

# PROPELLANT SLOSH IN CONFORMAL TANKS

by

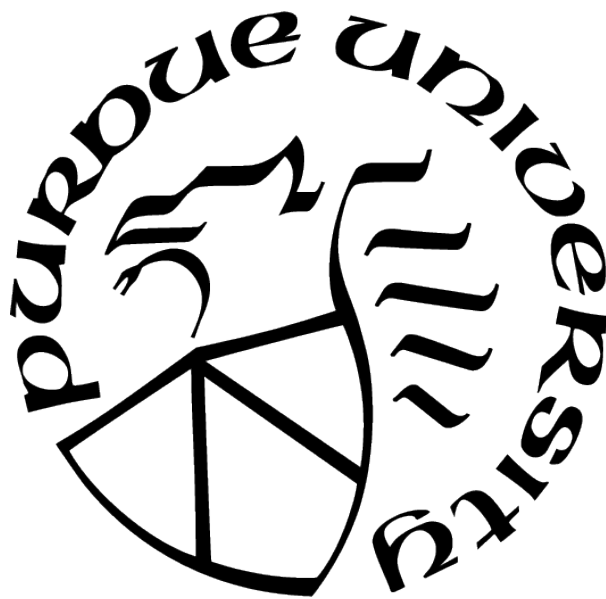
**Emily Beckman**

**A Thesis**

*Submitted to the Faculty of Purdue University*

*In Partial Fulfillment of the Requirements for the degree of*

**Master of Science in Aeronautics and Astronautics**



Aeronautics and Astronautics

West Lafayette, Indiana

December 2020

**THE PURDUE UNIVERSITY GRADUATE SCHOOL  
STATEMENT OF COMMITTEE APPROVAL**

**Dr. Steven Collicott, Chair**

School of Aeronautics and Astronautics

**Dr. Timothée Pourpoint**

School of Aeronautics and Astronautics

**Dr. Alina Alexeenko**

School of Aeronautics and Astronautics

**Approved by:**

Dr. Gregory Blaisdell

This thesis is dedicated to my partner, Silvio Mayolo, and my mom, Wendy Beckman, for their love and support, and to the Beckman cats, Pumpkin and Apollo, who supervised telework throughout the pandemic.

## ACKNOWLEDGMENTS

This work was supported by a NASA Space Technology Research Fellowship. The author appreciates the support of this program.

The author would also like to acknowledge the mentoring and support of David Benson from Goddard Space Flight Center, Bob Manning from NuSpace, and Brandon Williams, Hong Q. Yang, Jacob Brodnick, Marco Sansone, and Jeff West of Marshall Space Flight Center.

Finally, the author would like to thank her advisor, Steven Collicott, for his endless support and many opportunities provided throughout her time at Purdue University.



# TABLE OF CONTENTS

LIST OF TABLES . . . . .	9
LIST OF FIGURES . . . . .	11
LIST OF SYMBOLS . . . . .	16
ABBREVIATIONS . . . . .	17
ABSTRACT . . . . .	18
1 INTRODUCTION AND REVIEW OF LITERATURE . . . . .	19
1.1 Slosh . . . . .	19
1.1.1 SP-106 and Update . . . . .	20
1.1.2 Past Experimental Work . . . . .	21
1.1.3 Past CFD Work . . . . .	23
1.1.4 Oil and Natural Gas Industry . . . . .	25
1.2 Current Small Satellite Propulsion Systems . . . . .	27
1.2.1 VACCO Systems . . . . .	28
1.2.2 GOMSpace . . . . .	29
1.2.3 Aerojet-Rocketdyne . . . . .	29
1.2.4 Busek . . . . .	30
1.3 OpenFOAM Background . . . . .	31
2 EXPERIMENT . . . . .	32
2.1 Procedure . . . . .	32
2.1.1 Containers . . . . .	32
2.1.2 Setup . . . . .	35
2.1.3 Image Processor . . . . .	37
2.1.4 Parameter Calculation . . . . .	39
Damped Sinusoid . . . . .	39
Peak Curve Fit . . . . .	40

	Logarithmic Decrement . . . . .	41
2.1.5	Uncertainty . . . . .	41
2.2	Results . . . . .	42
2.2.1	Cylinder . . . . .	42
2.2.2	Cube . . . . .	46
2.2.3	Rectangular Prism . . . . .	49
2.2.4	Stair Step . . . . .	55
2.2.5	Trapezoid . . . . .	59
2.2.6	Quarter Annular Cylinder . . . . .	61
2.2.7	Analytical Comparison . . . . .	63
3	COMPUTATIONAL FLUID DYNAMICS . . . . .	65
3.1	Setup . . . . .	65
3.1.1	Server . . . . .	65
3.1.2	Solvers . . . . .	65
3.1.3	Initialization . . . . .	65
3.1.4	Properties . . . . .	66
3.1.5	Boundary Conditions . . . . .	67
3.1.6	Mesh . . . . .	67
	Cylinder . . . . .	67
	Cube . . . . .	70
	Rectangular Prism . . . . .	72
	Stair Step . . . . .	72
	Trapezoid . . . . .	74
	Quarter Annular Cylinder . . . . .	77
	Run Time . . . . .	77
3.1.7	Post-Processing Function Objects . . . . .	81
	Forces . . . . .	81
	Center of Mass . . . . .	83
3.1.8	Parameter Calculation . . . . .	84

3.2	Results . . . . .	85
3.2.1	Cylinder . . . . .	85
3.2.2	Cube . . . . .	89
3.2.3	Rectangular Prism . . . . .	96
3.2.4	Stair Step . . . . .	105
3.2.5	Trapezoid . . . . .	114
3.2.6	Quarter Annular Cylinder . . . . .	118
4	CONCLUSION . . . . .	134
5	RECOMMENDATIONS . . . . .	135
	REFERENCES . . . . .	137
A	OpenFOAM BACKGROUND . . . . .	141
A.1	System Directory . . . . .	142
A.1.1	Meshing . . . . .	142
A.1.2	controlDict . . . . .	148
A.1.3	fVSchemes . . . . .	151
A.1.4	fvSolution . . . . .	153
A.1.5	setFields . . . . .	158
A.1.6	decomposeParDict . . . . .	160
A.2	Constant Directory . . . . .	160
A.2.1	Gravity and Accelerations . . . . .	160
A.2.2	Transport Properties . . . . .	161
A.2.3	Turbulence Model . . . . .	161
A.2.4	Dynamic Mesh and Movement . . . . .	162
A.3	Time Directory . . . . .	163
A.3.1	Velocity . . . . .	164
A.3.2	Phase Fraction . . . . .	164
A.3.3	Pressure . . . . .	165
A.4	Running Simulations . . . . .	165

A.5	Post-Processing . . . . .	166
A.5.1	Para Foam . . . . .	166
A.5.2	Foam Log . . . . .	167
A.5.3	Function Objects . . . . .	167
B	CENTER OF MASS FUNCTION OBJECT . . . . .	169
B.1	CM.C . . . . .	169
B.2	CM.H . . . . .	173
C	PARAMETER CALCULATION . . . . .	176
C.1	Logarithmic Decrement . . . . .	176
C.2	Curve Fitting . . . . .	176
D	NON-DIMENSIONALIZATION . . . . .	179

## LIST OF TABLES

1.1	VACCO small satellite propulsion systems. . . . .	28
2.1	Camera specifications and usage by tank geometry. . . . .	37
2.2	Summary of Results for Cylindrical Tank . . . . .	42
2.3	Standard uncertainty in experimental values for the cylindrical container. . . . .	46
2.4	Summary of results for the cubic tank. . . . .	47
2.5	Standard uncertainty in experimental values for the cube tank. . . . .	48
2.6	Results for a rectangular tank. . . . .	51
2.7	Standard uncertainty in experimental values for the rectangular tank. . . . .	57
2.8	Experimental results for the stair step tank. . . . .	57
2.9	Standard uncertainty in experimental values for the stair step tank. . . . .	58
2.10	Experimental results for the trapezoid tank. . . . .	60
2.11	Standard uncertainty in experimental values for the trapezoid tank. . . . .	61
2.12	Experimental results for the quarter annulus tank. . . . .	61
2.13	Standard uncertainty in experimental values for the quarter annular cylinder tank. . . . .	64
3.1	Fluid properties used in CFD models. . . . .	66
3.2	Comparison of CFD, empirical, analytical, and experimental results in the cylindrical tank. . . . .	85
3.3	CFD results for the cube geometry at various fill fractions and slosh angles. . . . .	89
3.4	Summary of results for the rectangular tank at all slosh angles and fill levels. . . . .	96
3.5	Results of CFD simulations for the stair step tank. . . . .	105
3.6	Identifiable frequencies in the stair step tank. . . . .	107
3.7	Results of CFD simulations for the trapezoid tank. . . . .	115
3.8	Identifiable frequencies in the trapezoid tank. . . . .	117
3.9	Results of CFD simulations for the quarter annular cylinder tank. . . . .	119
3.10	Identifiable frequencies in the quarter annular cylinder tank. . . . .	126
A.1	Available OpenFOAM Interpolation Schemes. . . . .	152
A.2	Solvers available in OpenFOAM. . . . .	154
A.3	Available OpenFOAM preconditioners. . . . .	154
A.4	Available OpenFOAM smoothers. . . . .	155

A.5	Keywords for algorithms specifications in OpenFOAM. . . . .	157
A.6	Motion types available in OpenFOAM. . . . .	162

## LIST OF FIGURES

1.1	Slosh modelled in a natural gas tank with a vertical baffle [17]. . . . .	27
1.2	GOMSpace 3U cold gas propulsion system on the right, 6U system on the left [27], [28]. . . . .	29
1.3	GPIM Propulsion System Module for AF-M315E Propellant [30]. . . . .	30
1.4	Busek BGT-X5 green propellant propulsion system [34] . . . . .	31
2.1	Cube, rectangular prism, and cylinder tanks made out of clear acrylic. . . . .	32
2.2	CAD drawing of interior geometries for the stair step, trapezoid, and quarter annular cylinder tanks. . . . .	33
2.3	Images of the stair step, trapezoid, and quarter annular cylinder tanks. . . . .	33
2.4	Dimensions for the (a) stair step, (b) trapezoid, and (c) quarter annular cylinder containers. Top view on the left and side view on the right. All dimensions in millimeters. . . . .	34
2.5	Experimental setup with optional motor attached. . . . .	35
2.6	Illustration of slosh angle. . . . .	36
2.7	Comparison of slosh video frames before and after processing. (a) Original frame. (b) Frame after binarization. (c) Original frame image with surface points plotted. (d) Binarized frame with surface points plotted. . . . .	38
2.8	Example of data obtained for a slosh event in a cylindrical container as well as the best fit damped sinusoid. . . . .	43
2.9	Comparison of three fill fraction damping ratios to past experimental data as well as empirical correlations for a cylindrical tank [6] . . . . .	44
2.10	Comparison between experimental and analytical solutions for damping ratio in the cylindrical tank. . . . .	45
2.11	Comparison between experimental and analytical solutions for natural frequency in the cylindrical tank. . . . .	45
2.12	Example of data obtained for a slosh event in the cube container as well as the best fit damped sinusoid. This data is for a zero-degree slosh angle. . . . .	47
2.13	Damping ratio versus fill level for various slosh angles in the cube tank. . . . .	48
2.14	Comparison between experimental and analytical solutions for frequency in the cubic tank. . . . .	49
2.15	Example of data obtained for a zero-degree slosh event in the rectangular prism container as well as the best fit damped sinusoid. . . . .	50

2.16	Example of data obtained for a sixty-degree slosh event in the rectangular prism container as well as the best fit sum of two damped sinusoids. . . . .	50
2.17	Illustration of slosh modes and camera views. . . . .	52
2.18	Ratio of mode amplitudes over several slosh angles. . . . .	53
2.19	Damping ratio versus fill level for both modes and various slosh angles in the rectangular tank. . . . .	54
2.20	Natural frequency versus fill level for both modes and various slosh angles in the rectangular tank. . . . .	55
2.21	Example of data obtained for a slosh event in the stair step tank as well as the best fit sum of two damped sinusoids. . . . .	56
2.22	Power spectral density for the stair step tank at a sixty degree slosh angle. . . .	58
2.23	Example of data obtained for a slosh event in the trapezoid tank as well as the best fit sum of two damped sinusoids. . . . .	59
2.24	Power spectral density for the trapezoid tank at a forty-five degree slosh angle. .	60
2.25	Example of data obtained for a slosh event in the quarter annulus tank as well as the best fit sum of two damped sinusoids. . . . .	62
2.26	Power spectral density for the quarter tank at a forty-five degree slosh angle. . .	63
2.27	Experimental data for rectangular tanks fit to a hyperbolic tangent equation of the form used for the analytical solution. . . . .	64
3.1	Initialization of the liquid surface by angle, $\beta$ , from the horizontal. . . . .	66
3.2	Side view (left) and top view (right) of the (a) 83T cell mesh, (b) 2.2M cell mesh, and (c) 4.4M cell mesh for the cylinder. . . . .	68
3.3	Results of mesh convergence study in the cylindrical tank. . . . .	69
3.4	Results of mesh convergence study in the cube tank. . . . .	70
3.5	Side view (left) and top view (right) of the (a) 1.2M cell mesh, (b) 2.6M cell mesh, and (c) 4.2M cell mesh for the cube. . . . .	71
3.6	Results of mesh convergence study in the rectangular tank. . . . .	72
3.7	Side view (left) and top view (right) of the (a) 0.5M cell mesh, (b) 3M cell mesh, and (c) 4.4M cell mesh for the rectangular prism. . . . .	73
3.8	Results of mesh convergence study in the stair step tank. . . . .	74
3.9	Side view (left) and top view (right) of the (a) 1.2M cell mesh, (b) 2.4M cell mesh, and (c) 3.6M cell mesh for the stair step tank. . . . .	75
3.10	Results of mesh convergence study in the trapezoid tank. . . . .	76



3.11 Side view (left) and top view (right) of the (a) 1M cell mesh, (b) 2M cell mesh, and (c) 3.3M cell mesh for the trapezoid tank. . . . .	78
3.12 Results of mesh convergence study in the quarter annular cylinder tank. . . . .	79
3.13 Side view (left) and top view (right) of the (a) 1M cell mesh, (b) 2.4M cell mesh, and (c) 3M cell mesh for the quarter annular cylinder. . . . .	80
3.14 Run time for basic geometries. $t_R$ is run time, $t_S$ is simulation time, and $P$ is the number of processors. . . . .	81
3.15 Run time for complicated geometries. $t_R$ is run time, $t_S$ is simulation time, and $P$ is the number of processors. . . . .	82
3.16 Comparison of CFD, experimental, and analytical solutions for frequency in the cylindrical tank at tested fill levels. . . . .	86
3.17 Comparison of CFD, experimental, and average empirical solutions for damping ratio in the cylindrical tank at tested fill levels. . . . .	87
3.18 Plot of (a) center of mass, (b) forces, (c) moments, and (d) top-down center of mass trace for the cylindrical container. . . . .	88
3.19 Comparison of CFD, experimental, and analytical solutions for natural frequency in a cube container. . . . .	90
3.20 Comparison of CFD and experimental solutions for damping in a cube container. . . . .	91
3.21 Comparison of CFD and experimental solutions for damping with error cars at a fill level of (a) 51.4 mm, (b) 66.5 mm, and (c) 75.5 mm. . . . .	92
3.22 Center of mass over time in the cube for a slosh angle of (a) 0 degrees, (b) 30 degrees, (c) 45 degrees, and (d) 60 degrees. . . . .	93
3.23 Force on the walls over time in the cube for a slosh angle of (a) 0 degrees, (b) 30 degrees, (c) 45 degrees, and (d) 60 degrees. . . . .	94
3.24 Moment on the tank over time in the cube for a slosh angle of (a) 0 degrees, (b) 30 degrees, (c) 45 degrees, and (d) 60 degrees. . . . .	95
3.25 Top-down trace of the center of mass over time in the cube for a slosh angle of (a) 0 degrees, (b) 30 degrees, (c) 45 degrees, and (d) 60 degrees. . . . .	97
3.26 Comparison of CFD, experimental, and analytical solutions for mode one natural frequency in a rectangular container. . . . .	98
3.27 Comparison of CFD, experimental, and analytical solutions for mode two natural frequency in a rectangular container. . . . .	98
3.28 Comparison of CFD and experimental solutions for mode one damping ratio in a rectangular container. . . . .	99
3.29 Comparison of CFD and experimental solutions for mode two damping ratio in a rectangular container. . . . .	100

3.30	Comparison of CFD and experimental solutions for damping with error bars at a fill level of (a) 59.8 mm, (b) 70.2 mm, and (c) 80.8 mm. . . . .	101
3.31	Center of mass over time in the rectangular prism for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	102
3.32	Force on the walls over time in the rectangular prism for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	103
3.33	Moment on the tank over time in the rectangular prism for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	104
3.34	Top-down trace of the center of mass over time in the rectangular prism for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	106
3.35	Power spectral density of the CFD solution for the stair step tank at a zero degree slosh angle. . . . .	107
3.36	Comparison of overlapping frequency between CFD and experimental data in the stair step tank. . . . .	108
3.37	Comparison of overlapping damping ratio between CFD and experimental data in the stair step tank. . . . .	109
3.38	Center of mass over time in the stair step tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	110
3.39	Force on the walls over time in the stair step tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	111
3.40	Moment on the tank over time in the stair step tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	112
3.41	Top-down trace of the center of mass over time in the stair step tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	113
3.42	Independent slosh modes within the stair step tank acting along and perpendicular to the axis of symmetry. . . . .	114
3.43	Comparison of center of mass traced from CFD results and trace created by curve fitting. . . . .	115
3.44	Power spectral density of the CFD solution for the trapezoid tank at a thirty-degree slosh angle. . . . .	116
3.45	Comparison of overlapping frequency between CFD and experimental data in the trapezoid tank. . . . .	118
3.46	Comparison of overlapping damping ratio between CFD and experimental data in the trapezoid tank. . . . .	120
3.47	Center of mass over time in the trapezoid tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	121

3.48	Force on the walls over time in a trapezoid tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	122
3.49	Moment on the tank over time in the trapezoid tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	123
3.50	Top-down trace of the center of mass over time in the trapezoid tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	124
3.51	Power spectral density of the CFD solution for the quarter annular cylinder tank at a forty-five degree slosh angle. . . . .	125
3.52	Comparison of overlapping frequency between CFD and experimental data in the quarter annular cylinder. . . . .	127
3.53	Comparison of overlapping damping ratio between CFD and experimental data in the quarter annular cylinder. . . . .	128
3.54	Center of mass over time in the quarter annular cylinder for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	129
3.55	Force on the walls over time in a quarter annular cylinder for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	130
3.56	Moment on the tank over time in the quarter annular cylinder for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	131
3.57	Top-down trace of the center of mass over time in the quarter annular cylinder for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees. . . . .	132
3.58	Independent slosh modes within the quarter annular cylinder tank acting along and perpendicular to the axis of symmetry. . . . .	133
3.59	Comparison of center of mass traced from CFD results and trace created by curve fitting. . . . .	133
A.1	Numbering scheme for vertices and edges in blockMesh utility. . . . .	143
A.2	snappyHexMesh process of creating a background mesh, dividing cells, removing unwanted cells, and snappy boundary points to the object surface. . . . .	148
A.3	Illustration of rotatedBoxToCell region used in setFields. . . . .	159
A.4	ParaFoam view of a cube container with the surface initialized at an angle. . . .	167

## LIST OF SYMBOLS

$m$	mass
$v$	velocity
$\zeta$	damping ratio
$\omega$	frequency
$R$	radius
$g$	gravitational acceleration
$h$	height of liquid surface
$a$	edge length
$\Omega$	frequency of oscillation
$X$	amplitude of oscillation
$F$	force
$M$	moment
$m$	mass
$\lambda$	roots of eigenvalue equation
$\nu$	kinematic viscosity
$Re$	Reynold's number
$\theta$	slosh angle
$A$	amplitude
$t$	time
$b$	damping constant
$\Phi$	phase shift
$\beta$	surface initialization angle
$\rho$	density
$V$	volume
$\alpha$	volume fraction
$d$	distance from origin

## ABBREVIATIONS

CFD	computational fluid dynamics
PMD	propellant management device
NASA	national aeronautics and space administration
NSTRF	NASA space technology research fellowship
RCS	reaction control system
NEAR	near earth asteroid rendezvous
SP	special publication
OpenFOAM	open source field operation and manipulation
MarCO	mars cube one
GPIM	green propellant infusion mission
ATK	alliant techsystems
FPS	frames per second
MATLAB	matrix laboratory
M	million
T	thousand
CM	center of mass
PISO	pressure-implicit split-operator
SIMPLE	semi-implicit method for pressure-linked equations
VTK	visualization toolkit

# ABSTRACT

As small satellites are increasingly used in the space industry, creative solutions for the use of their limited volume will be required. Conformal tanks are one idea to better make use of this volume. These tanks are non-traditionally shaped and non-axisymmetric. Because slosh can have detrimental effects on a spacecraft, it should be understood. However, slosh in these more complicated geometries has not been thoroughly investigated in the past.

This research looks at slosh within six geometries, five of which are conformal tanks. These geometries are evaluated in both an experiment and using CFD simulations. It was found that the total slosh motion appears to be the sum of slosh behavior along each dimension. Slosh along a line of symmetry will have center of mass movement that stays along that line. Slosh off the line of symmetry will deviate from that line unless slosh frequency is the same in each direction.

# 1. INTRODUCTION AND REVIEW OF LITERATURE

Conformal tanks are designed such that they make efficient use of the available volume in a spacecraft. While the surface area to volume ratio of a sphere may be optimal, it may be preferable to choose a design which allows for more compact placement of components around the tank in small satellites. For example, it is significantly easier to pack electronics closely around a rectangular prism-shaped tank than it would be to pack the same electronics around a sphere.

The three basic tank geometries analyzed in this study were a cylinder, a cube, and a rectangular prism. The cylinder has already been well-studied, which allows comparison between this and past studies. The cube and rectangular prism are good examples of simple conformal tank geometries and therefore are important stepping-stones in understanding non-axisymmetric slosh. Three more-complicated conformal geometries including a stair step, trapezoid, and a quarter annular cylinder were also investigated to determine the impacts of this greater complexity.

## 1.1 Slosh

Slosh of liquid propellant in a spacecraft can be caused by normal spacecraft operations such as thrusters firing, reaction control system (RCS) operations, launch vehicle separation, and any other motion of the spacecraft. Slosh causes forces and moments on the spacecraft which can affect the stability. This can cause loss of observation time, especially for satellites with tight pointing requirements [1]. In more severe cases, the loss of the spacecraft is possible. The NEAR Shoemaker satellite was nearly lost at the beginning of the mission due to unexpected spacecraft movement, likely due to slosh. Although it was eventually recovered, the spacecraft entered safe mode and the mission was delayed by a year [2]. Because of these possible detrimental effects to the spacecraft, slosh should be understood and included in guidance, navigation, and control analysis.

Some of the most important slosh parameters are natural frequency, damping ratio, and the slosh mass [3]. Therefore, these are the parameters most investigated in this research.

### 1.1.1 SP-106 and Update

“The New “Dynamic Behavior of Liquids in Moving Containers”” is the update and addition to the NASA SP-106, a 1966 publication which summarized then-current slosh and related work [4], [5]. It contains much information about slosh, including analytical solutions and empirical correlations for some geometries. Of particular interest to this research are the analytical solutions for forces exerted on a rectangular or cylindrical container given an oscillatory amplitude, analytical solutions for natural slosh frequencies in a cylindrical or rectangular container, and empirical relations for the damping ratio of slosh in a cylindrical tank [5].

Most of the slosh of interest to this research is in the linear regime. The linear regime for slosh is the regime where the amplitude of the slosh is linearly proportional to the amplitude of the tank motion. Natural frequency and damping are independent from the slosh amplitude. This occurs when the amplitude of the slosh at the wall of the container is no larger than approximately 0.1R. This translates to an initial surface angle of no more than 5.7 degrees [5].

Assuming that the liquid is inviscid, incompressible, and has no vorticity, potential flow can be used to determine the natural frequency of slosh (Eqn 1.1) as well as the forces (Eqn 1.2) and moments (Eqn 1.3) exerted on the tank when a constant amplitude oscillatory motion is applied. In these equations,  $h$  is the height of the free surface at rest,  $a$  is the length of the rectangular prism in the direction of oscillation and slosh,  $\Omega$  is the frequency of oscillation, and  $X_0$  is the amplitude of oscillation [5].

$$\omega_n^2 = \pi(2n - 1) \left( \frac{g}{a} \right) \tanh \left[ \pi(2n - 1) \left( \frac{h}{a} \right) \right] \quad (1.1)$$

$$\frac{F_{x0}}{-i\Omega^2 X_0 m_{liq}} = 1 + 8 \frac{a}{h} \sum_{n=1}^N \frac{\tanh [(2n - 1)\pi h/a]}{(2n - 1)^3 \pi^3} \frac{\Omega^2}{\omega_n^2 - \Omega^2} \quad (1.2)$$

$$\begin{aligned} \frac{M_{y0}}{-i\Omega^2 X_0 m_{liq} h} &= \frac{1}{12} \left( \frac{a}{h} \right)^2 + 8 \frac{a}{h} \sum_{n=1}^N \frac{\tanh [(2n - 1)\pi h/a] \Omega^2}{(2n - 1)^3 \pi^3 (\omega_n^2 - \Omega^2)} \times \\ &\quad \left( \frac{1}{2} - \frac{2(a/h) \tanh [(2n - 1)\pi h/(2a)]}{(2n - 1)\pi} + \frac{g}{h\omega_n^2} \right) \frac{\Omega^2}{\omega_n^2 - \Omega^2} \end{aligned} \quad (1.3)$$



The full solutions to the cylinder are not given due to greater difficulty solving the same equations with boundary conditions not aligned with the cartesian grid. However, a solution for the natural frequency of slosh is given in Equation 1.4. In this equation,  $a$  is the tank radius.

$$\omega_n^2 = \frac{g\lambda_{mn}}{a} \tanh \left[ \frac{\lambda_{mn}h}{a} \right] \quad (1.4)$$

The  $\lambda_{mn}$  are roots of the eigenvalue equation. The  $m = 1$  modes are the modes of most interest because they are the ones which create net forces and moments. The  $\lambda_{1n}$  modes are denoted by  $\xi_n$  and are take the values given in Equation 1.5 [5].

$$\xi_1 = 1.1841 \quad \xi_2 = 5.331 \quad \xi_3 = 8.536 \quad \dots \quad \xi_{n+1} = \xi_n + \pi \quad (1.5)$$

In addition to the analytical solution to natural frequency available for cylindrical tanks, two empirical correlations are given for the damping ratio. The first was given by Mikishev and Dorozhkin (Eqn 1.6) and the second was given by Stephens (Eqn 1.7). Both require the Reynold's number as given in Equation 1.8. In these equations,  $R$  is the tank radius and  $\nu$  is the kinematic viscosity [5].

$$\zeta = 0.79\sqrt{Re_1} \left[ 1 + \frac{0.318}{\sinh(1.84h/R)} \left( 1 + \frac{1 - h/R}{\cosh(1.84h/R)} \right) \right] \quad (1.6)$$

$$\zeta = 0.83\sqrt{Re_1} \left[ \tanh(1.84h/R) \left( 1 + 2\frac{1 - h/R}{\cosh(3.68h/R)} \right) \right] \quad (1.7)$$

$$Re_1 = \frac{\nu}{\sqrt{gR^3}} \quad (1.8)$$

### 1.1.2 Past Experimental Work

As mentioned, “The New “Dynamic Behavior of Liquids in Moving Containers”” contains empirical correlations for damping ratio for some experimental slosh data [5]. While primarily about CFD work, Yang et. al. also summarizes some experimental slosh data. This data

is compared to the empirical correlations. The experimental slosh data follows the trend, although does vary, as is typical, for damping ratio. These correlations and experimental results are only for cylindrical containers [6].

The Southwest Research Institute did some research on the impact of compartmentalizing cylindrical tanks into sectors or an inner cylinder with an outer annulus [7]. This research was done with the objective of altering the natural frequency of the slosh. At that time, launch vehicles were increasing in size, resulting in concern that the natural frequency of the slosh was decreasing and getting close to the spacecraft control frequency. Compartmentalization was a solution to this [8]. The research reported the natural frequency of an uncompartmented tank, tanks with 180, 90, and 45-degree sectors, and annular tanks with three different ratios of inner to outer sector diameters. Damping ratio was not addressed in this research [7].

A rectangular tank has been investigated in the past with tanker truck applications. This experiment applied a sinusoidal excitation to the tank, with the excitation direction parallel to the long edge of the tank. No other slosh angles were investigated. The surface height and the pressure at the wall was recorded for use in matching CFD results. Neither the natural frequency nor damping ratio was reported [9].

A recent look at in-flight slosh was given for the Cassini spacecraft, which was about 54% propellant by mass at the beginning of the mission. Slosh during flight maneuvers was observable using telemetry data. The main propellant tanks were partially compartmented due to the use of a surface tension type PMD. Because of this, both full tank and sector slosh modes were observable. The paper found that the predicted sector slosh frequencies were within 1-3% of the actual frequencies and predicted full tank frequencies were within 6%. Low-gravity slosh results are reported but not discussed in depth [10].

Salzman and Masica took a look at slosh in low gravity using a drop tower in 1968. Using multiple liquids, accelerations, and size cylinders, they were able to find the natural frequency of slosh for many different bond numbers from 0.0008 to 14.9. Some cases also reported logarithmic decrement, but not all. These results may be used to help verify low gravity slosh simulations [11].

While this thesis only looks at the linear slosh regime, it is worth noting that slosh research in the past has looked at the impact of nearing or entering the nonlinear regime. The nonlinear regime occurs when slosh amplitudes are high, caused by either a large input force or an excitation frequency near the slosh natural frequency. At this point, there may be waves or breakage at the liquid surface which can cause swirl within the tank [12].

### 1.1.3 Past CFD Work

Coogan and Musgrove provide a detailed look at best practices discovered for CFD modeling of slosh. Their recommendations are specifically for the FLOW-3D program, but these recommendations may be taken as a starting point for other programs. When creating the mesh, they recommend that:

1. Cell aspect ratio should be below 3.
2. Ratio of adjacent cell lengths in any given direction should be less than 1.25.
3. The mesh should have at least 100 cells across the tank in any given direction at the liquid free surface.
4. If using a locally refined mesh near the liquid free surface, the refined area should extend at least 3-4 times the height of the initial slosh amplitude.
5. Size ratios between adjacent mesh blocks should be less than 2.
6. If the tank contains baffles, the mesh in that region should be at least as well defined as at the liquid free surface.

When creating their model, they specified all walls as no slip, initialized the surface at a five-degree angle (nonlinear slosh), and neglected heat transfer, phase change, and surface tension. Slosh parameters were calculated from pressure force at the tank walls. Viscous force is negligible compared to pressure force [3].

Another detailed look at slosh modeling practices is given by Yang, et al. This paper discusses the difficulties in modeling slosh with extremely low damping ratio. The researchers

used the CFD-ACE+ code to model slosh with the volume of fluid method. Using existing experimental data, they completed a mesh convergence study for multiple sizes of bare tanks as well as a tank with ring baffles [6].

Bare tanks were modeled with a no slip condition at all walls except the top which was set to a fixed pressure. A butterfly grid was used to avoid poor quality cells at the tank center. The liquid surface was given an initial slope of 5.7 degrees. The time step was allowed to vary with a Courant number of 0.1 and a maximum time step of 0.5 ms. They used a second order temporal scheme and a spatial scheme that was 99% second order and 1% first order upwind. Damping ratio was calculated using the log decrement method on the total center of mass displacement [6].

The geometry of most interest to this research is a tank with a radius of 3.8 cm which is similar to the tank size investigated in this research. The four mesh sizes investigated were 40 thousand, 256 thousand, and 1 million cells, corresponding to 60, 100, and 160 cells across the tank diameter. The natural frequency obtained from each mesh was nearly the same and matched the experimental data well. When compared to the experimental value for damping ratio, the different quality meshes resulted in an error of approximately 230%, 6.6%, and 1.8% respectively. It is clearly important to have a good quality mesh [6].

Marsell simulated slosh in an eight-inch diameter sphere using the Fluent CFD program and the volume of fluid method. No slip boundary conditions were applied to all walls except the top of the tank which was set as a pressure inlet at atmospheric pressure. The simulation had a time step of 0.01 seconds and used a laminar turbulence model. The model used second order schemes in space and first order schemes in time. Both the experiment and the CFD case were started with an initial small impulse instead of the more typical sloped surface offset [13].

This group used a curve fit of the form  $\exp(-bt)$  to find the damping rate,  $b$ . This fit was applied to the peaks of data from a force transducer in the experimental setup, and peaks in a nondimensional force coefficient in the CFD model. The natural frequency had a 3.12% error from the experimental [13].

Benson and Ng worked to validate slosh models using STAR-CCM+, another commercially available CFD code. The two tanks used for the verification were a bare cylinder and a

cylinder with a single ring baffle. The former will be most discussed here. Both a polyhedral and a trim mesh from 305 thousand to 1.19 million cells were tested. The center of mass displacement was used to determine the natural frequency and damping ratio because it shows less noise than either force or moment data. All meshes predicted the natural frequency well. The damping ratio was calculated using an exponential curve fit to the peaks and valleys of the center of mass data. The finest polyhedral mesh predicted the damping within 1.7% when compared to empirical correlations as explained above [1].

In general, studies find that matching the damping ratio is significantly more difficult than matching the natural frequency. For example, a recent paper by Manning found an error in the damping ratio of about 28% even for a mesh with about 12 million cells [14].

While most slosh studies have been done in commercially available codes, some work has been done in OpenFOAM, an open-source CFD software. For example, Chaichanasiri and Suvanjumrat used OpenFOAM to simulate slosh in rectangular tanks. This research was done for tanker trucks, as will be discussed in more detail in the next section. This study validated their OpenFOAM models using experimental data for surface height and pressure. The error between the CFD and experimental were 2.33% and 6.65% respectively. Interestingly, meshes using only up to 38,800 cells were used. The authors did not investigate the damping ratio of the slosh, which seem to be of less interest to fuel tankers than to space applications [9]. It is possible that both pressure and surface height would be easier to match with an unrefined mesh, similar to natural frequency.

#### 1.1.4 Oil and Natural Gas Industry

Another group interested in liquid slosh is the crude oil and natural gas industry. Both crude and natural gas are frequently transported from the location of production to the consumption region by ship [15]. Slosh within these carriers has been under investigation due to the possibility of structural damage to the cargo tanks. This is increasingly a problem due to the use of supersize carriers. The carriers sometimes transport their cargo with the tanks partially filled, increasing risk of large slosh waves and unknown behavior at these arbitrary fill levels [16]. Because these tanks are rectangular, natural gas research is relevant to this work.

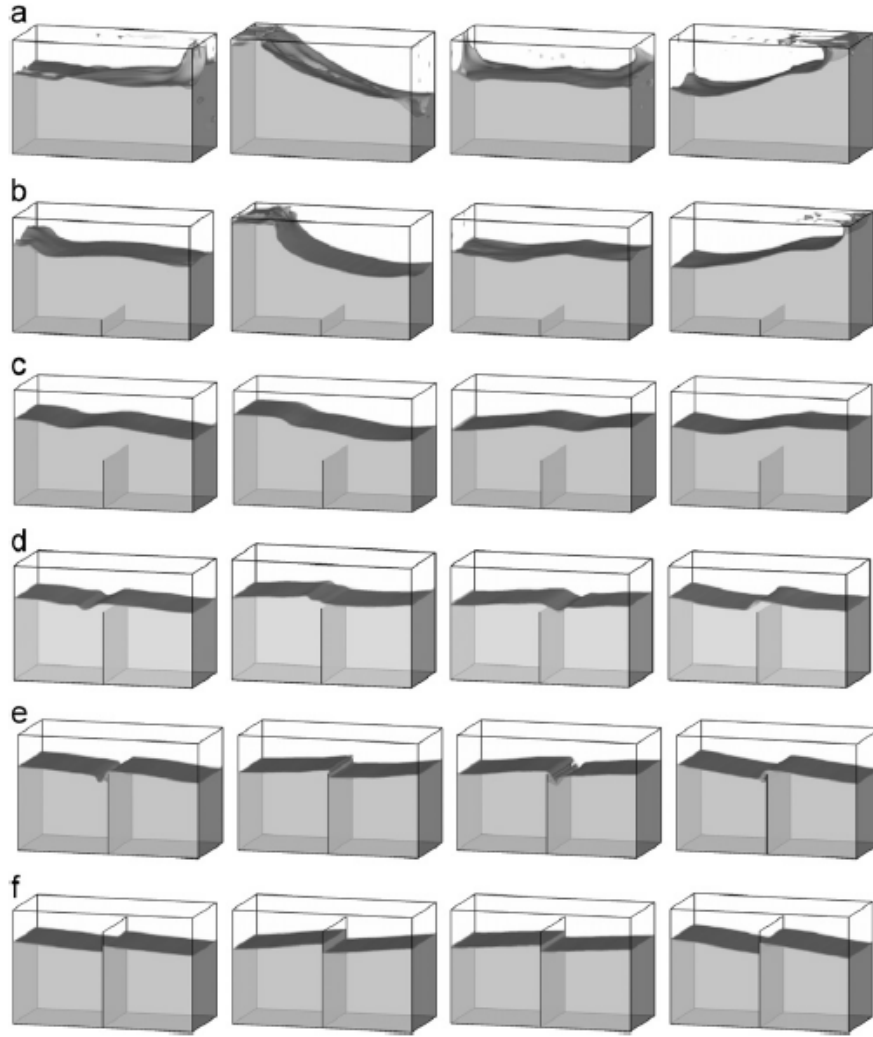
However, it appears that the natural gas industry is interested in different parameters, motions, and slosh regimes than the space industry. In general, the space industry is more interested in the frequency of slosh, the damping ratio, and center of mass movement due to the impact on the stability of the spacecraft. The natural gas industry appears to be more concerned about the pressure on walls due to large slosh events and the natural frequency of impacted walls when submerged in the liquid. For example, one study completed an experimental investigation of a rectangular tank made of acrylic except for one steel plate wall. The natural frequency of this steel plate was reported instead of the natural frequency of the slosh [15]. The expected natural frequency of the slosh for the given geometry is very different than that of the plate.

The input motion for this experiment was a rolling motion meant to imitate the motion of a ship in the ocean. It does not appear that the natural decay of the slosh at the end of the motion was investigated. Based on the reported results for surface profile, the experiment also looked at slosh in the nonlinear regime due to the presence of a breaking wave [15].

Another paper investigated slosh in a similar tank using CFD verified by the discussed experimental results. The authors discovered that change in the use of turbulent/laminar model, ratio of liquid and gas density, or ullage pressure did not appear to have a significant impact on the resulting slosh impact pressure on the walls. The exception was when the ullage pressure was set below the liquid critical pressure. Below this point, a different model would be needed to account for bubbles and thermodynamic effects [16].

Other studies have looked at the impact of a baffle on the slosh within a rectangular container. This baffle is vertical and is perpendicular to the slosh motion as shown in Figure 1.1. The main goal of the research appeared to be to investigate the impact of baffle height on the pressure at the wall and whether slosh waves reach the top of the tank [17]. The placement of the baffle is different than a surface tension type PMD which might be used in a satellite. This design would result in a significant volume of trapped propellant in a low gravity environment.

While the research done on rectangular tanks in the natural gas industry is of interest to this work, it does not sufficiently cover the material this work is interested in. The reported experimental data is different than that needed for spaceflight, because it looks at



**Figure 1.1.** Slosh modelled in a natural gas tank with a vertical baffle [17].

the nonlinear slosh regime, provides a different input motion, does not use a baffle of interest to space flight, and does not look at slosh along asymmetric angles. Research in the natural gas industry appears to look at slosh only on a zero-degree slosh angle. These are the gaps this work attempts to fill.

## 1.2 Current Small Satellite Propulsion Systems

The following is a brief look at currently available small satellite propulsion systems of interest to this research. These systems show the increasing interest in small satellite

propulsion systems, give a general idea about the size of propellant tank needed, and show types of propellants used.

### 1.2.1 VACCO Systems

The first interplanetary cube satellites were the Mars Cube One (MarCO) satellites which travelled to Mars and relayed communications between the InSight lander and Earth during the entry, descent, and landing phase [18]. This mission demonstrated the capabilities of cube satellites in deep space and will likely pave the way for more cubeSat applications in the future.

The MarCO satellites included a cold-gas propulsion system created by VACCO Industries [19]. This system is approximately 8.1 x 6.3 x 1.6 inches, contains four axial and four RCS 25mN thrusters, and runs on R236fa propellant [20]. The propulsion system has a 3.49 kg wet and a 1.65 kg dry mass [20], [21]. The total wet spacecraft mass was 13.5 kg [22]. This means that the spacecraft was approximately 13.6% propellant by mass.

VACCO has also created several other cube satellite propulsion systems which are summarized in Table 1.1.

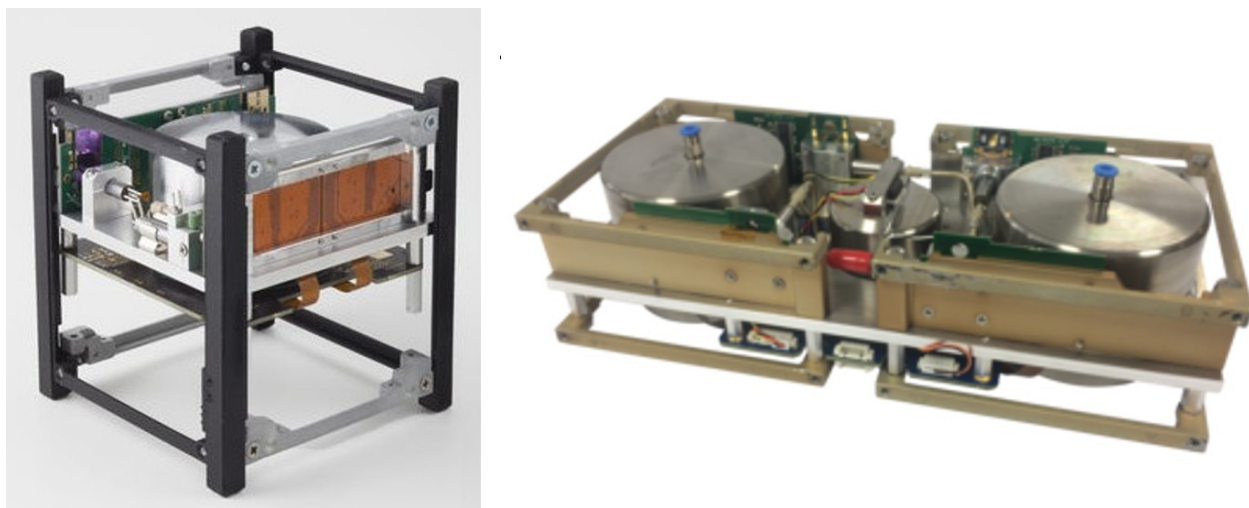
**Table 1.1.** VACCO small satellite propulsion systems.

System	Propellant	Size (in)	Thrust (mN)	Total Impulse (N-s)
MarCO Propulsion System [20]	R236fa	8.1x6.3x1.6	200	755
Propulsion Units of CubeSats [23]	Warm Gas	3.5x3.5x2.2 - 3.5x3.5x5.6	5.4	183 - 595
MEMS PicoSat Inspector Micro-Propulsion System [24]	Cold Gas	3.6x3.6x~1	53	23
Palomar Micro Propulsion System [24]	Isobutane/Variable	3.9x3.9x4.2	35	85
Reaction Control Propulsion Module [25]	R134a	3.9x3.7x4	200	166
Standard Micro-Propulsion System [26]	R134a	3.5x3.5x1.2 - 3.5x3.5x3.9	10	44 - 250



### 1.2.2 GOMSpace

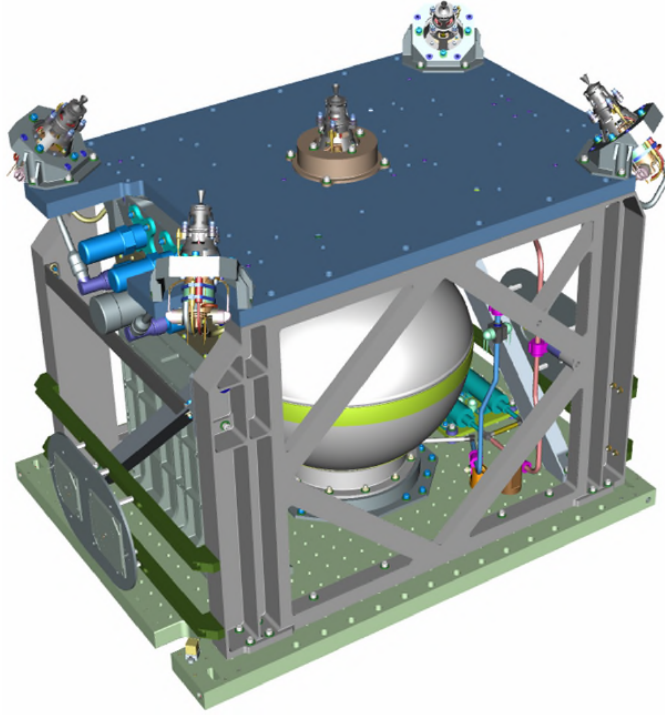
GOMSpace Sweden, formerly NanoSpace, has developed a cold gas propulsion system which can be used on either a 3U or 6U satellite. The module designed for a 3U satellite was successfully flight tested in 2015. This module is capable of producing 1mN of thrust with a total impulse of 40 Ns. The mass is 350 g wet and uses Butane as a propellant. The propulsion system required 0.5U volume of the spacecraft [27]. The 6U version is very similar to the 3U module but has two Butane propellant tanks. The total impulse of this design is 80 Ns with a thrust of 1 mN or 10 mN [28]. Both operate at low pressures of 29-72.5 psi. Both designs appear to utilize cylindrical tanks and are picture in Figure 1.2 [27], [28].



**Figure 1.2.** GOMSpace 3U cold gas propulsion system on the right, 6U system on the left [27], [28].

### 1.2.3 Aerojet-Rocketdyne

Aerojet-Rocketdyne has developed multiple small satellite propulsion systems including the propulsion system for the Green Propellant Infusion Mission (GPIM). This mission was launched in June 2019 and successfully demonstrated the use of a green propellant propulsion system in space [29]. The propulsion system included five 1-N class thrusters and an ATK 80581 diaphragm tank [30]. This tank is spherical with a 12.8-inch inner diameter [31].

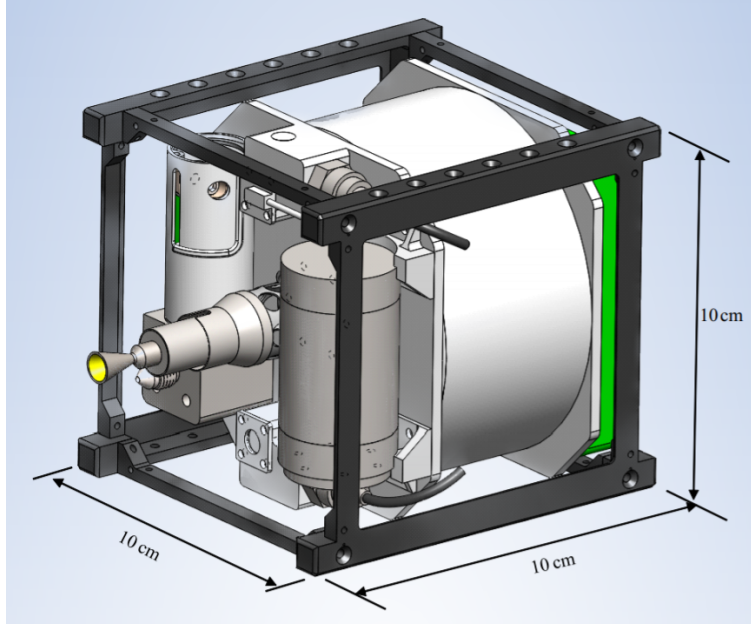


**Figure 1.3.** GPIM Propulsion System Module for AF-M315E Propellant [30].

In addition to the propulsion system created for the GPIM mission, Aerojet-Rocketdyne has developed a line of CubeSat propulsion systems ranging from 1U to 8U and designed for either hydrazine or green propellant. The 1U and 2U designs are piston fed, while the 4, 6, and 8U systems are pump fed with PMD tanks. From the provided images, it appears that the piston fed tanks are cylindrical and the pump fed tanks are rectangular. Total impulse for these propulsion systems range from 1130 to 19360 N-sec [32].

#### 1.2.4 Busek

Busek Co. Inc. offers several small satellite compatible propulsion systems including radio frequency ion, green monopropellant, micro-pulsed plasma, and hall effect thrusters [33]. The green propellant system is likely most relevant to this research. This system has a 1U volume, 500 mN thrust, and 565 N-sec total impulse. This system is pictured in Figure 1.4 and appears to have a cylindrical propellant tank [34].



**Figure 1.4.** Busek BGT-X5 green propellant propulsion system [34]

### 1.3 OpenFOAM Background

Open Source Field Operation and Manipulation (OpenFOAM) is an open source software package which may be used to develop computational fluid dynamics (CFD) codes. This software package is the CFD code most used in this research. It is based on a framework of C++ libraries and, in addition to the pre-built solvers and utilities, users may create their own applications. OpenFOAM does not have a user interface, so all simulations are coded in a system of directories and text files which specify solvers, constants, and conditions. Detailed information about constructing simulations is available in Appendix A.

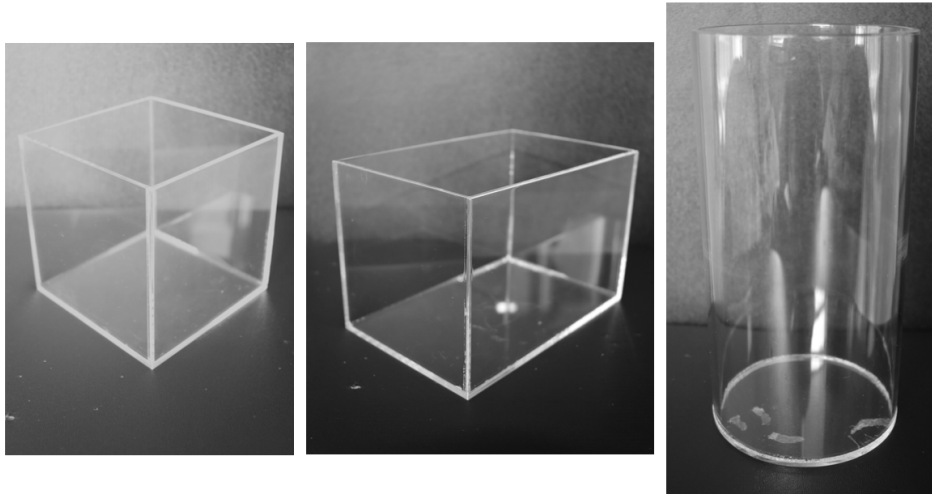
## 2. EXPERIMENT

Due to the lack of experimental data for slosh in non-axisymmetric containers, especially at non-zero slosh angles, an experiment was designed to obtain the damping ratio and frequency of slosh in the linear slosh regime for various tank geometries. Free decay behavior was recorded and the video processed to obtain the surface height over time. This data was used to calculate the desired slosh parameters.

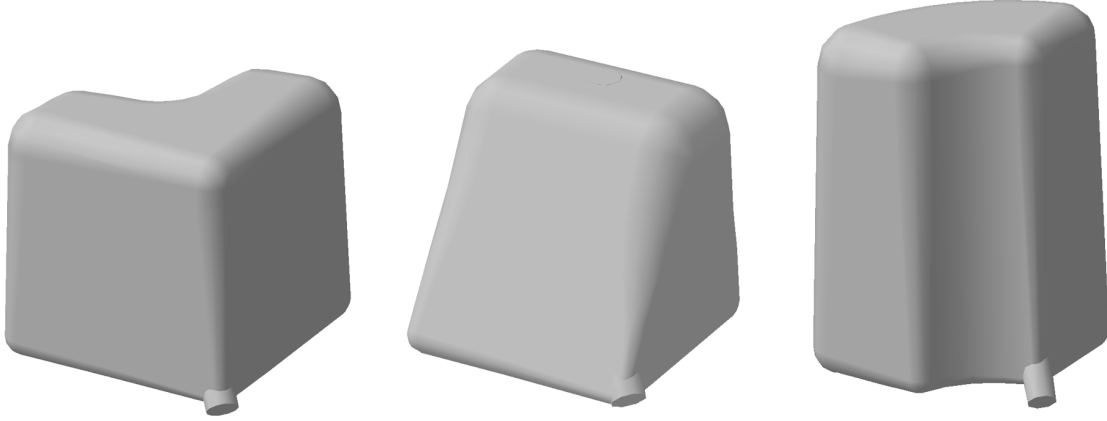
### 2.1 Procedure

#### 2.1.1 Containers

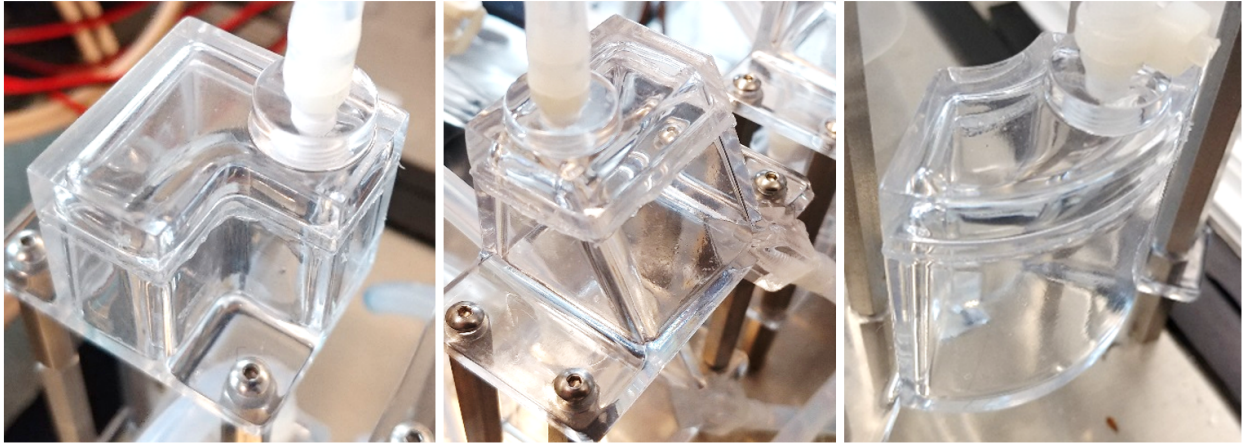
Six containers were used in this experiment. The first three are referred to as basic geometries and were a cylinder, cube, and rectangular prism. All three were made of clear acrylic for good visualization of the liquid-gas interface. The cylindrical container was chosen due to prior experimental investigations of the geometry. Results in this experiment can be compared to past experiments to increase the confidence in this experimental setup. The cube and rectangular prism were chosen as basic non-axisymmetric geometries which are good for initial investigations of slosh behavior. These basic geometries are pictured in Figure 2.1.



**Figure 2.1.** Cube, rectangular prism, and cylinder tanks made out of clear acrylic.



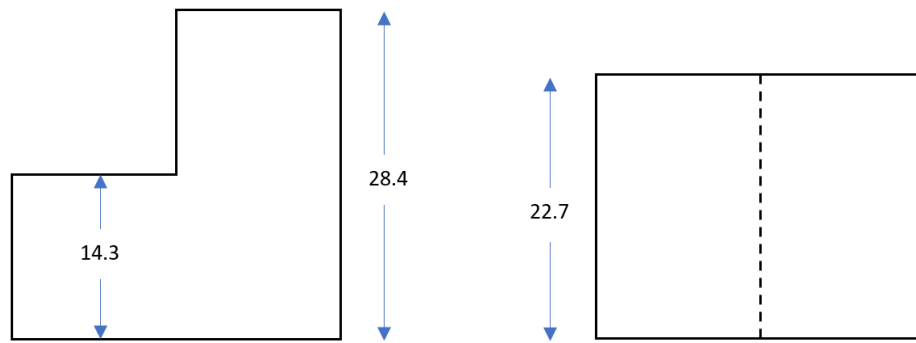
**Figure 2.2.** CAD drawing of interior geometries for the stair step, trapezoid, and quarter annular cylinder tanks.



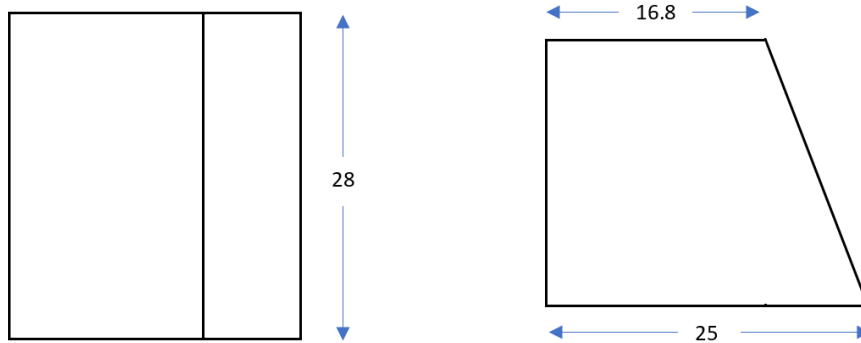
**Figure 2.3.** Images of the stair step, trapezoid, and quarter annular cylinder tanks.

The diameter of the cylinder is 94.5 mm, the cube is 96.1 mm in width, and the rectangular prism is 95.9 by 146.8 mm. These are inner dimensions. Liquid depths are discussed in the results section.

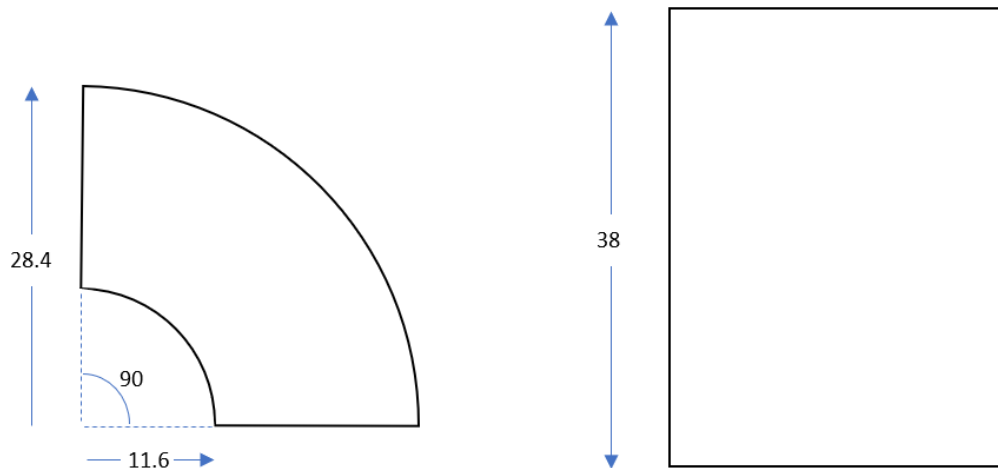
The other three containers are more complicated with more varied geometries and edge fillets designed to allow full draining during zero gravity conditions. These consist of a stair step, a trapezoid, and a quarter annular cylinder. CAD of the interior geometries are pictured in Figure 2.2, images of the tanks are given in Figure 2.3, and basic dimensions are given in Figure 2.4. These tanks were 3D printed in transparent material with surface polishing and coating to provide a smooth finish.



(a)



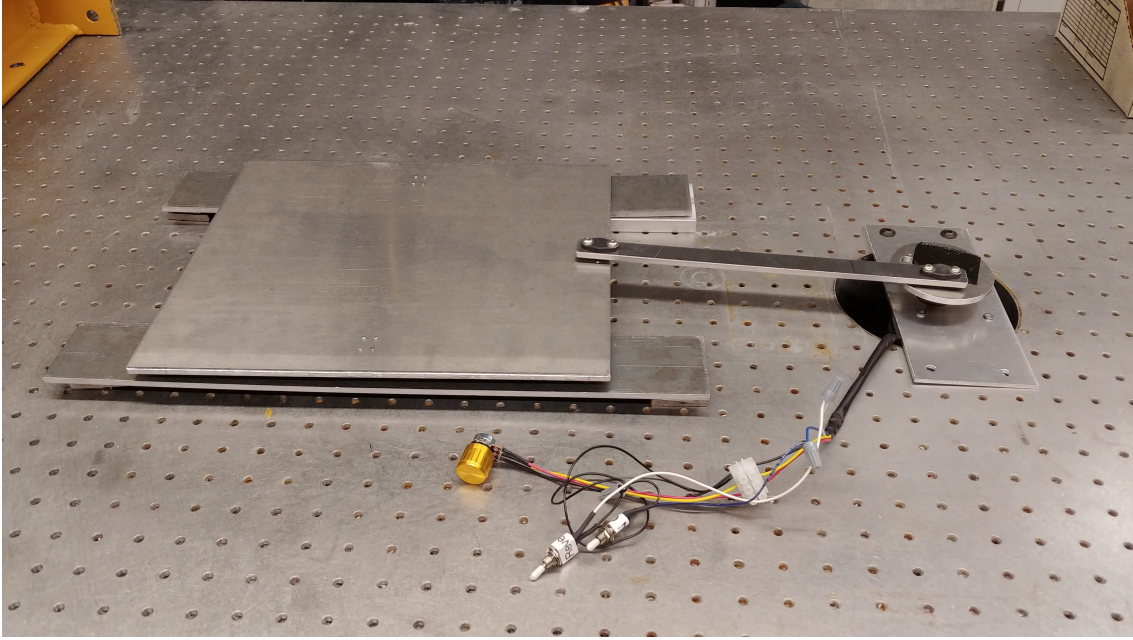
(b)



(c)

**Figure 2.4.** Dimensions for the (a) stair step, (b) trapezoid, and (c) quarter annular cylinder containers. Top view on the left and side view on the right. All dimensions in millimeters.





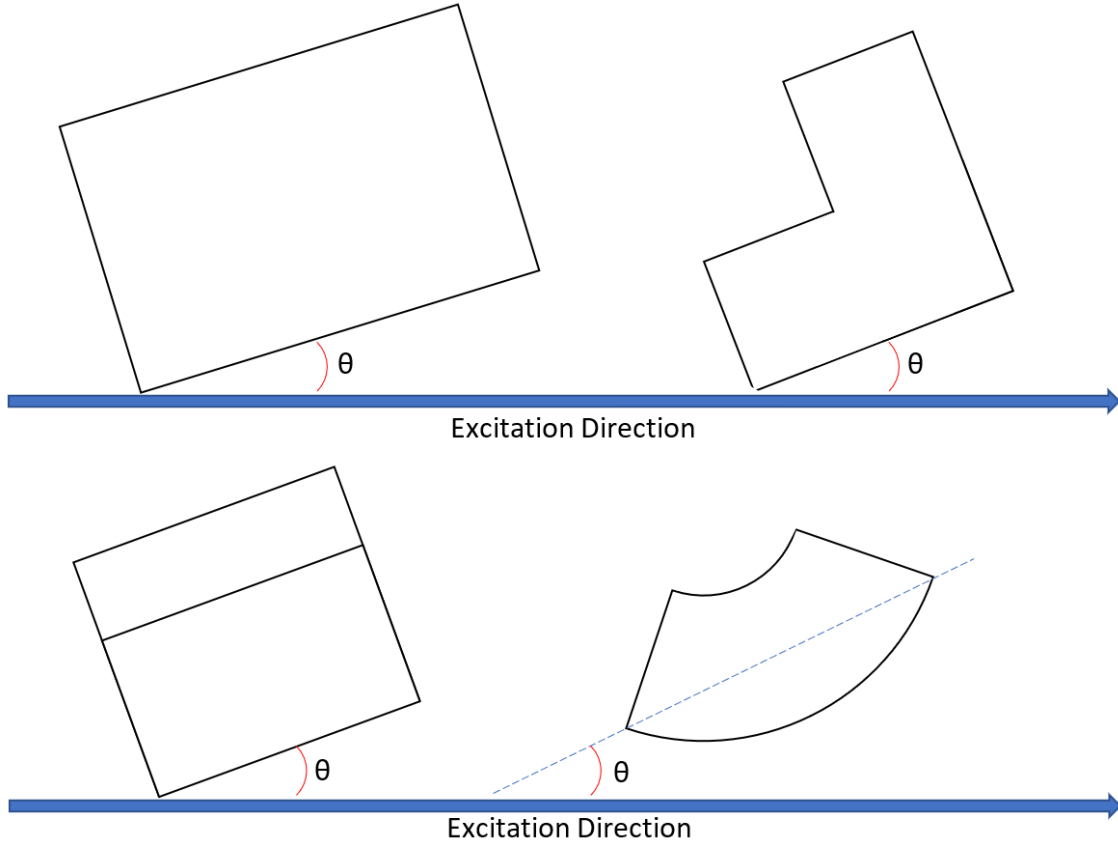
**Figure 2.5.** Experimental setup with optional motor attached.

### 2.1.2 Setup

A translating table was created using a flat metal plate and two sets of ball bearing carriages on guide rails. This allowed the plate to translate smoothly in a single horizontal direction. An image of this setup is given in Figure 2.5.

The container of interest was placed on the translating table at the desired slosh angle,  $\theta$ . In this study, slosh angle is defined as the angle between the direction of excitation and the longest side of the container where applicable, as illustrated in Figure 2.6. This figure also illustrates the slosh angle for other non-uniform geometries. The slosh angle is measured from the edge of the translating table.

Video was recorded using a camera placed on the translating table along with the containers. For the cylinder and quarter annular cylinder, the camera was placed such that the view was perpendicular to the direction of excitation. For the other containers, the camera was placed such that the view was perpendicular to a flat container wall. This allowed for the clearest recording of the surface. The height of the camera was determined so that the camera view was centered on the liquid-gas interface. A light-colored and well-lit back-



**Figure 2.6.** Illustration of slosh angle.

ground was used so that the liquid-gas interface appeared dark against a light background both above and below the interface. This was very important for video processing.

A Nikon D5600 camera was used to record slosh in the basic geometries. This camera records at 60 frames per second with a resolution of 1080 by 1920 pixels. Due to the smaller size of the more complicated geometries, a higher frequency and damping ratio was expected. A GoPro Hero 8 camera with the same resolution but faster frame rate of 240 fps was used for these geometries to appropriately capture slosh behavior. Camera usage is summarized in Table 2.1

The translating table was setup to allow for instigation of slosh by either an impulse or oscillatory motion, provided by an attached motor. Due to some shaking caused by the motor, it was determined that the impulse method provided a cleaner start and better data. The translating table was given a small, single impulse to initiate slosh in the container. Slosh was allowed to decay freely with no other outside inputs.



**Table 2.1.** Camera specifications and usage by tank geometry.

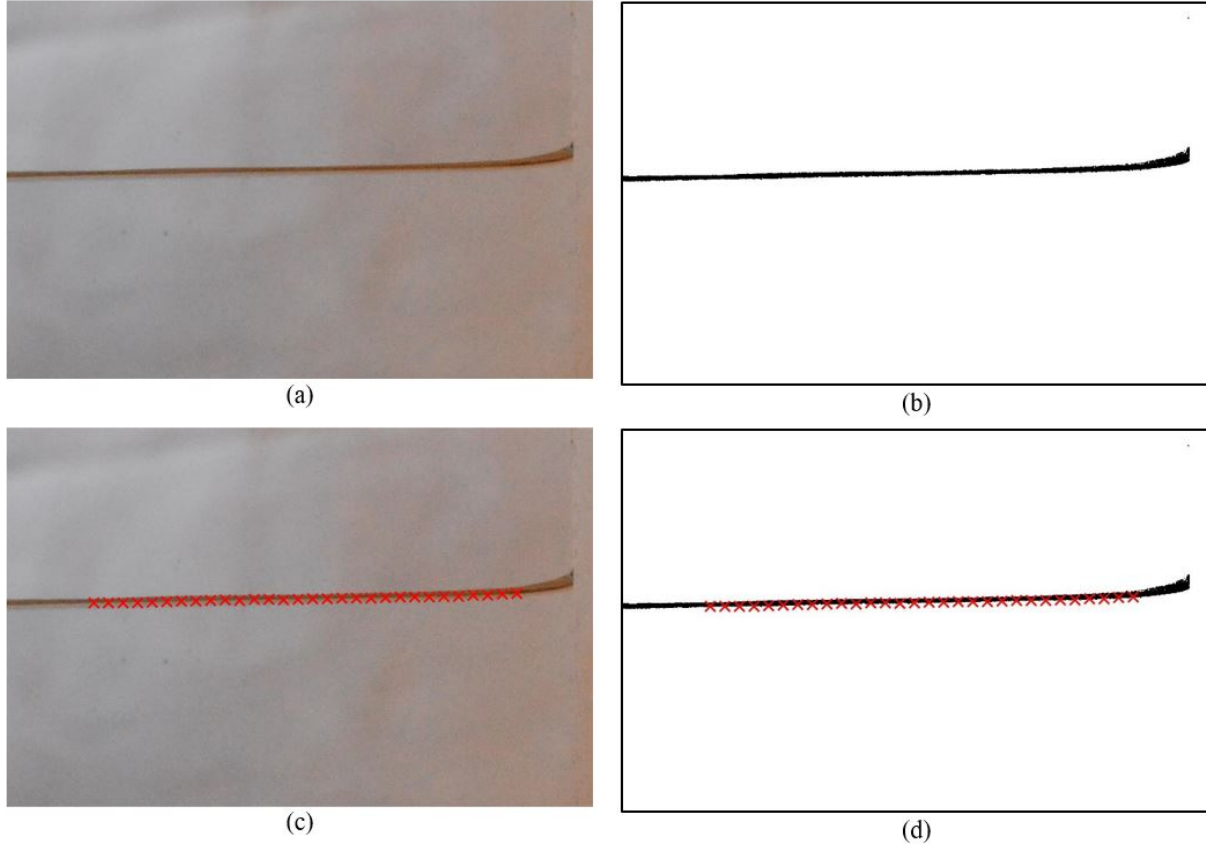
Camera	FPS	Resolution	Geometries
Nikon D5600	60	1080 x 1920	Cylinder
			Cube
			Rectangle
GoPro Hero 8	240	1080 x 1920	Stair Step
			Trapezoid
			Quarter Annular Cylinder

### 2.1.3 Image Processor

The video of the slosh event was processed in MATLAB to obtain the surface height over time in multiple locations. As needed, the video was cropped to the area of interest around the free surface. Because these slosh events are in the linear regime, slosh amplitude is low and the surface did not reach the top and bottom of the frame in many cases. Cropping to the actual slosh area improved processing time because the codes could work with smaller matrices.

By using the described light background and back-lighting, the resulting video of the slosh event shows a dark colored liquid-gas interface with both the area above and below the surface appearing light colored. Each frame of the video is converted to grayscale, then binarized. This creates an image where the liquid and background are solid white (value of 1 in the image matrix), and the liquid-gas interface is solid black (value of 0 in the image matrix). Binarization improves the ability of the code to recognize the surface [35]. The surface height is found by locating the position of the pixel where the color changes from white to black. This is done at many locations along the length of the frame so that multiple calculations of slosh parameters may be done for a single slosh event. These points are plotted over the top of the original video for review to ensure that this process has not picked up any shadows instead of the liquid-gas interface. Figure 2.7 illustrates this process. Surface height is measured from the bottom of the image to the first black pixel because measurement at the bottom of the meniscus was desired.

In Figure 2.7, a curve in the surface near the right side of the image can be seen. This is where the liquid has climbed the corner of the cube. Surface height measurements were



**Figure 2.7.** Comparison of slosh video frames before and after processing. (a) Original frame. (b) Frame after binarization. (c) Original frame image with surface points plotted. (d) Binarized frame with surface points plotted.

not taken in this region. In some frames where the liquid-gas interface is near its lowest height, the interface appears so thin that the video and image processor are unable to pick up the location. In these frames, the code records a value of 1 for the pixel height of the surface. This value is significantly lower than the values recorded for good data, so it is easy to process the data further to remove only the points with errors.

Before calculating damping ratio and natural frequency, some data is cut off from the beginning and end of the recording. At the beginning, there is some data before the impulse which must be removed. After the impulse, a few seconds of data are also removed so start-up behavior is not used in calculations. For good slosh parameter calculation, only the clean, free-decay behavior is desired. Data at the end of the slosh event is cut off when the surface

amplitude is so small that it is only changing by a few pixels. This is clearly seen in the data when it begins to have the appearance of a step function instead of a smooth sinusoid.

#### 2.1.4 Parameter Calculation

There are several methods which may be used to calculate the damping ratio and natural frequency of the slosh from the data for surface height over time.

#### Damped Sinusoid

The entire set of data may be curve fit with a damped sinusoid to obtain both the natural frequency and damping ratio. Depending on the data set, either a single damped sinusoid of the form given in Equation 2.1 or the sum of two damped sinusoids of the form given in Equation 2.2 can be used.

$$f(x) = A \sin(\omega_d t + \phi) \exp(bt) + C \quad (2.1)$$

$$f(x) = A_1 \sin(\omega_1 t + \phi_1) \exp(b_1 t) + A_2 \sin(\omega_2 t + \phi_2) \exp(b_2 t) + C \quad (2.2)$$

These equations give the damped natural frequency and the damping constant. Because the damping ratio is extremely small, the natural frequency is approximately the same as the damped natural frequency and the damping ratio may be calculated using Equation 2.3. An explanation for the derivation of the best fit equation and the approximation is given in Appendix C.

$$\zeta = \frac{b}{\omega_n} \quad (2.3)$$

The MATLAB Curve Fitting Toolbox was used to fit these equations to the data. In most cases, restricting the damping ratio to a negative value (decay) as well as an initial guess for frequency within about 1 rad/s of the expected value was sufficient to get a good curve fit. All other parameters could be nearly any arbitrary value and still converge to the

same curve fit. This is the method used for calculating both the damping ratio and frequency for experimental data.

## Peak Curve Fit

Another option for calculating the damping ratio and natural frequency is by using only the peaks in the data. This can be done by either using just the local maximums, or by centering the data about the x-axis, then finding the local maximums of the magnitude of the data. In the latter case, both the original peaks and valleys in the data are analyzed and the number of data points is doubled. This is the method used in this research.

The maximums are found using the built-in MATLAB findpeaks function, which gives as output both the peak values and the locations of the peak values. The natural frequency can be found using the difference in time between neighboring peaks. The natural frequency is given by Equation 2.4 when only the original peaks are used (full period) and by Equation 2.5 when both the original peaks and valleys are used (half period). In both equations,  $t$  is the time at which a maximum occurs.

$$\omega_{n_{fullperiod}} = \frac{2\pi}{t_{i+1} - t_i} \quad (2.4)$$

$$\omega_{n_{halfperiod}} = \frac{\pi}{t_{i+1} - t_i} \quad (2.5)$$

The damping ratio may be found by using the MATLAB Curve Fitting Toolbox to fit an equation of the form given in Equation 2.6. This equation is the same for both methods discussed here. Again, this curve fit gives the damping constant, and the damping ratio may be calculated by Equation 2.3.

$$f(x) = A \exp(bt) \quad (2.6)$$

## Logarithmic Decrement

The final method for calculating the damping ratio is by calculating the logarithmic decrement between peaks. Peaks are found in the same way as described above. The damping ratio is given by Equation 2.7 when only the original peaks are used (full period) and by Equation 2.8 when both the original peaks and valleys are used (half period). In both equations,  $A$  is the amplitude of each successive maximum.

$$\zeta_{fullperiod} = \frac{1}{2\pi} \log \left( \frac{A_i}{A_{i+1}} \right) \quad (2.7)$$

$$\zeta_{halfperiod} = \frac{1}{\pi} \log \left( \frac{A_i}{A_{i+1}} \right) \quad (2.8)$$

### 2.1.5 Uncertainty

The uncertainty in each slosh parameter was calculated. Both the damping ratio and frequency had three similar sources of uncertainty. The first was uncertainty between parameter values calculated at various locations along the surface for a single slosh event. The maximum standard deviation between these parameters was used. The second is uncertainty between parameter values across different slosh events. The standard deviation between the average parameter for each slosh event was used. Finally, there is uncertainty in video processing. Random scatter was applied to 25% of the data points for a slosh event, then slosh parameters recalculated. The difference between the original and artificially scattered data is taken to be the uncertainty. This is considered a very conservative estimate because the scattered data is significantly poorer quality than any data used for actual parameter calculations.

In addition to these three sources of uncertainty, the frequency also depends on the frame rate of the camera. For a conservative estimate, it was assumed that there was one less frame per ten seconds of recorded data. For example, this would mean that the frame rate would be 59.9 fps instead of 60 fps. The difference in the calculated frequency is the uncertainty due to this possible timing error.

The root sum square of each source of uncertainty was used to calculate the total uncertainty for each slosh parameter [35].

## 2.2 Results

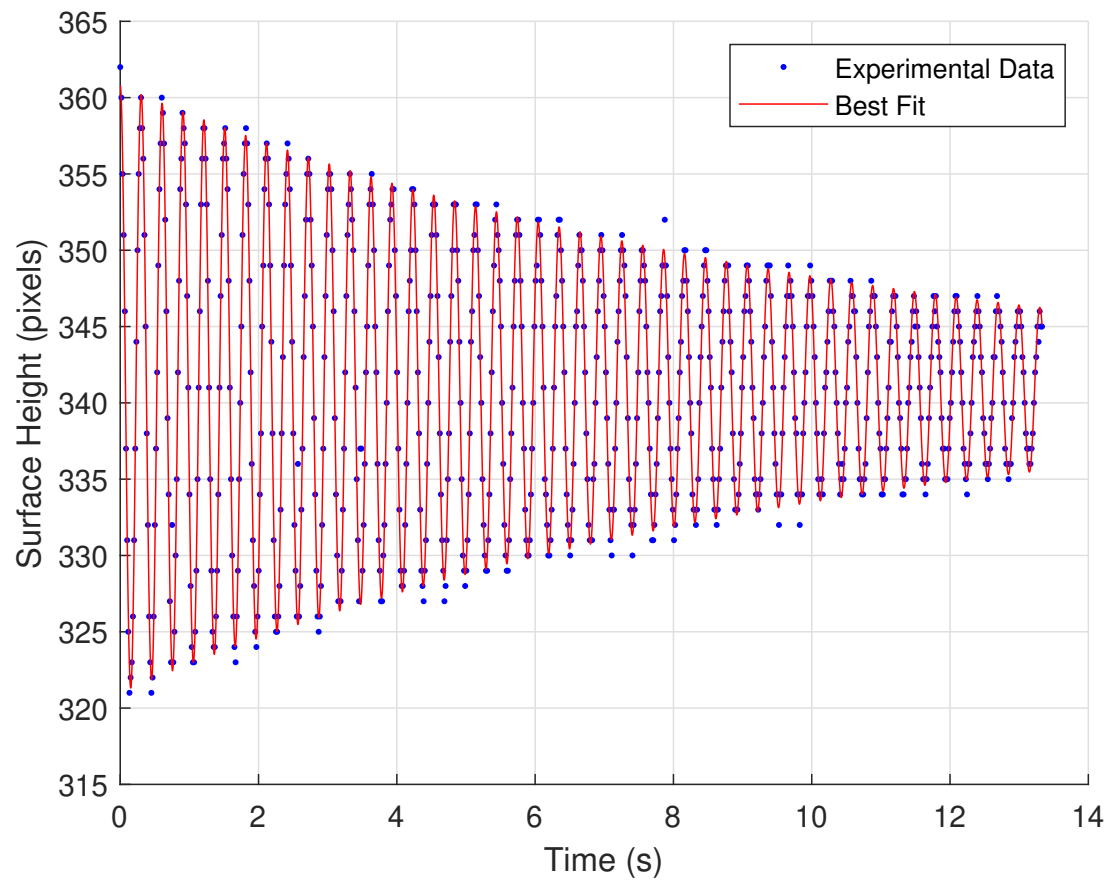
### 2.2.1 Cylinder

Because a larger body of previous work was available for cylindrical containers, the cylinder was the first container investigated. This comparison helped raise the confidence in the experimental methods used. Figure 2.8 shows an example of the surface height over time as well as a best fit equation. As expected, the surface height over time is in the shape of a damped sinusoid with only the first natural frequency clearly visible. The best fit equation was of the form given in Equation 2.1. The damping constant was restricted to negative values because slosh should damp out with no excitation other than the initial impulse. The analytical solution for natural frequency in a cylinder of this size was used as the initial guess of the natural frequency in the best fit equation. Parameters do not vary with time.

Table 2.2 summarizes the data obtained for the cylinder. The experimental natural frequency was within 7% of the analytical solution for natural frequency. The experimental damping ratio was between 12-36% of the existing damping ratio correlations for cylinders given in Equations 1.6 and 1.7.

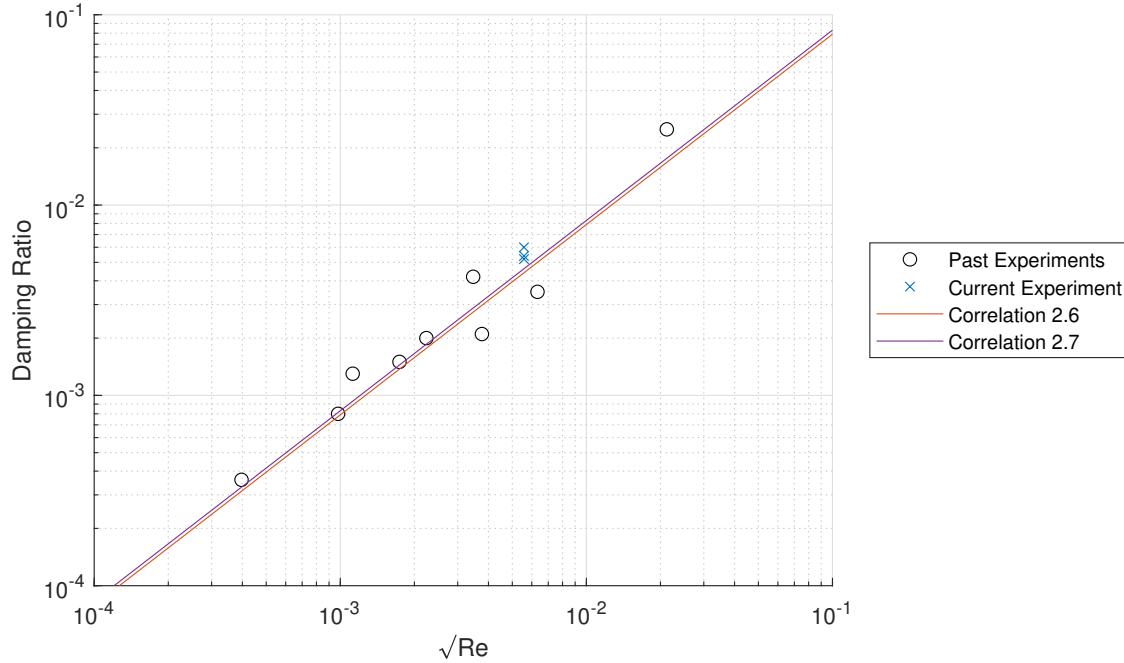
**Table 2.2.** Summary of Results for Cylindrical Tank

	<b>Height 106 mm (1.12D)</b>	<b>Height 120.8 mm (1.28D)</b>	<b>Height 141.6 mm (1.5D)</b>
<b>Natural Frequency (<math>\omega_n</math>)</b>	-	-	-
Experimental Average (rad/s)	20.78	20.87	20.91
Analytical (rad/s)	19.55	19.55	19.55
Difference (%)	6.31	6.75	6.96
<b>Damping Ratio (<math>\zeta</math>)</b>	-	-	-
Experimental Average (%)	0.52	0.54	0.60
Equation 1.6 [5] (%)	0.44	0.44	0.44
Difference from 1.6 (%)	16.9	21.9	35.9
Equation 1.7 [5] (%)	0.46	0.46	0.46
Difference from 1.7 (%)	12.6	16.8	29.7



**Figure 2.8.** Example of data obtained for a slosh event in a cylindrical container as well as the best fit damped sinusoid.

While this may seem like a large difference between the experimental and empirical damping ratio, this is reasonably in line with previous experimental data for cylindrical containers. Figure 2.9 is replicated from Yang et. al. with the addition of the data from this experiment [6]. The experimental data for this cylinder has similar scatter from the empirical correlations as past experimental data.

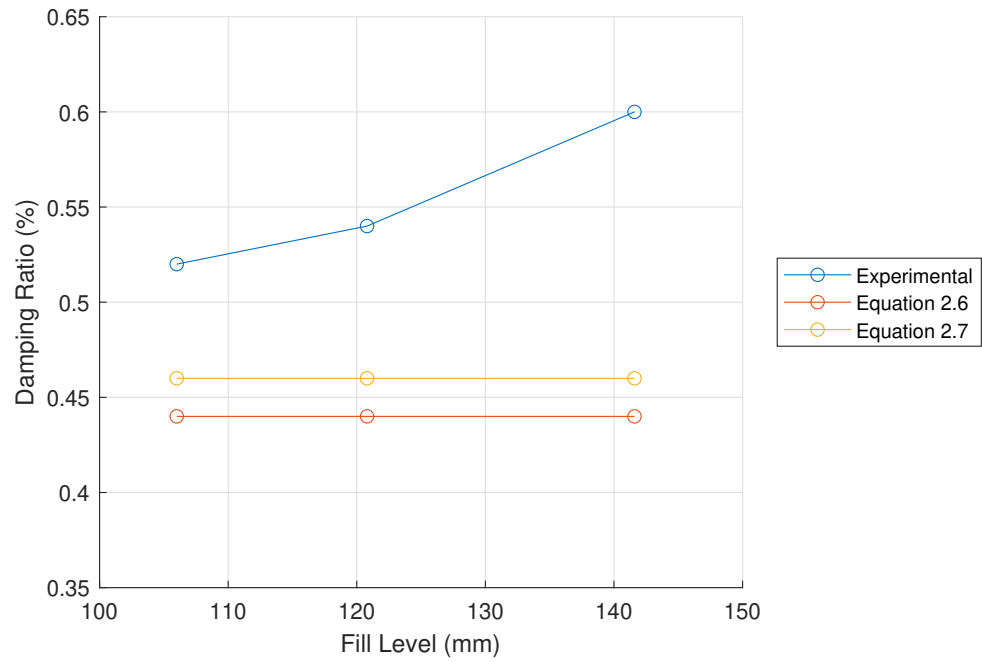


**Figure 2.9.** Comparison of three fill fraction damping ratios to past experimental data as well as empirical correlations for a cylindrical tank [6]

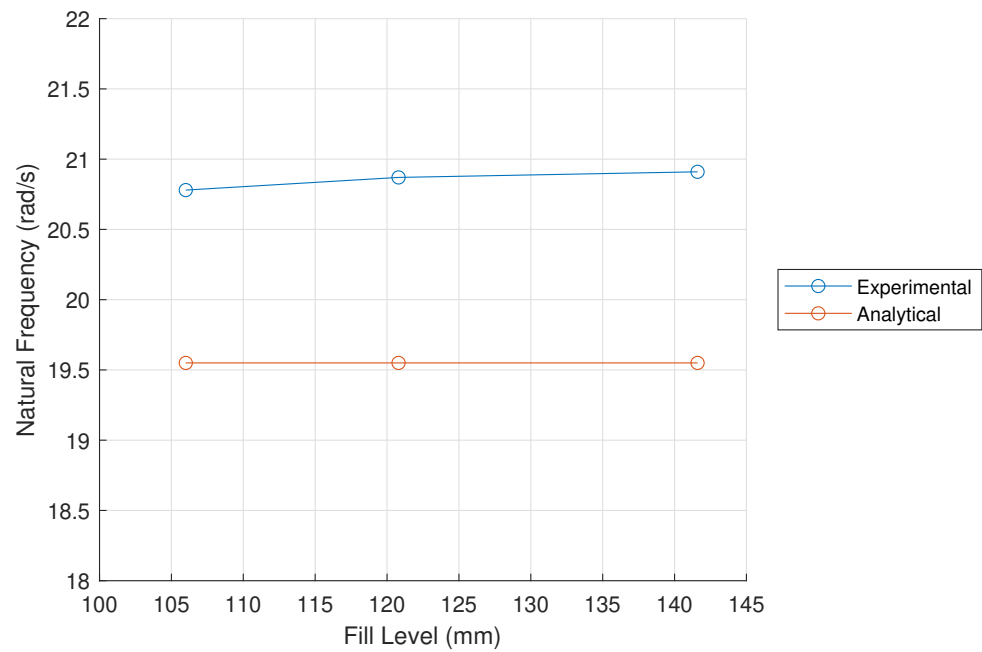
Figure 2.10 shows a comparison between the experimental data and analytical solutions for damping ratio at the tested fill levels. Because the tested fill levels were above twice the radius, it is expected that the damping ratio would remain constant. However, the damping ratio increased slightly as the fill level increased. While unexpected, the damping ratio is small enough in magnitude that it does not seem completely out of line.

Figure 2.11 shows a comparison of the experimental data and analytical solutions for natural frequency at the test fill levels. Again, the natural frequency is expected to remain constant at these fill levels. The experimental natural frequency increased only slightly with fill level.





**Figure 2.10.** Comparison between experimental and analytical solutions for damping ratio in the cylindrical tank.



**Figure 2.11.** Comparison between experimental and analytical solutions for natural frequency in the cylindrical tank.

Uncertainty in the presented slosh parameters is given in Table 2.3. Plots showing error bars are given with the CFD results to better show agreement with CFD results.

**Table 2.3.** Standard uncertainty in experimental values for the cylindrical container.

Fill Level	1.12D	1.28D	1.5D
Uncertainty in $\zeta$ (%)	0.115	0.067	0.070
Uncertainty in $\omega$ (rad/s)	0.039	0.039	0.047

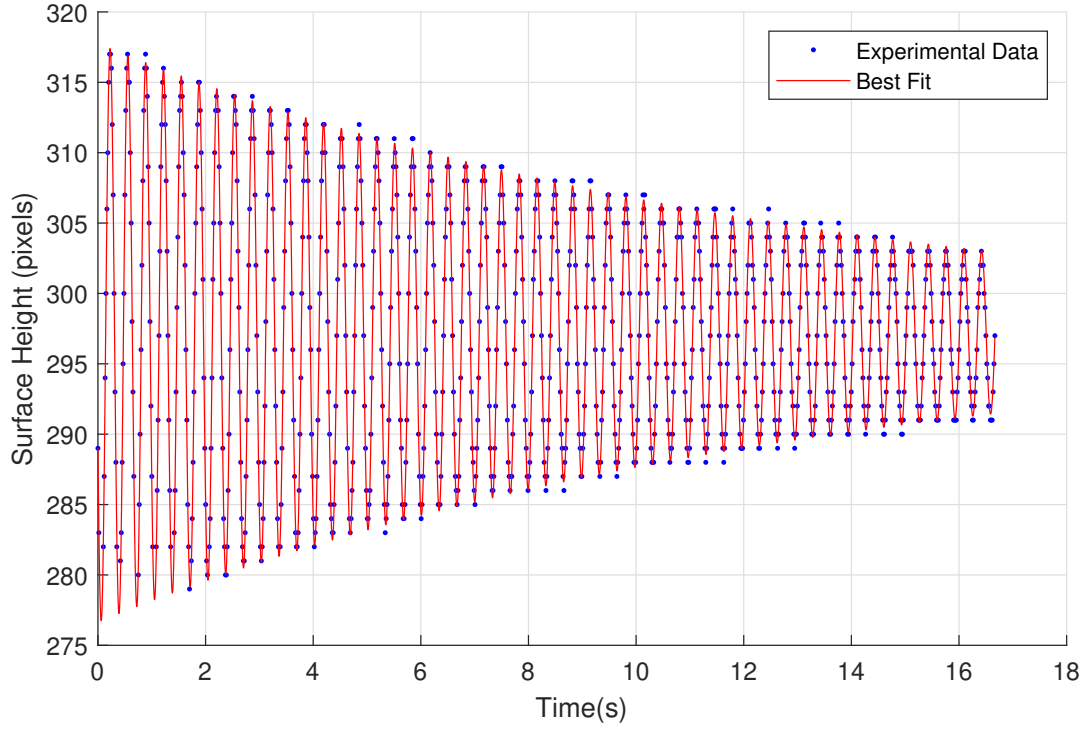
### 2.2.2 Cube

The cube tank was analyzed at a 0, 30, 45, and 60-degree slosh angle. The 90-degree slosh angle was not investigated because it would be a repeat of the 0-degree case due to equal side length. Figure 2.12 gives an example of the data obtained for a cube slosh event as well as the best fit damped sinusoid. The best fit equation was of the form given in Equation 2.1. The damping constant was restricted to negative values because slosh should damp out with no excitation other than the initial impulse. The analytical solution for natural frequency in a rectangular prism of this size was used as the initial guess of the natural frequency in the best fit equation. Parameters do not vary with time. Unexpectedly, the data for slosh at asymmetric angles also appear to be composed of a single damped sinusoid.

A summary of the experimental data is given in Table 2.4. As shown, the natural frequency is nearly the same at all slosh angles. For the zero-degree slosh angle, an analytical solution for the natural frequency exists as given in Equation 1.1. At each fill level, there was between a 6.39% and 7.57% difference between the analytical solution and the experimental result at a zero-degree slosh angle.

Figure 2.13 shows a comparison between the experimental data and analytical solutions for damping ratio at the tested fill levels. With the exception of slosh at zero degrees, the damping ratio appears to increase with increasing fill level.

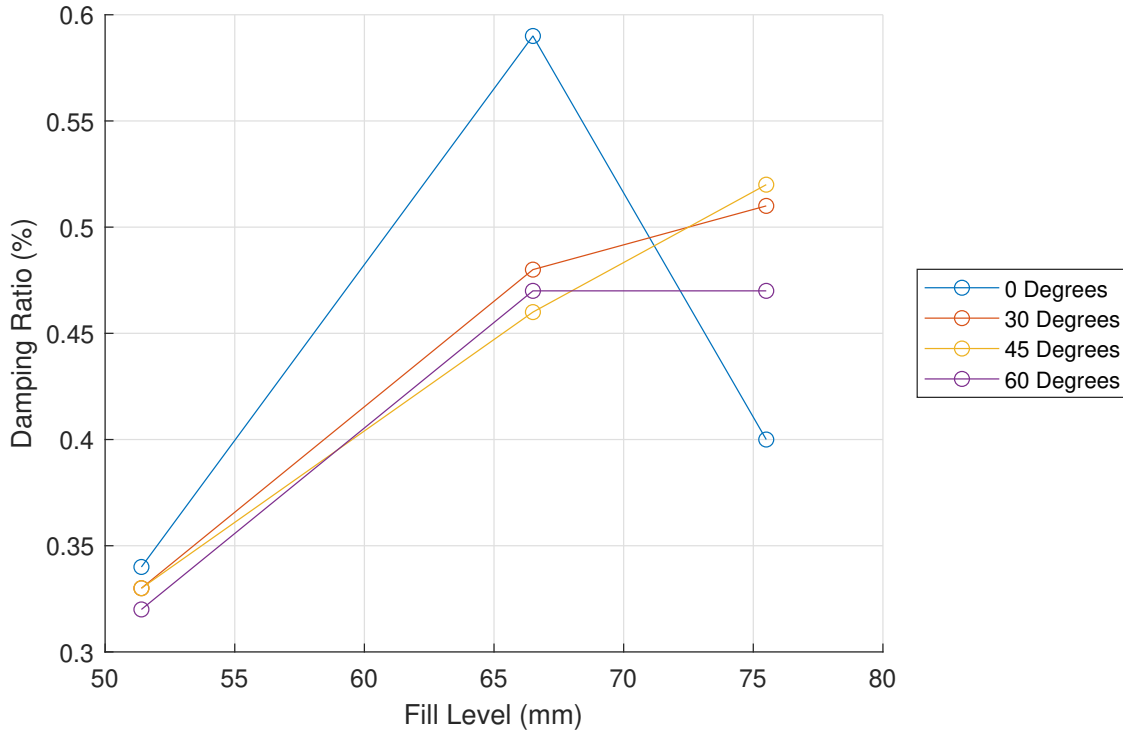
Figure 2.14 shows a comparison of the experimental data and analytical solutions for natural frequency at the test fill levels. The natural frequency of the slosh slightly increases at the fill level increases, which is expected based on the behavior of the analytical solution. It appears that the experimental results are simply shifted upward from the analytical solution.



**Figure 2.12.** Example of data obtained for a slosh event in the cube container as well as the best fit damped sinusoid. This data is for a zero-degree slosh angle.

**Table 2.4.** Summary of results for the cubic tank.

	Surface Height of 51.4 mm		Surface Height 66.5 mm		Surface Height 75.5 mm	
Slosh Angle	Average $\omega_n$ (rad/s)	Average $\zeta$ (%)	Average $\omega_n$ (rad/s)	Average $\zeta$ (%)	Average $\omega_n$ (rad/s)	Average $\zeta$ (%)
0	18.61	0.34	18.81	0.59	19.01	0.40
30	18.60	0.33	18.92	0.48	18.96	0.51
45	18.59	0.33	18.91	0.46	18.98	0.52
60	18.58	0.32	18.96	0.47	19.02	0.47
Analytical	17.30	-	17.68	-	17.78	-
% Diff	7.57	-	6.39	-	6.92	



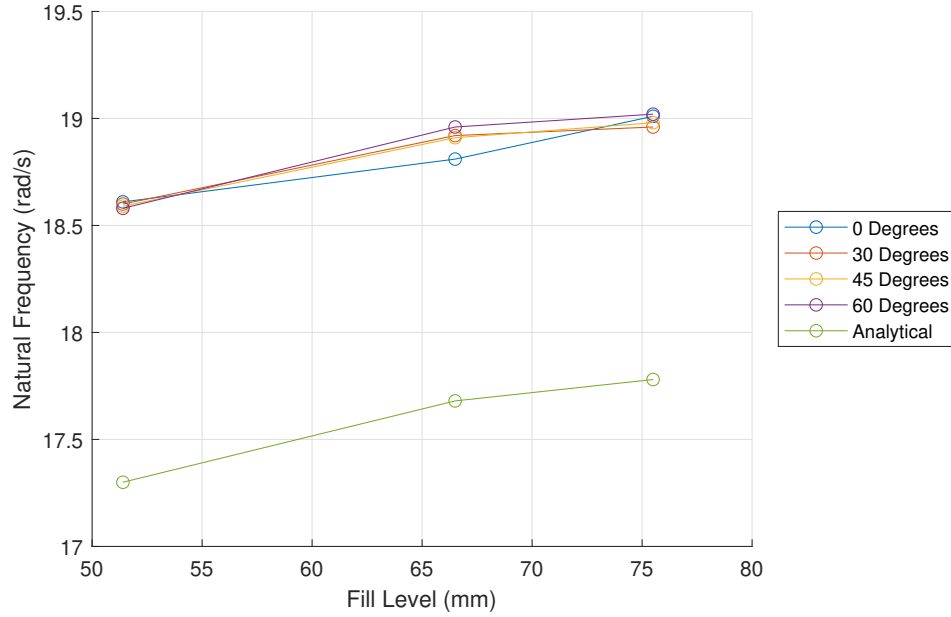
**Figure 2.13.** Damping ratio versus fill level for various slosh angles in the cube tank.

When reviewing top-down video of cube slosh, it appears as if the slosh quickly aligns itself with the walls. It is possible that there are actually two waves travelling between each pair of walls. Because these two waves would have the same frequency, the single sinusoid seen in the data could actually be two overlapping sinusoids which appear as one.

Uncertainty in the presented slosh parameters is given in Table 2.5. Plots showing error bars are given with the CFD results to better show agreement with CFD results.

**Table 2.5.** Standard uncertainty in experimental values for the cube tank.

Fill Level	Slosh Angle	0	30	45	60
51.4 mm	Uncertainty in $\zeta$ (%)	0.042	0.033	0.032	0.033
	Uncertainty in $\omega$ (rad/s)	0.033	0.033	0.033	0.033
66.5 mm	Uncertainty in $\zeta$ (%)	0.058	0.035	0.033	0.078
	Uncertainty in $\omega$ (rad/s)	0.033	0.034	0.033	0.034
75.5 mm	Uncertainty in $\zeta$ (%)	0.033	0.048	0.078	0.034
	Uncertainty in $\omega$ (rad/s)	0.035	0.035	0.034	0.034

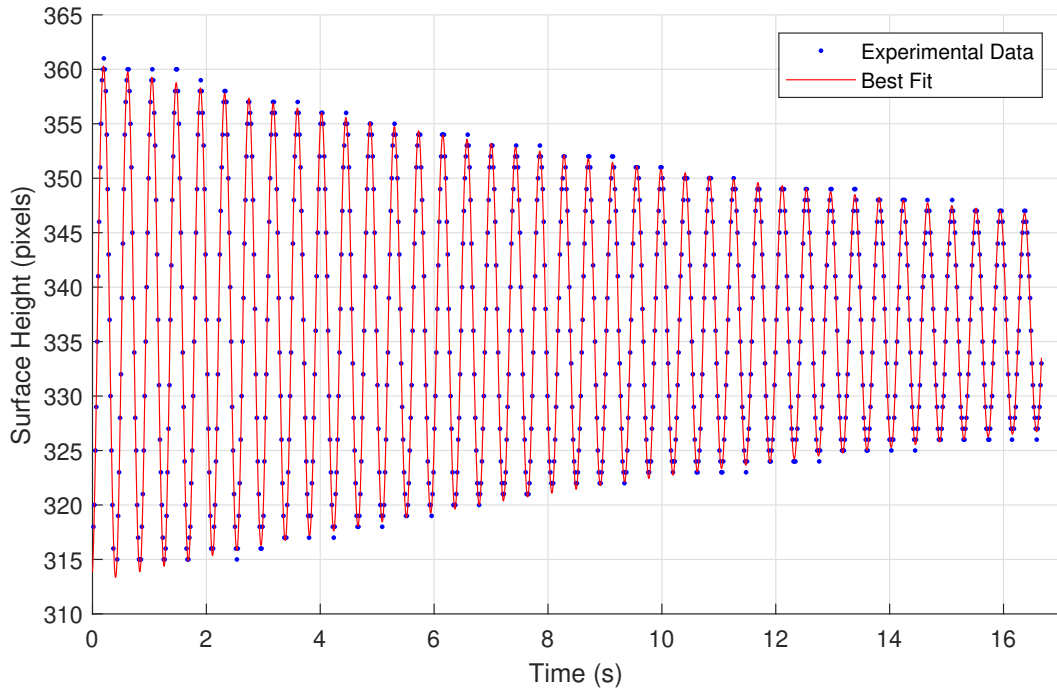


**Figure 2.14.** Comparison between experimental and analytical solutions for frequency in the cubic tank.

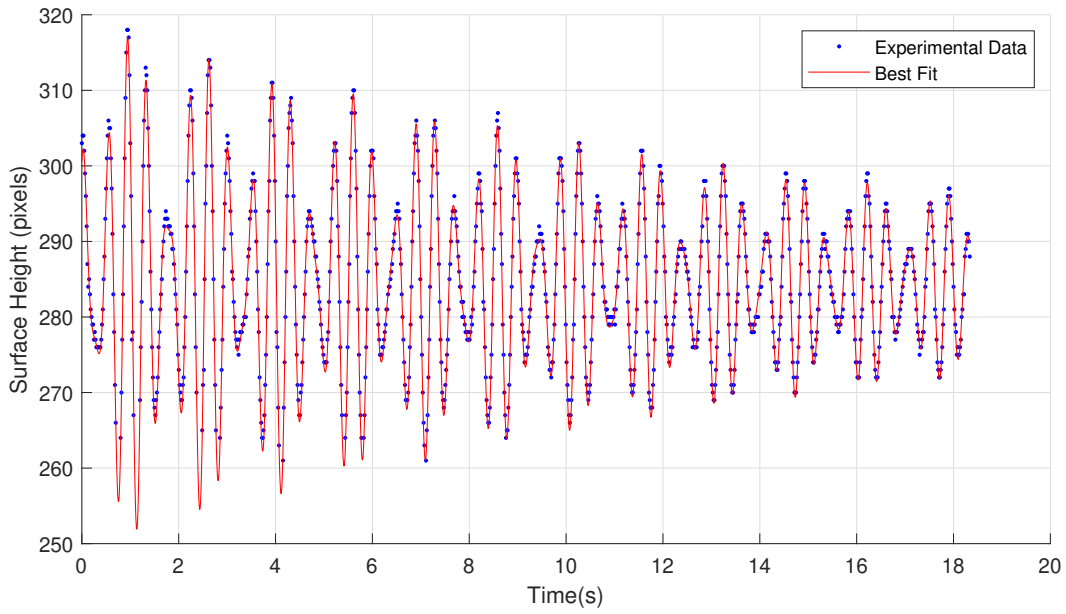
### 2.2.3 Rectangular Prism

The rectangular prism container was analyzed at a 0, 30, 45, 60, and 90-degree slosh angle. Figure 2.15 shows an example of surface height over time for the rectangular prism at a zero-degree slosh angle. This has the appearance of a single damped sinusoid as expected and can be modeled with Equation 2.1. All cases at either a zero- or ninety-degree angle exhibited similar behavior. Figure 2.16 is an example of slosh at a sixty-degree slosh angle. The behavior appears to be at least two overlapping damped sinusoids and can be modeled using Equation 2.2. The same general behavior was observed at 30, 45, and 60-degree slosh angles. The damping constant was restricted to values less than one because slosh should decay as there are no inputs after the initial excitation. The frequency of slosh for the 0- and 90-degree slosh cases were used as initial guess values for each frequency in the curve fit but were not constrained to be near these values. Parameters do not vary with time.

A summary of the experimental data is given in Table 2.6. Figure 2.17 illustrates the camera views and slosh modes referenced in Table 2.6. For the zero- and ninety-degree cases, an analytical solution for the natural frequency exists [5]. There is a 5.13% and 6.28%



**Figure 2.15.** Example of data obtained for a zero-degree slosh event in the rectangular prism container as well as the best fit damped sinusoid.



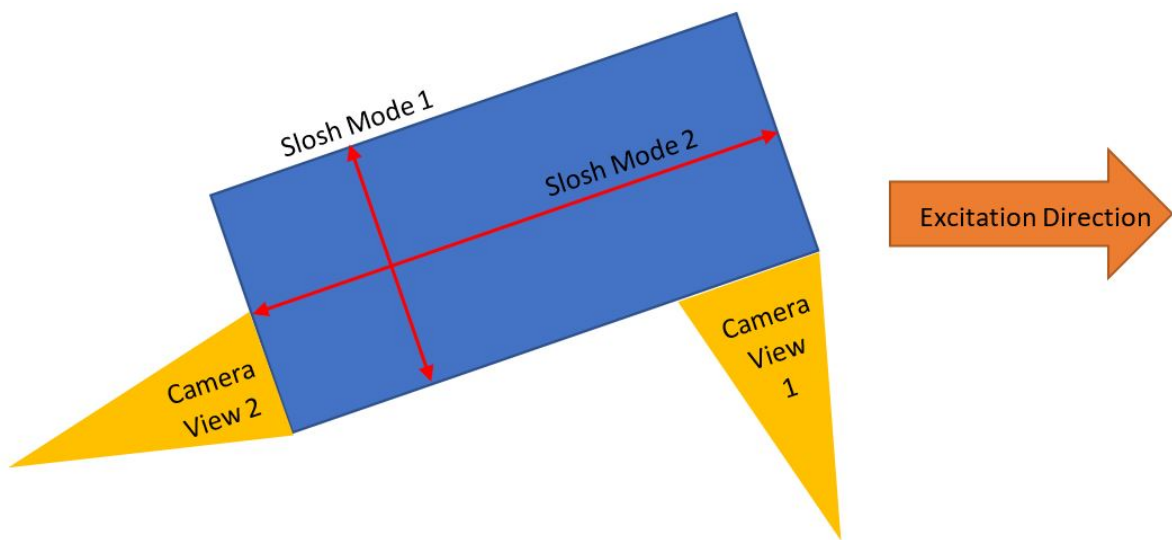
**Figure 2.16.** Example of data obtained for a sixty-degree slosh event in the rectangular prism container as well as the best fit sum of two damped sinusoids.

difference between the experimental and the analytical solution for zero- and ninety- degree slosh angles respectively. At any slosh angle tested, one mode had nearly the same values for natural frequency and damping ratio as the single sinusoid at a zero-degree slosh angle and the other mode had nearly the same values as slosh at ninety degrees. This seems to indicate that slosh at any angle can be decomposed into two waves, each corresponding to slosh between one pair of walls.

**Table 2.6.** Results for a rectangular tank.

Surface Height of 59.8 mm						
Angle	Camera View	Mode 1 $\omega_n$ (rad/s)	Mode 2 $\omega_n$ (rad/s)	Mode 1 $\zeta$ (%)	Mode 2 $\zeta$ (%)	$A_1/A_2$
0	1	-	14.16	-	0.38	-
30	1	18.7	14.17	0.37	0.42	0.18
45	1	18.7	14.17	0.35	0.40	0.26
60	2	18.7	14.17	0.34	0.38	0.32
90	2	18.7	-	0.35	-	-
Analytical	-	17.57	13.41	-	-	-
Surface Height of 70.2 mm						
Angle	Camera View	Mode 1 $\omega_n$ (rad/s)	Mode 2 $\omega_n$ (rad/s)	Mode 1 $\zeta$ (%)	Mode 2 $\zeta$ (%)	$A_1/A_2$
0	1	-	14.51	-	0.42	-
30	1	18.81	14.50	0.58	0.45	0.14
45	1	18.81	14.51	0.55	0.44	0.26
60	2	18.80	14.51	0.53	0.43	0.41
90	2	18.83	-	0.47	-	-
Analytical	-	17.75	13.79	-	-	-
Surface Height of 80.8 mm						
Angle	Camera View	Mode 1 $\omega_n$ (rad/s)	Mode 2 $\omega_n$ (rad/s)	Mode 1 $\zeta$ (%)	Mode 2 $\zeta$ (%)	$A_1/A_2$
0	1	-	14.76	-	0.33	-
30	1	18.93	14.80	0.33	0.33	0.22
45	1	18.94	14.80	0.28	0.29	0.31
60	1	18.93	14.80	0.33	0.34	0.67
60	2	18.95	14.81	0.30	0.31	0.60
90	2	18.95	-	0.31	-	-
Analytical	-	17.84	14.04	-	-	-

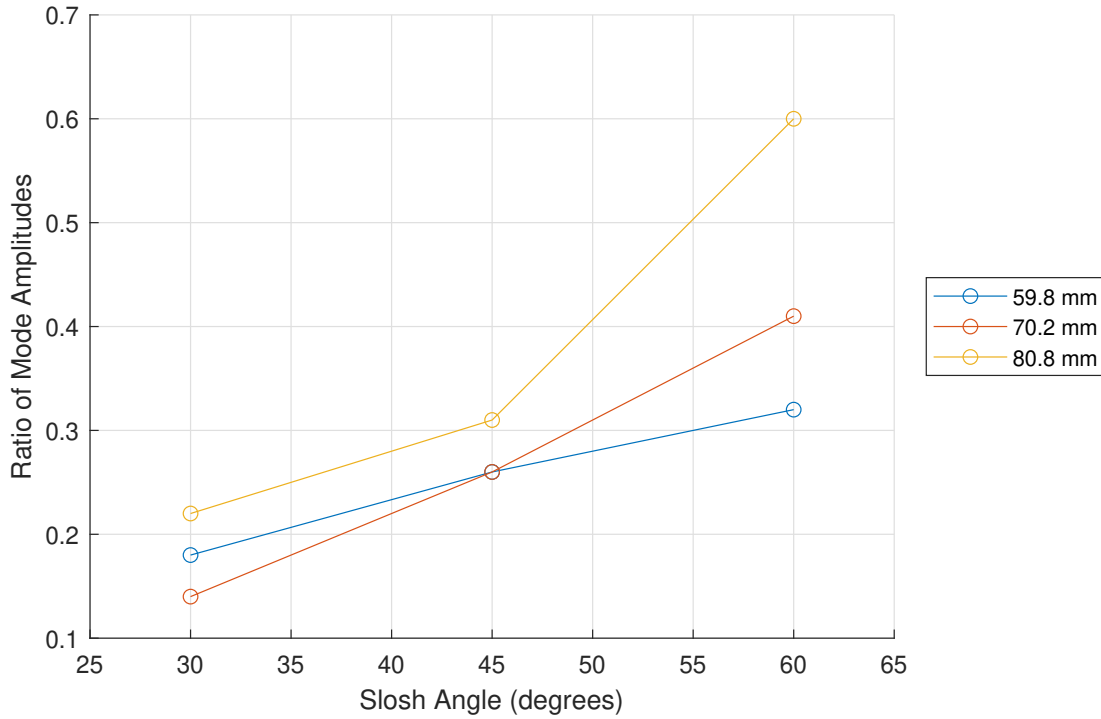
The ratio of amplitudes between the two modes was also calculated for each slosh angle. There was significantly more scatter in this data than in natural frequency and damping



**Figure 2.17.** Illustration of slosh modes and camera views.



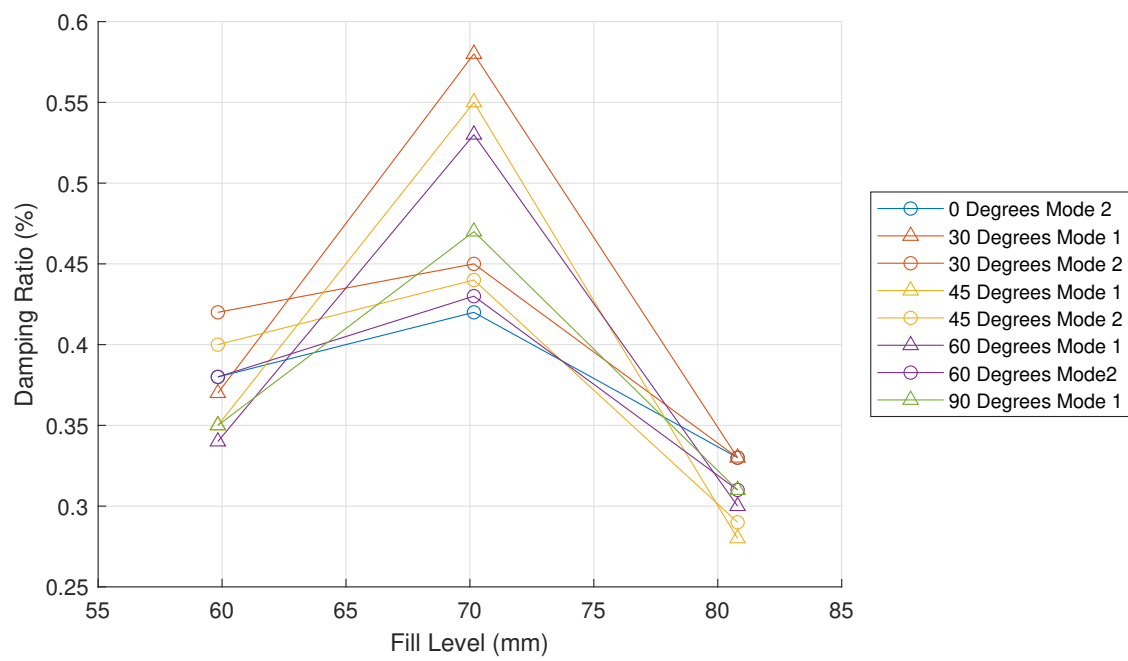
ratio. In general, however, higher slosh angles resulted in a larger ratio of mode 1 amplitude to mode 2 amplitude as illustrated in Figure 2.18. This is reasonable, because at higher slosh angles, the component of the long wall normal to the excitation direction is larger than the component of short wall normal to the excitation direction.



**Figure 2.18.** Ratio of mode amplitudes over several slosh angles.

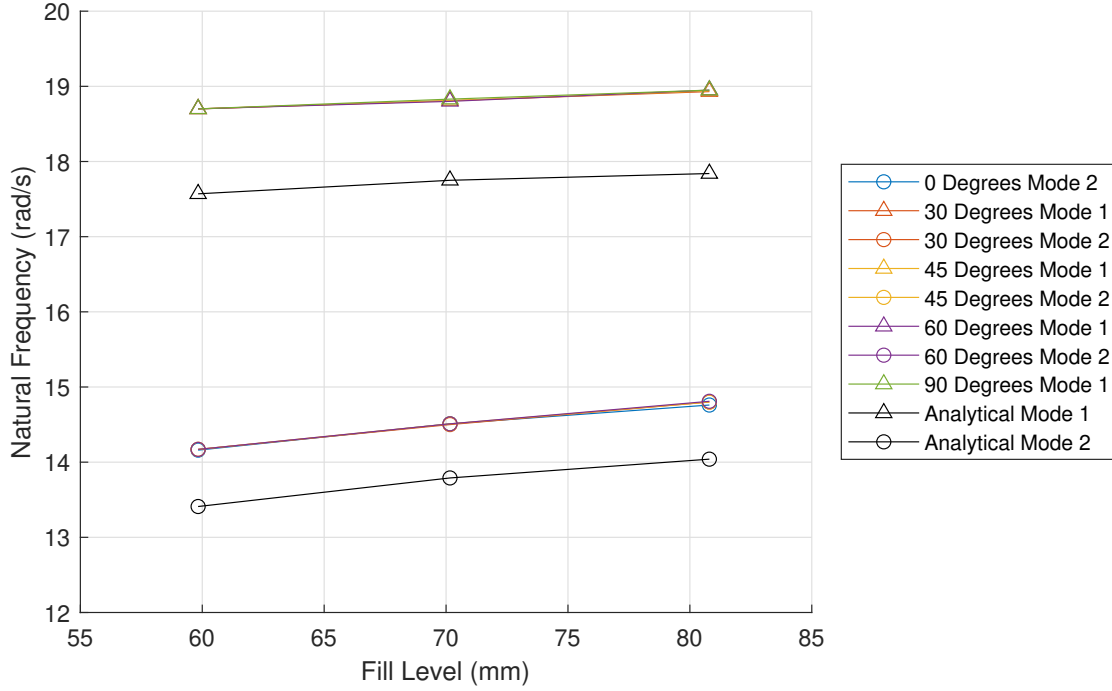
The damping ratio for both modes increased slightly from the lowest to the mid-fill fraction, then decreased in the highest fill fraction. This is shown in Figure 2.19. While the two modes have similar damping ratios, mode one has a higher damping constant due to its higher frequency (see Equation 2.3). Because of this, the slosh in that direction decays more quickly. Slosh toward the end of the recording period appears more like a single sinusoid with the same frequency as mode two. This is expected. It is logical that the longer tank dimension (mode two) would have a lower damping constant, and therefore slosh will align with that direction toward the end of the slosh event.

The natural frequency slightly increased as the fill level increased as shown in Figure 2.20. The value for each mode's natural frequency was nearly the same at each slosh angle. The



**Figure 2.19.** Damping ratio versus fill level for both modes and various slosh angles in the rectangular tank.

experimental natural frequency exhibited the same behavior as the analytical solution but was shifted up by a near-constant amount. This is typical of all container geometries.

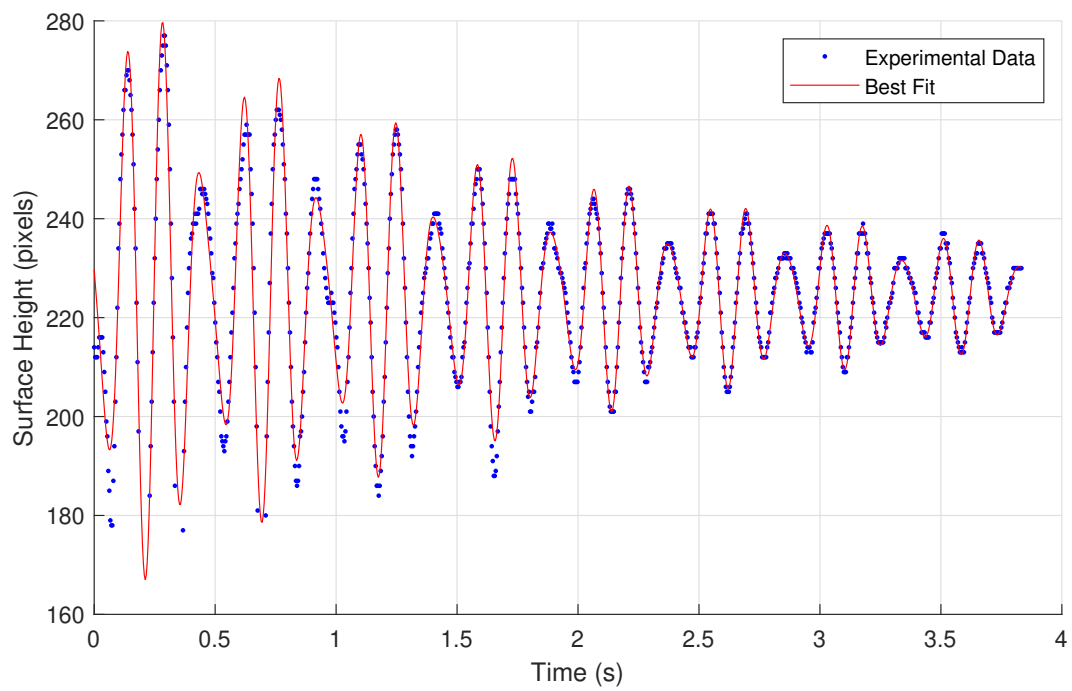


**Figure 2.20.** Natural frequency versus fill level for both modes and various slosh angles in the rectangular tank.

Uncertainty in the presented slosh parameters is given in Table 2.7. Plots showing error bars are given with the CFD results to better show agreement with CFD results.

#### 2.2.4 Stair Step

The stair step tank was analyzed at a 0, 30, 45, 60, and 90-degree slosh angle. Figure 2.21 shows an example of surface height over time data obtained for this geometry. At all slosh angles, the data has the appearance of the sum of at least two damped sinusoids and can be modeled with Equation 2.2. The damping constant was again restricted to negative values. Parameters do not vary in time.



**Figure 2.21.** Example of data obtained for a slosh event in the stair step tank as well as the best fit sum of two damped sinusoids.

**Table 2.7.** Standard uncertainty in experimental values for the rectangular tank.

Fill Level	Mode	Slosh Angle	0	30	45	60	90
59.8 mm	1	Uncertainty in $\zeta$ (%)	-	0.101	0.092	0.092	0.094
		Uncertainty in $\omega$ (rad/s)	-	0.034	0.033	0.033	0.033
	2	Uncertainty in $\zeta$ (%)	0.093	0.092	0.092	0.092	-
		Uncertainty in $\omega$ (rad/s)	0.026	0.026	0.026	0.026	-
70.2 mm	1	Uncertainty in $\zeta$ (%)	-	0.035	0.040	0.069	0.023
		Uncertainty in $\omega$ (rad/s)	-	0.034	0.034	0.034	0.033
	2	Uncertainty in $\zeta$ (%)	0.028	0.029	0.033	0.069	-
		Uncertainty in $\omega$ (rad/s)	0.027	0.026	0.027	0.027	-
80.8 mm	1	Uncertainty in $\zeta$ (%)	-	0.072	0.053	0.026	0.025
		Uncertainty in $\omega$ (rad/s)	-	0.044	0.044	0.045	0.044
	2	Uncertainty in $\zeta$ (%)	0.027	0.026	0.032	0.024	-
		Uncertainty in $\omega$ (rad/s)	0.039	0.039	0.039	0.039	-

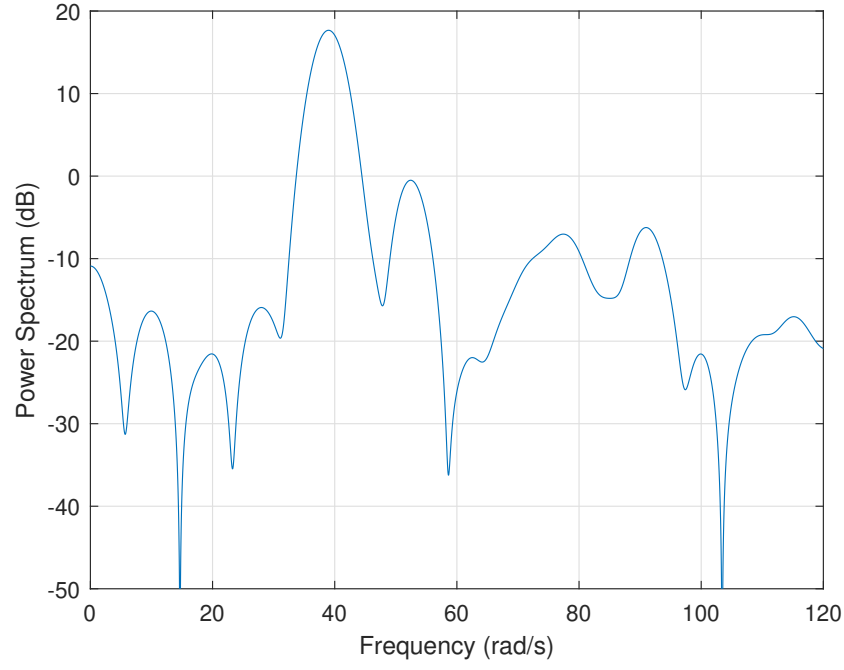
A summary of the experimental data is given in Table 2.8. Both the damping ratio and natural frequency are relatively consistent across all slosh angles, although the frequency was more so. These modes will be discussed further in the CFD section.

**Table 2.8.** Experimental results for the stair step tank.

	Mode 1		Mode 2	
Slosh Angle	$\omega_n$ (rad/s)	$\zeta$ (%)	$\omega_n$ (rad/s)	$\zeta$ (%)
0	39.2	1.29	52.3	1.60
30	39.1	1.27	52.3	1.11
45	39.0	1.55	52.2	1.36
60	39.0	1.30	52.2	1.11
90	39.0	1.02	52.3	1.04

The two frequencies presented in this table are generally the most prominent when looking at the power spectrum of the data, such as in Figure 2.22. In some cases, however, there is also a prominent peak around 63 rad/s.

Uncertainty in the presented slosh parameters is given in Table 2.9. Plots showing error bars are given with the CFD results to better show agreement with CFD results.



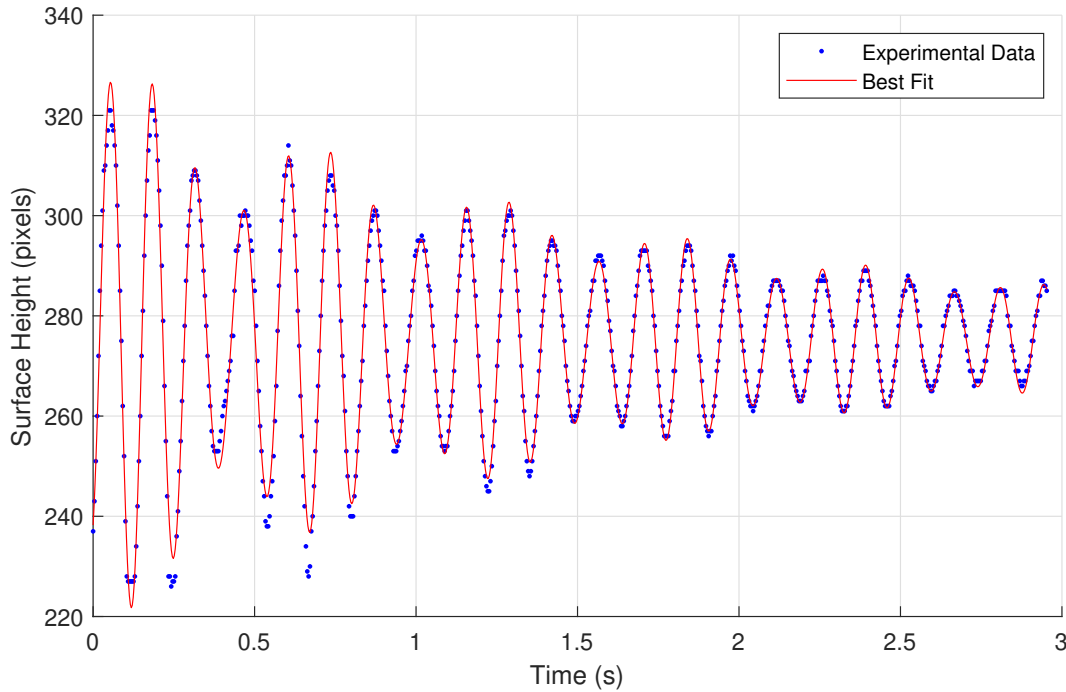
**Figure 2.22.** Power spectral density for the stair step tank at a sixty degree slosh angle.

**Table 2.9.** Standard uncertainty in experimental values for the stair step tank.

Mode	Slosh Angle	0	30	45	60	90
1	Uncertainty in $\zeta$ (%)	0.138	0.550	0.354	0.103	0.131
	Uncertainty in $\omega$ (rad/s)	0.103	0.303	0.272	0.073	0.073
2	Uncertainty in $\zeta$ (%)	0.666	0.650	0.342	0.317	0.541
	Uncertainty in $\omega$ (rad/s)	0.553	0.787	0.115	0.118	0.179

### 2.2.5 Trapezoid

The trapezoid tank was analyzed at a 0, 30, 45, 60, and 90-degree slosh angle. Figure 2.23 shows an example of surface height over time data obtained for this geometry. At all slosh angles, the data has the appearance of the sum of at least two damped sinusoids and can be modeled with Equation 2.2. The damping constant was again restricted to negative values. Parameters do not vary in time.



**Figure 2.23.** Example of data obtained for a slosh event in the trapezoid tank as well as the best fit sum of two damped sinusoids.

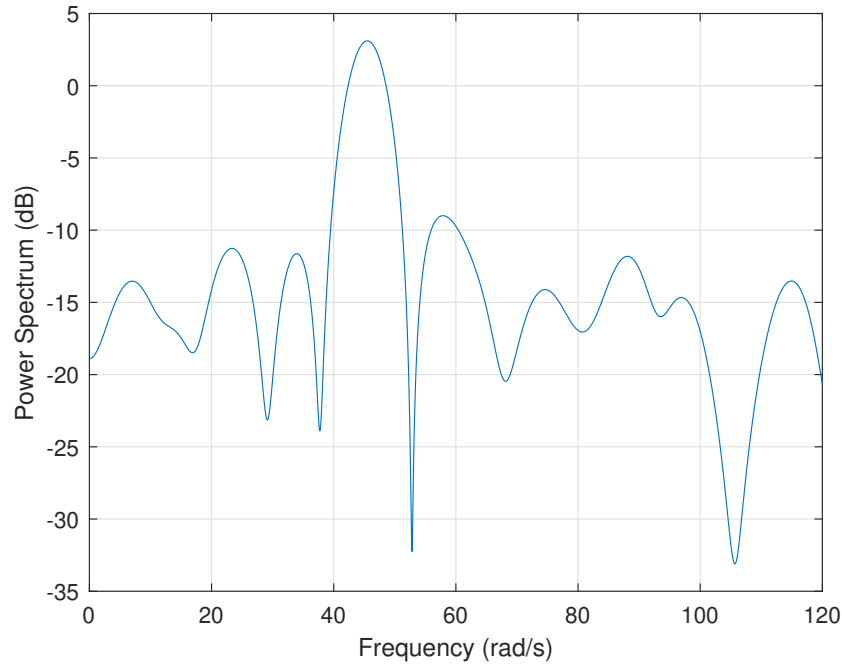
A summary of the experimental data is given in Table 2.10. Again, the damping ratio and natural frequency are relatively consistent, although the natural frequency does vary more than previously discussed geometries.

The two modes presented in Table 2.10 are the modes corresponding to the two dominant frequencies in the results. There are, however, quite a few frequencies contributing to the behavior in this geometry. Figure 2.24 shows an example of the power spectral density. The power spectral density did vary somewhat between slosh events even for the same slosh

**Table 2.10.** Experimental results for the trapezoid tank.

	<b>Mode 1</b>		<b>Mode 2</b>	
<b>Slosh Angle</b>	$\omega_n$ (rad/s)	$\zeta$ (%)	$\omega_n$ (rad/s)	$\zeta$ (%)
<b>0</b>	45.5	1.08	59.2	1.67
<b>30</b>	45.5	1.13	56.0	1.09
<b>45</b>	45.6	1.47	56.9	1.86
<b>60</b>	45.7	1.32	58.2	1.69
<b>90</b>	45.7	1.15	58.2	1.50

angle. For example, some cases did not have as prominent of a peak at approximately 34 rad/s. However, the two largest peaks were generally the modes listed in Table 2.10.

**Figure 2.24.** Power spectral density for the trapezoid tank at a forty-five degree slosh angle.

It is less clear in this geometry which frequency corresponds to slosh along which direction. Along the longest dimension, the first and second natural frequencies are 32.0 and 57.5 rad/s assuming a rectangular tank of that dimension. By assuming the slanted wall was actually a vertical wall at the fill level width, the first and second natural frequencies would be 39.2 and 68.5 rad/s. It would make sense that the two most prominent frequencies would be the



first natural frequency in each direction, but this would result in a percent difference of over 40%. This is further discussed in the CFD chapter. It is likely that the two slosh modes found in the experiment are actually the first natural frequency in the short direction and the second natural frequency in the long direction.

Uncertainty in the presented slosh parameters is given in Table 2.11. Plots showing error bars are given with the CFD results to better show agreement with CFD results.

**Table 2.11.** Standard uncertainty in experimental values for the trapezoid tank.

Mode	Slosh Angle	0	30	45	60	90
1	Uncertainty in $\zeta$ (%)	0.130	0.182	0.606	0.378	0.382
	Uncertainty in $\omega$ (rad/s)	0.127	0.088	0.118	0.152	0.121
2	Uncertainty in $\zeta$ (%)	0.609	0.534	0.474	1.072	1.096
	Uncertainty in $\omega$ (rad/s)	3.178	2.348	0.412	3.055	2.749

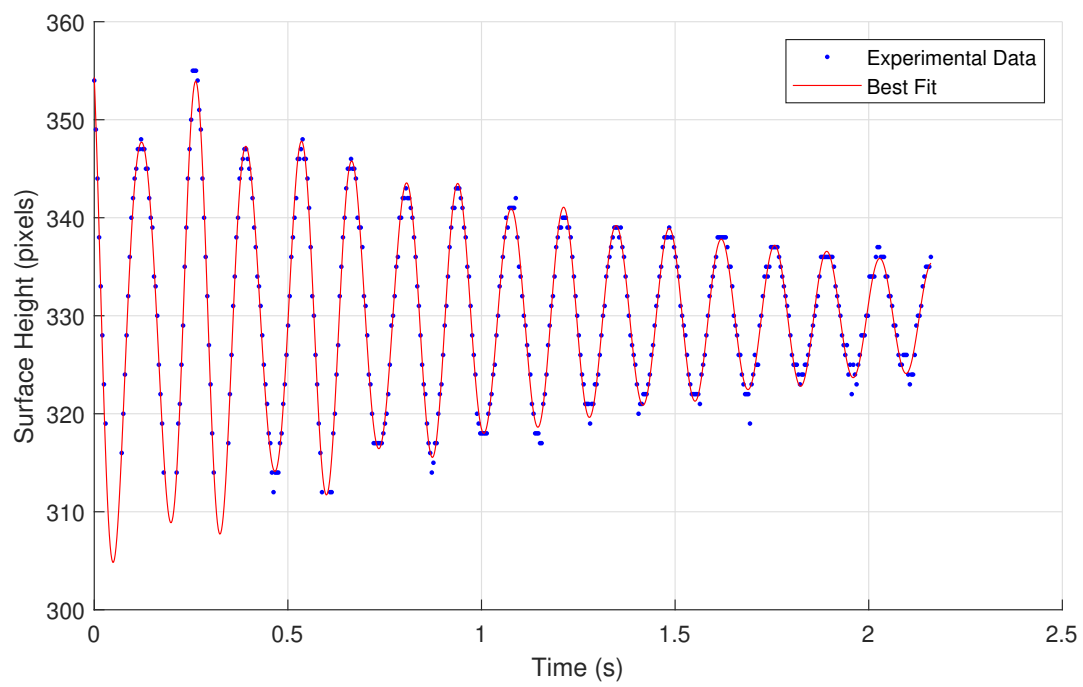
### 2.2.6 Quarter Annular Cylinder

The quarter annular cylinder was analyzed at a 0, 30, 45, 60, and 90-degree slosh angle. Figure 2.25 shows an example of surface height over time data obtained for this geometry. At all slosh angles, the data has the appearance of the sum of at least two damped sinusoids and can be modeled with Equation 2.2. The damping constant was again restricted to negative values. Parameters do not vary in time.

A summary of the experimental data is given in Table 2.12. Similarly to the trapezoid tank, the results are relatively consistent but with more variability in the natural frequency than simpler geometries.

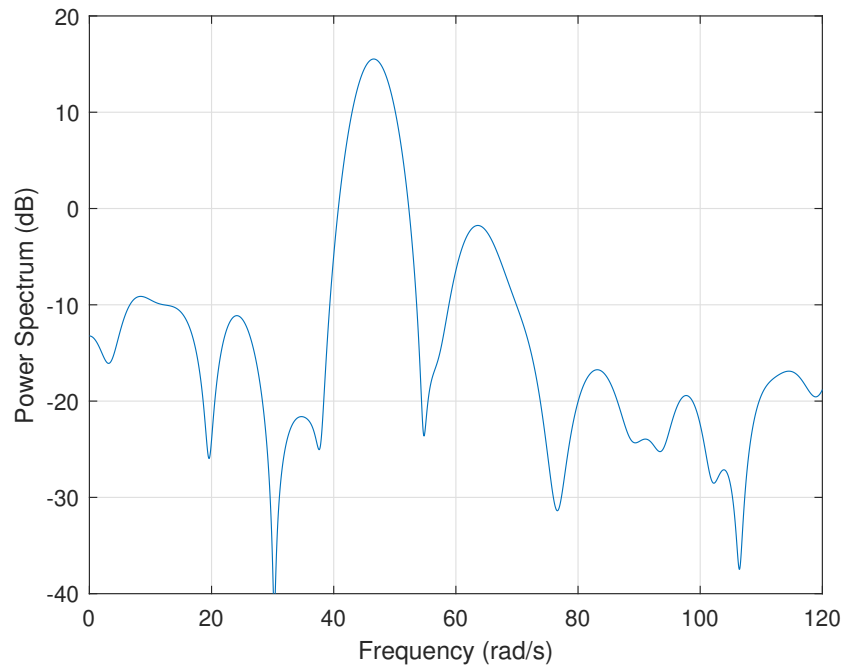
**Table 2.12.** Experimental results for the quarter annulus tank.

	Mode 1		Mode 2	
Slosh Angle	$\omega_n$ (rad/s)	$\zeta$ (%)	$\omega_n$ (rad/s)	$\zeta$ (%)
0	46.3	1.48	65.8	2.77
30	46.1	1.45	65.6	2.46
45	46.5	1.57	67.5	2.06
60	46.5	1.38	63.3	2.18
90	46.1	1.74	63.9	2.07



**Figure 2.25.** Example of data obtained for a slosh event in the quarter annulus tank as well as the best fit sum of two damped sinusoids.

The two modes presented in Table 2.12 are the most prominent peaks in the power spectrum such as the one shown in Figure 2.26. Other peaks are visible, but generally quite a bit smaller than the two main modes. The SP-106 presents an analytical solution for only a quarter sector tank, which is similar to this tank but with an inner radius equal to zero. For this type of tank, two slosh modes would be expected. One would act along the circumferential direction, and one along the radial direction. It is likely that the two most prominent modes are acting in these two direction as well.



**Figure 2.26.** Power spectral density for the quarter tank at a forty-five degree slosh angle.

Uncertainty in the presented slosh parameters is given in Table 2.13. Plots showing error bars are given with the CFD results to better show agreement with CFD results.

### 2.2.7 Analytical Comparison

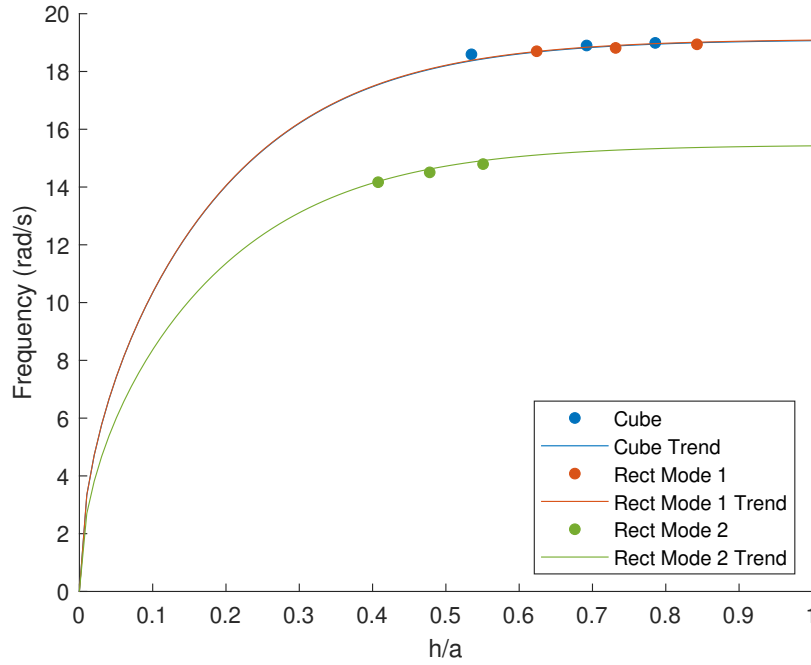
While the natural frequencies in the basic geometries tended to be greater than the analytical solutions, it was found that the frequencies did follow a hyperbolic tangent. While there is not enough data to be very confident in the correlation, an equation was curve fit to

**Table 2.13.** Standard uncertainty in experimental values for the quarter annular cylinder tank.

Mode	Slosh Angle	0	30	45	60	90
1	Uncertainty in $\zeta$ (%)	0.149	0.092	0.481	0.058	0.069
	Uncertainty in $\omega$ (rad/s)	0.084	0.092	0.092	0.099	0.221
2	Uncertainty in $\zeta$ (%)	1.347	1.345	0.898	1.218	0.863
	Uncertainty in $\omega$ (rad/s)	1.878	0.950	2.027	0.397	0.554

data for the cube and rectangular tanks to determine the difference between the analytical solution and experimental data. The result was Equation 2.9. It is very similar to the analytical solution, Equation 1.1, which instead has the value of  $\pi$  for each constant. This curve fit is plotted along with the available experimental data in Figure 2.27.

$$\omega^2 = 3.578 \left( \frac{g}{a} \right) \tanh \left[ 3.017 \left( \frac{h}{a} \right) \right] \quad (2.9)$$



**Figure 2.27.** Experimental data for rectangular tanks fit to a hyperbolic tangent equation of the form used for the analytical solution.

### 3. COMPUTATIONAL FLUID DYNAMICS

CFD simulations for each experimental geometry were created in an effort to show the ability of CFD to model conformal geometries and the capabilities of an open-source CFD code.

#### 3.1 Setup

The following sections describe how the CFD simulations were set up and verified.

##### 3.1.1 Server

OpenFOAM simulations were run on Amazon Web Services Elastic Cloud Compute [36]. This service allows users to create an image of a base system that can then be used to start multiple instances at once without the need to reinstall software every time. OpenFOAM v8 was installed on this server and was used for all results presented here except for some of the smallest meshes in convergence studies [37]. These smallest meshes were run on a desktop running OpenFOAM v5. One case was run in each version, with no difference between results. For consistency, all final results were run on Amazon Web Services. However, the author is confident in the use of different OpenFOAM versions for mesh convergence studies.

##### 3.1.2 Solvers

The interFoam volume-of-fluid solver was used for these simulations. More details about the interFoam solver and exact specifications are given in the examples in Appendix A.

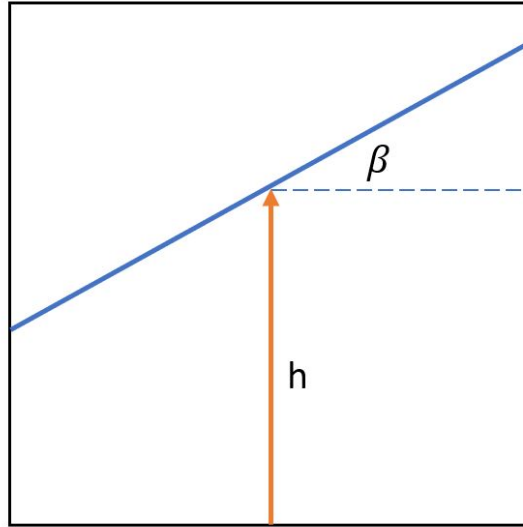
A laminar turbulence model was used because these simulations only looked at the linear regime of slosh with no large motions. This is typical of slosh simulations of this type.

An adjustable time step is used with a maximum values of 1 ms and a maximum Courant number of 0.5. The simulation is run for a total of twenty seconds.

##### 3.1.3 Initialization

The liquid surface was initialized at a 5 degree angle from the horizontal in order to keep slosh within the linear regime. The center point of the initialized surface is at the undisturbed

surface height,  $h$ , as shown in Figure 3.1. This surface is rotated counterclockwise by the slosh angle,  $\theta$ .



**Figure 3.1.** Initialization of the liquid surface by angle,  $\beta$ , from the horizontal.

### 3.1.4 Properties

Standard properties for distilled water and air were used in CFD modeling as given in Table 3.1.

**Table 3.1.** Fluid properties used in CFD models.

<b>Water</b>	
Kinematic Viscosity ( $m^2/s$ )	$1 \times 10^{-6}$
Density ( $kg/m^3$ )	998.2
<b>Air</b>	
Kinematic Viscosity ( $m^2/s$ )	$1.48 \times 10^{-5}$
Density ( $kg/m^3$ )	1
<b>Interface</b>	
Surface Tension ( $N/m$ )	0.07

### 3.1.5 Boundary Conditions

These CFD simulations model closed tanks, so all boundaries are assigned to a wall type and have the same conditions applied. There are three boundary conditions used in these simulations- velocity, pressure, and contact angle. The velocity condition is no-slip. The pressure boundary condition is set to fixed flux pressure, which adjusts the pressure gradient at the boundary to fit the velocity no-slip boundary condition. The contact angle between the liquid and the container wall is set to zero gradient, which is the equivalent of a ninety-degree contact angle.

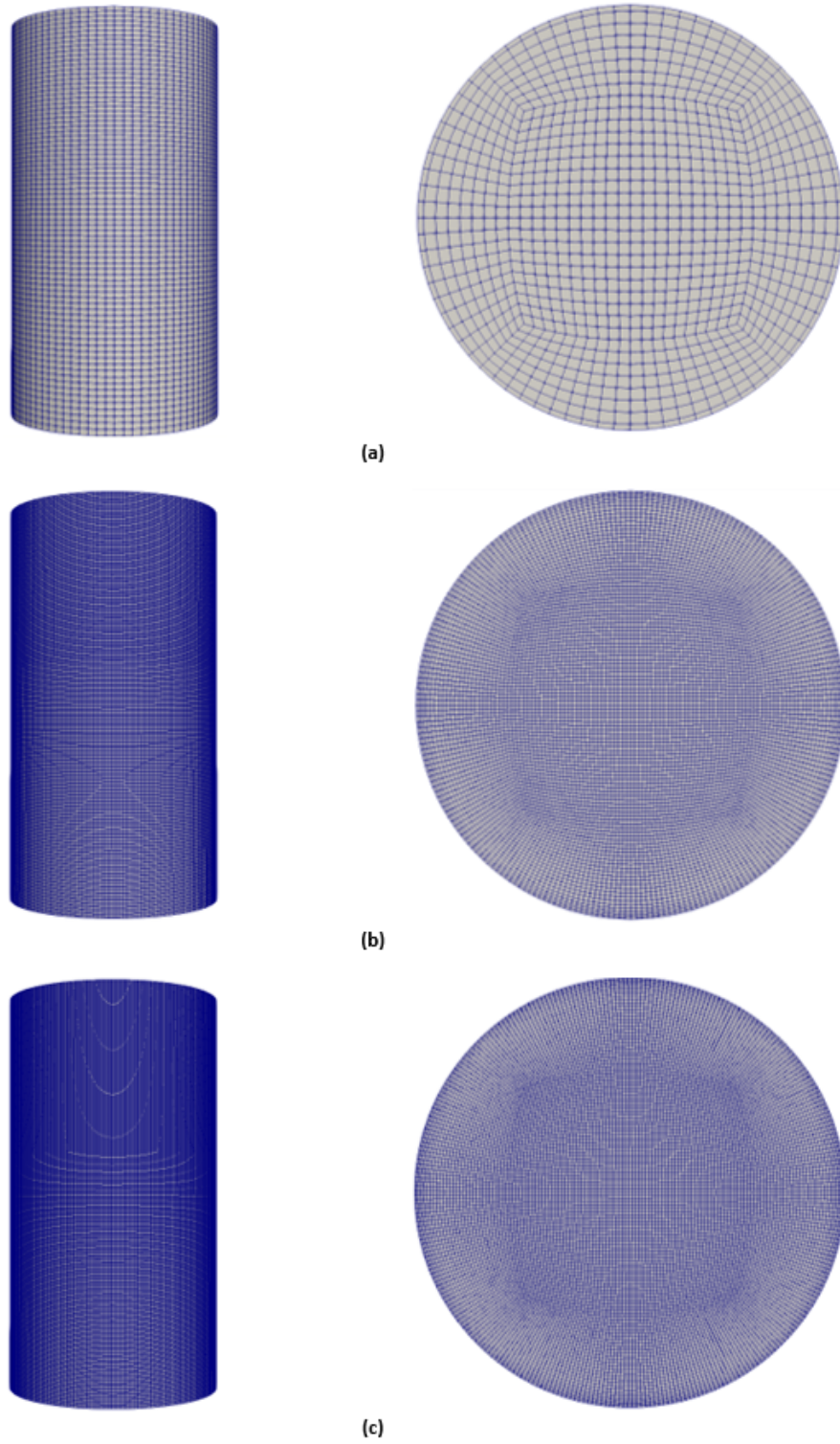
### 3.1.6 Mesh

Mesher for each geometry were created such that the maximum aspect ratio was below three, there were at least 100 elements across the length of the tank, and the refined region surrounding the surface extended at least 3-4 times the initial amplitude of the surface [3]. Areas near the walls and near the surface were more refined.

## Cylinder

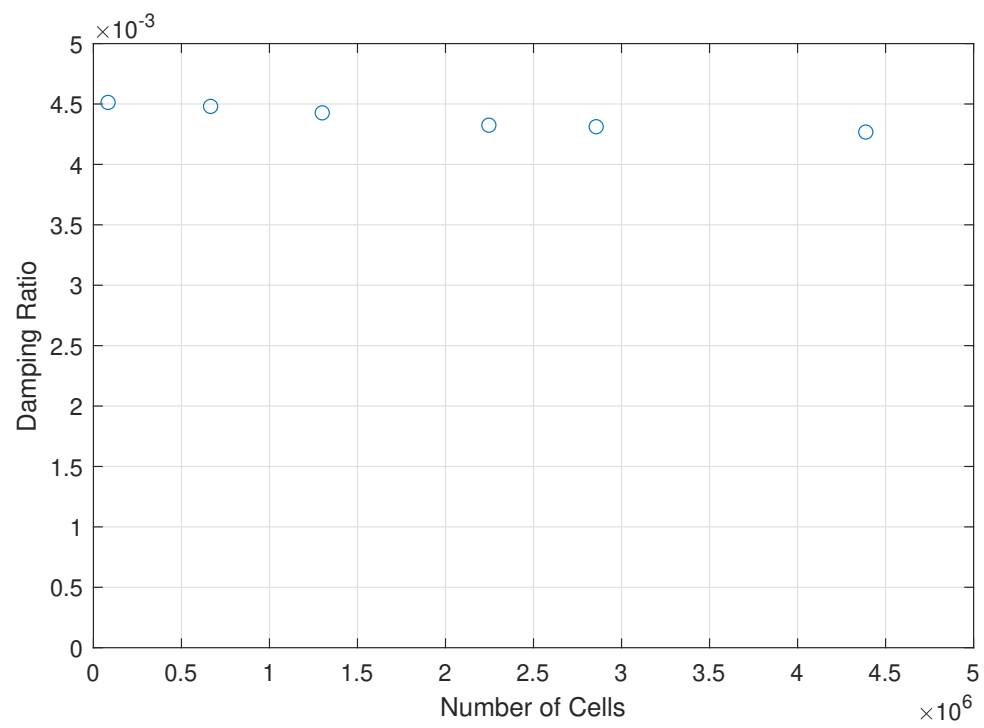
The mesh used for the cylindrical geometry was a slightly modified butterfly mesh to prevent a singularity at the axis. While many butterfly meshes have a rectangular prism for the center block, this mesh uses a center block with edges that follow a slight arc. This prevents highly non-orthogonal cells near the corners of the center block. This geometry may be most clearly seen in the least refined mesh shown in Figure 3.2.

Six different refinement levels were investigated for the cylindrical tank. Overall, there was very little change in the damping ratio, even between the most refined mesh and meshes which did not meet the condition that 100 elements should span the diameter. This is shown in Figure 3.3. The mesh appeared to have converged at around 2.2M cells, so this mesh was used for the final solutions. This grid has a maximum aspect ratio of 2.60 and 108 cells across the tank diameter.

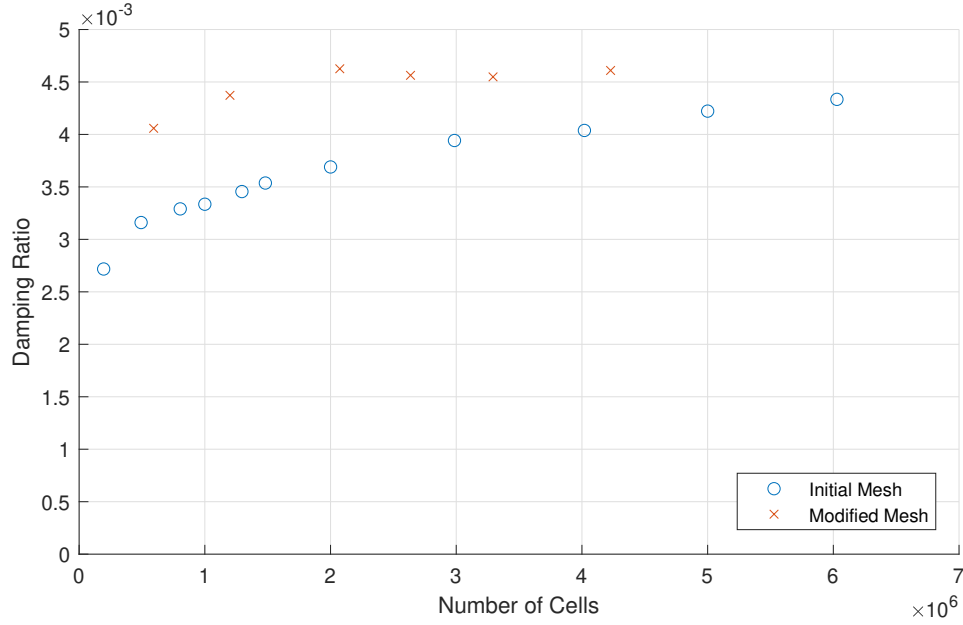


**Figure 3.2.** Side view (left) and top view (right) of the (a) 83T cell mesh, (b) 2.2M cell mesh, and (c) 4.4M cell mesh for the cylinder.





**Figure 3.3.** Results of mesh convergence study in the cylindrical tank.

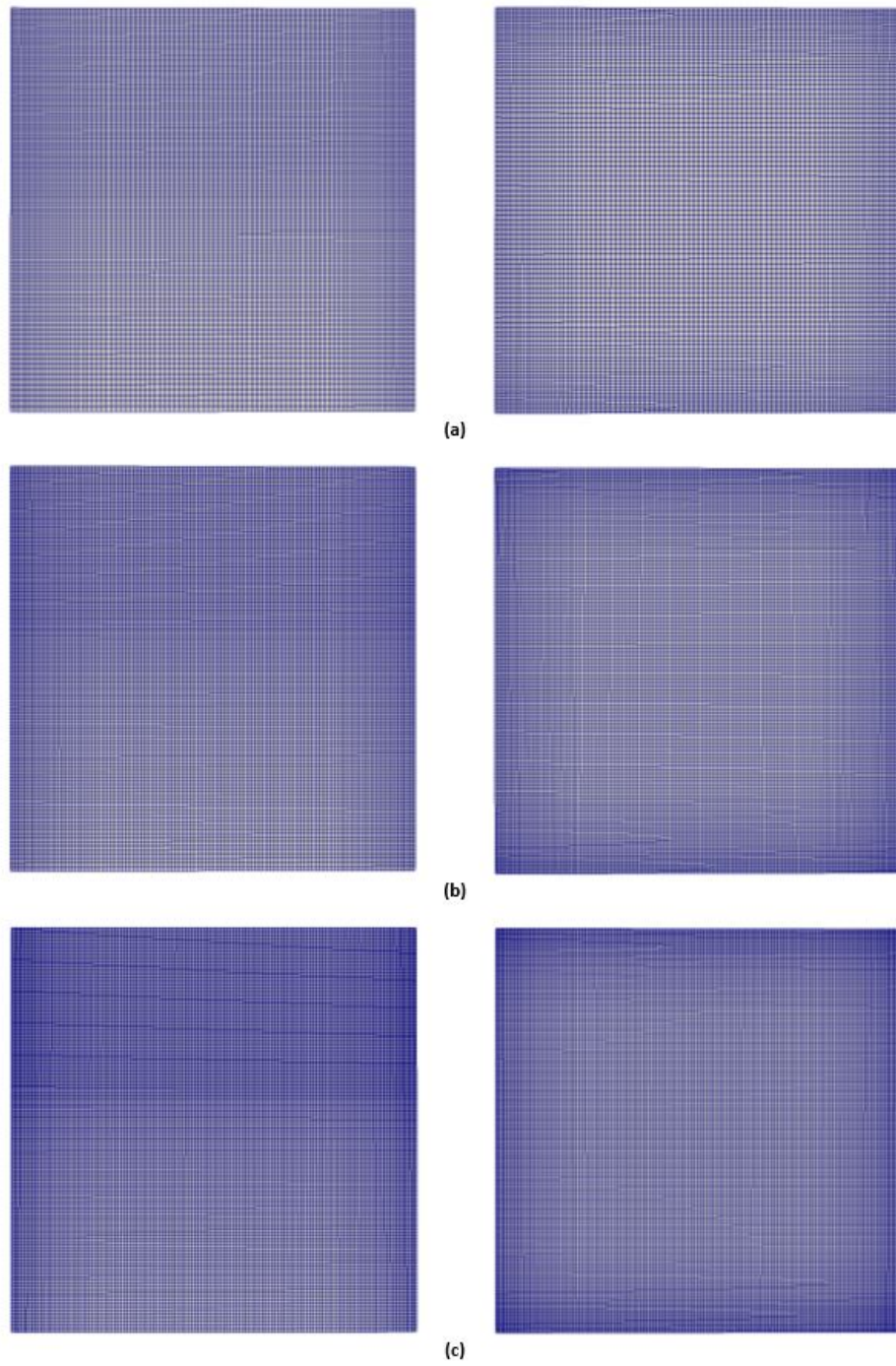


**Figure 3.4.** Results of mesh convergence study in the cube tank.

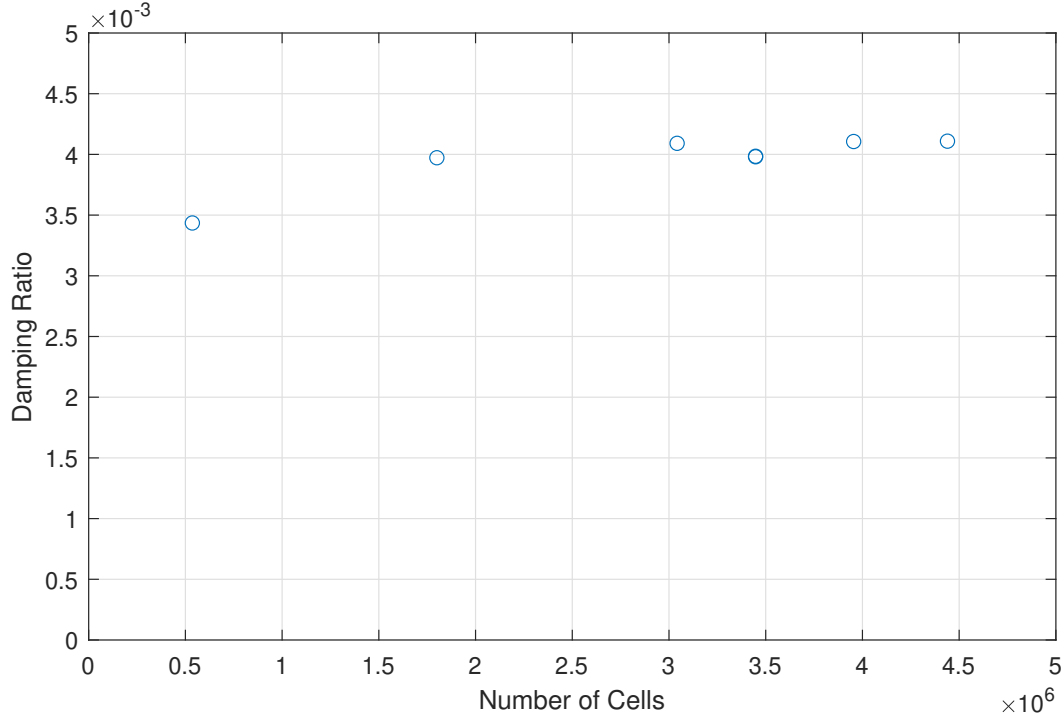
## Cube

Two different meshes for the cube geometry were tested. The first did not have sufficient grid refinement near the walls and free surface, so it did not converge even for a mesh size of approximately 6 million cells. A modified mesh was created with further refinement near the walls and liquid surface while still keeping the maximum overall aspect ratio below the recommended value of three [3]. This mesh converged more quickly and approached the solution with about 2.6 million cells as shown in Figure 3.4. There was some fluctuation in the value for more refined meshes, but the solution did not diverge. The fluctuations are likely due to the difficulty in calculating a damping ratio with such a small value.

Six different levels of grid refinement were tested for the modified mesh. Three of these grids are pictured in Figure 3.5. There appears to be little change in the solution beyond 2.6M cells, so this grid was used for simulations of the cube geometry. This grid has a maximum aspect ratio of 2.54 and 130 cells across the tank length. The refined region surrounding the free surface extends more than three times the initial slosh height past the position of the free surface at rest.



**Figure 3.5.** Side view (left) and top view (right) of the (a) 1.2M cell mesh, (b) 2.6M cell mesh, and (c) 4.2M cell mesh for the cube.



**Figure 3.6.** Results of mesh convergence study in the rectangular tank.

### Rectangular Prism

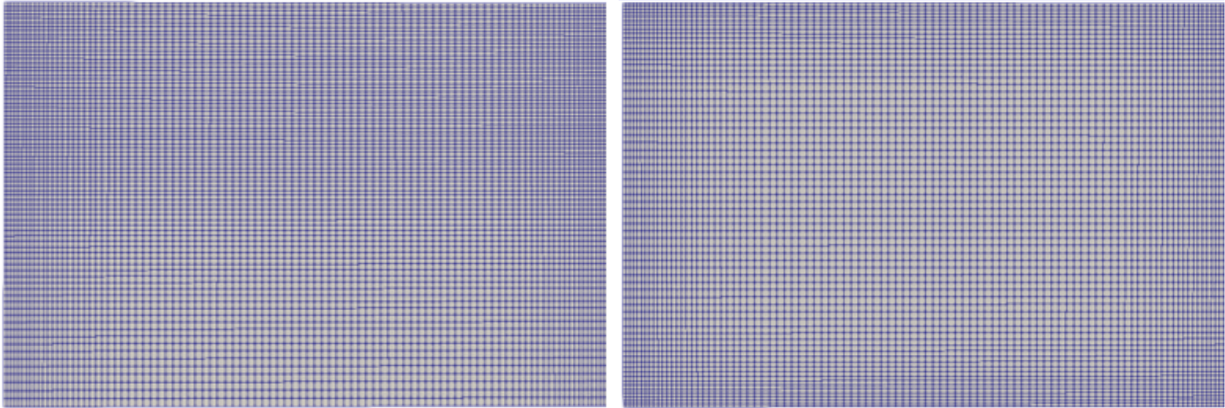
The rectangular mesh was created using the cube mesh as a starting point. For example, at least 130 cells are needed along the shortest side. The longest side was further refined to maintain an appropriate aspect ratio. As shown in Figure 3.6, there was some slight fluctuation of the damping ratio among the most refined meshes, however, this is again likely due to difficulty in calculating a small damping ratio.

Figure 3.7 shows three levels of mesh refinement tested, including the 3M cell mesh used for final solutions. This grid has a maximum aspect ratio of 2.96, 150 cells across the longest edge, and 130 cells across the shortest edge.

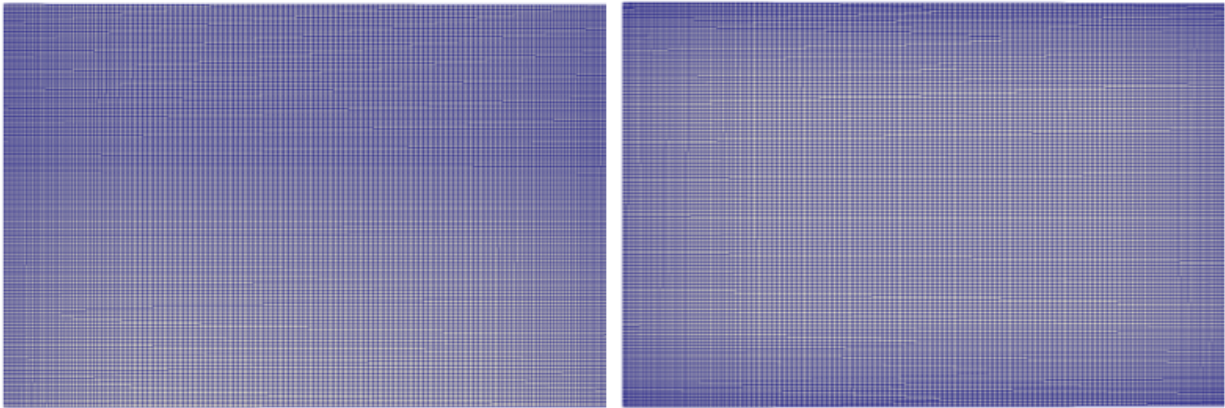
### Stair Step

Four levels of grid refinement were tested for the stair step tank, as shown in Figure 3.8. Three of these levels are shown in Figure 3.9. The mesh was made of three blocks, each

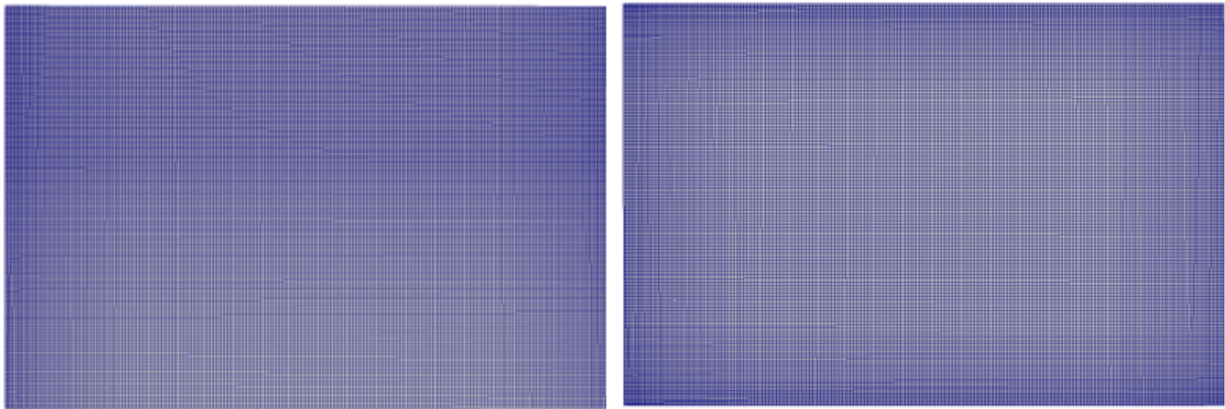




(a)



(b)

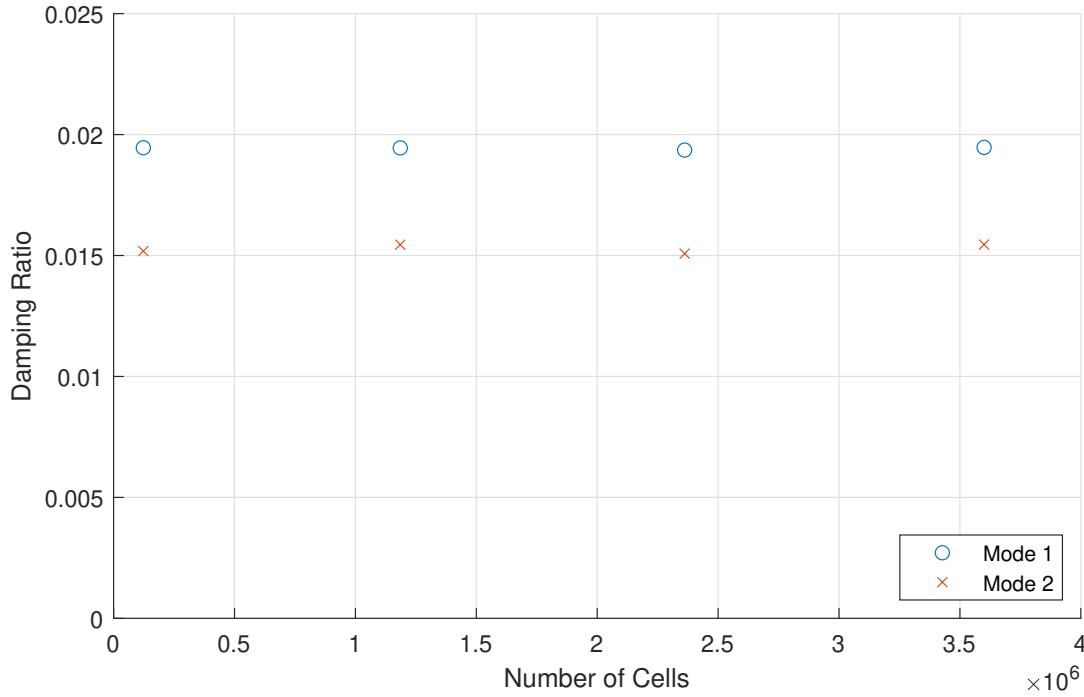


(c)

**Figure 3.7.** Side view (left) and top view (right) of the (a) 0.5M cell mesh, (b) 3M cell mesh, and (c) 4.4M cell mesh for the rectangular prism.

of which was refined more at the edges and in the area of the free surface. This creates a well-refined region surrounding the center corner, as well as at the walls in general.

There was little change in damping ratio at all levels of grid refinement tested. A mesh with 1.2M cells, a maximum aspect ratio of 2.89, and 138 cells across the tank length was used for final solutions.

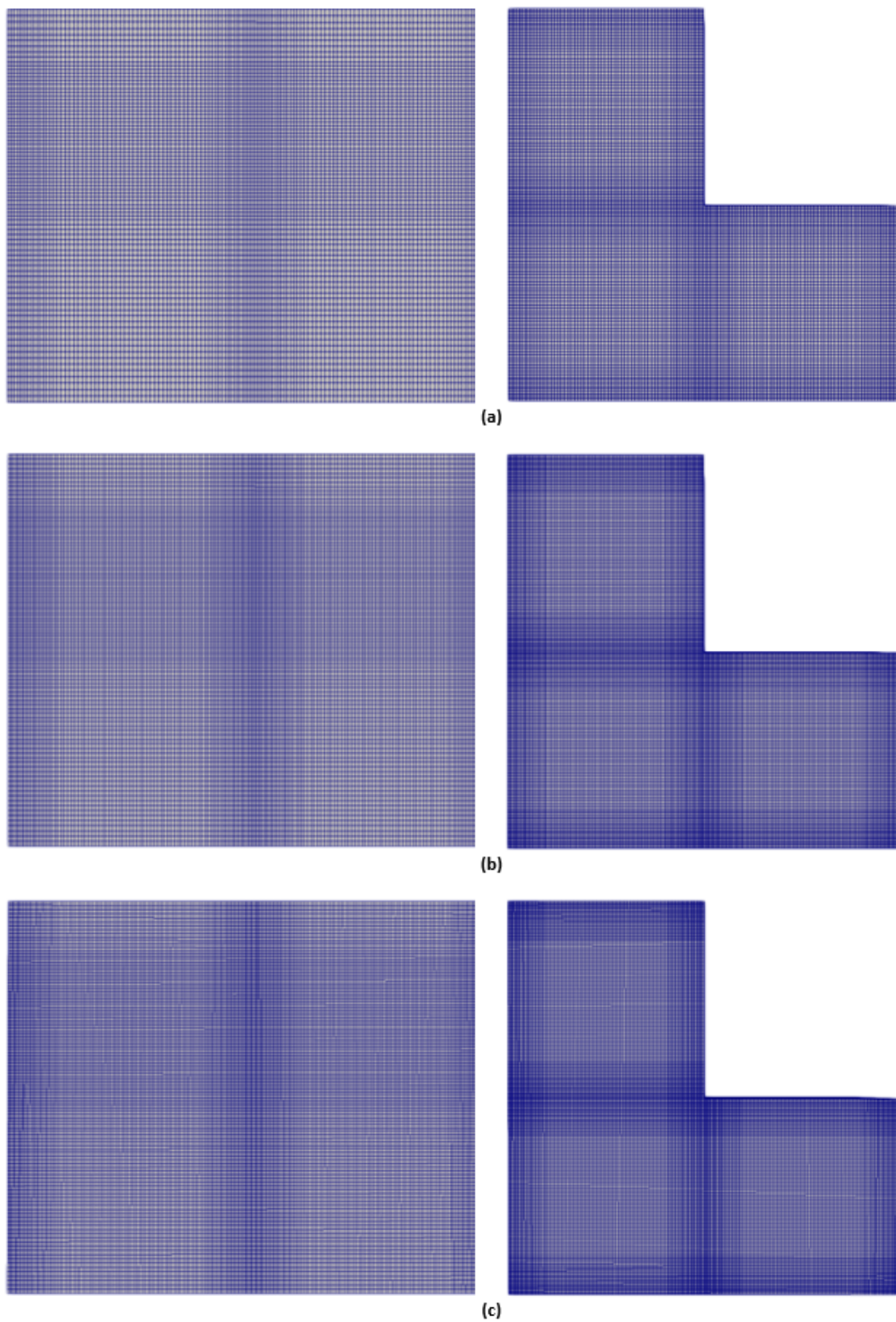


**Figure 3.8.** Results of mesh convergence study in the stair step tank.

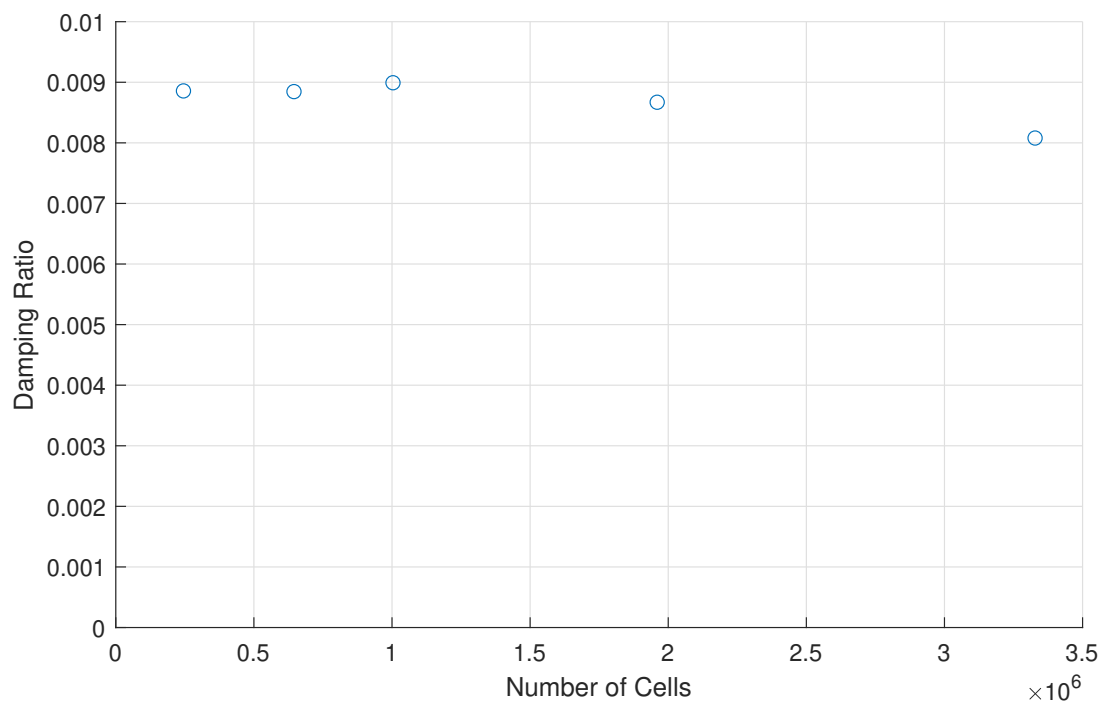
## Trapezoid

Five levels of grid refinement were examined for the trapezoid tank, as shown in Figure 3.10. Three of these are pictured in Figure 3.11. Unfortunately, the mesh did not appear to converge even for a mesh with 3.3M cells. Because of higher natural frequencies and small motions of the liquid caused by the slightly slanted wall, the time step was very low and significantly increased run time compared to other simulations. For example, the 3.3M mesh took 12.7 days on 8 processors to model 6 seconds of simulation time. These run times were prohibitively large for the scope of this research.





**Figure 3.9.** Side view (left) and top view (right) of the (a) 1.2M cell mesh, (b) 2.4M cell mesh, and (c) 3.6M cell mesh for the stair step tank.



**Figure 3.10.** Results of mesh convergence study in the trapezoid tank.



For final solutions, a mesh with 1M cells was used. This mesh had a maximum aspect ratio of 2.7, 112 cells along the x axis, and 80 cells along the y axis. While it would be preferred to have a fully converged mesh, this un-converged solution can still give the general slosh behavior. The frequency changes little between refinement levels, with a maximum of 0.15% difference between any of the solutions and the 3.5M mesh. The damping ratio has a maximum 11.3% difference between any of the solutions and the 3.5M mesh. Therefore, confidence in the final solution frequencies is high, even if it is possible that the damping ratios are not quite converged.

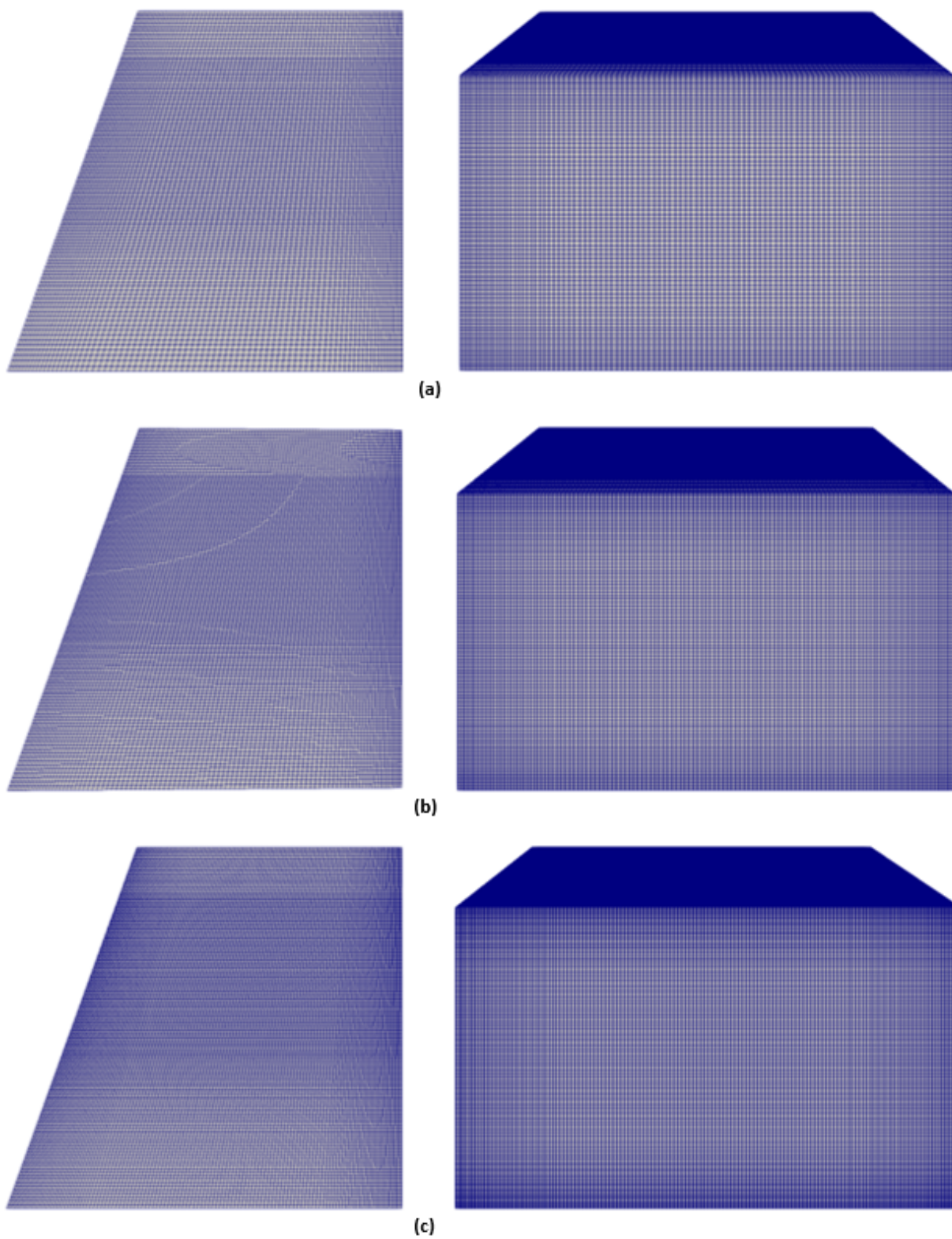
### **Quarter Annular Cylinder**

Results of the quarter annular tank convergence study are shown in Figure 3.12. Six levels of grid refinement were tested, including the three pictured in Figure 3.13. There was again little change in damping ratio at all levels of grid refinement tested. The mesh used for final solutions has 2.4M cells, a maximum aspect ratio of 2.87, 100 cells radially, and 120 cells along the arc.

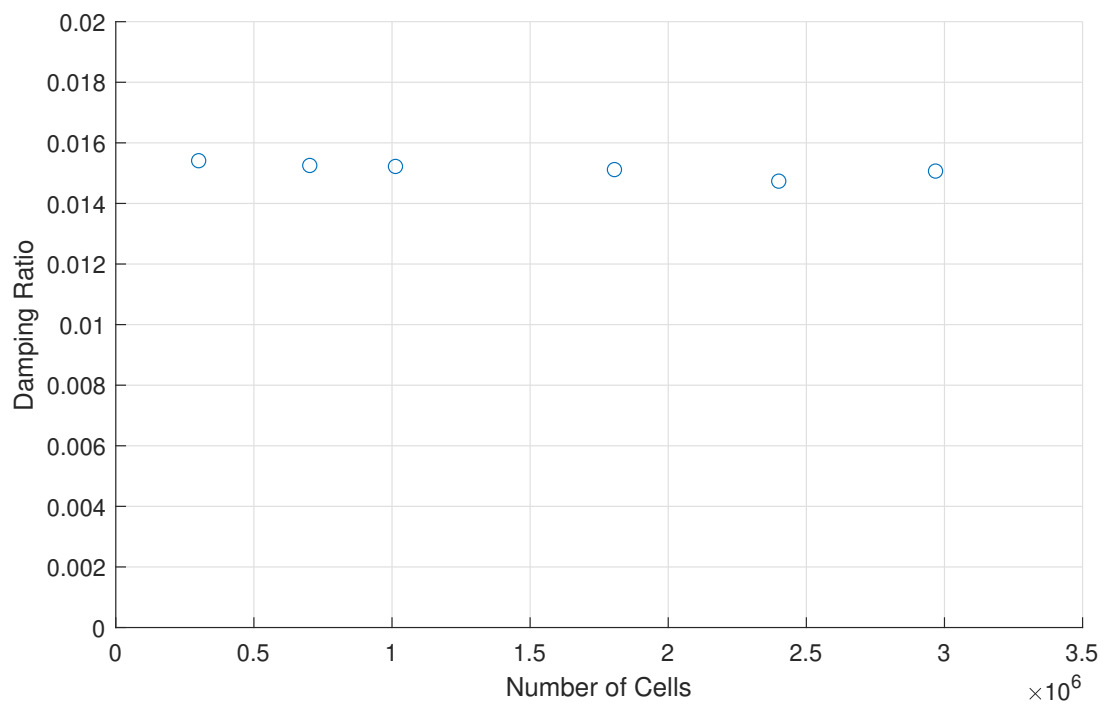
### **Run Time**

CFD simulations of the basic geometries ran significantly quicker than the more complicated geometries. This is due to the fact that the basic geometries are larger with lower frequencies, combined with simpler flow in the simpler geometries. Run time for the basic geometries are shown in Figure 3.14. Some simulations were run on a different number of processors, so total run time is multiplied by this number of processors in an attempt to make a good comparison. However, due to diminishing returns when running simulations on more processors, this does cause the variations seen in the plot. In general, the cylinder, cube, and rectangle ran for a similar amount of time for a similar level of grid refinement.

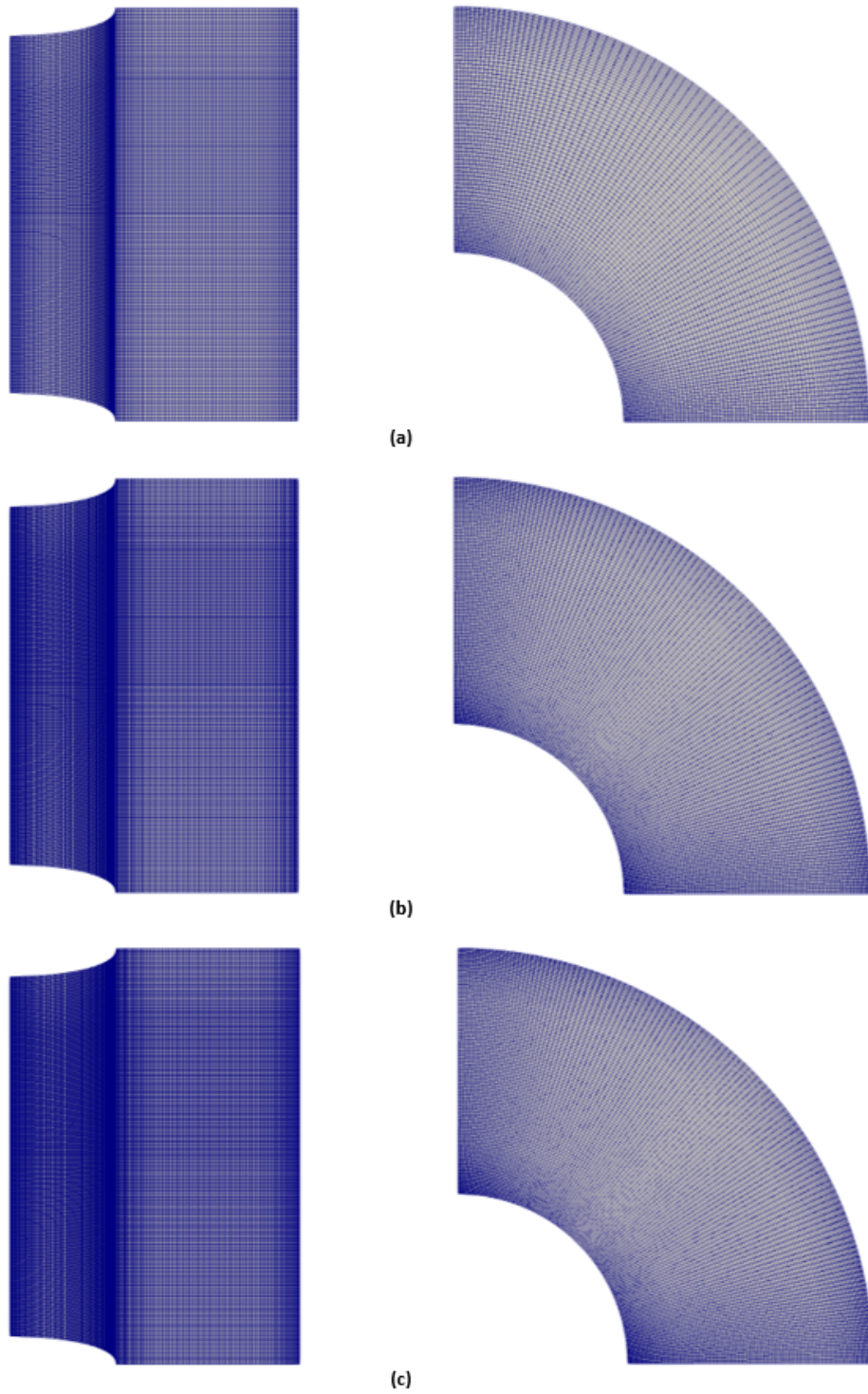
Figure 3.15 shows the run time for the more complicated geometries. All of these simulations were run on the same number of processors, so the plot strictly increases for larger meshes unlike for the basic geometries. As shown, the trapezoid tank does have signifi-



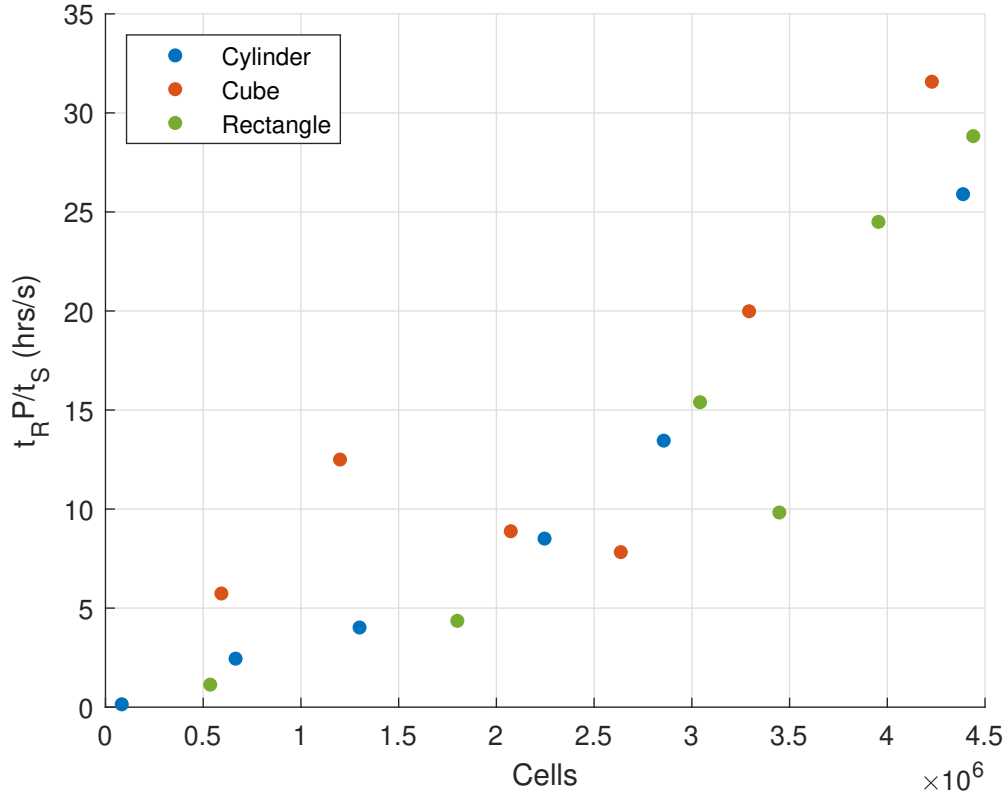
**Figure 3.11.** Side view (left) and top view (right) of the (a) 1M cell mesh, (b) 2M cell mesh, and (c) 3.3M cell mesh for the trapezoid tank.



**Figure 3.12.** Results of mesh convergence study in the quarter annular cylinder tank.



**Figure 3.13.** Side view (left) and top view (right) of the (a) 1M cell mesh, (b) 2.4M cell mesh, and (c) 3M cell mesh for the quarter annular cylinder.



**Figure 3.14.** Run time for basic geometries.  $t_R$  is run time,  $t_S$  is simulation time, and  $P$  is the number of processors.

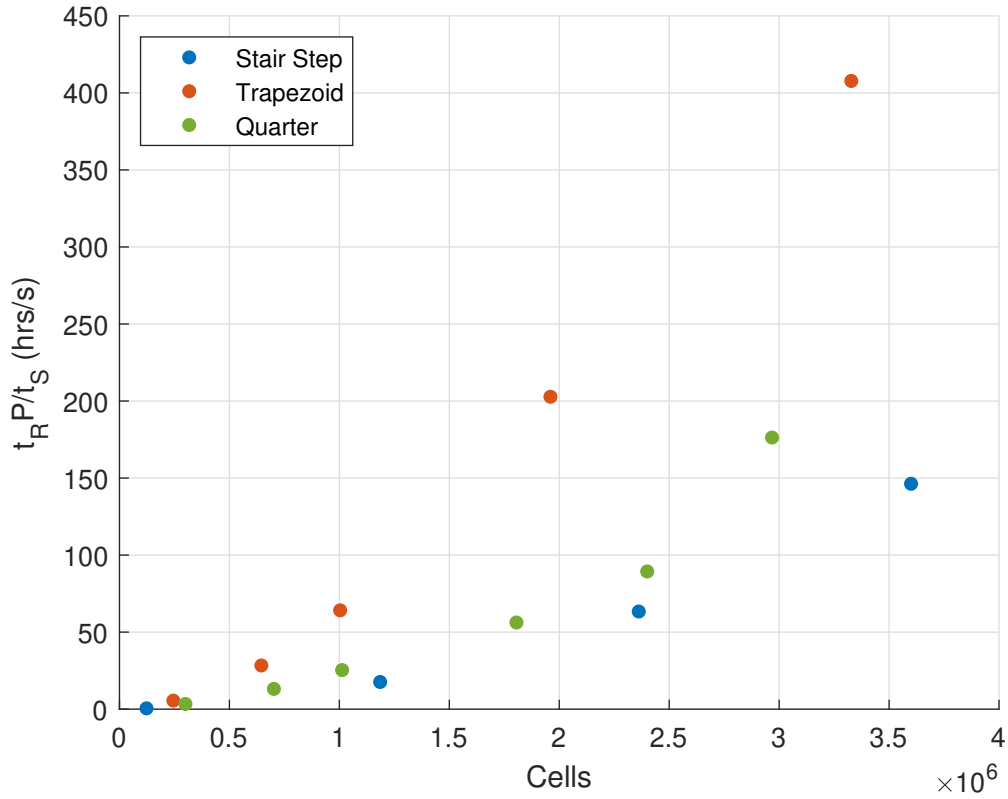
cantly higher run time than either the quarter or stair step tank. This could indicate that a better-designed mesh could be needed.

### 3.1.7 Post-Processing Function Objects

OpenFOAM post-processing function objects are explained in more detail in [Appendix A](#). Two function objects were used in this research.

#### Forces

The first function object used, forces, is a built in function object which gives as output the total forces and moments on a specified boundary. The required inputs are the name of the boundary or boundaries where the forces and moments should be computed, the value



**Figure 3.15.** Run time for complicated geometries.  $t_R$  is run time,  $t_S$  is simulation time, and  $P$  is the number of processors.

for the density of the fluid, and the center of rotation to be used for moment calculation. For example, the forces function could be specified as follows:

```

forces                                     //Name of directory for output files
{
  type      forces;                       //Name of function object
  libs      ("libforces.so");             //Library name
  patches    (walls);                     //Boundaries where forces are computed
  rho        rhoInf;                      //Which density to use
  rhoInf     1000;                        //Value of density to use
  CofR       (0 0 0);                     //Center of rotation for moment calculation
}

```



## Center of Mass

The second function object was user-defined and calculated the center of mass of the liquid at each time step. This code was created because existing methods for calculating the center of mass took place after the simulation completed. A center of mass data point could only be obtained at time steps where a full time directory had been written during simulation. Because of this, time directories had to be written frequently in order to obtain sufficient center of mass data points for good calculation of slosh parameters. Time directories require significant storage space, especially for large meshes. Calculating the center of mass after the simulation finished was not optimal, so a function object was written to calculate the center of mass at each time step while the simulation runs, without the requirement of a time directory.

A center of mass calculation code written by Bob Manning was used as an example for how to reference OpenFOAM parameters and use these to calculate the center of mass. The total volume, Equation 3.1, and the volume of the liquid, Equation 3.2, were first calculated as a simple double check to ensure the code was working correctly. This also allows the user to check that fields were set correctly in the model at the first time step. Total volume of the liquid over time can also be tracked to ensure that the value does not change.

$$V_{total} = \sum V_i \quad (3.1)$$

$$V_{liquid} = \sum V_i \alpha_i \quad (3.2)$$

Center of mass is calculated as the sum of the volume of an individual cell multiplied by the volume fraction and by the distance from the origin in the direction of interest (x/y/z). This is then divided by the total volume of the liquid as shown in Equation 3.3.

$$CM_{liquid} = \frac{\sum V_i \alpha_i d_i}{\sum V_i \alpha_i} \quad (3.3)$$

The full center of mass function object code is available in Appendix B. The function object may be called in the controlDict file using the following code:

```

center                                //Name of directory for output files
{
type CM;                             //Name of function object
libs ("libCMFunctionObject.so"); //Library name
}

```

This function object has been shared with the OpenFOAM community through CFD Online forums [\[38\]](#).

### 3.1.8 Parameter Calculation

Parameter calculation methods are described in Section [2.1.4](#). The damped sinusoid curve fit was the best method for calculating parameters for experimental data due to the imperfect nature of experimental data. Other methods could result in a false detection of a peak value where there was a "wobble" in the data. For CFD data, false detection of a peak is not an issue because the data is "smoother." Because of this, a peak curve fit was used to determine parameters when possible. The frequency was calculated using Equation [2.5](#) and the damping ratio was calculated using the curve fit given in Equation [2.6](#). This method is considered slightly more accurate because the sine curve fit may show a good fit by slightly under-predicting peak values as long as data between the peaks fits well. There is minimal difference in the results of the methods. For example, at the highest fill fraction of the cube tank, there is less than a 1% difference in the damping ratio between the methods at any given slosh angle.

A few CFD cases, the stair step and quarter annular cylinder tank, could not use the peak curve fit method because multiple frequencies were visible in the data. In these cases, the damped sinusoid curve fit method was used.

The basic geometries were simulated for a total of twenty seconds and slosh parameters were calculated during the last ten seconds of the simulation. The complicated geometries were simulated for a total of six seconds and slosh parameters were calculated during the last four seconds of the simulation. This is because the surface is initially completely flat, a surface shape not likely to occur during real slosh. The surface develops a more natural shape with a slight curvature as the simulation progresses. Therefore, the latter part of the simulation will more accurately simulate real physics. The small, complicated geometries



**Table 3.2.** Comparison of CFD, empirical, analytical, and experimental results in the cylindrical tank.

Fill Level	$\zeta_{CFD}$ (%)	$\zeta_{emp}$ (%)	$\zeta_{exp}$ (%)	$\omega_{CFD}$ (rad/s)	$\omega_{ana}$ (rad/s)	$\omega_{exp}$ (rad/s)
<b>141.6 mm</b>	0.432	0.452	0.604	19.61	19.55	20.91
<b>120.8 mm</b>	0.448	0.453	0.537	19.61	19.55	20.87
<b>106 mm</b>	0.447	0.453	0.520	19.61	19.55	20.78

could be run for a shorter time due to their high natural frequencies. Sufficient cycles could be observed in a shorter time span.

## 3.2 Results

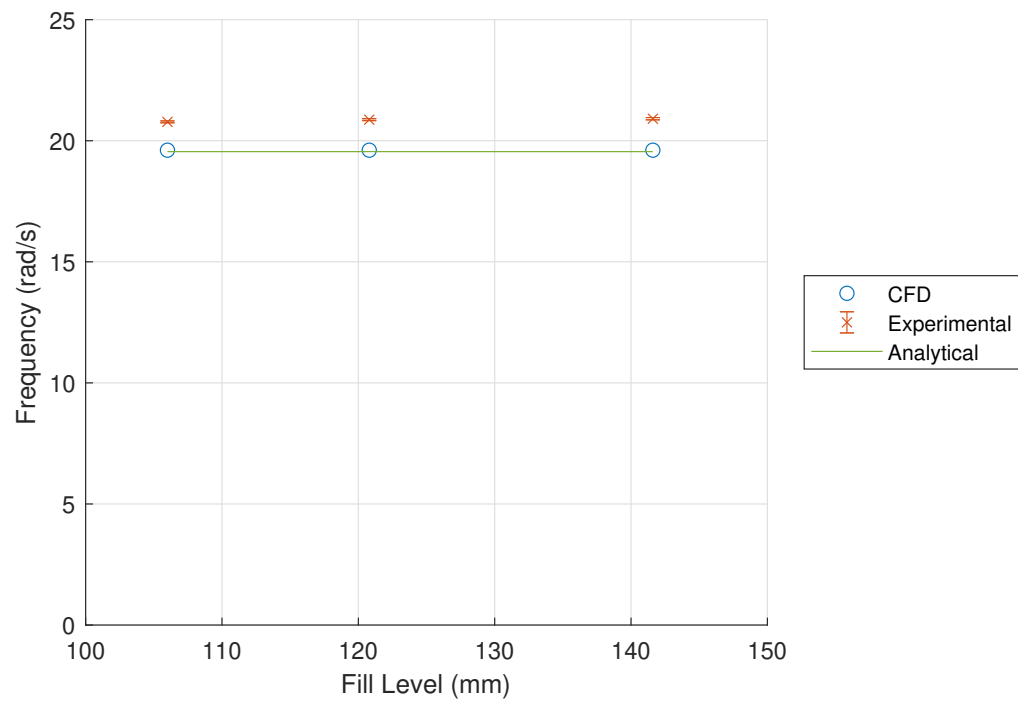
The following sections describe the results of CFD simulations for each geometry.

### 3.2.1 Cylinder

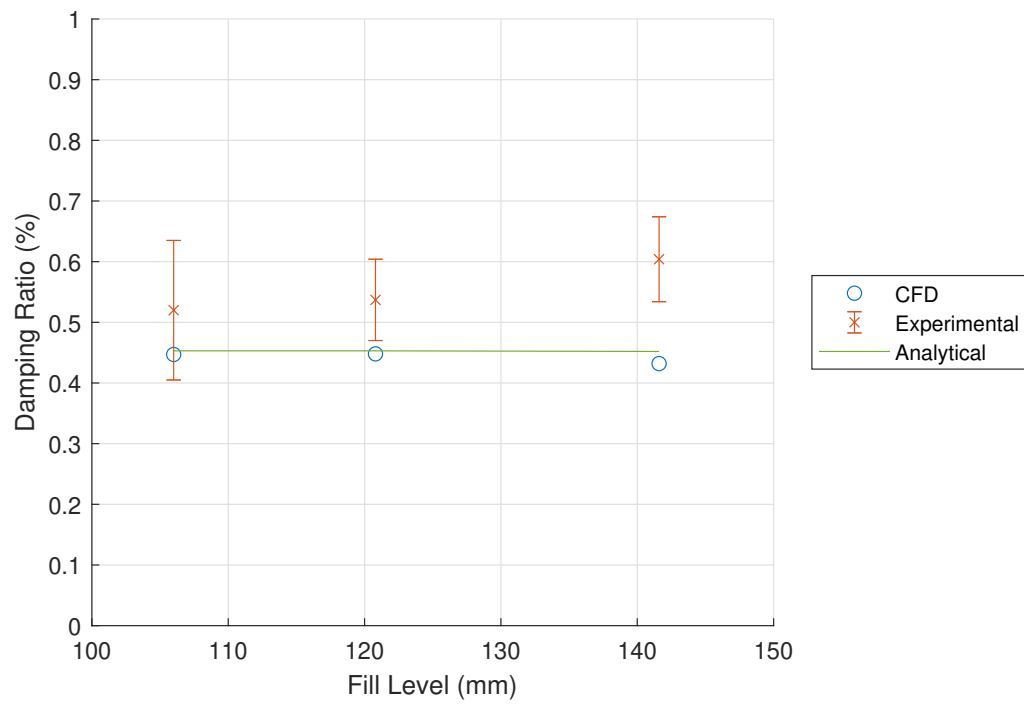
CFD cases of the cylindrical geometry were run at all three slosh angles tested in the experiment. The results are summarized in Table 3.2, where the subscript *exp* refers to the experimental data, *emp* refers to the empirical solutions for damping ratio, and *ana* refers to analytical solutions for frequency. In this table and subsequent plots, the average of the two empirical solutions for damping ratio are used.

The CFD simulation was able to match the analytical frequency within 0.31% at all fill levels. There was up to 6.21% difference between the frequency from the CFD solution and the experimental data. Figure 3.16 gives a comparison of the CFD, experimental, and analytical natural frequencies at each fill level. The CFD solution was not within the uncertainty of the experimental values, as uncertainty in the frequency was very low.

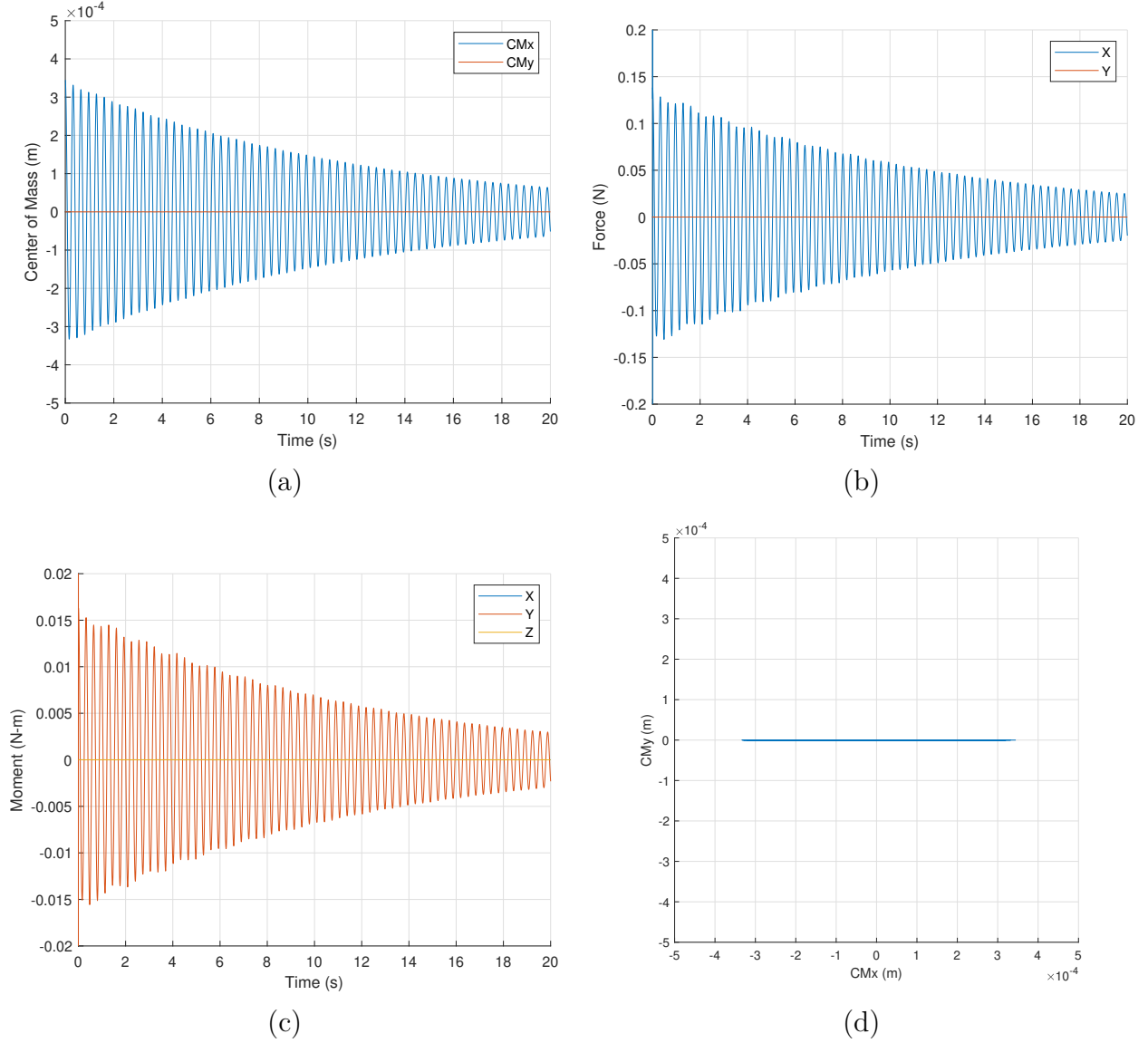
As expected based on previous research, the CFD simulation was unable to match the experimental damping ratio as well as the frequency. There was between 14 and 28.4% difference between the CFD and experimental values at the various fill levels. The CFD simulation better matched the empirical correlations for damping ratio with a maximum difference of 4.32%. These results are shown in Figure 3.17. At one fill level, the CFD predicted the damping ratio with the standard uncertainty.



**Figure 3.16.** Comparison of CFD, experimental, and analytical solutions for frequency in the cylindrical tank at tested fill levels.



**Figure 3.17.** Comparison of CFD, experimental, and average empirical solutions for damping ratio in the cylindrical tank at tested fill levels.



**Figure 3.18.** Plot of (a) center of mass, (b) forces, (c) moments, and (d) top-down center of mass trace for the cylindrical container.

Figure 3.18 shows the CFD results for center of mass, force, and moments on the tank over time, as well as a top-down trace of the center of mass over time. All plots are as expected. More fluctuations due to the second natural frequency can be seen in the plots of forces and moments. This is why the center of mass was used for damping ratio calculations. The top-down trace shows that the center of mass stayed along the x-axis as expected due to the surface initialization in that direction. There was no swirl.

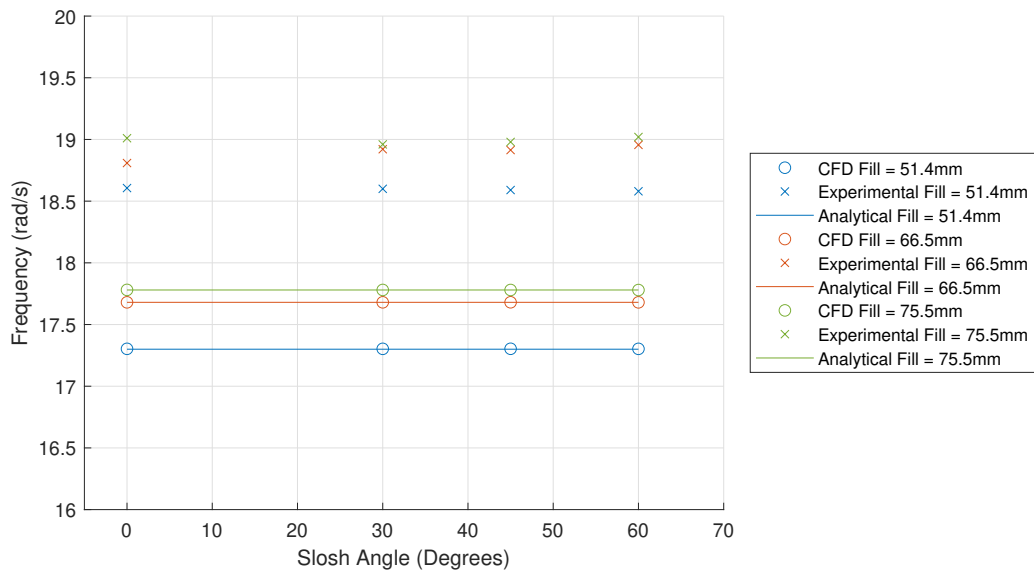
### 3.2.2 Cube

CFD cases of the cube geometry were run at slosh angles of 0, 30, 45, and 60 degrees at each fill level examined in the experiment. The results are summarized in Table 3.3 where the subscript  $m$  refers to parameters calculated using the magnitude of the center of mass offset,  $exp$  refers to the experiment, and  $ana$  refers to the analytical solution.

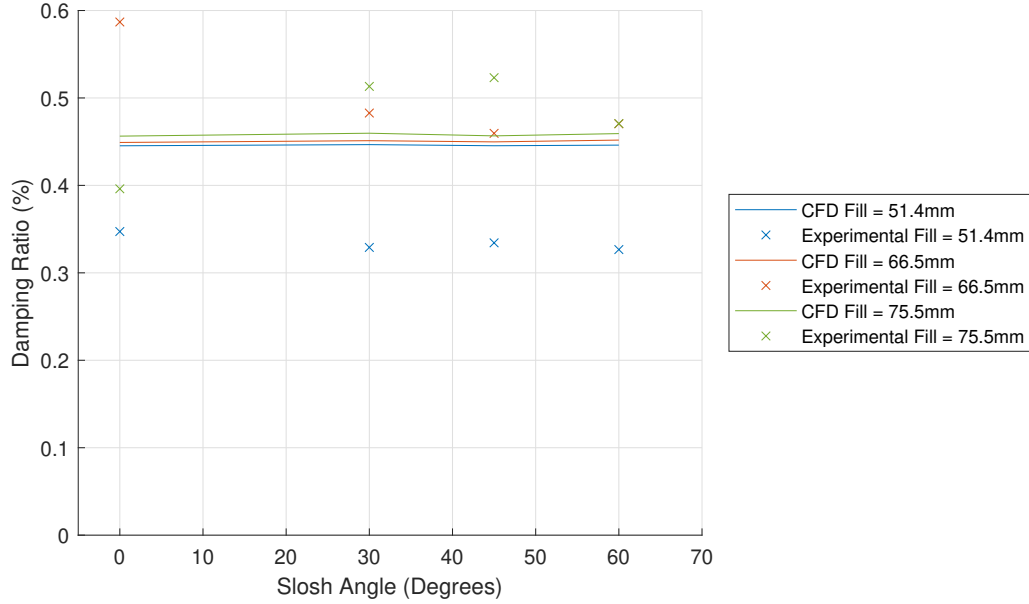
The CFD simulation was able to exactly match the analytical solution for the natural frequency. There was up to a 7% difference between the CFD solution for natural frequency and the experimentally determined natural frequency. Figure 3.19 gives a comparison of the CFD, experimental, and analytical natural frequencies at each slosh angle. As mentioned in the analysis of experimental data, it appears that the experimental natural frequency is simply shifted upward from the CFD and analytical solutions.

**Table 3.3.** CFD results for the cube geometry at various fill fractions and slosh angles.

<b>51.4 mm Fill</b>									
<b>Angle</b>	$\zeta_x$ (%)	$\omega_x$ (rad/s)	$\zeta_y$ (%)	$\omega_y$ (rad/s)	$\zeta_m$ (%)	$\omega_m$ (rad/s)	$\zeta_{exp}$ (%)	$\omega_{exp}$ (rad/s)	$\omega_{ana}$ (rad/s)
<b>0</b>	0.445	17.30	-	-	0.445	17.30	0.345	18.61	17.30
<b>30</b>	0.448	17.30	0.442	17.30	0.447	17.30	0.332	18.60	17.30
<b>45</b>	0.445	17.30	0.446	17.30	0.445	17.30	0.329	18.59	17.30
<b>60</b>	0.441	17.30	0.448	17.30	0.446	17.30	0.323	18.58	17.30
<b>66.5 mm Fill</b>									
<b>Angle</b>	$\zeta_x$ (%)	$\omega_x$ (rad/s)	$\zeta_y$ (%)	$\omega_y$ (rad/s)	$\zeta_m$ (%)	$\omega_m$ (rad/s)	$\zeta_{exp}$ (%)	$\omega_{exp}$ (rad/s)	$\omega_{ana}$ (rad/s)
<b>0</b>	0.449	17.68	-	-	0.449	17.68	0.587	18.81	17.68
<b>30</b>	0.452	17.68	0.448	17.68	0.451	17.68	0.483	18.92	17.68
<b>45</b>	0.449	17.68	0.450	17.68	0.450	17.68	0.460	18.91	17.68
<b>60</b>	0.447	17.68	0.453	17.68	0.452	17.68	0.470	18.96	17.68
<b>75.5 mm Fill</b>									
<b>Angle</b>	$\zeta_x$ (%)	$\omega_x$ (rad/s)	$\zeta_y$ (%)	$\omega_y$ (rad/s)	$\zeta_m$ (%)	$\omega_m$ (rad/s)	$\zeta_{exp}$ (%)	$\omega_{exp}$ (rad/s)	$\omega_{ana}$ (rad/s)
<b>0</b>	0.456	17.78	-	-	0.456	17.78	0.396	19.01	17.78
<b>30</b>	0.461	17.78	0.456	17.78	0.460	17.78	0.513	18.96	17.78
<b>45</b>	0.456	17.78	0.457	17.78	0.457	17.78	0.523	18.98	17.78
<b>60</b>	0.454	17.78	0.461	17.78	0.459	17.78	0.471	19.02	17.78



**Figure 3.19.** Comparison of CFD, experimental, and analytical solutions for natural frequency in a cube container.

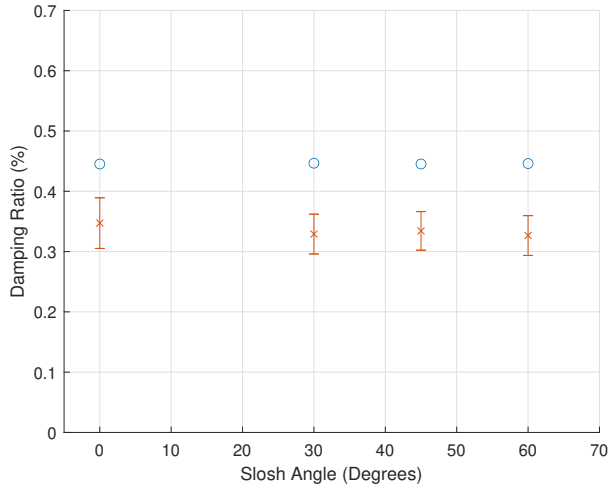


**Figure 3.20.** Comparison of CFD and experimental solutions for damping in a cube container.

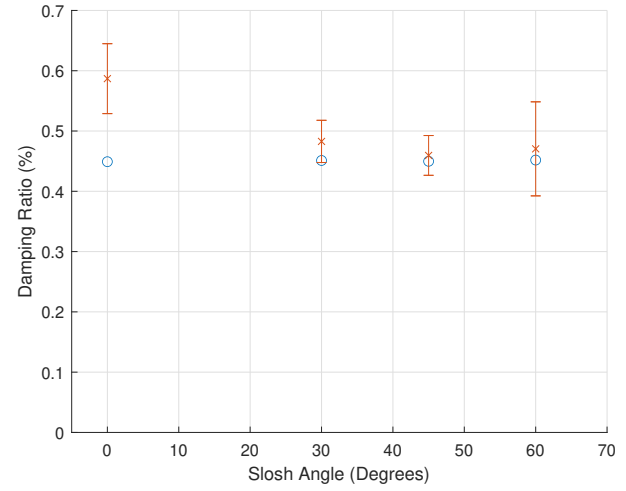
The CFD simulation was not able to predict the damping ratio as closely as it was able to predict the natural frequency as shown in Figure 3.20. This was expected based on the results of previous research. There was up to a 38%, 24%, and 16% error between the CFD and experimental damping ratio at the 51.4mm, 66.5mm, and 75.5mm fills levels respectively. It might be that the CFD simulation is less able to accurately model bottom effects, and therefore the larger error in damping ratio at the low fill fractions.

Uncertainty in the frequency was again very low. A comparison of CFD and experimental damping ratio with error bars is given in Figure 3.21. Other than at the lowest fill fraction, the CFD predicts or nearly predicts the damping ratio to within the standard uncertainty. This again may indicate difficulty modeling bottom effects.

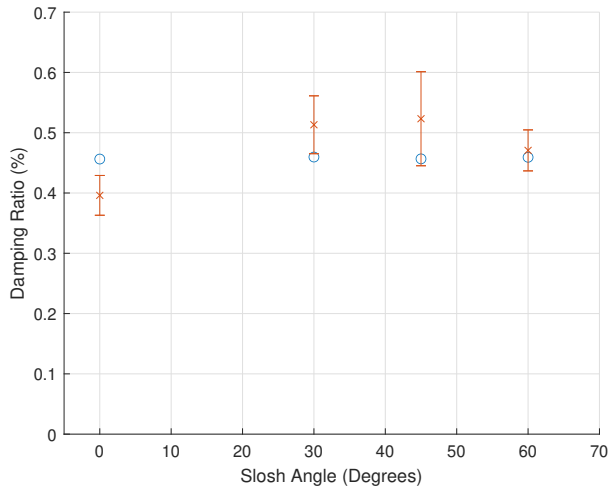
Figure 3.22 gives a comparison of the center of mass over time in the x and y directions for each slosh angle. This is the data which was used to calculate the damping ratio and frequency for each case. The center of mass data also shows that the 30 and 60 degree cases are the same, but with the x and y values switched. The 45 degree case has x and y values which are equal at all times. Both of these results were expected based on experimental results.



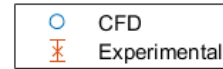
(a)



(b)

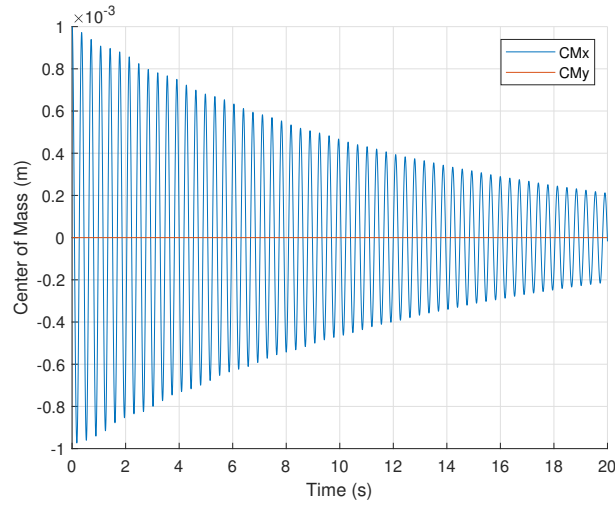


(c)

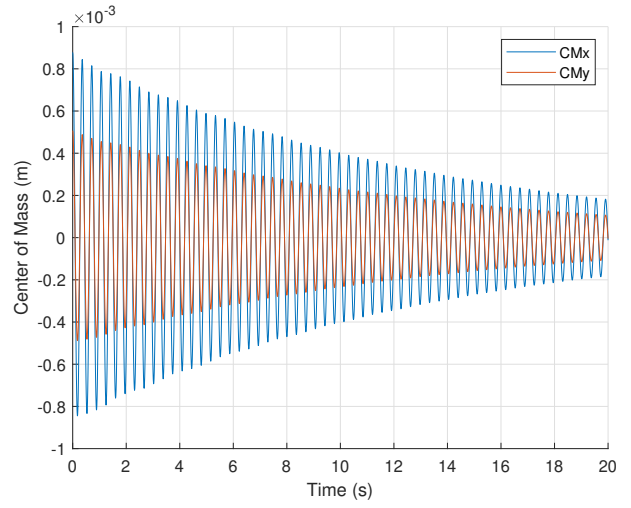


**Figure 3.21.** Comparison of CFD and experimental solutions for damping with error cars at a fill level of (a) 51.4 mm, (b) 66.5 mm, and (c) 75.5 mm.

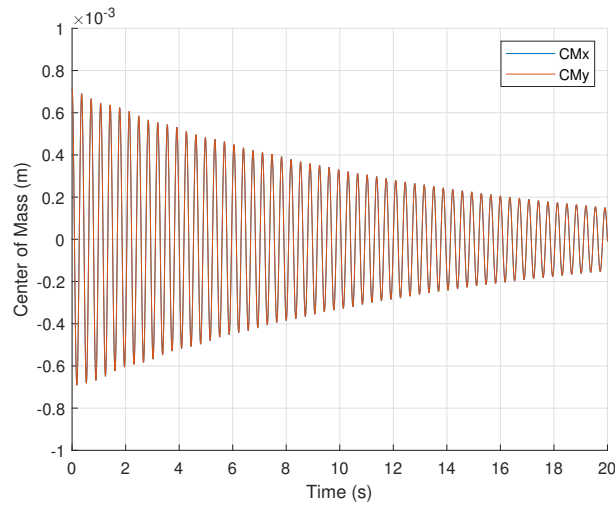




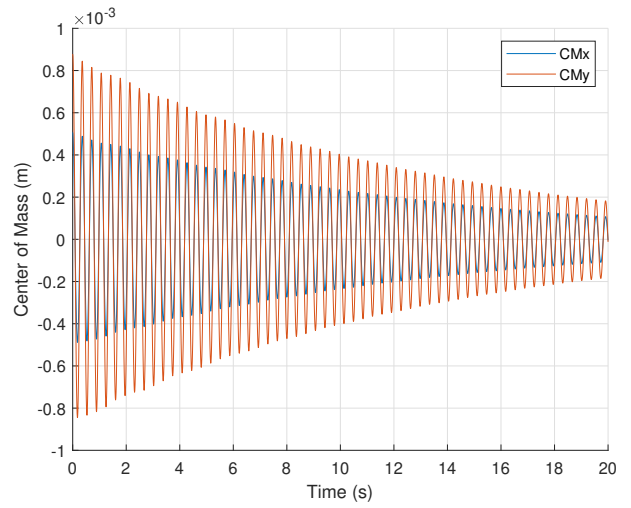
(a)



(b)

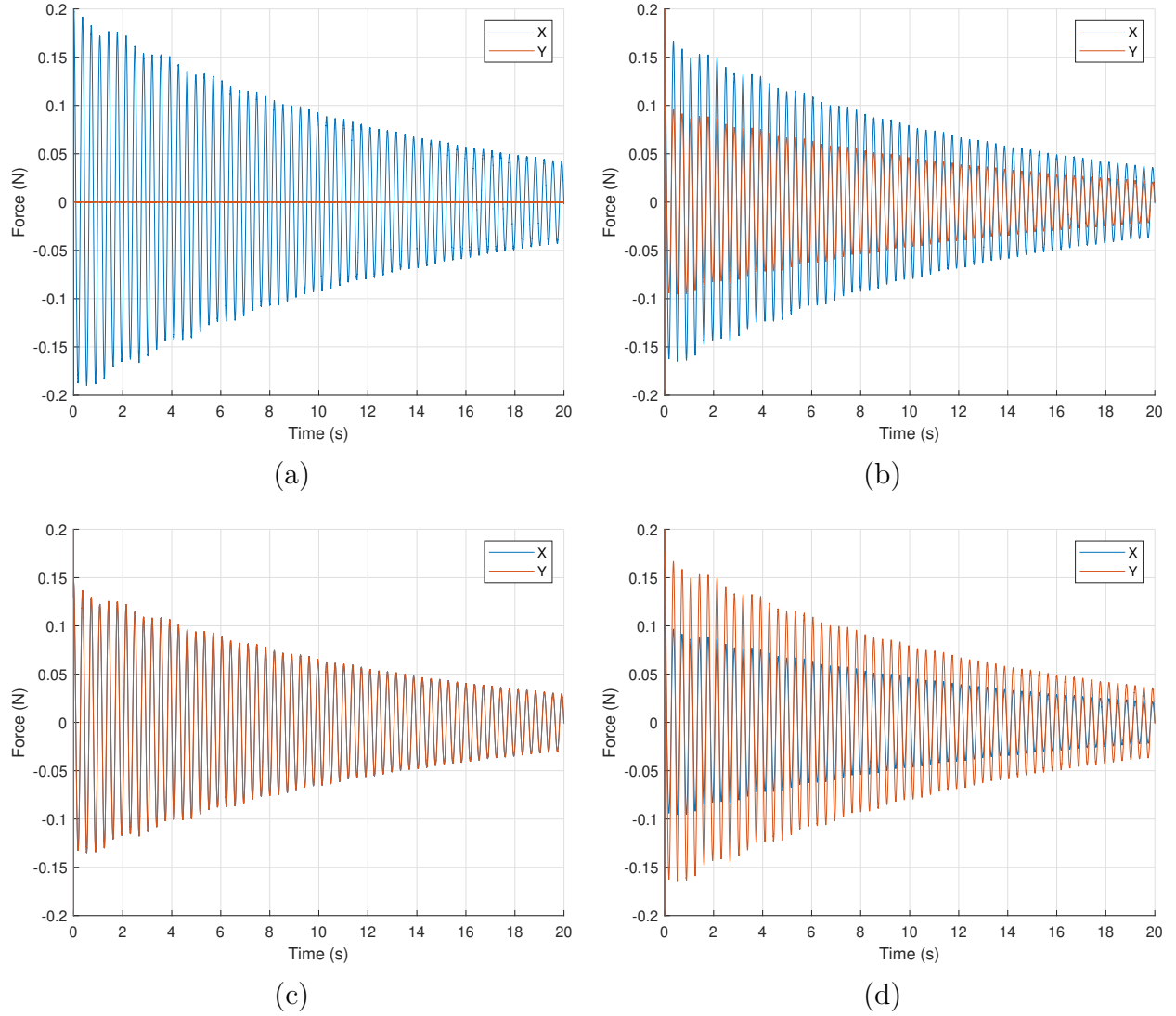


(c)



(d)

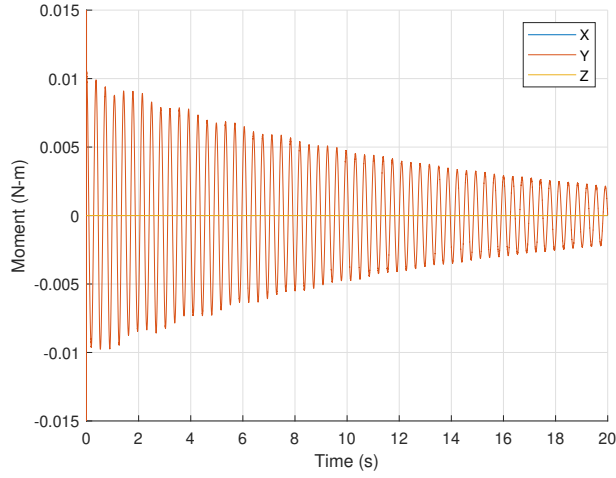
**Figure 3.22.** Center of mass over time in the cube for a slosh angle of (a) 0 degrees, (b) 30 degrees, (c) 45 degrees, and (d) 60 degrees.



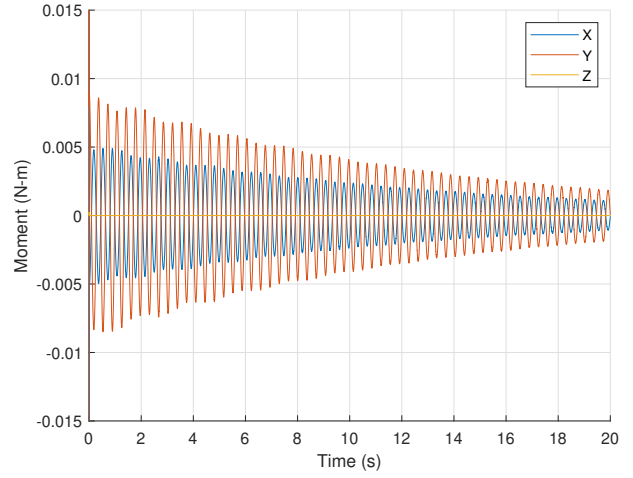
**Figure 3.23.** Force on the walls over time in the cube for a slosh angle of (a) 0 degrees, (b) 30 degrees, (c) 45 degrees, and (d) 60 degrees.

Figures 3.23 shows the forces on the tank and Figure 3.24 shows the moments on the tank at each slosh angle. In both plots, the second natural frequency in the tank can be seen slightly at the beginning of the simulation. This is why the center of mass was used for parameter calculation instead of either forces or moments.

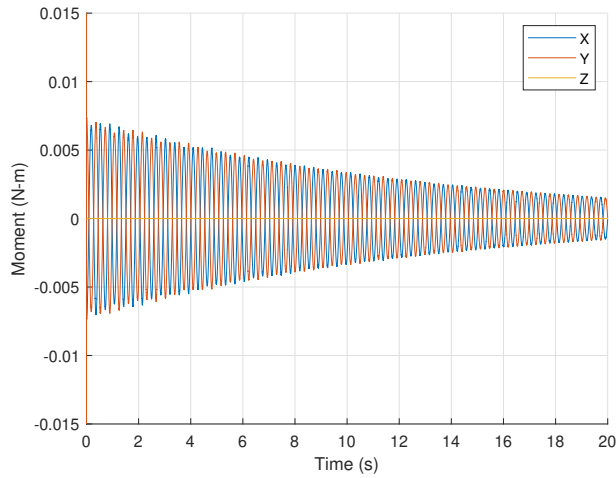
Figure 3.25 gives a top-down trace of the center of mass over the time period investigated. These plots show that the center of mass remained along a single line corresponding to the



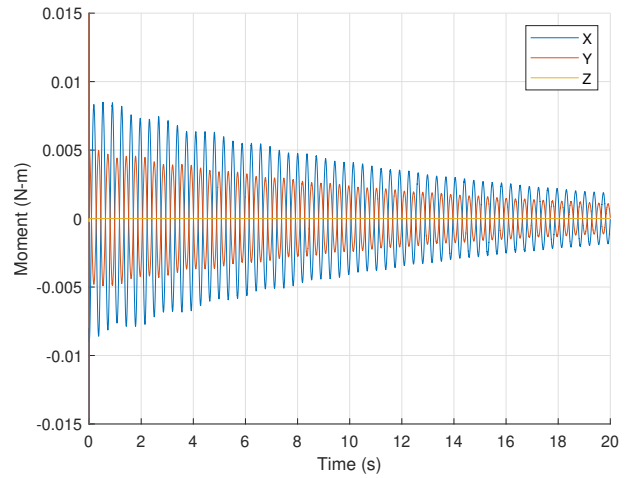
(a)



(b)



(c)



(d)

**Figure 3.24.** Moment on the tank over time in the cube for a slosch angle of (a) 0 degrees, (b) 30 degrees, (c) 45 degrees, and (d) 60 degrees.

**Table 3.4.** Summary of results for the rectangular tank at all slosh angles and fill levels.

<b>Slosh Angle</b>	<b>Mode 1</b>					<b>Mode 2</b>				
	$\zeta_{CFD}$ (%)	$\zeta_{exp}$ (%)	$\omega_{CFD}$ (rad/s)	$\omega_{exp}$ (rad/s)	$\omega_{ana}$ (rad/s)	$\zeta_{CFD}$ (%)	$\zeta_{exp}$ (%)	$\omega_{CFD}$ (rad/s)	$\omega_{exp}$ (rad/s)	$\omega_{ana}$ (rad/s)
<b>80.8 mm Fill</b>										
<b>0</b>	-	-	-	-	-	0.417	0.331	14.00	14.76	14.04
<b>30</b>	0.396	0.332	17.85	18.93	17.84	0.417	0.330	14.00	14.80	14.04
<b>45</b>	0.393	0.275	17.85	18.94	17.84	0.419	0.293	14.01	14.80	14.04
<b>60</b>	0.392	0.313	17.85	18.94	17.84	0.422	0.327	14.00	14.81	14.04
<b>90</b>	0.393	0.306	17.85	18.95	17.84	-	-	-	-	-
<b>70.2 mm Fill</b>										
<b>0</b>	-	-	-	-	-	0.396	0.418	13.76	14.51	13.79
<b>30</b>	0.381	0.577	17.77	18.81	17.75	0.396	0.449	13.76	14.50	13.79
<b>45</b>	0.378	0.547	17.77	18.81	17.75	0.398	0.438	13.76	14.51	13.79
<b>60</b>	0.375	0.528	17.77	18.80	17.75	0.401	0.427	13.76	14.51	13.79
<b>90</b>	0.374	0.465	17.76	18.83	17.75	-	-	-	-	-
<b>59.8 mm Fill</b>										
<b>0</b>	-	-	-	-	-	0.399	0.380	13.39	14.16	13.41
<b>30</b>	0.374	0.373	17.60	18.70	17.57	0.399	0.415	13.39	14.17	13.41
<b>45</b>	0.372	0.348	17.60	18.70	17.57	0.401	0.398	13.39	14.17	13.41
<b>60</b>	0.368	0.344	17.60	18.70	17.57	0.405	0.375	13.39	14.17	13.41
<b>90</b>	0.368	0.353	17.59	18.70	17.57	-	-	-	-	-

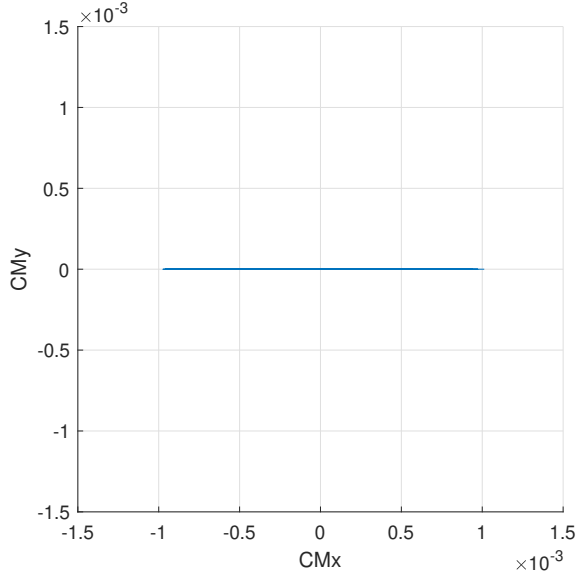
slosh angle. There was very slight variation at 30 and 60 degrees, but the displacement from the line was very small.

### 3.2.3 Rectangular Prism

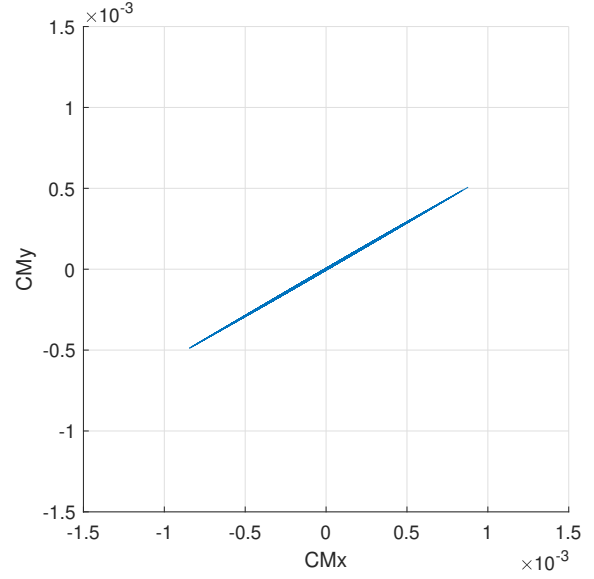
CFD cases of the rectangular geometry were run at slosh angles of 0, 30, 45, 60, and 90 degrees and at every fill level examined in the experiment. The results are summarized in Table 3.4.

The CFD simulations were able to match the analytical solutions for frequency within 0.27% for both slosh modes. The CFD simulation was also within 5.91% of the experimental value for natural frequency. A comparison of the three solutions are given in Figure 3.26 and Figure 3.27.

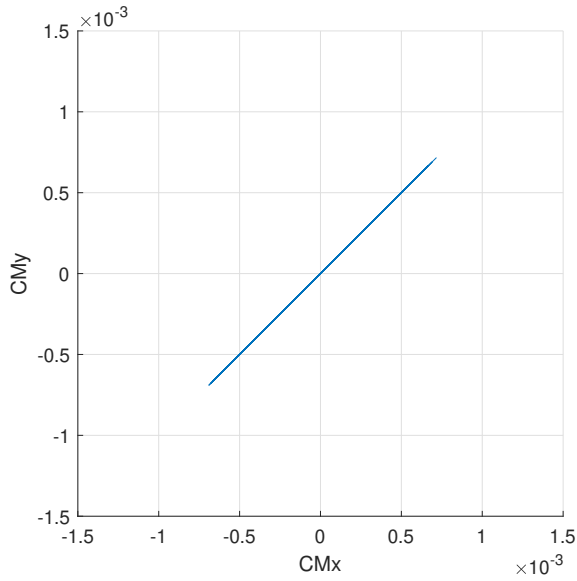
Again, damping ratio did not match as well as the frequency. For mode one, the damping ratio was between 0.25 and 43.2% of the experimental values. For mode two, it was between



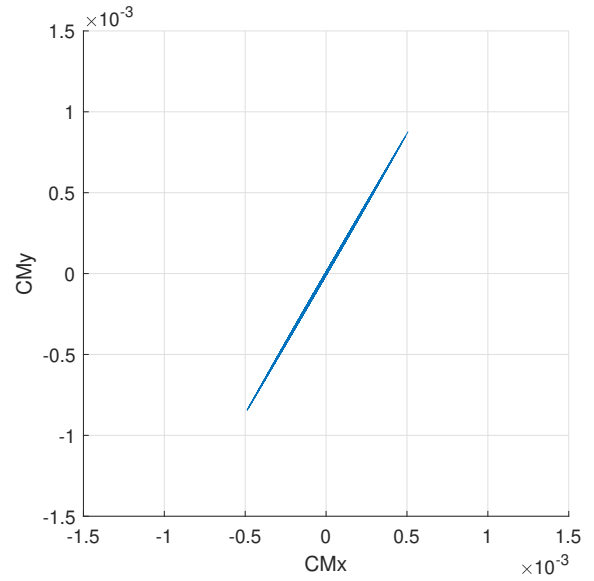
(a)



(b)

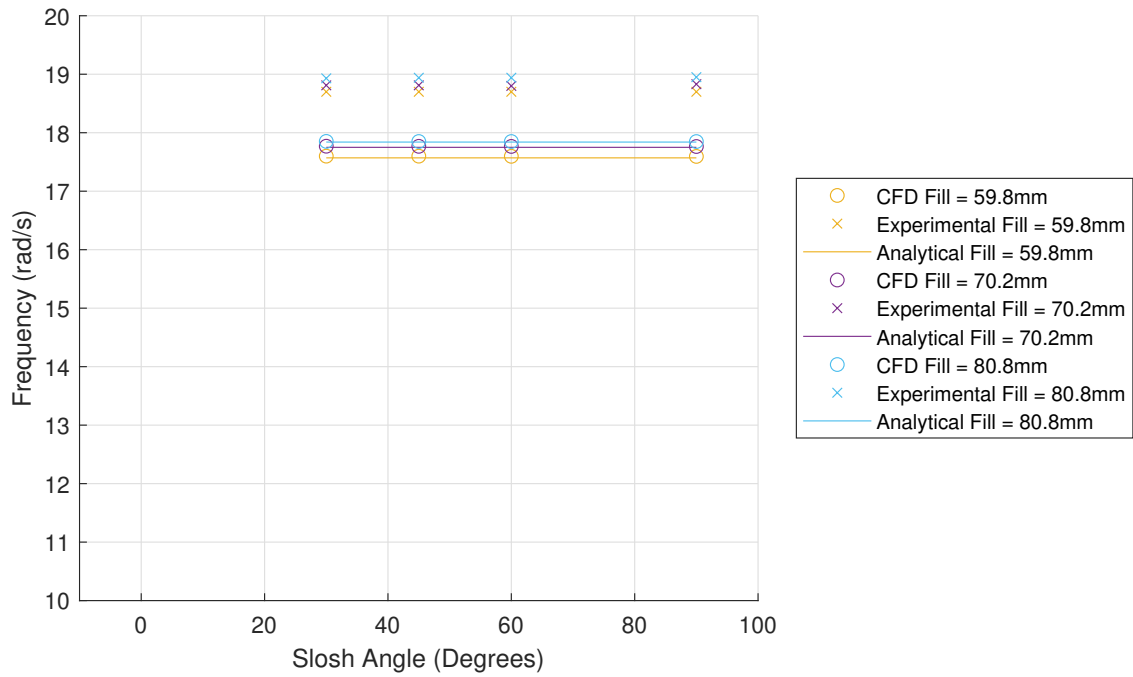


(c)

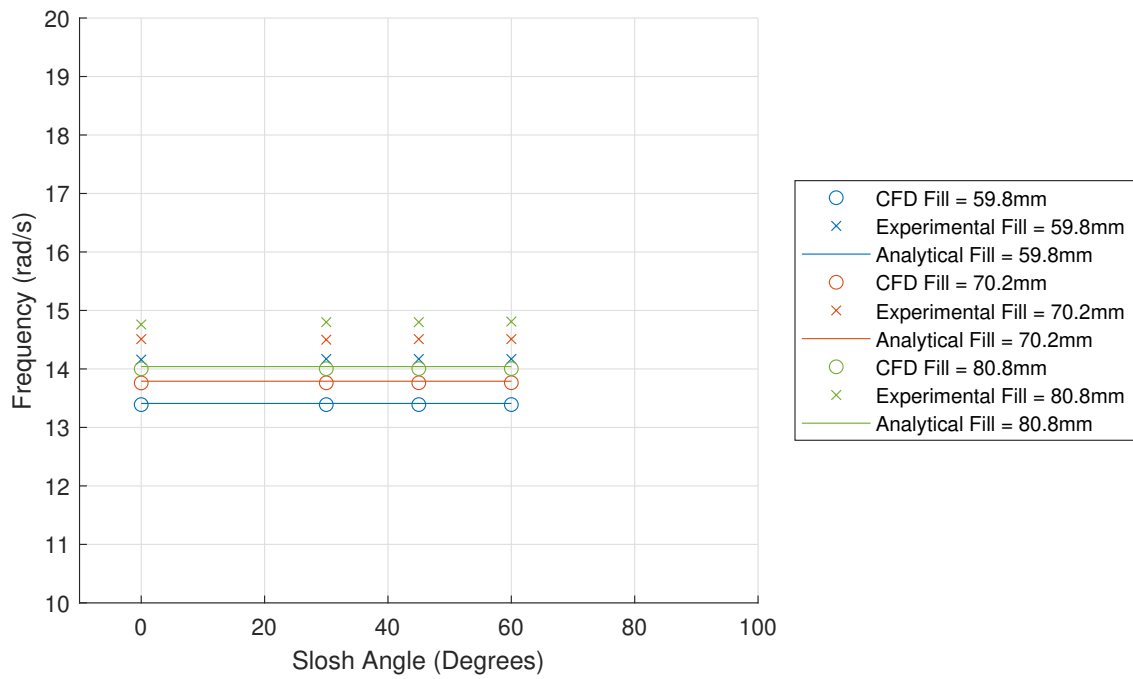


(d)

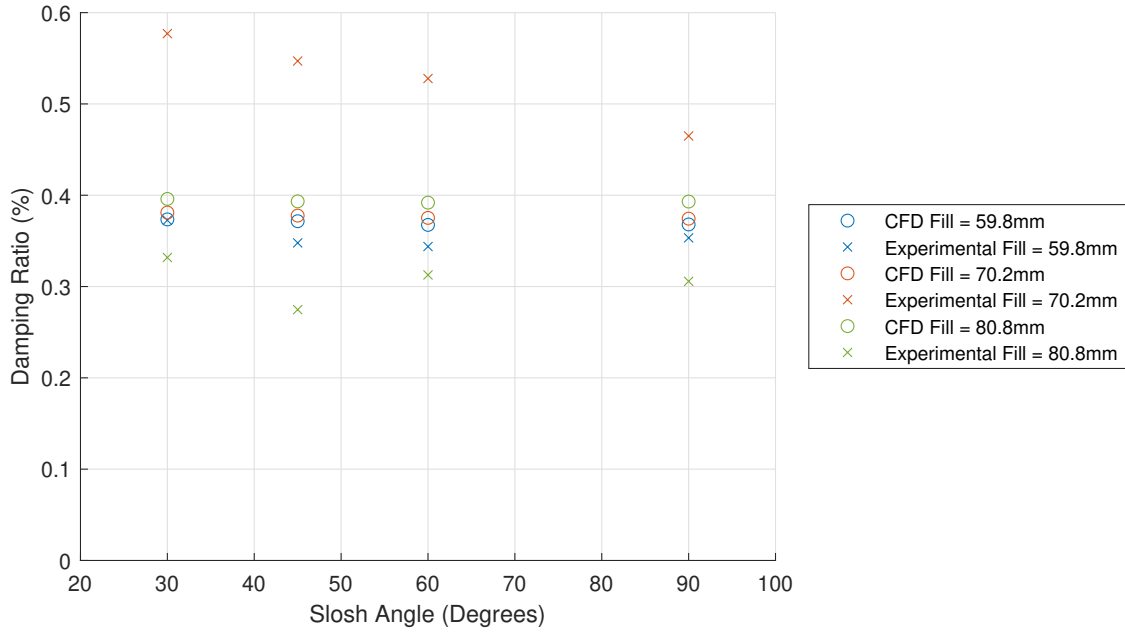
**Figure 3.25.** Top-down trace of the center of mass over time in the cube for a slosch angle of (a) 0 degrees, (b) 30 degrees, (c) 45 degrees, and (d) 60 degrees.



**Figure 3.26.** Comparison of CFD, experimental, and analytical solutions for mode one natural frequency in a rectangular container.



**Figure 3.27.** Comparison of CFD, experimental, and analytical solutions for mode two natural frequency in a rectangular container.



**Figure 3.28.** Comparison of CFD and experimental solutions for mode one damping ratio in a rectangular container.

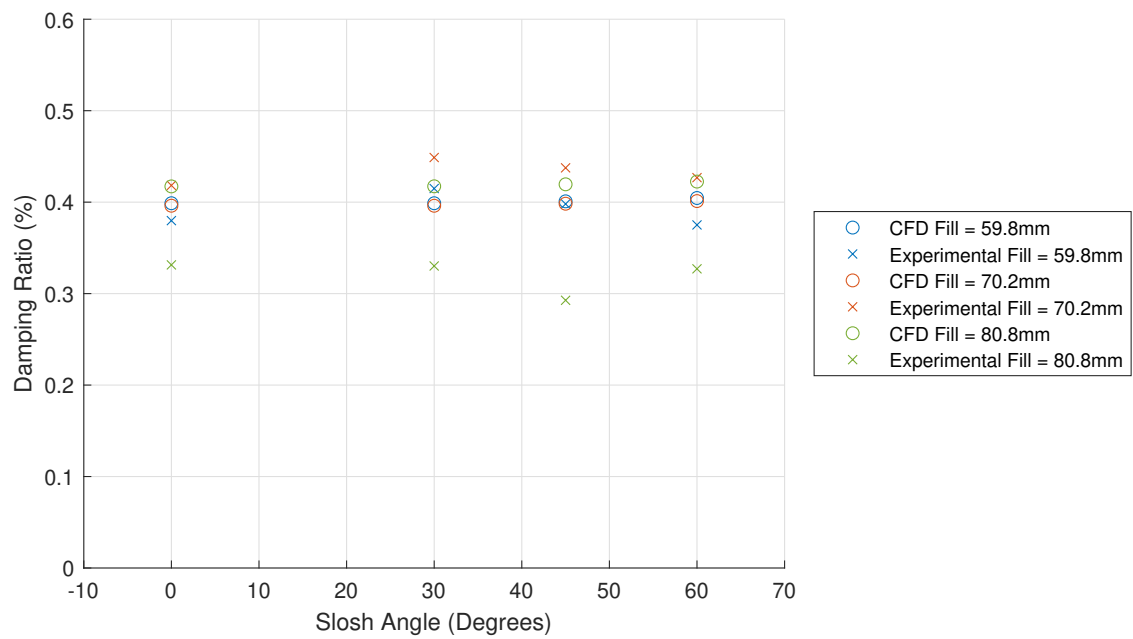
0.73 and 43.3%. A comparison of the CFD and experimental results are given in Figure 3.28 and Figure 3.29 for modes one and two respectively.

Figure 3.30 gives a comparison of the CFD and experimental values for damping ratio at each fill fraction. While the CFD did not always predict the damping ratio within the standard uncertainty, it was typically very close.

Figure 3.31 shows plots of the center of mass over time at each tested slosh angle. Similarly to the cube geometry, the increasing ratio of the y displacement to the x displacement can be seen with increasing slosh angles.

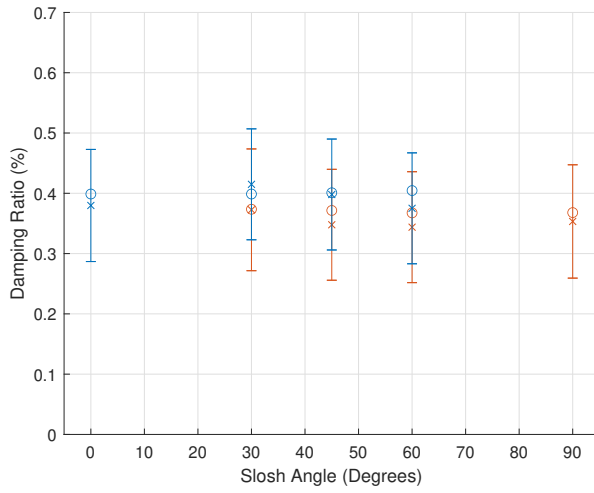
Figure 3.32 shows the forces on the tank and Figure 3.33 shows the moments on the tank at each slosh angle. Again, the second natural frequency may be seen in each of these plots and is the reason why center of mass is used for parameter calculation.

Figure 3.34 gives a top-down trace of the center of mass over time. These plots are not simple lines corresponding to the slosh angle as was found in both the cylinder and cube cases. This is expected because the natural frequency along the two coordinate axes differ.

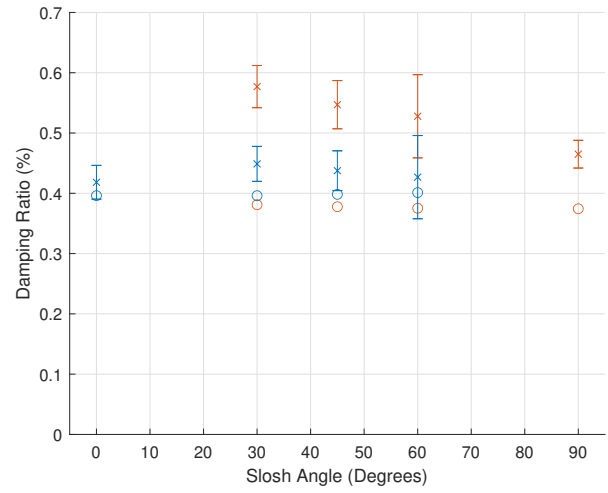


**Figure 3.29.** Comparison of CFD and experimental solutions for mode two damping ratio in a rectangular container.

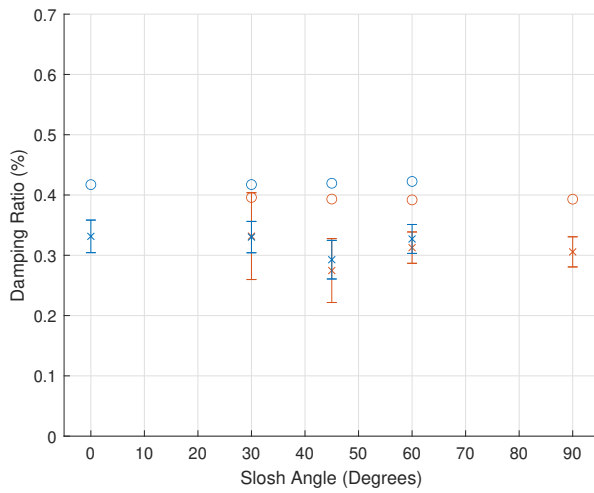




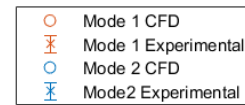
(a)



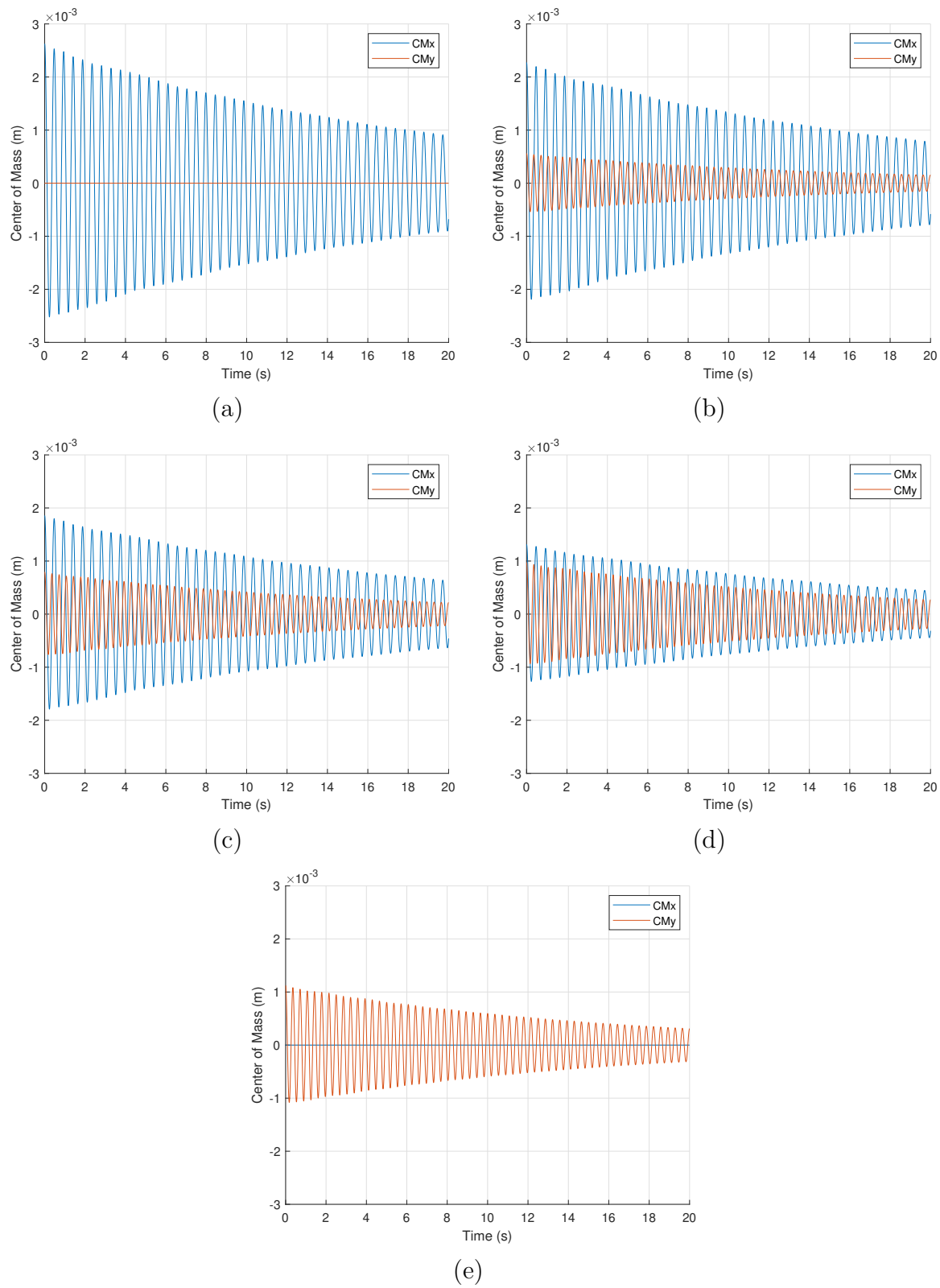
(b)



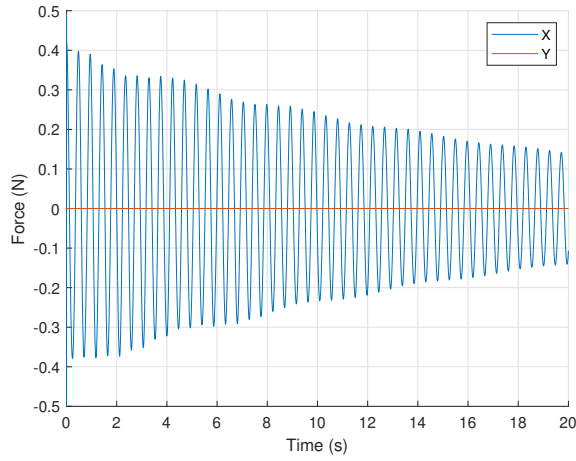
(c)



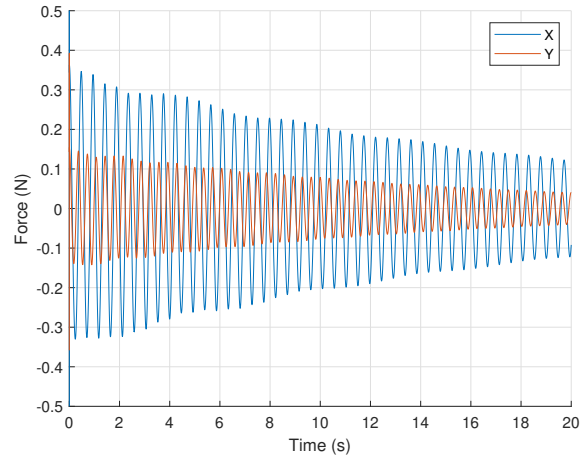
**Figure 3.30.** Comparison of CFD and experimental solutions for damping with error bars at a fill level of (a) 59.8 mm, (b) 70.2 mm, and (c) 80.8 mm.



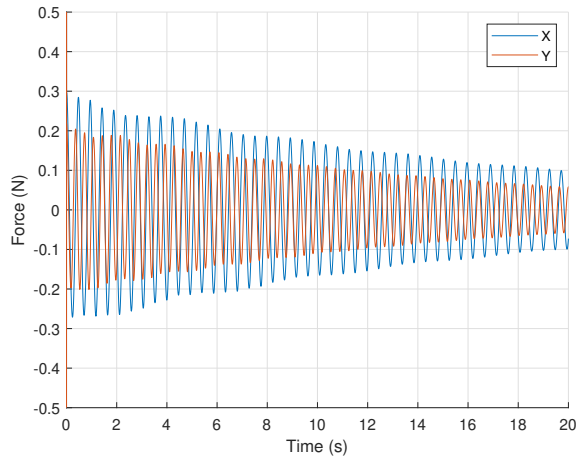
**Figure 3.31.** Center of mass over time in the rectangular prism for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.



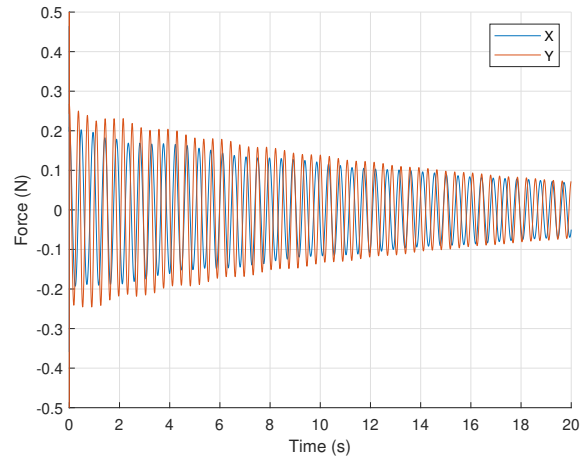
(a)



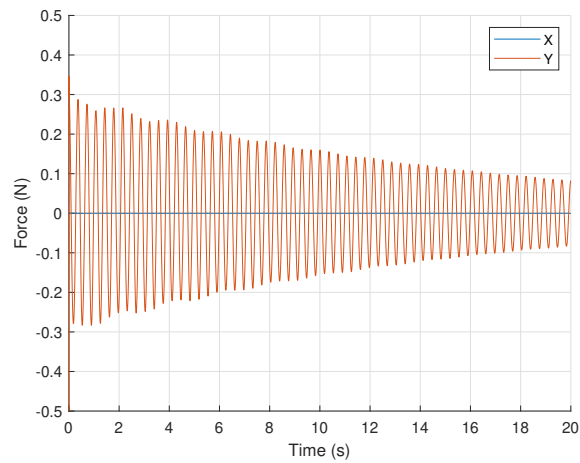
(b)



(c)

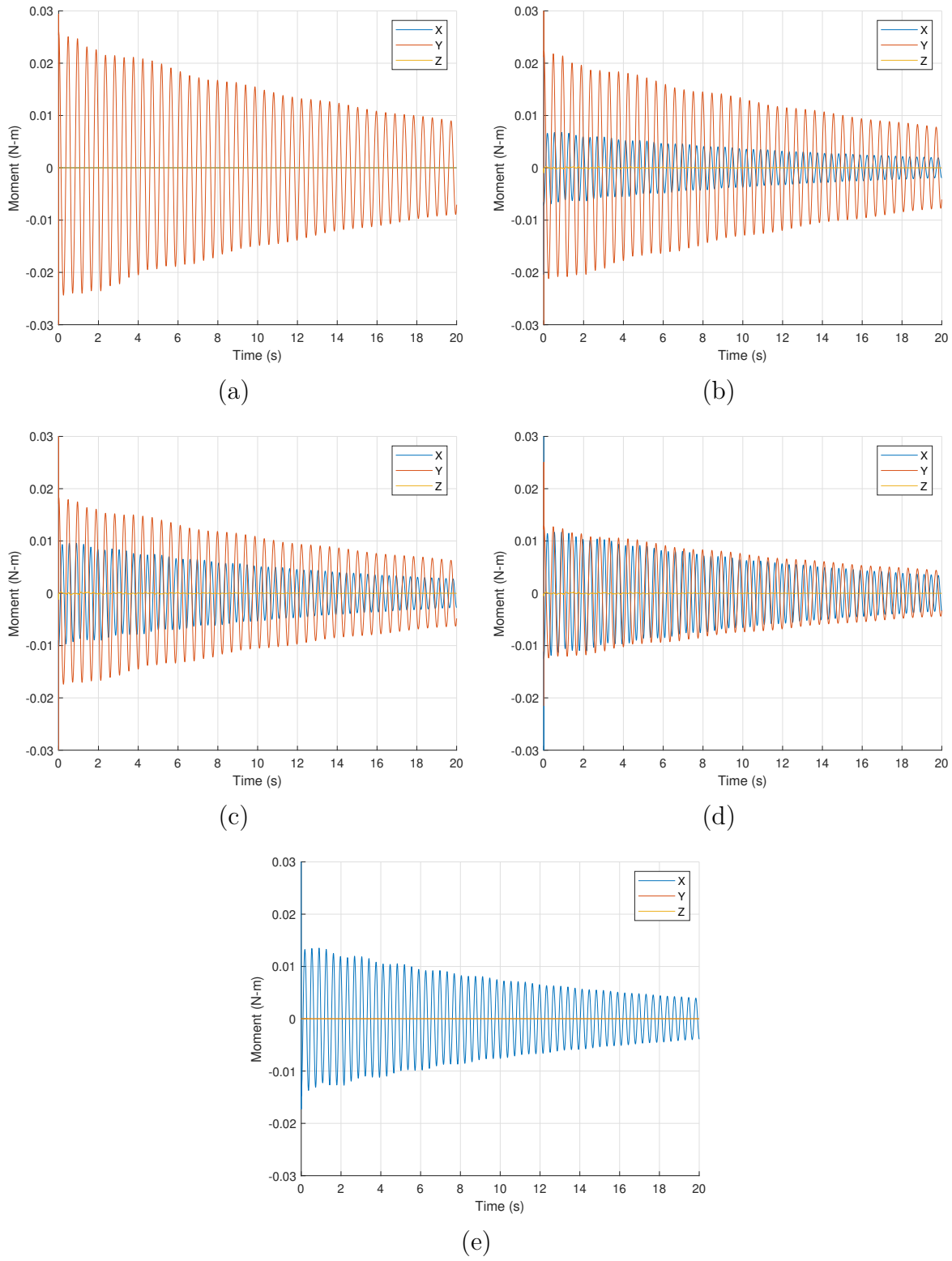


(d)



(e)

**Figure 3.32.** Force on the walls over time in the rectangular prism for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.



**Figure 3.33.** Moment on the tank over time in the rectangular prism for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.

The liquid motion within the rectangular tank is more complicated than the motion in either a cube or cylinder.

### 3.2.4 Stair Step

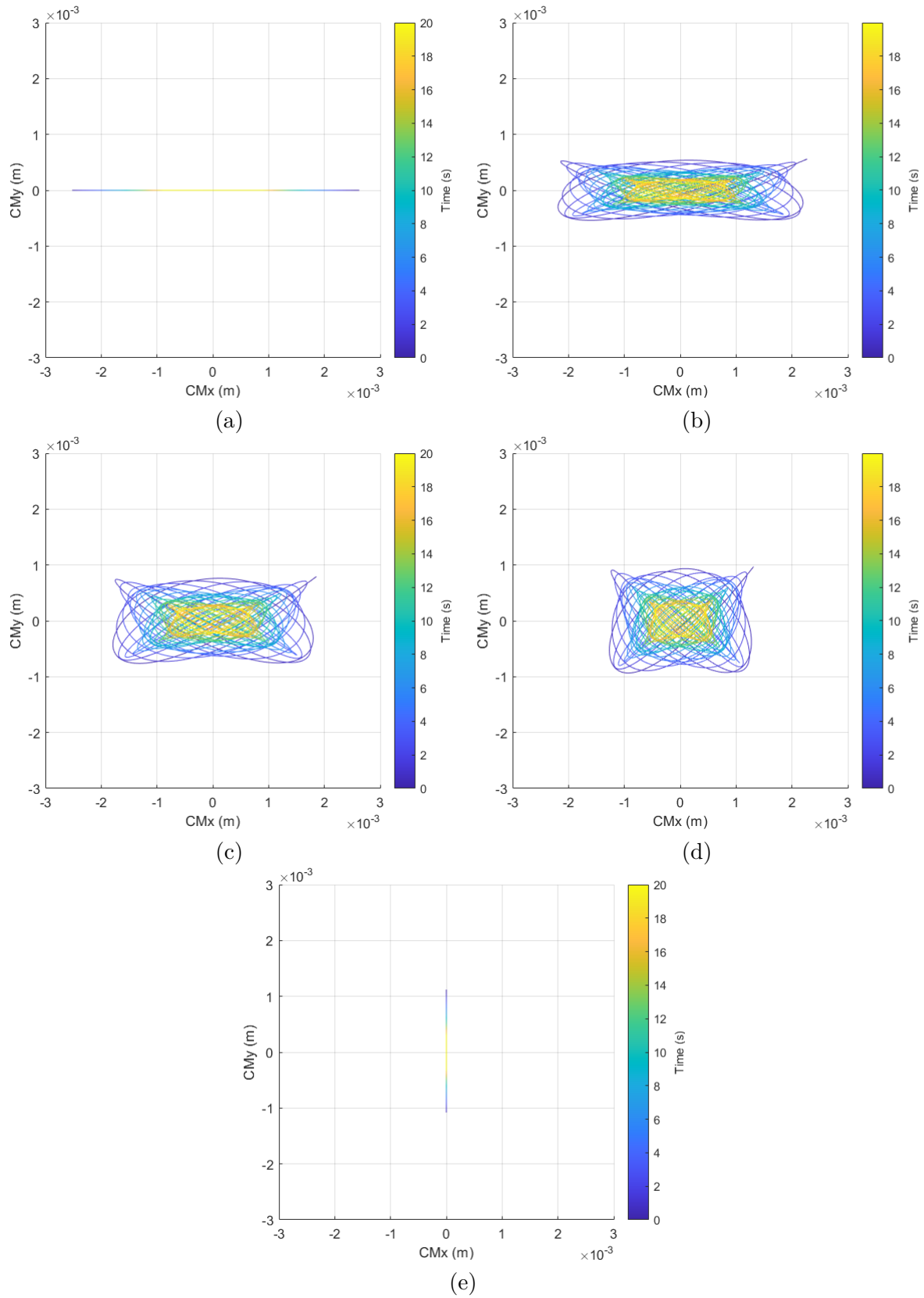
CFD cases of the stair step geometry were run at slosh angles of 0, 30, 45, 60, and 90 degrees at the fill level examined in the experiment. The results are summarized in Table 3.5.

**Table 3.5.** Results of CFD simulations for the stair step tank.

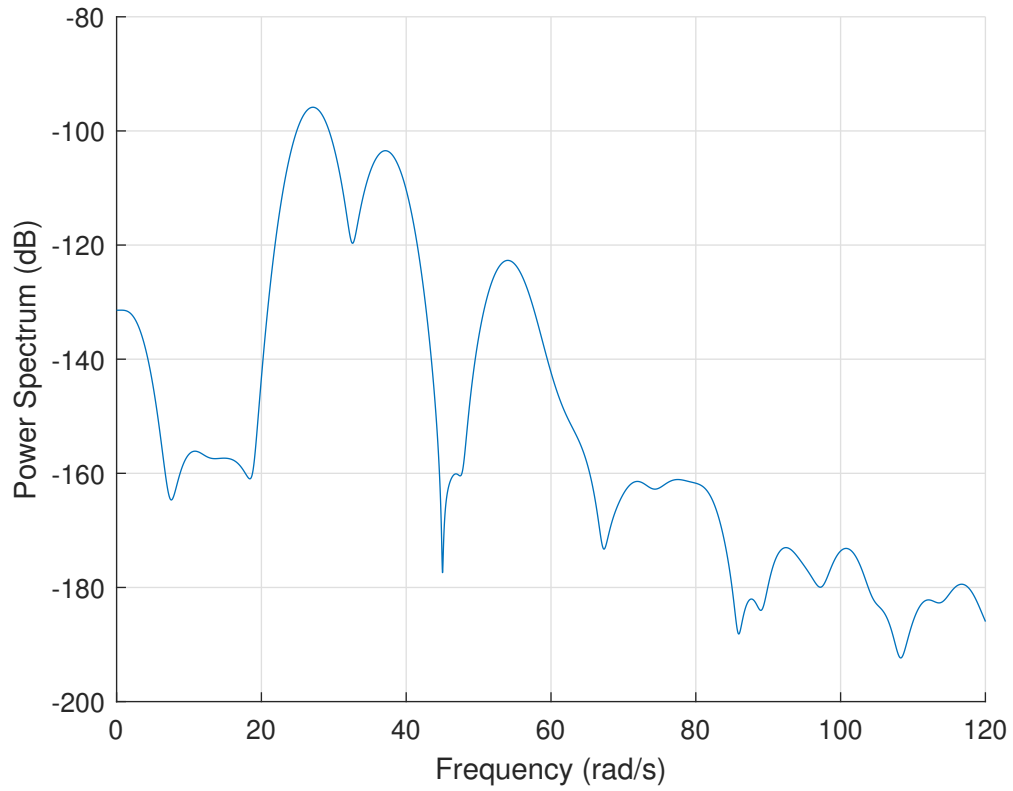
	<b>Mode 1</b>		<b>Mode 2</b>	
<b>Slosh Angle</b>	$\zeta$ (%)	$\omega_n$ (rad/s)	$\zeta$ (%)	$\omega_n$ (rad/s)
<b>0</b>	1.92	27.1	1.56	37.1
<b>30</b>	1.80	27.1	1.59	37.2
<b>45</b>	-	-	1.56	37.1
<b>60</b>	1.92	27.1	1.56	37.1
<b>90</b>	1.89	27.1	1.57	37.2

The power spectral density of the CFD results are shown in Figure 3.35. The two most dominant frequencies visible in the CFD simulations are around 27.1 and 37.1 rad/s. This is different than the experimental results which has dominant frequencies at 39.0 and 52.4 rad/s. While the two most dominant frequencies are different, both results have peaks at similar locations in their power spectral density, even if the peaks are less prominent.

Table 3.6 summarizes four of the main frequencies visible in both the CFD and experimental results. These are compared to the first and second natural frequencies along both the longest and shortest directions in the tank. There is less than a 5% difference between the CFD and experimental values over all four frequencies. There is up to a 20% difference between the analytical and CFD solutions and up to a 19% difference between the analytical and experimental solutions. Based on this analysis, it appears that the two dominant frequencies in the CFD results correspond to the first natural frequency of slosh in both the shortest and longest direction. Because the values are not as close to the analytical as simpler geometries, it's likely that the greater geometrical complexity did shift the natural frequencies slightly.



**Figure 3.34.** Top-down trace of the center of mass over time in the rectangular prism for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.



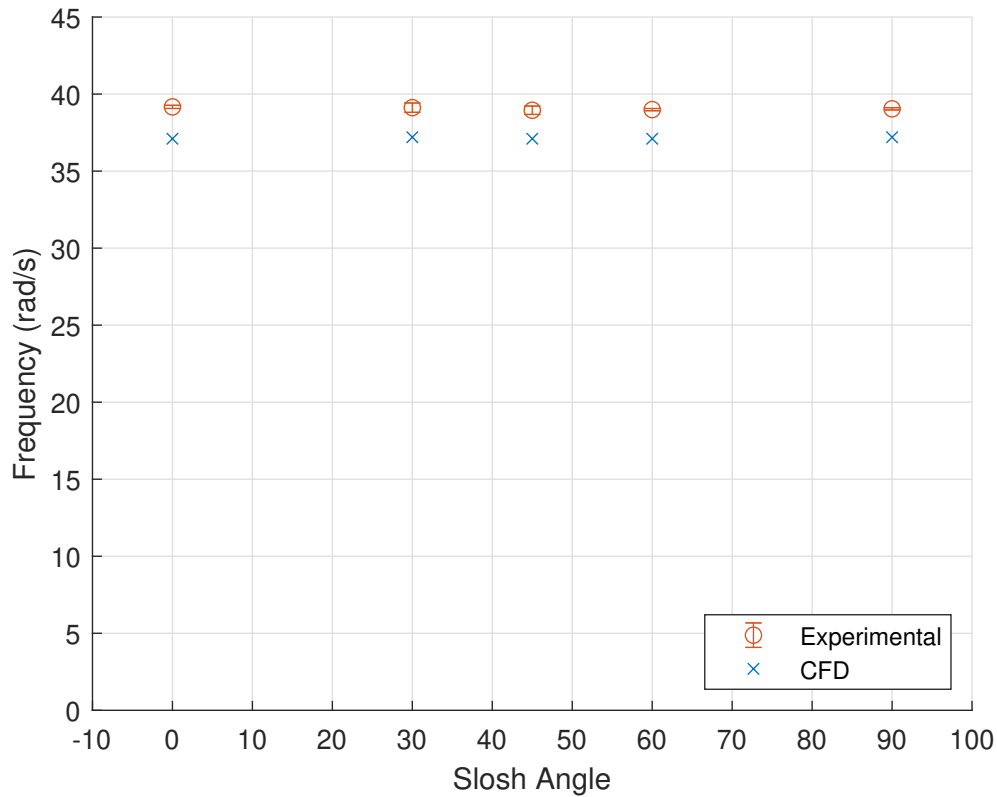
**Figure 3.35.** Power spectral density of the CFD solution for the stair step tank at a zero degree slosh angle.

**Table 3.6.** Identifiable frequencies in the stair step tank.

Analytical (rad/s)	CFD (rad/s)	Experimental (rad/s)
31.8	27.1	28.0
46.4	37.1	39.0
57.1	54.0	52.4
80.4	77.5	77.4

It is more difficult to compare the damping ratios in the CFD and experimental solutions because of the different dominant modes. However, both share a dominant frequency near 38 rad/s. There is between a 0.1 and 53.9% difference between the damping ratio in the experimental and CFD solutions at this frequency. It is unclear why there are different dominant frequencies.

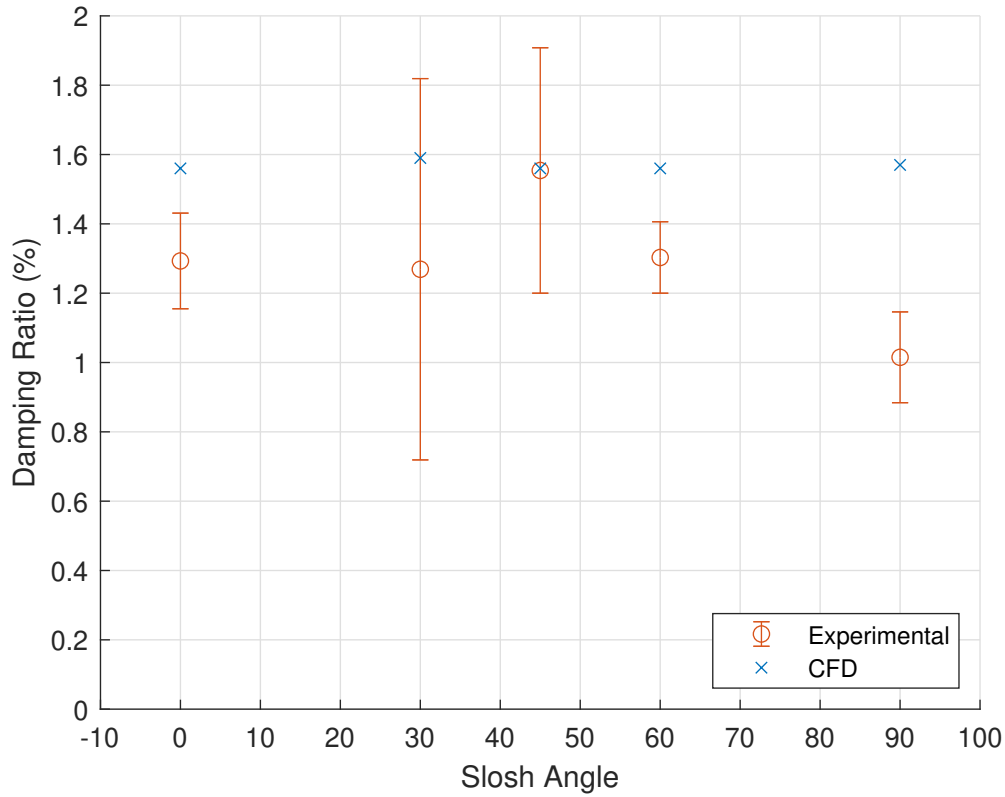
Figures 3.36 and 3.37 compare the CFD and experimental values for damping ratio and frequency in the overlapping slosh mode. Similarly to the basic geometries, the uncertainty in the frequency is very low. Uncertainty in the damping ratio was larger and the CFD model was able to predict the damping ratio within the uncertainty for two slosh angles.



**Figure 3.36.** Comparison of overlapping frequency between CFD and experimental data in the stair step tank.

Figure 3.38 shows plots of the center of mass over time at each tested slosh angle. Unlike the simple geometries where only one frequency is visible in each direction, the results for





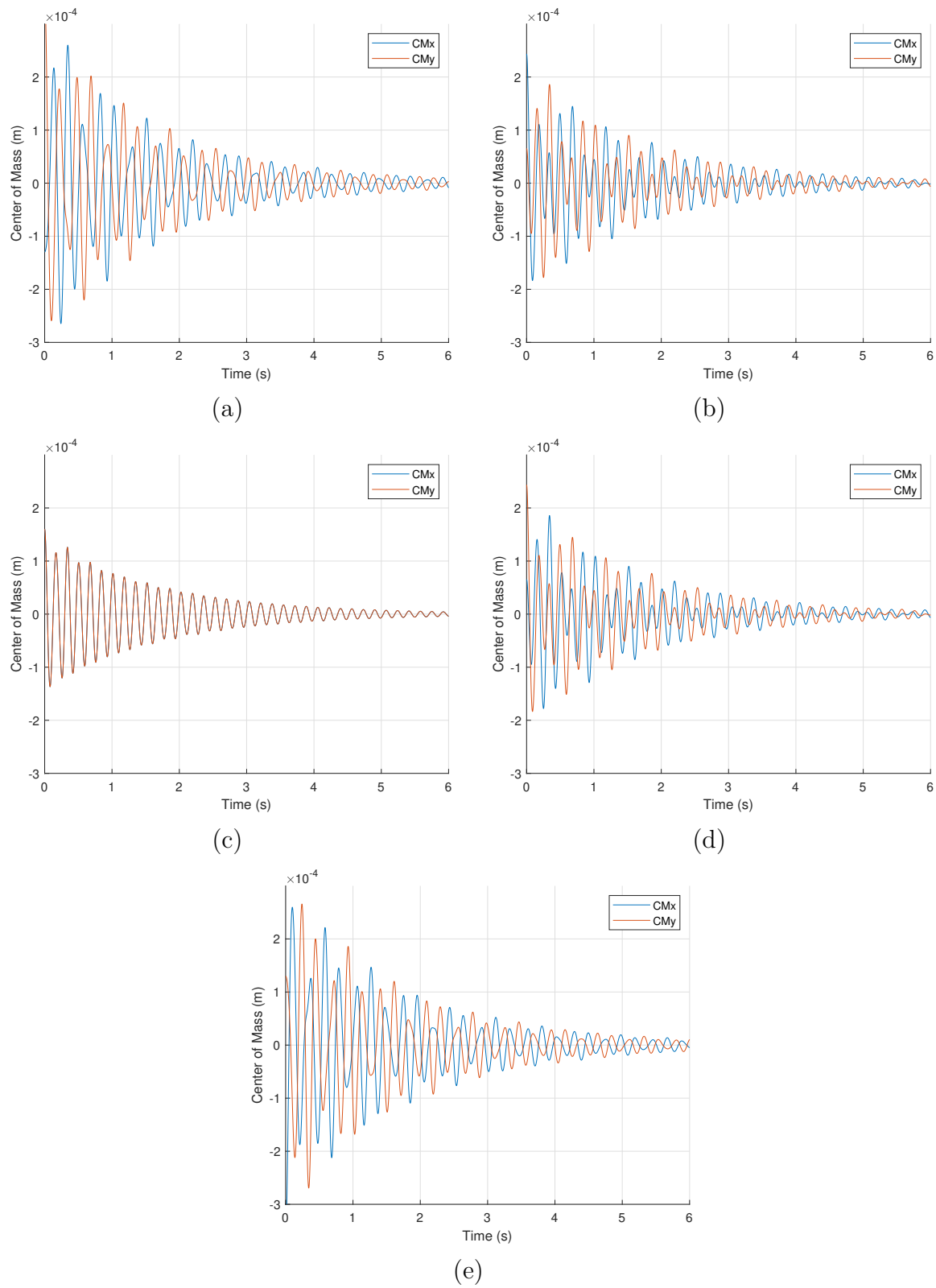
**Figure 3.37.** Comparison of overlapping damping ratio between CFD and experimental data in the stair step tank.

the stair step tank have the appearance of the sum of at least two damped sinusoids. This is because there is more than one dimension in each direction.

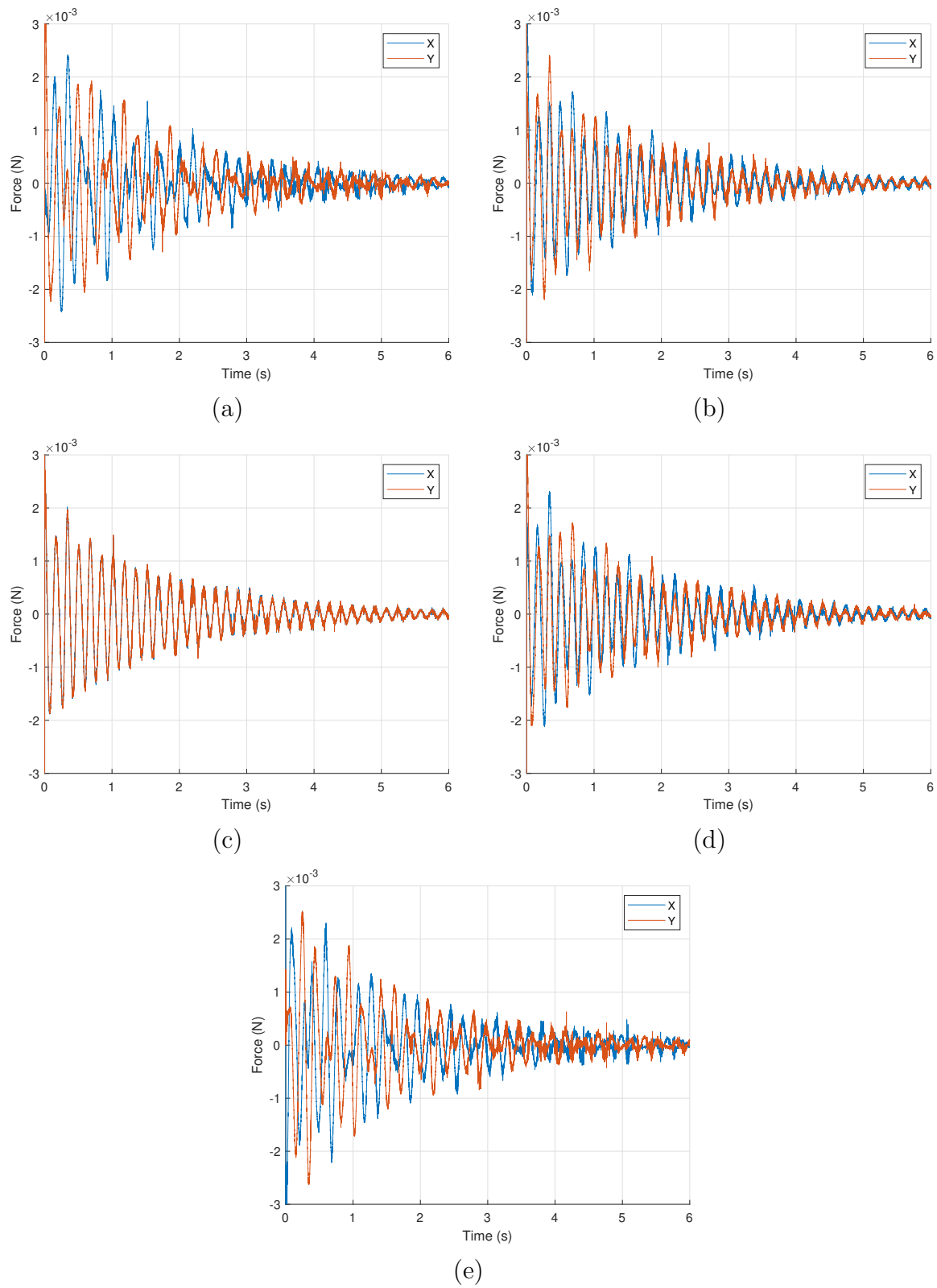
The forces and moments over time are shown in Figures 3.39 and 3.40 respectively. These results appear less "clean" than the basic geometries. This could be due to more complicated liquid motion and the smaller time steps due to this complex motion.

The top-down trace of the center of mass over time is given in Figure 3.41. Even at zero and ninety-degree slosh angles, the trace of the center of mass appears to be slanted. This makes sense because the geometry's line of symmetry is at a forty-five degree angle. Accordingly, the center of mass trace at a forty-five degree slosh angle remains along this line of symmetry.

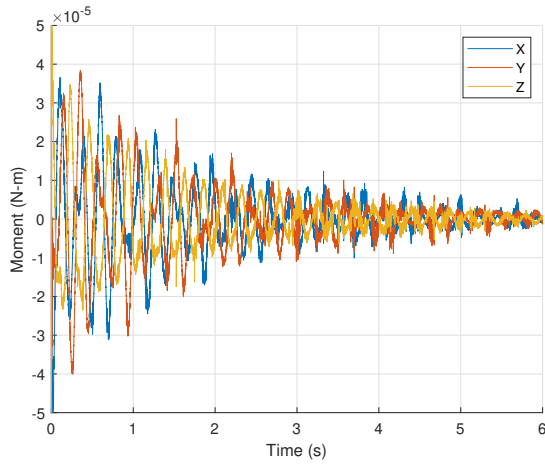
Using curve fits to the CFD data in each direction, the data can be split into a wave corresponding to each frequency. When each of these modes are plotted, the result is that



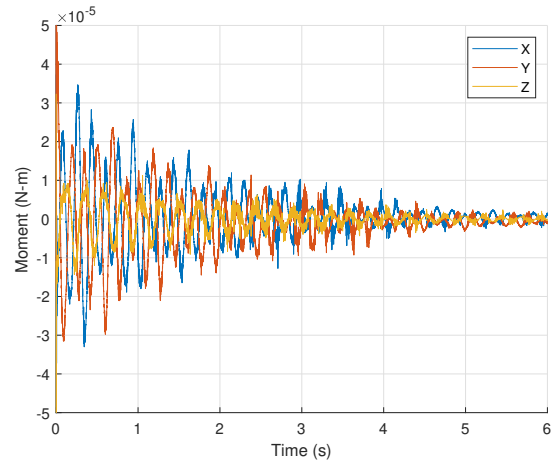
**Figure 3.38.** Center of mass over time in the stair step tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.



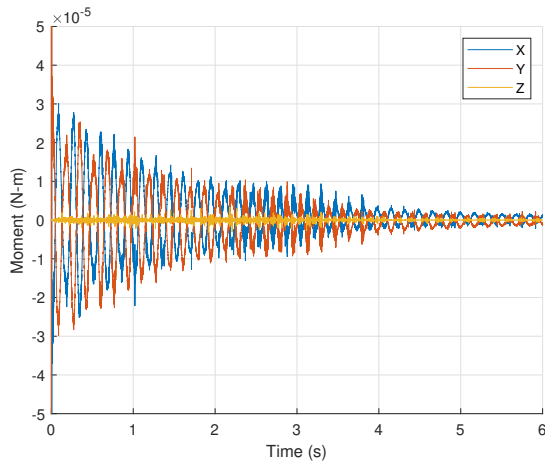
**Figure 3.39.** Force on the walls over time in the stair step tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.



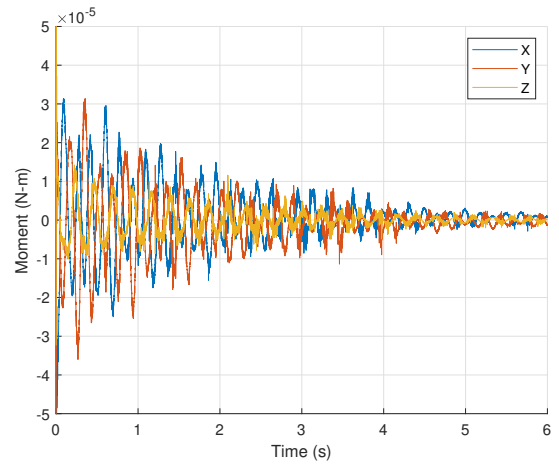
(a)



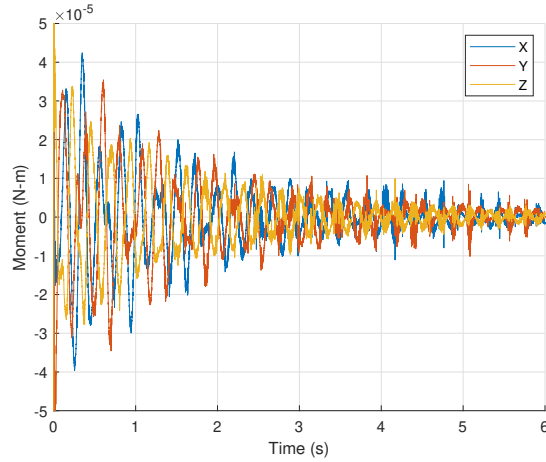
(b)



(c)

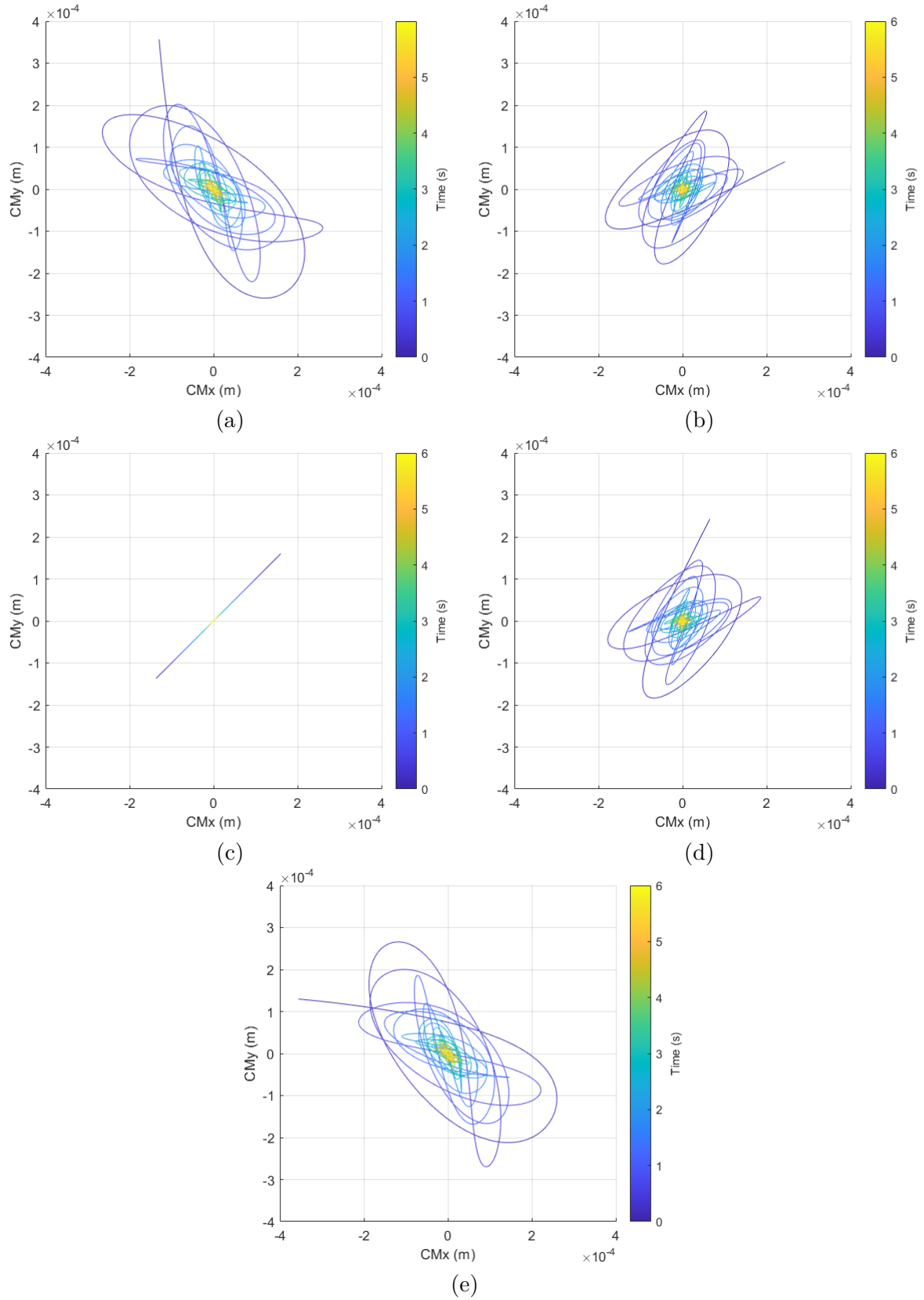


(d)



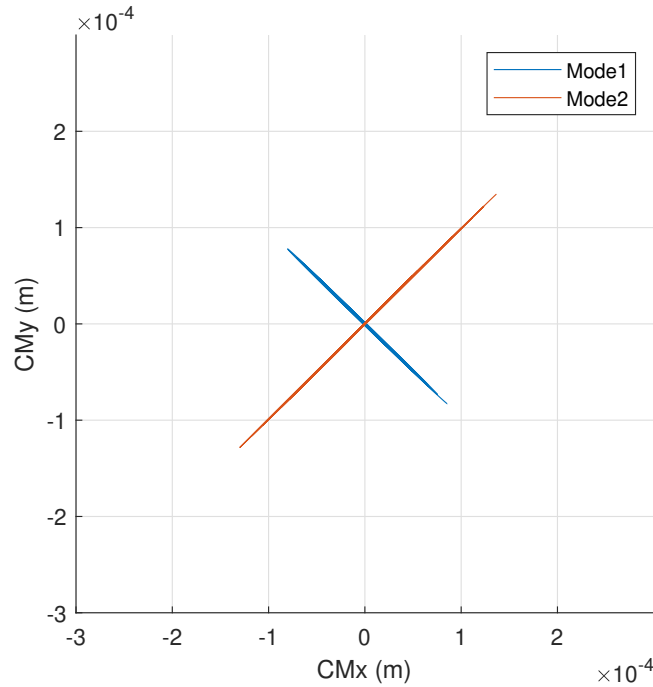
(e)

**Figure 3.40.** Moment on the tank over time in the stair step tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.



**Figure 3.41.** Top-down trace of the center of mass over time in the stair step tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.

one mode travels along the axis of symmetry, while the other acts at a right angle to this axis, such as shown in Figure 3.42.



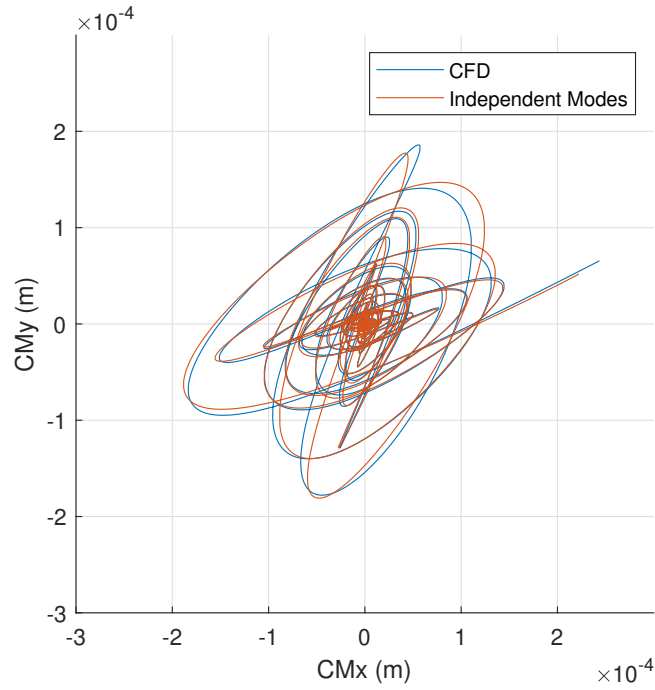
**Figure 3.42.** Independent slosh modes within the stair step tank acting along and perpendicular to the axis of symmetry.

These curve fits are able to match the center of mass trace very well, although some peaks are under- or over-shot. A comparison is given in Figure 3.43. This curve fit was done for the entire simulation period. Had only the last four seconds of simulation time been used, like for calculation of slosh parameters, the fit would be better.

### 3.2.5 Trapezoid

CFD cases of the trapezoid tank were run at slosh angles of 0, 30, 45, 60, and 90 degrees at the same fill level examined in the experiment. The results are summarized in Table 3.7.

The power spectral density of the CFD results are shown in Figure 3.44. The two most dominant frequencies visible in the CFD simulations are around 34.0 rad/s in the x-direction and 42.4 rad/s in the y-direction. This is again different than the experimental results which have dominant frequencies around and 45.6 and 57.7 rad/s. While the two most dominant

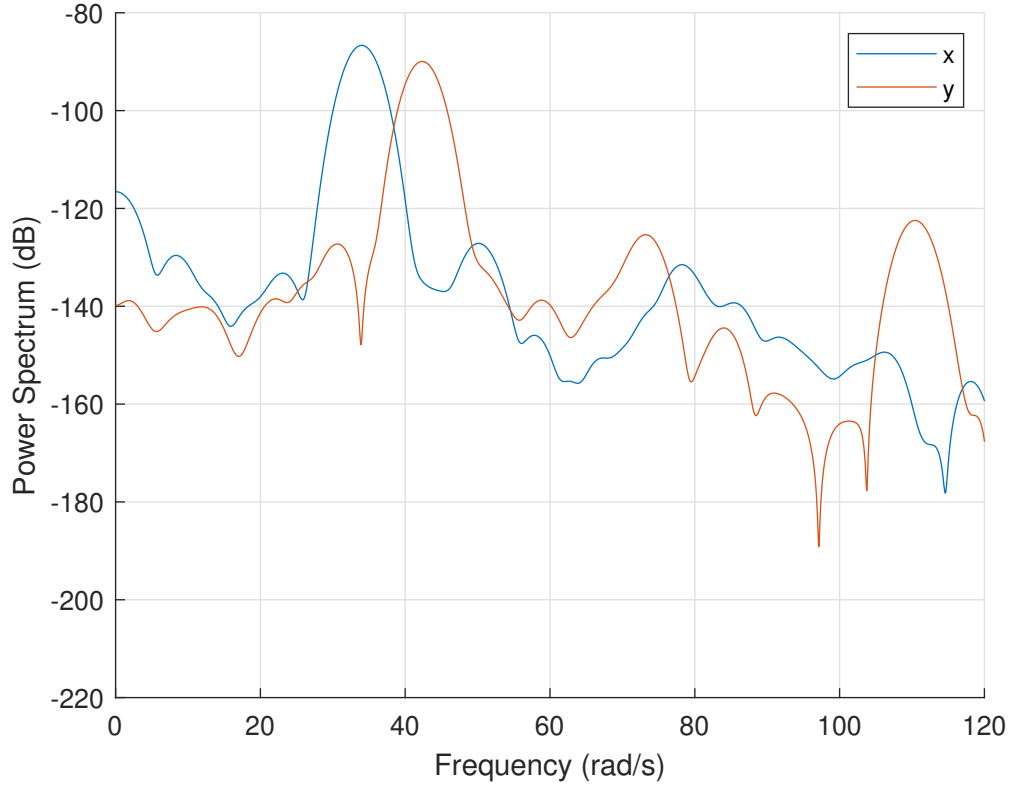


**Figure 3.43.** Comparison of center of mass traced from CFD results and trace created by curve fitting.

**Table 3.7.** Results of CFD simulations for the trapezoid tank.

Slosh Angle	Mode 1		Mode 2	
	$\zeta$ (%)	$\omega_n$ (rad/s)	$\zeta$ (%)	$\omega_n$ (rad/s)
0	0.87	34.0	0.42	42.5
30	0.97	34.0	0.37	42.4
45	0.93	34.0	0.41	42.4
60	0.98	34.0	0.43	42.4
90	-	-	0.33	43.3

frequencies are different, both results have peaks at similar locations, even if they are less prominent in the experimental results.



**Figure 3.44.** Power spectral density of the CFD solution for the trapezoid tank at a thirty-degree slosh angle.

Table 3.8 summarizes four of the main frequencies visible in both the CFD and experimental results. These are compared to the analytical solutions for the first and second natural frequencies along both the longest and shortest directions in the tank at the liquid free surface. There is a maximum of 13.2% difference between the CFD and experimental values over all four frequencies. There is up to a 14.8% difference between the analytical and CFD solutions and up to a 14.0% difference between the analytical and experimental solutions. It is again unclear why there are different most dominant frequencies for the CFD and experimental results.

It seems likely that there is more coupling between slosh in each direction in this geometry than others examined here. In the power spectral density, it appears that there are peaks,



**Table 3.8.** Identifiable frequencies in the trapezoid tank.

Analytical (rad/s)	Experimental (rad/s)	CFD (rad/s)
32.0	33.9	34.0
39.2	45.6	42.3
57.5	57.7	50.1
68.5	74.6	73.2

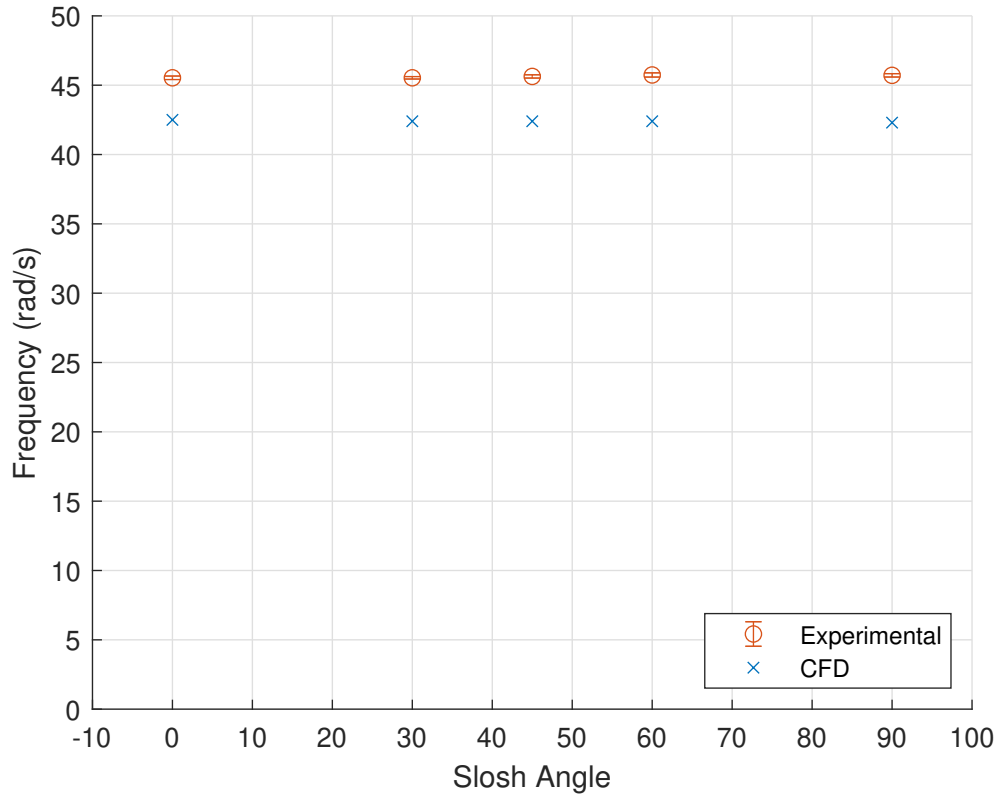
even if small peaks, at the sum and difference of the four most identifiable frequencies. For example, in the x-direction there is a peak around 8-9 rad/s in all but the 90-degree case. This is approximately the difference between the first natural frequency in the y-direction and the first natural frequency in the x-direction. This greater degree of coupling may be due to the slanted wall, as all other geometries have only vertical walls.

A comparison of CFD and experimental results for the overlapping frequency is given in Figure 3.45.

It is more difficult to compare the damping ratios in the CFD and experimental solutions because of the different dominant modes. However, both share a dominant frequency near 45 rad/s. There is a large difference of between the experimental and CFD results for damping ratio at this frequency, between 61 and 72%. The CFD simulation predicts a much lower damping ratio than the experiment, as shown in Figure 3.46. This could be due to the fact that the mesh for this geometry did not converge within a reasonable run-time. As mentioned earlier, confidence in frequencies are high, but damping ratio is likely problematic. This is especially true because this mode comes from slosh in the shortest direction, which would have the fewest cells spanning the container length.

Figure 3.47 shows plots of the center of mass over time at each tested slosh angle. In each direction, the center of mass appears to be a single damped sinusoid. At a zero-degree slosh angle, there is some slight movement in the y-direction. Because of the defined initial surface offset, the slanted wall of the trapezoid causes a small initial shift in the center of mass in the y-direction. At a 90-degree slosh angle, there is almost no movement in the x-direction because this is along a tank axis of symmetry.

The forces and moments over time are shown in Figures 3.48 and 3.49 respectively. The results for force in the y-direction and moment in the x-direction appear "rougher" than the



**Figure 3.45.** Comparison of overlapping frequency between CFD and experimental data in the trapezoid tank.

plots of other geometries. Forces and moments in these directions are the result of slosh impacts on the slanted wall and the vertical wall opposite it. The slanted wall may be the cause of this roughness.

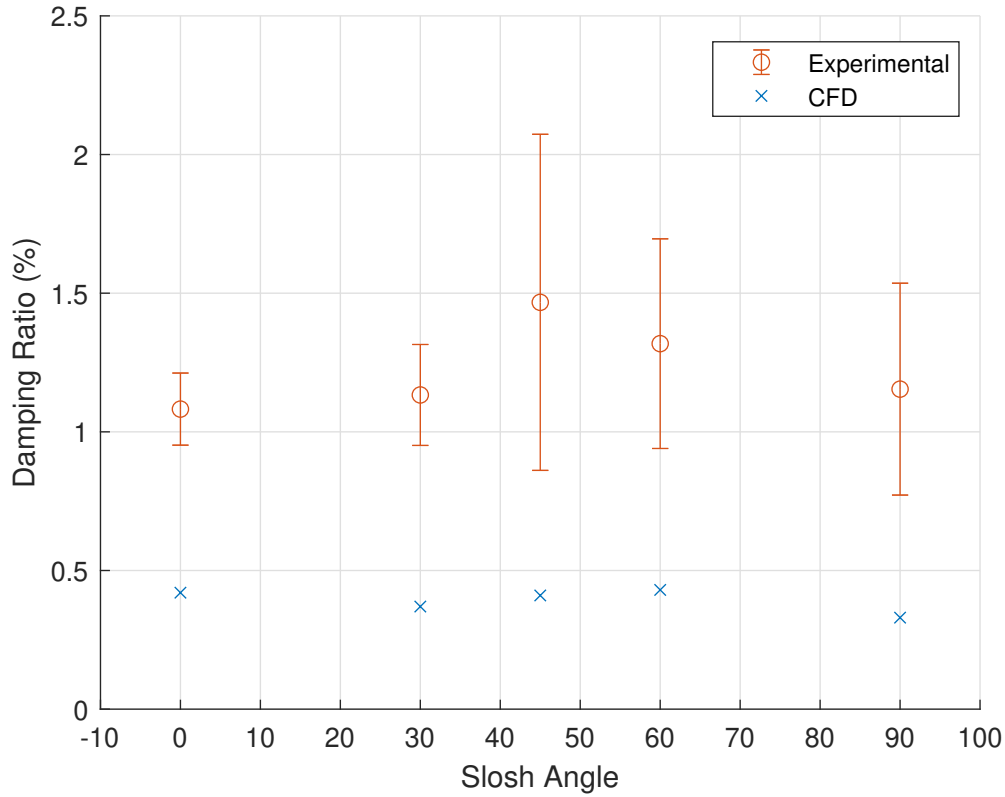
The top-down trace of the center of mass over time is given in Figure 3.50. There is some movement in each direction for all slosh angles except 90-degrees. At 90-degrees, slosh is only along the line of symmetry.

### 3.2.6 Quarter Annular Cylinder

CFD cases of for quarter annular cylinder tank were run at slosh angles of 0, 30, 45, 60, and 90 degrees at the same fill level examined in the experiment. The results are summarized in Table 3.9.

**Table 3.9.** Results of CFD simulations for the quarter annular cylinder tank.

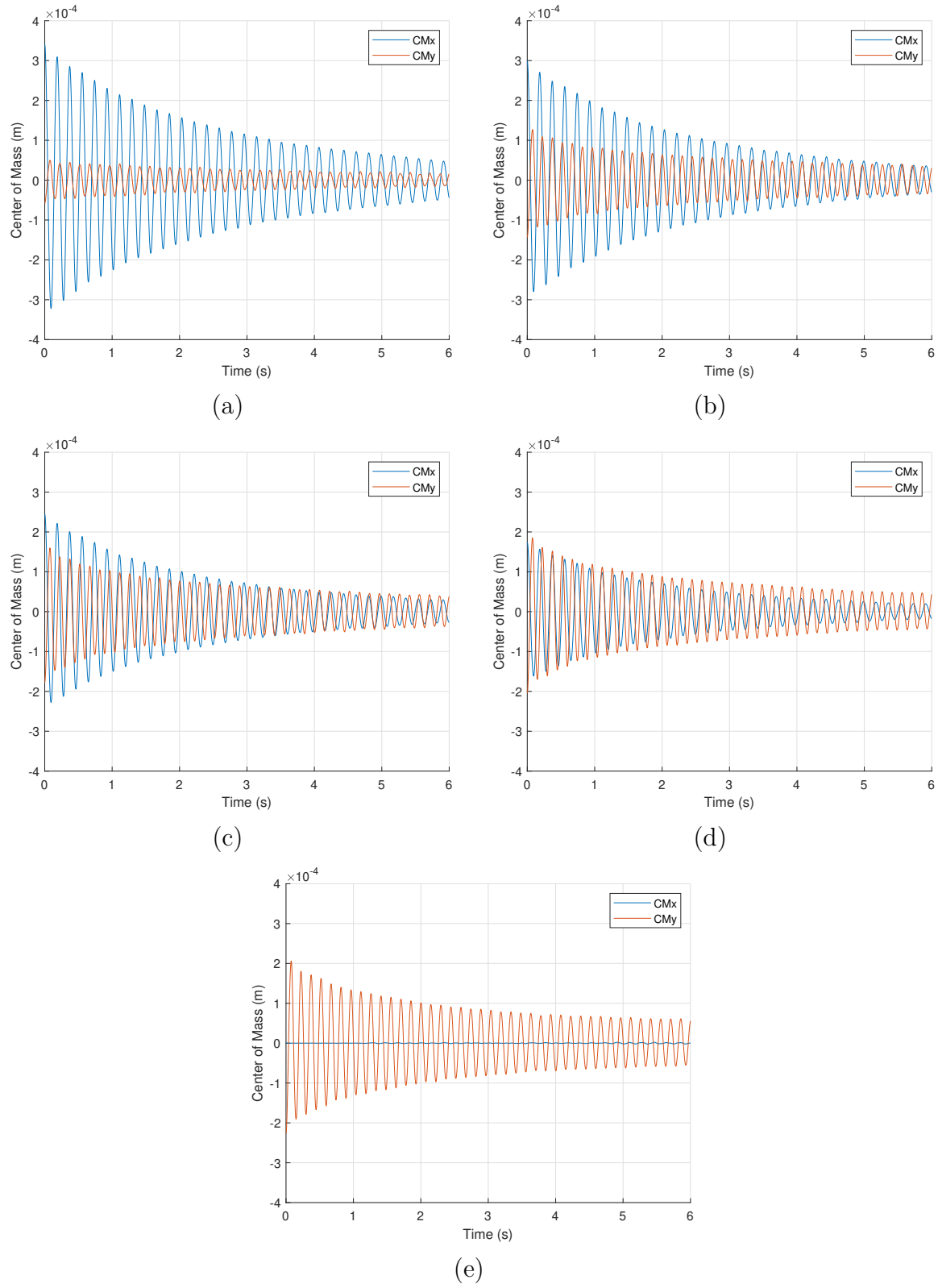
	<b>Mode 1</b>		<b>Mode 2</b>	
<b>Slosh Angle</b>	$\zeta$ (%)	$\omega_n$ (rad/s)	$\zeta$ (%)	$\omega_n$ (rad/s)
<b>0</b>	1.49	31.8	-	-
<b>30</b>	1.50	31.8	1.33	48.6
<b>45</b>	1.49	31.8	1.40	48.6
<b>60</b>	1.51	31.8	1.36	48.5
<b>90</b>	-	-	1.37	48.5



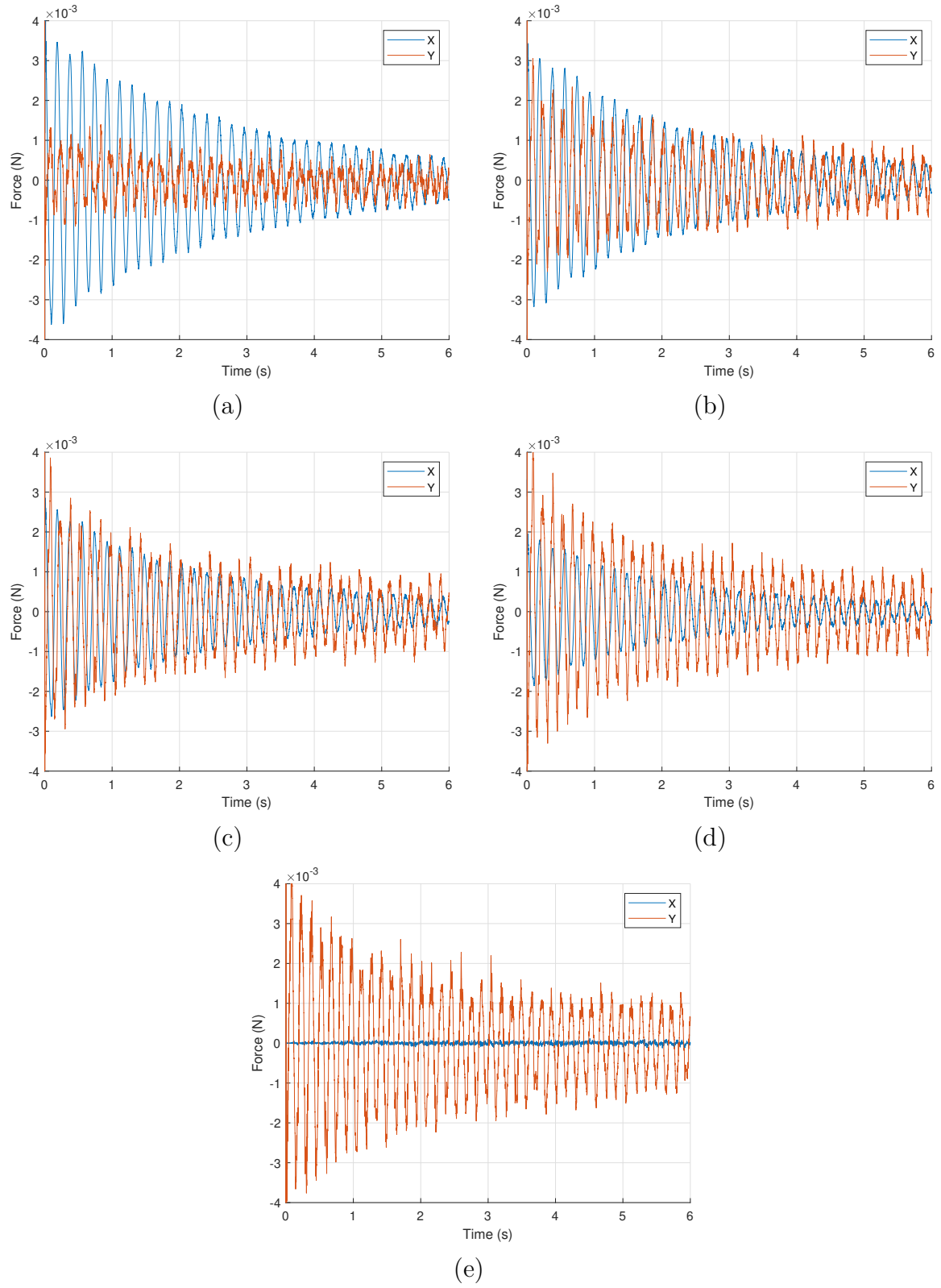
**Figure 3.46.** Comparison of overlapping damping ratio between CFD and experimental data in the trapezoid tank.

The power spectral density of the CFD results are shown in Figure 3.51. The two most dominant frequencies visible in the CFD simulations are around 31.8 and 48.5 rad/s. This is again different than the experimental results which have dominant frequencies around 46.6 and 63.6 rad/s. While the two most dominant frequencies are different, both results have peaks at similar locations, even if they are less prominent.

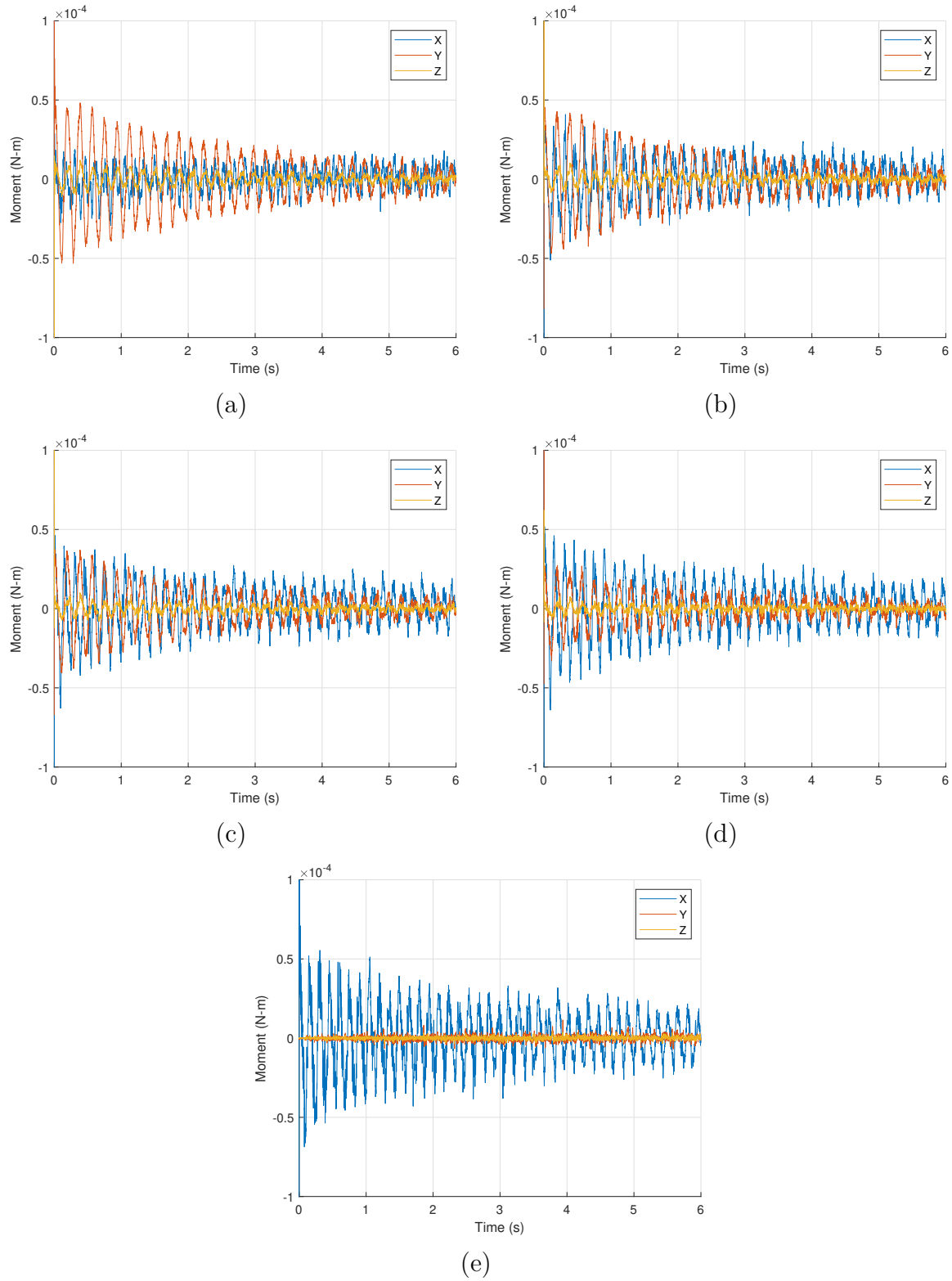
Table 3.10 summarizes four of the main frequencies visible in both the CFD and experimental results. These are compared to the analytical solutions for the first and second natural frequencies along both the longest (farthest corner-to-corner distance) and shortest (radial) directions in the tank. There is a maximum of 8.4% difference between the CFD and experimental values over all four frequencies. There is up to a 13.3% difference between the analytical and CFD solutions and up to a 10.8% difference between the analytical and experimental solutions. Sector tanks usually have one slosh mode that acts in the radial



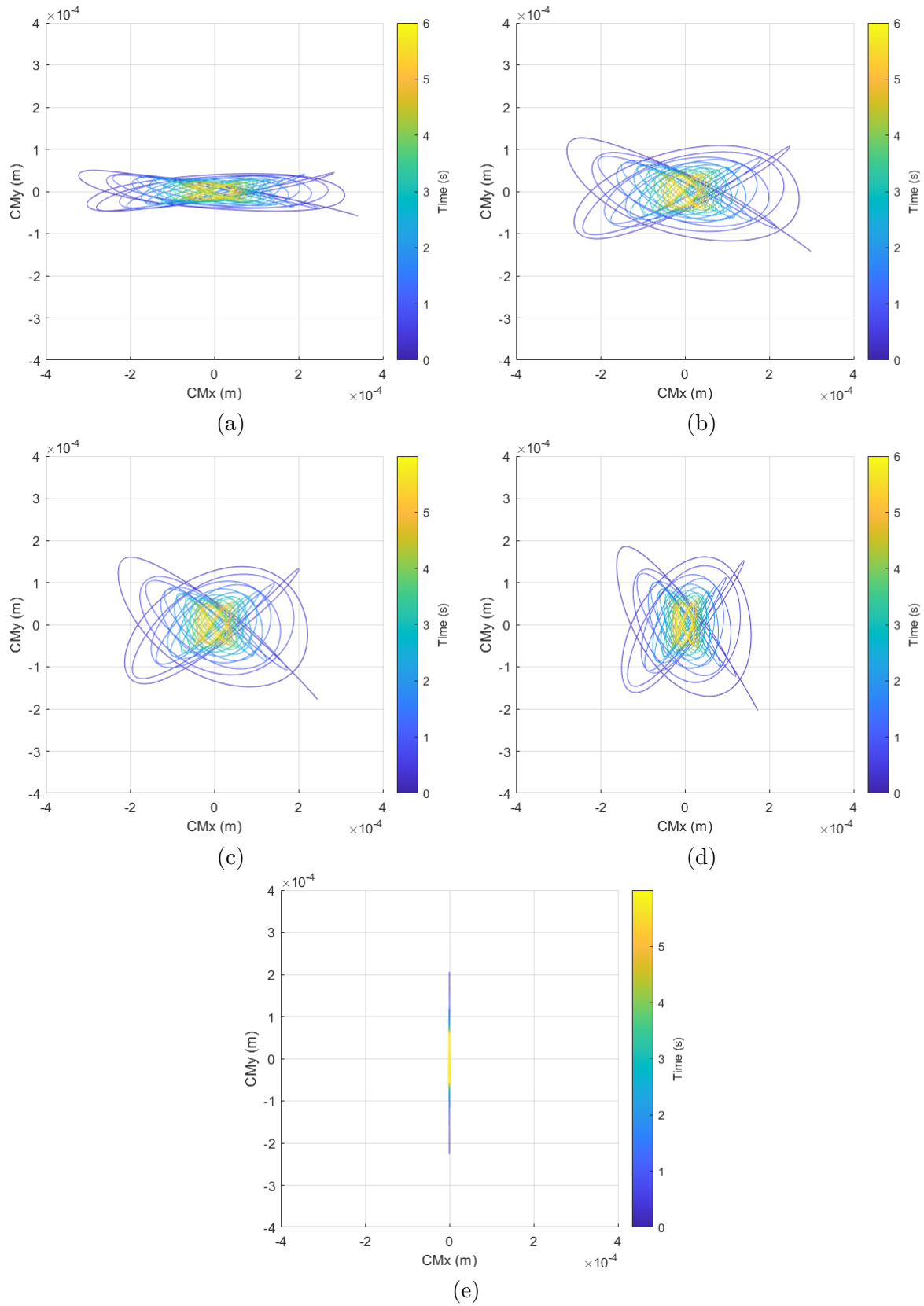
**Figure 3.47.** Center of mass over time in the trapezoid tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.



**Figure 3.48.** Force on the walls over time in a trapezoid tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.

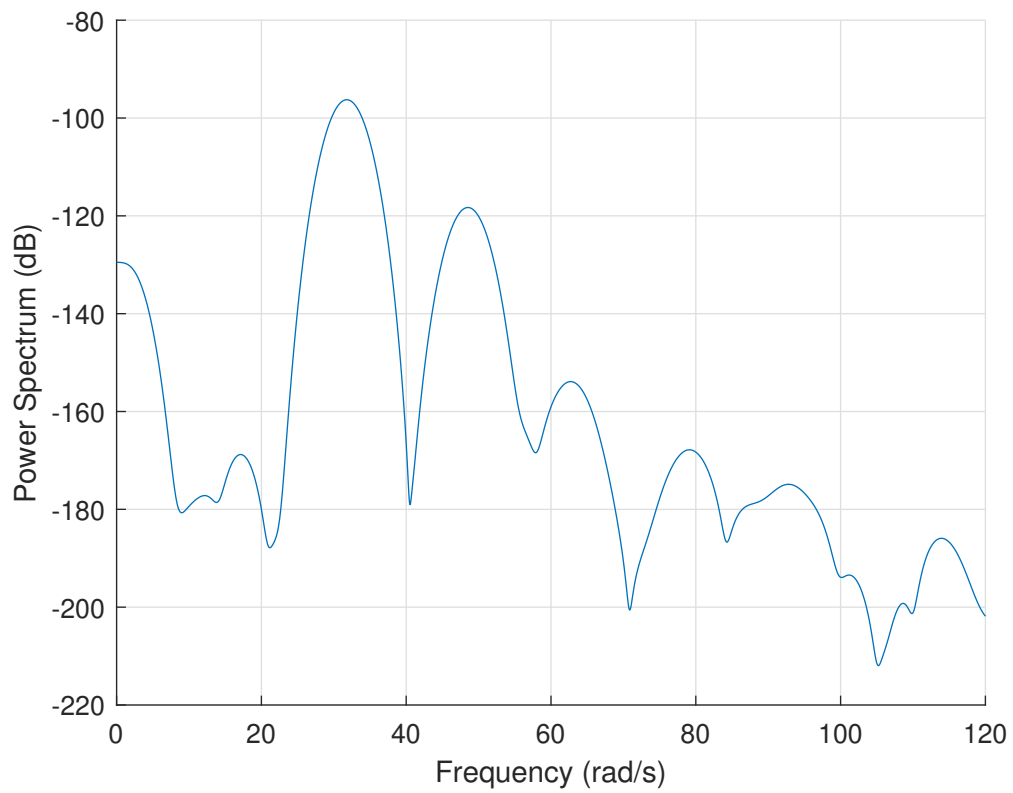


**Figure 3.49.** Moment on the tank over time in the trapezoid tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.



**Figure 3.50.** Top-down trace of the center of mass over time in the trapezoid tank for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.





**Figure 3.51.** Power spectral density of the CFD solution for the quarter annular cylinder tank at a forty-five degree slosh angle.

direction, and one in the circumferential direction [5]. This is likely what these two slosh modes correspond to. The analytical values presented here are approximations assuming a rectangular tank of the same dimension. It is again unclear why there are different dominant frequencies for the CFD and experimental results.

**Table 3.10.** Identifiable frequencies in the quarter annular cylinder tank.

Analytical	Experimental	CFD
32.8	34.7	31.8
42.8	46.6	48.5
57.1	63.6	62.7
74.2	83.2	79.1

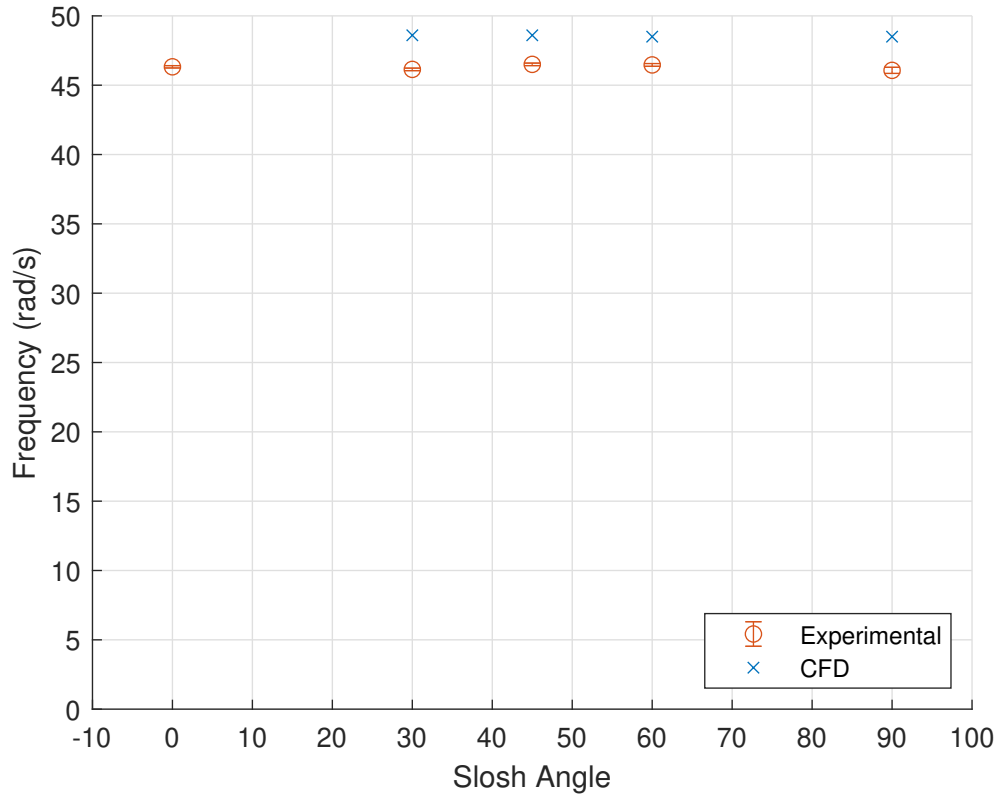
A comparison between the CFD and experimental results for natural frequency are given in Figure 3.52 for the one overlapping dominant frequency. Again, uncertainty in the natural frequency is low.

It is more difficult to compare the damping ratios in the CFD and experimental solutions because of the different dominant modes. However, both share a dominant frequency near 47 rad/s. There is between a 1.4 and 21.3% difference between the damping ratio in the experimental and CFD solutions. These results are shown in Figure 3.53

Figure 3.54 shows plots of the center of mass over time at each tested slosh angle. The results for the quarter annular cylinder tank have the appearance of the sum of at least two damped sinusoids for 30, 45, and 60-degree slosh angles. There appears to be only one dominant frequency in each direction for the 0 and 90 degree slosh angles. At a 0-degree slosh angle, the most identifiable frequency is Mode 1, acting in the circumferential direction. At the 90-degree slosh angle, the frequency is Mode 2, acting in the radial direction. This makes sense due to the tank orientation at these angles.

The forces and moments over time are shown in Figures 3.55 and 3.56 respectively. There appears to be more static in the results for forces and moments toward the end of the time period, especially at the 90-degree slosh angle. This is likely due to extremely small motions at this point making calculations difficult.

The top-down trace of the center of mass over time is given in Figure 3.57. At a 0-degree slosh angle, slosh is mainly in the circumferential direction. It is a little bit surprising that

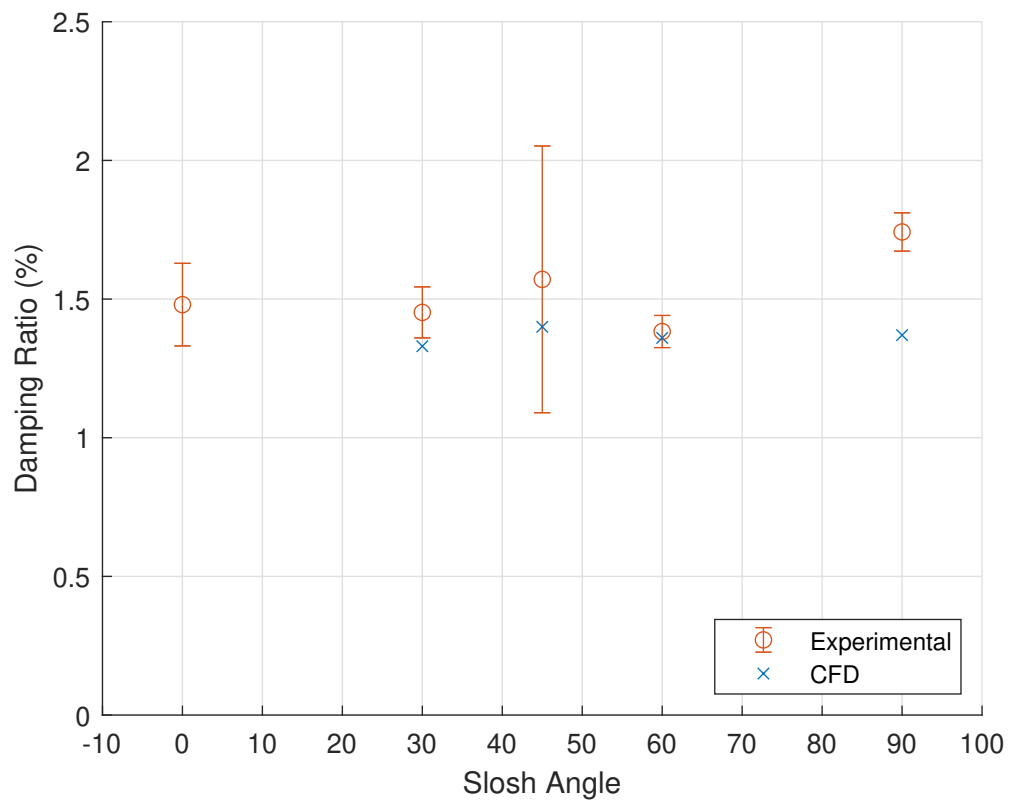


**Figure 3.52.** Comparison of overlapping frequency between CFD and experimental data in the quarter annular cylinder.

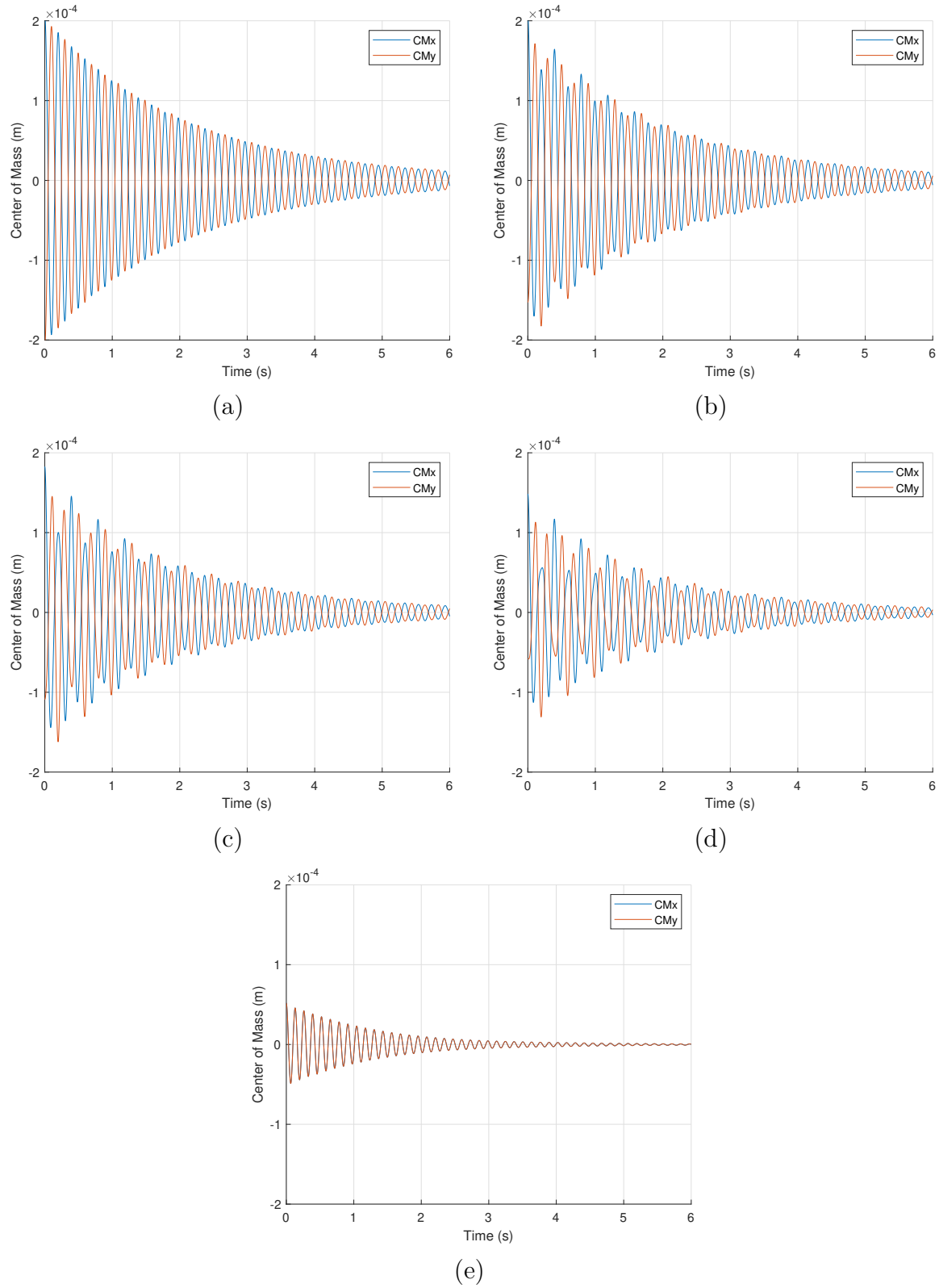
the center of mass trace is nearly a straight line. It was expected that the line would be slightly curved or with more divergence from the line. The 90-degree angle trace is along the line of symmetry which was expected.

Similarly to the stair step tank, the data can be split into one wave corresponding to each frequency. When each of these modes are plotted, the result is that one mode travels along the axis of symmetry, while the other acts at a right angle to this axis, such as shown in Figure 3.58.

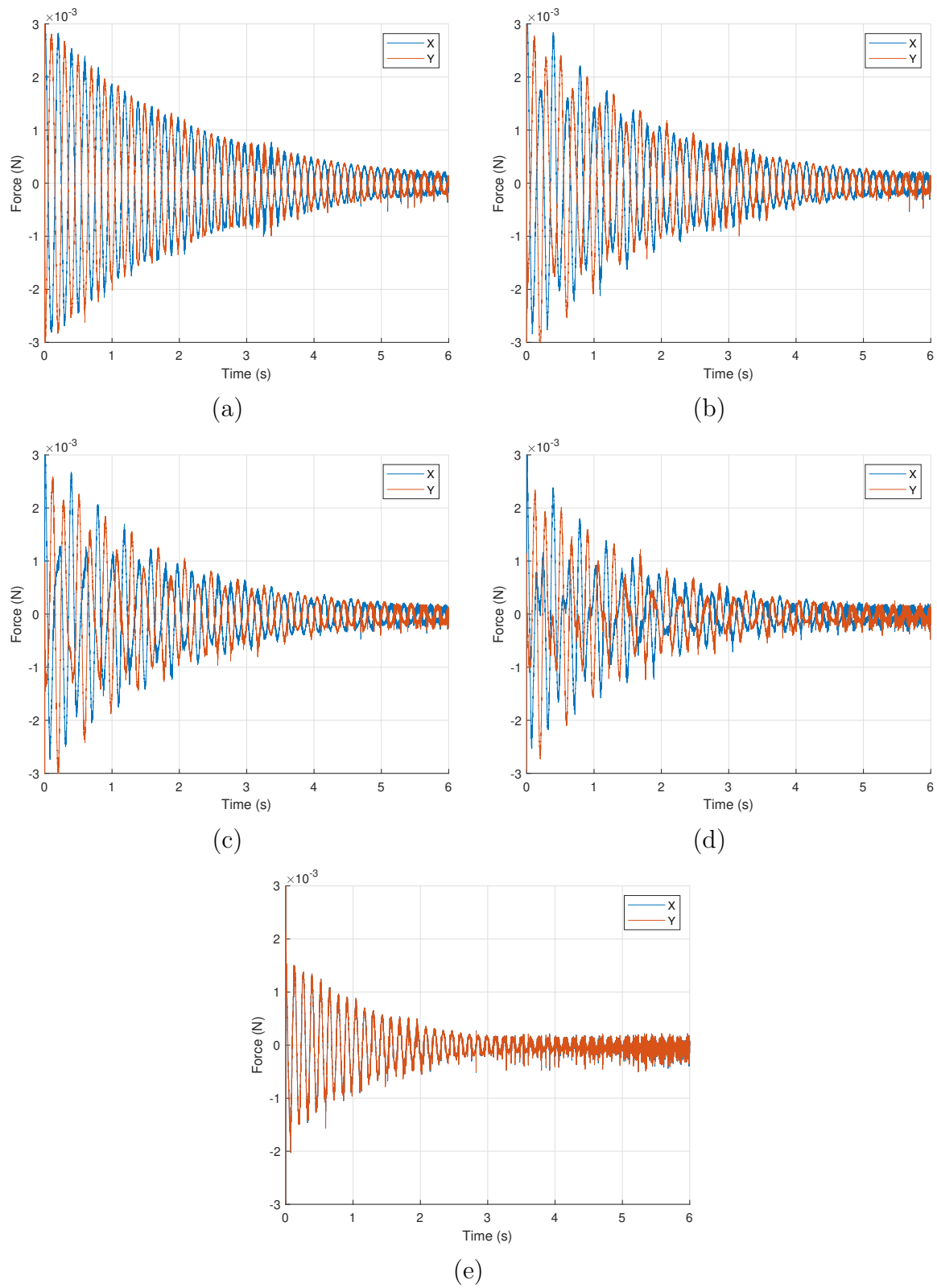
These curve fits are able to match the center of mass trace very well, although some peaks are under- or over-shot. A comparison is given in Figure 3.59. This curve fit was done for the entire simulation period. Had only the last four seconds of simulation time been used, like for calculation of slosh parameters, the fit would be better.



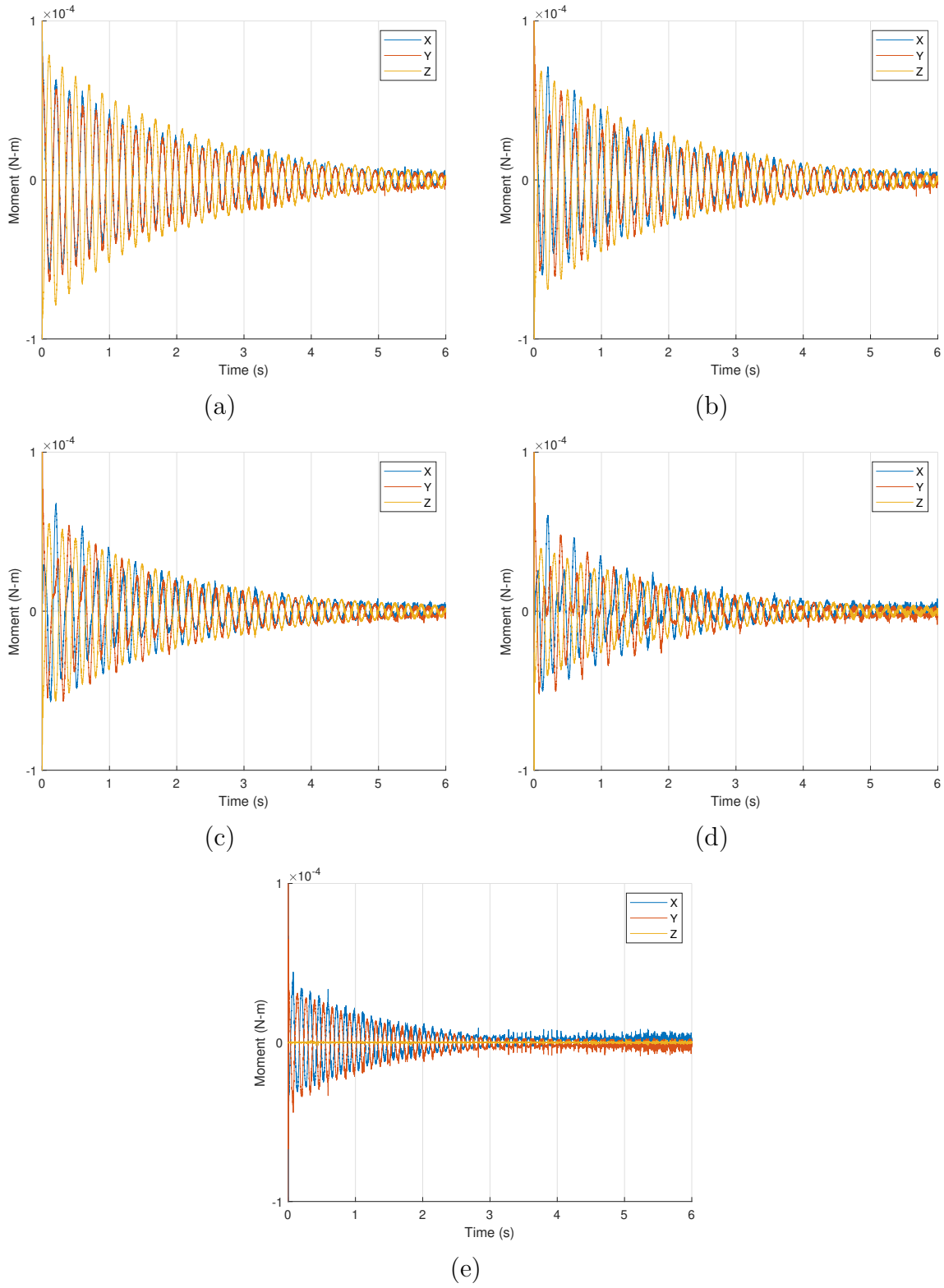
**Figure 3.53.** Comparison of overlapping damping ratio between CFD and experimental data in the quarter annular cylinder.



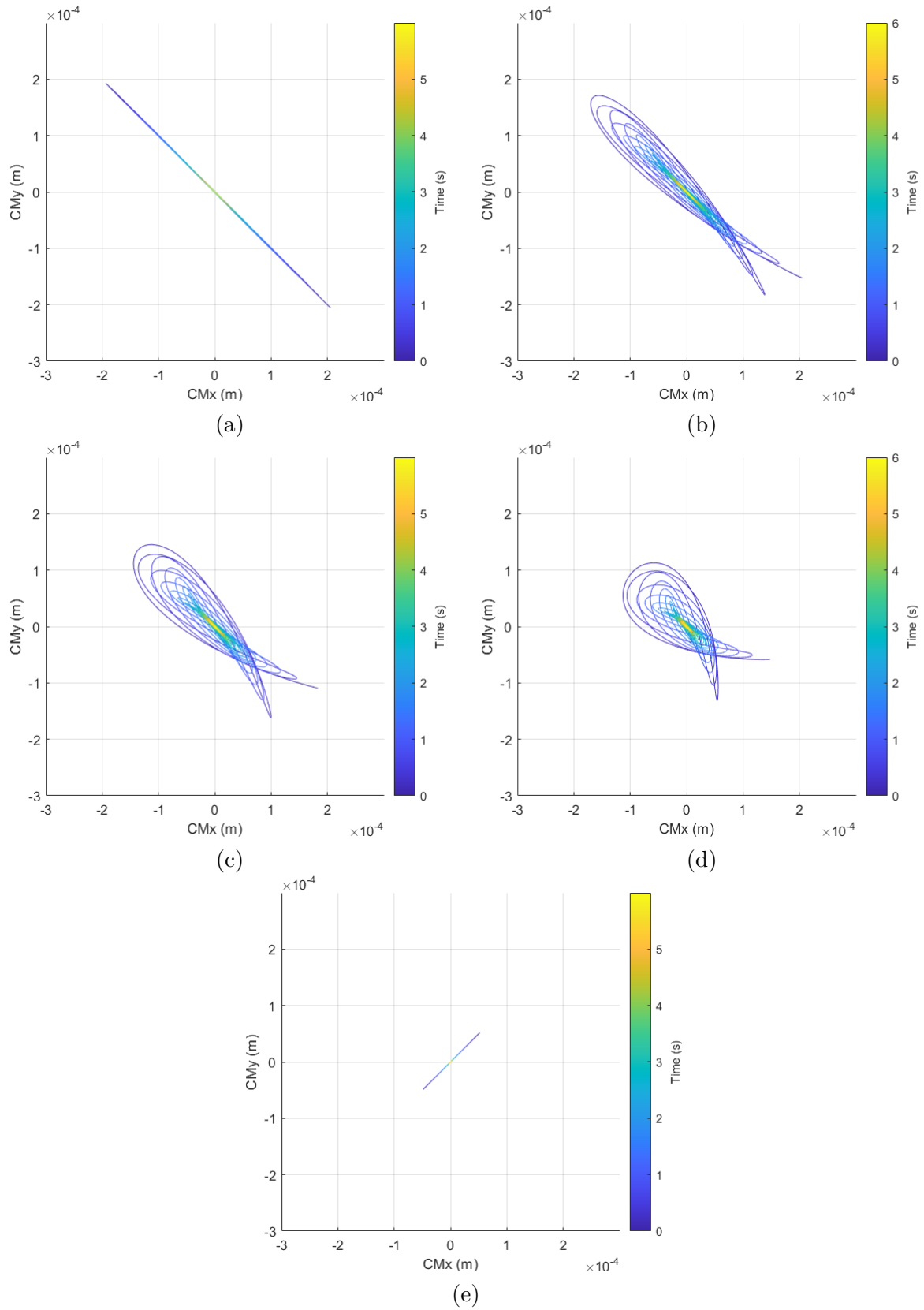
**Figure 3.54.** Center of mass over time in the quarter annular cylinder for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.



**Figure 3.55.** Force on the walls over time in a quarter annular cylinder for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.

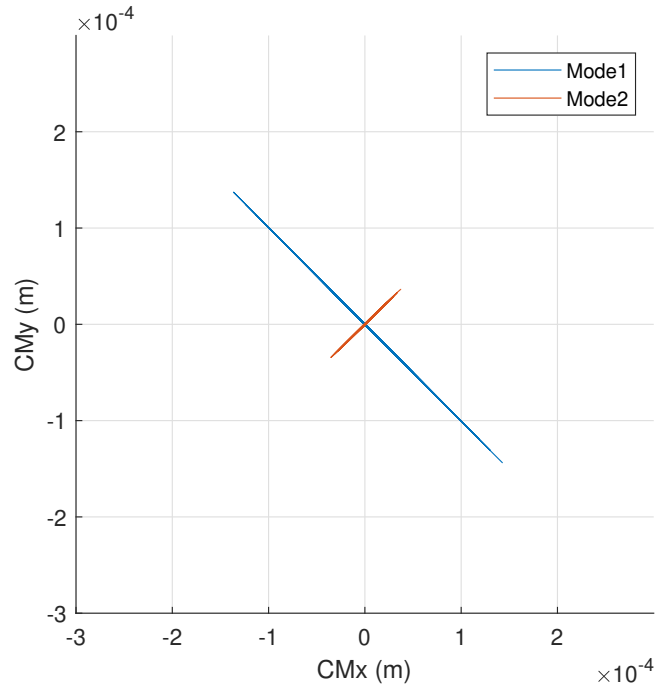


**Figure 3.56.** Moment on the tank over time in the quarter annular cylinder for a sosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.

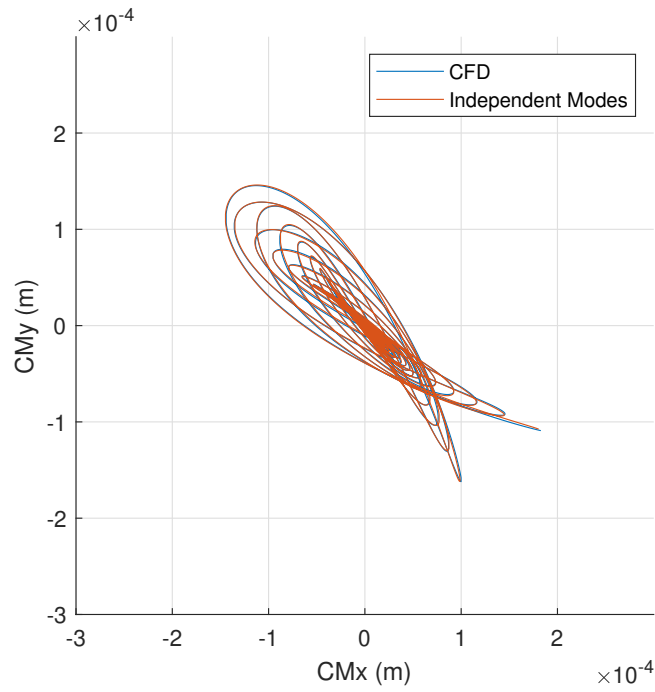


**Figure 3.57.** Top-down trace of the center of mass over time in the quarter annular cylinder for a slosh angle of (a) 0, (b) 30, (c) 45, (d) 60, and (e) 90 degrees.





**Figure 3.58.** Independent slosh modes within the quarter annular cylinder tank acting along and perpendicular to the axis of symmetry.



**Figure 3.59.** Comparison of center of mass traced from CFD results and trace created by curve fitting.

## 4. CONCLUSION

Experimental data was collected and CFD models created for six different tank geometries, five of which are non-traditional geometries for spacecraft tanks. These conformal tanks have the potential to improve volume usage in small satellites. Both experimental and CFD results identified the dominant frequencies and damping ratios in the tanks, two parameters of great interest to guidance, navigation, and control.

It appears that especially for the basic conformal geometries, the total behavior is the sum of slosh along each dimension. For example, in each direction in the rectangular tank, slosh motion appeared to exhibit only one natural frequency. Adding the behavior in each direction resulted in total behavior that took the appearance of the sum of two damped sinusoids.

The same general idea applied to more complicated geometries. However, instead of one frequency in each direction, multiple frequencies were visible. For example, in the stair step tank, two frequencies were observable in both the x and y direction. Each of these two frequencies roughly corresponded to slosh in the longest and the shortest dimension. However, natural frequencies in the complicated geometries did not match the analytical values as closely as the basic geometries did. This is likely due to the fact the analytical solutions are for a simple rectangular tank instead of a more complicated geometry. The use of these analytical approximations are simply useful to check that the results presented here are reasonable.

When slosh is aligned with an axis of symmetry in the tank, the center of mass trace remains along a single line in that direction. This makes sense because the same slosh motion should be taking place on either side of that line of symmetry. It also appears that for slosh not on this line of symmetry, slosh can be broken down into one mode which travels along the axis of symmetry, and one which travels perpendicular to this axis.

CFD simulations in this research were created using open-source software, OpenFOAM. The relatively good agreement between CFD simulations, experimental results, and analytical solutions support the idea that this open-source software can be a valuable tool for engineers without the need for expensive software.

## 5. RECOMMENDATIONS

Basic geometries in this study had edges with right angles and no fillets. This would not be the best design for spaceflight, because liquid would become trapped along these edges in low gravity. This would prevent the tank from fully draining. A better design would be one with intelligently designed edge fillets [39]. Therefore, it would be useful to examine the difference in slosh behavior between these basic geometries and similar geometries with edge fillets added.

The complicated tanks used in this experiment did have tapered fillets, although the CFD solutions did not so that the frequencies caused by the walls alone could be isolated. It would be very beneficial to complete more CFD simulations which fully match the tank geometries with edge fillets. The difference between the two simulations would show the impact of the edge fillets. Due to the reasonably good match between the experimental data and CFD results presented here, there may be little difference in slosh behavior between a tank with and without edge fillets.

It appears clear that the total behavior of slosh in a rectangular container is the sum of the behavior in each of the two dimensions. Analytical solutions for frequency may be used to predict the actual frequency. However, no correlations for damping ratio could be found. It might be useful to complete further experiments and CFD simulations for more rectangular geometries and fill levels. If a trend in the damping ratio could be determined, it would be useful for those designing conformal tanks in the future.

All of these experiments and simulations were done under Earth's gravity ( $9.81 \text{ m/s}^2$ ). It would be beneficial to investigate slosh in these conformal tanks under low gravity, using existing data as CFD verification [11]. This would give a better look at slosh that might occur while a satellite is in orbit.

Using the data presented here, it would be beneficial to investigate the applicability of equivalent mechanical models to these geometries. Equivalent mechanical models are important for guidance, navigation, and control, and are generally included in control system modeling. These models would no doubt be more complicated for some conformal tanks

than those required for traditional tank geometries, but will be important for flight use of the tanks.

Finally, it would be useful to examine the impact of PMDs on slosh in these tanks. PMDs generally introduce additional slosh modes, so it is expected that the resulting slosh would be more complicated.

## REFERENCES

- [1] D. J. Benson and W. Ng, “Validation of slosh modeling approach using star-ccm+,” *3AF Space Propulsion 2018*, May 14, 2018. [Online]. Available: <https://ntrs.nasa.gov/citations/20180003007>.
- [2] R. Zhou, M. Vergalla, S. Chintalapati, D. Kirk, and H. Gutierrez, “Experimental and numerical investigation of liquid slosh behavior using ground-based platforms,” *Journal of Spacecraft and Rockets*, Nov. 2012. DOI: [10.2514/1.A32052](https://doi.org/10.2514/1.A32052).
- [3] G. Musgrove and S. Coogan, “Validation and rules-of-thumb for computational predictions of liquid slosh dynamics,” *51st AIAA Joint Propulsion Conference*, Jul. 2015. DOI: [10.2514/6.2015-4076](https://doi.org/10.2514/6.2015-4076).
- [4] H. N. Abramson, “The dynamic behavior of liquids in moving containers with applications to space vehicle technology,” Southwest Research Institute, Washington, DC, NASA SP-106, 1966.
- [5] F. T. Dodge, “The new dynamic behavior of liquids in moving containers,” Southwest Research Institute, San Antonio, Texas, 2000.
- [6] H. Q. Yang, R. Purandare, J. Peugeot, and J. West, “Prediction of liquid slosh damping using a high resolution cfd tool,” *48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit 2012*, Jul. 2012. DOI: [10.2514/6.2012-4294](https://doi.org/10.2514/6.2012-4294).
- [7] H. N. Abramson, L. R. Garza, and D. D. Kana, “Some notes on liquids sloshing in compartmented cylindrical tanks,” Southwest Research Institute, San Antonio, Texas, Technical Report No. 1, Feb. 15, 1962.
- [8] H. F. Bauer, “Liquid sloshing in a cylindrical quarter tank,” *AIAA Journal*, vol. 1, pp. 2601–2606, 11 Nov. 1963. DOI: [10.2514/3.2118](https://doi.org/10.2514/3.2118).
- [9] E. Chaichanasiri and C. Suvanjumrat, “Simulation of three dimensional liquid-sloshing models using c++ open source code cfd software,” *Kasetsart Journal - Natural Science*, vol. 46, pp. 978–995, Nov. 2012.
- [10] A. Y. Lee and J. Stupik, “In-flight characterization of the cassini spacecraft propellant slosh modes,” *Journal of Spacecraft and Rockets*, vol. 54, pp. 417–425, 2 2017. DOI: [10.2514/1.A33636](https://doi.org/10.2514/1.A33636).
- [11] J. A. Salzman and W. J. Masica, “Lateral sloshing in cylinders under low-gravity conditions,” Lewis Research Center, Cleveland, Ohio, NASA TN D-5058, Feb. 1969.

- [12] H. N. Abramson, W.-H. Chu, and D. D. Kana, "Some studies of nonlinear lateral sloshing in rigid containers," Southwest Research Institute, San Antonio, Texas, NASA CR-375, Jan. 1966.
- [13] B. Marsell, S. Gangadharan, Y. Chatman, J. Sudermann, K. Schlee, and J. Ristow, "A cfd approach to modeling spacecraft fuel slosh," *47th AIAA Aerospace Sciences Meeting*, Jan. 2009. DOI: [10.2514/6.2009-366](https://doi.org/10.2514/6.2009-366).
- [14] R. E. Manning, I. Ballinger, M. Bhatia, and M. Dowdy, "Design of the europa clipper propellant management device," *AIAA Propulsion and Energy 2019 Forum*, Aug. 2019. DOI: [10.2514/6.2019-3858](https://doi.org/10.2514/6.2019-3858).
- [15] D. Y. Lee and H. S. Choi, "Study on sloshing in cargo tanks including hydroelastic effects," *Journal of Marine Science and Technology*, vol. 4, pp. 27–34, 1999. DOI: [10.1007/s007730050004](https://doi.org/10.1007/s007730050004).
- [16] D. H. Lee, M. H. Kim, S. H. Kwon, J. W. Kim, and Y. B. Lee, "A parametric sensitivity study on lng tank sloshing loads by numerical simulations," *Ocean Engineering*, vol. 34, pp. 3–9, 1 Jan. 2007. DOI: [10.1016/j.oceaneng.2006.03.014](https://doi.org/10.1016/j.oceaneng.2006.03.014).
- [17] J. H. Jung, H. S. Yoon, and S. C. Shin, "Effect of the vertical baffle height on the liquid sloshing in a three-dimensional rectangular tank," *Ocean Engineering*, vol. 44, pp. 79–89, Apr. 2012. DOI: [10.1016/j.oceaneng.2012.01.034](https://doi.org/10.1016/j.oceaneng.2012.01.034).
- [18] S. Asmar and S. Matousek, "Mars cube one (marco) shifting the paradigm in relay deep space operations," *SpaceOps 2016 Conference*, May 2016. DOI: [10.2514/6.2016-2483](https://doi.org/10.2514/6.2016-2483).
- [19] J. Schoolcraft, A. Klesh, and T. Werne, "Marco: Interplanetary mission development on a cubesat scale," *SpaceOps 2016 Conference*, May 2016. DOI: [10.2514/6.2016-2491](https://doi.org/10.2514/6.2016-2491).
- [20] VACCO, *Jpl marco- micro cubeset propulsion system*, 2019. [Online]. Available: [https://cubesat-propulsion.com/wp-content/uploads/2015/11/X14102000-01\\_2019update.pdf](https://cubesat-propulsion.com/wp-content/uploads/2015/11/X14102000-01_2019update.pdf) (visited on 10/11/2020).
- [21] VACCO, *Micro propulsion system summary*, 2018. [Online]. Available: <https://www.cubesat-propulsion.com/wp-content/uploads/2018/09/VACCO-Micro-Propulsion-Systems-Summary-web2-Sept2018.pdf> (visited on 10/11/2020).
- [22] NASA. (2019). "Marco (mars cube one)," [Online]. Available: <https://solarsystem.nasa.gov/missions/mars-cube-one/in-depth/> (visited on 10/11/2020).
- [23] VACCO, *Propulsion unit for cubesats*, Sep. 2019. [Online]. Available: [https://cubesat-propulsion.com/wp-content/uploads/2016/03/11044000-01\\_PUC\\_2019update.pdf](https://cubesat-propulsion.com/wp-content/uploads/2016/03/11044000-01_PUC_2019update.pdf) (visited on 10/11/2020).

- [24] VACCO, *Mepsi micro propulsion system*, Aug. 2013. [Online]. Available: [https://www.vacco.com/images/uploads/pdfs/MicroPropulsionSystems\\_0714.pdf](https://www.vacco.com/images/uploads/pdfs/MicroPropulsionSystems_0714.pdf) (visited on 10/11/2020).
- [25] VACCO, *Reaction control propulsion system*, Jul. 2014. [Online]. Available: [https://www.vacco.com/images/uploads/pdfs/ReactionControlPropulsionModule-X13003000-01\\_0414.pdf](https://www.vacco.com/images/uploads/pdfs/ReactionControlPropulsionModule-X13003000-01_0414.pdf) (visited on 10/11/2020).
- [26] VACCO, *Standard micro propulsion system*, Jun. 2015. [Online]. Available: [https://www.vacco.com/images/uploads/pdfs/MiPS\\_standard\\_0714.pdf](https://www.vacco.com/images/uploads/pdfs/MiPS_standard_0714.pdf) (visited on 10/11/2020).
- [27] GOMSpace, *Nanoprop cgp3*. [Online]. Available: [https://gomspace.com/UserFiles/Subsystems/flyer/gomspace\\_nanoprop\\_cgp3.pdf](https://gomspace.com/UserFiles/Subsystems/flyer/gomspace_nanoprop_cgp3.pdf) (visited on 10/11/2020).
- [28] GOMSpace, *Nanoprop 6u*. [Online]. Available: [https://gomspace.com/UserFiles/Subsystems/flyer/gomspace\\_nanoprop\\_cgp6\\_flyer.pdf](https://gomspace.com/UserFiles/Subsystems/flyer/gomspace_nanoprop_cgp6_flyer.pdf) (visited on 10/11/2020).
- [29] R. K. Masse, R. Spores, and M. Allen, “Af-m315e advanced green propulsion – gpim and beyond,” *AIAA Propulsion and Energy 2020 Forum*, Aug. 2020. DOI: [10.2514/6.2020-3517](https://doi.org/10.2514/6.2020-3517).
- [30] R. K. Masse, R. Spores, M. Allen, S. Kimbrel, E. Lorimor, and P. Myers, “Gpim af-m315e propulsion system,” *51st AIAA/SAE/ASEE Joint Propulsion Conference*, Jul. 2015. DOI: [10.2514/6.2015-3753](https://doi.org/10.2514/6.2015-3753).
- [31] Northrup Grumman. (2020). “Diaphragm tanks data sheets – sorted by volume,” [Online]. Available: <https://www.northropgrumman.com/space/diaphragm-tanks-data-sheets-sorted-by-volume/> (visited on 10/11/2020).
- [32] Aerojet Rocketdyne, *Modular propulsion systems*. [Online]. Available: <https://www.rocket.com/sites/default/files/documents/CubeSat%20Mod%20Prop-2sided.pdf> (visited on 10/11/2020).
- [33] Busek. (2019). “Propulsion for cubesats and nanosats,” [Online]. Available: [http://busek.com/technologies\\_\\_cubesatprop.htm](http://busek.com/technologies__cubesatprop.htm) (visited on 10/11/2020).
- [34] Busek, *Bgt-x5 green monopropellant thruster*, 2016. [Online]. Available: [http://busek.com/index\\_htm\\_files/70008517E.pdf](http://busek.com/index_htm_files/70008517E.pdf) (visited on 10/11/2020).
- [35] R. S. Figliola and D. E. Beasley, *Theory and Design for Mechanical Measurements*, 6th ed. Wiley, 2015, ISBN: 9781118881279.
- [36] Amazon Web Services, *Amazon elastic cloud compute*. [Online]. Available: <https://docs.aws.amazon.com/ec2/index.html>.

- [37] OpenFOAM Foundation, *Openfoam*, version v8, Sep. 1, 2020. [Online]. Available: <https://openfoam.org/version/8/> (visited on 10/11/2020).
- [38] E. Beckman. (Dec. 5, 2020). “Summation in function object for multiple processors,” [Online]. Available: <https://www.cfd-online.com/Forums/openfoam-post-processing/230887-summation-function-object-multiple-processors.html#post789859> (visited on 12/05/2020).
- [39] S. H. Collicott, E. A. Beckman, and P. Srikanth, “Conformal tanks: Small-sat propellant management technology,” *AIAA Propulsion and Energy 2019 Forum*, Aug. 2019. DOI: [10.2514/6.2019-3874](https://doi.org/10.2514/6.2019-3874).
- [40] OpenFOAM Foundation, *Openfoam user guide*, 2020. [Online]. Available: <https://cfd.direct/openfoam/user-guide/> (visited on 10/11/2020).
- [41] N. Herreras and J. Izarra, “Two-phase pipeflow simulations with openfoam,” *Masters of Energy and Process Engineering*, Norwegian University of Science and Technology, Jun. 2013. [Online]. Available: [https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/235256/651439\\_FULLTEXT01.pdf?sequence=1](https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/235256/651439_FULLTEXT01.pdf?sequence=1) (visited on 10/11/2020).



## A. OpenFOAM BACKGROUND

Unless otherwise noted, the source for information presented in this appendix is the OpenFOAM user's guide [40]. Open Source Field Operation and Manipulation (OpenFOAM) is an open source software package which may be used to develop computational fluid dynamics (CFD) codes. It is based on a framework of C++ libraries and, in addition to the pre-built solvers and utilities, users may create their own applications. OpenFOAM does not have a user interface, so all simulations are coded in a system of directories and text files which specify solvers, constants, and conditions. The overall structure of each case file system is as follows:

```
system
|---> blockMeshDict
|---> controlDict
|---> decomposeParDict
|---> fvSchemes
|---> fvSolution
|---> setFieldsDict
constant
|---> dynamicMeshDict
|---> g
|---> transportProperties
|---> turbulenceProperties
0
|---> alpha.water.orig
|---> p_rgh
|---> U
```

The time directory, typically beginning with zero, defines conditions at the initial time step. The constant directory defines physics, properties, and mesh motion that will be the used throughout the simulation. The system directory contains files which define the mesh, fluid locations, how the solvers will function, and how the simulation will run. Each of these directories as well as the operation of OpenFOAM will be discussed in detail in the following sections.

## A.1 System Directory

### A.1.1 Meshing

The two most used OpenFOAM meshing utilities are blockMesh and snappyHexMesh. The blockMesh utility is best for creating a structured hexahedral mesh. However, all edges must be defined, and it is therefore difficult to use for more complicated geometries. snappyHexMesh may be used to take the original mesh created by blockMesh and make it conform to a given geometry.

The blockMesh utility is controlled by a file called blockMeshDict. In this file, the vertices, blocks, edges, and boundaries are defined. Before these required file sections, the user has the option to define variables which may be referenced throughout the file. This is useful when a value will be used multiple times throughout the file. By defining and referencing the variable, it is then easier for the user to update the variable with lower chances of error. These variables are defined by starting a line with the variable name followed by the value assigned to that variable with at least one space in between. Each line must end with a semi-colon. These variables may be referenced later in the file by typing the dollar sign followed by the variable name.

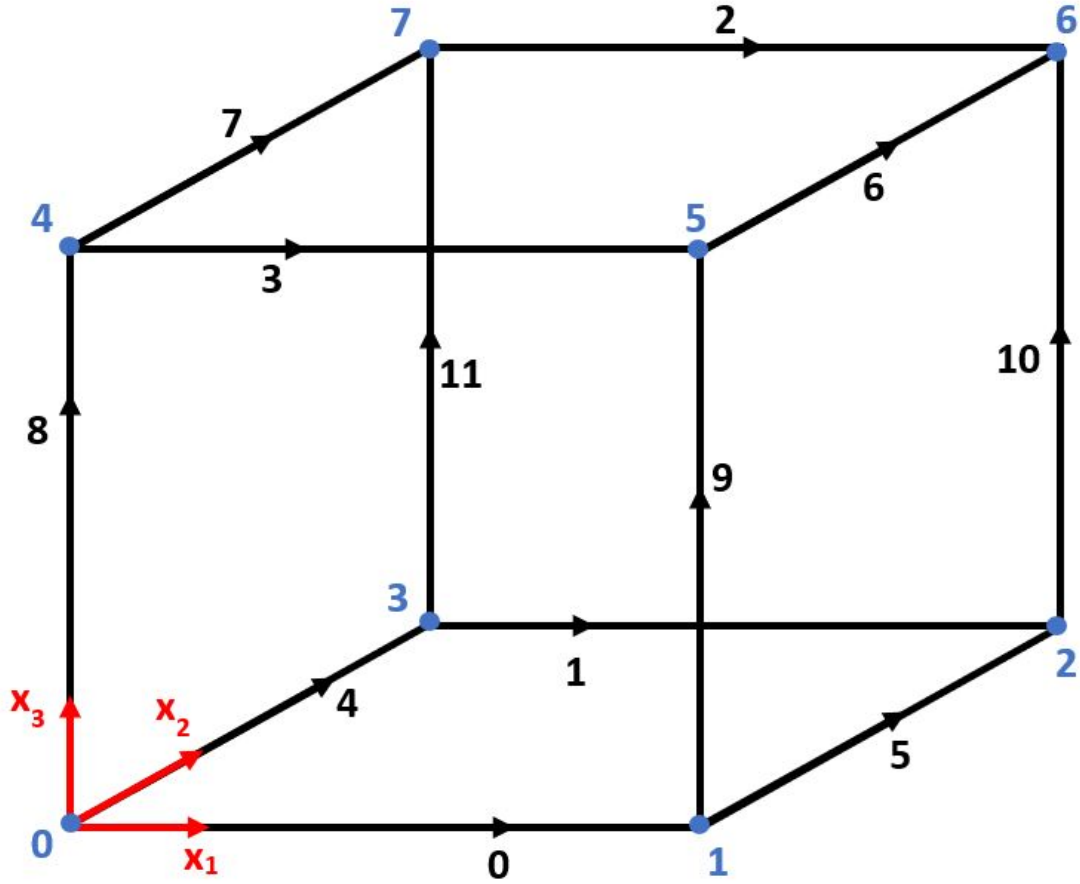
The user may also use calculations within the files. These calculations are started with #calc, followed by the equation in quotation marks. The following example shows the use of variables defined as simple numbers as well as variables defined using calculations based on these defined variables.

```
sloshang    0;           //slosh angle
surang      5;           //angle of the liquid surface
p           10;          //vector length

ix #calc "2*$p*cos(degToRad($sloshang))"; //vector component in x
iy #calc "2*$p*sin(degToRad($sloshang))"; //vector component in y
iz #calc "2*$p*tan(degToRad($surang))";    //vector component in z
```

The first required section of the blockMeshDict file is a list of vertices. This section begins with the word vertices and an open parenthesis. The bulk of the section is a list of vertices in cartesian coordinates. Vertices are numbered in the order which the user lists

them, starting with zero. For ease of use, it is recommended to list the vertices for each block in the order shown in Figure A.1.



**Figure A.1.** Numbering scheme for vertices and edges in blockMesh utility.

Multiple blocks may be defined as needed. Blocks with fewer than eight vertices are possible but are not used in this research so are not discussed here. The vertices section ends with a closed parenthesis and a semicolon. For example, we could write a vertices section as shown below for a rectangular prism with a corner at the origin:

```
vertices
(
(0 0 0) //vertex 0
($Lx 0 0) //vertex 1
($Lx $Ly 0) //vertex 2
(0 $Ly 0) //vertex 3
(0 0 $Lz) //vertex 4
($Lx 0 $Lz) //vertex 5
)
```

```

($Lx $Ly $Lz)    //vertex 6
(0   $Ly $Lz)    //vertex 7
);

```

Some pre-programmed geometries are available to the user to make the definition of vertex and edge locations easier. Some of these geometries include a box, cone, cylinder, disk, rotated box, and sphere. The cylinder is described here due to its use in this research, but the usage for other geometries may be found in the OpenFOAM User Guide online. The cylinder is defined as `searchableCylinder`. There are three needed inputs. Two points are given, each on the axis at one end of the cylinder. The other input is the radius of the cylinder.

```

cylinder
{
    type      searchableCylinder;
    point1    (1.5 1 -0.5);
    point2    (3.5 2 0.5);
    radius    0.05;
}

```

The user may require a point or edge to lie on the cylinder or other geometry by placing the keyword “project” in front of the point or line definition, then following the usual definition with the name of the geometry in parenthesis. An example of a point projected onto the cylinder is given directly below.

```

project ($Lx $Ly $Lz) (cylinder)

```

The edges section contains a list of each edge along with the edge geometry. This section is optional if the vertices were defined according to Figure A.1 and all edges are straight. Each entry in the section starts with the keyword defining the type of edge followed by the two vertices at the start and end of the edge. See Figure A.1 again for the conventional order and direction of edges. The options for edge type include line (the default), arc, spline, polyline, BSpline, and project. The arc option requires a single interpolation point. The two spline options and polyLine require a list of interpolation points. Project will project the point on the edge onto a geometry already defined in the file, such as the cylinder already discussed in this section. The project option requires the geometry the user is projecting

onto to be listed in parenthesis. Edges could be defined as shown below. Note that this is not defining any particular geometry. This simply shows the function of various edge definitions used in this research.

```
edges
(
line 0 1           //edge 0
line 3 2           //edge 1
project 7 6 (cylinder) //edge 2
project 4 5 (cylinder) //edge 3
arc 0 3 (1 3 4)     //edge 4
arc 1 2 (5 3 6)     //edge 5
line 5 6           //edge 6
line 4 7           //edge 7
project 0 4 (cylinder) //edge 8
project 1 5 (cylinder) //edge 9
arc 2 6 (7 2 6)     //edge 10
arc 3 7 (-5 7 6)    //edge 11
);
```

The blocks section consists of a list of each block. These blocks must have vertex numbers, number of cells in each direction, and cell expansion ratios defined. The first part of each block definition is the keyword describing the type of block. In this research, only the hexahedral type block is used (keyword hex). However, other block types including wedge, prism, pyramid, and tetrahedron are possible. Following the block type, a list of the vertices in this block are given within parenthesis. These vertices should again be given in the order shown in Figure A.1. Next, the number of cells along each coordinate direction is given in parentheses.

Finally, the grading along each edge must be given. The two options are simpleGrading and edgeGrading. simpleGrading specifies expansion along each coordinate direction and requires three expansion ratios. edgeGrading specifies a different expansion ratio along each edge and requires a number of expansion ratios equal to the number of edges in the block. The expansion ratio is the ratio of the length of the last cell along the edge to the length of the first cell along the edge. An expansion ratio of one means that the length of each cell along that direction are equal. An expansion ratio of two would mean that the last cell was twice as long as the first cell along that edge. An example of a simple block definition is

given below. This block would have 100 cells in the x, 200 cells in the y, and 300 cells in the z direction. The cells would have equal lengths along each coordinate direction.

```
blocks
(
hex (0 1 2 3 4 5 6 7) (100 200 300) simpleGrading (1 1 1)
);
```

Multi-grading may be used with simpleGrading to create a mesh with expansion ratios that vary along each direction. This may be done by specifying the number of cells and expansion ratio along various specified divisions. In the example below, both the x and z directions use a single expansion ratio, similar to the example above. The y direction is split into three sections. In this example, 20% of the edge length on each side has a non-unity expansion ratio, while the middle 60% has cells of equal length along this direction. This type of grading is especially useful for refining the mesh near the boundary, but not in the center of the mesh.

```
blocks
(
hex (0 1 2 3 4 5 6 7) (100 200 300)
simpleGrading
(
1          //x direction has expansion ratio of 1
(
(0.2 0.25 2.0) //20% of length has 25% of cells, with expansion
                ratio of 2
(0.6 0.50 1.0) //60% of length has 50% of cells, with expansion
                ratio of 1
(0.2 0.25 0.5) //20% of length has 25% of cells, with expansion
                ratio of 0.5
)
2          //z direction has expansion ratio of 2
)
);
```

Boundaries are defined in a section which lists each type of boundary and the corresponding faces. The user may choose the name used to identify each boundary. For each boundary, a type is given. For the purposes of this research, all containers are closed, so only a wall type patch is needed. Other types include patch, symmetry Plane, symmetry, empty, wedge,

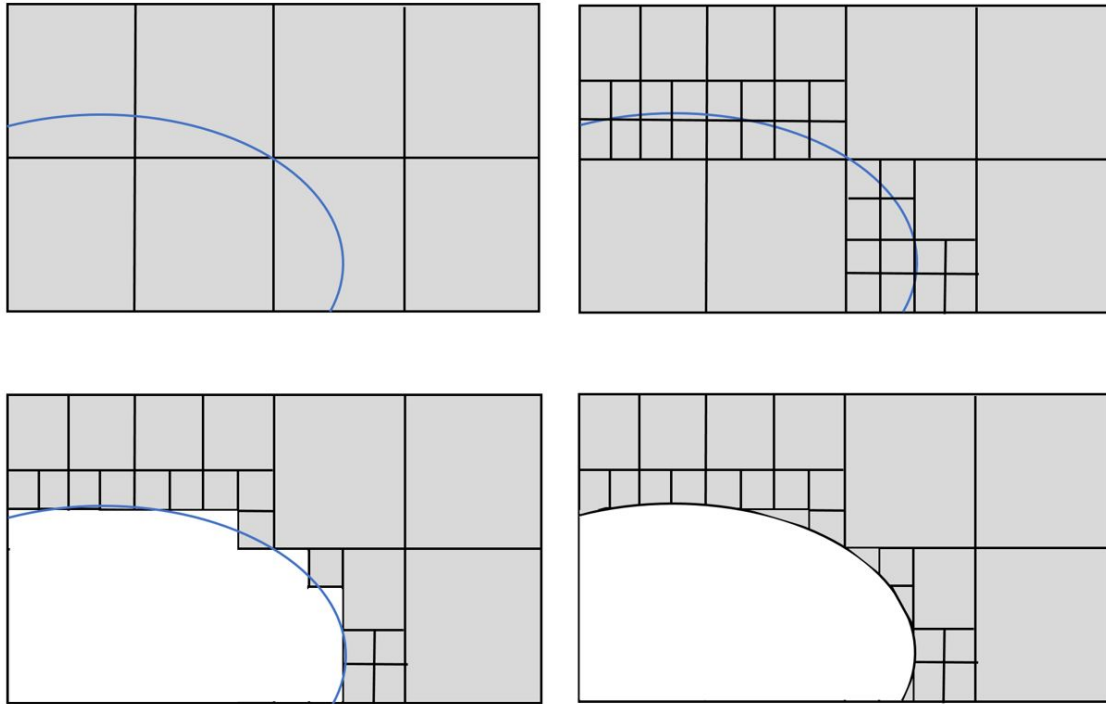
cyclic, cyclicAMI and processor. Following the type definition, a list of the faces included on this boundary are given. Each face is defined by a list of the vertices defining the edge of this face. The vertices should be given in a counterclockwise order when looking from the outside of the block. An example of a wall type boundary useful for this research is given below where the boundary is given the name “walls” by the user.

```
boundary
(
walls      //name of the boundary
{
type wall; //type of boundary
faces
(
(3 0 4 7)  //list of faces in this boundary
(1 2 6 5)
(0 3 2 1)
(4 5 6 7)
(0 1 5 4)
(2 3 7 6)
);
}
);
```

When multiple blocks are defined, some faces must be combined. The blockMesh utility can handle some combinations on its own. Using face matching, the faces which are connected are just left out of the boundary list and blockMesh will determine which faces are collocated and are not an external boundary. These faces are then combined into a single face with the cells in each block connected. Face merging can be used for more complicated geometries, for example when two faces overlap but not completely. This requires a mergePatchPairs section which defines the master and slave patch. The faces on the slave patch are projected onto the master patch. Only face matching is used in this research.

Another available mesher is snappyHexMesh. This utility creates a mesh using a geometry described by an object file from the user’s choice of CAD software. Before use of snappyHexMesh, a background hexahedral mesh must be created. The simplest way to achieve this, and the method used in this research, is to use the blockMesh utility to create a simple rectangular prism which encompasses what will be the final geometry. The snappyHexMesh utility first divides the cells in the region of the background mesh which passes

through the surfaces of the object file. Cells are then removed according to the location the user specifies as internal to the mesh. Cells may then be refined and split again in specified regions. To complete the process, cells are "snapped" to the object surface to remove the jagged edges of the split cells. See Figure A.2 for a visualization of this process.



**Figure A.2.** snappyHexMesh process of creating a background mesh, dividing cells, removing unwanted cells, and snappy boundary points to the object surface.

If the user prefers to use other meshing software, OpenFOAM also has several utilities available for mesh conversion from other formats. It may convert .msh files in ASCII format from Fluent, .geo files from CFX, .ans files in Ansys format from I-DEAS, .neu files from Gambit, and .ccm files from STAR-CCM+.

### A.1.2 controlDict

The controlDict file is where the type of simulation, time stepping, and data writing format is controlled. The first thing defined in this file is the application, the OpenFOAM solver which will be used in the model. For this research, the interFoam solver is used. This



is a volume of fluid solver for two “incompressible, isothermal, and immiscible fluids” [40]. The application is defined as follows:

```
application    interFoam;    //use the interFoam VOF solver
```

Several lines control when the simulation starts and ends. The `startFrom` entry has three options. The `firstTime` option starts the simulation at the earliest time directory, `latestTime` starts the simulation at the latest time directory, and `startTime` begins at a time specified by the `startTime` entry. The `stopAt` entry has four options for ending the simulation. The `writeNow` option stops at the current time step and writes out the data, the `noWriteNow` option stops on the current time step without writing out any of the data, `nextWrite` continues to run the simulation until the end of the next scheduled write step, and `endTime` ends the simulation at a specified time. This time is specified in the `endTime` entry. This section of the control dictionary might be written as follows:

```
startFrom    latestTime;    //Will start from latest time directory
startTime    0;             //Earliest time is zero, not really necessary
                                //for latestTime specification
stopAt        endTime;     //Stop at the time specified by endTime
endTime       30;          //Time at which it will stop
```

The way the simulation writes the data is also defined in the `controlDict` file. The `writeControl` entry tells the simulation when to write out a data file. The options include `timeStep`, which writes every specified number of time steps, and `runTime`, which writes data after a specified amount of simulation time. Other options include `adjustableRunTime`, `cpuTime`, and `clockTime` which are less useful to this research. The `writeInterval` entry either specifies the number of time steps or the amount of simulated time between write steps depending on whether the user chose `timeStep` or `runTime`. If the user wishes to only retain the last several time directories, `purgeWrite` can be used to specify the number of time directories to retain. A value of zero will retain all time directories. The `writeFormat` entry tells OpenFOAM to write in either ASCII or binary format and `writePrecision` is the number of decimal places recorded when writing the data. The `writeCompression` option can be used to compress the data when it's written to a zip file. Time directory names are controlled with `timeFormat` and `timePrecision`. `timeFormat` specifies whether the time

should be written in scientific or standard format. The timePrecision entry is the number of decimal places recorded in the time directory names.

```

writeControl      timeStep;    //writes the time files every nth time
                                //step as given by writeInterval
writeInterval     20;          //number of time steps between each
                                //written time directory
purgeWrite        0;           //do not overwrite any time files
writeFormat       ascii;       //write in ascii format, not binary
writePrecision    6;           //number of decimal places written
writeCompression  off;         //don't compress files
timeFormat        general;     //choose whether scientific and general
timePrecision     6;           //decimal places in time directory names

```

Sometimes it is helpful for the user to be able to edit the controlDict file while running. Turning this on allows the user to change things such as the time at which the simulation ends and is especially useful when first testing out simulations. However, it does take some time for the simulation to reread the controlDict file at the beginning of every time step, so turning this option off may be useful to slightly improve run time.

```

runTimeModifiable yes; //yes means this file is reread and can
                        //be modified while running

```

Time stepping may be as simple as one entry, deltaT which is the time step used in the simulation. However, variable time steps are desired in many applications. This is turned on using the adjustTimeStep entry. In this case, the deltaT is the time step used during the first time step. For an adjustable time step, both the courant/CFL number and the maximum time step must be specified. These are specified in maxCo, maxAlphaCo, and maxDeltaT as shown below:

```

deltaT           0.001; //First time step size
adjustTimeStep   yes;   //Time step
maxCo            0.5;   //Maximum courant/CFL number of everything but
                        //phase fields
maxAlphaCo       0.5;   //Maximum courant/CFL number in phase fields
maxDeltaT        0.001; //Maximum value for delta T using the adjustable
                        //time step

```

### A.1.3 fvSchemes

The fvSchemes file specifies the numerical schemes that will be used in the simulation. The first section defines the discretization of time. The options are steadyState, Euler (first order implicit), backward (second order implicit), CrankNicolson (second order implicit with a specified off-centering coefficient), and localEuler (pseudo transient and first order implicit). The example given below is for the CrankNicolson method with 90% second order and 10% first order implicit.

```
ddtSchemes
{
    default    CrankNicolson 0.9; //90\% Crank Nicolson, 10% Euler
}
```

The method used for calculating the gradient is specified in the gradSchemes section. The default and most commonly used scheme is Gauss linear. Other options include leastSquares (second order) and Gauss cubic (third order). An example using Gauss linear is given below:

```
gradSchemes
{
    default    Gauss linear;    //linear interpolation of values from cell
                                //centers to faces
}
```

The divergence schemes section specifies the discretization and interpolation method to be used for the divergence of specified properties. The only option for the discretization scheme is Gauss. A list a commonly used interpolation schemes is given in Table A.1. A common use of this section is for convection schemes, which transport a property due to the velocity field. For example, a convection scheme specified in this section is  $\nabla \cdot (U\alpha)$  which is coded as div(phi, alpha). This refers to the transport of the volume fraction due to the velocity field.

```
divSchemes
{
    div(rhoPhi,U)      Gauss vanLeerV;    //divergence of rho*U*U
    div(phi,alpha)     Gauss vanLeer;     //divergence of U*alpha
    div(phirb,alpha)   Gauss venLeer;     //divergence of Urb*alpha
    div(((rho*nuEff)*dev2(T(grad(U))))   Gauss linear;    //recommended
```

**Table A.1.** Available OpenFOAM Interpolation Schemes.

Interpolation Scheme	Description
linear	Second order, unbounded
skewLinear	Second order, unbounded, skewness correction
vanLeer	Second order, unbounded
cubicCorrected	Fourth order, unbounded
upwind	First order, bounded
linearUpwind	First/second order, bounded
QUICK	First/second order, bounded
TVD schemes	First/second order, bounded
SFCD	Second order, bounded
NVD schemes	First/second order, bounded

```

}
//scheme by
//user manual

```

The interpolationSchemes section has few options. The linear scheme is used for nearly every simulation except direct numerical simulation and some stress analyses.

```

interpolationSchemes
{
    default    linear;    //linear is the only scheme used except for DNS
                        //and stress analysis
}

```

Surface normal gradient schemes are defined in the snGradSchemes section. This is the “component, normal to the face, of the gradient of values at the centers of the two cells that the face connects.” Options for this scheme include corrected, limited corrected, orthogonal, and uncorrected. The corrected scheme is generally recommended, although uncorrected may be used for meshes with low non-orthogonality.

```

snGradSchemes
{
    default    corrected; //generally recommended scheme
}

```

The laplacianSchemes section is defined based on the interpolation and surface normal gradient schemes. The Gauss scheme must be used for discretization. The interpolation

scheme is almost always linear as discussed, and the `snGradScheme` must be the same as the user chose in that section.

```
laplacianSchemes
{
    default    Gauss linear corrected; //must match interpolation and
                                           //snGradSchemes
}
```

#### A.1.4 fvSolution

The `fvSolution` file contains information about the solvers, algorithms, and relaxation factors used in the simulation. The solvers section specifies how each equation is solved for various terms. This section of the file begins with the keyword “solvers” and is enclosed within braces. For a multiphase problem, the first solver will likely be that of the volume fraction of the fluid of interest. In the example below, the solver is for the volume fraction of water. The number of calculations for volume fraction is specified by `nAlphaCorr` and a value of 1 or 2 is typical for transient simulations. The number of sub-cycles used in calculating the volume fraction per time step is given in `nAlphaSubCycles`, usually with a value between 2 and 4. The compression term in the modified transport equation is given by `cAlpha`. This is deactivated with a given value of 0, conservative compression with a value of 1, and enhanced compression with a value greater than 1 [\[41\]](#).

```
alpha.water          //volume fraction of water
{
    nAlphaCorr        1;    //loops over the volume fraction of water
    nAlphaSubCycles    3;    //sub-calculations of volume fraction of water
                           //per time step
    cAlpha             1.5;  //compression at the interface
}
```

Other solvers are for fields such as those listed in time directories. Table [A.2](#) gives a list of the solvers available.

The GAMG method has several options which may be specified if the default value is not desired. The number of cells in the coarsest mesh size is given by `nCellsInCoarsestLevel` and has a default value of 10. The number of levels which the solver coarsens or refines at once

**Table A.2.** Solvers available in OpenFOAM.

<b>Solver</b>	<b>Description</b>
PCG	Preconditioned Conjugate Gradient Method (symmetric matrices)
PBiCGStab	Stabilized Preconditioned Biconjugate Gradient Method (asymmetric matrices)
PBiCG	Preconditioned Biconjugate Gradient Method (asymmetric matrices)
smoothSolver	Solver that uses a smoothing algorithm
GAMG	Generalized Geometric Algebraic Multi-Grid Method
diagonal	Diagonal Solver (explicit only)

is given by mergeLevels. The default value is one which is the most common. Higher values can only be used for very simple meshes. The user can also turn on or off whether the solver caches the agglomeration strategy (default true) or uses a direct solver when solving the coarsest level (default false) using cacheAgglomeration and directSolveCoarsest respectively.

The user may specify preconditioners using the “preconditioner” keyword. The available preconditioners are given in Table A.3.

**Table A.3.** Available OpenFOAM preconditioners.

<b>Preconditioner</b>	<b>Description</b>
DIC	Diagonal Incomplete Cholesky (symmetric matrices)
DILU	Diagonal Incomplete LU (asymmetric matrices)
FDIC	Fast Diagonal Incomplete Cholesky
diagonal	Diagonal Preconditioning
GAMG	Geometric Algebraic Multi-Grid
none	Do not use a preconditioner

Smoothing algorithms are defined using the “smoother” keyword. The Table A.4 describes the common smoothers used. While quite a few options are available, the most commonly specified option for smoothers is the number of sweeps the algorithm makes before recalculating the residual, nSweeps.

There are three conditions which can be specified as a condition for the solver to stop and continue on to the next step. The solver will stop iterations if any one of the conditions is met, not if all three are met. The first condition is the tolerance, which is the value below which the residual should fall. The maximum number of iterations is given by maxIter. The relative tolerance, keyword “relTol,” is the ratio of current to initial residuals. The solver

**Table A.4.** Available OpenFOAM smoothers.

Smoother	Description
GaussSeidel	Gauss-Seidel
symGaussSeidel	Symmetric Gauss-Seidel
DIC	Diagonal Incomplete Cholesky (symmetric matrices)
DILU	Diagonal Incomplete LU (asymmetric matrices)
DICGaussSeidel	Diagonal Incomplete Cholesky with Gauss-Seidel

will stop if the ratio falls below the specified value. In many cases, this is specified as zero to force the solver to converge to the other two conditions.

```

"pcorr.*" //pressure corrections
{
    solver          PCG;      //use PCG solver
    preconditioner
    {
        preconditioner  GAMG;          //use GAMG as the preconditioner
        tolerance       1e-06;        //residual must reach a value
                                         //below 1e-06
        relTol          0;            //must meet tolerance condition,
                                         //not relative tolerance
        smoother        DICGaussSeidel; //use DICGaussSeidel as a smoother
        cacheAgglomeration no;        //do not cache the agglomeration
                                         //for GAMG
    }

    tolerance         1e-06; //residual must reach a value below 1e-06
    relTol             0;    //must meet tolerance or maxIter condition,
                              //not relative tolerance
    maxIter            100;  //maximum of 100 iterations
}

p_rgh //specifications for first pressure solutions in each time step
{
    solver          GAMG;      //use GAMG solver
    tolerance       1e-08;    //tolerance should reach below 1e-08 or relTol
    relTol          0.01;    //relative tolerance should be less than 0.01
    smoother        DIC;      //use DIC as a smoother
}

p_rghFinal //final pressure solution in each time step
{
    solver          PCG;      //use PCG solver

```

```

preconditioner
{
    preconditioner    GAMG;    //use GAMG as a preconditioner
    tolerance         2e-09;   //residual must fall below 2e-09
    relTol            0;       //must meet tolerance condition,
                                //not relative tolerance
    nVcycles          2;       //number of V-cycles used in the
                                //multigrid method
    smoother          DICGaussSeidel; //use DICGaussSeidel method for
                                //the smoother
    nPreSweeps        2;       //number of sweeps when coarsening
                                //or refining the grid
}

tolerance            2e-09;   //residual should fall below 2e-09
relTol               0;       //must meet tolerance condition,
                                //not relative tolerance
maxIter              20;      //maximum of 20 iterations
}

U    //velocity field
{
    solver            smoothSolver; //use smoothSolver
    smoother          GaussSeidel;  //use the GaussSeidel smoother
    tolerance         1e-06;        //residual should fall below 1e-06
    relTol            0;             //must meet tolerance condition, not
                                //relative tolerance
    nSweeps           1;            //number of sweeps before the residual
                                //is calculated
}

```

The three algorithms used by OpenFOAM are the pressure-implicit split-operator (PISO), semi-implicit method for pressure-linked equations (SIMPLE), and a combination of the two (PIMPLE) algorithms. SIMPLE is primarily used for steady-state problems, while the other two are generally used for transient problems. The `fvSolution` file contains a section which specifies the algorithm run as well as the how the algorithm solves equations. There are five main options which may be set, as well as pressure referencing terms used only for closed incompressible systems. These are described in Table A.5.

```

PIMPLE
{
    momentumPredictor    no;    //do not solve the momentum predictor,

```



**Table A.5.** Keywords for algorithms specifications in OpenFOAM.

Term	Description
momentumPredictor	Turns on or off whether the algorithm solves the momentum predictor. Off is typical for both multiphase and low Reynold's number simulations
nCorrectors	Times the algorithm will solve the pressure equation and momentum corrector within each time step. Typical values are 2 or 3 (only for PISO and PIMPLE algorithms)
nNonOrthogonalCorrectors	Number of solutions to the pressure equation for the non-orthogonal correction. Typical values are 0 or 1.
nOuterCorrectors	Loops over full system of equations within each time step. Typical value is 1 (only for PIMPLE algorithm)
correctPhi	Turns on or off flux correction functions which ensure continuity especially for startup and moving meshes (cite API guide)
pRefPoint	Point within the mesh from which other pressure will be referenced. Necessary for closed incompressible systems where pressure is relative to other points and absolute pressures are not important
pRefValue	Value of the pressure at given pressure reference point. Used to compute all other relative pressures

```

nCorrectors          2;    //typically off for multiphase
                        //times the pressure equation and
                        //momentum corrector are solved
nNonOrthogonalCorrectors 0; //number of solutions to the pressure
                        //equation to update explicit
                        //non-orthogonal correction
nOuterCorrectors     1;    //loops over full system of equations
correctPhi           no;   //no flux correction functions

pRefPoint    (0.15 0.15 0.15); //location in mesh to ref pressure
pRefValue    1e5;              //value of pressure at location
}

```

The relaxationFactors section specifies the under-relaxation used for both fields and equations. Under-relaxation improves the stability but limits the change from one iteration to the next, which slows down the simulation. The relaxation factor must be greater than zero and less than or equal to one. A value of zero would mean that the solution would not

change between iterations and a value of one guarantees that the matrix will be diagonally dominant. A value of one for equations is typical of transient cases.

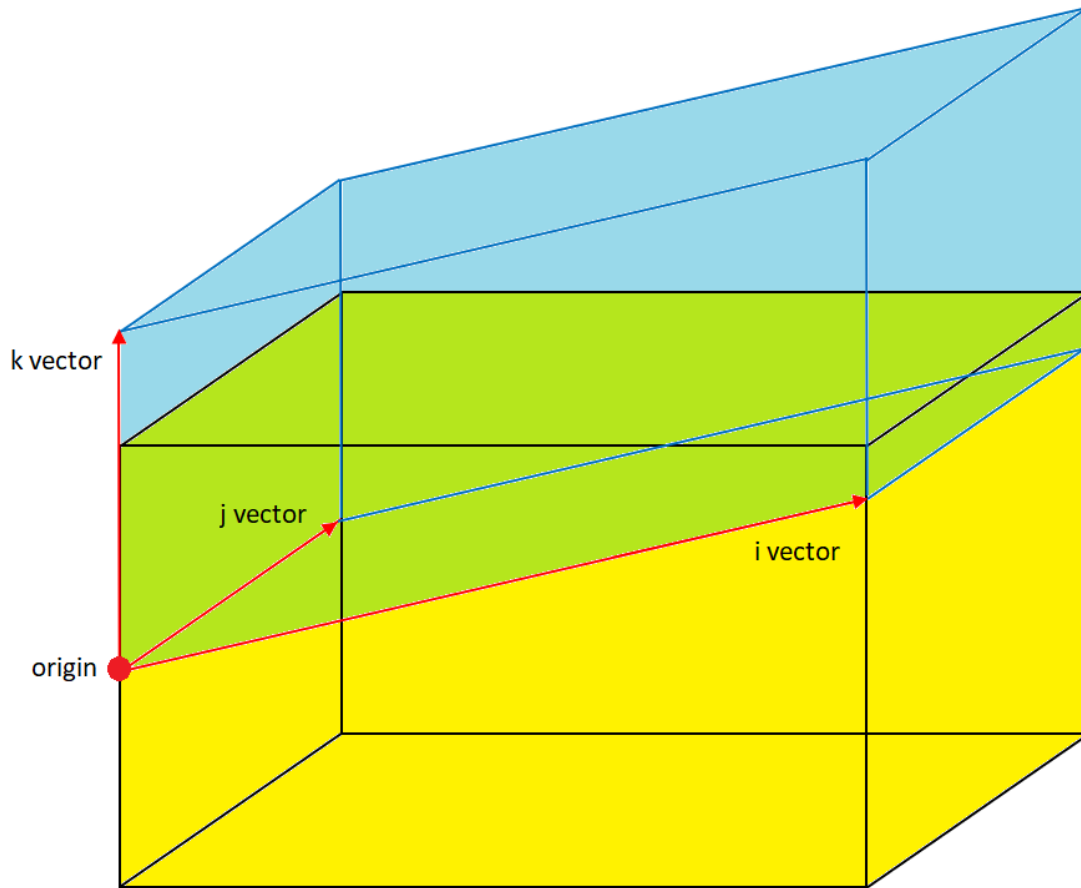
```
relaxationFactors
{
    equations
    {
        "."* 1; //ensure all equations/matrices are diagonally dominant
    }
}
```

### A.1.5 setFields

The setFieldsDict file is required to specify where each of the two phases are located within the mesh. This file references a file in the time directory called alpha.phaseName. The phase name is the name chosen for the liquid (in this example water is used). Fields are assigned by setting the entire mesh to one phase, then specifying a region within the mesh where the other phase is located. The entire mesh is assigned a phase in the defaultFieldValues section. The default value is set to one if the user wants everything to be the phase given in the alpha.phaseName file and set to zero if the user wants to set the entire mesh to be assigned the gas phase. The regions section then specifies areas where the other phase is located. There are many options for specifying the region which may be found in the OpenFOAM User's Manuel under topoSet.

The method most commonly used in this research is the rotatedBoxToCell method. This specifies a parallelepiped within which cells will take the specified fieldValues. For this method, an origin and three edge vectors must be specified as shown in Figure A.3. In the figure, the yellow rectangular prism represents the mesh and the blue parallelepiped is the region defined by the origin and vectors. The parallelepiped may extend beyond the mesh. The volume where the original mesh and the parallelepiped overlap is the region which is set to the specified volume fraction, shown in green in the illustration.

```
defaultFieldValues
(
    volScalarFieldValue alpha.water 1
);
```



**Figure A.3.** Illustration of rotatedBoxToCell region used in setFields.

```
regions
(
    rotatedBoxToCell
    {
        origin (0 0 0);
        i      (1 0 1);
        j      (0 1 0);
        k      (0 0 1);

        fieldValues
        (
            volScalarFieldValue alpha.water 0
        );
    }
);
```

### A.1.6 decomposeParDict

If the simulation will be run on multiple processors, a decomposeParDict file is required. This file specifies the number of processors in the numberOfSubdomains entry and the type of decomposition in the method field. The methods available are simple, hierarchical, scotch, and manual. The simple method splits the domain by coordinate directions and requires a specification of the number of divisions along each direction (n) as well as the cell skew factor (delta). The hierarchical method is the same as the simple method, but the user may specify which coordinate direction should be split first. The scotch method does not require user inputs and tries to minimize the number of processor boundaries itself. Manual decomposition requires the user to specify which cell is assigned to each processor. The user also has the option to distribute the data across multiple disks. This option is turned off and on using the distributed keyword. The simple method of decomposition is used most frequently in this research and an example is given below:

```
numberOfSubdomains 8;      //Number of processors this will be run on
method              simple; //Split domain by coordinate direction
simpleCoeffs
{
  n      (2 2 2); //Division of domain in each coordinate direction
  delta  0.001;   //Default and suggested value for cell skew factor
}
distributed no;    //Not distributed across multiple disks
```

## A.2 Constant Directory

The constant directory contains information about properties that will remain constant throughout the simulation.

### A.2.1 Gravity and Accelerations

The g file defines the gravitational or other constant accelerations on the body. This is defined by specifying the dimensions of the acceleration ( $m/s^2$ ), followed by the value of the acceleration, formatted as a vector. The example below would be a normal gravitational acceleration of  $9.81 m/s^2$  in the negative z direction.

```

Dimensions    [0 1 -2 0 0 0 0];
value         (0 0 -9.81);

```

### A.2.2 Transport Properties

The transportProperties file defines the fluid properties in the model such as viscosity and density. The Newtonian model is the only model used in this research because viscosity is constant. Other available models include the Bird-Carreau, Cross Power Law, Power Law, Herschel-Bulkley, Casson, and general strain rate model. First, each of the phases used in the model are defined. phases (water air);

Each phase is then called by name and the model and properties defined. For the Newtonian transport model, both the kinematic viscosity and the density are required. An example of how water and air phases may be defined is given below:

```

water
{
    transportModel    Newtonian;    //assume kinematic viscosity is constant
    nu                1e-06;        //kinematic viscosity in m2/s
    rho               998.2;        //density in kg/m3
}

air
{
    transportModel    Newtonian;    //assume kinematic viscosity is constant
    nu                1.48e-05;     //kinematic viscosity in m2/s
    rho               1;            //density in kg/m3
}

```

For simulations with more than one fluid, such as the simulations in this research, the surface tension between the two phases is given by sigma:

```

sigma    0.07;    //surface tension in N/m

```

### A.2.3 Turbulence Model

The turbulence model used is specified in the turbulenceProperties file. Laminar turbulence models are generally used for slosh modelling of this nature due to the fact that only the linear regime with low slosh amplitudes is investigated. Other available models

in OpenFOAM include Reynolds Averages Simulation (RAS) and Large Edgy Simulation (LES). The turbulence model is specified as below:

```
simulationType    laminar;
```

#### A.2.4 Dynamic Mesh and Movement

The user is able to specify motion using the dynamicMeshDict file. There are four main types of motion that may be specified as described in the table below. The only motion type used in this research is solidBodyMotionFvMesh.

**Table A.6.** Motion types available in OpenFOAM.

Motion Name	Description
staticFvMesh	No motion (for use in debugging solvers which require a defined mesh motion)
solidBodyMotionFvMesh	Defined solid body motion of the mesh
dynamicRefineFvMesh	Refines sections of the mesh based on specified fields without changing the mesh shape (useful for simulations with small regions of quickly changing variables such as shock waves)
dynamicMotionSolverFvMesh	Mesh motion is calculated based on pressure at specified boundaries

The two solidBodyMotionFvMesh motion functions used in this research were oscillatingLinearMotion and tabulated6DoFMotion. The first specifies oscillating motion along a specified axis. A vector specifying the amplitude and direction of motion as well as the oscillation frequency in radians per second are specified.

```
dynamicFvMesh solidBodyMotionFvMesh;    //use solid body motion
solidBodyMotionFvMeshCoeffs              //begin to specify the solid body motion
{
    solidBodyMotionFunction oscillatingLinearMotion;
    oscillatingLinearMotionCoeffs
    {
        amplitude    (3.0 0.5 2.0); //amplitude and direction of motion
        omega         5.25;         //oscillation frequency (rad/s)
    }
}
```

For more control over the motion, the user may use `tabulated6DoFMotion`. This allows the user to specify the center of gravity of the body around which rotation will occur as well as the location of a data file which specifies all motion.

```
dynamicFvMesh solidBodyMotionFvMesh;
solidBodyMotionFvMeshCoeffs
{
    solidBodyMotionFunction tabulated6DoFMotion;
    tabulated6DoFMotionCoeffs
    {
        CofG    ( 1.0  3.0 2.5 );
        timeDataFileName    " $FOAM_CASE_PATH/constant/datafile.dat ";
    }
}
```

The data file begins with the number of data points in the file. This is followed by a list of points specifying the time and motion. The first vector specifies the total linear displacement in the unit system used by the simulation. The second vector specifies the total angular displacement about the center of gravity in radians. It is important to note that both vectors are total displacement, not relative displacement compared to the last point. The time values are not required to be the same as the time steps because the simulation will linearly interpolate between each given time point.

```
4 //number of data points in the file
(
  //(time    ( (linear displacement)    (angular displacement) ) )
    (0      ( (1 1 1)                    (0.1 0.2 0.3) ) )
    (1      ( (2 2 2)                    (0.2 0.3 0.4) ) )
    (2      ( (2 1 2)                    (0.3 0.4 0.5) ) )
    (3      ( (2 1 1)                    (0.4 0.5 0.6) ) )
)
```

### A.3 Time Directory

The time directories contain information about the current state of the simulation at the time given by the directory name. In general, the user will begin with a time directory named zero. As the simulation runs, more time directories will be created by the simulation based on the write frequency specified in the `controlDict` file. Simulations may be continued

from these later time directories instead of having to start over at the zero-time directory. The fields most commonly used in this research are discussed here, although others available for other simulation types.

### A.3.1 Velocity

The U file defines the initial conditions for velocity. The internal field specifies the value throughout the mesh. Specifying that this is uniform means that the value is the same in every cell. For a fluid that is not initially moving, the initial condition for the internal field would be (0 0 0). The boundary field specifies the velocity boundary conditions. In this research, the only boundary type is a wall. Boundary condition options include slip, noSlip, partialSlip, translatingWallVelocity, rotatingWallVelocity, movingWallVelocity, fixedNormalSlip, and fixedValue. Most past slosh studies have used no slip boundary conditions.

```
dimensions      [0 1 -1 0 0 0 0];
internalField    uniform (0 0 0);
boundaryField
{
    walls
    {
        type      noSlip;
    }
}
```

### A.3.2 Phase Fraction

Alpha files define the boundary between the fluid and the walls. These files must be named as alpha.phaseName using the same phase name given in the transportProperties file. For example, if the phase name is water, the alpha file must be named alpha.water.orig. The internal field assigns the volume fraction of this phase to all cells. For multiphase flow with a setFieldsDict, this internal field value has little affect. The value assigned will be overwritten when the setFields command is run. The boundary field is where the boundary condition is given. Options for this include zeroGradient (90-degree contact angle) and constantAlphaContactAngle (specified contact angle).



```

dimensions      [0 0 0 0 0 0 0];
internalField    uniform 0;
boundaryField
{
    walls
    {
        type      zeroGradient;    //90-degree contact angle
    }
}

```

### A.3.3 Pressure

In multiphase simulations such as these, hydrostatic pressure has an impact on the pressure within each cell. Because of this, the initial pressure is defined using the file `p_rgh` instead of `p`. The definition of `p_rgh` is the total pressure minus the hydrostatic pressure  $\rho gh$ . The internal field value is the value at which internal cells are set and may be set as uniform or nonuniform. The boundary field sets the condition at the boundary. Some options for pressure include `fixedValue`, `fixedFluxPressure`, and `fixedFluxExtrapolatedPressure`. In this research, primarily `fixedFluxPressure` is used because this condition “sets the pressure gradient to the provided value such that the flux on the boundary is that specified by the velocity boundary condition” [40].

```

dimensions      [1 -1 -2 0 0 0 0];
internalField    uniform 0;
boundaryField
{
    wall
    {
        type      fixedFluxPressure;    //pressure gradient adjusted to fit
                                           //velocity boundary condition
    }
}

```

## A.4 Running Simulations

The simulation is usually run by first creating the mesh, setting the volume fraction field, deconstructing the domain for use on multiple processors, and then running the solver. These commands are all run in the Linux terminal. The mesh is created using “`blockMesh`”,

followed by “snappyHexMesh” if the user requires this utility. The volume fraction fields are set by running “setFields”. If running the simulation on multiple processors, the user then runs “decomposePar”. To run the simulation on multiple processors, the command “mpirun -np [numberProcessors] interFoam -parallel” is run. If the user just wants to run the simulation on a single processor, simply “interFoam” may be used. To output the results to a file instead of the command window, the user may append “> logName” to the end of any of these commands. Typically, the log name convention is log.commandName. For example, the following series of commands may be run.

```
blockMesh > log.blockMesh
setFields > log.setFields
decomposePar > log.decomposePar
mpirun -np 16 interFoam -parallel > log.interFoam
```

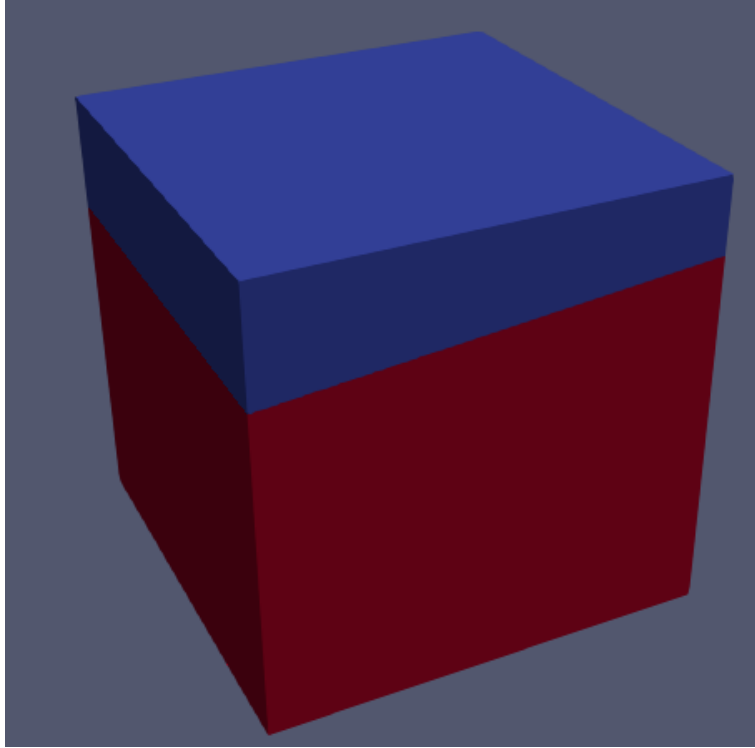
When running the setFields utility, OpenFOAM rewrites the alpha.water.orig file to a new alpha.water file which specifies the volume fraction for each cell. If the user needs to clean and restart a case, the alpha.water file should be deleted before setFields is run again. Otherwise the alpha.water file will be used instead of the alpha.water.orig file.

## A.5 Post-Processing

### A.5.1 Para Foam

The most used post-processing utility is paraFoam. Running this command in the Linux terminal will open the OpenFOAM case in Paraview. Paraview can be used to view visualizations of various field values such as volume fractions, create plots, view the liquid-gas interface, and more. It may also be used before running the simulation to check that setFields has run as expected. For example, Figure A.4 shows an example of a paraFoam view used to check that the liquid-gas interface initialized at an angle as expected.

If the user needs to view the simulation in Paraview on a different computer system, the simulation may be packaged for viewing by running the command foamToVTK. This creates a VTK directory which can be transferred and opened in Paraview from any system. This is helpful to the user because the VTK directory takes up significantly less space than the entire case directory.



**Figure A.4.** ParaFoam view of a cube container with the surface initialized at an angle.

### A.5.2 Foam Log

The foamLog command may be used to create a directory of logs containing information about the residuals for various fields. This command is run on the interFoam log. If using the standard log naming method, the command “foamLog log.interFoam” would create the desired directory and files.

### A.5.3 Function Objects

Function objects are OpenFOAM post-processing tools which may be used either at run-time or after the simulation has finished. In general, it is preferred to use function objects at run-time so the simulation may be monitored and to reduce the required number of write directories. If a function object is run after the simulation completes, it may only provide information at times with a written directory. If run with the simulation, a function object may provide information at any time step, not just those where a time directory is written.

Some time directories are desired for paraFoam visualization, but fewer are needed than for function objects.

OpenFOAM has many built in function objects, such as forces and residuals. Because OpenFOAM is open-source, it is also possible for the user to write a new function object.

Function objects are defined in a "functions" section at the end of the controlDict file. For example, if one wished to use only the forces function object, the function section might look like the following:

```
functions
{

forces
{
type forces;
libs ("libforces.so");
patches (walls);
rho rhoInf;
log yes;
rhoInf 1000;
CofR (0 0 0);
}

}
```

## B. CENTER OF MASS FUNCTION OBJECT

### B.1 CM.C

```
/*-----*\
=====
\\      /   F i e l d      |   OpenFOAM: The Open Source CFD Toolbox
\\      /   O p e r a t i o n      |   Website:  https://openfoam.org
\\      /   A n d      |   Copyright (C) 2020 OpenFOAM Foundation
  \\    M a n i p u l a t i o n      |
-----
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

\*-----*/

#include "CM.H"
#include "Time.H"
#include "fvMesh.H"
#include "addToRunTimeSelectionTable.H"
#include "fvCFD.H"
#include "OSspecific.H"
#include <iostream>

// * * * * * Static Data Members * * * * * //

namespace Foam
{
namespace functionObjects
{
    defineTypeNameAndDebug(CM, 0);
    addToRunTimeSelectionTable(functionObject, CM, dictionary);
}
```

```

}
}

// * * * * * Constructors * * * * * //

Foam::functionObjects::CM::CM
(
    const word& name,
    const Time& runTime,
    const dictionary& dict
)
:
    fvMeshFunctionObject(name, runTime, dict),
    logFiles(obr_, name)
{
    read(dict);
    resetName(name);
}

// * * * * * Destructor * * * * * //

Foam::functionObjects::CM::~CM()
{}

// * * * * * Member Functions * * * * * //

void Foam::functionObjects::CM::writeFileHeader(const label i)
{
    //Print out a file header with what each number is
    file() << "Time(s)" << " " << "totalVol" << " " << "waterVol" << " " <<
    "CMx" << " " << "CMy" << " " << "CMz" << endl;
}

bool Foam::functionObjects::CM::read(const dictionary& dict)
{
    return true;
}

bool Foam::functionObjects::CM::execute()
{
    return true;
}

bool Foam::functionObjects::CM::end()

```

```

{
    return true;
}

bool Foam::functionObjects::CM::write()
{
    #include "volFields.H"
    #include "uniformDimensionedFields.H"

    //Assign names to the quantities we want
    //Volume of each cell in the mesh
    auto volume = mesh_.V();
    //Volume fraction of each cell in the mesh
    const volScalarField& alpha = mesh_.lookupObject<volScalarField>("alpha.water");
    //Position of each cell in the mesh
    auto position = mesh_.C();

    //Calculate the values we want on each processor
    //Volume of the mesh
    double volsum = sum(volume).value();
    //Volume of water
    double watervol = sum(volume*alpha.internalField()).value();
    //Sum of the volume*volume_fraction*cell_position
    auto waterposit = sum(volume*alpha.internalField()*position).value();
    //Extract x component of sum of the volume*volume_fraction*cell_position
    double waterpositX = waterposit.component(0);
    //Extract y component of sum of the volume*volume_fraction*cell_position
    double waterpositY = waterposit.component(1);
    //Extract z component of sum of the volume*volume_fraction*cell_position
    double waterpositZ = waterposit.component(2);

    //Create a variable proci which is the number of the current processor
    const label proci=Pstream::myProcNo();

    //Create lists which have the length equal to the number of processors
    List<double>volList(Pstream::nProcs());
    List<double>watervolList(Pstream::nProcs());
    List<double>waterXList(Pstream::nProcs());
    List<double>waterYList(Pstream::nProcs());
    List<double>waterZList(Pstream::nProcs());

    //Assign each variable a different position in the list based on processor
    volList[proci]=volsum;
    watervolList[proci]=watervol;
    waterXList[proci]=waterpositX;

```

```

waterYList[proci]=waterpositY;
waterZList[proci]=waterpositZ;

//Collapse the lists onto one processor
Pstream::gatherList(vollList);
Pstream::gatherList(watervollList);
Pstream::gatherList(waterXList);
Pstream::gatherList(waterYList);
Pstream::gatherList(waterZList);

//On the master processor
if (Pstream::master())
{
    //Assign a name to the time step value
    auto& runTime = fileObr_.time();

    logFiles::write();

    //Calculate the sum of each value over all of the processors
    double totalVol = sum(vollList)/(Pstream::nProcs());
    double waterVol = sum(watervollList)/(Pstream::nProcs());
    double volX = sum(waterXList)/(Pstream::nProcs());
    double volY = sum(waterYList)/(Pstream::nProcs());
    double volZ = sum(waterZList)/(Pstream::nProcs());

    //Center of mass is the
    //sum(volume*position*volume_fraction)/sum(volume*volume_fraction)
    double CMx = volX/waterVol;
    double CMY = volY/waterVol;
    double CMz = volZ/waterVol;

    //Write the variables we want to the postprocessing log file
    file() << runTime.timeName() << " " << totalVol << " "
    << waterVol << " " << CMx << " " << CMY << " " << CMz << endl;
}

return true;
}

// ***** //
```



## B.2 CM.H

```
/*-----*\
=====
\\      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
\\      /  O p e r a t i o n      | Website:  https://openfoam.org
\\      /  A n d      | Copyright (C) 2020 OpenFOAM Foundation
  \\    M a n i p u l a t i o n      |
-----
```

### License

This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with OpenFOAM. If not, see <<http://www.gnu.org/licenses/>>.

### Class

Foam::functionObjects::CM

### Description

This function object...

Example of function object specification:

```
\verbatim
CM1
{
    type          CM;
    libs ("libCMFunctionObject.so");
    ...
    wordData      someWord;
    scalarData    1.0;
    labelData     1;
}
\endverbatim
```

### SourceFiles

```

CM.C

/*-----*/

#ifndef CM_H
#define CM_H

#include "fvMeshFunctionObject.H"
#include "logFiles.H"

// * * * * *

namespace Foam
{
namespace functionObjects
{

/*-----*\
                        Class CM Declaration
\*-----*/

class CM
:
    public fvMeshFunctionObject,
    public logFiles
{

protected:
    virtual void writeFileHeader(const label i);

public:

    //- Runtime type information
    TypeName("CM");

    // Constructors

    //- Construct from Time and dictionary
    CM
    (
        const word& name,
        const Time& runTime,
        const dictionary& dict
    );

```

```

        //- Disallow default bitwise copy construction
        CM(const CM&) = delete;

    //- Destructor
    virtual ~CM();

    // Member Functions

        //- Read the CM data
        virtual bool read(const dictionary&);

        //- Execute, currently does nothing
        virtual bool execute();

        //- Execute at the final time-loop, currently does nothing
        virtual bool end();

        //- Write the CM
        virtual bool write();

    // Member Operators

        //- Disallow default bitwise assignment
        void operator=(const CM&) = delete;
};

// * * * * *

} // End namespace functionObjects
} // End namespace Foam

// * * * * *

#endif

// *****

```

## C. PARAMETER CALCULATION

### C.1 Logarithmic Decrement

The logarithmic decrement can be used to find the damping ratio,  $\zeta$ , of a damped sinusoid using Equations C.1 and C.2 where  $A_n$  is the amplitude of a peak.

$$\delta = \frac{1}{n} \ln \frac{A_n}{A_{n+1}} \quad (\text{C.1})$$

$$\zeta = \frac{1}{\sqrt{1 + (\frac{2\pi}{\delta})^2}} \quad (\text{C.2})$$

When  $\zeta \ll 1$ , C.3 may be used [6].

$$\zeta = \frac{1}{2\pi} \ln \frac{A_n}{A_{n+1}} \quad (\text{C.3})$$

### C.2 Curve Fitting

Another way to find the damping ratio is by curve fitting an equation to the data. The equation for a damped sinusoid is given by the following where  $\zeta$  is the damping ratio and  $\omega_n$  is the natural frequency.

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = 0 \quad (\text{C.4})$$

We know that the solution will be of the type:

$$x(t) = Ce^{st}$$

$$\dot{x}(t) = Cse^{st}$$

$$\ddot{x}(t) = Cs^2e^{st}$$

So now we can plug that in to get:

$$Cs^2e^{st} + 2\zeta\omega_n Cse^{st} + \omega_n^2 Ce^{st} = 0 \quad (\text{C.5})$$

Cancel out the C and exponential:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \quad (\text{C.6})$$

Now we can solve for what s should be, which is:

$$s = \frac{-2\zeta\omega_n \pm \sqrt{(2\zeta\omega_n)^2 - 4\omega_n^2}}{2} \quad (\text{C.7})$$

$$s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \quad (\text{C.8})$$

We will be underdamped for any situation we expect to be in, so  $\zeta < 1$  and we can rewrite the equation as:

$$s = -\omega_n(\zeta \pm i\sqrt{1 - \zeta^2}) \quad (\text{C.9})$$

Therefore, the full equation will be:

$$x(t) = Ce^{-\omega_n(\zeta \pm i\sqrt{1 - \zeta^2})t} \quad (\text{C.10})$$

This may also be written as:

$$x(t) = Ce^{-\omega_n\zeta t} [\cos(\omega_n\sqrt{1 - \zeta^2}t) + i \sin(\omega_n\sqrt{1 - \zeta^2}t)] \quad (\text{C.11})$$

Only the real or imaginary part is required so we will use the sine:

$$x(t) = Ce^{-\omega_n\zeta t} \sin(\omega_n\sqrt{1 - \zeta^2}t) \quad (\text{C.12})$$

For simplicity's sake and to allow better curve fitting, this equation can be written as the following where  $\omega_d = \omega_n\sqrt{1 - \zeta^2}$  and  $b = \omega_n\zeta$ .

$$x(t) = Ce^{-bt} \sin(\omega_d t) \quad (\text{C.13})$$

The values obtained from curve fitting are  $\omega_d$  and  $b$ . To obtain the damping ratio and natural frequency, the two values we actually want, we can solve for those in terms of the values obtained by curve fitting.

$$\zeta = \sqrt{\frac{b^2}{\omega_d^2 - b^2}} \quad (\text{C.14})$$

$$\omega_n = \frac{\omega_d}{\sqrt{1 - \zeta^2}} \quad (\text{C.15})$$

For very small damping ratios,  $\zeta \ll 1$ , the natural frequency is approximately the damped natural frequency. The damping ratio may then be approximated as the ratio of the damping constant to the damped natural frequency.

$$\omega_n \approx \omega_d \quad (\text{C.16})$$

$$\zeta = \frac{b}{\omega_n} \approx \frac{b}{\omega_d} \quad (\text{C.17})$$

## D. NON-DIMENSIONALIZATION

While primarily dimensional parameters are used in this research, such as frequency and surface height, others may wish to use non-dimensional parameters. This appendix presents a method for non-dimensionalizing the parameters used in this research. Equation D.1 is the basic equation governing the change in the surface height at any particular location along the surface.

$$x(t) = Ce^{-bt} \sin(\omega t) \quad (\text{D.1})$$

We can define the following, where  $*$  denotes a non-dimensional quantity, as well as  $\zeta$  which denotes damping ratio, a non-dimensional quantity. Damping ratio is given by Equation D.2, non-dimensional frequency by Equation D.3, non-dimensional surface height by Equation D.4, non-dimensional constant Equation D.5, and non-dimensional time by Equation D.6.

$$\zeta = \frac{b}{\omega} \quad (\text{D.2})$$

$$\omega^* = \omega \sqrt{\frac{L}{g}} \quad (\text{D.3})$$

$$x^* = \frac{x}{L} \quad (\text{D.4})$$

$$C^* = \frac{C}{L} \quad (\text{D.5})$$

$$t^* = t \sqrt{\frac{g}{L}} \quad (\text{D.6})$$

Inserting these non-dimensional quantities, we get Equation D.7 which simplifies to Equation D.8

$$x^* L = C^* L e^{-\zeta \omega^* \sqrt{\frac{g}{L}} t^* \sqrt{\frac{L}{g}}} \sin \left[ \omega^* \sqrt{\frac{g}{L}} t^* \sqrt{\frac{L}{g}} \right] \quad (\text{D.7})$$

$$x^* = C^* e^{-\zeta \omega^* t^*} \sin [\omega^* t^*] \quad (\text{D.8})$$