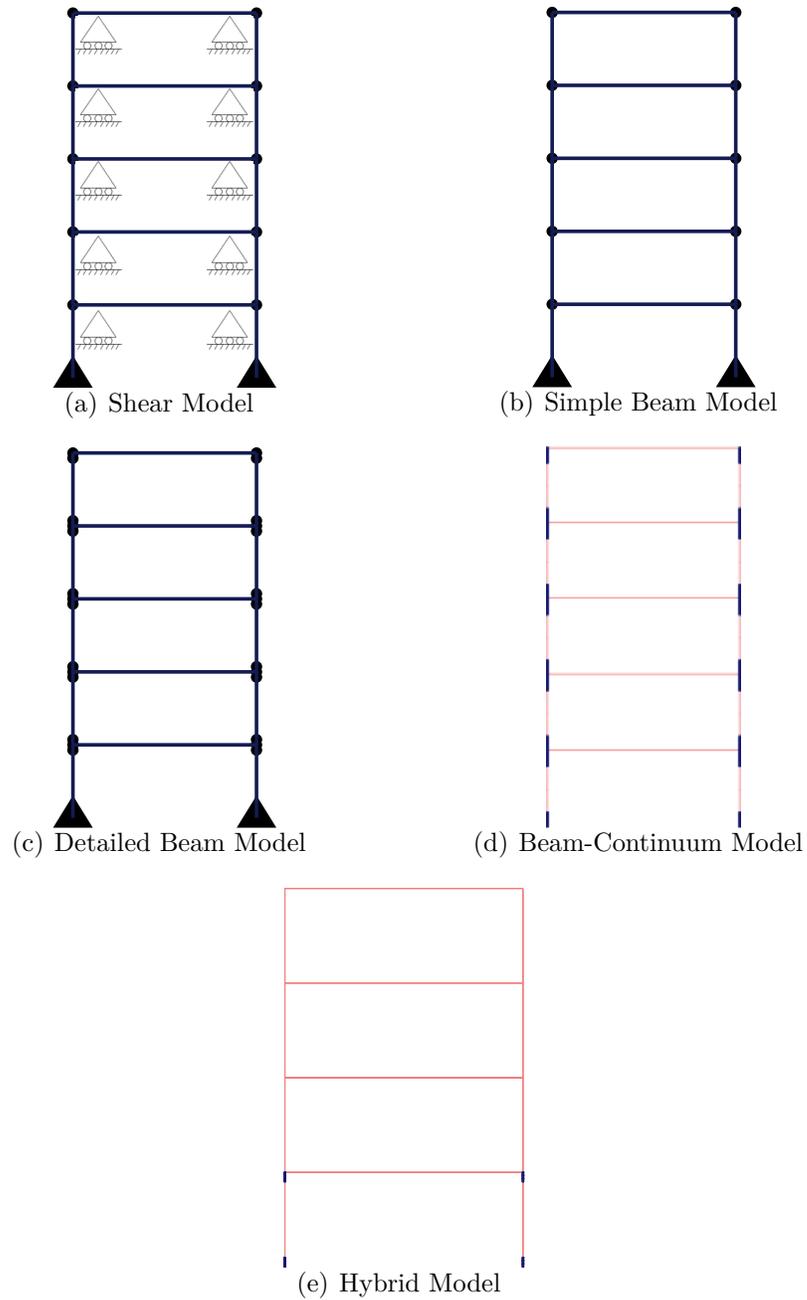


Figure 4.14.: Experimental Models

#### 4.5.5 Linear Continuum Model

A linear continuum model, composed entirely of continuum elements, was considered, however it was not possible to solve such a model within the real-time constraints.

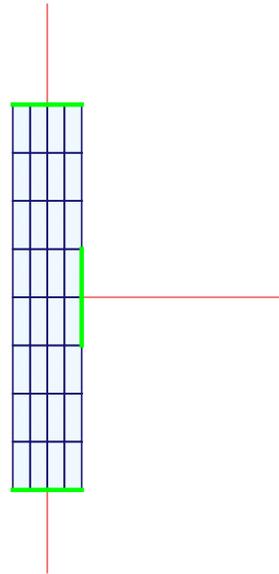


Figure 4.15.: Beam Continuum Model Column-Floor Detail

As an alternative, beam-continuum models were constructed that contained linear and nonlinear Q4 elements around the floor-column joints.

#### 4.5.6 Linear Beam-Continuum Model

The Linear Beam-Continuum model uses beam elements for the majority of the columns, but continuum elements around the connections. Similarly to the pure continuum model, four continuum elements are meshed across the width of the beam. For the rigid floors, a single beam element is used, rather than the large number of continuum elements required in the pure continuum model. Figure 4.14(d) shows the beam-continuum model, with the ends of the column modeled as continuum elements. The model is composed of 490 nodes, 25 beam elements, and 320 continuum elements. Figure 4.15 shows the mesh detail around the beam and column intersection. The dark blue Q4 elements are solved at a small time step (Subdomain B), and the light red beam elements are solved at the large time step. The beam-continuum coupling interactions are shown with thick green lines.

#### 4.5.7 Nonlinear Beam-Continuum Model

The nonlinear beam-continuum model uses the same mesh as the linear beam-continuum model, but with J2 plasticity activated. The model has a yield stress of 500 MPa, and a plastic modulus of 1.66 GPa. These values were chosen to match the experimental data acquired in [105, 106].

#### 4.5.8 Hybrid Models

In addition to the aforementioned models, some hybrid models were constructed as combinations of the aforementioned models. For example, if only the first two floors are expected to go into the nonlinear range, the third, fourth and fifth floors can be modeled with linear elements. If a detailed analysis of how these floors are behaving is not essential, the upper floors can be modeled with the beam and shear models. In the case of RTHS, a truly hierarchical approach can be applied – there is a hard computational limit on the experiment, and the computational resources must be distributed in a way that gives the best results. Figure 4.14(e) shows a hybrid numerical substructure where the second floor is a beam-continuum nonlinear model, and the third, fourth, and fifth floors are the detailed beam model. The model is composed of 104 nodes, 32 beam elements, and 48 continuum elements.

### 4.6 Experimental Validation

Each numerical model was validated through a series of experimental tests. An impact hammer tests was used to determine the natural frequencies of the system, a nonlinear cyclic static test was used to determine hysteretic behavior, and shake table tests were conducted for experimental validation.

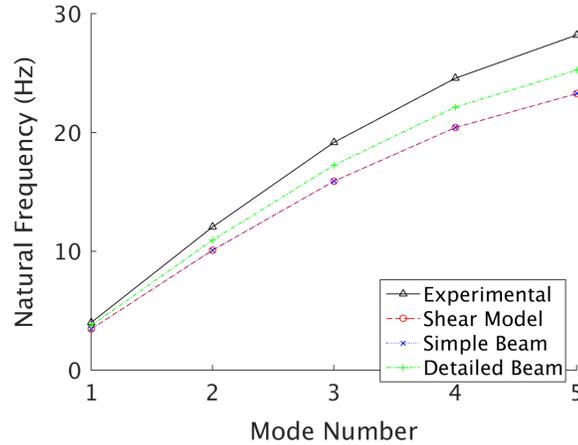


Figure 4.16.: Natural Frequencies of Numerical Models

#### 4.6.1 Impact Hammer Test

The first test was a hammer test, used to determine the natural frequencies of the system. Figure 4.16 shows the undamped natural frequencies of each model, along with the experimental results. In the linear range, both the linear and nonlinear beam-continuum models show the same results. From these results, it can be seen that the experimental structure is slightly stiffer than each numerical models.

#### 4.6.2 Nonlinear Cyclic Static Test

The second test was a static cyclic test, where a quasi static specified displacement was placed on a one story frame, and loaded repeatedly in both directions. Figures 4.17 - 4.21 show the results from these experiments, along with the corresponding results from each model. Figure 4.17 shows that each linear model is shown to matches the slope of the initial loading curve. The nonlinear beam-continuum model replicates the initial slope of the loading curve (elastic stiffness), as well as the initial slope of the the "backbone" curve in the hysteresis, and the loading/unloading behavior of the structure. The nonlinear beam-continuum model does diverge some from the experimental model with a displacement of over 2.5 cm, however the dynamic experiments will fall short

of that linear range. In the event that the experiment does reach such a displacement, the structure will likely fail under the load conditions.

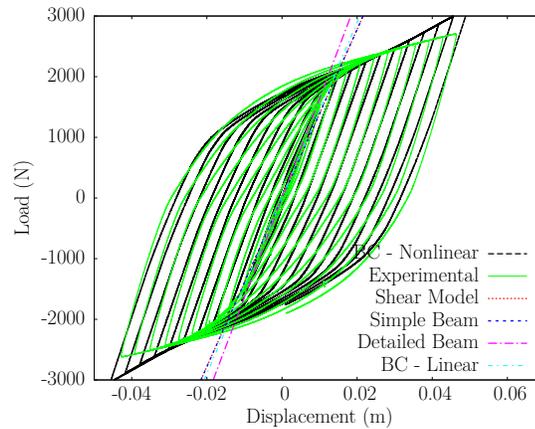


Figure 4.17.: One Story Cyclic Static Test

Figure 4.18 highlights the early non-linearities in the static cyclic test. Note that even though the structure is in the expected linear loading range, the slope of the displacement load curve changes. Three sets of nonlinear cyclic static tests were conducted. Experiment 1 was conducted on an undamaged frame, and Experiments 2 and 3 were conducted on the same frame immediately after the previous tests. Experiments 2 and 3 both show evidence of being damaged by the prior experiments, and both have initial asymmetry.

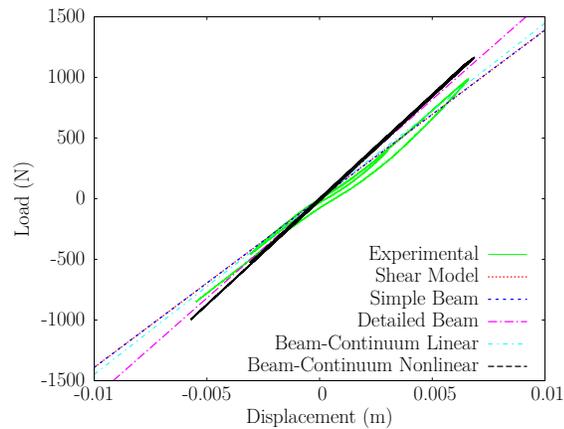


Figure 4.18.: One Story Cyclic Static Test - Linear Range

Figure 4.19 shows the reaction force at the top of the one story frame required to achieve the specified displacement.

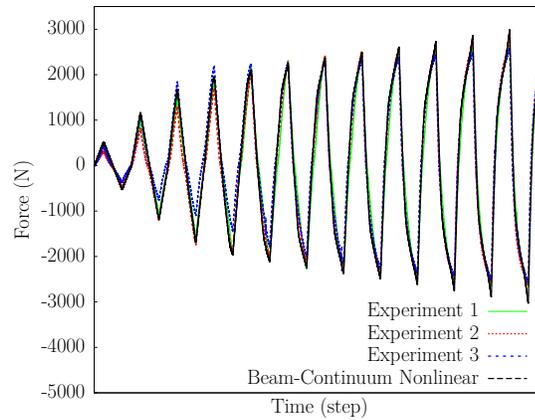


Figure 4.19.: One Story Cyclic Static Test

Figure 4.20 shows a zoomed in version of Figure 4.19, and highlights some initial nonlinear in the experimental models. Each of the first two peaks is well within the expected linear range of the structure, but the reaction force doesn't increase in a linear behavior. This may be due to some "play" in the floor-column joints, where a small amount of motion is allowed with minimal force.

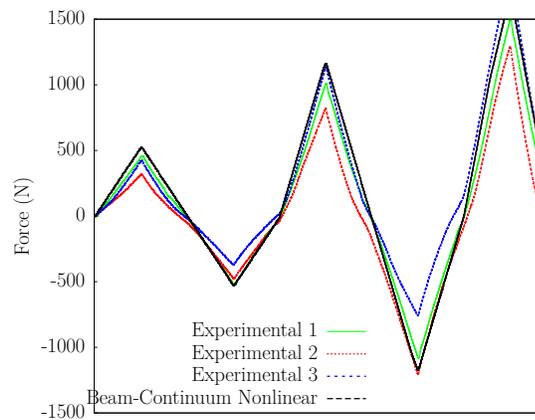


Figure 4.20.: One Story Cyclic Static Test

Figure 4.21 shows the absolute value of the reaction force at each load step, and highlights asymmetry in the physical models. Both Experiment 1 and Experiment 3

show distinct asymmetry in the reaction force, likely originating in the beam-column joints.

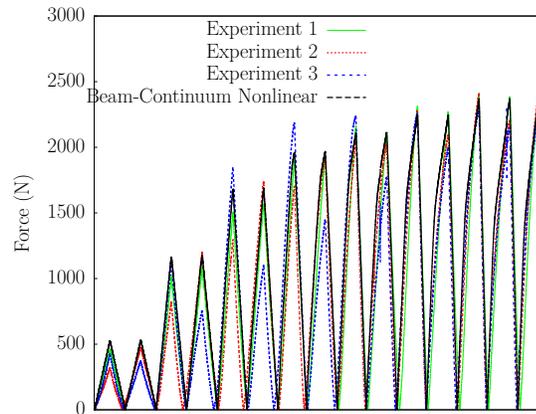


Figure 4.21.: One Story Cyclic Static Test

#### 4.6.3 Five Story Shake Table Tests

For the final validation, the entire five story structure was constructed and placed on a shake table, where it was loaded with scaled versions of the 19xx El-Centro earthquake. Shake Table Test A applied the 58 second El-Centro earthquake loading with a magnitude scaling factor of 0.2. Shake Table Test B applied the 58 second El-Centro earthquake loading with a magnitude scaling factor of 0.5. The results from each floor of Shake Table Test A are shown in figure 4.22. The results from each floor of Shake Table Test B are shown in figure 4.23. The results from each floor of Shake Table Test C are shown in figure 4.24. The results from each floor of Shake Table Test D are shown in figure 4.25.

Shake Table Test C applied the 58 second El-Centro earthquake loading with a magnitude scaling factor of 1.0. Shake Table Test D applied the 58 second El-Centro earthquake loading with a magnitude scaling factor of 2.0. Each of these tests occurred within the linear response range of the structure. Figures 4.26 - 4.28 show the roof displacement from each physical experiment, with the results scaled inversely with the

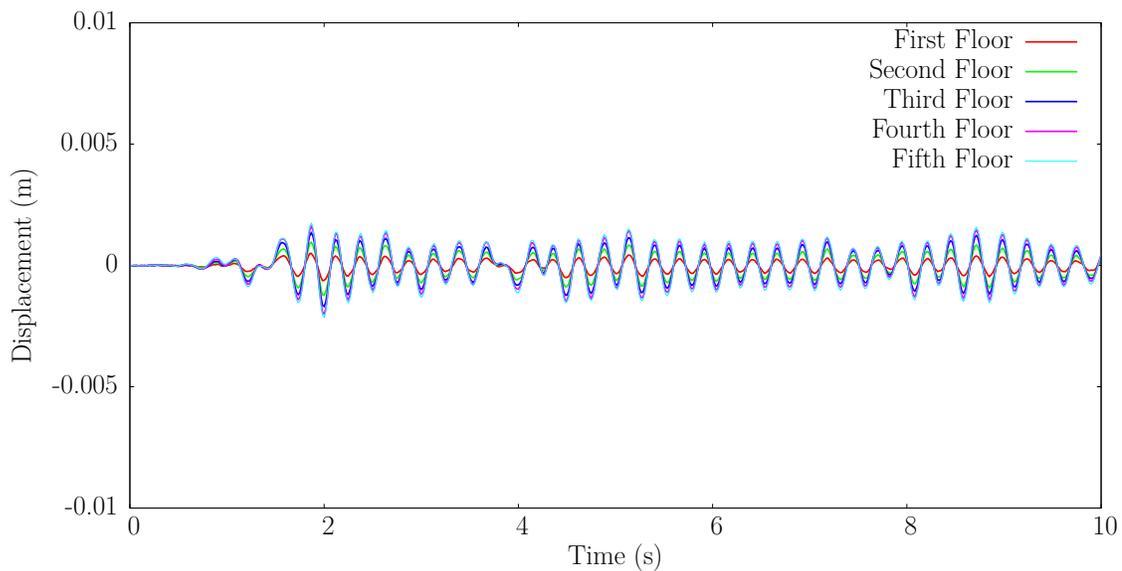


Figure 4.22.: Shake Table Test A Results

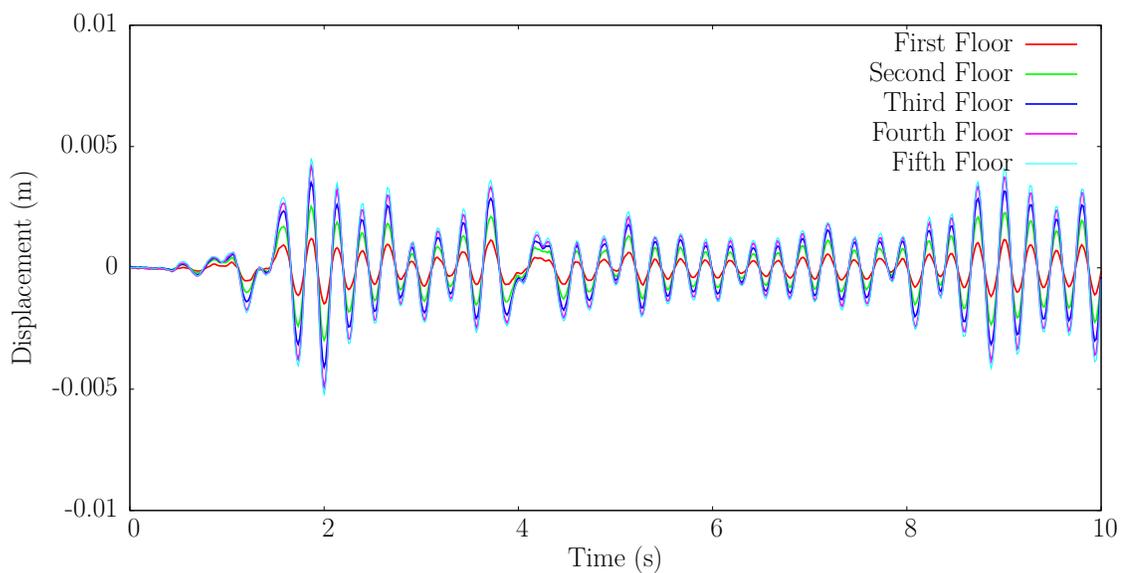


Figure 4.23.: Shake Table Test B Results

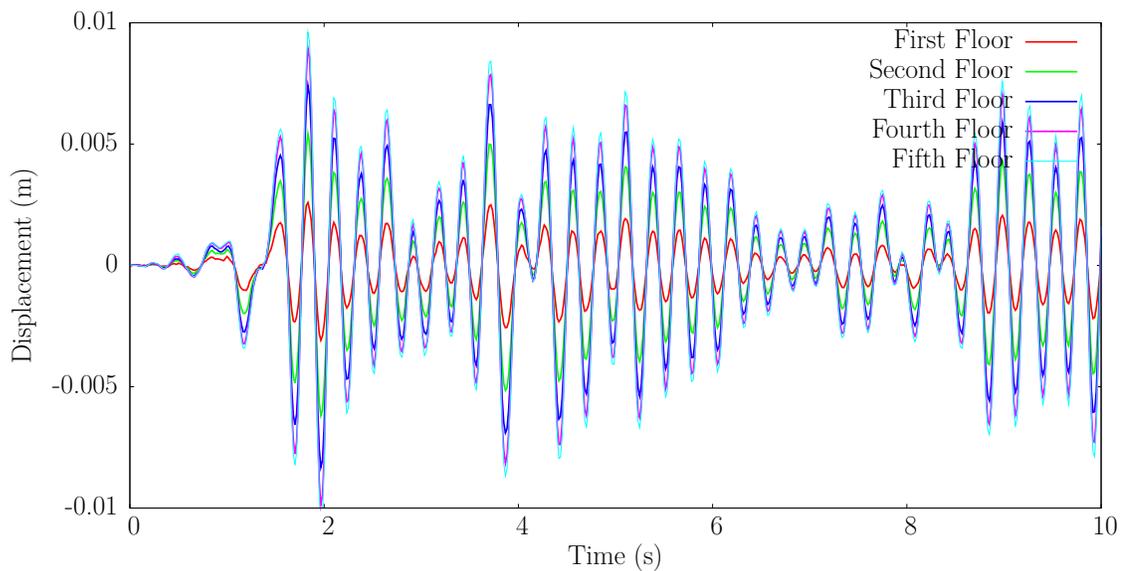


Figure 4.24.: Shake Table Test C Results

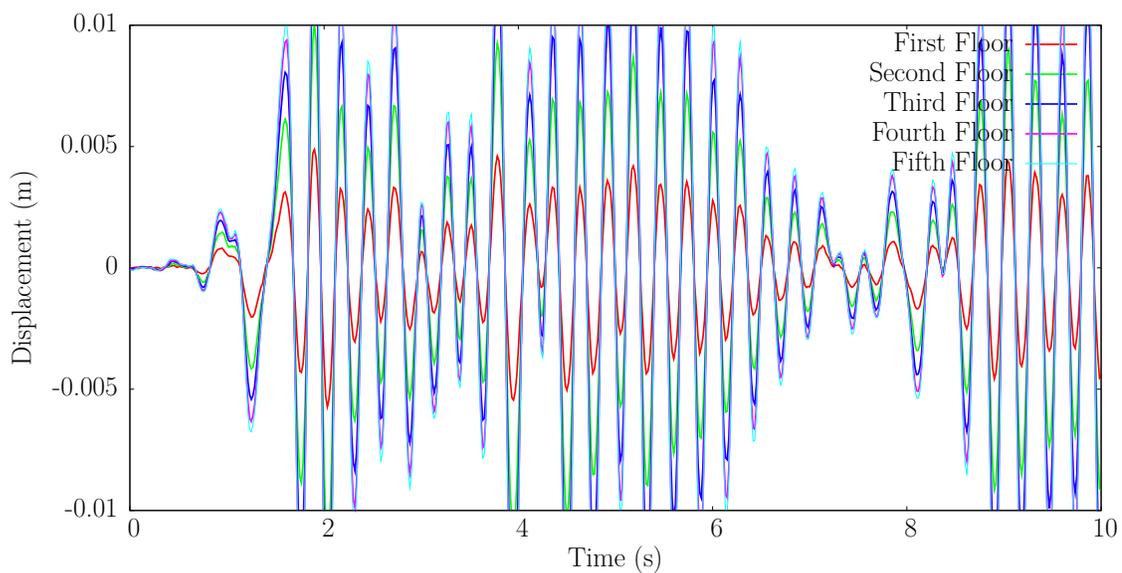


Figure 4.25.: Shake Table Test D Results

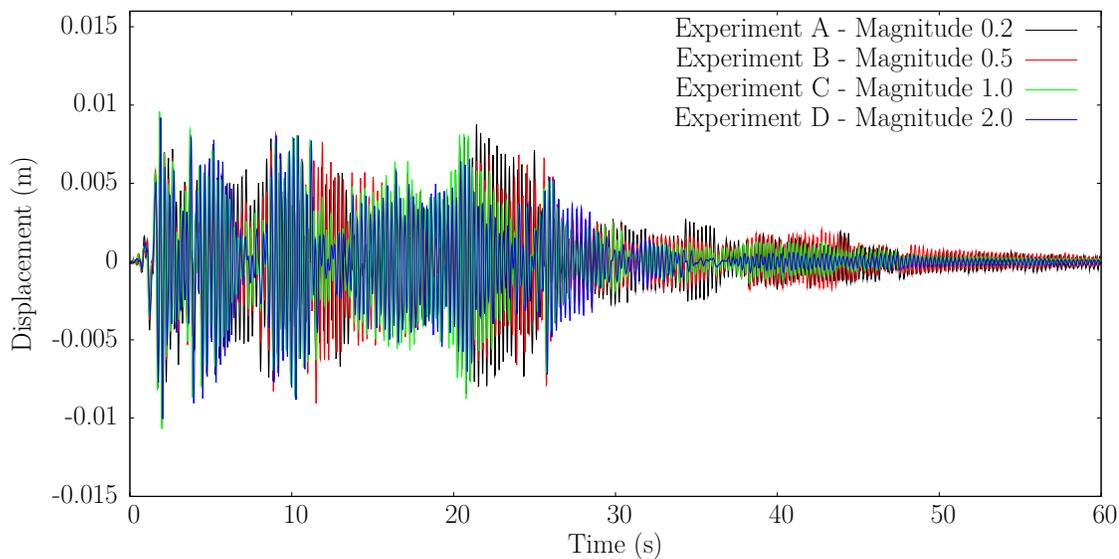


Figure 4.26.: Scaled Experiment Comparison (Roof Displacement)

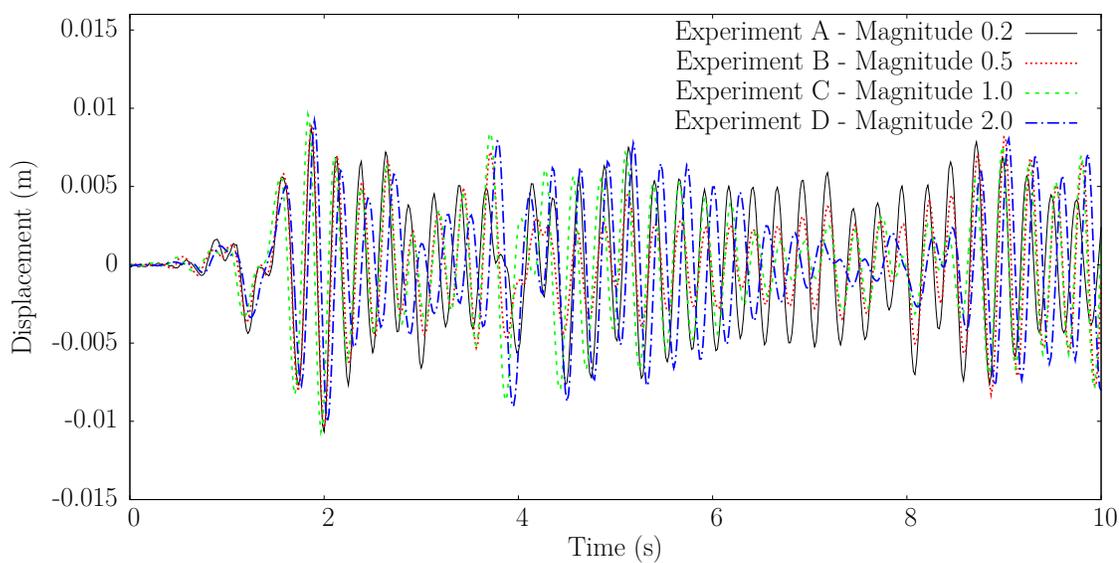


Figure 4.27.: Scaled Experiment Comparison (Roof Displacement)

original loading. It can be seen here that while the peak displacement and behavior at the beginning of the earthquake scales linearly with magnitude, the damping does not. The comparison between experiment A and the other experiments show that at small displacements, the experimental structure moves in a nearly undamped behavior.

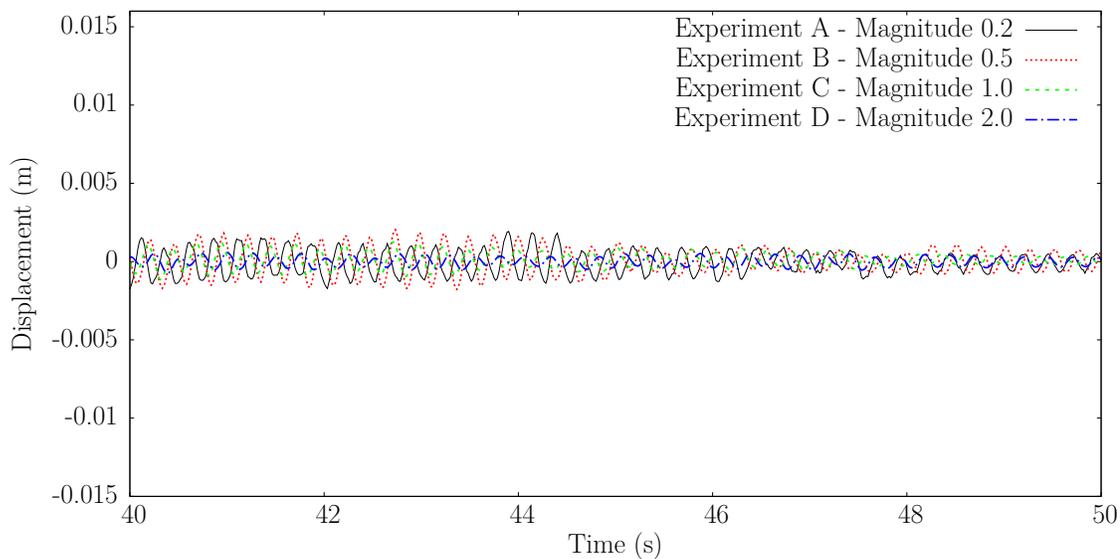


Figure 4.28.: Scaled Experiment Comparison (Roof Displacement)

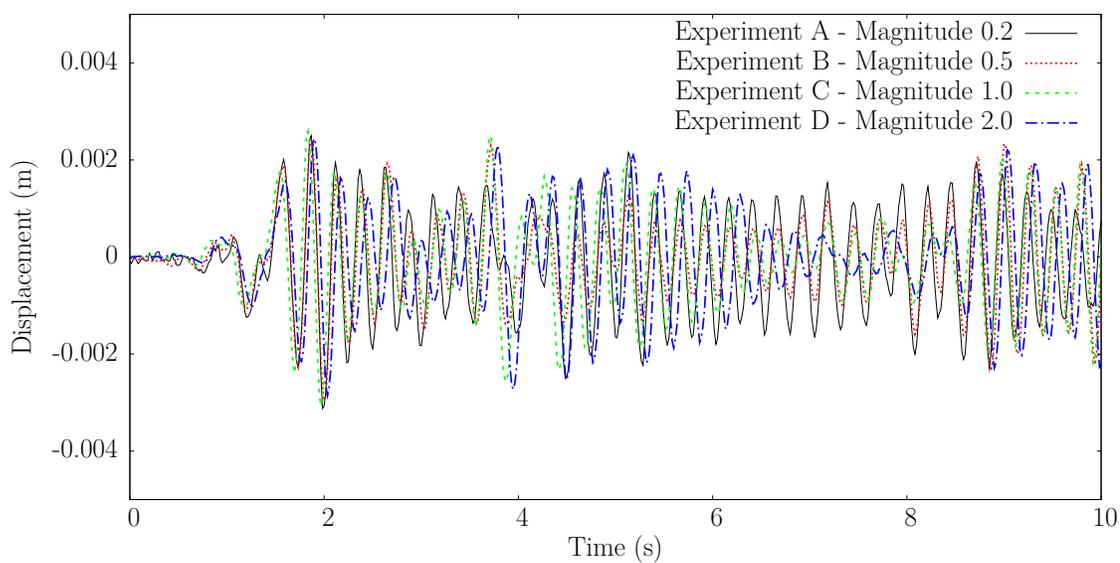


Figure 4.29.: Scaled Experiment Comparison (First Floor Displacement)

Figure 4.29 shows the scaled first floor displacements of the experimental structure. Figure 4.26 and 4.29 highlight the need for RTHS. While the natural frequencies of the structure match well, and the nonlinear static response match well, the numerical models are still unable to capture the dynamic response of the building. Specifically,

the building seems to oscillate at low values of displacement, where these oscillations are damped out in the numerical model. Any inclusion of physical results into the simulation will capture this behavior, even if a full scale experiment is not possible.

#### 4.6.4 Real Time Hybrid Experiment

The building was excited with the El-Centro earthquake with a magnitude to 10%. The Numerical substructure was the top four floor of the building, and the physical substructure was the bottom floors, as shown in Figure 4.30. Displacement based control was used for each experiment. In each experiment, the simulated physical substructure is the nonlinear beam-continuum model, running in isolation as a physical simulated substructure. The only communication between the numerical and physical substructures is through the NIDAQ cards on two different computers, in the form of desired displacement and reaction force. A desired displacement was sent from the numerical substructure to the compensation and control loop, and the measured force was returned from the physical substructure to the numerical substructure, as shown in Figures 4.31 and 4.32.

### 4.7 Results

#### 4.7.1 Pure Numerical Results

Figure 4.33 - 4.35 shows the results from the pure numerical experiments. Figure 4.33 shows the roof displacements for each numerical model. Figure 4.34 shows the first floor displacement, at the interface, for each model. Figure 4.35 shows the interaction force from the physical substructure to the numerical substructure.

#### 4.7.2 RTHS Results

Figures 4.36 - 4.38 show the results from the RTHS experiments with a stiffness member acting as the physical substructure. Figure 4.36 shows the roof displacement

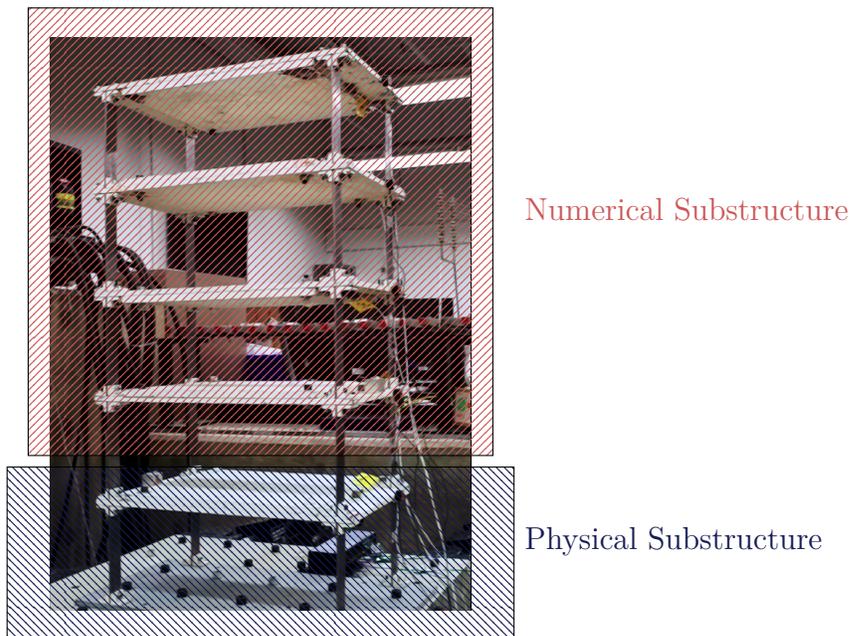


Figure 4.30.: RTHS Setup for Five Story Frame

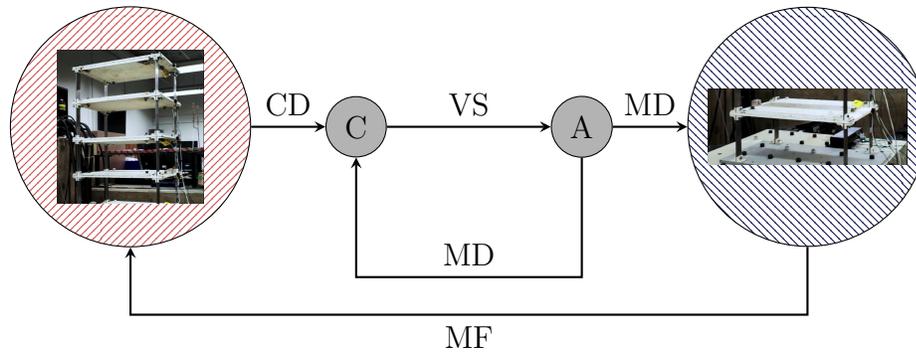


Figure 4.31.: RTHS Control Flow Schematic

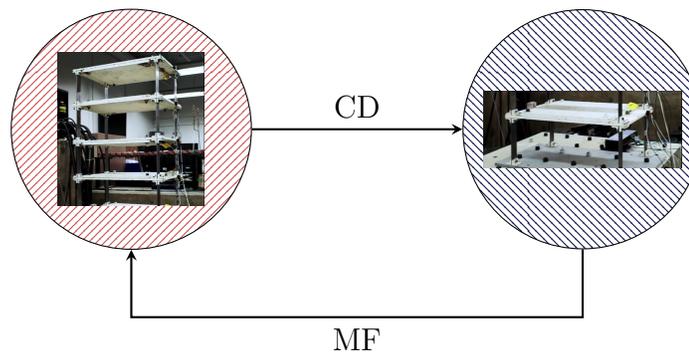


Figure 4.32.: vRTHS Control Flow Schematic

of each model from the RTHS results. Figure 4.37 shows the first floor displacement of the RTHS, at the interaction point. This is the *desired displacement* sent to the physical substructure.

Figure 4.38 shows the interaction force between the first and second floors of the RTHS experiment. This is the *measured force* sent from the physical substructure to the numerical substructure.

#### 4.7.3 Nonlinear Behavior

Figure 4.39 shows the results of the Hybrid model with an earthquake magnitude 1.0, and the scaled results of the Hybrid model with an earthquake magnitude 2.5, demonstrating the nonlinear behavior of the model. The softening of the model at

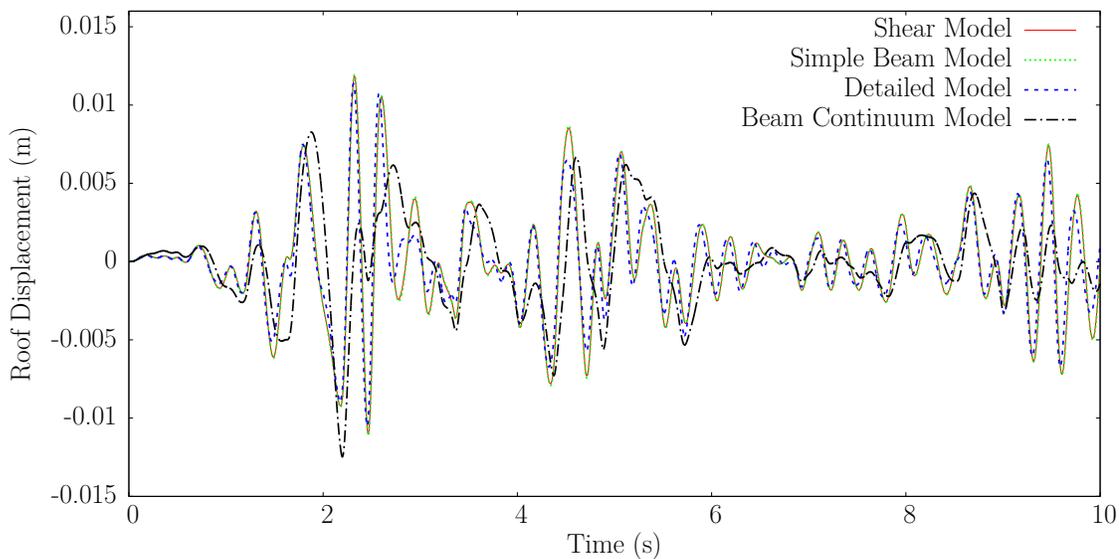


Figure 4.33.: Roof Displacement of Each Model (Numerical)

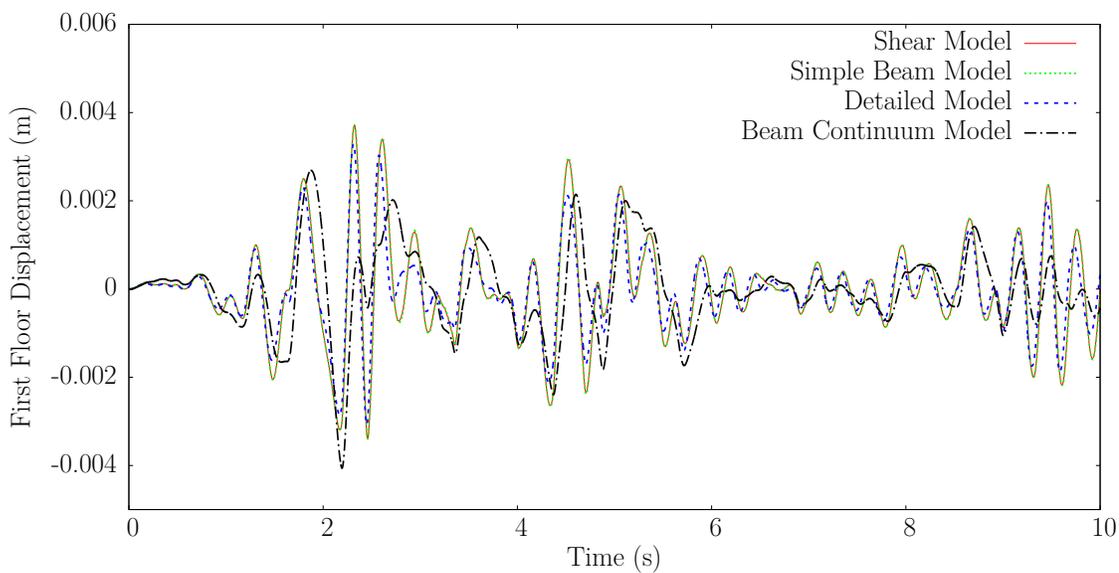


Figure 4.34.: First Floor Displacement (Numerical)

high displacements can be seen as the difference between the peaks of the two curves. If the model were operating in the linear range, the scaled results would match exactly.

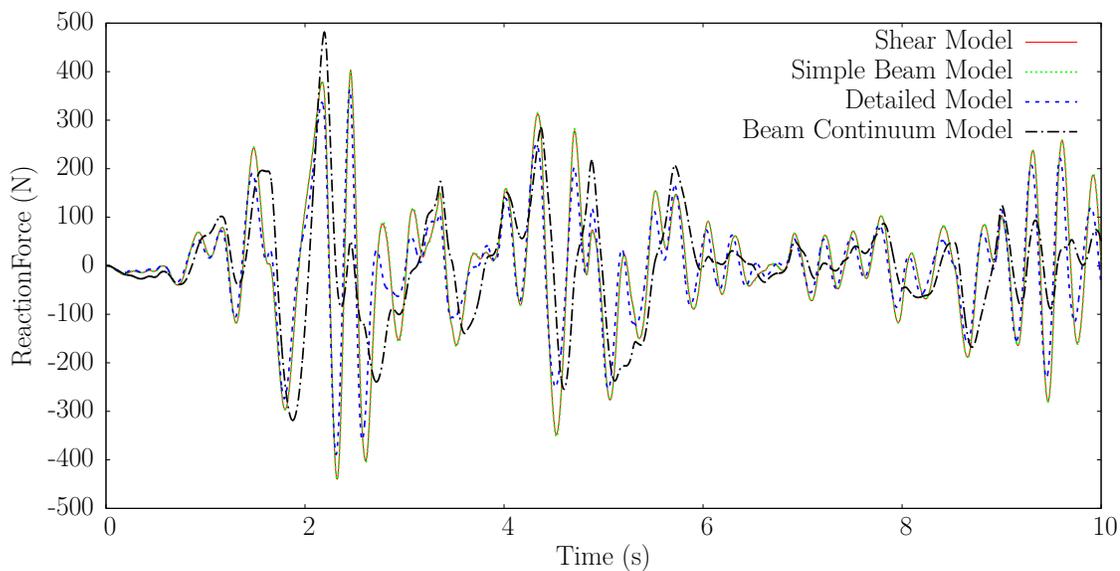


Figure 4.35.: Interaction Force (Numerical)

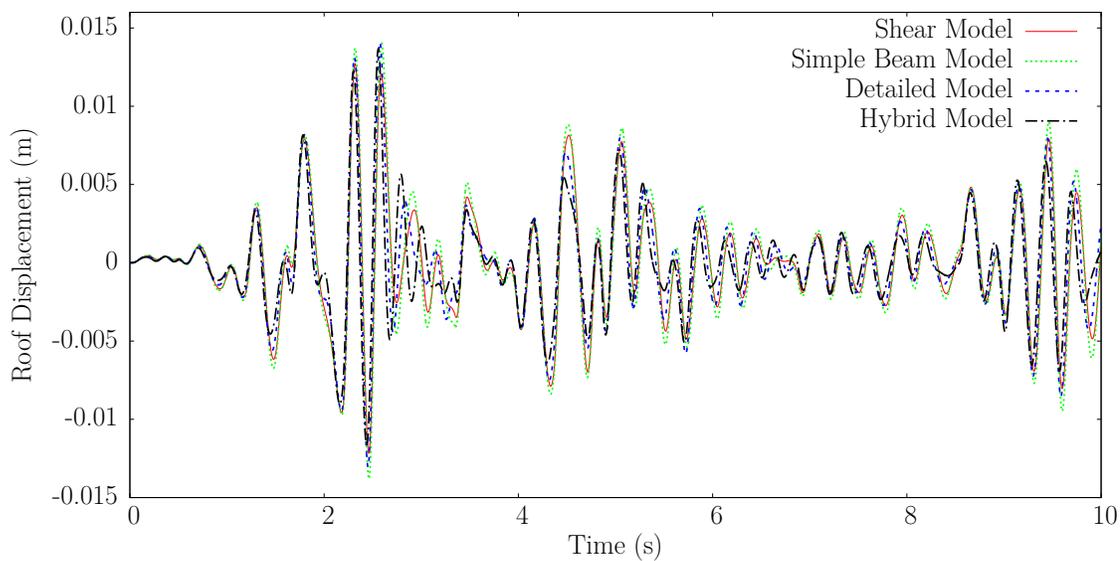


Figure 4.36.: Roof Displacement of Each Model (RTHS)

#### 4.8 Discussion

These results clearly demonstrate both the benefits of both RTHS and high fidelity numerical substructures. Despite the seemingly simple structural design, it is extremely difficult to match all of the experimental results. Even in the small displacement range,

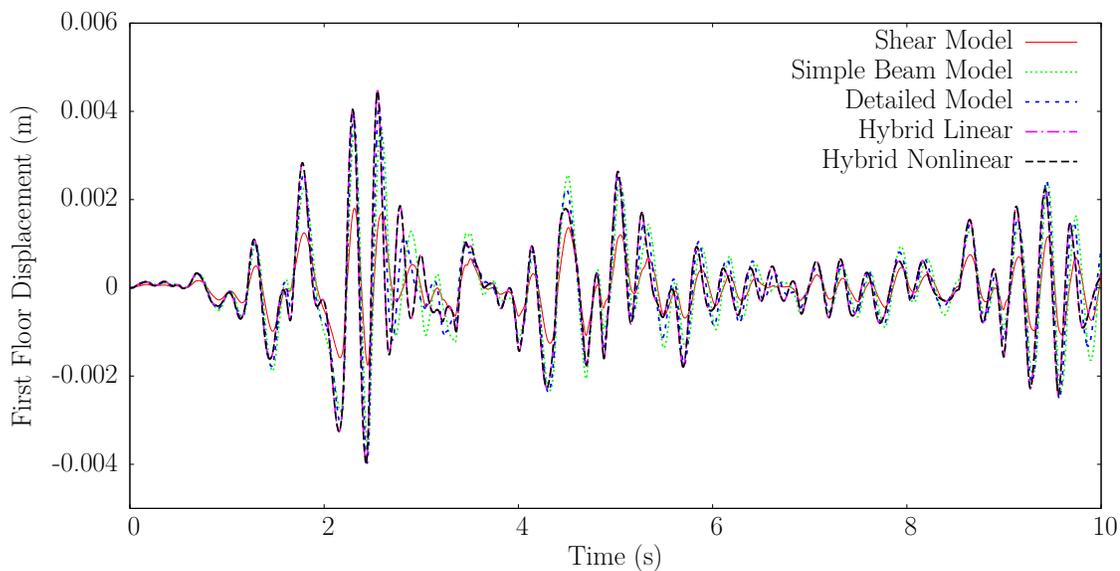


Figure 4.37.: First Floor Displacement (RTHS)

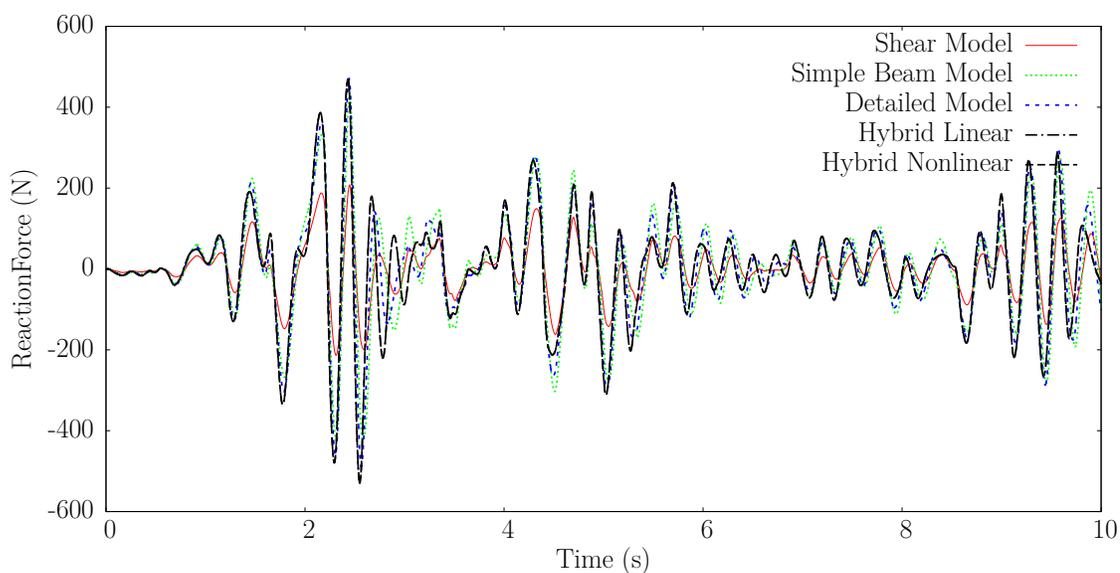


Figure 4.38.: Interaction Force (RTHS)

there is some nonlinear behavior in the connections. While this nonlinearity can be understood in this case by building a physical model, in general this is expensive and infeasible. In these cases, there is a clear benefit to having high fidelity models of the numerical substructure. While no numerical model will match experimental results

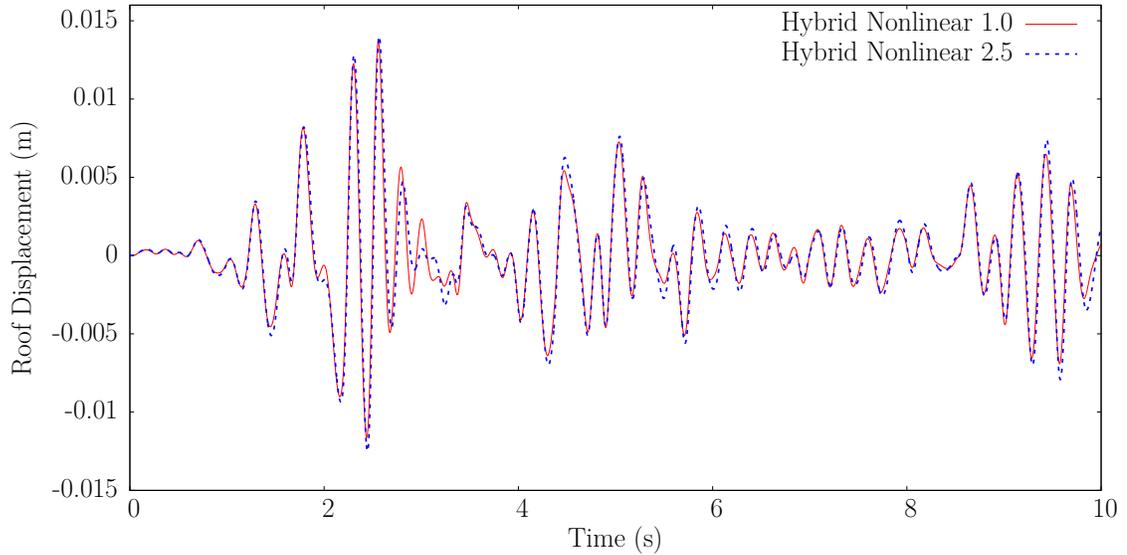


Figure 4.39.: Roof Displacement of Each Model (RTHS)

exactly, it is important to get as high fidelity a model as possible. Real Time Hybrid Simulation allows the coupled numerical-physical experiment to provide better results than a pure numerical experiment. The ability to capture nonlinear results is essential to using high fidelity RTHS.

#### 4.9 Conclusions

In conclusion, it has been demonstrated that advance numerical modeling techniques such as the MTS method and beam-continuum coupling can be combined with nonlinear material models in Real Time Hybrid Simulation. While these techniques are computationally beyond the reach of traditional RTHS, careful modeling and algorithm selection can allow for nonlinear behavior in the numerical substructure. It has been demonstrated that these numerical models give better results than shear models and other small linear models.

## 5. PARALLEL ELLIPSOIDAL PERFECTLY MATCHED LAYERS FOR ACOUSTIC HELMHOLTZ PROBLEMS ON EXTERIOR DOMAINS

Exterior acoustic problems occur in a wide range of applications, making the finite element analysis of such problems a common practice in the engineering community. Various methods for truncating infinite exterior domains have been developed, including absorbing boundary conditions, infinite elements, and more recently, perfectly matched layers (PML). Perfectly matched layers are gaining popularity due to their generality, ease of implementation, and effectiveness as an absorbing boundary condition. PML formulations have been developed in Cartesian, cylindrical, and spherical geometries, but not ellipsoidal. Some recent studies have demonstrated the solution of the PML Helmholtz problem in parallel. In this study we examine the conditioning and performance of the PML Helmholtz problem, and we compare with an infinite element approach that is based on high order basis functions.

To enable a fair comparison between PML and infinite elements, we perform a We show that, because of the poor conditioning related to the high order basis functions, the number of Helmholtz solver iterations

In this paper we present a parallel, ellipsoidal PML formulation for acoustic Helmholtz problems. performance To facilitate the meshing process, the ellipsoidal PML layer is generated by a simple extrusion. Though the complex stretching is defined along ellipsoidal contours, we modify the Jacobian to include an additional mapping back to Cartesian coordinates in the weak formulation of the finite element equations. This allows the equations to be solved in Cartesian coordinates, which is more compatible with existing finite element software. As essentially any shape can be efficiently embedded in a minimal volume ellipsoid, ellipsoidal PML provides an

efficient domain truncation strategy that does not have the extra complications of corners.

We present massively parallel implementations of both PML and infinite elements on ellipsoidal domains, and compare their performance on representative exterior acoustic problems. We also examine the conditioning of the linear systems generated by the two techniques by examining the number of Krylov-iterations needed for convergence to a fixed solver tolerance.

## 5.1 Introduction

Perfectly Matched Layers (PML) were originally developed in 1994 [20] for simulating electromagnetic waves on exterior domains. This method provides an elegant and effective means of terminating a finite element acoustic mesh without reflections. A layer of elements is added around the acoustic domain that, at least on the continuous level, absorbs waves of all frequencies and of all angles of incidence.

Recent research efforts by Hohage et al [107] and Lassas et al [108, 109] have proven the convergence of the exterior acoustic problem with PML boundaries as the thickness of the PML layer tends to infinity. These references provide a theoretical basis for convergence of the PML methods in general. However, in practice one would like to use the smallest (most efficient) acoustic mesh, such as an ellipsoidal mesh, and truncate with a finite-thickness PML.

Despite the versatility of PML, some gaps currently exist in the literature for these methods. First, though infinite elements [110–118] and absorbing boundary conditions [119] have been developed for ellipsoidal domains, existing PML formulations have been developed for Cartesian, cylindrical, and spherical domains [21–23, 120, 121]. For many applications that involve long, slender structures, an ellipsoidal formulation would allow for the PML to more naturally match the acoustic meshes of interest, and would avoid the extra complications of corners and edges in the formulation. In time-domain PML, corners are an extra complication that requires additional computation.

Additionally, Cartesian PML can be viewed as a special case of curvilinear PML where each flat face of the Cartesian PML is constructed from a curvilinear PML with a very large radius of curvature. Thus the curvilinear approach is general and offers significant versatility.

Parallel implementations of PML, and an understanding of the conditioning of and solution of the resulting system of equations with parallel iterative solvers, are additional gaps in the PML literature. Given that many structural acoustic systems result in large linear systems, parallel implementations and preconditioners are highly desirable for these problems.

This paper presents a parallel formulation and implementation of perfectly matched layers in an ellipsoidal coordinate system. Given an interior, unstructured acoustic mesh, perfectly matched layers are extruded into a structured mesh that surrounds the acoustic domain of interest. This conformal meshing strategy allows for easy manipulation of PML thickness, and could also be extended to non-conformal PML layers, wherein the PML is embedded in meshes with different underlying geometry [121]. In order to facilitate the implementation in a finite element code that is based on Cartesian coordinates, we present a mapped PML in which the weak formulation is mapped back to Cartesian coordinates. This follows recent work for spherical and cylindrical PML [22, 121], and allows the PML parameters such as thickness and number of elements to be changed without the need to remesh the entire domain, which can be computationally intensive in large models. The PML is implemented in parallel, and compared with the absorbing boundary condition and the infinite element solution for both accuracy and linear solver performance.

## 5.2 Theory

Given a structure  $S$  surrounded by bounded interior domain  $\Omega_i$ , and an exterior domain  $\Omega_e$ , the exterior acoustics problem consists of determining the acoustic pressure,  $p$ , in domain  $\Omega_e \cup \Omega_i$ . We refer to Figure 5.1 for a schematic of the geometry. In

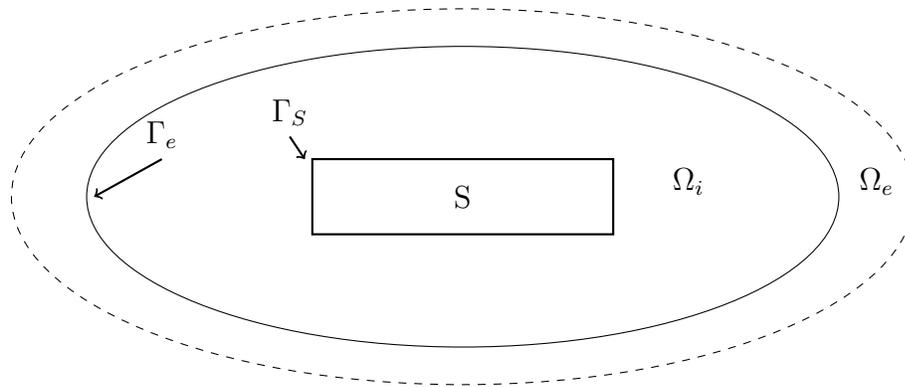


Figure 5.1.: Domains  $\Omega_i$  and  $\Omega_e$  and interface  $\Gamma$  for the exterior acoustic problem.

a domain truncation strategy, boundary conditions are applied to the outermost boundary  $\Gamma_e$  of  $\Omega_i$ . Several strategies can be used for this purpose, including absorbing boundary conditions, infinite elements, and PML. In this section we derive formulations for ellipsoidal PML. To facilitate the development of the theory, we start by first presenting formulations for Cartesian, rotated Cartesian, and spherical PML.

To illustrate the ideas, we assume an acoustic pressure wave propagating in the  $x$ -direction, with wavenumber  $k = \frac{\omega}{c}$ , where  $\omega$  is the circular frequency, and  $c$  is the speed of sound. The wave takes the form

$$p(x) = p_0 e^{ikx} \quad (5.1)$$

As written, this wave is undamped, and will propagate indefinitely with no change of shape. However, if we allow the wave to propagate on a coordinate system that has *complex* coordinates  $\tilde{x} = a(x) + ib(x)$ , where  $a(x)$  and  $b(x)$  are functions of  $x$ , then the equation of the wave becomes [122]

$$p(\tilde{x}) = p_0 e^{ik\tilde{x}} = p_0 e^{i(-ka(x) + ikb(x))} = p_0 e^{-kb(x)} e^{ika(x)} \quad (5.2)$$

We observe that this wave now corresponds to *damped* wave propagation, with decay coefficient equal to  $kb(x)$ . This wave will decay exponentially fast.

In order for equation 5.2 to be a solution to a wave equation, that wave equation must itself be written in a coordinate system that is complex, rather than real-valued. On the other hand, the corresponding finite element implementation is most easily derived on a real-valued coordinate system. Thus, though the governing partial differential equations of the PML are written in a complex coordinate field, the corresponding weak formulation is mapped to a real coordinate system, to facilitate the finite element implementation.

In order to build up to the ellipsoidal PML formulation, the following sections provide derivations of rectangular, rotated rectangular, and spherical PML. These provide the building blocks for the ellipsoidal case. We will subsequently show that the ellipsoidal formulation reduces to the spherical and rectangular cases under certain conditions, and thus includes these formulations as special cases.

#### Cartesian PML

We define the PML domain as being a parallelepiped of dimension  $(2a, 2b, 2c)$ , centered at the origin, with an interior parallelepiped hole of dimension  $(2\bar{a}, 2\bar{b}, 2\bar{c})$ . Practically, this would correspond to the case where the structure of interest, as complex shape it may have, was surrounded by an acoustic mesh that terminated at the boundary of the inner parallelepiped. The PML would then occupy the region between the inner and outer parallelepiped boundaries. A simple shift can be applied if the domain is not origin-centered.

**Step 1. Analytic continuation.** The PML equations can be written in either first or second order form. Here we consider the implementation of second order form. In the interior  $\Omega = \Omega_I$ , the acoustic pressure must satisfy the acoustic Helmholtz equation

$$-\Delta p - k^2 p = 0 \tag{5.3}$$

where  $k = \frac{\omega}{c}$ , and  $p$  is the acoustic pressure, a prescribed Neumann boundary condition on  $\Gamma_S$

$$\frac{\partial p}{\partial n} = g(x, \omega) \quad (5.4)$$

and the Sommerfeld radiation condition at infinity [22]

$$\frac{\partial p}{\partial n} + ikp \in L^2(\Omega_E) \quad (5.5)$$

where  $k = \frac{\omega}{c}$ . We note that equation 5.3 involves constant coefficients, meaning that the speed of sound and density in the fluid are assumed to be constant. More specifically, equation 5.3 is undamped, meaning that the waves will not attenuate as they propagate through the medium.

Equation 5.3 is written in terms of real coordinates. As illustrated earlier, in order to have the waves decay in the PML, we need to have the waves propagate on a complex grid rather than a real-valued grid. Thus, we use *analytic continuation* to map the Helmholtz equation into the complex plane

$$-\tilde{\Delta}p - k^2p = 0 \quad (5.6)$$

where the change of coordinates is defined as

$$\tilde{x} = x - \frac{i}{\omega} \int_x^a \sigma(\xi) d\xi \quad a < x < \bar{a} \quad (5.7)$$

$$\tilde{x} = x + \frac{i}{\omega} \int_a^x \sigma(\xi) d\xi \quad -\bar{a} < x < -a \quad (5.8)$$

**Step 2. Weak formulation over complex-valued domain.** We note that the weak formulation of equation 5.6 can be constructed using either a *bilinear* or *sesquilinear* formulation [123, 124]. The difference is only whether complex conjugation is applied to the test functions. In standard finite element methods for acoustics, these formulations lead to the same discrete system of equations. However, with PML the formulations yield different numerical methods. In this paper we take the bilinear

approach, since it yields a complex-symmetric system of linear equations that can be exploited in the linear solver. The bilinear weak form of equation 5.6 seeks  $p \in V_f(\tilde{\Omega}_I)$  such that

$$\int_{\tilde{\Omega}_I} \langle \tilde{\nabla} p, \tilde{\nabla} q \rangle - k^2 p q \, d\tilde{\Omega}_I = \int_{\tilde{\Gamma}_S} g q \, d\tilde{\Gamma}_S \quad (5.9)$$

where the tildes indicate quantities defined over the complex extension of the domain  $\Omega_I$ .

**Step 3: Apply the chain rule.** From equations 5.8 and the Fundamental Theorem of Calculus, we see that

$$\frac{\partial \tilde{x}}{\partial x} = \gamma_x(x) = 1 + \frac{i}{\omega} \sigma(x) \quad (5.10)$$

Similar expressions hold for the  $y$  and  $z$  coordinates. This implies that the gradients of acoustic pressure can be transformed between the real and complex domains using a Jacobian

$$\nabla p = \mathbf{J}_{cart} \tilde{\nabla} p \quad (5.11)$$

where the Jacobian matrix for the Cartesian coordinate system  $\mathbf{J}_{cart}$  is defined as

$$\mathbf{J}_{cart} = \begin{bmatrix} \gamma_x & 0 & 0 \\ 0 & \gamma_y & 0 \\ 0 & 0 & \gamma_z \end{bmatrix} \quad (5.12)$$

Conversely, we can map from the complex to the real derivatives using the inverse of the Jacobian.

$$\tilde{\nabla} p = \mathbf{J}_{cart}^{-1} \nabla p \quad (5.13)$$

where

$$\mathbf{J}_{cart}^{-1} = \begin{bmatrix} \frac{1}{\gamma_x} & 0 & 0 \\ 0 & \frac{1}{\gamma_y} & 0 \\ 0 & 0 & \frac{1}{\gamma_z} \end{bmatrix} \quad (5.14)$$

The scale factor that maps  $\tilde{\Omega}_I$  into  $\Omega_I$  is simply the determinant of the Jacobian,

$$W_{cart} = \gamma_x \gamma_y \gamma_z \quad (5.15)$$

**Step 4: Revert to real-valued weak formulation.** Using the previous results and the determinant relation from equation 5.15, the corresponding weak version of the Helmholtz equation is given as follows. Find  $p \in V_f(\Omega_I)$  such that

$$\int_{\Omega_I} [(\mathbf{J}_{cart}^{-1} \nabla p) \cdot (\mathbf{J}_{cart}^{-1} \nabla q) - k^2 pq] W_{cart} d\Omega_I = \int_{\Gamma_S} g q dS. \quad (5.16)$$

We note that we can turn this into a Helmholtz equation with variable coefficients as follows

$$\int_{\Omega_I} \mathbf{A} \langle \nabla p, \nabla q \rangle - k^2 W_{cart} p q d\Omega_I = \int_{\Gamma_S} g q d\Gamma_S \quad (5.17)$$

where  $\mathbf{A} = W_{cart} \mathbf{J}_{cart}^{-1} \mathbf{J}_{cart}^{-T}$ . We note that  $\mathbf{A}$  is a symmetric matrix, which follows from our choice to use a bilinear formulation rather than sesquilinear.

Note that equation 5.17 achieves all of the goals that were set from the beginning - a symmetric weak formulation over the real-valued domain, but with built-in dissipative properties stemming from the transformation to complex coordinates.

In the following sections, we will derive PML equations for rotated Cartesian, spherical, and ellipsoidal coordinates. In all cases, the weak formulation will be precisely the same as in equation 5.17, but with a different Jacobian matrix  $\mathbf{J}$  and corresponding determinate  $W$ . Thus, we will only derive expressions for  $\mathbf{J}$  in each of the coordinate systems.

#### Rotated Cartesian Coordinates

In this section we consider the case where the PML surface is extruded from a flat plane that is oriented at an arbitrary angle in three-dimensional space. If we

define  $\mathbf{x} = x_i, i = 1, 2, 3$  as the unrotated coordinates and  $\mathbf{x}' = x'_i, i = 1, 2, 3$  as the coordinates in the rotated coordinate system, we have

$$\mathbf{R} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (5.18)$$

where  $a_{ij}$  is the direction cosine between the  $x_i$  and  $x'_i$  axis. This defines the transformation as follows

$$\mathbf{x}' = \mathbf{R}\mathbf{x} \quad (5.19)$$

The Jacobian matrix for this case can be computed from the chain rule [121]

$$\mathbf{J}_{rotcart} = \frac{\partial(\tilde{x}, \tilde{y}, \tilde{z})}{\partial(x, y, z)} = \frac{\partial(\tilde{x}, \tilde{y}, \tilde{z})}{\partial(x', y', z')} \frac{\partial(x', y', z')}{\partial(x, y, z)} = \begin{bmatrix} \gamma_x & 0 & 0 \\ 0 & \gamma_y & 0 \\ 0 & 0 & \gamma_z \end{bmatrix} \mathbf{R} = \mathbf{J}_{cart} \mathbf{R} \quad (5.20)$$

The inverse of this matrix is given as

$$\mathbf{J}_{rotcart}^{-1} = \mathbf{R}^T \mathbf{J}_{cart}^{-1} \quad (5.21)$$

Thus, the coefficient matrix for this case is given by

$$\begin{aligned} \mathbf{A} &= W_{rotcart} \mathbf{J}_{rotcart}^{-1} \mathbf{J}_{rotcart}^{-T} \\ &= W_{rotcart} \mathbf{R}^T \mathbf{J}_{cart}^{-1} (\mathbf{J}_{cart} \mathbf{R})^{-T} \\ &= W_{cart} \mathbf{R}^T \mathbf{J}_{cart}^{-1} \mathbf{J}_{cart}^{-1} \mathbf{R} \end{aligned} \quad (5.22)$$

where we have used the fact that  $W_{rotcart} = W_{cart}$ . We see that this involves a simple rotation tensor transformation applied to the diagonal Jacobian matrix given in the unrotated case, equation 5.14. Thus, equation 5.17 applies, and can be used to construct the weak formulation in the rotated Cartesian case, but with a modified coefficient matrix  $\mathbf{A}$  given in equation 5.22.

## Spherical Coordinates

In a similar manner, we can derive the Jacobian matrix for a spherical PML. Though other researchers [120,125] have chosen to solve the spherical PML equations directly in spherical coordinates, we prefer to map the equations back to the Cartesian system to facilitate the finite element implementation. Thus, in this case our Jacobian needs to account for this additional transformation. The formulation for this case is given in [121]. The mapping from spherical to Cartesian coordinates is given as

$$\begin{aligned}x &= r \sin(\phi) \cos(\theta) \\y &= r \sin(\phi) \sin(\theta) \\z &= r \cos(\phi)\end{aligned}\tag{5.23}$$

The corresponding analytically continued coordinates are given as

$$\begin{aligned}\tilde{x} &= \tilde{r} \sin(\phi) \cos(\theta) \\\tilde{y} &= \tilde{r} \sin(\phi) \sin(\theta) \\\tilde{z} &= \tilde{r} \cos(\phi)\end{aligned}\tag{5.24}$$

With these definitions the Jacobian matrix is given by the chain rule

$$\begin{aligned}\mathbf{J}_{spherical} &= \frac{\partial(\tilde{x}, \tilde{y}, \tilde{z})}{\partial(x, y, z)} = \frac{\partial(\tilde{x}, \tilde{y}, \tilde{z})}{\partial(r, \phi, \theta)} \frac{\partial(x, y, z)}{\partial(r, \phi, \theta)}^{-1} \\&= \begin{bmatrix} \tilde{r}' \sin(\phi) \cos(\theta) & \tilde{r} \cos(\phi) \cos(\theta) & -\tilde{r} \sin(\phi) \sin(\theta) \\ \tilde{r}' \sin(\phi) \sin(\theta) & \tilde{r} \cos(\phi) \sin(\theta) & \tilde{r} \sin(\phi) \cos(\theta) \\ \tilde{r}' \cos(\phi) & -\tilde{r} \sin(\phi) & 0 \end{bmatrix} \\&\quad \begin{bmatrix} \sin(\phi) \cos(\theta) & r \cos(\phi) \cos(\theta) & -r \sin(\phi) \sin(\theta) \\ \sin(\phi) \sin(\theta) & r \cos(\phi) \sin(\theta) & r \sin(\phi) \cos(\theta) \\ \cos(\phi) & -r \sin(\phi) & 0 \end{bmatrix}^{-1}\end{aligned}\tag{5.25}$$

Once again, equation 5.17 applies, and can be used to construct the weak formulation in the case of spherical coordinates, but with a modified coefficient matrix  $\mathbf{A}$  given in equation 5.25.

We note that an advantage of the curvilinear PML formulation is that it is one-dimensional in the sense that the stretching only happens in one of the coordinate directions, in this case the radial direction. Thus, we simply can define the stretching as being in the radial direction only. This takes the form

$$\tilde{r} = r + i \int_R^r \sigma(\epsilon) d\epsilon \quad (5.26)$$

which implies that

$$\tilde{r}' = \frac{\partial \tilde{r}}{\partial r} = \gamma(r) = 1 + i\sigma(r) \quad (5.27)$$

### Ellipsoidal Coordinates

In the case of ellipsoidal coordinates, we first must choose an appropriate coordinate system for the complex stretching of the PML. Ellipsoidal coordinates can be expressed in various ways, but we have found use of the coordinates developed by Burnett [114] to be the most convenient for defining the PML. As in the spherical case, we prefer to solve the final equations in Cartesian coordinates rather than ellipsoidal. Thus, we will apply complex stretching to the ellipsoidal coordinate system, but will map the resulting equations back to Cartesian coordinates for the finite element solution. Once again, all of these transformations can be applied with the Jacobian.

We define an ellipsoidal radius [114] as

$$r = \frac{c_1 + c_2}{2} \quad (5.28)$$

where  $c_1$  and  $c_2$  are the distances of a given point on the ellipse to the two foci. We note that on the ellipsoidal surface,  $r$  is a constant, and is essentially a generalization of the notion of radial distance in the case of a sphere. Given the major and minor

radii  $a$  and  $b$  of the ellipse, the distance to the focus along the major axis is given by  $f = \sqrt{a^2 - b^2}$ .

In terms of PML, we choose the direction of complex stretching to be along the direction defined in equation 5.28. We note that unlike the radial direction for a sphere, equation 5.28 defines curvilinear lines, and thus the PML layer will produce damping along those directions. This is necessary since if we were to define damping along straight-line paths (say in the direction normal to the ellipsoid surface), then the complex stretching would occur in all three directions  $r$ ,  $\phi$ ,  $\theta$ .

Given these parameters, the ellipsoidal coordinate system is defined as

$$\begin{aligned} x &= \sqrt{r^2 - f^2} \sin(\phi) \cos(\theta) \\ y &= \sqrt{r^2 - f^2} \sin(\phi) \sin(\theta) \\ z &= r \cos(\phi) \end{aligned} \tag{5.29}$$

Note that in the case of a sphere,  $a = b$  which implies that  $f = 0$ , and these coordinates reduce to the spherical case. The stretched coordinates in the ellipsoidal case are given by

$$\begin{aligned} \tilde{x} &= \sqrt{\tilde{r}^2 - f^2} \sin(\phi) \cos(\theta) \\ \tilde{y} &= \sqrt{\tilde{r}^2 - f^2} \sin(\phi) \sin(\theta) \\ \tilde{z} &= \tilde{r} \cos(\phi) \end{aligned} \tag{5.30}$$

This implies that the transformation matrix is given as

$$\begin{aligned}
 \mathbf{J}_{\text{ellipsoidal}} &= \frac{\partial(\tilde{x}, \tilde{y}, \tilde{z})}{\partial(x, y, z)} = \frac{\partial(\tilde{x}, \tilde{y}, \tilde{z})}{\partial(r, \phi, \theta)} \frac{\partial(x, y, z)}{\partial(r, \phi, \theta)}^{-1} \\
 &= \begin{bmatrix} \frac{\tilde{r}\tilde{r}'}{\sqrt{\tilde{r}^2 - f^2}} \sin(\phi) \cos(\theta) & \sqrt{\tilde{r}^2 - f^2} \cos(\phi) \cos(\theta) & -\sqrt{\tilde{r}^2 - f^2} \sin(\phi) \sin(\theta) \\ \frac{\tilde{r}\tilde{r}'}{\sqrt{\tilde{r}^2 - f^2}} \sin(\phi) \sin(\theta) & \sqrt{\tilde{r}^2 - f^2} \cos(\phi) \sin(\theta) & \sqrt{\tilde{r}^2 - f^2} \sin(\phi) \cos(\theta) \\ \tilde{r}' \cos(\phi) & -\tilde{r} \sin(\phi) & 0 \end{bmatrix} \\
 &\quad \begin{bmatrix} \frac{r}{\sqrt{r^2 - f^2}} \sin(\phi) \cos(\theta) & \sqrt{r^2 - f^2} \cos(\phi) \cos(\theta) & -\sqrt{r^2 - f^2} \sin(\phi) \sin(\theta) \\ \frac{r}{\sqrt{r^2 - f^2}} \sin(\phi) \sin(\theta) & \sqrt{r^2 - f^2} \cos(\phi) \sin(\theta) & \sqrt{r^2 - f^2} \sin(\phi) \cos(\theta) \\ \cos(\phi) & -r \sin(\phi) & 0 \end{bmatrix}^{-1} \quad (5.31)
 \end{aligned}$$

Ellipsoidal Coordinates with X axis as Major axis

The previous section assumed that the major axis of the ellipse was oriented along the  $z$  direction. For completeness, we show here how to adjust the formulation in the case when the major axis is along the  $x$  direction. In this case the ellipsoidal coordinate system is defined as

$$\begin{aligned}
 x &= r \cos(\phi) \\
 y &= \sqrt{r^2 - f^2} \sin(\phi) \sin(\theta) \\
 z &= \sqrt{r^2 - f^2} \sin(\phi) \cos(\theta)
 \end{aligned} \quad (5.32)$$

Note that in the case of a sphere,  $a = b$  which implies that  $f = 0$ , and these coordinates reduce to the spherical case. The stretched coordinates in the ellipsoidal case are given by

$$\begin{aligned}
 \tilde{x} &= \tilde{r} \cos(\phi) \\
 \tilde{y} &= \sqrt{\tilde{r}^2 - f^2} \sin(\phi) \sin(\theta) \\
 \tilde{z} &= \sqrt{\tilde{r}^2 - f^2} \sin(\phi) \cos(\theta)
 \end{aligned} \quad (5.33)$$

This implies that the Jacobian matrix is given as

$$\begin{aligned} \mathbf{J} &= \frac{\partial(\tilde{x}, \tilde{y}, \tilde{z})}{\partial(x, y, z)} = \frac{\partial(\tilde{x}, \tilde{y}, \tilde{z})}{\partial(r, \phi, \theta)} \frac{\partial(x, y, z)}{\partial(r, \phi, \theta)}^{-1} \\ &= \begin{bmatrix} \tilde{r}' \cos(\phi) & -\tilde{r} \sin(\phi) & 0 \\ \frac{\tilde{r}\tilde{r}'}{\sqrt{\tilde{r}^2 - f^2}} \sin(\phi) \sin(\theta) & \sqrt{\tilde{r}^2 - f^2} \cos(\phi) \sin(\theta) & \sqrt{\tilde{r}^2 - f^2} \sin(\phi) \cos(\theta) \\ \frac{\tilde{r}\tilde{r}'}{\sqrt{\tilde{r}^2 - f^2}} \sin(\phi) \cos(\theta) & \sqrt{\tilde{r}^2 - f^2} \cos(\phi) \cos(\theta) & -\sqrt{\tilde{r}^2 - f^2} \sin(\phi) \sin(\theta) \end{bmatrix} \\ &\quad \begin{bmatrix} \cos(\phi) & -r \sin(\phi) & 0 \\ \frac{r}{\sqrt{r^2 - f^2}} \sin(\phi) \sin(\theta) & \sqrt{r^2 - f^2} \cos(\phi) \sin(\theta) & \sqrt{r^2 - f^2} \sin(\phi) \cos(\theta) \\ \frac{r}{\sqrt{r^2 - f^2}} \sin(\phi) \cos(\theta) & \sqrt{r^2 - f^2} \cos(\phi) \cos(\theta) & -\sqrt{r^2 - f^2} \sin(\phi) \sin(\theta) \end{bmatrix}^{-1} \end{aligned} \quad (5.34)$$

### 5.2.1 Relations Between the PML Formulations

It is clear that as the minor and major axis become equal,  $a = b$ , and hence  $f = 0$ . This implies that the Jacobian for ellipsoidal coordinates in equation 5.31 reduces to the spherical Jacobian given in equation 5.25.

As an additional step, we consider that the spherical Jacobian reduces to that of the Cartesian in the limiting case of a large radius of the inner sphere defining the PML boundary. This can be seen by considering equations 5.26 and 5.27, which we repeat here for convenience

$$\tilde{r} = r + i \int_R^r \sigma(\epsilon) d\epsilon \quad (5.35)$$

which implies that

$$\tilde{r}' = \frac{\partial \tilde{r}}{\partial r} = \gamma(r) = 1 + i\sigma(r) \quad (5.36)$$

As  $r \rightarrow \infty$ , we see from equation 5.26 that if  $r - R$  is constant, then  $\tilde{r} \rightarrow r$ , since the imaginary term will become vanishingly small compared to  $r$ . However, from equation

5.27 we see no limiting change in  $\tilde{r}'$  as  $r$  becomes large. Thus, going back to equation 5.25, we have

$$\begin{aligned}
\mathbf{J}_{spherical} &= \frac{\partial(\tilde{x}, \tilde{y}, \tilde{z})}{\partial(x, y, z)} = \frac{\partial(\tilde{x}, \tilde{y}, \tilde{z})}{\partial(r, \phi, \theta)} \frac{\partial(x, y, z)}{\partial(r, \phi, \theta)}^{-1} \\
&= \begin{bmatrix} \tilde{r}' \sin(\phi) \cos(\theta) & \tilde{r} \cos(\phi) \cos(\theta) & -\tilde{r} \sin(\phi) \sin(\theta) \\ \tilde{r}' \sin(\phi) \sin(\theta) & \tilde{r} \cos(\phi) \sin(\theta) & \tilde{r} \sin(\phi) \cos(\theta) \\ \tilde{r}' \cos(\phi) & -\tilde{r} \sin(\phi) & 0 \end{bmatrix} \\
&\quad \begin{bmatrix} \sin(\phi) \cos(\theta) & r \cos(\phi) \cos(\theta) & -r \sin(\phi) \sin(\theta) \\ \sin(\phi) \sin(\theta) & r \cos(\phi) \sin(\theta) & r \sin(\phi) \cos(\theta) \\ \cos(\phi) & -r \sin(\phi) & 0 \end{bmatrix}^{-1} \\
&\rightarrow \begin{bmatrix} \tilde{r}' \sin(\phi) \cos(\theta) & r \cos(\phi) \cos(\theta) & -r \sin(\phi) \sin(\theta) \\ \tilde{r}' \sin(\phi) \sin(\theta) & r \cos(\phi) \sin(\theta) & r \sin(\phi) \cos(\theta) \\ \tilde{r}' \cos(\phi) & -r \sin(\phi) & 0 \end{bmatrix} \quad (5.37) \\
&\quad \begin{bmatrix} \sin(\phi) \cos(\theta) & r \cos(\phi) \cos(\theta) & -r \sin(\phi) \sin(\theta) \\ \sin(\phi) \sin(\theta) & r \cos(\phi) \sin(\theta) & r \sin(\phi) \cos(\theta) \\ \cos(\phi) & -r \sin(\phi) & 0 \end{bmatrix}^{-1} \\
&= \begin{bmatrix} \gamma_r & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sin(\phi) \cos(\theta) & r \cos(\phi) \cos(\theta) & -r \sin(\phi) \sin(\theta) \\ \sin(\phi) \sin(\theta) & r \cos(\phi) \sin(\theta) & r \sin(\phi) \cos(\theta) \\ \cos(\phi) & -r \sin(\phi) & 0 \end{bmatrix} \\
&\quad \begin{bmatrix} \sin(\phi) \cos(\theta) & r \cos(\phi) \cos(\theta) & -r \sin(\phi) \sin(\theta) \\ \sin(\phi) \sin(\theta) & r \cos(\phi) \sin(\theta) & r \sin(\phi) \cos(\theta) \\ \cos(\phi) & -r \sin(\phi) & 0 \end{bmatrix}^{-1} = \begin{bmatrix} \gamma(r) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

Thus, the limiting case of a large radius for the PML surface reduces to a one-dimensional PML layer. Constructing a tensor product with PML layers in the other two directions produces a diagonal Jacobian matrix as given for the Cartesian case in equation 5.12. Thus, one could use ellipsoidal PML to construct Cartesian PML simply by artificially defining the radius of the PML layer to be very large.

We have shown that the ellipsoidal PML reduces to spherical when the major and minor axis coincide, and that the spherical reduces to Cartesian in the limiting case of a large inner radius of the PML boundary. Thus, one could simply have an ellipsoidal formulation in an existing finite element code and use it to construct spherical and/or Cartesian PML layers as desired. This generality makes the ellipsoidal formulation a convenient choice.

### 5.3 LossFunction

One parameter of the PML formulation is the choice of  $\sigma(\xi)$  from Equation 5.8. The choice of  $\sigma$  defines how rapidly the complex transformation occurs in the boundary layers. While the choice of the loss function  $\sigma(d)$  is discussed in the literature [21, 22, 126], papers in the literature use a range of formulations and implementations, and it is still unclear what the best choice is for any given problem. Bermudez [21] recommends singular loss functions (Equation 5.38), which is unbounded at the outer boundary.

$$\sigma(\xi) = \frac{c_1}{t - \xi} \quad (5.38)$$

Michler et al. [22] uses fifth and sixth order polynomials. The consensus in the literature is that the loss function should start at a low value (often zero) on the inner boundary to minimize numerical reflections, but should increase at an increasing rate to maximize the loss terms near the outer boundary.

Rather than present another analysis of possible loss functions for this formulation, a simple loss function is selected, that includes only the constant, linear, quadratic, and cubic terms.

For the polynomial loss function, four parameters define the loss function

$$\sigma(\xi) = c_1 + c_2 \frac{\xi}{t} + c_3 \frac{\xi^2}{t^2} + c_4 \frac{\xi^3}{t^3} \quad (5.39)$$

Where  $c_1$ ,  $c_2$ ,  $c_3$ , and  $c_4$  are specific to the problem,  $\xi$  is the distance along the normal from the Gauss point to the inner ellipsoid boundary, and  $t$  is the total thickness of the PML layer. The loss function is normalized such that changing the thickness of the PML layer does not change the maximum value of  $\sigma$ . The calculation for  $\bar{\sigma}$  is performed analytically, and is shown to be

$$\bar{\sigma}(\xi) = c_1\xi + c_2\frac{\xi^2}{2t} + c_3\frac{\xi^3}{3t^2} + c_4\frac{\xi^4}{4t^3} \quad (5.40)$$

To further simplify the search for the best loss parameter,  $c_1$  was set equal to 0, and the other constants were set equal to each other ( $c_2 = c_3 = c_4$ ). This simplifies the search for an optimal loss function, as there is only one parameter that needs to be “tuned”. It is hypothesized that widening this search space may further improve the results presented in Section 5.7.

In most structural-acoustic applications the frequencies of interest, acoustic material properties, and acoustic element size are often the same between runs, with the structural model primarily changing.

The loss functions can be selected for a purely acoustic mesh with a known solution, and then those same parameters can be used for a coupled structural acoustics problem.

## 5.4 Implementation

The PML boundary condition was set up such that six noded wedge (WEDGE6) elements are extruded from the exterior of the acoustic domain of interest, where the exterior acoustic surface is discretized with linear triangles. For the Cartesian formulation, the WEDGE6 elements are extruded normal to the surface. For the spherical formulation, the WEDGE6 elements are extruded on radial lines originating from the center of the sphere. For the ellipsoidal formulation, the WEDGE6 elements to increasing larger confocal ellipsoids, along radii that are not necessarily straight lines. Figure 5.2 shows the ellipsoidal confocal meshing. The elements have a relatively thin thickness near the ends of the ellipse, and a relatively thick thickness along the

sides. Note that these elements are of equal thickness in ellipsoidal coordinates, due to the ellipsoidal transformation, the PML loss function reaches the same maximum across the surface of the exterior boundary.

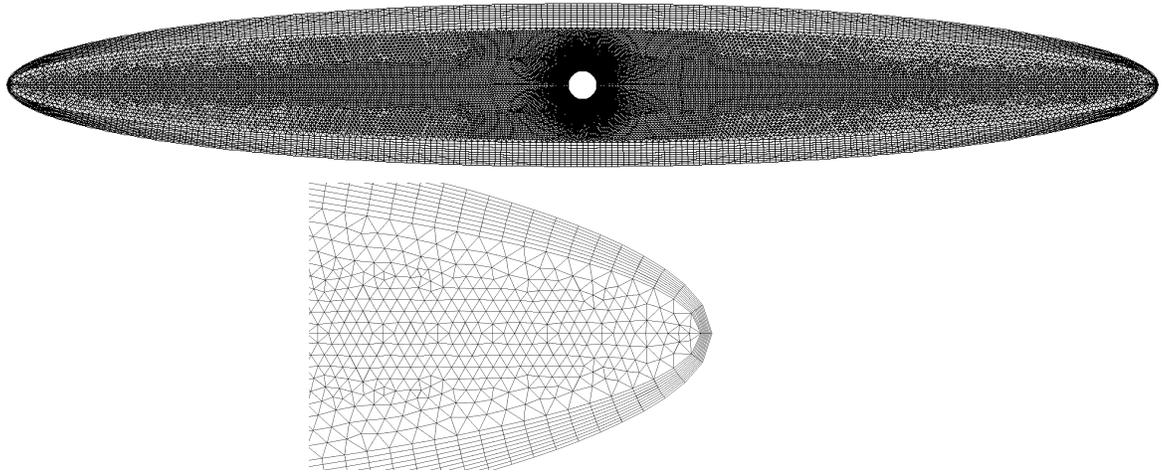


Figure 5.2.: 2D Slice of a 3D Acoustic Mesh with PML Ellipsoidal Boundary

## 5.5 Verification

In the majority of existing structural acoustic meshes, the ellipsoidal acoustic domain is meshed with four noded tetrahedral (TET4) elements. Each mesh that is currently solved with Infinite Elements needs to be solvable with PML without needing to re-mesh or decompose. Thus, TET4 acoustic elements were selected for this analysis.

The PML formulation was verified using three problems. The first problem is a 3D representation of a 1D Waveguide, which was not oriented along any of the coordinate axes. The second problem is an “Offset Sphere”, where an acoustic source is placed asymmetrically in a spherical domain. The third problem is a prolate ellipsoid with a major axis of 10 meters in the x direction, and minor axis of 1 meter the y and z directions. Each model was run in parallel on 16 processors.

### 5.5.1 Oriented Waveguide

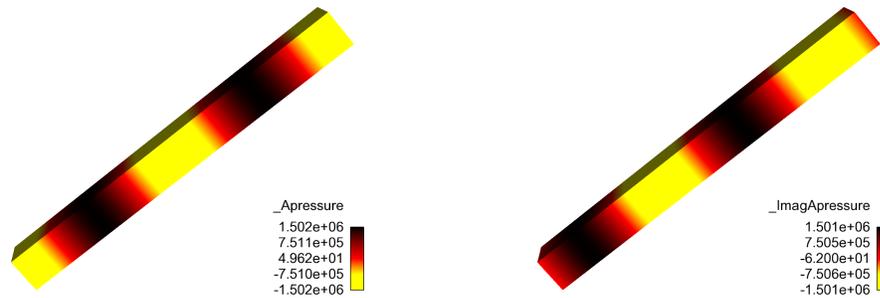


Figure 5.3.: Numerical Solution for Oriented Waveguide (150 Hz)

Figure 5.3 shows the numerical solution to the oriented Waveguide Problem. The mesh has dimensions  $20 \times 1 \times 1$  meters. It is composed of 271,000 TET4 elements, and 51,000 nodes. The boundary where PML is applied has 1548 nodes, and 516 element faces. The material modeled was water, where  $\rho = 1000 \frac{\text{kg}}{\text{m}^3}$  is the density of the fluid, and  $c_0 = 1500 \frac{\text{m}}{\text{s}}$  is the speed of sound.

The exact solution is given as

$$P(D) = \rho c_0 V_0 e^{ikD} \quad (5.41)$$

where the applied velocity  $V_0 = 1$ , and  $D(x)$  is the distance from the acoustic source to the a point in the acoustic mesh. The wavelength  $k = \Omega/c_0$ .

### 5.5.2 Offset Sphere

Figure 5.4 shows the numerical solution for the offset sphere problem. The radius of the outer sphere is 5 meters, and the radius of the inner sphere is 1 meter. There sphere is composed of 850,000 TET4 elements, and 145,000 nodes. The material modeled is air, where  $\rho = 1.293 \frac{\text{kg}}{\text{m}^3}$ ,  $c_0 = 332.0 \frac{\text{m}}{\text{s}}$

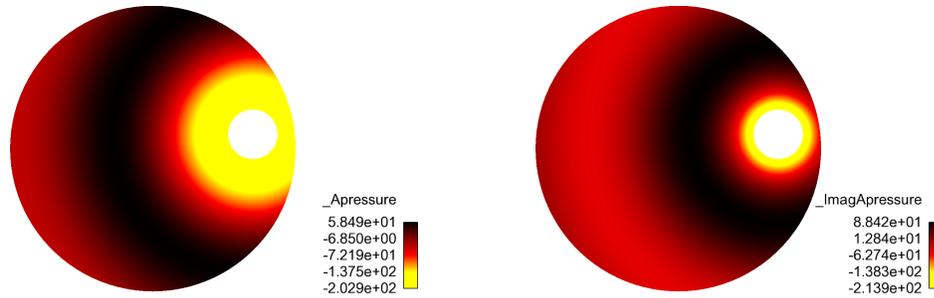


Figure 5.4.: Numerical Solution for Offset Sphere (50 Hz)

A 2D representation of the spherical result cut along the plane  $y=0$  is shown. Note that the solution is spherically symmetric about the acoustic source. The exact solution is given as

$$P(r) = \frac{iV_0\Omega\rho a^2}{r(1+ika)} e^{ik(r-a)} \quad (5.42)$$

where  $r$  is the distance from the center of the inner sphere to a point in the mesh, and  $a$  is the radius of the inner sphere.

### 5.5.3 Ellipse

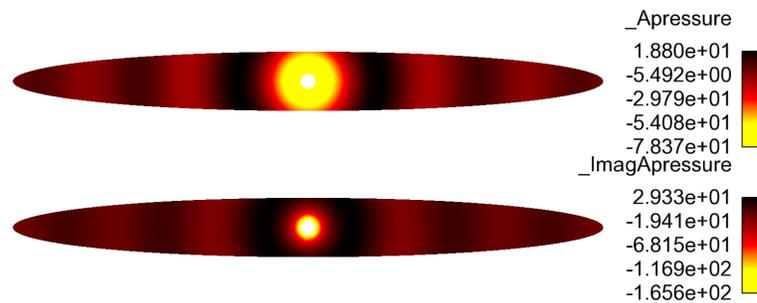


Figure 5.5.: Numerical Solution for Ellipsoid (100 Hz)

Figure 5.5 shows the numerical solution for a ten to one aspect ratio ellipsoid. The major axis has a radius of 10 meters, and each minor axis has a radius of 1 meter. The inner sphere has a radius of 0.25 meters. The mesh is composed of 2,800,000

TET4 elements, and 500,000 nodes. The face of the inner sphere, where the acoustic velocity is applied, contains 125,000 elements faces and 373,000 nodes. The outer face of the ellipsoid, contains 18,500 elements and 56,000 nodes. The material is air, with the same material properties as the offset sphere. A 2D representation of the ellipsoidal result along the plane  $y=0$  is shown. Note that the solution is once again spherically symmetric about the acoustic source. The exact solution is the same as that of the sphere, given in Equation 5.42. The only difference is the shape of the acoustic domain, and the size of the inner sphere.

## 5.6 Solver

The linear equations in this study were solved to a relative residual tolerance of  $10^{-11}$  using a right-preconditioned GMRES algorithm [127]. Specifically, for each linear system of the form  $Ax = b$ , the calculated approximate solution  $x_a$  satisfies  $\|b - Ax_a\|/\|b\| \leq 10^{-11}$ . Although the presence of energy dissipation (damping) in the form of absorbing boundaries can be helpful, the solution of acoustic Helmholtz problems on exterior domains can still be a challenge. As is often the case, the success of iterative Krylov solvers such as GMRES depends strongly on the availability of a good preconditioner.

### 5.6.1 Preconditioner

The preconditioner used in this study is based on a classic two-level overlapping Schwarz approach [128, 129]. The coarse space for the preconditioner is constructed from solutions of problems local to each subdomain [130]. Further, the coarse part of the preconditioner is applied in a multiplicative form [128].

For subdomains consisting entirely of acoustic elements without any damping, the restriction of the coefficient matrix  $A$  to an overlapping subdomain may be singular depending on the excitation frequency  $\omega$ . This is true because these overlap matrices are indefinite for large enough values of  $\omega$ . In order to deal with this potential

complication, a preconditioner is constructed for a perturbed matrix  $\tilde{A} := A + A_d$ , where the matrix  $A_d$  introduces additional energy dissipation into the problem. We note that although  $\tilde{A}$  is used in the construction of the preconditioner, the original linear system  $Ax = b$  remains the one that is solved.

A fairly recent approach for selecting  $A_d$  is given by

$$A_d = i\gamma\omega^2 M$$

and goes by the name shifted Laplacian [131, 132]. Here,  $i$  is the imaginary unit,  $\gamma$  is a dimensionless parameter, and  $M$  is the mass matrix for the acoustic domain (not including the effects of infinite elements or PML).

Another approach investigated here is to choose

$$A_d = i\gamma K,$$

where  $K$  is the stiffness matrix for the acoustic domain that does not include any infinite element or PML contributions. This choice of  $A_d$  is motivated by the idea of using structural damping [133] to model energy dissipation in structures. Indeed, both forms of  $A_d$  shown above provide additional damping for the preconditioner. We found in our numerical studies that either of the two choices for  $A_d$  resulted in similar numbers of iterations and solution times for the same value of  $\gamma$ ; in this study we used  $\gamma = 0.12$ .

## 5.7 Results

In this section, the relationship between the thickness of the PML boundary, the discretization of the elements within the PML boundary, and the selection of loss parameters is investigated. For each of the problems, the numerical problem is solved with varying thickness, number of elements, and loss parameters. The discrete L2

error norm of the solution is compared between the PML formulations, the absorbing boundary conditions, and infinite elements of various orders.

### 5.7.1 Oriented Waveguide

Figure 5.6 shows the results for the Oriented Waveguide at a frequency of 100Hz. Both the absorbing boundary conditions and any infinite element solution perfectly absorb outgoing waves, and give theoretically exact results. The difference between the numerical solutions and the exact solution is due to discretization error of the interior acoustic mesh. Figure 5.6 shows that with the correct PML parameters, the PML solution also converges to the discretization error of the interior mesh. For the waveguide problem, where outgoing waves are always orthogonal to the boundary condition, it is clear that the computationally inexpensive absorbing boundary condition is likely the best choice. The optimal parameters for all three PML formulation are shown to be loss parameters of 2500, with a thickness of 2 meters and 20 elements in the boundary.

### 5.7.2 Offset Sphere

Figure 5.7 shows the results for the Offset Sphere at a frequency of 50Hz. For this case, the outgoing waves are not perpendicular to the boundary surface, and the absorbing boundary condition fails to absorb the waves. Increasing the infinite element order beyond 4 does not improve the solution, and the error norm of the fourth order infinite elements represents the discretization error for the mesh. While the ellipsoidal and spherical PML formulations converge to the discretization error of the mesh, the Cartesian PML fails to absorb the outgoing waves. The discretization error is achieved with 12 elements, a loss parameter of 600, and a thickness of 2 meters. Fourth order infinite elements are the most computationally efficient method to solve this problem.

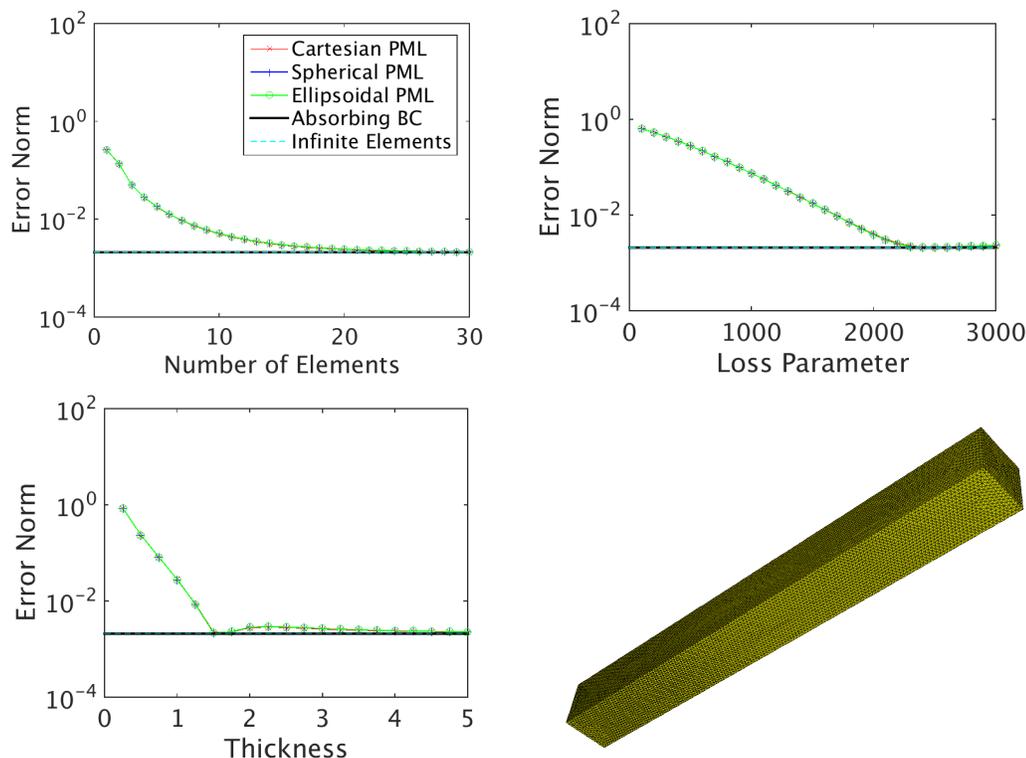


Figure 5.6.: Parameter Studies for Oriented Waveguide (100 Hz)

### 5.7.3 Ellipsoid

Figure 5.8 shows the results for the 10:1 aspect ratio ellipse, at a frequency of 100 Hz. The absorbing boundary condition, Cartesian PML, and spherical PML all fail to absorb the outgoing waves. 14th Order infinite elements are needed to reach the discretization error of the mesh. PML can reach the same discretization error with only 10 elements, a loss parameter of 2500, and a thickness of 0.15 meters. Here the ellipsoidal PML is the most computationally efficient solution for this problem.

Each comparison was done at a single frequency primarily because the discretization error of a mesh is frequency dependent. A comparison of error norms across frequencies would place higher weight on higher frequencies, where mesh discretization can drive up error. Thus, the error norm at the highest frequency would govern the total error norm, and the norms at lower frequencies would have little impact. Note that

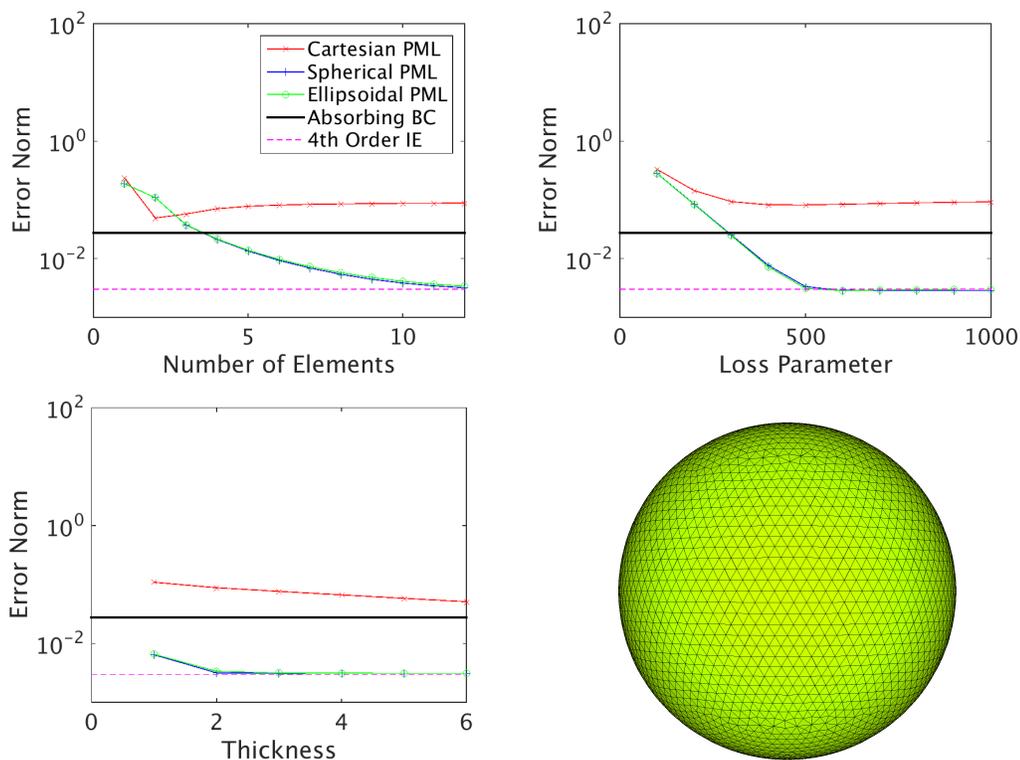


Figure 5.7.: Parameter Studies for OffsetSphere (50 Hz)

as discretization error of the mesh increases, the benefit of PML decreases. When discretization error of the mesh decreases, the benefit of PML increases.

#### 5.7.4 Comparison of Infinite Elements to PML

Figure 5.9 shows directly how infinite element order and the PML number of elements affects the error norm, solve time, and number of iterations for the preconditioned GMRES solver. Each order of infinite elements or PML elements adds one additional node for each node on the acoustic domain boundary. Thus, 6th Order Infinite Elements represents the same number of degrees of freedom and linear system size as a PML formulation with 6 elements. Figure 5.9(a) shows that for the same problem size, PML results in a lower error norm than Infinite Elements. Figure 5.9(b) shows the number of iterations required for convergence of the linear solver. It is

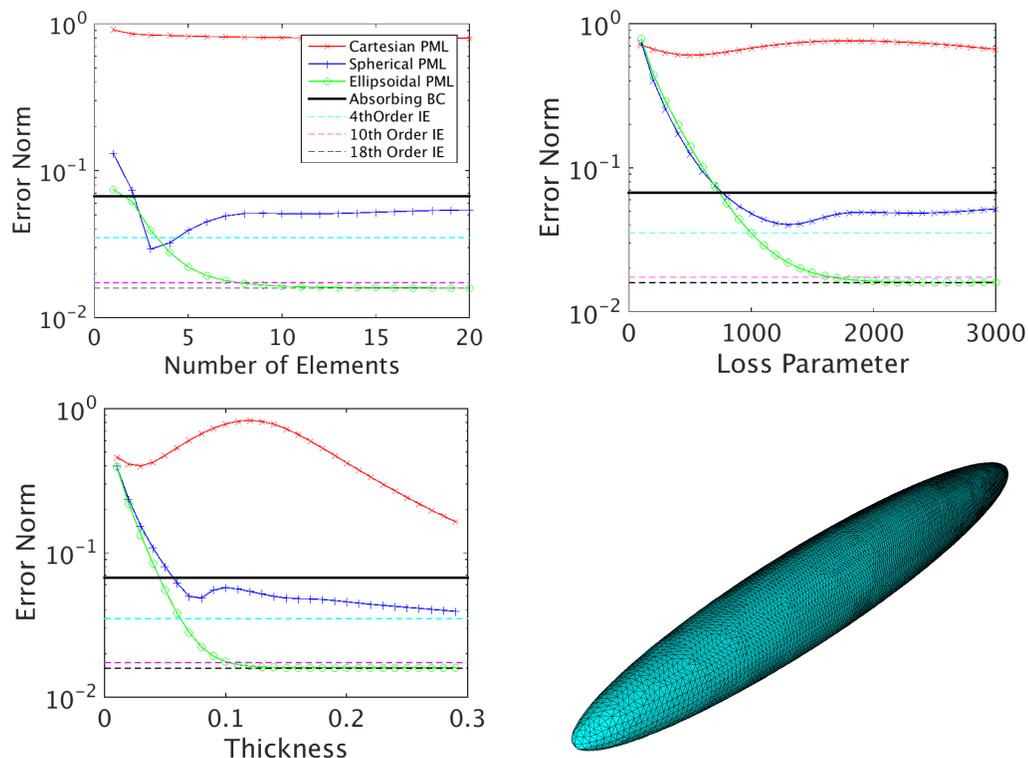


Figure 5.8.: Parameter Studies for Ellipsoid (100 Hz)

shown that Infinite elements at higher orders require significantly more iterations than PML to solve the same size problem. Figure 5.10(a) shows that Infinite elements require longer solve times than PML. The longer solve times are primarily due to the increase in iteration count, but each iteration also takes longer.

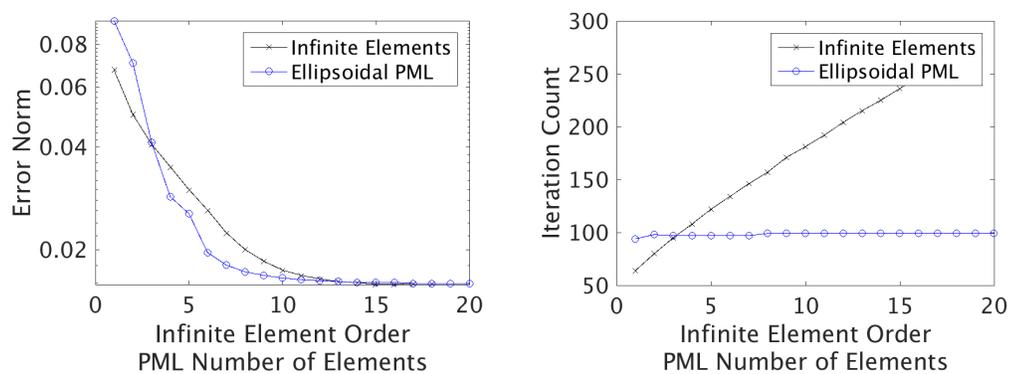


Figure 5.9.: Comparison Between IE and PML (100 Hz)

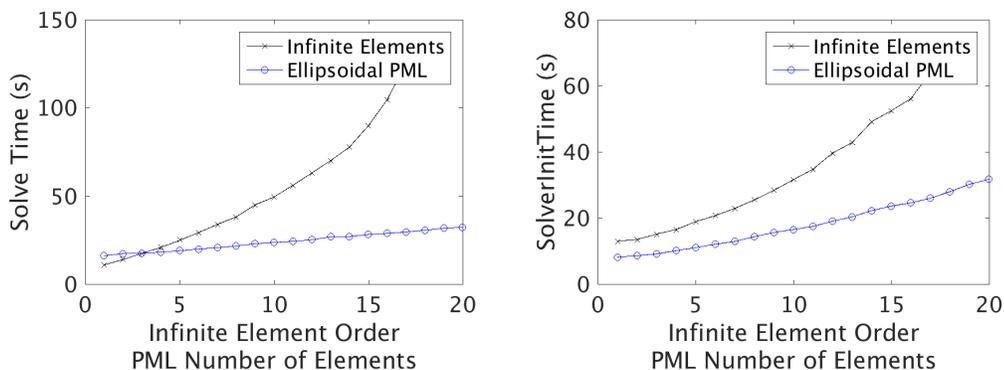


Figure 5.10.: Comparison Between IE and PML (100 Hz)

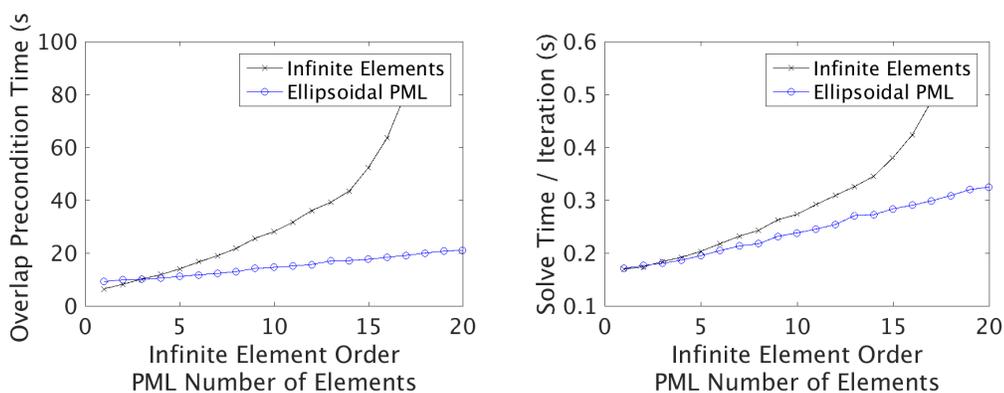


Figure 5.11.: Comparison Between IE and PML (100 Hz)

The results for a 10:1 aspect ratio ellipse consisting of only acoustic elements is presented here, the broader impact of these methods can be seen when even larger problems are run. Meshes with large interior structural components will suffer more from poor matrix conditioning caused by infinite elements, as the increased solver time due to conditioning is compounded by higher numbers of processors and degrees of freedom in the mesh.

## 5.8 Conclusions

The primary motivation for this work was to explore an alternative to higher order infinite elements on ellipsoidal meshes. The high iteration count caused by these

infinite elements has a large impact on solve time and scalability of structural-acoustic meshes. In this paper, we have presented the existing derivations for Cartesian and spherical PML, and presented two novel derivations for rotated Cartesian and ellipsoidal PML. For each method, we have presented a way to map the formulation back to Cartesian space, allowing for novel implementations of spherical and ellipsoidal PML formulations in meshes defined in Cartesian coordinates. For the ellipsoidal method, we have presented a method to extrude the boundary to confocal ellipsoids of increasing radii, ensuring the same maximum loss function across the surface of the outermost ellipsoid.

The formulation has been verified for three specific cases. A 10-1 aspect ratio ellipse was used to verify the ellipsoidal formulation. An asymmetric spherical mesh with non-orthogonal outgoing waves was used to verify the spherical and ellipsoidal formulations. A 1D Waveguide in 3D space was used to verify the rotated Cartesian, spherical, and ellipsoidal formulations.

The first parallel implementation of PML available in the literature has been presented, and the effect on the parallel linear solver has been studied. The linear solver iteration counts and solve times of the PML formulation have been compared with equivalent infinite elements, and it has been demonstrated that PML outperforms higher order infinite elements for large meshes. Further, it has been demonstrated that this benefit scales with mesh size. The greater impact of this work is the reduction in solution times for existing structural-acoustic problems on ellipsoidal meshes.

## 6. SUMMARY AND CONCLUSIONS

In this work, different numerical techniques are used to advance state of the art RTHS, allowing for higher fidelity models in real-time. Aspects of the multi-time-step integration method are investigated, and a thorough cost vs. error analysis for the MTS method with two subdomains is conducted. Next, the MTS method is used in a virtual Real-Time Hybrid Simulation (RTHS). The algorithm is adapted to account for the limitations in RTHS, specifically the inability to apply updates back in time when a physical substructure is involved. Using the computational time made available by the MTS method, higher fidelity numerical models are used in a MTS+RTHS method. Nonlinear models that utilize beam-continuum coupling are used to obtain better results than the purely linear beam models that are typically used in RTHS. Additional work developing an ellipsoidal PML formulation lays the groundwork for soil structure interaction in future RTHS.

In Chapter 2, the details of the MTS method are studied, and careful breakdown of computational cost vs. computed error is conducted for a series of decompositions. A system is developed to find an optimal decomposition. While the system proposed in Chapter 2 is prohibitively expensive for large problems, it is possible to use this system to pick a decomposition for RTHS problems. Additionally, further research can be conducted to use error estimators to circumvent the need for an exact solution, and to pick a decomposition on the fly under changing loading conditions.

In Chapter 3, an implementation of the MTS method for RTHS is presented. The benefits, particularly in regards to computational cost, are outlined, and the algorithmic changes necessary to implement MTS+RTHS are presented. A predictor is used on the linear subdomain coupling updates, allowing for a consistent MTS integration scheme coupled with the physical experiment. A vRTHS on a nine-story

benchmark structure is used to demonstrate that MTS+RTHS is a much better option than using simple shear models.

In Chapter 4, the additional time gained by MTS+RTHS is used to allow bigger models in RTHS, with more degrees of freedom, more complicated geometry, and nonlinear material models. Beam-Continuum coupling is used to transition regions of the model between coarsely meshed beam elements and finely meshed linear and nonlinear continuum elements. The results from the simple linear model, beam-continuum linear model, and beam-continuum non-linear model are compared to experimental results obtained from a full physical test. Once again, it is demonstrated that these numerical techniques allow for high fidelity RTHS.

In Chapter 5, an ellipsoidal PML formulation is presented, as well as a detailed comparison to infinite elements when used in a massively parallel environment for the Helmholtz problem. It is noted that the benefits of PML scale with problem size, particularly due to the limitations of solving the poorly conditioned infinite element system matrices in a massively parallel environment. Future work may extend this formulation to the time domain, to be yet another tool for large scale high fidelity RTHS.

Each of these chapters works towards a common goal of better, faster, numerical and RTHS modeling. This research is a step in the direction of developing high-fidelity multi-scale models for full scale RTHS when existing modeling techniques are not sufficient to capture the behavior of a structure.

## LIST OF REFERENCES

## LIST OF REFERENCES

- [1] Japan National Committee on Earthquake Engineering. Niigata earthquake, 1964. In *Proceedings of the 3rd World Conference in Earthquake Engineering*, volume III, pages 78–105, 1965.
- [2] B. Phillips, Z. Jiang, J.M. Ricles, S.J. Dyke, Y. Chae, B.F. Spencer, R.E. Christenson, and A. Agrawal. Real-time hybrid simulation benchmark study with a large-scale MR damper. *Proc. of the 5th World Conference on Structural Control and Monitoring*, pages 12–14, 2010.
- [3] Victor Saouma, Dae-Hung Kang, and Gary Haussmann. A computational finite-element program for hybrid simulation. *Earthquake Engineering and Structural Dynamics*, 41:375–289, May 2011.
- [4] A. Maghareh, S.J. Dyke, A. Prakash, G. Bunting, and P. Lindsay. Evaluating modeling choices in the implementation of real-time hybrid simulation. In *2012 Joint Conference of the Engineering Mechanics Institute and the 11th ASCE Joint Specialty Conference on Probabilistic Mechanics and Structural Reliability*, 2012.
- [5] Y. Chae, S. Tong, T. Marullo, and J.M. Ricles. Real-time hybrid simulation studies of complex large-scale systems using multi-grid processing. In *20th Analysis and Computation Specialty Conference*, pages 359–370. ASCE, 2012.
- [6] A.M. Reinhorn, M.V. Sivaselvan, Z. Liang, X. Shao, M. Pitman, and S. Weinreber. Large scale real time dynamic hybrid testing technique—shake tables substructure testing. In *The First International Conference on Advances in Experimental Structural Engineering*, volume 1, pages 457–464, 2005.
- [7] Victor Saouma, Gary Haussmann, Dae-Hung Kang, and Wassim Ghannoum. Real time hybrid simulation of a non ductile reinforced concrete frame. *Journal of Structural Engineering*, 140(2):04013059, 2013.
- [8] Nestor Castaneda, Xiuyu Gao, and Shirley J Dyke. Computational tool for real-time hybrid simulation of seismically excited steel frame structures. *Journal of Computing in Civil Engineering*, 29(3):04014049, 2013.
- [9] David Ferry, Gregory Bunting, Amin Maghareh, Arun Prakash, Shirley Dyke, Kunal Agrawal, Chris Gill, and Chenyang Lu. Real-time system support for hybrid structural simulation. In *Proceedings of the 14th International Conference on Embedded Software*. ACM, 2014.
- [10] David Ferry, Amin Maghareh, Gregory Bunting, Arun Prakash, Kunal Agrawal, Chris Gill, Chenyang Lu, and Shirley Dyke. On the performance of a highly parallelizable concurrency platform for real-time hybrid simulation. In *The Sixth World Conference on Structural Control and Monitoring*, 2014.

- [11] T. Belytschko, H.-J. Yen, and R. Mullen. Mixed methods for time integration. *Computer Methods in Applied Mechanics and Engineering*, 17/18:259–275, 1979.
- [12] P. Smolinski, S. Sleith, and T. Belytschko. Stability of an explicit multi-time step integration algorithm for linear structural dynamics equations. *Computational Mechanics*, 18:236–244, 1996.
- [13] W. J. T. Daniel. The subcycled Newmark algorithm. *Computational Mechanics*, 20:272–281, 1997.
- [14] A. Gravouil and A. Combescure. Multi-time-step explicit-implicit method for non-linear structural dynamics. *International Journal for Numerical Methods in Engineering*, 50:199–225, 2001.
- [15] A. Combescure and A. Gravouil. A numerical scheme to couple subdomains with different time-steps for predominantly linear transient analysis. *Computer Methods in Applied Mechanics and Engineering*, 191:1129–1157, 2002.
- [16] A. Prakash and K. D. Hjelmstad. A FETI based multi-time-step coupling method for Newmark schemes in structural dynamics. *International Journal for Numerical Methods in Engineering*, 61:2183–2204, 2004.
- [17] Arun Prakash, Ertugrul Taciroglu, and Keith D. Hjelmstad. Computationally efficient multi-time-step method for partitioned time integration of highly nonlinear structural dynamics. *Computers & Structures*, 133:51–63, 2014.
- [18] Hansen Pitandi. Geometrically consistent coupling of beam and continuum models for studying collapse of structures, 2011.
- [19] George Mylonakis and George Gazetas. Seismic soil-structure interaction: beneficial or detrimental? *Journal of Earthquake Engineering*, 4(3):277–301, 2000.
- [20] Jean-Pierre Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics*, 114(2):185–200, 1994.
- [21] A Bermúdez, L. Hervella-Nieto, A. Prieto, et al. An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems. *Journal of Computational Physics*, 223(2):469–488, 2007.
- [22] Ch Michler, Leszek Demkowicz, J. Kurtz, and D. Pardo. Improving the performance of perfectly matched layers by means of hp-adaptivity. *Numerical Methods for Partial Differential Equations*, 23(4):832–858, 2007.
- [23] Joseph J. Shirron and Thomas E. Giddings. A finite element model for acoustic scattering from objects near a fluid–fluid interface. *Computer Methods in Applied Mechanics and Engineering*, 196(1):279–288, 2006.
- [24] David S. Burnett and Richard L. Holford. An ellipsoidal acoustic infinite element. *Computer Methods in Applied Mechanics and Engineering*, 164(1):49–76, 1998.
- [25] Fang Q Hu. Absorbing boundary conditions. *International Journal of Computational Fluid Dynamics*, 18(6):513–522, 2004.
- [26] Garth M Reese, Manoj Kumar Bhardwaj, Daniel Joseph Segalman, Kenneth Fredrick Alvin, Brian James Driessen, Kendall Pierson, and Timothy Walsh. Salinas-user’s notes. *SAND report*, 2801, 1999.

- [27] Garth Reese, Manoj K Bhardwaj, and Timothy Walsh. Salinas–theory manual. 2004.
- [28] Leszek Demkowicz. Computing with hp-adaptive finite elements, vol. 1: One and two dimensional elliptic and Maxwell problems. Applied Mathematics and Nonlinear Science Series. *Book*, 2006.
- [29] Anil K. Chopra. *Dynamics of Structures, Theory and Applications to Earthquake Engineering, 4th ed.* Prentice-Hall, 2012.
- [30] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory.* Springer, 2005.
- [31] Y. Fragakis and M. Papadrakakis. The mosaic of high performance domain decomposition methods for structural mechanics: Formulation, interrelation and numerical efficiency of primal and dual methods. *Computer Methods in Applied Mechanics and Engineering*, 192:3799–3830, 2003.
- [32] Y. Fragakis and M. Papadrakakis. The mosaic of high-performance domain decomposition methods for structural mechanics - Part II: Formulation enhancements, multiple right-hand sides and implicit dynamics. *Computer Methods in Applied Mechanics and Engineering*, 193:4611–4662, 2004.
- [33] N. M. Newmark. A method of computation for structural dynamics. *Journal of Engineering Mechanics, ASCE*, 85:67–94, 1959.
- [34] E.L. Wilson, I. Farhoomand, and K.J. Bathe. Nonlinear dynamic analysis of complex structures. *Earthquake Engineering & Structural Dynamics*, 1(3):241–252, 1972.
- [35] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, 5:283–292, 1977.
- [36] W.L. Wood, M. Bossak, and O.C. Zienkiewicz. An alpha modification of Newmark’s method. *International Journal for Numerical Methods in Engineering*, 15:1562–1566, 1981.
- [37] D. Levy and E. Tadmor. From semidiscrete to fully discrete: Stability of Runge-Kutta schemes by the energy method. *SIAM Review*, 40:40–73, 1998.
- [38] Pål G Bergan and Egil Mollestad. An automatic time-stepping algorithm for dynamic problems. *Computer Methods in Applied Mechanics and Engineering*, 49(3):299–318, 1985.
- [39] K.C. Park and P.G. Underwood. A variable-step central difference method for structural dynamics analysis—Part 1. theoretical aspects. *Computer Methods in Applied Mechanics and Engineering*, 22(2):241–258, 1980.
- [40] J.J. Sanchez-Gasca, R. D’Aquila, W.W. Price, and J.J. Paserba. Variable time step, implicit integration for extended-term power system dynamic simulation. In *Power Industry Computer Application Conference, 1995. Conference Proceedings., 1995 IEEE*, pages 183–189. IEEE, 1995.

- [41] Fernando J Barros. Comparing synchronous and asynchronous variable step size explicit ODE solvers: A simulation study. In *Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation*, pages 32–37. IEEE Computer Society, 2007.
- [42] William Fong, Eric Darve, and Adrian Lew. Stability of asynchronous variational integrators. *Journal of Computational Physics*, 227(18):8367–8394, 2008.
- [43] K. G. Kale and A. J. Lew. Parallel asynchronous variational integrators. *International Journal for Numerical Methods in Engineering*, 70:291–321, 2007.
- [44] T. Belytschko and R. Mullen. Stability of explicit-implicit mesh partitions in time integration. *International Journal for Numerical Methods in Engineering*, 12:1575–1586, 1978.
- [45] P. Smolinski, T. Belytschko, and W. K. Liu. Stability of multi-time-step partitioned integrators for first-order systems of equations. *Computer Methods in Applied Mechanics and Engineering*, 65:115–125, 1987.
- [46] C. Farhat and F. X. Roux. A method for finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32:1205–1227, 1991.
- [47] M. O. Neal and T. Belytschko. Explicit-explicit subcycling with non-integer time step ratios for structural dynamic systems. *Computers and Structures*, 31:871–880, 1989.
- [48] P. Smolinski. Subcycling integration with non-integer time steps for structural dynamics problems. *Computers and Structures*, 59:273–281, 1996.
- [49] C.A. Felippa, K.C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190(24):3247–3270, 2001.
- [50] Alain Combescure, Najib Mahjoubi, Anthony Gravouil, and Nicolas Greffet. A time variational method to couple heterogeneous time integrators. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 19(1-3):11–24, 2010.
- [51] Najib Mahjoubi, Anthony Gravouil, Alain Combescure, and N Greffet. A monolithic energy conserving method to couple heterogeneous time integrators with incompatible time steps in structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 200(9):1069–1086, 2011.
- [52] A Gravouil, A Combescure, and M Brun. Heterogeneous asynchronous time integrators for computational structural dynamics. *International Journal for Numerical Methods in Engineering*, 2014.
- [53] Chenyang Liu, Muhammad Hasan Jamal, Milind Kulkarni, Arun Prakash, and Vijay Pai. Exploiting domain knowledge to optimize parallel computational mechanics codes. In *Proceedings of the 27th international ACM conference on International conference on supercomputing, ICS '13*, pages 25–36, New York, NY, USA, 2013. ACM.

- [54] K. C. Park and C. A. Felippa. Partitioned transient analysis procedures for coupled-field problems: Accuracy analysis. *Journal of Applied Mechanics, ASME*, 47:919–926, 1980. Transactions of the ASME.
- [55] Ignacio Romero and Luis M. Lacoma. Analysis of error estimators for the semidiscrete equations of linear solid and structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 195:2674–2696, 2006.
- [56] O.C. Zienkiewicz and Y.M. Xie. A simple error estimator and adaptive time stepping procedure for dynamic analysis. *Earthquake Engineering & Structural Dynamics*, 20(9):871–887, 1991.
- [57] N.E. Wiberg and X.D. Li. A post-processing technique and an a posteriori error estimate for the newmark method in dynamic analysis. *Earthquake Engineering & Structural Dynamics*, 22(6):465–489, 1993.
- [58] Chang-Koon Choi and Heung-Jin Chung. Error estimates and adaptive time stepping for various direct time integration methods. *Computers & structures*, 60(6):923–944, 1996.
- [59] Y. Ohtori, R.E. Christenson, BF Spencer Jr, and S.J. Dyke. Benchmark control problems for seismically excited nonlinear buildings. *Journal of Engineering Mechanics*, 130(4):366–385, 2004.
- [60] Horace Lamb. On the propagation of tremors over the surface of an elastic solid. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, pages 1–42, 1904.
- [61] M.J. Kuhn. A numerical study of lamb’s problem. *Geophysical prospecting*, 33(8):1103–1137, 1985.
- [62] Dimitri Komatitsch and Jean-Pierre Vilotte. The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures. *Bulletin of the Seismological Society of America*, 88(2):368–392, 1998.
- [63] Dimitri Komatitsch, Jean-Pierre Vilotte, Rossana Vai, Jose M Castillo-Covarrubias, and Francisco J Sanchez-Sesma. The spectral element method for elastic wave equations-application to 2-d and 3-d seismic problems. *International Journal for numerical methods in engineering*, 45(9):1139–1164, 1999.
- [64] P.S.B. Shing and S.A Mahin. Computational aspects of a seismic performance test method using on-line computer control. *Earthquake Engineering & Structural Dynamics*, 13(4):507–526, 1985.
- [65] Naru Nakata, SJ Dyke, Jian Zhang, Gilberto Mosqueda, Xiaoyun Shao, Hussam Mahmoud, Monique Hite Head, Michael Bletzinger, Gemez A Marshall, Gaby Ou, and Cheng Song. Hybrid simulation primer and dictionary. *Network for Earthquake Engineering Simulation, NEES*, 2014.
- [66] A. Gravouil and A. Combescure. Multi-time-step explicit–implicit method for non-linear structural dynamics. *International Journal for Numerical Methods in Engineering*, 50(1):199–225, 2001.

- [67] A Bonelli, OS Bursi, L He, G Magonette, and P Pegon. Convergence analysis of a parallel interfield method for heterogeneous simulations with dynamic substructuring. *International Journal for Numerical Methods in Engineering*, 75(7):800–825, 2008.
- [68] OS Bursi, A Gonzalez-Buelga, L Vulcan, SA Neild, and DJ Wagg. Novel coupling rosenbrock-based algorithms for real-time dynamic substructure testing. *Earthquake Engineering & Structural Dynamics*, 37(3):339–360, 2008.
- [69] Xiuyu Gao, Nestor Castaneda, and Shirley J. Dyke. Experimental validation of a generalized procedure for mdof real-time hybrid simulation. *Journal of Engineering Mechanics*, 140(4):04013006, 2013.
- [70] Anthony Friedman, Shirley J. Dyke, Brian Phillips, Ryan Ahn, Baiping Dong, Yunbyeong Chae, Nestor Castaneda, Zhaoshuo Jiang, Jianqiu Zhang, Youngjin Cha, Ali Ozdagli, B. F. Spencer, James Ricles, Richard Christenson, Anil Agrawal, and Richard Sause. Large-scale real-time hybrid simulation for evaluation of advanced damping system performance. *Journal of Structural Engineering*, 141(6):04014150, 2014.
- [71] Amin Maghareh, Shirley Dyke, Arun Prakash, and Gregory Bunting. Establishing a predictive performance indicator for real-time hybrid simulation. *Earthquake Engineering and Structural Dynamics*, 43(15):2299–2318, 2014.
- [72] P.B. Shing and S.A. Mahin. Experimental error propagation in pseudodynamic testing. *Rep. No. UCB/EERC-83*, 12, 1983.
- [73] Xiaoyun Shao, Andrei M. Reinhorn, and Mettupalayam V. Sivaselvan. Real-time hybrid simulation using shake tables and dynamic actuators. *Journal of Structural Engineering*, 137(748), 2011.
- [74] Cheng Chen and James M. Ricles. Improving the inverse compensation method for real-time hybrid simulation through a dual compensation scheme. *Earthquake Engineering and Structural Dynamics*, 38:1237–1255, February 2009.
- [75] Amin Maghareh, Jacob P. Waldbjørn, Shirley J. Dyke, Arun Prakash, and Ali I. Ozdagli. Adaptive multi-rate interface: development and experimental verification for real-time hybrid simulation. *Earthquake Engineering & Structural Dynamics*, 2016.
- [76] Amin Maghareh, Shirley Dyke, Siamak Rabieniaharatbar, and Arun Prakash. Predictive stability indicator: a novel approach to configuring a real-time hybrid simulation. *Earthquake Engineering & Structural Dynamics*, 2016.
- [77] A.P. Darby, M.S. Williams, and A Blakeborough. Stability and delay compensation for real-time substructure testing. *Journal of Engineering Mechanics*, 128(12):1276–1284, 2002.
- [78] C. Thewalt and M. Roman. Performance parameters for pseudodynamic tests. *Journal of Structural Engineering*, 120(9):2768–2781, 1994.
- [79] C. Chen and J.M. Ricles. Analysis of actuator delay compensation methods for real-time testing. *Engineering Structures*, 31(11):2643–2655, 2009.

- [80] A. Blakeborough, M.S. Williams, A.P. Darby, and D.M. Williams. The development of real-time substructure testing. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 359(1786):1869–1891, 2001.
- [81] M. Ahmadizadeh, G. Mosqueda, and AM Reinhorn. Compensation of actuator delay and dynamics for real-time hybrid structural simulation. *Earthquake Engineering & Structural Dynamics*, 37(1):21–42, 2008.
- [82] C. Chen, J.M. Ricles, T.M. Marullo, and O. Mercan. Real-time hybrid testing using the unconditionally stable explicit or integration algorithm. *Earthquake Engineering & Structural Dynamics*, 38(1):23–44, 2008.
- [83] G. Mosqueda and M. Ahmadizadeh. Combined implicit or explicit integration steps for hybrid simulation. *Earthquake Engineering & Structural Dynamics*, 36(15):2325–2343, 2007.
- [84] C.R. Thewalt and S.A. Mahin. *Hybrid solution techniques for generalized pseudodynamic testing*, volume 87. Earthquake Engineering Research Center, College of Engineering, University of California; Springfield, Va.: available from the National Technical Information Service, 1987.
- [85] C.P. Lamarche, A. Bonelli, O.S. Bursi, and R. Tremblay. A Rosenbrock-w method for real-time dynamic substructuring and pseudo-dynamic testing. *Earthquake Engineering & Structural Dynamics*, 38(9):1071–1092, 2009.
- [86] C. Chen and J.M. Ricles. Development of direct integration algorithms for structural dynamics using discrete control theory. *Journal of Engineering Mechanics*, 134(8):676–683, 2008.
- [87] Masayoshi Nakashima and Nobuaki Masaoka. Real-time on-line test for MDOF systems. *Earthquake engineering & structural dynamics*, 28(4):393–420, 1999.
- [88] P. Pegon and A.V. Pinto. Pseudo-dynamic testing with substructuring at the ELSA laboratory. *Earthquake engineering & structural dynamics*, 29(7):905–925, 2000.
- [89] O.S. Bursi and P.S.B. Shing. Evaluation of some implicit time-stepping algorithms for pseudodynamic tests. *Earthquake engineering & structural dynamics*, 25(4):333–355, 1996.
- [90] O. S. Bursi, G. Abbiati, B. Wu, and G. Xu. Partitioned time integration methods for hybrid simulators. In *Proceedings of 15th World Conference on Earthquake Engineering*, 2012.
- [91] Inc MathWorks. *MATLAB: the language of technical computing. Desktop tools and development environment, version 7*, volume 9. MathWorks, 2005.
- [92] Gregory Bunting, Arun Prakash, Shirley Dyke, and Amin Maghareh. Characterizing errors and evaluating performance of transient simulations using multi-time-step integration. *ASCE Journal of Computing in Civil Engineering*, 2016.
- [93] S.J. Dyke, B.F. Spencer Jr, P. Quast, and M.K. Sain. Role of control-structure interaction in protective system design. *Journal of Engineering Mechanics*, 121(2):322–338, 1995.

- [94] Ge Ou, Ali Irmak Ozdagli, Shirley J. Dyke, and Bin Wu. Robust integrated actuator control: experimental verification and real-time hybrid-simulation implementation. *Earthquake Engineering & Structural Dynamics*, 44(3):441–460, 2015.
- [95] Anthony Friedman and Zhaoshuo Jiang. Phenomenological Bouc-Wen Model for 200 kN Large-Scale Magneto-Rheological Fluid (MR) Damper. <https://nees.org/resources/4122>, 2012.
- [96] B.F. Spencer Jr, J. Suhardjo, and M.K. Sain. Frequency domain optimal control strategies for aseismic protection. *Journal of Engineering Mechanics*, 120(1):135–158, 1994.
- [97] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity*. Springer, Berlin, 1998.
- [98] Jing Li, Jian Jia Chen, Kunal Agrawal, Chenyang Lu, Chris Gill, and Abusayeed Saifullah. Analysis of federated and global scheduling for parallel real-time tasks. In *2014 26th Euromicro Conference on Real-Time Systems*, pages 85–96. IEEE, 2014.
- [99] Zhang Xianyi, Wang Qian, and Zaheer Chothia. OpenBLAS: An optimized BLAS library. <http://www.openblas.net/>.
- [100] Speedgoat. *Speedgoat real-time simulation and testing*. Liebefeld Switzerland, 2016.
- [101] dSPACE Inc. *dSPACE Real-Time Systems*. Wixom, MI, 2016.
- [102] Gregory Bunting, Payton Lindsay, Amin Maghareh, Arun Prakash, and Shirley Dyke. Using multi-timestepping in finite element models to meet real time constraints. In *2012 Joint Conference of the Engineering Mechanics Institute and the 11th ASCE Joint Specialty Conference on Probabilistic Mechanics and Structural Reliability*, 2012.
- [103] P. Lindsay, M. Parks, and A. Prakash. Enabling fast, stable and accurate peridynamic computations using multi-time-step integration. *Computer Methods in Applied Mechanics and Engineering*, 2016.
- [104] J. Chakrabarty. *Theory of plasticity*. McGraw-Hill, New York, 1987.
- [105] Wei Song and Shirley Dyke. Real-time dynamic model updating of a hysteretic structural system. *Journal of Structural Engineering*, 140(3):04013082, 2013.
- [106] Wei Song. *Dynamic model updating with applications in structural and damping systems: From linear to nonlinear, from off-line to real-time*. PhD thesis, Purdue University, 2011.
- [107] T. Hohage, F. Schmidt, and L. Zschiedrich. Solving time-harmonic scattering problems based on the pole condition, II. convergence of the PML method. *SIAM J. Math. Anal.*, 35:547–560, 2003.
- [108] M. Lassas and E. Somersalo. On the existence and convergence of the solution of the PML equations. *Computing*, 60:229–241, 1998.

- [109] M. Lassas and E. Somersalo. Analysis of the PML equations in general convex geometry. *Proceedings of the Royal Society of Edinburgh*, pages 1183–1207, 2001.
- [110] J.L. Cippola and M.J. Butler. Infinite elements in the time domain using a prolate spheroidal multipole expansion. *International Journal for Numerical Methods in Engineering*, 43:889–908, 1998.
- [111] R.J. Astley. Mapped spheroidal wave-envelope elements for unbounded wave problems. *International Journal for Numerical Methods in Engineering*, 41(7):1235–1254, 1998.
- [112] R.J. Astley and J.P. Coyote. The performance of spheroidal infinite elements. *International Journal for Numerical Methods in Engineering*, 52(12):1379–1396, 2001.
- [113] R.J. Astley. Infinite elements for wave problems: a review of current formulations and an assessment of accuracy. *International Journal for Numerical Methods in Engineering*, 49(7):951–976, 2000.
- [114] D. S. Burnett and R. L. Holford. An ellipsoidal acoustic infinite element. *Computer Methods in Applied Mechanics and Engineering*, 164(1-2):49–76, 1998.
- [115] D. S. Burnett. A three-dimensional acoustic infinite element based on a prolate spheroidal multipole expansion. *JASA*, 96:2798–2816, 1994.
- [116] Y. Rui-Liang and W. Hong-zhen. A novel ellipsoidal acoustic infinite element. *Appl. Math. Mech.*, 26(2):261–268, 2005.
- [117] L. Demkowicz and J. Shen. A few new facts about infinite elements. *Computer Methods in Applied Mechanics and Engineering*, 195(29):3572–3590, 2006.
- [118] T.F. Walsh, A. Jones, M. Bhardwaj, C. Dohrmann, G. Reese, and R. Wilson. A comparison of transient infinite elements and transient kirchhoff integral methods for far field acoustic analysis. *Journal of Computational Acoustics*, 21(2), 2013.
- [119] D.S. Burnett. Radiation boundary conditions for the Helmholtz equation for ellipsoidal, prolate spheroidal, oblate spheroidal, and spherical domain boundaries. *Journal of Computational Acoustics*, 20(4), 2012.
- [120] F. Collino and P. Monk. The perfectly matched layer in curvilinear coordinates. *SIAM Journal on Scientific Computing*, 19(6):2061–2090, 1998.
- [121] P.J. Matuszyk and L. Demkowicz. Parametric finite elements, exact sequences, and perfectly matched layers. *Computational Mechanics*, 51(1):35–45, 2013.
- [122] Steven G Johnson. Notes on perfectly matched layers (pmls). *Lecture notes, Massachusetts Institute of Technology, Massachusetts*, 2008.
- [123] L. Demkowicz. *Computing with hp-Adaptive Finite Elements, Volume 1: One and Two Dimensional Elliptic and Maxwell Problems*. Chapman and Hall, CRC, 2007.

- [124] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszynski, W. Rachowicz, and A. Zdunek. *Computing with hp-Adaptive Finite Elements, Volume 2: Frontiers, Three Dimensional Elliptic and Maxwell Problems with Applications*. Chapman and Hall, CRC, 2008.
- [125] J.L. Shirron and T.E. Giddings. A finite element model for acoustic scattering from objects near a fluid-fluid interface. *Computer Methods in Applied Mechanics and Engineering*, 196:279–288, 2006.
- [126] J-P Berenger. Perfectly matched layer for the fdtd solution of wave-structure interaction problems. *Antennas and Propagation, IEEE Transactions on*, 44(1):110–117, 1996.
- [127] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.
- [128] Barry F. Smith, Petter Bjørstad, and William Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, New York, 1996.
- [129] Andrea Toselli and Olof B. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin Heidelberg New York, 2005.
- [130] Clark R. Dohrmann and Olof B. Widlund. An overlapping Schwarz algorithm for almost incompressible elasticity. *SIAM J. Numer. Anal.*, 47(4):2897–2923, 2009.
- [131] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee. On a class of preconditioners for solving the Helmholtz equation. *Appl. Numer. Math.*, 50(3-4):409–425, 2004.
- [132] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik. A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Sci. Comput.*, 27(4):1471–1492, 2006.
- [133] Jr. Roy R. Craig. *Structural Dynamics: An Introduction to Computer Methods*. John Wiley and Sons, New York, 1981.

VITA

## VITA

Gregory Bunting is a doctoral candidate in Civil Engineering. He obtained his BS (2010) and MS (2011) in Civil Engineering from Clemson University. While at Purdue, he has served as vice president of the Civil Engineering Graduate Student Advisory Council, and currently serves on the Intramural Student Advisory Board. He works primarily in Dr. Arun Prakash's Computational Solid and Structural Mechanics Laboratory, and also works in Dr. Shirley Dyke's Intelligent Infrastructure System Lab at the Bowen Laboratory for large-scale civil engineering research. He has interned with the Structural Dynamics and Solid Mechanics group at Sandia National Laboratories since May 2013. His areas of interest include computational mechanics and structural dynamics.