



# **Computational Modeling of Aeroelastic Characteristics of Transonic Wing with Sloshing in an Embedded Fuel Tank**

*By*

**Shashank Srivastava**

*(B.Tech., IIT Patna)*

**A THESIS  
SUBMITTED FOR THE DEGREE OF**

**Doctor of Philosophy  
Department of Mechanical Engineering  
National University of Singapore**

**2019**

Supervisor

**Professor Boo Cheong Khoo**

Reviewed by

**Dr Zhang Huangwei**  
Assistant Professor  
NUS

**Dr Teo Chiang Juay**  
Associate Professor  
NUS

**Dr Liu Feng**  
Professor  
University of California

## Declaration

I hereby declare that this thesis entitled "*Computational Modeling of Aeroelastic Characteristics of Transonic Wing with Sloshing in Embedded Fuel Tank*" submitted by me for the partial fulfillment of the **Doctor of Philosophy** degree from National University of Singapore, is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Date: 9<sup>th</sup> December 2019

Signature



Shashank Srivastava

A0146556X

## Abstract

Aeroelasticity is the study of the interaction between structural dynamics and unsteady flow around the structure. It is an important design consideration for aircrafts as it governs flight performance and stability. Liquid fuel carried by transonic flights sloshes upon structural excitation, and in turn modifies the aeroelastic behavior of the vehicle. The effects of fuel sloshing on aeroelasticity is relatively unexplored in open literature, largely because of the absence of computational tools for this multiphysics problem. This study focuses on the development of computational models for assessment of fuel sloshing effects on aeroelasticity of transonic flights.

Firstly, a coupling methodology is developed which utilizes existing open-source CFD solvers to obtain high-fidelity numerical solutions. An existing open-source solver interface is modified to enable interaction between the single physics solvers participating in the coupling via data exchange during runtime. However, the coupling is computationally expensive and impractical for time marching solutions such as flutter boundary problems that require multiple simulation runs. The computational cost of the coupled problem is reduced by representing sloshing by an Equivalent Mechanical System (EMS), which comprises of a series of mass-spring systems to simulate linear sloshing. A surrogate model based on Radial Basis Function Neural Network (RBF-NN) is developed to efficiently describe the dominant nonlinear dynamic characteristics of sloshing in an enclosed vessel. Both EMS and RBF-NN sloshing modes are coupled with CFD based aeroelastic solver to compute the transonic flutter boundary of an airfoil.

The study is extended to the development of machine learning surrogate models for both aerodynamics and sloshing. Data-driven recurrent neural network based surrogate models are utilized for construction of a computational framework to assess aeroelastic motion of an airfoil and a wing embedded with partially filled fuel tanks. Finally, a Physics-Informed Neural Network (PINN) is developed to alleviate the data-dependence of black-box surrogate models. PINN incorporates the governing equation of the underlying physical system in the learning process. In this work, PINN is substituted for the aeroelastic equation of motion for demonstration of its implementation and prediction accuracy.

## Acknowledgments

I owe thanks to a number of people who have helped and supported me in shaping my work and research presented in this thesis. First and foremost, I want to thank my advisors Prof Boo Cheong Khoo and Dr Murali Damodaran for their continuous guidance and motivation throughout PhD study. Their immense knowledge and patience has helped me in my research and improved my technical writing skills. They have taken the pain to go through various details of my numerical experiments and taught me the nitty-gritty of fluid mechanics, aeroelasticity and numerical computations as and when needed.

This PhD thesis would not have been possible without the continuous support of my labmates in the Aeroelasticity and Flow Control Group. I owe thanks to my colleague Mr Rahul Halder for the countless discussions we have had during the course of my PhD. His technical insights and ideas have been invaluable. My deep appreciation goes to Dr Saeil Lee who always gave me good advice whenever I found myself struggling with my codes, and especially when I was getting started with open-source numerical solvers. I also want to thank my colleagues Mr. Yu Qijing and Mr. Eng Yong who have always welcomed me whenever I needed help. Their high quality of work ethic and research has always pushed me to do better work.

I gratefully acknowledge the funding received towards my PhD from the NUS Research Scholarship via the Department of Mechanical Engineering. I owe thanks to Prof Shu Chang who has always motivated me in my research. Thanks to A/Prof Senthil Kumar who motivated me to explore various research fields within Mechanical Engineering and collaborative work when he interviewed me in New Delhi, India for admission as a PhD candidate at NUS. Many thanks to NUS HPC and their staff for facilitating faster computations essential to my study and always helping me with software related issues. I also want to thank the ME staff at NUS who has always been helpful for providing and managing computational resources. I want to

thank Fluid Mechanics Lab staff members Ms Lee Cheng Fong Carolyn, Mr. Rajendran and Mr Looi Siew Wah for their help in lab sessions and my duties as a Teaching Assistant.

I owe a great debt of gratitude to my alma mater, Indian Institute of Technology Patna. My sincere thanks to the professors in the Department of Mechanical Engineering, especially Dr Akhilendra Singh and Dr Md Kaleem Khan, for their efforts in teaching and overall guidance whenever I strayed from the path of learning.

I am also very grateful to all those at Hydroinformatics Institute, Singapore for hosting me as an intern for almost eight months. I owe thanks Prof Vladan Babovic, Mr Abhishek Saha, Mr Srinesh and Dr Serene Tay for helping me with my projects there.

I am indebted to my family scattered throughout the world. My Father and Mother, having doctorates in Economics and Music, respectively, have always encouraged me to pursue PhD. A heartfelt thanks to my aunt Reshma, my grandfather Girish and grandmother Asha for always encouraging and motivating me. Special thanks to my uncle Theodore who has always been curious about my work and helped me during my conference in California and took me all around the state. Thanks to my siblings Shubhangi, Ritwick, and Sakshi for always believing in me. And a big thank you to my partner Pooja, without whom I would not have had the courage to cross the finish line of my PhD.

And finally, my friends who have always been by my side through thick and thin. A special thanks to Abhinav Tripathi for being there for me since the beginning of my bachelor's at IIT Patna and decided to take this journey of PhD at NUS with me. Animesh Prasad, Harshit Shah, and Vishwaraj Anand, who have been close friends since my days at IIT Patna, followed by being the best neighbors in Singapore, and still going strong. Last but not least, the wonderful friends I have made for life at NUS Singapore, especially Dr Fahid Riaz, Dr Ravi Ranjan, and

Pratibha Baveja. They have been a constant source of encouragement and this PhD would not be possible without them.

# Table of Contents

	<i>Page No.</i>
<b>List of Tables</b>	iv
<b>List of Figures</b>	v
<b>List of Abbreviations and Symbols</b>	xi
<b>1. Introduction</b>	1
1.1 Introduction and Background	1
1.2 Motivation for this Research	13
1.3 Contributions of this Thesis	14
<b>2. Computational Models</b>	18
2.1 Aerodynamic Model	18
2.2 Fluid Sloshing	22
2.2 A. Volume of Fluid Method	23
2.2 B. Equivalent Mechanical System	25
2.3 Aero-Structural Model	32
<b>3. Integrated Computational Framework</b>	38
3.1 Coupling Methodology	40
3.2 SU2 Adaptation for Coupling	51
3.3 OpenFOAM Adaptation for Coupling	55
<b>4. Machine Learning Surrogate Models</b>	61
4.1 Principles of Dynamic Systems Modeling	61
4.2 Radial Basis Function Neural Network	64
4.3 Recurrent Neural Network	73
4.4 Integrated Frameworks	80

4.5 Physics-Informed Neural Network	87
<b>5. Results and Discussions</b>	<b>90</b>
5.1 Flutter Boundary of NACA64A010 with RBF-NN Based Sloshing Model	90
5.1 A. Grid Dependence Studies for the Inviscid Aeroelastic Solver within SU2	91
5.1 B. Validation of Compressible Flow Solver SU2 for the Inviscid Flutter Analysis of NACA64A010	92
5.1 C. Unsteady Sloshing Force Predictions Using Surrogate Model and Validation Using CFD	94
5.1 D. Airfoil Flutter Analysis Using the Computational Framework	96
5.2 Flutter Boundary of NACA64A010 with EMS Model for Sloshing	100
5.3 Sloshing Effects on Aeroelastic Motion of NAC0012	108
5.3 A. Validation of Compressible Flow Solver for Steady and Unsteady Flow for NACA0012	108
5.3 B. Validation of Multiphase Flow Solver for Sloshing Flow in Fuel Tank	111
5.3 C. Pertinent Parameters of Interest in the Machine Learning Surrogates	112
5.3 D. Single Step Prediction Test – Aero-Structural Surrogate Model	116
5.3 E. Single Step Prediction Test for RNN Surrogate Model for Tank Fuel Sloshing Loads Prediction	122
5.3 F. Prediction of Aeroelastic Responses Using RNN	125
5.4 Sloshing Effects on Aeroelastic Motion of NASA BSCW	133
5.4 A. Validation of Inviscid Compressible Solver for NASA BSCW	133
5.4 B. RNN Surrogate Model Validation for External Flow with Parametric Variation	140
5.4 C. Prediction of Coupled Aeroelastic Motion of BSCW with Sloshing Tank	144

5.4 D. Computational Cost Analysis of High Fidelity CFD and Surrogate Predictive Model	146
5.5 Physics Informed Neural Network for Aeroelastic Equation of Motion	147
5.5 A. Aeroelastic Equation Parameters in Loss Function	148
5.5 B. Data Samples for Training Structural PINN	149
5.5 C. Structural Displacements Prediction Results of PINN	150
<b>6. Conclusions and Future Works</b>	152
<b>Bibliography</b>	156

## List of Tables

	<i>Page No.</i>
<b>Table 5.1:</b> Computational Cost Analysis of High Fidelity CFD and Surrogate Predictive Model	132
<b>Table 5.2:</b> Structural Properties of NASA Benchmark Supercritical Wing	145
<b>Table 5.3:</b> Computational Cost Analysis of High Fidelity CFD and Surrogate Predictive Model	147
<b>Table 5.4:</b> Structural Parameters of NACA0012	148

## List of Figures

	<i>Page No.</i>
<b>Figure 1.1:</b> Aeroelastic (Collar's) triangle of forces	1
<b>Figure 1.2:</b> A F-16 with external stores containing liquid fuel which is prone to slosh (Source: aviation.stackexchange.com/aviation/f16_with_conformal_fuel_tanks)	6
<b>Figure 1.3:</b> (a) A 2-DOF airfoil with an embedded tank immersed in an external flowfield, and (b) Interaction of structural solver with aerodynamic and sloshing solvers	9
<b>Figure 1.4:</b> Solvers interaction via an interface by exchanging data	10
<b>Figure 1.5:</b> A Radial Basis Function Neural Network for prediction of sloshing loads	15
<b>Figure 1.6:</b> Recurrent neural network for prediction of aerodynamic load coefficients with flow parameterization	16
<b>Figure 2.1:</b> Spatial discretization of computational domain	20
<b>Figure 2.2:</b> Partially filled fuel tank with normal and shear forces acting on the tank walls due to sloshing fluid	24
<b>Figure 2.3:</b> Half-filled fuel tank with tank dimensions and reference axes used for derivation of equivalent mechanical system parameters	29
<b>Figure 2.4:</b> Equivalent mechanical system (EMS) consisting of mass-spring system to represent sloshing loads of a partially filled fuel tank	30
<b>Figure 2.5:</b> A 2-DOF airfoil immersed in a flowfield free to plunge and pitch	33
<b>Figure 2.6:</b> A 2-DOF airfoil with a partially filled embedded fuel tank immersed in a flowfield free to plunge and pitch	35
<b>Figure 3.1:</b> Coupling structure of compressible flow solver for aerodynamics and multiphase solver with structural solver	39
<b>Figure 3.2:</b> A simple comparison of partitioned and monolithic schemes for the present problem	40
<b>Figure 3.3:</b> Staggered coupling of aeroelastic solver and sloshing solver in time	47
<b>Figure 3.4:</b> Coupling library <i>preCICE</i> for runtime solver interactions data transfer	48
<b>Figure 3.5:</b> Aeroelastic motion of NASA BSCW immersed in transonic flow and under the influence of sloshing fuel in internal fuel tank coupled using <i>preCICE</i>	49
<b>Figure 3.6:</b> (a) Coupling strategy of <i>SU2</i> and <i>OpenFOAM</i> via an interface, and (b) Coupling structure of <i>SU2</i> and <i>OpenFOAM</i> flow solvers using <i>preCICE</i>	51

<b>Figure 3.7:</b> (a) Representation of configuration file for solver interface <i>precice-config.xml</i> , and (b) block diagram representation of the configuration files for <i>preCICE</i> and <i>SU2</i>	53
<b>Figure 3.8:</b> Computation of displacement of the elastic axis of the airfoil in terms of pitch and plunge from displacement information of surface points of the airfoil.	54
<b>Figure 3.9:</b> (a) Coupling directory containing participant solvers, and (b) <i>OpenFOAM</i> working directory with <i>preCICE</i> adapter	57
<b>Figure 3.10:</b> Code snippet added to <i>controlDict</i> file in <i>OpenFOAM</i> to activate the coupling	58
<b>Figure 4.1:</b> A radial basis function neural network with $N$ inputs and $M$ neurons in the hidden layer that produces an output $\tilde{y}$	64
<b>Figure 4.2:</b> Functional approximation of noisy data using radial basis function networks	65
<b>Figure 4.3:</b> Different beta values depicting differences in the spread of gaussian-type function	67
<b>Figure 4.4:</b> ARX based network with input delay-order of 2 and output delay-order of 1	69
<b>Figure 4.5:</b> Radial basis function neural network with ARX-type architecture containing time history of previous inputs and outputs of the prediction network	70
<b>Figure 4.6:</b> (a) White noise signal used to generate APRBS and cut off higher frequencies, (b) Power spectral density comparison of APRBS and white noise signals, (c) APRBS with level change modulation, and (d) Amplitude response to excitation frequencies comparison between APRBS with and without level change parameters	72
<b>Figure 4.7:</b> Recurrent neural network cells in abstract and unfolded representation	74
<b>Figure 4.8:</b> Activation functions (a) linear, (b) sigmoid, (c) tanh, (d) ReLU, (e) leaky ReLU, and (f) ELU	76
<b>Figure 4.9:</b> Densely connected deep recurrent neural network in multiple-input multiple-output configuration for (a) the overall aeroelastic model, and (b) sloshing loads in the fuel tank model	77
<b>Figure 4.10:</b> Densely connected deep recurrent neural network in multiple-input multiple-output configuration for the aeroelastic model with parameterized flow conditions	80
<b>Figure 4.11:</b> Computational framework with radial basis function neural network based sloshing surrogate for studying effects of sloshing on NACA64A010 flutter boundary	81
<b>Figure 4.12:</b> Computational framework with recurrent neural network based surrogates for studying effects of sloshing on aeroelastic airfoil in transonic flow regime	83
<b>Figure 4.13:</b> Flowchart showing the coupling of aero-structural model for both forced and free aeroelastic motions	85

<b>Figure 4.14:</b> Computational framework with recurrent neural network based surrogates for studying effects of sloshing on aeroelastic motion of BSCW in transonic flow regime	86
<b>Figure 4.15:</b> Physics-informed neural network for 2-DOF aeroelastic equation	88
<b>Figure 5.1:</b> (a) CL vs. 1/N Variation, and (b) Grid 3 mesh in the vicinity of the wing section	91
<b>Figure 5.2:</b> (a) Neutral response $M_\infty = 0.825$ , $V_f = 0.612$ , (b) Damped response $M_\infty = 0.85$ , $V_f = 0.439$ , and (c) Diverging (flutter) response $M_\infty = 0.875$ , $V_f = 1.420$	92-93
<b>Figure 5.3:</b> Flutter boundary computation and comparison with Alonso and Jameson [17]	94
<b>Figure 5.4:</b> (a) The structural input provided as pitching angle to fuel tank, (b) comparison of forces on east wall using CFD output and surrogate model prediction, (c) comparison of forces on west wall using CFD output and surrogate model prediction, (d) zoomed-in comparison of forces on east wall and corresponding errors, and (e) zoomed-in comparison of forces on west wall and corresponding errors.	95-96
<b>Figure 5.5:</b> (a) Computed Mach contours for steady state flow past NACA64A010 at $M_\infty = 0.80$ and $V_f = 0.70$ , (b) Unsteady flowfield at the end of 2 forced pitching cycles, (c) Wing Section in $\pi/4$ position in the 5th free cycle going into flutter, and (d) Wing Section at the end of 5th free cycle	97-98
<b>Figure 5.6:</b> (a) Volume fraction of fuel in rectangular container (50% fill level), (b) Free surface at the end of 2 forced pitching cycles, and (c) Tank at $\pi/4$ position in the 5th free cycle, and (d) Tank at the end of 5 <sup>th</sup> free cycle	98
<b>Figure 5.7:</b> Flutter boundary of NACA64A010 with and without fuel sloshing effects modeled by CFD and RBF-NN based surrogate model	99
<b>Figure 5.8:</b> (a) Normalized sloshing modal mass for first 8 sloshing modes, (b) Modal spring constants, and (c) Modal (natural) frequencies of sloshing modes	101-102
<b>Figure 5.9:</b> Cumulative modal forces with increasing number of sloshing modes	102
<b>Figure 5.10:</b> Force amplitude response to forcing frequency	103
<b>Figure 5.11:</b> Flutter boundary of NACA64A010 with and without the effects of fuel sloshing	104
<b>Figure 5.12:</b> Lateral forces acting on the tank walls as a result of fuel sloshing due to pitching motion of the fuel tank at $f = 15.91$ Hz using CFD and potential flow solution	105
<b>Figure 5.13:</b> Lateral sloshing forces on tank walls for forced pitching with increasing amplitude	105
<b>Figure 5.14:</b> Sloshing forces and moment time-history obtained from CFD and Fourier analysis for combined pitch-plunge motion with $\omega=3.14$ rad/sec and (a) $\alpha = 1^\circ$ , $h = 2.5\%$ , (b) $\alpha = 2^\circ$ , $h = 5.0\%$ , (c) $\alpha = 4^\circ$ , $h = 10.0\%$ , and (d) $\alpha = 6^\circ$ , $h = 15.0\%$ of tank height	106

<b>Figure 5.15:</b> Sloshing forces and moment time-history obtained from CFD and Fourier analysis for combined pitch-plunge motion with plunging amplitude $h = 5.0\%$ of tank height, pitching amplitude $\alpha = 4^\circ$ and forcing frequency of (a) $\omega=3.14$ rad/sec, (b) $\omega=6.28$ rad/sec, (c) $\omega=12.56$ rad/sec, and (d) $\omega=25.12$ rad/sec	107
<b>Figure 5.16:</b> (a) Mach contours of NACA0012 airfoil at $M_\infty=0.80$ and $\alpha = 1.25^\circ$ , (b) Coefficient of pressure distribution of the upper and lower airfoil surfaces, (b) Coefficient of pressure distribution of the upper and lower airfoil surfaces, (c) Lift coefficient variation with angle of attack for harmonic pitching comparison of present study with published results, and (d) Moment coefficient variation comparison with published data for the same motion	109-110
<b>Figure 5.17:</b> Comparison of (a) dynamic pressure, and (b) transverse sloshing forces obtained from present study with experimental and numerical measurements of Yusong et al.	111
<b>Figure 5.18:</b> Mean squared error variation with training epochs	113
<b>Figure 5.19:</b> (a) One step prediction of $C_L$ based on input sequence with batch length 40, (b) Prediction of $C_M$ for batch length 40, (c) $C_L$ prediction with batch length 30, (d) $C_M$ prediction with batch length 30, (e) $C_L$ prediction with batch length 20, and (f) $C_M$ prediction with batch length 20	114
<b>Figure 5.20:</b> (a) One step prediction of $C_L$ for RNN with 92 and 60 neurons in 2 hidden layers, (b) Corresponding $C_M$ prediction, (c) One step prediction of $C_L$ for RNN with 62 and 40 neurons in 2 hidden layers, and (d) Corresponding $C_M$ prediction	116
<b>Figure 5.21:</b> (a) Pitching displacements of the airfoil in free motion with a random batch of inputs for testing the RNN, (b) Corresponding variation of lift and moment coefficients; Variation of (c) MSE, and (d) MSE (in log scale) vs. epochs during network training for aerodynamic data.	117-118
<b>Figure 5.22:</b> The training instance, the expected output and the network prediction for lift coefficient using randomly chosen batch for (a) lift, and (b) moment coefficients	119
<b>Figure 5.23:</b> Time history of (a) plunging and pitching displacements of the airfoil in free motion with a random batch of inputs for testing the RNN, (b) Corresponding lift and moment coefficients; Variation of (c) MSE, and (d) MSE (in log scale) vs. epochs during network training for aerodynamic data	120
<b>Figure 5.24:</b> (a) The training instance, the expected output and the network prediction for lift coefficient using randomly chosen batch, and (b) Same parameters shown for moment coefficient	121
<b>Figure 5.25:</b> (a) Pitch and plunge motion time series structural inputs for tank motion, and (b) Sloshing loads in response to tank motion; Variation of (c) MSE, and (d) MSE (in log scale) vs. epochs during network training for aerodynamic data.	123
<b>Figure 5.26:</b> The training sample, expected values and prediction values of surrogate model for, (a) lateral force, (b) vertical force, and (c) moment	124
<b>Figure 5.27:</b> (a) Plunging and pitching displacements of the airfoil in free motion, (b) corresponding lift and moment coefficients obtained for the free motion of the airfoil (c) phase plot for plunging, and (d) phase plot for pitching	126

<b>Figure 5.28:</b> (a) Plunging data divided into test-train sets for free-aeroelastic motion of NACA0012 airfoil, and (b) pitching data divided into test-train sets for the same motion	127
<b>Figure 5.29:</b> (a) Airfoil displacements and lift and moment coefficients as training data inputs, testing targets and surrogate model prediction outputs, (b) phase plot of comparison of lift coefficient against plunge, (c) phase plot comparison of moment coefficient against pitch (d) phase plot comparison of moment coefficient against pitch, and (e) phase plot comparison of moment coefficient against plunge	127-128
<b>Figure 5.30:</b> Sloshing loads due to tank motion in form of forces and moments as training data inputs, testing targets and surrogate model prediction outputs	129
<b>Figure 5.31:</b> (a) The effective lift obtained by coupling sloshing tank with the aeroelastic wing section using <i>RNN</i> based predictive model, (b) Corresponding moment coefficient, (c) Comparison of plunge motion of airfoil computed from surrogate model and high-fidelity CFD, and (d) Comparison of pitching motion of airfoil computed from surrogate model and high-fidelity CFD	130-131
<b>Figure 5.32:</b> (a) NASA Benchmark Supercritical Wing as 2-DOF plunge and pitch wing, (b) BSCW in wind tunnel, (c) Pressure transducers at 60% span length of the wing, and (d) Cross-section at 60% span length	134
<b>Figure 5.33:</b> (a) Mach number contours on upper surface of BSCW, (b) Mach number contours on lower surface, (c) Coefficient of pressure contours on wing surface and 60% span, (d) Mach number contours on wing surface and 60% span, (e) Comparison of $c_p$ at 60% span with numerical and experimental data, and (f) Mach number distribution at 60% span length	135
<b>Figure 5.34:</b> Steady state Mach number contours on BSCW surface and cross-section at span length (a) 20%, (b) 40%, (c) 60%, (d) 80%, and (e) 100%	136
<b>Figure 5.35:</b> (a) Lift coefficient, and (b) Drag coefficient and moment coefficient for BSCW OTT forced pitching at 10 Hz, (c) Lift coefficient vs angle of attack for forced pitching motion, and (d) Drag coefficient vs angle of attack for forced pitching motion	137
<b>Figure 5.36:</b> Mach number contours for BSCW OTT forced pitching of a cycle at 10 Hz at (a) 0 deg, (b) 90 deg, (c) 180 deg, (d) 270 deg, and (e) 360 deg	138
<b>Figure 5.37:</b> (a) $c_p$ vs spatial grids at 60% span of BSCW for 1 cycle of motion divided into 36 steps, (b) FRF Magnitude OTT Case 1 at $M_\infty = 0.70$ , and (c) FRF Phase OTT Case 1 at $M_\infty = 0.70$	139
<b>Figure 5.38:</b> (a) Lift coefficient of BSCW for pitching motion, (b) Moment coefficient of BSCW for pitching motion; Variation of (c) MSE, and (d) MSE (in log scale) vs. epochs during network training for aerodynamic data	141
<b>Figure 5.39:</b> (a) Sample data red to RNN at Mach 0.70, (b) The training instance, the expected output and the network prediction for lift coefficient at $M_\infty = 0.70$ , (c) The training instance, the expected output and the network prediction for moment coefficient at $M_\infty = 0.70$ , (d) The training instance, the expected output and the network prediction for lift coefficient at $M_\infty = 0.73$ , and (e) The training instance, the expected output and the network prediction for moment coefficient at $M_\infty = 0.73$	142-143

<b>Figure 5.40:</b> Training data, testing data and RNN prediction of (a) Lift coefficient at $M_\infty = 0.70$ , (b) Moment coefficient at $M_\infty = 0.70$ ; Testing data and RNN prediction of (c) Lift coefficient at $M_\infty = 0.73$ , and (d) Moment coefficient at $M_\infty = 0.73$	144
<b>Figure 5.41:</b> (a) Aeroelastic plunging motion of BSCW with and without sloshing at $M_\infty = 0.72$ , and (b) Aeroelastic pitching motion of BSCW with and without sloshing at $M_\infty = 0.72$	146
<b>Figure 5.42:</b> Aeroelastic motion profiles NACA0012 for training structural PINN at $M_\infty = 0.60$ and flutter speed index (a) $V_f = 0.400$ , (b) $V_f = 0.420$ , (c) $V_f = 0.425$ , and (d) $V_f = 0.500$	149
<b>Figure 5.43:</b> Comparison of aeroelastic motion of NACA0012 computed by CFD and predicted by PINN for (a) plunging motion, and (b) corresponding pitching motion; (c) plunging motion, and (d) corresponding pitching motion; (e) plunging motion, and (f) corresponding pitching motion	150-151

## List of Abbreviations and Symbols

<i>ALE</i>	Arbitrary Eulerian-Lagrangian
<i>JST</i>	Jameson-Schmidt-Turkel
<i>DLM</i>	Doublet Lattice Method
<i>VLM</i>	Vortex Lattice Method
<i>LCO</i>	Limit Cycle Oscillation
<i>CFD</i>	Computational Fluid Dynamics
<i>MAC</i>	Marker and Cell
<i>SPH</i>	Smoothed Particle Hydrodynamics
<i>VOF</i>	Volume of Fluid
<i>POD</i>	Partial Orthogonal Decomposition
<i>ROM</i>	Reduced Order Model
<i>MLP-NN</i>	Multi-Layer Perceptron Neural Network
<i>DOF</i>	Degree of Freedom
<i>BSCW</i>	Benchmark Supercritical Wing
<i>EMS</i>	Equivalent Mechanical System
<i>RBF-NN</i>	Radial Basis Function Neural Network
<i>ARX</i>	Autoregressive Exogenous
<i>APRBS</i>	Amplitude-modulated Pseudo-Random Binary Signal
<i>RNN</i>	Recurrent Neural Network
<i>PINN</i>	Physics-Informed Neural Network
<i>FFT</i>	Fast Fourier Transform
<i>mse</i>	Mean-squared error
$\rho$	Fluid density
$\vec{V}$	Fluid velocity
$E$	Total energy
$H$	Total enthalpy
$p$	Fluid pressure
$\mu$	Fluid viscosity
$\dot{\vec{u}}_m$	Mesh velocity
$\vec{U}$	Conservative variables

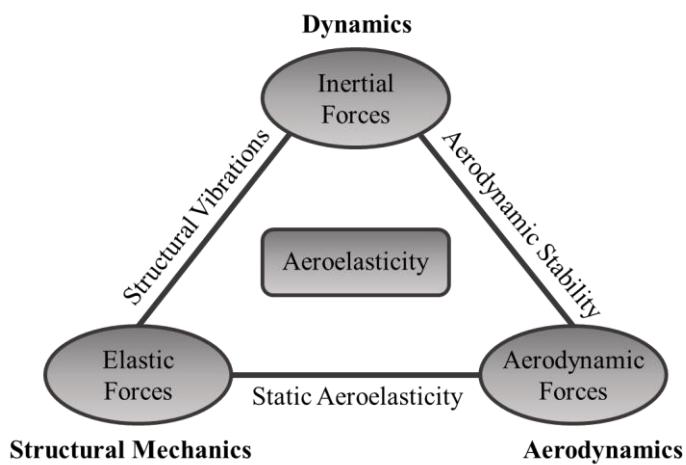
$\vec{F}$	Flux terms
$s_{i,j}$	Cell area
$\vec{f}_B$	External body forces due to gravity per unit mass
$\vec{f}_v$	External body forces due to tank motion per unit mass
$f$	Fluid fraction in multiphase sloshing
$\Phi$	Velocity potential
$\delta(x,t)$	Free surface displacement
$m_f$	Frozen mass in Equivalent Mechanical Systems
$h$	Plunging motion
$\alpha$	Pitching motion
$c$	Chord length of airfoil/wing
$b$	Semi-chord length
$S_a$	Static imbalance
$I_\alpha$	Moment of Inertia of airfoil about elastic axis
$r_\alpha$	Radius of gyration
$C_L$	Coefficient of lift
$C_M$	Coefficient of moment
$c_p$	Pressure coefficient
$\mu_{mass}$	Reduced mass
$V_f$	Flutter speed index
$F_{aero}$	Loads on airfoil/wing due to aerodynamics
$F_{slosh}$	Loads on tank imbedded in airfoil/wing due to sloshing fluid
$M_\infty$	Free-stream Mach number

# Chapter 1

## Introduction

### 1.1 Introduction and Background

Aeroelasticity is the study of the interaction between fluids and structure immersed in it. The interaction between the inertial forces, elastic forces, and aerodynamic forces during the exposure of an elastic structure to fluid flow. The study of the interaction of flexible structures with steady-state aerodynamics is termed as *static aeroelasticity*. It involves the deformation of the structure due to flow loads. In the context of aerospace problems, it involves phenomenon like static divergence and control reversal in case of wings with ailerons or other control surfaces. However, when interaction with inertial forces comes into the picture, much more interesting dynamic phenomena occur which collectively falls under the term *dynamic aeroelasticity*, which will be simply referred to as aeroelasticity from hereon. The interaction of the three interdisciplinary activities namely dynamics, structural mechanics and aerodynamics can be summarized in the Collar's aeroelastic triangle shown in Fig. 1.1.



**Figure 1.1: Aeroelastic (Collar's) triangle of forces**

Aeroelastic implications are common in fields like civil engineering, e.g. stability of tall structures and bridges; mechanical engineering, e.g., blades of turbomachinery, flow

interaction with subsea pipelines; etc., however, the advancement of the field has been centered around aeronautical applications. Aeroelastic interactions determine the flight performance of any aircraft. It determines the aerodynamic loads created by the lifting surfaces and their interaction with the motion of the structure. Estimation and accurate prediction of aeroelastic stability of aircraft has been an age-old problem. The crash of Langley's monoplane attributed to torsional divergence in 1903, only eight days before the Wright brothers' first successful flight, is probably the earliest noted failure due to an aeroelastic problem. Aeroelastic characteristics remain an important consideration even today, especially for modern high-speed aircrafts. The principle aeroelastic phenomena of interest are the dynamic instability, also termed as 'flutter', and response to various other dynamic loads originated from external sources such as gusts or fuel sloshing. Flutter is one of the most dramatic aeroelastic phenomena that can lead to large amplitude motion and often catastrophic failure of the structure. The earliest noted incidence of flutter was in Handley Page O/400 by coupling of two structural modes in 1916. The problem was solved by interconnecting the aircraft elevators by a torque by as detailed by Lancaster [1]. Wing-aileron flutter was soon encountered, followed by primary surface flutter in 1925 and servo-tab flutter predicted by Collar [2] in the 1930s up until as late as 1956. The detailed history of flutter occurrences is detailed in the works of Tolve [3] and Garrick [4].

Recent research has focused on reductions in fuel consumption by aircrafts. This has a direct consequence on the reduction of the structural weight, as well as, increased wing flexibility. Further, research driven by the desirability of higher flight speeds has made transonic and supersonic flights not only possible but affordable for catering to the masses. These two effects combined, intensifies the interaction between the aerodynamics and structures and leads to problems such as excessive gust loading, unacceptable flutter margins, and limit cycle oscillations. Bulk of commercial flights operate in the transonic regime, typically ranging

between Mach numbers of 0.7 and 0.9. In this range of Mach number, the flow is highly nonlinear and dominated by the presence of moving shock waves. This results in a phenomenon called ‘transonic dip’, which adversely affects the stability of the aircraft as a sharp drop in the flutter boundary is experienced. This also led to the emergence of new aeroelastic problems such as transonic control surface buzzing [5]. The aeroelastic stability of the aircraft has also known to be affected by the presence of external stores. The external stores carried by high-speed aircrafts such as F-16, F-18, and F-111, when present in a certain combination, produce a post flutter phenomenon, limit cycle oscillations (LCO) as reported by Norton [6, 7]. LCO is characterized by almost constant small-amplitude harmonic vibrations. However, instances of high amplitude vibrations have been recorded and it remains an undesirable phenomena. The aeroelastic characteristics are also sensitive to external loading on the wing structures such as gusts, as well as intricate phenomenon such as shock-boundary layer interaction [8] in high Mach number flows. To the present day, the transonic flutter remains one of the most active research areas and a lot remains to be explored.

Aeroelasticity is the study of the interaction of aerodynamics, structure, and inertia and hence the solutions to all three facets comprising aeroelasticity are critical. The most challenging part of solving the aeroelastic motion of a structure is the computation of aerodynamic loads and it makes sense to understand and review the methods for solving the fluid dynamics component of aeroelasticity. The aerodynamic flow characteristics are highly sensitive to flow conditions and hence appropriate solution methodology is required for accurate and efficient results. Classical aeroelastic theory considers linear nature of aerodynamics and reduces the problem to a set of linear equations that can be solved easily [9]. Tang [10] used vortex lattice method (VLM) to account for incompressible aerodynamics which was coupled with nonlinear structure. Conyers [11] used doublet lattice method (DLM) to model the aerodynamics to analyze the effects of a hole on the aeroelasticity of a rectangular wing. However, at higher

flow speeds in high-subsonic and transonic regime, these linear theory leads to inaccurate solutions. Phenomenon like shocks, transonic dip, and flow separation cannot be captured by these methods.

Aeroelastic phenomenon can result from both linear and non-linear interaction of structures and fluids. Significant attention has been given to aeroelastic phenomenon such as static divergence, flutter, and control reversal in the past few decades. Various time-domain and frequency-domain methods have been developed to predict the flutter boundary of aeroelastic systems such as Harmonic Balance method and continuation method. However, these methods are mostly confined to a linear assumption of aerodynamics and/or structures.

The progress in the field of computational fluid dynamics in the last three decades cannot be overlooked. Solution accuracy, robustness, and availability of solvers in the open-source make CFD as one of the most popular and versatile tools to compute aerodynamic flows in a wide range of flow conditions. A framework that utilizes CFD for aerodynamics solutions is desirable for aeroelastic applications but the computational cost incurred by time marching aeroelastic solutions acts as a barrier for its widespread usage. However, the improvements in computing resources and the advent of smaller and faster machines have encouraged time-domain CFD solutions of aeroelastic computations. Bendiksen and Kousen [12, 13] used Euler equations coupled with a structural integrator to compute the flutter boundary for NACA64A010 airfoil. Guruswamy [14, 15] utilized Euler and Navier-Stokes equations for 2-D and 3-D aeroelastic computations. The efficiency of CFD solutions using Euler equations was improved by Jameson [16] by developing a multi-grid driven implicit approach. He used central difference in space and temporal implicit multistep discretization and used steady-state marching in pseudo time to obtain converged solutions at physical time steps. CFD codes based on Alonso and Jameson's [17] work were developed at Stanford University and are available

in an open-source suite, *SU2* [18]. It offers an efficient tool to compute aeroelastic solutions using Euler and Navier-Stokes equations for flow solutions.

Fuel sloshing has shown to play an important role in determining the dynamics of the enclosing vessel. Sloshing refers to the motion of fluid excited by motion of containing vessel. Essentially the free surface of liquid must be disturbed from its initial state of rest in a partially filled container for sloshing to occur. Carrier vessels ranging from large scale marine LNG carriers, ground vehicles, to aircrafts with partially fuel tanks are prone to the effects of sloshing on its motion and dynamics. Marine LNG carrier industry has enjoyed prolonged attention of the research community and substantial research exists to efficiently design container vessels with sloshing loads taken into consideration. Coupled ship-tank motion has been investigated using models test in the last decade by Nasar et al. [19] and Nam and Kim [20]. Many numerical studies including finite-differencing method to simulate sloshing flow in anti-rolling tank as well as using impulse-response-function formulation to study coupling effects on linear ship motion are done by Kim [21,22]. Abrahamson's [23] work on sloshing effects on the motion of space vehicles was one of the earliest to simplified mechanical models to represent the dynamics of sloshing. Sloshing of fuel in an enclosed tank attached to an aircraft wing has known to affect the aeroelastic characteristics of the wing. Smaller high-speed aircrafts designed for high maneuverability such as F-16 have external stores as fuel tanks, shown in Fig. 1.2. Larger commercial aircrafts contain liquid fuels inside the wing structure. Although studied to a lesser extent, there have been some important studies on the effects of sloshing on the aeroelastic motion of the wing. One of the earliest attempts on studying the effects of fuel sloshing on aircraft motion was published by Luskin and Lapin [24] in 1952. The effects of sloshing on upper subsonic and transonic flights have been recognized by the aircraft design community such as Cazier et al. [25], Farhat [26], Chiu and Farhat [27], Firdous-Abadi [28] and Hall [29]. The approaches taken by the researchers vary in the modeling of aerodynamics,

as well as sloshing flows. Simplified methods such as hydro-elastic models and linear models have been used by Farhat [26] and Firdous-Abadi [28]. Smoothed particle hydrodynamics method [30] has been used by Banim [31] to model fuel sloshing in the wing, but applied gust as the forcing function for the wing instead of aerodynamic loads. Despite the efforts of these researchers, comparably lesser attention has been given to the study of the effects of sloshing on aeroelastic motion in the open literature. It is important to review and understand methods for solving sloshing flows in an enclosed tank before delving into the details of the solution of coupled aeroelasticity with fuel sloshing.



**Figure 1.2: A F-16 with external stores containing liquid fuel which is prone to slosh (Source: [aviation.stackexchange.com/aviation/f16\\_with\\_conformal\\_fuel\\_tanks](https://aviation.stackexchange.com/aviation/f16_with_conformal_fuel_tanks))**

Accurate prediction of sloshing behavior of fluids partially occupying the volume of a container subjected to motion has been a long-standing problem for researchers. The dynamic response of sloshing fluid interacting with the container depends on multiple factors such as shape of the container, type of motion imparted, e.g. rectilinear, rotatory, oscillatory, etc., frequency and amplitude of motion, fill level of the container, physical properties of the fluid and structural properties of the containing vessel such as rigidity, porosity, etc. The fluid response can vary from simple planar, nonplanar, rotational, irregular beating, symmetric, asymmetric, quasi-

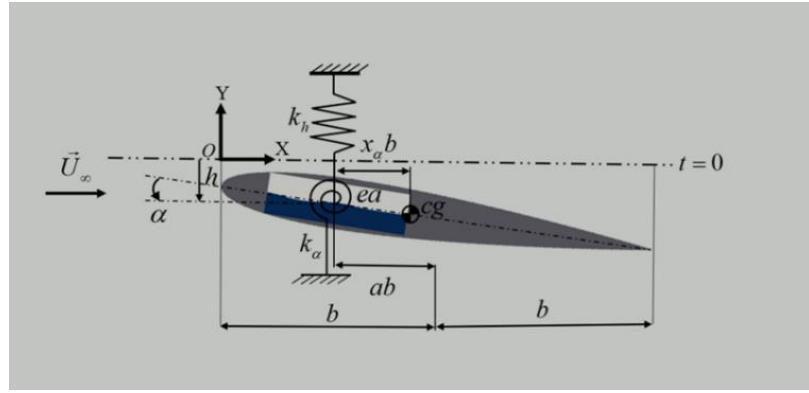
periodic to chaotic. Sloshing dynamics are dominated by surface tension and the free surface may break into multiple blobs of liquid in case of micro-gravity conditions as in space vessels.

Luskin and Lapin [24] used Graham's [32] method of representing oscillating fuel in a rectangular fuel by an equivalent system consisting of a pendulum and a fixed mass which produces the same loads. This idea was further developed and extended to cover a variety of shapes of the tank. It utilized linearized flow equations to compute sloshing loads for small excitation of the tank. Later an Equivalent Mechanical System (EMS) consisting of configurations of mass-spring systems was developed to represent the motion of oscillating fluid, with each set of mass-spring system corresponding to a sloshing mode computed from linear potential theory formulations. This is a simple yet powerful method for modeling sloshing flow as a combination of different sloshing modes. Ibrahim's [33] work provides the detailed derivation of the EMS parameters and corresponding forces and moments generated for a few standard motions of containing tank. This method is restricted to linear domain when excitation frequencies are not close to natural frequency of any of the sloshing modes.

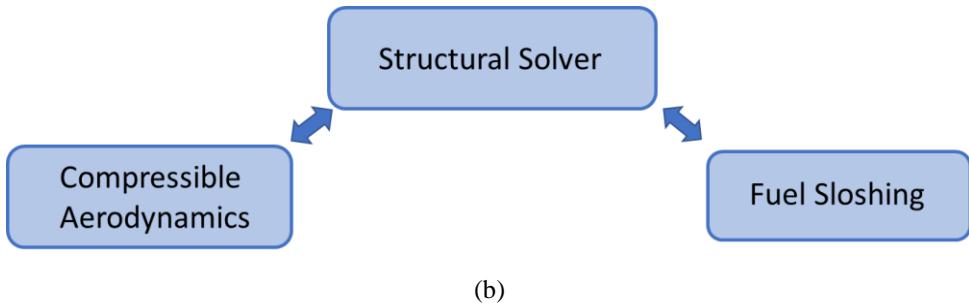
Researchers continued to work on high-order methods to capture nonlinearities of sloshing flows. Feng [34] developed the 3-D version of the famous Marker and Cell (MAC) method for a rectangular tank. This approach assigns volume fraction of phases in each cell and the interface is constructed by joining all the adjacent cells with fraction phase volume. Although this approach could capture nonlinearities of the flow, it consumed huge computational resources at the time and the presence of numerical instabilities was reported. Faltinsen [35] suggested a nonlinear analytical method that satisfied the nonlinear boundary condition at the surface to simulate sloshing. However, the coupling effect between the fluid and the structural part was missing in the study. The preliminary study of this coupling effect was studied by Haroun [36, 37] for cylindrical tanks. A high-fidelity sloshing solution methodology was still desirable which utilized Euler or Navier-Stokes based solver. In 1981, Hirt and Nicholas [38]

developed the Volume of Fluid (VOF) method based on the Navier-Stokes equation for two incompressible and immiscible fluids capable of tracking the interface. Like MAC, it used the phase fraction method to track the free surface location. Partom [39] presented the applications of VOF method for a cylindrical tank in 1987. Other methods such as Boundary Element Technique (BET) were developed for transient free-surface potential flow and more recently Smoothed Particle Hydrodynamics (SPH) is developed and utilized for sloshing computations by Hall [28]. However, the VOF method still remains popular and has been implemented in *OpenFOAM*'s [40] multiphase solvers *interFoam* and *interDyMFoam*.

With a detailed review of solution methodologies for aeroelastic and sloshing problem, the question still remains: Why lack of extensive research in sloshing effects on transonic aeroelasticity? This can be attributed to the difficulty in modeling a physical system containing both compressible aerodynamic flow and incompressible multiphase sloshing. Furthermore, the lack of multi-physics solvers in the open-source to tackle such problems also acts as a barrier to studying the combined effects of sloshing and aerodynamics on aeroelasticity. The problem can be centered around the motion of the wing, which is of primary concern. Hence the flow loads from compressible aerodynamics and sloshing must be cast as forcing terms in the equation of motion of the structure. An equivalent of the coupled system is shown in Fig. 1.3 (a) with a 2-DOF airfoil with an embedded partially filled fuel tank immersed in an external flow. This is represented in Fig. 1.3 (b) where the structural solver interacts with both flow solvers.



(a)

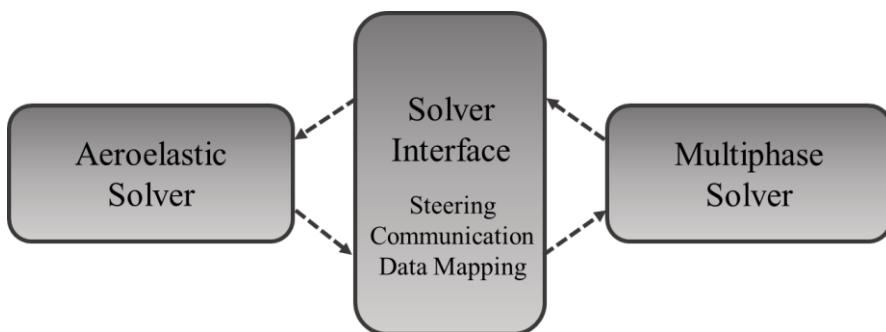


(b)

**Figure 1.3: (a) A 2-DOF airfoil with embedded tank immersed in an external flowfield, and (b) Interaction of structural solver with aerodynamic and sloshing solvers**

A solution to the lack of multiphysics solvers in open-source is to facilitate coupling of single physics solvers using a third-party interface. Many tools have been developed to facilitate coupling between existing solvers since multiphysics simulations have been gaining importance in recent times. One of the first tools is *MpCCI* [41] which was developed as a communication library between softwares. Its objective was to communicate surface data between codes but its capabilities have been expanded to data mapping functions for non-conforming grids, steering solvers and data exchange. This coupler had limited parallel-scalability and was limited by simple numerics schemes. These shortcomings were improved upon by *ADVENTURE\_Coupler* [42] which was developed as an open-source tool for particular use in an open-source general-purpose computational mechanics system ADVanced ENgineering analysis Tool for Ultra large Real world (*ADVENTURE*). The communication between the solvers was improved by using a server instead of a direct interaction between the codes. *EMPIRE* (Enhanced Multi Physics Interface Research Engine) is another code that implements the concept of a server for interaction between solvers and utilizes communication

using MPI which has parallel capabilities. It provides data-mapping between non-conforming grids at the coupling interface using interpolation functions. However, it still lacks effective solver communication with massively parallel simulations. The problem with communication and data exchange in heavily parallel codes especially those running on HPCs was substantially resolved by another open-source software, *OpenPALM* [43]. Some other couplers that have been developed with the same objective of communication and parallel data exchange for solver interaction include *OASIS* coupler [44] based on Model Coupling Toolkit (MCT) [45], *Physics Integrated Kernels* (PIKE) [46] and *preCICE* [47]. *preCICE* offers a multiphysics simulation environment emphasizing on realization of plug-and-play capability along with other functionalities such as parallel communication, solver steering, and data mapping. This acts as a solver interface between two single physics solvers and enables interaction via surface coupling. In its native state, it supports various open-source and commercial solvers such as *OpenFOAM* [40], *SU2* [18], *foam-extend* [48], *CalculiX* [49], *Code\_Aster* [50], *ANSYS Fluent* [51], *COMSOL* [52] etc. Hence *preCICE* is an ideal candidate for solving aeroelastic problems coupled with sloshing effects in an attached fuel tank. However, in its native form, *preCICE* does not facilitate such coupling and needs modification for its utilization for the aeroelasticity problem with sloshing fuel. The desired solver coupling between pure aeroelastic solver and multiphase sloshing solver is represented in Fig. 1.4.



**Figure 1.4: Solvers interaction via an interface by exchanging data**

With an appropriate solver interface, high-fidelity solutions of the coupled problem can be obtained. The component CFD solvers march the solution in time. This can be computationally very expensive. The nature of the problem requires the expensive task of deforming grids to accommodate structural motions. Furthermore, the sloshing solution typically requires a smaller time step for a converged solution. Therefore, advancement in one structural time step in the solution may require multiple smaller time steps for the multiphase solver. The computational cost dramatically increases when the problem demands flutter boundary computations. In that case, multiple runs at subsequently higher flutter speed indices at a fixed Mach number are required to compute the flutter boundary. Hence the computation of flutter boundary throughout the transonic flow range requires tedious time-domain computations.

This calls for the development of a low-cost surrogate model that can efficiently and accurately predict the dominant dynamics of coupled aeroelastic-sloshing multiphysics system which can take flow nonlinearities into account. Lucia et al. [53] and Dowell and Hall [54] provide a comprehensive overview of several reduced-order techniques such as harmonic balance, Volterra theory [55] and Proper orthogonal decomposition (POD) while demonstrating their application to aeroelastic test cases. There are approaches that use linear system identification concepts to obtain a reduced order model (ROM); however, such methods based on state-space approach cannot accurately capture nonlinearities of the sloshing flows, large amplitude vibrations and limit cycle oscillations, which require specialized methods for nonlinear system identification. Faller and Schreck [56] proposed a recurrent multi-layer-perceptron neural network (MLP-NN) to predict unsteady loads for aeroelastic applications and this study was subsequently followed up by Voitcu and Wong [57] and Mannarino and Mantegazza [58] leading to a systems model aeroelastic behavior of airfoils and wings based on non-recurrent MLP-NN. The dynamic loads of sloshing fluid in a fuel tank not only depends on the current state but also on the previous states and inputs since the fluid is always in a transition. In order

to include dynamic memory effects, the temporal history of the excitation signal is added to the input vector of the neural network. Neural networks have been shown to offer a powerful tool in modeling nonlinear systems over a compact set rather than a small neighborhood around the dynamically linearized steady-state ROM based approach for linear systems. It provides a powerful tool for learning complex input-output mappings and has simulated many studies for the identification of dynamic systems with unknown nonlinearities as outlined in a number of studies such as Narendra and Parthsarathy [59] and Elanayar and Shin [60].

The Radial basis function neural networks (RBF-NNs) belongs to the domain of artificial neural networks, which is able to approximate any nonlinear function to an arbitrary degree of accuracy with a finite number of neurons. Originally published by Lowe and Broomhead [61], they are derived from the theory of functional approximation and conventional approximation theory. RBF-NNs have proved to a good alternative to multi-layer perceptron (MLP) networks because of their high learning rate [62] and robustness to noise. The predictive capability of RBF-NN can be utilized for developing a surrogate model for predicting sloshing loads on the container due to its motion. Fuel sloshing is a transient phenomenon where previous structural state inputs, sloshing loads as well as the current structural state inputs are required for the prediction of the loads for the next time step (the future state). One widely used system identification technique is the Autoregressive technique with Exogenous inputs (ARX) as outlined in Billings [63] which assumes that the known relationship between a finite series of former inputs and previous outputs is sufficient to predict system response to subsequent inputs. Winter [64] has used a similar technique for the prediction of aerodynamic loads using RBF-NNs.

Although the machine learning models such as recurrent neural networks [65], RBF-NNs and other MLP neural networks have been successfully utilized in system identification and surrogate modeling, they still remain black-box models whose capability is restricted by the

quality and richness of training data. Hence the performance of these models outside of the learning parameters domain cannot be relied upon. Recent trends indicate the inclination of the scientific community towards models that utilize laws governing the dynamics of physical systems in determining the parameters of the surrogate model. One such model is Physics-Informed Neural Network introduced by Raissi [66] that incorporates the governing equations of the underlying physical system in the learning process of the neural network. Incremental work based on PINN has popular for solutions of systems governed by partial differential equations recently.

## 1.2 Motivation for this Research

The motivation for the present research is threefold: absence of a general and open-source numerical tool to study the effects of sloshing on aeroelasticity of transonic flights, lack of application of surrogate and reduced order modeling techniques to model sloshing loads using, and need for a low-cost computational platform that can accurately emulate coupled motion of a wing embedded with a partially filled fuel tank for quick estimation of aeroelastic characteristics of the coupled aero-structural-fuel tank system

1. The lack of multi-physics solvers in the open-source to study sloshing effects on transonic flights acts as a barrier to this field of research. Numerical solvers such as *SU2* and *OpenFOAM* are capable of solving problems involving single physics. With the development of an appropriate coupling tool, a computational framework can be materialized to obtain high-fidelity CFD solutions for the coupled aero-structural-fuel tank problem.
2. There has been significant progress in the development of surrogate and reduced order models for aerodynamics in literature ranging from physics-based intrusive methods to black-box methods. However, these concepts have not been extended and tested for solution of sloshing problems. Researchers resort to low fidelity methods based on linearized theories for

sloshing problems when computationally inexpensive models are desired. This calls for the development of a low-cost surrogate model for sloshing which can emulate its nonlinear behavior as well.

3. Development of a computational framework based on data-driven models for both compressible aerodynamic flows and sloshing flows that can predict the aeroelastic motion of the coupled aero-structural-fuel tank system at a very small computational cost. The surrogate model must efficiently and accurately predict the unsteady aerodynamic and sloshing loads such that they can be utilized for predicting the onset of flutter, limit cycle oscillations and flutter mitigation directed aerodynamic wing design for wing with embedded fuel tank.

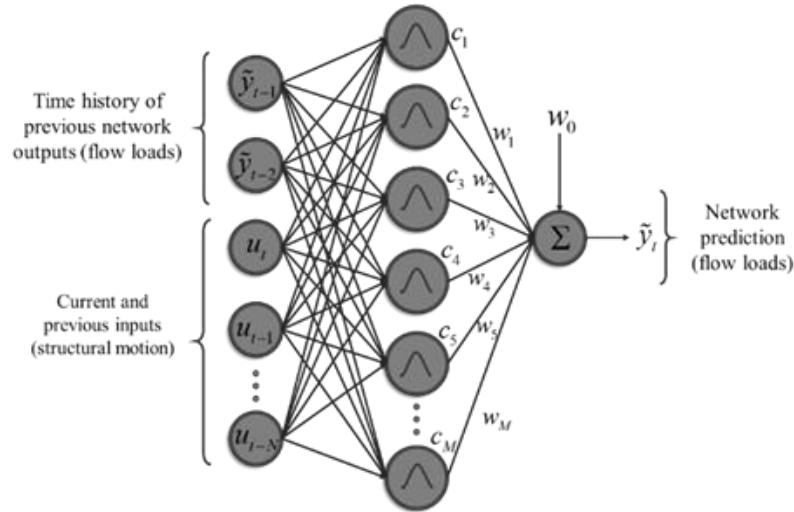
### **1.3 Contributions of this Thesis**

This thesis is focused on the development of computational frameworks for the assessment of the effects of fuel sloshing on transonic flights. Some of the major contributions of this work are listed as follows:

1. A methodology is developed to integrate open-source CFD solvers *SU2* and *OpenFOAM* for solution of coupled aero-structural-fuel tank system. The *preCICE* solver interface has been modified in the present work to accommodate the coupling of aeroelastic solver and multiphase sloshing solver by enabling the exchange of data between the solvers during runtime. The component CFD solvers march the solution in time. High-fidelity numerical solutions obtained from this method are considered ground truth for comparison and evaluation of other models and frameworks developed in this work. The flutter boundary of NACA64A010 is computed using this framework.

2. Coupled CFD solutions for aero-structural-fuel tank system is computationally expensive. Therefore, usage of cheaper models and the development of computationally inexpensive surrogate models for such aeroelastic analysis becomes a natural choice. A surrogate model

based on RBF-NN is developed with the aim to efficiently simulate the dominant static and dynamic characteristics of the sloshing tank. The dynamic memory effect is accounted for by supplying previous outputs of RBF-NN in form of loads, as well as, previous inputs in the form of structural displacement to the tank as shown in Fig.1.5. A limited set of CFD-based data is exploited for calibrating the surrogate. Once trained, the RBF-NN is subsequently fed with new inputs in the form of structural motion, as well as its previous outputs to predict sloshing loads at the present time step. This is coupled with the aeroelastic solver for time marching solutions. The flutter boundary of NACA64A010 embedded with a partially filled rectangular fuel tank is compared for the prediction accuracy of RBF-NN in this thesis.

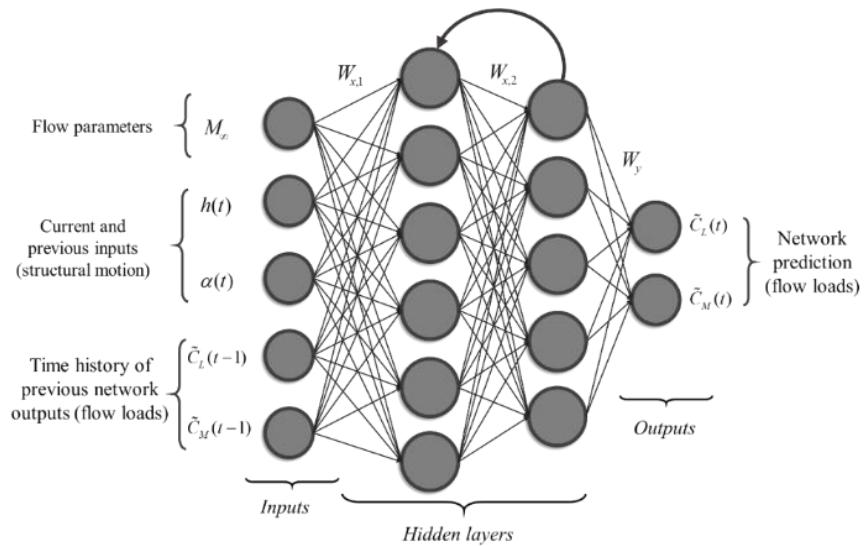


**Figure 1.5: A Radial Basis Function Neural Network for prediction of sloshing loads**

This work also utilizes the EMS model to demonstrate the effect of fuel sloshing on the flutter boundary of NACA64A010 airfoil in transonic regime. The EMS parameters are derived by comparing its force and moment equation with those obtained from potential flow formulations for sloshing in rectangular tanks. The EMS model simulates inertial and convective components of sloshing liquids by combinations of linear sloshing modes. The flutter boundary of NACA64A010 is computed by simulating sloshing loads with the EMS model. The

limitations of usage of the EMS model are demonstrated by studying sloshing load response to parametrically varying excitation frequency.

3. Recurrent neural network based surrogates are developed to efficiently and accurately predict unsteady aerodynamic loads of a free plunging and pitching airfoil, as well as, fuel sloshing loads in a partially filled fuel tank subjected to the same displacements of the airfoil. These models, when fed with temporal sequences of structural displacement, and load coefficients, predict the load coefficients of the subsequent time step. The structure of this model is shown in Fig. 1.6. These surrogates are coupled to construct a computational framework that can predict the aeroelastic motion of airfoil with an embedded fuel tank. The surrogate for aerodynamics is further improved to incorporate parametric flow variations of the free-stream Mach number. The performance of these frameworks is demonstrated by predicting the aeroelastic motion of NACA0012 and NASA BSCW, respectively, with and without the influence of fuel sloshing. The computational savings for these frameworks are recorded to demonstrate its prediction efficiency.



**Figure 1.6: Recurrent neural network for prediction of aerodynamic load coefficients with flow parameterization**

4. A Physics-Informed Neural Network is developed to emulate the aeroelastic structural equation of motion for a 2-DOF aeroelastic system. The network parameters of PINN are learned by minimizing a modified loss function containing a residue term computed from the solution of aeroelastic equation of motion. The prediction capability and accuracy of the model is demonstrated for aeroelastic motion of NACA0012 airfoil without the effects of sloshing.

## Chapter 2

### Computational Models

The present research entails flow solutions for compressible high-speed aerodynamics and incompressible multiphase sloshing in an enclosed container. The external aerodynamic flow is computed by Euler flow equations cast into Arbitrary Lagrangian-Eulerian formulation to incorporate dynamics meshed for unsteady flows. The multiphase flow is computed by Volume of Fluid method which calculates the phase fraction of fluids in each computational cell. The present study also includes a linear model of sloshing using potential flow equations for approximation of an Equivalent Mechanical System which can predict sloshing loads. The fluid-structure interaction problem models the airfoils and wing structures by 2-DOF aeroelastic equation of motion. The forcing terms in the equation of motion correspond to aerodynamic and sloshing loads in the form of forces and moments. The numerical schemes utilized for this research for flow solutions, structural interactions and couplings are elaborated in this chapter. When appropriate, conflicting notations and nomenclatures are reconciled.

#### 2.1 Aerodynamic Model

##### Inviscid External Flow (Euler Equations)

Euler equation is cast in Arbitrary Lagrangian-Eulerian (ALE) form in integral formulation as detailed by Alonso and Jameson [17],

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \vec{U}(\vec{u}_m, t) d\Omega + \int_{\partial\Omega(t)} \vec{F} \cdot \vec{n} dS = 0 \quad (2.1)$$

where control volume represented by  $\Omega$  and its boundary represented by  $\partial\Omega$  are time dependent.

$$\vec{U} = \begin{bmatrix} \rho \\ \rho\vec{V} \\ \rho E \end{bmatrix} \quad (2.2)$$

$$\vec{F} = \begin{bmatrix} \rho(\vec{V} - \dot{\vec{u}}_m) \\ \rho\vec{V} \otimes (\vec{V} - \dot{\vec{u}}_m) + \vec{I} p \\ \rho E(\vec{V} - \dot{\vec{u}}_m) + p\vec{V} \end{bmatrix} \quad (2.3)$$

where  $\rho$ ,  $\vec{V}$ ,  $E$  and  $p$  are the density, velocity vector, the total energy and the pressure of the ambient fluid. Arbitrary Lagrangian-Eulerian formulation is a mixed reference frame containing mesh velocity  $\dot{\vec{u}}_m$ , in the flux term.

### Discretization of convective flux term using Jameson-Schmidt-Turkel (JST) scheme

The 2-D computational domain is divided into cells whose cell centers are denoted by the subscripts  $i,j$  as shown in Fig. 2.1. In the finite volume setting, the values of the dependent variables are considered to have uniform value within the cell and the information is stored at the center of the cell, as detailed by Jameson [67]. Consider quadrilateral cell, applying integral of conservation laws, i.e.

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \vec{U}(\vec{u}_m, t) d\Omega + \int_{\partial\Omega(t)} \vec{F} \cdot \vec{n} dS = 0 \quad (2.4)$$

leads to a system of ordinary differential equations

$$\frac{d}{dt} (s_{i,j} U_{i,j}) + F_{i,j} = 0 \quad (2.5)$$

where  $s_{i,j}$  is the cell area and  $F_{i,j}$  is the net flux of the cell given by,

$$\sum_{k=1}^4 (\Delta y_k f_k - \Delta x_k g_k) \quad (2.6)$$

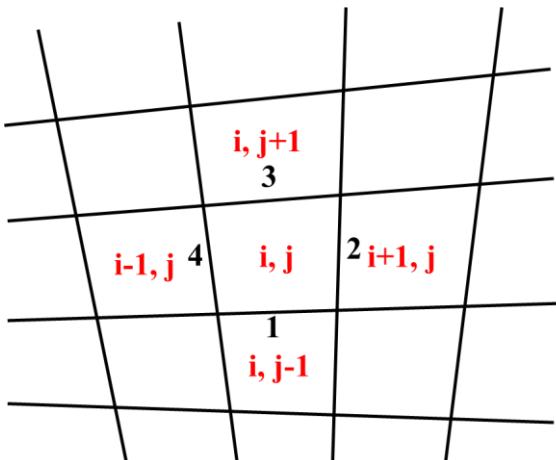
where  $\Delta x_k$  and  $\Delta y_k$  are the increments of  $x$  and  $y$  along the edge with appropriate signs,  $f_k$  and  $g_k$  are the values of the flux vectors  $f$  and  $g$  at the  $k^{th}$  edge and is summed over all four edges of the cell. Each element of the flux vectors is calculated as the average of the values on either side of each edge. For example, the flux calculated on edge 2 of the computational cell with center  $i,j$  as shown in Fig. 2.1 is given by,

$$f_2 = \frac{1}{2}(f_{i,j} - f_{i+1,j}) \quad (2.7)$$

The flux balance for x-momentum is given by,

$$\frac{d}{dt}\{(\rho u)_{i,j} s_{i,j}\} + \sum_{k=1}^4 \{F_k (\rho u)_k + \Delta y_k p_k\} = 0 \quad (2.8)$$

The numerical scheme has second-order spatial accuracy.



**Figure 2.1: Spatial discretization of computational domain**

To suppress the tendency of odd and even point decoupling and to limit undesirable overshoots near discontinuities such as shock waves, Eqn. 2.8 has to be supplemented with artificial viscosity term  $D_{i,j}$  i.e.

$$\frac{d}{dt}(s_{i,j} U_{i,j}) + F_{i,j} - D_{i,j} = 0 \quad (2.9)$$

Artificial viscosity terms are a blend of second and fourth differences in flow variables. For example, the artificial viscosity terms for density equation is,

$$D_{i,j} = d_{i+1/2,j} - d_{i-1/2,j} + d_{i,j+1/2} - d_{i,j-1/2} \quad (2.10)$$

where the dissipative flux  $d_{i+1/2,j}$  is defined by,

$$d_{i+1/2,j} = \frac{h_{i+1/2,j}}{\Delta t} \left\{ \varepsilon_{i+1/2,j}^{(2)} (\rho_{i+1,j} - \rho_{i,j}) - \varepsilon_{i+1/2,j}^{(4)} (\rho_{i+2,j} - 3\rho_{i+1,j} + 3\rho_{i,j} - \rho_{i-1,j}) \right\} \quad (2.11)$$

where  $\varepsilon^{(2)}$  and  $\varepsilon^{(4)}$  are adaptive parameters defined as,

$$\varepsilon_{i+1/2,j}^{(2)} = k^{(2)} \max(v_{i+1,j}, v_{i,j}) \quad (2.12)$$

and

$$\varepsilon_{i+1/2,j}^{(4)} = \max(0, k^{(4)} - \varepsilon_{i+1/2,j}^{(2)}) \quad (2.13)$$

where

$$v_{i,j} = \frac{|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}|}{|p_{i+1,j}| + |2p_{i,j}| + |p_{i-1,j}|} \quad (2.14)$$

The typical values of  $k^{(2)}$  and  $k^{(4)}$  are,

$$k^{(2)} = \frac{1}{4} \quad \text{and} \quad k^{(4)} = \frac{1}{256} \quad (2.15)$$

The dissipative terms for the remaining equations are obtained by substituting  $\rho u$ ,  $\rho v$  and either  $\rho E$  or  $\rho H$  for  $\rho$  in these formulas.

### Time Marching Using Dual Time Stepping Method

Dual-time method [68] consists of using an outer time-step for the formulation of the fully-implicit equations, i.e. without linearization, and an inner time-step loop for the resolution of a

linear system at each outer time-step. Local time-step, multigrid and implicit algorithms can be used to achieve faster convergence of the linear system. Governing equation can be cast into the following form,

$$\frac{d}{dt}(\vec{U})^{n+1} + R(\vec{U}^{n+1}) = 0 \quad (2.16)$$

where  $(n+1)$  denotes the time level  $(n+1)\Delta t$  and temporal operator is approximated by a backward difference formula of second-order accuracy. Time derivative can be discretized as,

$$\frac{dU_i}{dt} = \frac{3U_i^{n+1} - 4U_i^n + U_i^{n-1}}{2\Delta t} \quad (2.17)$$

The set of coupled ordinary differential equations can be solved at each time step by defining a modified residual  $R^*(\vec{U})$  as,

$$R^*(\vec{U}) = \frac{d}{dt}(\vec{U})^{n+1} + R(\vec{U}^{n+1}) \quad (2.18)$$

The governing equation is marched in fictitious time  $t^*$ . Convergence will give time-accurate solution at  $(n+1)^{th}$  time step.

$$\frac{d}{dt}(\vec{U})^{n+1} + R^*(\vec{U}^{n+1}) = 0 \quad (2.19)$$

The above governing equation is solved by LU-SGS implicit algorithm.

## 2.2 Fluid Sloshing

While a number of models have been used by various researchers as reported in literature, in this work two approaches of different fidelities are used for modeling the sloshing effects of the internal fluid in a rectangular fuel tank to assess its effect on aeroelastic characteristics and flutter prediction. The first approach is based on the full-order CFD approach which utilizes

high fidelity flow computations based on the open-source CFD software *OpenFOAM*. The second approach represents sloshing with an equivalent mechanical system (EMS) which is derived from potential flow based formulations for sloshing in rectangular tanks.

### A. Volume of Fluid Method:

Fuel sloshing in a partially filled rectangular fuel tank is modeled using a multi-phase fluid model based on the incompressible Navier-Stokes Equations consisting of the continuity and momentum equations as follows:

$$\int_{\Omega(t)} (\nabla \cdot \vec{U}) d\Omega = 0 \quad (2.20)$$

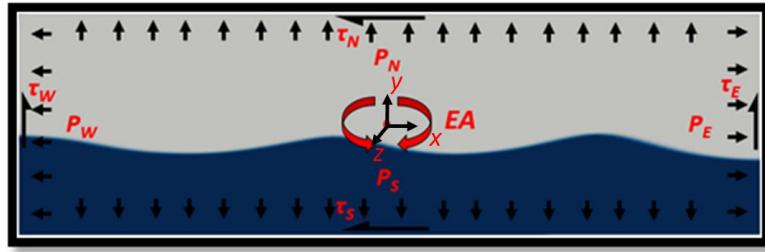
$$\frac{\partial}{\partial t} \int_{\Omega(t)} \vec{U} d\Omega + \int_{\partial\Omega(t)} (\vec{U}(\vec{U} \cdot \vec{n})) dS = -\frac{1}{\rho} \int_{\partial\Omega(t)} (\nabla p - \mu(\nabla \cdot \vec{U}) \vec{U}) dS + \int_{\Omega(t)} (\vec{f}_B + \vec{f}_v) d\Omega$$

$$(2.21)$$

where  $\vec{U}$  denotes the velocity of fluid relative to the tank,  $\Omega(t)$  and  $\partial\Omega(t)$  denote are time-dependent control volume and its boundary,  $p$  the pressure,  $\rho$  and  $\mu$  the density and viscosity, respectively,  $\vec{f}_B$  represents the external body forces per unit mass for the fluid due to gravity and  $\vec{f}_v$  represents body force per unit mass on the fluid influenced by tank motion. Fuel sloshing in the tank consists of two phases namely fuel and the vapor regions in the tank. The volume of fluid (VOF) method of Hirt and Nicholas [38] is used for tracking the volume of these phases in a fixed Eulerian mesh for the internal flow in the tank. In this method, a single set of momentum equations is shared by the fluid phases and the volume fraction of each fluid is tracked throughout the domain. A scalar function  $f$  is used to characterize the free surface deformation, the value of which is set based upon the fluid volume fraction of a cell. The volume fractions are updated by the equation written as

$$\frac{D}{Dt} \int_{\Omega(t)} f d\Omega = \int_{\Omega(t)} \frac{\partial f}{\partial t} d\Omega + \int_{\partial\Omega(t)} (f(\vec{U} \cdot \vec{n})) dS = 0 \quad (2.22)$$

After computation of the volume fractions in each cell, the equivalent density  $\rho_{cell}$  and viscosity  $\mu_{cell}$  are estimated as  $\rho_{cell} = \rho_{gas} + f(\rho_{fuel} - \rho_{gas})$  and  $\mu_{cell} = \mu_{gas} + f(\mu_{fuel} - \mu_{gas})$  where  $\rho_{gas}$  and  $\mu_{gas}$  are the density and viscosity of the vapor respectively, while  $\rho_{fuel}$  and  $\mu_{fuel}$  are the density and viscosity of the fuel in the tank. The fuel sloshing problem is solved in time-domain and at each time-step, the sloshing forces and moments acting on the tank wall are obtained by integrating the pressure fields and shear forces along the tank walls. The sloshing of the fuel in a partially filled tank is computed using the *interDyMFoam* solver which implements the VOF method in *OpenFOAM* [40]. The pressure and shear forces from the computation are integrated over the wetted area of the fuel tank wall as shown in Fig. 2.2 and are used to estimate the sloshing loads as follows:



**Figure 2.2: Partially filled fuel tank with normal and shear forces acting on the tank walls due to sloshing fluid**

$$F_{x,sl} = \int_{-h/2}^{h/2} p_E(y)w dy - \int_{-h/2}^{h/2} p_W(y)w dy + \int_{-l/2}^{l/2} \tau_S(x)w dx + \int_{-l/2}^{l/2} \tau_N(x)w dx \quad (2.23)$$

$$F_{y,sl} = \int_{-l/2}^{l/2} p_N(x)w dx - \int_{-l/2}^{l/2} p_S(x)w dx + \int_{-h/2}^{h/2} \tau_W(y)w dy + \int_{-h/2}^{h/2} \tau_E(y)w dy \quad (2.24)$$

The corresponding moments about the geometric center of the tank are estimated as follows:

$$\begin{aligned}
M_z = & \int_{-h/2}^{h/2} p_E(y)w(y - y_{EA})dy + \int_{-l/2}^{l/2} p_N(x)w(x - x_{EA})dx - \int_{-l/2}^{l/2} p_w(y)w(y - y_{EA})dy \\
& - \int_{-l/2}^{l/2} p_S(x)w(x - x_{EA})dx + \int_{-h/2}^{h/2} \tau_W(y)w(x - x_{EA})dy + \int_{-l/2}^{l/2} \tau_N(x)w(y - y_{EA})dx \\
& - \int_{-h/2}^{h/2} \tau_E(y)w(x - x_{EA})dy - \int_{-l/2}^{l/2} \tau_S(x)w(y - y_{EA})dx
\end{aligned} \quad (2.25)$$

The sloshing forces are the vector sum of the forces on the wall, and hence there is no need to take special care of inertial forces. The local axis,  $xyz$ , is attached to the geometric center of the fuel tank and the sloshing loads namely,  $F_x$ ,  $F_y$  and  $M_z$ , are computed relative to this frame of reference. The sloshing loads are then projected in the inertial frame i.e.  $XYZ$  to form the coupled aero-structural model of Fig 1 using the transformation,

$$F_{X,sl} = F_{x,sl} \cos \alpha + F_{y,sl} \sin \alpha \quad (2.26)$$

$$F_{Y,sl} = F_{y,sl} \cos \alpha - F_{x,sl} \sin \alpha \quad (2.27)$$

$$M_{OZ,sl} = M_{oz,sl} \quad (2.28)$$

## B. Equivalent Mechanical System (EMS):

The fundamental problems of sloshing require approximation of hydrodynamic pressure distribution, integrated pressures to compute forces and moments generated, location of free surface and response frequencies of the bulk liquid. Broadly, there are two components of hydrodynamic pressure; an inertial component as a direct consequence of container acceleration, and a convective component representing free-surface liquid motion. This demarcation spurs the understanding of sloshing dynamics and also plays an important role in modeling sloshing loads using a mass-spring system representations.

The fluid is assumed to have inviscid behavior and the motion is assumed to be without vorticity, hence a velocity potential can be derived, represented by  $\Phi$ . For a 2-D problem, the velocity components,  $u$  and  $v$ , computed from the potential function as per Eqn. 2.29. The sloshing fluid is considered incompressible and hence the Laplace equation given in Eqn. 2.30 and Eqn. 2.31. The equations are also presented in vector form to maintain generality with velocity represented by  $\vec{u}$ . A partially filled fuel tank is shown in Fig. 2.3 with the coordinate axes  $x, y$  that is fixed to and moves with the tank, whereas the inertial coordinate system  $X, Y$  is stationary. This derivation is based on Ibrahim's [33] formulations for EMS modeling.

$$u = \frac{\partial \Phi}{\partial x} \text{ and } v = \frac{\partial \Phi}{\partial y} \text{ or } \vec{u} = \nabla \Phi \quad (2.29)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (2.30)$$

hence,

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0 \text{ or } \nabla^2 \Phi = 0 \quad (2.31)$$

The Euler equation of hydrodynamics is given as Eqn. 2.32. Applying the irrotationality condition in Eqn. 2.33 and considering small velocity such that higher terms vanish (linearizing the equation), as per Eqn. 2.34, we get the general Bernoulli's equation at any point  $P$  as given in Eqn. 2.35. The pressure and fluid density are represented by  $p$  and  $\rho_L$ , respectively.

$$\frac{\partial \vec{u}}{\partial t} + (\nabla \times \vec{u}) \times \vec{u} + \nabla \left( \frac{p}{\rho_L} + \frac{1}{2} |u|^2 \right) = \vec{g} \quad (2.32)$$

$$\nabla \times \vec{u} = 0 \text{ or } \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = 0 \quad (2.33)$$

$$|u|^2 \simeq 0 \quad (2.34)$$

$$\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho_L} \nabla p = \vec{g} \text{ or } \nabla \left( \frac{\partial \Phi}{\partial t} + \frac{p}{\rho_L} - g(x - x_p) \right) = 0 \quad (2.35)$$

The solution of the equation of motion can be obtained by applying the boundary conditions at the tank walls and the free surface. The surrounding gas density is considered negligible as compared to the liquid density, and the pressure at the free surface is considered as a static value,  $p_0$ . The tank with height  $h$  is half-filled for the present case and at  $t=0$  and the free surface is at  $y = h/2$ . The equation of motion at the free surface is:

$$\frac{\partial \Phi(x, y, t)}{\partial t} + g\delta(x, t) = \frac{-p_0}{\rho} \text{ at } y=h/2 \quad (2.36)$$

where  $\delta(x, t)$  is free surface displacement above the initial level. Eqn. 2.36 is the dynamics boundary condition. The kinetic boundary condition is obtained by relating the free surface velocity with velocity potential as given in Eqn. 2.37.

$$\frac{\partial \delta}{\partial t} = v = \frac{\partial \Phi}{\partial y} \text{ at } y=h/2 \quad (2.37)$$

The boundary conditions can be combined to eliminate the surface perturbation  $\delta$  (or velocity potential,  $\Phi$ ) by computing the temporal derivative of  $\delta$  in Eqn. 2.36. The resulting equation is given as follows:

$$\frac{\partial^2 \Phi}{\partial t^2} + g \frac{\partial \Phi}{\partial y} = 0 \text{ at } y=h/2 \quad (2.38)$$

The sloshing fluid is considered as inviscid and hence the boundary condition at the tank walls does not contain any viscous stress terms. The only condition at the boundary should be no flow condition, i.e. the normal velocity at the walls should be zero as per Eqn. 2.39. Now, this

condition is imposed with the excitation of the tank. For forced pitching excitation with an amplitude  $\psi_0$  and forcing frequency  $\Omega$  about  $z$ -axis given by Eqn. 2.40, the boundary condition is given in Eqn. 2.41.

$$\left. \frac{\partial \Phi}{\partial x} \right|_{x=\pm l/2} = 0 \quad \text{and} \quad \left. \frac{\partial \Phi}{\partial y} \right|_{y=0} = 0 \quad (2.39)$$

$$\psi(t) = \psi_0 \sin(\Omega t) \quad (2.40)$$

$$n \cdot \nabla \Phi = i \psi_0 \Omega e^{i \Omega t} \quad (2.41)$$

The natural frequencies of the free surface can be calculated using Eqn. 2.31 imposed with boundary conditions given in Eqn. 2.39. These natural frequencies correspond to *sloshing modes*, which are analogous to structural modes. The natural frequency formulation is given in Eqn. 2.42 as follows:

$$\omega_n^2 = (2n-1) \frac{\pi g}{l} \tanh\left((2n-1) \frac{\pi h_f}{l}\right) \quad (2.42)$$

where  $h_f$  is the height of the free surface at  $t=0$ . For the present study, the fill level of the fuel tank is 50% i.e.  $h_f = h/2$ . As apparent from the above equation, the natural frequencies are dependent on the fill level, and the tank dimensions. Hence, these are not influenced by the type of motion the tank configuration is subjected to. Solving the equation of motion with the above boundary conditions, the horizontal forces along the  $x$ -axis and moment about the  $z$ -axis are derived as follows:

$$F_x = (\rho g h_f l b) \frac{\Omega^2 h_f}{g} \Psi_0 \sin(\Omega t) \left\{ \frac{l^2}{12 h_f^2} + \sum_{n=1}^N \frac{8 \tanh\left((2n-1) \frac{\pi h_f}{l}\right)}{\pi^3 (2n-1)^3} \left( \frac{\Omega^2}{\omega_n^2 - \Omega^2} \right) \times \left[ \frac{1}{2} - \frac{2l \tanh\left((2n-1) \frac{\pi h_f}{2l}\right)}{\pi (2n-1) h_f} + \frac{g}{h \omega_n^2} \right] \right\}$$

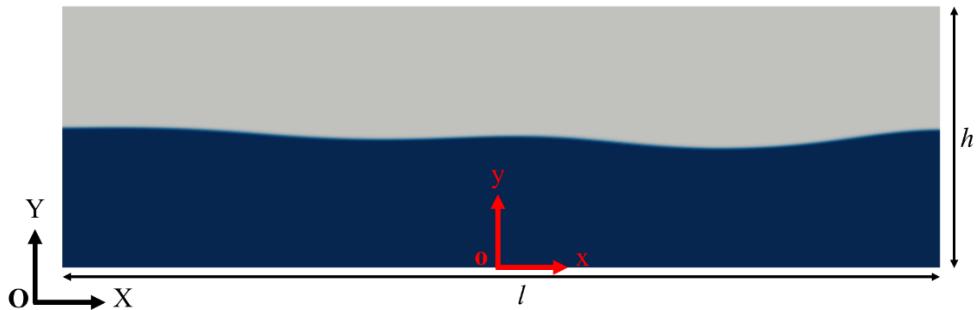
(2.43)

$$\begin{aligned}
 M_z = & \left( \rho g h_f^2 l b \right) \frac{\Omega^2 h_f}{g} \Psi_0 \sin(\Omega t) \left\{ \frac{I_{fz}}{m_z h_f^2} + \frac{gl^2}{12h_f^3 \Omega^2} + \sum_{n=1}^N \frac{8 \tanh\left((2n-1)\frac{\pi h_f}{l}\right)}{\pi^3 (2n-1)^3 h_f} \right. \\
 & \times \left[ \frac{1}{2} - \frac{2l \tanh\left((2n-1)\frac{\pi h_f}{2l}\right)}{\pi(2n-1)h_f} + \frac{g}{2h_f \omega_n^2} \right] \frac{g}{h_f \omega_n^2} \Bigg] \\
 & + \sum_{n=1}^N \frac{8l \tanh\left((2n-1)\frac{\pi h_f}{a}\right)}{\pi^3 (2n-1)^3 h_f} \left[ \frac{1}{2} - \frac{2l \tanh\left((2n-1)\frac{\pi h_f}{2l}\right)}{\pi(2n-1)h_f} + \frac{g}{h_f \omega_n^2} \right]^2 \left( \frac{\Omega^2}{\omega_n^2 - \Omega^2} \right) \Big\} \quad (2.44)
 \end{aligned}$$

where  $I_{fz}$  is the effective mass moment of inertia of the fluid about  $z$ -axis given by Eqn. 2.45,

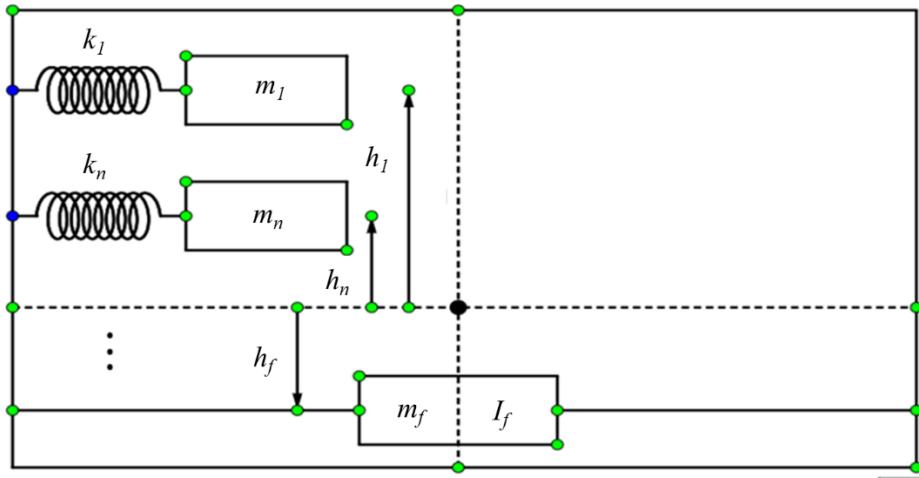
$$I_{fz} = I_{sz} \left\{ 1 - \frac{4}{1 + \left( \frac{h_f}{l} \right)^2} + \frac{768}{\pi^3 h_f (l^2 + h_f^2)/l^3} \sum_{n=1}^N \frac{\tanh\left((2n-1)\frac{\pi h_f}{2l}\right)}{(2n-1)^5} \right\} \quad (2.45)$$

where  $I_{sz}$  is the mass moment of inertia of the solidified liquid about the  $z$ -axis at  $t=0$ .



**Figure 2.3: Half-filled fuel tank with tank dimensions and reference axes used for derivation of equivalent mechanical system parameters**

Now, with the derived force and moment formulation, the tank is represented by a configuration of mass-spring systems shown in Fig. 2.4, each set representative of a fluid mode. A frozen mass,  $m_f$ , moves in unison with the tank and represents the inertial component of sloshing, and the series of movable masses,  $m_n$ , represents the convective part of sloshing. Each mass is attached to the tank by a spring  $k_n$ , representing sloshing modes. From here, these masses will be referred to as *modal masses*.



**Figure 2.4: Equivalent mechanical system (EMS) consisting of mass-spring system to represent sloshing loads of a partially filled fuel tank**

Now, the modal masses, their location, and the corresponding spring constants can be computed using a series of constraints as follows:

The sum of rigid and modal masses must be equal to the total fluid mass:

$$m_T = m_f + \sum_{n=1}^N m_n \quad (2.46)$$

Mass moment of inertia about  $z$ -axis that passes through the center of mass of solidified liquid:

$$I_T = I_f + m_f h_f^2 + \sum_{n=1}^N m_n h_n^2 \quad (2.47)$$

Preservation of center of mass:

$$m_f h_f - \sum_{n=1}^N m_n h_n = 0 \quad (2.48)$$

Calculation of spring constants using modal mass and natural frequency relation:

$$\omega_n^2 = K_n / m_n \quad (2.49)$$

Now, this EMS configuration is subjected to forced structural excitation about the *z-axis* as given by Eqn. 2.50. It should be noted that this structural excitation is the same as that used for potential flow formulation in Eqn. 2.40. This is done deliberately such that the obtained forces and moments can be directly comparable with each other.

$$\psi(t) = \psi_0 \sin(\Omega t) \quad (2.50)$$

The response force and moment obtained for the EMS configuration is given by Eqn. 2.51 and Eqn. 2.52, respectively.

$$F_x = \Psi_0 \Omega^2 \sin(\Omega t) \left\{ m_f h_f + \sum_{n=1}^{\infty} m_n (h_n \Omega^2 + g) \frac{\omega_n^2}{(\omega_n^2 - \Omega^2)} \right\} \quad (2.51)$$

$$M_z = \Psi_0 \Omega^2 \sin(\Omega t) \left\{ I_f + m_f h_f^2 + \sum_{n=1}^{\infty} m_n (h_n \Omega^2 + g)^2 \frac{\omega_n^2}{(\omega_n^2 - \Omega^2)} \right\} \quad (2.52)$$

The computed force and moment are compared with the force and moment response of the potential flow based model given in Eqn. 2.43 and Eqn. 2.44 to compute the modal masses, spring constants, and modal mass locations. The computed parameters are given as follows:

$$\frac{m_n}{m_f} = \frac{8}{\pi^3} \frac{\tanh((2n-1)\pi h_f / l)}{(2n-1)^3 h / l} \quad (2.53)$$

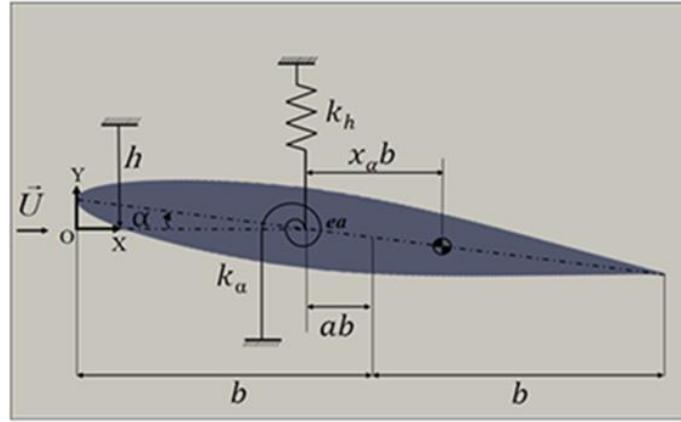
$$K_n = \frac{gm_f}{h_f} \frac{8 \tanh((2n-1)\pi h_f / l)}{(2n-1)^2} \quad (2.54)$$

$$\frac{h_n}{h} = \frac{1}{2} - \frac{\tanh((2n-1)\pi h_f / 2l)}{(2n-1)\pi h / 2l} \quad (2.55)$$

It should be noted that the modal masses, spring constants, and their respective locations are functions of tank configuration only. It is worthwhile to point out that the force and moment formulations given in Eqn. 2.43, Eqn. 2.44, Eqn. 2.51 and Eqn. 2.52 are linear in amplitude of the forcing motion,  $\psi_0$ . This is important because although the formulations and EMS parameters are derived for forced motion of constant amplitude, the force and moment can be easily computed for motions with varying amplitudes, such as aeroelastic motion of the airfoil as long as the frequency of motion is constant. Therefore, different possible aeroelastic motions of the airfoil such as damped response, neutral response, divergent response (flutter) and limit cycle oscillations can be addressed by the same formulation. The same configuration is used for plunging motion. The motion is considered in the linear domain and nonlinear phenomenon like wave breaking are not accounted for, the vertical loads comprise of the total mass of the fluid. The EMS model performance evaluation and coupled results with airfoil are detailed in the Results and Discussions Chapter of this thesis.

### 2.3 Aero-Structural Model

The motion of a 2-DOF airfoil free to plunge and pitch is governed by the aeroelastic equations of motion. The system is represented in Fig. 2.5 where the airfoil motion is modeled by equivalent springs in the plunging and pitching hinged at the elastic axis. In the case of pure aeroelasticity with no sloshing effects, the aeroelastic equation comprises of force and moment balance equations given in Eqn. 2.56 and Eqn. 2.57.



**Figure 2.5: A 2-DOF airfoil immersed in a flowfield free to plunge and pitch**

$$m\ddot{h} + S_\alpha \ddot{\alpha} + c_h \dot{h} + k_h h = -L \quad (2.56)$$

$$S_\alpha \ddot{h} + I_\alpha \ddot{\alpha} + c_\alpha \dot{h} + k_\alpha \alpha = M \quad (2.57)$$

where  $m$  is the structural mass per unit length of the airfoil,  $S_\alpha$  is the static imbalance,  $c_h$  is the damping coefficient for the plunging motion,  $k_h$  is the spring constant for plunging motion,  $h$  is the plunging displacement of the elastic axis of the airfoil measured against its initial position,  $I_\alpha$  is the moment of inertia of the airfoil about its elastic axis,  $c_\alpha$  is the damping constant for pitching motion,  $k_\alpha$  is the spring constant for pitching motion,  $\alpha$  is the pitching angle relative to zero angle of attack position of the airfoil and  $M$  is the moment about the elastic axis of the airfoil due to pressure distribution of external flow around the airfoil.

Non dimensionalizing this equation,

$$\ddot{\bar{h}} + x_\alpha \ddot{\alpha} + \frac{c_h}{m\omega_\alpha} \dot{\bar{h}} + \frac{k_h}{m\omega_\alpha^2} \bar{h} = \frac{1}{2} \frac{\rho U_\infty^2 C_L S_{ref}}{mb\omega_\alpha^2} \quad (2.58)$$

$$x_\alpha \ddot{\bar{h}} + r_\alpha^2 \ddot{\alpha} + \frac{c_\alpha}{mb^2\omega_\alpha} \dot{\bar{h}} + \frac{k_\alpha}{mb^2\omega_\alpha^2} \alpha = \frac{1}{2} \frac{\rho U_\infty^2 C_m S_{ref} c}{mb^2\omega_\alpha^2} \quad (2.59)$$

where the parameter  $x_\alpha$  indicates the nondimensional offset of the elastic axis and the center of gravity of the airfoil defined by Eqn. 2.60. Static imbalance is defined in terms of  $x_\alpha$  in Eqn. 2.61, moment of inertia,  $I_\alpha$ , is defined by Eqn. 2.62 and  $r_\alpha$  is the radius of gyration of the airfoil defined by Eqn. 2.63.

$$x_\alpha = \frac{2(x_{ea} - x_{cg})}{c} \quad (2.60)$$

$$S_\alpha = mx_\alpha \frac{c}{2} \quad (2.61)$$

$$I_\alpha = \frac{k_\alpha}{\omega_\alpha^2} \quad (2.62)$$

$$r_\alpha^2 = \frac{I_\alpha}{m} \quad (2.63)$$

Speed index can be defined as

$$V_f = \frac{U_\infty}{\omega_\alpha b \sqrt{\mu_{mass}}} \quad (2.64)$$

where  $\mu_{mass}$  represents reduced mass defined as,

$$\mu_{mass} = \frac{m}{\rho_\infty b^2} \quad (2.65)$$

The governing equation can be cast into the following matrix form,

$$[M]\{\ddot{q}\} + [C]\{\dot{q}\} + [K]\{q\} = \{F_{aero}\} \quad (2.66)$$

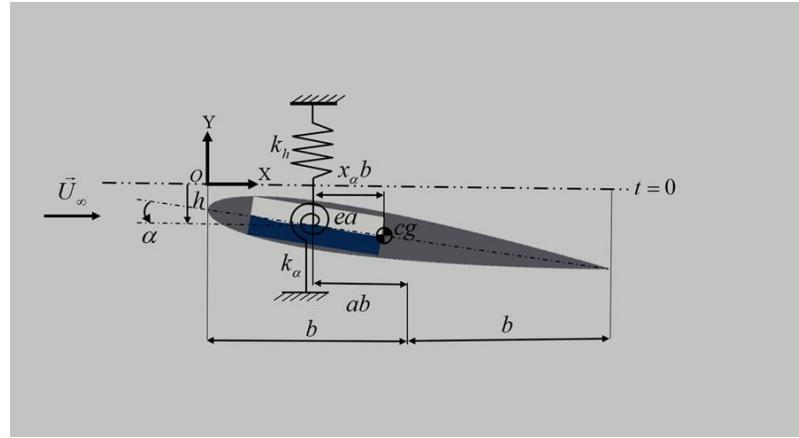
where,

$$[M] = \begin{bmatrix} 1 & x_\alpha \\ x_\alpha & r_\alpha^2 \end{bmatrix}, [C] = \begin{bmatrix} \frac{c_h}{m\omega_\alpha} & 0 \\ 0 & \frac{c_\alpha}{mb^2\omega_\alpha} \end{bmatrix}, [K] = \begin{bmatrix} \left(\frac{\omega_h}{\omega_\alpha}\right)^2 & 0 \\ 0 & r_\alpha^2 \end{bmatrix}, \{q\} = \begin{bmatrix} \bar{h} \\ \alpha \end{bmatrix}, \{F_{aero}\} = \frac{V^*}{\pi} \begin{bmatrix} -C_L \\ 2C_M \end{bmatrix}$$

For the present study, structural damping is ignored and the equation of motion is reduced to the following form,

$$[M]\{\ddot{q}\} + [K]\{q\} = \{F_{aero}\} \quad (2.67)$$

The aforementioned system of equations and physical quantities are utilized in *SU2* for solution of aeroelastic equations. These equations are modified to study the effects of sloshing on aeroelastic characteristics of the airfoil. The equivalent system of an airfoil embedded with a rectangular fuel tank is shown in Fig. 2.6. The equation of motion of this system is derived using the first principle: force and moment balance about the elastic axis of the airfoil.



**Figure 2.6: A 2-DOF airfoil with a partially filled embedded fuel tank immersed in a flowfield free to plunge and pitch**

$$m_{tot}\ddot{h} + S_{\alpha,tot}\ddot{\alpha} + k_h h = -L + F_{Y,ea} \quad (2.68)$$

$$S_{\alpha,tot}\ddot{h} + I_\alpha\ddot{\alpha} + k_\alpha\alpha = M + M_{OZ,ea} \quad (2.69)$$

where total mass  $m_{tot}$  is defined as the sum of the structural mass of the airfoil and mass of the contained fluid expressed by

$$m_{tot} = m + m_f \quad (2.70)$$

The equations are non-dimensionalized by dividing the force balance Eqn. 2.68 with  $mb\omega_\alpha^2$  and the moment balance Eqn. 2.69 with  $mb^2\omega_\alpha^2$  to obtain Eqn. 2.71 and Eqn. 2.72, respectively.

$$\frac{m_{tot}\ddot{h}\omega_\alpha^2}{m_{tot}b\omega_\alpha^2} + \frac{S_{\alpha,tot}\ddot{\alpha}\omega_\alpha^2}{m_{tot}b\omega_\alpha^2} + \frac{k_h h}{m_{tot}b\omega_\alpha^2} = -\frac{L}{m_{tot}b\omega_\alpha^2} + \frac{F_{Y,ea}}{m_{tot}b\omega_\alpha^2} \quad (2.71)$$

$$\frac{S_{\alpha,tot}\ddot{h}\omega_\alpha^2}{m_{tot}b^2\omega_\alpha^2} + \frac{I_\alpha\ddot{\alpha}\omega_\alpha^2}{m_{tot}b^2\omega_\alpha^2} + \frac{k_\alpha\alpha}{m_{tot}b^2\omega_\alpha^2} = \frac{M}{m_{tot}b^2\omega_\alpha^2} + \frac{M_{OZ,ea}}{m_{tot}b^2\omega_\alpha^2} \quad (2.72)$$

Cancelling the common terms in the previous equations and combining them together in a single matrix equation yields Eqn. 2.73, written in a condensed form in Eqn. 2.74.

$$\frac{m_{tot}}{m} \begin{bmatrix} 1 & x_\alpha \\ x_\alpha & r_\alpha^2 \end{bmatrix} \begin{Bmatrix} \ddot{h} \\ \ddot{\alpha} \end{Bmatrix} + \begin{bmatrix} \left(\frac{\omega_h}{\omega_\alpha}\right)^2 & 0 \\ 0 & r_\alpha^2 \end{bmatrix} \begin{Bmatrix} \bar{h} \\ \alpha \end{Bmatrix} = \frac{V^{*2}}{\pi} \begin{Bmatrix} -C_L \\ 2C_M \end{Bmatrix} + \frac{r^3}{mb\omega_\alpha^2} \begin{Bmatrix} F_{Y,ea} \\ 2rM_{OZ,ea} \end{Bmatrix} \quad (2.73)$$

$$\frac{m_{tot}}{m} [M] \{ \ddot{q} \} + [K] \{ q \} = \{ F_{aero} \} + \{ F_{slosh} \} \quad (2.74)$$

The obtained Eqn. 2.74 is very similar to the standard 2-DOF aeroelastic equation Eqn. 2.67 with the obvious difference of the forcing term on the right-hand side of the equation and a factor attached to the mass matrix. This factor is the ratio of the combined mass of the airfoil with embedded fuel to the mass of airfoil alone. For an empty tank, the equation becomes the same as the pure aeroelastic motion. Since this ratio is always more than unity for any amount of non-zero fuel, the effect is called *added mass effect*. Although the equation is derived for a 2-D embedded fuel tank inside an airfoil, it holds true for any configuration of the fuel, internal or external, and 2-D or 3-D, as long as the moment due to sloshing is computed about the elastic

axis of the airfoil and the system is of 2-DOF. The forcing terms,  $\mathbf{F}_{aero}$  and  $\mathbf{F}_{slosh}$  can be computed using simple aerodynamic models like DLM or VLM, high fidelity CFD for aerodynamics or sloshing, EMS model for sloshing, or any black-box surrogate model.

## Chapter 3

### Integrated Computational Framework

Simultaneous simulation of the multiple interacting phenomena is required to predict their combined effect. These numerical simulations for multiphysics systems require communication channels for interactions between different solvers for each of the physical systems. The present research involves a fluid-structure interaction system sub-divided into an aeroelastic solver for airfoil motion upon interaction with aerodynamics and a multiphase flow solver for the solution of sloshing loads due to the motion of the fluid container. This problem can be dealt with either developing an implicit formulation for the physical system, or by explicitly coupling different solvers. Here, an explicit coupling approach is followed by coupling individual solvers for each physical system. This chapter details existing coupling methods, their advantages, and disadvantages, the coupling approach taken for the present research and implementation details of those approaches.

The computational tools are divided into subsystems, where these subsystems are numerically modeled individually without having the coupled problem in mind. These solvers interact through bidirectional periodic data transfer at pre-prescribed time. The mechanism of data transfer must provide a data-mapping technique for non-conforming meshes for surface coupling between domains. Furthermore, the mechanism should be suited for black-box solvers as well, which in the present research corresponds to surrogate models comprising of radial basis function neural network.

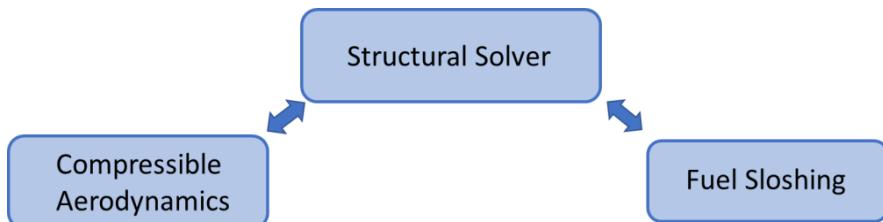
The aeroelastic subsystem consists of an airfoil immersed in a transonic flowfield which is free to plunge and pitch. The plunging and pitching motion is governed by the equation of motion of the structure with forcing terms computed from the pressure field obtained from the compressible flow solver. The airfoil has an embedded fuel tank partially filled with

incompressible fluid. The coupling is solely input and output data based on single physics solvers, which in this case are *SU2* and *OpenFOAM* for aeroelastic and multiphase flow solutions, respectively. Ideally, an implicit equation containing external aerodynamic loads and internal sloshing loads as forcing terms should produce the most accurate solutions. However, the present problem utilizes *SU2* and *OpenFOAM* which solve the aeroelastic and multiphase flow problem and casting the coupling in that form is challenging. The aeroelastic solver in *SU2* is already developed for implicit solutions. Furthermore, developing an explicit formulation provides convenience of playing with different model representations of individual physical systems.

The aeroelastic equation comprises of the structural part which is described by a mass matrix and a stiffness matrix which is excited by forcing terms from the fluid flow interactions. A simplified form of the equation of motion for the system is given by Eqn. 3.1,

$$[M]\{\ddot{q}\} + [K]\{q\} = \{F_{aero}\} + \{F_{slosh}\} \quad (3.1)$$

Here,  $M$  and  $K$  represent the equivalent mass and stiffness matrices of the system,  $q$  represents the modal displacements and the forcing terms  $F_{aero}$  and  $F_{slosh}$  represent the generalized loads due to external compressible flow and internal sloshing, respectively. The present research constrains airfoil and wing motions to plunge and pitch, however, this formulation holds for multi-mode motions as well. The computational structure is represented in Fig. 3.1 with structural solver at the coupling step.

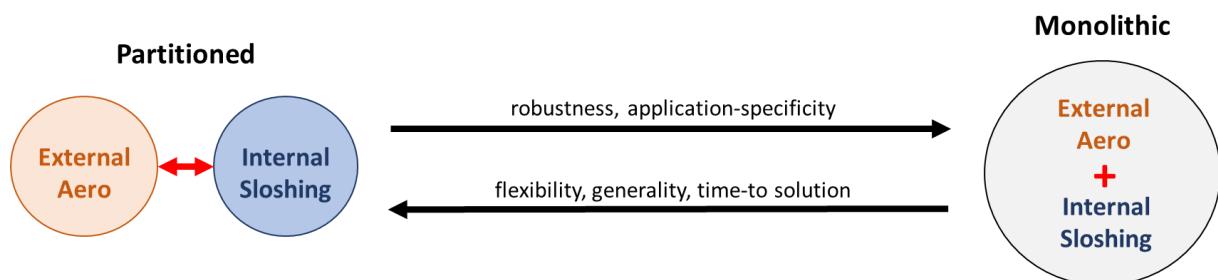


**Figure 3.1: Coupling structure of compressible flow solver for aerodynamics and multiphase solver with structural solver**

The following sections detail present methods for solver coupling, computational tools used and developed for this study and their implementation.

### 3.1 Coupling Methodology

Fluid-structure interaction problems are generally solved either by partitioning methods or by monolithic methods. Monolithic methods treat fluid and structure interaction at the interface synchronously and typically require multiple fluid-structure interaction iterations. Monolithic schemes maintain conservation properties at the interface, which guarantee unconditional stability. The direct consequence of this is the flexibility in choosing the time step size, which is limited only by the accuracy and resolution of the solution required. In partitioned schemes, the fluid and structural equations are integrated in time alternatively and the interface conditions are enforced asynchronously. A simple comparison of both approaches for the present research problem of coupling aerodynamic and sloshing loads with an aeroelastic airfoil or a wing is shown in Fig. 3.2. A partitioned based approach is followed by native *SU2* solver which is capable of solving fluid-structure interaction problems has been explained step-by-step in this section.



**Figure 3.2:** A simple comparison of partitioned and monolithic schemes for the present problem

## Fluid-Structure interaction using partitioned based method

The partitioned approach allows separate solvers for fluid equations and structural equations with appropriate boundary conditions at the interface at each global time step. The fluid and structural equations are alternately integrated in time with the enforcement of interface conditions.

The problem is broken down into two parts: implicit coupling of airfoil structural motion with external flowfield and explicit coupling of airfoil structure with internal sloshing loads. Structural coupling with aerodynamics or pure aeroelasticity of airfoil is solved using native fluid solver in *SU2* casted into arbitrary Lagrangian-Eulerian (ALE) formulation to take moving grids into account. Vertex based control volume meshes are used for flow integration in space and dual-time stepping is followed for time marching. Surface coupling in implicit formulation is followed for the solution of coupled problem. Native methodology followed in *SU2* is used as it is for the pure aeroelastic problem. The step-by-step operations for solving a fluid-structure interaction problem are shown in Fig. 3.3.

Time coupling at the fluid-structure interface along with boundary conditions as detailed by Sanchez [68, 69] are detailed as follows:

1. The computational domain is initialized with fluid and structural variables. The airfoil or wing is considered a fixed shaped body with a rigid motion in the pitch and plunge directions. A third entity, interface, is defined for partitioned schemes where the fluid and the structure interact.



2. Fluid state and mesh velocity  $\dot{u}_m$  is predicted from the previous time step i.e.  $\vec{V}_f, \rho, E, P$  predicted from  $t = n - 1$  or the initial condition in the fluid solver

$$\lambda_{f,\Gamma} = T_f(\vec{V}_f, \rho, E, P, \dot{u}_m) \quad (3.2)$$

where,  $T_f$  is the fluid traction force and  $\lambda_{f,\Gamma}$  is defined as  $\lambda_{f,\Gamma} = \int_{\partial\Omega_\Gamma(t)} \sigma_f \vec{n}_f d\Omega_\Gamma(t)$

$\vec{V}, \rho, E, P$  predicted from  $t = n - 1$



3. At the fluid and solid interface, the structural and fluid tractions must be equal. Hence, the traction boundary condition is applied at the interface.

Traction force of the structure is defined as  $T_{s,\Gamma} = \lambda_{s,\Gamma}(\vec{u}_s, \dot{\vec{u}}_s)$

where,  $\lambda_{s,\Gamma} = \int_{\partial\Psi_\Gamma(t)} \sigma_s \vec{n}_s d\Psi_\Gamma(t)$

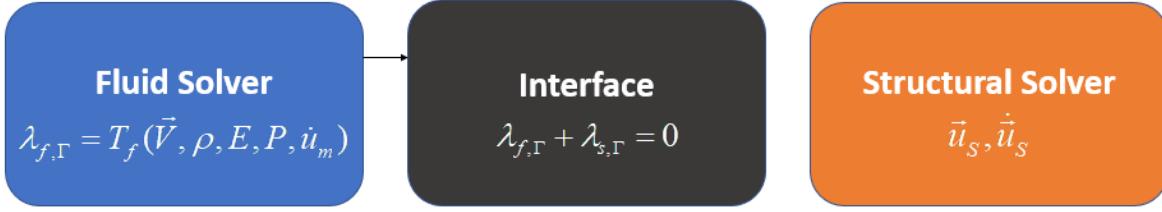
Traction boundary condition,

$$\lambda_{f,\Gamma} + \lambda_{s,\Gamma} = 0 \quad (3.4)$$

Continuity boundary condition at the interface is given by,

$$\vec{V}_f = \vec{u}_\Gamma = \vec{u}_s \quad (3.5)$$

where  $\vec{V}_f$  is the fluid velocity at the interface,  $\vec{u}_m$  is the interface velocity at the interface and  $\vec{u}_s$  is the structural velocity at the interface.



where the fluid traction is defined as,

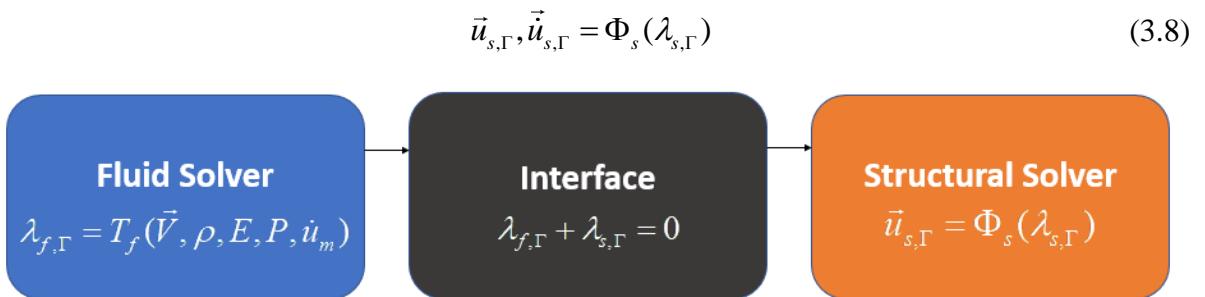
$$\lambda_{f,\Gamma} = \int_{\partial\Omega_\Gamma(t)} \sigma_f \vec{n}_f d\Omega_\Gamma(t) \quad (3.6)$$

Here, the fluid domain is represented by  $\Omega_\Gamma$  and  $\sigma_f$  represents stresses due to the fluid flow. The structural traction is defined as,

$$\lambda_{s,\Gamma} = \int_{\partial\Psi_\Gamma(t)} \sigma_s \vec{n}_s d\Psi_\Gamma(t) \quad (3.7)$$

where  $\Psi_\Gamma$  represents the structural domain and  $\sigma_s$  represents the structural stresses.

4. Structural displacement and mesh velocity calculated from the traction boundary condition at the interface. The structural displacements computation is represented by a function  $\Phi_s$  as shown.

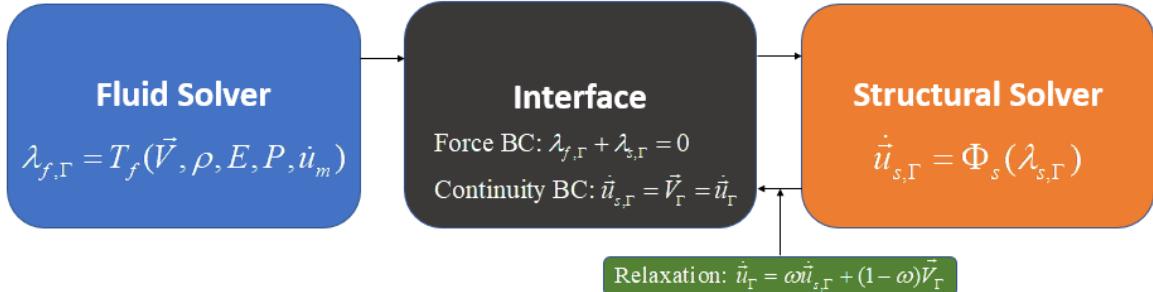


5. The structural displacements and mesh velocity computed from the traction boundary conditions at the interface are fed back to the interface after a relaxation operation. This operation is necessary for stability and convergence when large structural deformations are encountered, as illustrated by Tallec and Mouro [70]. The relaxed displacements are updated to the common interface for the application of continuity boundary condition.

$$\dot{\vec{u}}_{\Gamma} = \omega \dot{\vec{u}}_{s,\Gamma} + (1-\omega) \vec{V}_{\Gamma} \quad (3.9)$$

Continuity boundary condition applied at interface

$$\dot{\vec{u}}_{s,\Gamma} = \vec{V}_{\Gamma} = \dot{\vec{u}}_{\Gamma} \quad (3.10)$$



6. Mesh velocity updated from the common interface is transmitted to the fluid domain mesh points at the interface as per Eqn. 3.11

$$\dot{\vec{u}}_{m,\Gamma} = \dot{\vec{u}}_{\Gamma} \quad (3.11)$$

The mesh velocity and deformed mesh positions of the rest of the fluid domain is calculated using spring analogy formulation given by Eqn. 3.12,

$$\dot{\vec{u}}_{\Gamma} = \mathbf{F}_{spring}(\dot{\vec{u}}_{m,\Gamma}) \quad (3.12)$$

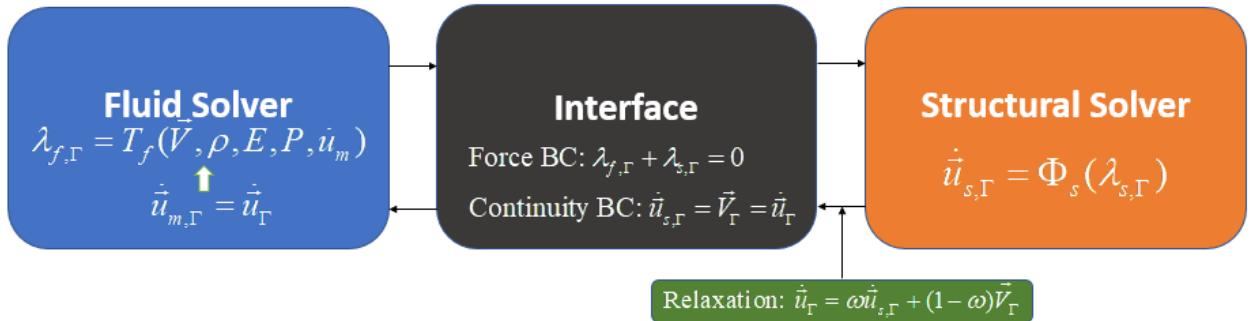
The dynamic mesh model uses a spring-based smoothing method in which the edges between any two nodes are idealized as a network of interconnected springs. The initial spacing of the edges before any boundary motion constitutes the equilibrium state of the mesh. A displacement given at a boundary node generates a force proportional to the displacement along the spring connected to the node. Force can be written using Hooke's law given in Eqn. 3.14,

$$\vec{F}_i = \sum_j^{n_i} k_{ij} (\Delta \vec{x}_i - \Delta \vec{x}_j) \quad (3.14)$$

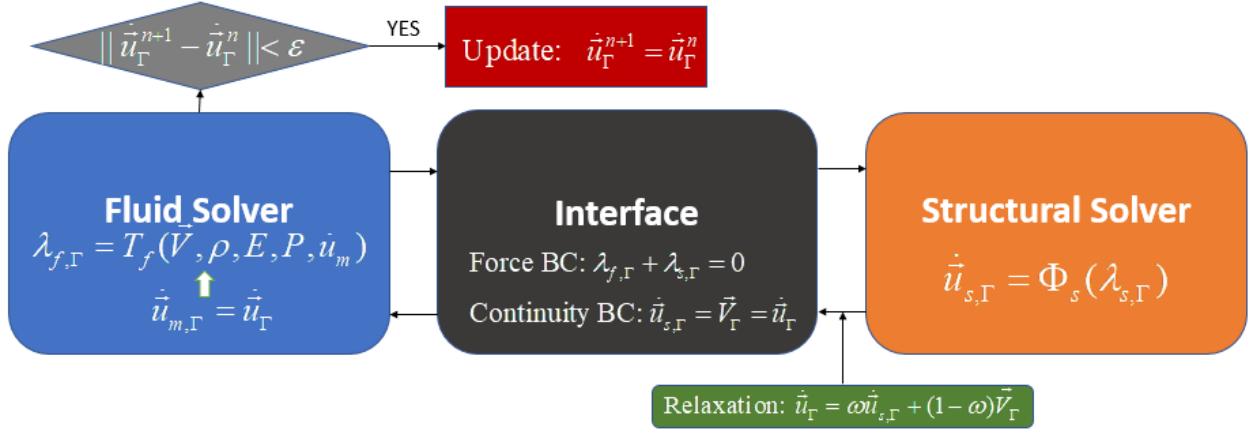
where  $\vec{F}_i$  is the net force on the  $i^{th}$  node,  $\Delta\vec{x}_i$  and  $\Delta\vec{x}_j$  are the displacements of the  $i^{th}$  and neighboring  $j^{th}$  node,  $n_i$  is the number of neighbors of the node  $i$  and spring constant of the edge connecting the node  $i$  and node  $j$  is defined as,

$$k_{ij} = \frac{1}{\sqrt{|\vec{x}_i - \vec{x}_j|}} \quad (3.15)$$

Net force on all the nodes of the mesh is zero at each instance of dynamics equilibrium i.e.,  $\vec{F}_i = 0$ . Volume conservation of the mesh elements should be satisfied for closure of the above system of equations.



7. Local convergence of mesh velocity to meet a residual criteria are followed. This is done to avoid instabilities due to staggered scheme followed in the coupling. This method tries to eliminate energy errors resulting from not enforcing the coupling conditions on the coupled time integration.



The fluid variables are updated in the updated domain and **Step 1 to Step 6** are repeated until local convergence is reached for each global time step.

Convergence criteria,

$$\|\dot{u}_\Gamma^{n+1} - \dot{u}_\Gamma^n\| < \epsilon \quad (3.13)$$

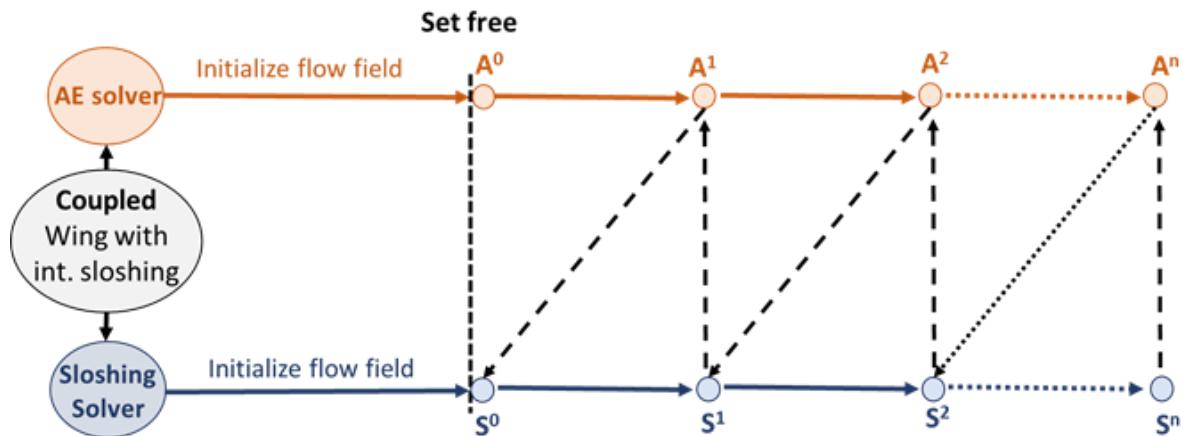
then, update  $\dot{u}_\Gamma^{n+1} = \dot{u}_\Gamma^n$  and  $t \rightarrow t + \Delta t$

This sequential process allows solver modularity which enhances flexibility in solver type with plug-and-play features. It requires only one fluid and structural solution per time step of the coupled problem, individual solvers can have multiple solutions with smaller time steps. This staggered type arrangement can lose the conservation properties of the continuum fluid-structure system due to time lag. Hence, the coupling time step size must be carefully tuned to minimize the lag as well as maintain the stability of the solution.

The native aeroelastic solver of *SU2* is put to use for numerical validations and computations requiring pure aeroelastic motion without the effects of sloshing fluid. The flutter boundary of NACA64A010 is validated with the numerical results of Alonso [17] in the transonic regime. The three key aeroelastic responses namely, damped motion, neutral motion, and divergent (flutter) motion are observed at various flow conditions. Furthermore, limit cycle oscillations are also observed post-flutter when longer simulation run-times were chosen. This solver was

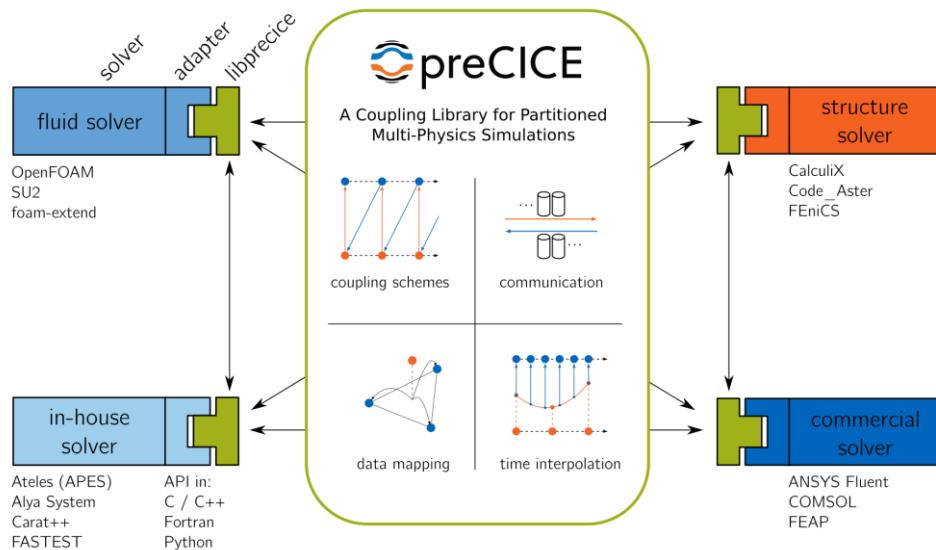
also used to simulate aeroelastic motion on NACA0012 airfoil. The numerical results for steady and unsteady motions are validated with published results. The aeroelastic properties of NACA0012 are tweaked for the present study in order to maintain low natural frequency for plunging and pitching motion. This is done to ensure a non-chaotic sloshing response in the embedded fuel tank. Lastly, NASA benchmark supercritical wing (BSCW) has been used to extend the study to 3D flowfields and native *SU2* solver is utilized to generate steady and unsteady solutions. The detailed description and results for the aforementioned airfoil and wing geometries are given in the Results and Discussions chapter of this thesis.

The *SU2* aeroelastic solver is required to be coupled with a multiphase flow solver to study the combined effects on structural motion. *OpenFOAM*'s multiphase solver *interDyMFoam* is used to solve the sloshing flow inside the containing tank susceptible to motion. Coupling the solvers in time-domain is required to achieve the framework shown in Fig. 3.1. The coupling library utilized for this problem, *preCICE* [47], offers coupling options for both implicit coupling scheme which requires iteration over a coupled equation until the solution converges, and explicit scheme that comes with a fixed number of solver call for a given number of time steps. The aeroelastic and multiphase solvers are coupled and integrated in time using a partitioned approach adopting a loosely-coupled staggered procedure as shown in Fig. 3.3.



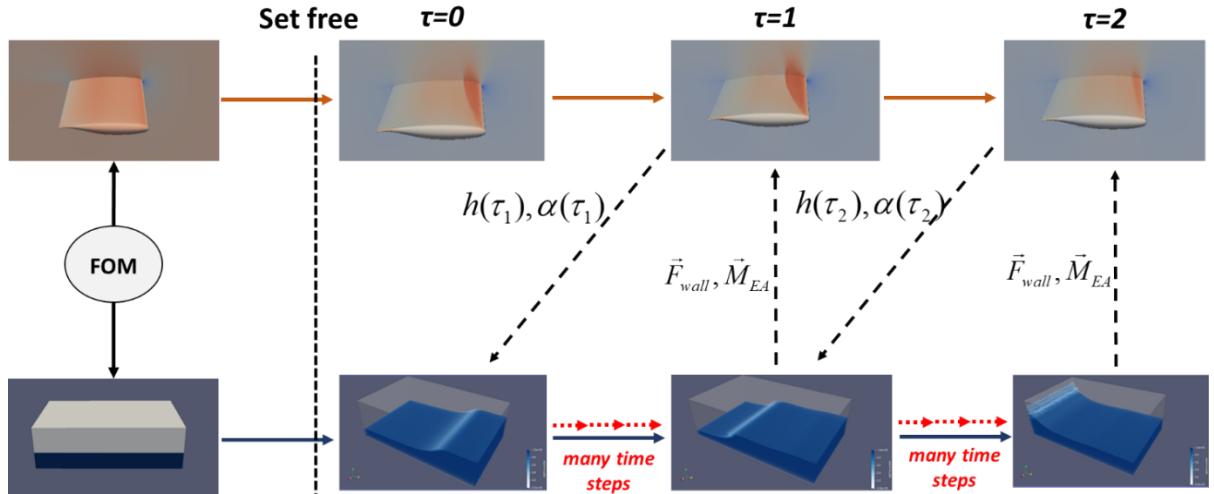
**Figure 3.3: Staggered coupling of aeroelastic solver and sloshing solver in time**

The external flowfield is first solved for a steady solution about the mean position of the airfoil or wing. This is followed by a few cycles of forced pitching motion to perturb the flowfield. The airfoil is then set free to move with its motion governed by aeroelastic equation of motion. The sloshing solver is activated at the same time and the flow loads due to sloshing are coupled with the aeroelastic solver in a staggered manner. The coupling library, *preCICE*, which provides the solver interface is modified for the present problem. The *preCICE* structure constitutes a central solver interface that handles coupling schemes, the establishment of communication channels between the flow solvers, data mapping and handling of non-conforming meshes in different solvers for surface coupling, spatial and temporal interpolation, etc. This central structure communicates with an *adapter* that is specific to a solver involved in the coupling process. The solver-specific adapters act as a communication link between the solver and the interface, which is able to process variables in the solver format as well as readable for the solver interface. The *preCICE* library structure, as well as some solvers compatible with this library, are shown in Fig. 3.4 borrowed from its online documentation [71].



**Figure 3.4: Coupling library *preCICE* for runtime solver interactions data transfer**

The library is equipped with adapters for various fluid solvers, structural solvers, and in-house codes. The adapters for both *SU2* and *OpenFOAM* are developed and provided by *preCICE*. However, these adapters are devised for specific cases and deal with a specific set of flow and structural variables that cannot be used in this work in the native state. Furthermore, the adapters are designed for surface coupling, i.e. the solvers share a common surface that is part of both physical domains. The rectangular fuel tank embedded in the airfoil/wing in this problem does not share any common surface with the external aerodynamics and hence the coupling procedure must be modified to accommodate communication of integrated field quantities, in this case, boundary pressures, and structural displacements in the form of plunge and pitch between *SU2* and *OpenFOAM*. The desired coupling operation is shown in Fig. 3.5 demonstrating the aeroelastic motion of NASA BSCW under the influence of external transonic flow and internal fuel sloshing computed by *SU2* and *OpenFOAM*, respectively.



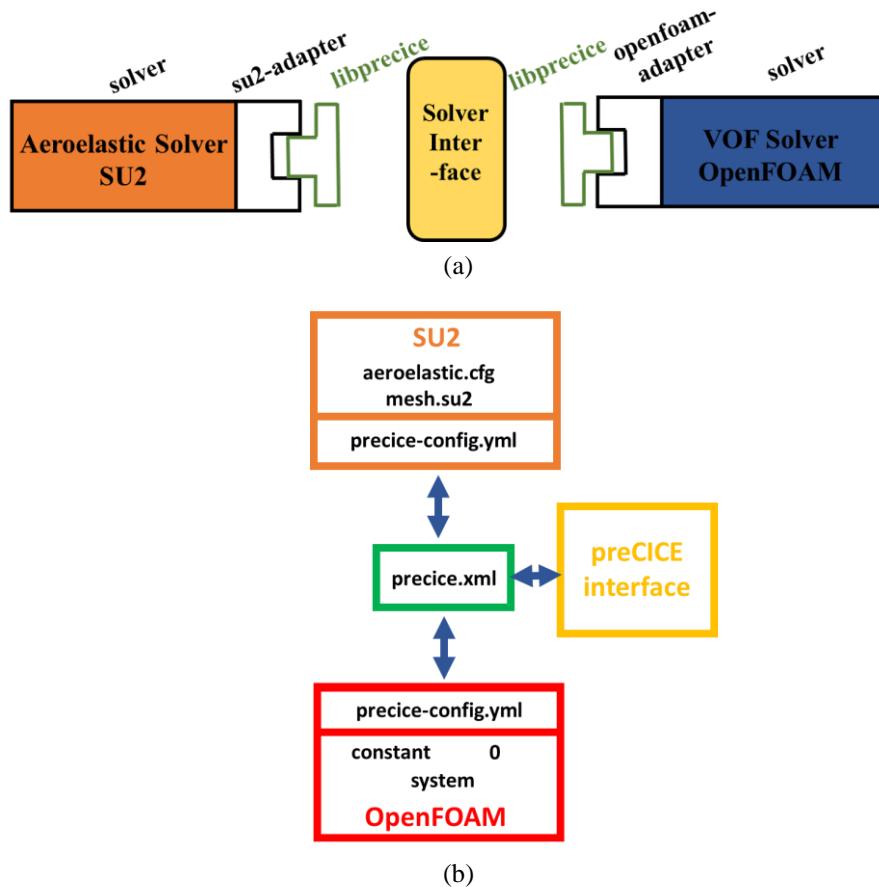
**Figure 3.5: Aeroelastic motion of NASA BSCW immersed in transonic flow and under the influence of sloshing fuel in internal fuel tank coupled using *preCICE***

The coupling is done explicitly in a staggered manner as previously demonstrated by Fig. 3.3. The wing is set free for aeroelastic motion after perturbation of the external flowfield as described earlier. The structural displacements in the form of plunge and pitch represented by

$h$  and  $\alpha$ , respectively, at the present time step,  $\tau_l$ , computed for pure aeroelastic interaction with external flowfields and the motion information is transmitted to the multiphase solver via solver interface. The multiphase solver excites the containing tank with the same motion. However, the time step size of the multiphase solver may not be the same as that of the aeroelastic solver. Usually, it requires about 2-10 times smaller time steps for the multiphase solver to get converged results. Therefore, the motion information received from the aeroelastic solver is subdivided over the number of time steps utilized by *interDyMFoam* to generate a smooth motion. This problem could be avoided by allocating the same time step value to both solvers. However, this increases the computational cost of the system without significantly improving its accuracy. The fluid enclosed in the tank sloshes due to its structural motion and in return generates a pressure field comprising of dynamic and static pressure components. The pressure fields are integrated along the tank surface to compute forces and moments in the direction of aerodynamic lift and moment along the elastic axis location of the enclosing wing. These integrated quantities are computed by the multiphase solver, *interDyMFoam*, at each physical time step corresponding to the structural time and the information is transmitted back to the aeroelastic solver via the interface. The sloshing loads are taken into account to solve the aeroelastic equation for the next time step, and so on. To implement this idea, the existing *preCICE* adapters for both *SU2* and *OpenFOAM* must be modified as per the requirements of this problem. The simplified coupling structure for *SU2* and *OpenFOAM* using *preCICE* library is shown in Fig. 3.6.

The runtime settings for *preCICE* are configured by *.xml* file which specifies the solver participants for the coupled simulation. It also includes information of which data values are to be exchanged between the solvers, the mesh information, which data mapping techniques are to be used and the establishment of the communication channels. However, the present problem does not need to utilize all these capabilities inherent to the *preCICE* libraries such as mesh

mapping and rely upon the communication channel establishment capability of the library. The information from the flow solvers interacts with the solver interface via the dedicated adapters. The modification in the existing *preCICE* adapters are detailed in the coming sections starting with *SU2*, then *OpenFOAM* and finally an adapter to enable coupling with black-box solvers.



**Figure 3.6: (a) Coupling strategy of *SU2* and *OpenFOAM* via an interface, and (b) Coupling structure of *SU2* and *OpenFOAM* flow solvers using *preCICE***

### 3.2 SU2 Adaptation for Coupling

*SU2* is an open-source software developed at the Aerospace Design Laboratory of Stanford University. *SU2* is mainly written in C++ using a modular code structure. It is designed for both steady and unsteady flow solutions and has built-in aeroelastic solver. *SU2* requires two files to run a regular simulation, a *configuration* file that contains all the information required by the software for the run, and a *mesh* file that contains the dimensionality, geometry

information, computational grid points, connectivity information and marker names. For cases that initialize from a flow solution, one or two solution files are required depending on the discretization scheme used. For the present work, aeroelastic solution is required to be coupled with sloshing solutions in run time via data exchange through the solver interface. Euler equations cast into ALE form are utilized for solving the external flow. Dynamic meshes are used to accommodate the motion of the wing structure. The intrinsic FSI capability of *SU2* for 2-DOF airfoil motion is not altered in the present work.

The *preCICE* documentation provides FSI tutorial with *SU2* and *CalculiX* utilizing *SU2* for fluid simulation and *CalculiX* for finite element. The *SU2* adapter is designed to write fluid forces and read structural displacements. This adapter must be modified such that *SU2* can read integrated pressure loads on the tank walls and write structural displacements obtained from the solution of the aeroelastic equation. The *preCICE* class encapsulates the coupling related capabilities in *SU2*. The library has an invasive nature and its adapted version is integrated in the source code for *SU2*. However, it comes with a provision to turn off the coupling modules and run *SU2* as original.

The coupling information is given in the configuration file of *SU2*, which has the following new options given with their default name:

`PRECICE_USAGE = YES, NO` (3.14)

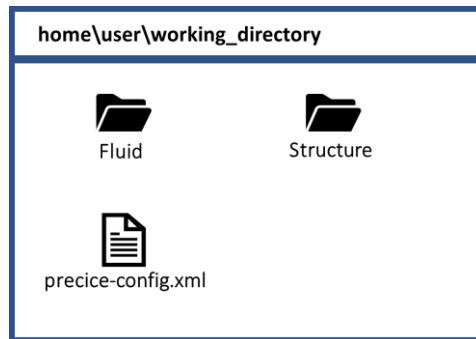
`PRECICE_CONFIG_FILENAME = precice-config.xml` (3.15)

`PRECICE_VEBOSITYLEVEL_HIGH = NO, YES` (3.16)

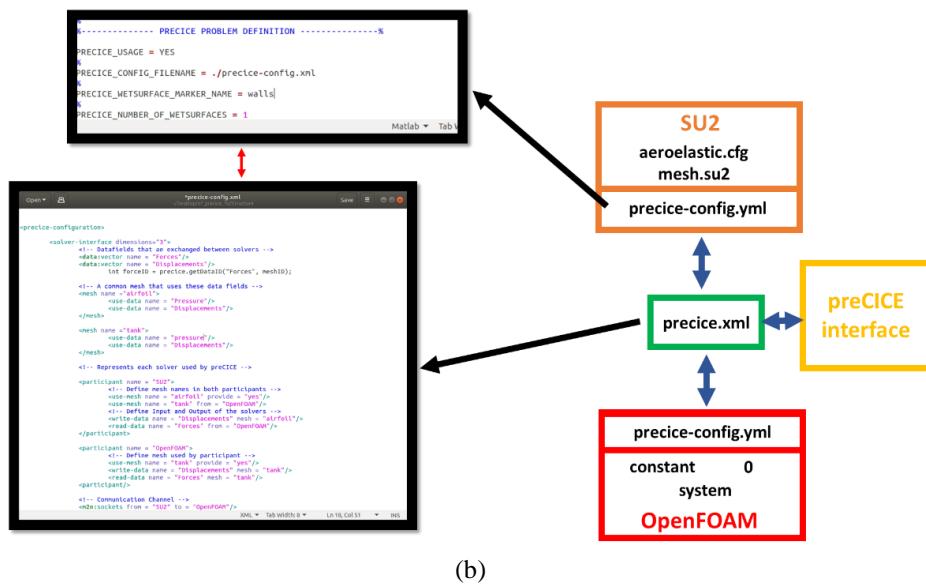
`PRECICE_WETSURFACE_MARKER_NAME = wetSurface` (3.17)

The first option is the most obvious, whether or not to engage *preCICE* for the given simulation run. The remaining three options become irrelevant if ‘*NO*’ is chosen. If set to ‘*YES*’. The

second line states the name of the configuration file for *preCICE* which contains coupling details such as the dimensionality of the problem, data fields to be exchanged, the identifiers for meshes engaged in coupling for both participating solvers, data fields that are to be read and data fields that are to be written by each participant solver, details of the communication channel to be set-up, the type of data mapping and a post-processing block. A simplified working of this *precice.xml* is shown with the help of a block diagram in Fig. 3.6 (b). For ease of implementation, the working location of this file and snapshot of the file for the present work with reference to the block diagram in Fig. 3.6 (b) are shown in Fig. 3.7 (a) and Fig. 3.7 (b), respectively. The directory named *Fluid* contains the simulation files for the sloshing case and *Structure* contains the files required to run the aeroelastic simulation.



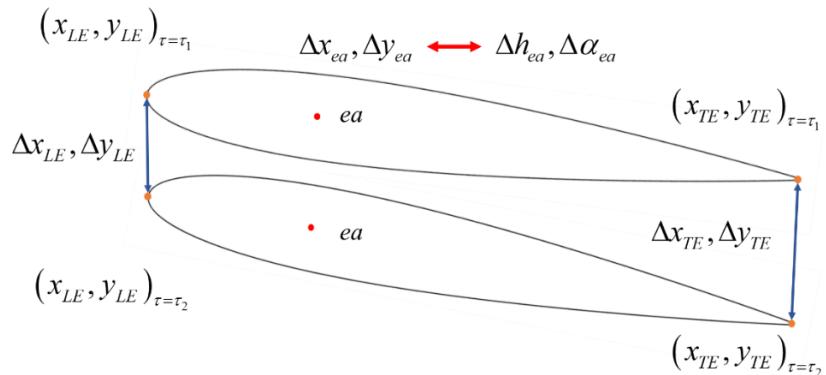
(a)



(b)

**Figure 3.7: (a) Representation of configuration file for solver interface *precice-config.xml* , and (b) block diagram representation of the configuration files for *preCICE* and *SU2***

The third configuration option PRECICE\_VERBOSITYLEVEL\_HIGH gives the user the option to output the working information of the solver interface at a certain checkpoint if checked ‘YES’. The fourth and the final option PRECICE\_WETSURFACE\_MARKER\_NAME denotes the *SU2* marker that partakes in coupling. It must be reiterated that *preCICE*, by default, is a surface coupler. It requires complete information of the coupled or ‘shared’ surface in terms of geometric coordinates, internal reference of the solver and values of the physical quantities. However, the problem at hand requires interaction of condensed information. The aeroelastic solver, *SU2*, is required to write structural displacements and read sloshing force in the lift direction and moment due to sloshing at the elastic axis. These quantities are utilized in the 2-DOF aeroelastic equation of motion. The marker required for the coupling is chosen as the airfoil surface. By default, *preCICE* writes the displacement of all the grid points in the computational domain. However, the aeroelastic motion considered for this problem is constrained to plunge and pitch only, and no shape deformation of the airfoil is considered. Hence, the information of motion of any two known grid points on the airfoil surface is enough to determine the motion of the elastic axis in terms of pitching and plunging. This property of rigid bodies is utilized in this study. The locations of the leading edge and the trailing edge of the airfoil are tracked to compute plunge and pitching displacement of the elastic axis as illustrated in Fig. 3.8.



**Figure 3.8: Computation of displacement of the elastic axis of the airfoil in terms of pitch and plunge from displacement information of surface points of the airfoil.**

Dynamic mesh motion is required to accommodate the aeroelastic motion of the airfoil. *SU2* provides certain options for mesh motions like *RIGID* and *DEFORMING* for oscillating airfoil motion. The solver uses flow equations cast into ALE formulation as detailed in the earlier chapter on flow solvers. *preCICE*, however, mandates a new grid movement option given as,

**GRID\_MOVEMENT = YES**

**GRID\_MOVEMENT\_KIND = PRECICE\_MOVEMENT, RIGID, DEFORMING**

These capabilities of *preCICE* are not modified in the current study. The idea is to make as little changes in the source code as possible to serve the coupling problem. The steering capability of *preCICE* is of utmost importance. This is not only important for ending the coupled simulation, but also holds the individual solver until the data from the other solver is not communicated. The time step size requirement for converged and stable solutions for aeroelastic solver is different than that for the multiphase flow solver. The coupling interface provides flexibility in choosing different time steps for different participating solvers. The standalone aeroelastic results generated in this study utilizes about 36 to 72 time steps for a typical aeroelastic cycle i.e. one pitching cycle at its natural frequency. The sloshing solutions, however, sometimes require a smaller time step for stable computation. In this work, the time step size is typically kept about 0.5 to 0.2 times the aeroelastic time step for high frequency motion, e.g. free motion of NACA64A010. However, for smaller natural frequencies, e.g. around 1 Hz for NACA0012, the time step size for sloshing solver is kept the same as that of the aeroelastic solver. The next section will discuss the modification in *OpenFOAM* to facilitate the coupling with *SU2*.

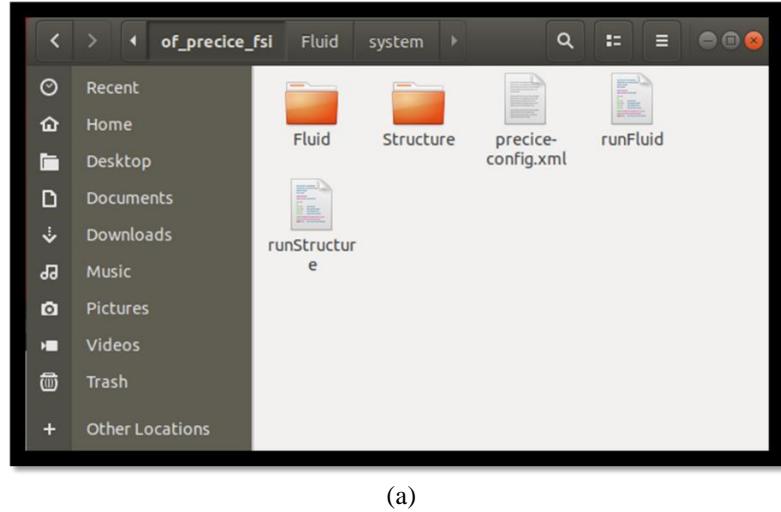
### 3.3 OpenFOAM Adaptation for Coupling

The *preCICE* adapter for *OpenFOAM* in the native form is developed for two sets of problems: conjugate heat transfer (CHT) and fluid-structure interaction (FSI) problems. The main class

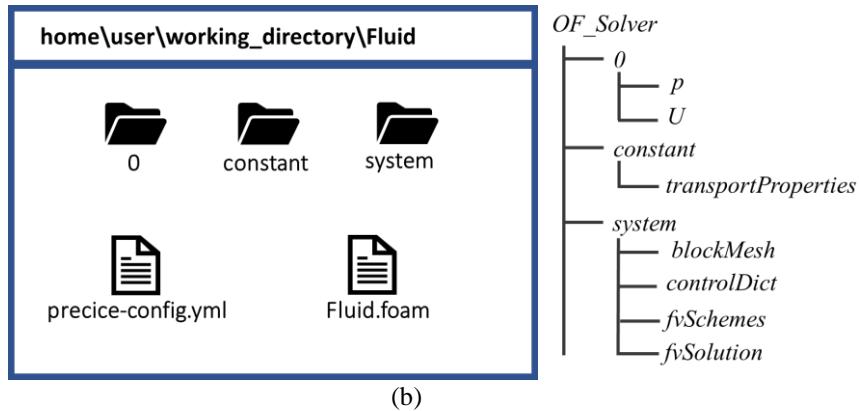
*Adapter* provides methods that can be called from the modified adapter specific to *OpenFOAM*. The *preCICE* adapter for *OpenFOAM* has been designed for extendibility in mind. Chourdakis's [72] thesis details the generalization procedure for the *OpenFOAM* adapter. Minimal changes in the source code of the adapter and the solver are desired. Although the work mainly deals with the conjugate heat transfer problem, the same ideas can be extended for FSI capabilities using OpenFOAM solvers.

This problem employs *interDyMFoam* solver of *OpenFOAM* to compute the sloshing loads due to structural motion. The coupling is initialized when the airfoil is set free after perturbing the flow field by a few cycles of forced pitching motion with a small amplitude, typically  $1^\circ$  at natural pithing frequency. The fuel tank in the form of a rectangular container is partially filled with incompressible fluid. The structural motion of the airfoil is obtained at the convergence of the first time step. Since the system is 2-DOF, the airfoil motion can be expressed by plunging and pitching only and no shape deformation is considered. As detailed in the previous section, the effective motion of the elastic axis of the airfoil is computed and transferred to the sloshing solver.

Now, the configuration file coupling, *precice-config.xml* is common to both participating solvers and is located as shown in the representative picture in Fig. 3.7 (a). Since the coupling parameters and settings that can be configured by this file in both solvers are already explained, the settings specific only to *OpenFOAM* will be discussed here. The multiphase solver in this coupling is referred to as *Fluid* and a screenshot of the directory used for coupling in this work is shown in Fig. 3.9 (a). *OpenFOAM* file structure generally contains three directories to provide the complete information of the simulation, often placed in the same directory as shown by a representative picture in Fig. 3.9 (b).



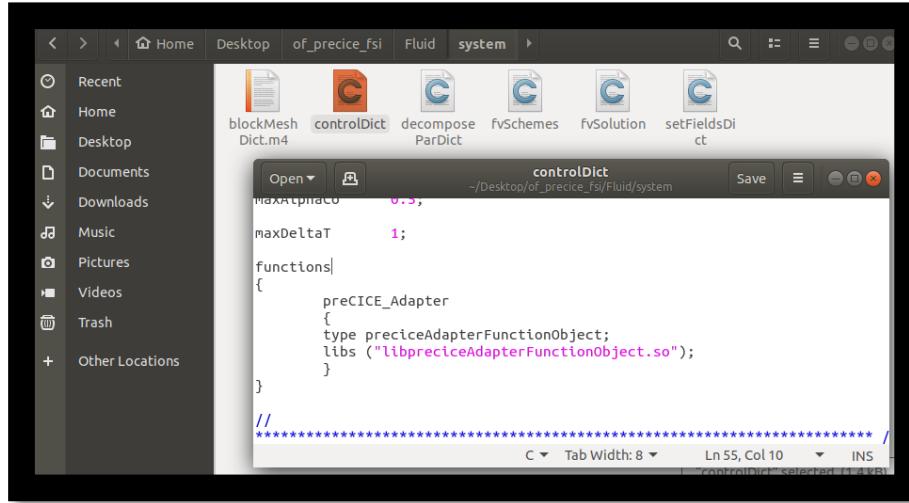
(a)



(b)

**Figure 3.9:** (a) Coupling directory containing participant solvers, and (b) *OpenFOAM* working directory with *preCICE* adapter

The initial conditions for the flow are defined in the “0” dictionary, the model and physical properties of the variables involved in the problem are defined in the “constant” directory, and “system” contains configuration files that control the numerical schemes, simulation settings such as time step size and total simulation run time in *controlDict*, and geometry and mesh connectivity information settings in *blockMesh*. The simulation run requires sequential execution of these files in a specific order, generation of the computational domain, initialization of variables and running the time marching solver. In order to run coupled simulation via *preCICE*, a short snippet of code must be appended in the *controlDict*, as shown in Fig. 3.10. This is analogous to the code snippet added to the *SU2* configuration file to activate coupling in the previous section.



**Figure 3.10: Code snippet added to *controlDict* file in *OpenFOAM* to activate the coupling**

As apparent from Fig. 3.9 (b), another configuration file is required within the *OpenFOAM* directory, *precice\_config.yml*. This contains solver information and acts as a bridge between the solver and the central *.xml* configuration file. The local configuration file contains the identifier for the participant, i.e. *Fluid* for this case, the reference to the central configuration, coupling mesh and patch (surface) and the data to be read and written by the solver, in this case, *displacement* and *loads* (force in lift direction and moment about the elastic axis location of the airfoil). A code snippet from the *.yml* file is given here.

---

```

participant: Fluid
precice-config-file: precice-config.xml
interfaces:
- mesh: tank_mesh
  patches:
  - walls
write-data: Forces
read-data: Displacements

```

---

The default structure of the solver coupler is designed for surface coupling and extracts pressure data from the *wetSurface*, which is the tank wall in case of sloshing. The present problem, however, does not demand surface coupling and the integrated pressure values along the surface is required to be written. *OpenFOAM* allows users to define and implement custom functions for the computation of forces and moments at each time step as an output. The integration of vertical forces along the tank surface and moments about the airfoil elastic axis is shown earlier in Fig. 2.2. This has been implemented in the present study. A code snippet added to *controlDict* for computation of forces on the patch “wall” is given below.

---

functions

```
{
    force_walls
    {
        type      forces;
        functionObjectLibs ("libforces.so");
        outputControl  outputTime;
        patches      (walls);
        pName       p;
        UName       U;
        rhoName     rhoInf;
        porosity   no;
        log         yes;
        rhoInf     1000;
        CofR       (0 0 0);
    }
}
```

---

It has been mentioned before that the time step size for participating solvers do need not be the same for the individual participating solvers in the coupling. In fact, the time step size for the multiphase solver is usually smaller than that of aeroelastic solver for high natural frequency airfoils, such as NAC64A010. The coupling takes place at larger of the two solver's time step size. The sloshing forces and moments, however, take place at every solver time step.

The interaction of *SU2* and *OpenFOAM* via solver interface helps achieve the staggered type coupling explained earlier and shown in Fig. 3.3. The single physics participant solvers are validated against published numerical and experimental results for both steady and unsteady cases, as applicable. The validations, results, and comparisons of aerodynamics, aeroelasticity, and sloshing are given in detail in the Results and Discussions chapter. This thesis explores surrogate models for both aerodynamics and sloshing and uses those models to study the aeroelastic behavior of the airfoil. The coupled results obtained from this methodology are used as reference for those studies and are considered as ground truth.

## Chapter 4

# Machine Learning Surrogate Models

The coupled aero-structural-fuel sloshing problem requires the coupling of an aeroelastic solver and a multiphase flow solver during runtime as demonstrated in the previous chapter. The solvers interact through a solver interface. In this study, aeroelastic solver, *SU2*, has been coupled with multiphase flow solver, *interDyMFoam*, by modifying *preCICE* solver interface codes, as demonstrated in the previous chapter. Coupling solvers through the interface alleviates the problem of coupling. However, the computational costs for such high fidelity CFD solutions still remain an outstanding issue. To mitigate the issue of expensive computations for every run of coupled CFD solutions, the development of surrogate reduced order models is a natural choice for time marching solutions. This chapter discusses the principles of surrogate modeling of dynamic systems, the choice of surrogate models for different physical systems involved in the coupled problems and the computational savings achieved.

### 4.1 Principles of Dynamic Systems Modeling

Solving the governing equations of physical systems may not be an easy task. Some of the common problems faced are inadequate spatial and temporal discretization of the computational domain, unstable numerical scheme, multiphysics problems posing solutions requirements from different physical systems, and high computational costs of the solution. Constructing simplified and cheap models that can accurately represent the underlying physical systems is an extremely effective solution to this problem.

The important question that arises is the approach for modeling any system based on the prior knowledge of its behavior. In general, there are three distinctly different methods for modeling

engineering systems, namely, black-box, white-box, and grey-box modeling, as detailed by Sjöberg [73] and Ljung [74].

- (i) Black-box modeling: Black-box models are functional relationships that map the system's inputs to outputs. Black-box models do not necessarily require a detailed understanding of laws governing the system's behavior. These models are usually defined from a set of flexible existing model structures, that are tuned as per available data to match the dynamic behavior of the source system. Some of the models most commonly used for such approach are ARX, NARX, ARMAX, state-space and neural networks.
- (ii) White-box modeling: White-box models (also called glass box or clear box) are based on diligent and extensive physical modeling using first principles. Essentially all necessary information is known, i.e. what type of functions relate to different system variables.
- (iii) Grey-box models: Most system identification problems involve a system that is somewhere between a black-box and a white-box model. This means certain information about the system behavior is known; however, this information is not enough to precisely determine the system response. In other words, grey-box system identification has a known structure with unknown parameters. Accurate approximation of these parameters from given information is the objective of good surrogate modeling efforts.

The objective is to construct surrogate models for the following physical two systems, (i) aeroelastic model of 2-DOF airfoil and wings, and (ii) incompressible fluid sloshing inside a moving fuel tank. In the present study, the CFD solutions for the aforementioned physical systems are considered as ground truth for training and validation of surrogate models'

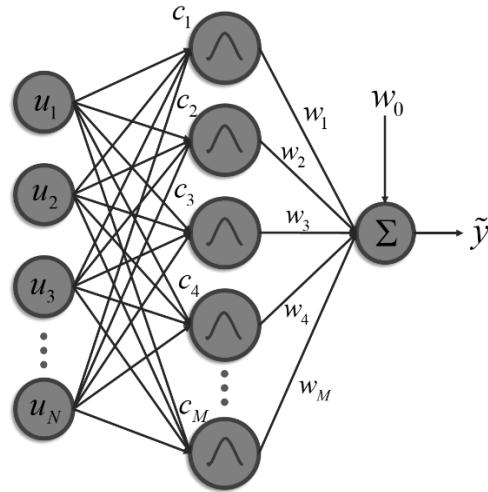
prediction accuracy. It must be restated that the development of surrogate models is to find a computationally cheaper alternate to make the coupled solution more feasible for early design stage iterative studies and optimization problems.

A few CFD runs of stand-alone aeroelastic motion of an airfoil or a wing immersed in a transonic flowfield reveal its aeroelastic characteristics. For example, the aeroelastic characteristics of NACA64A010 is observed by varying the flutter speed index at a fixed Mach number to obtain a sub-flutter damped motion response, a neutral response with a near-constant amplitude of motion, and a divergent (flutter) response, where the motion amplitude increases with time until limit cycle oscillations set in. By observing the resultant free-motion amplitudes and frequencies of the airfoil, the range of expected coupled motion response can be approximated. Since the behavior of physical systems only in the limited range of motions is of interest, the surrogate model is required to mimic the systems' dynamic response on the given range. As shown in later sections, this simplifies the surrogate model architecture choice, as well as the generation of training data for '*fitting*' the surrogate models.

The choice of surrogate model architecture also plays a key role in determining the method of generation of training data from CFD solvers. For example, for a linear system, a state-space like formulation has the ability to capture the dynamic behavior of the system. The training data must comprise of a rich dataset of input and output data in the expected range of operation. Furthermore, a small dataset would be required as the system is assumed to exhibit linear behavior. With the aforementioned understanding of surrogate modeling, the following sections explore the development of relatively computationally inexpensive surrogate models for transonic aerodynamics, aeroelastic and fluid sloshing systems, based on their observed responses, respectively.

## 4.2 Radial Basis Function Neural Network

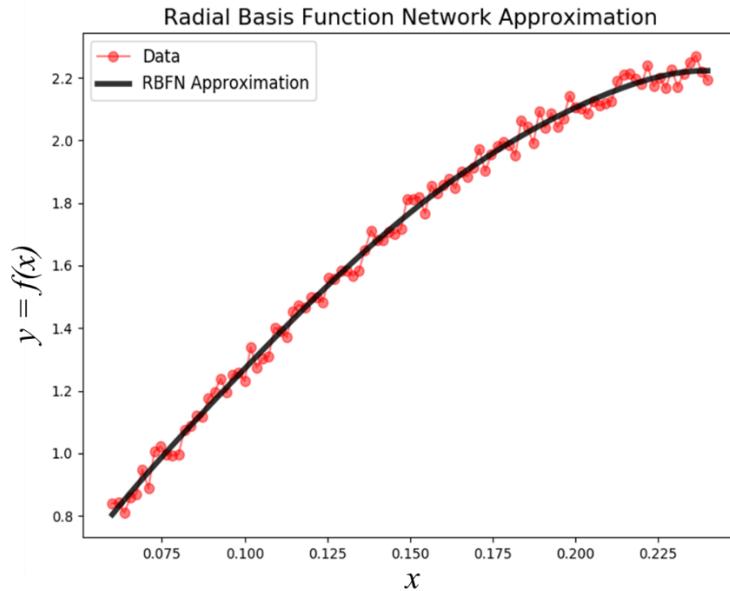
Radial basis function networks, originally published by Lowe and Broomhead [61] in 1988, were derived from the theory of functional approximation and conventional approximation theory [75]. The Radial basis function neural networks (RBF-NNs) belong to the domain of artificial neural networks and are able to approximate any nonlinear function to an arbitrary degree of accuracy with a finite number of neurons. They have proved to be one of the most popular neural networks not only due to the fact that they are universal approximators but also because of their simple 3-layer feedforward architecture. They adopt Gaussian based activation functions for the neurons in the hidden layer. RBF-NNs benefit from having a simple and compact architecture in regards that parameter adjustment is simple. A radial basis function neural network having  $N$  inputs containing  $M$  neurons in its hidden layer producing an output  $\tilde{y}$  is shown in Fig 4.1



**Figure 4.1: A radial basis function neural network with  $N$  inputs and  $M$  neurons in the hidden layer that produces an output  $\tilde{y}$**

RBF-NNs have proved to be a good alternative to multi-layer perceptron (MLP) networks because of their high learning rate [62] and robustness to noise. RBF-NNs have been successfully applied to problems involving system identification [76], time series forecasting [77] and nonlinear controls [78]. The accuracy of prediction of RBF-NN can be estimated by

calculating the relative errors which are an indication of absolute difference in the target value and actual prediction. However, this may not be a good indicator of the *quality* of prediction. This is demonstrated by RBF-NN prediction of a noisy signal shown in Fig. 4.2. The RBF-NN approximation does not pass through many actual data points which leads to a substantial mean error, the *quality* of prediction is good as the data set is clearly noisy. Hence an important question arises: What is a reliable indicator of good quality of prediction? The balance between prediction accuracy and smoothing effect depends on the nature of data. These under-fittings and over-fittings of predictive models can be better understood by understanding the detailed working of RBF-NN.



**Figure 4.2: Functional approximation of noisy data using radial basis function networks**

The workflow of the RBF-NN initiates at the input layer, where the network inputs are distributed to the hidden layer. The hidden layer, containing neurons which receive the input processes the information in a nonlinear manner by evaluating the Euclidian distance between the input and the neuron center and produces the output by computing a weighted average. One single neuron allows only a comparatively simple operation. However, the combination of

many neurons makes this approach powerful. The mathematical formulation for RBF-NN with an output vector consisting of  $N_y$  elements ( $1 \leq i \leq N_y$ ) can be written as

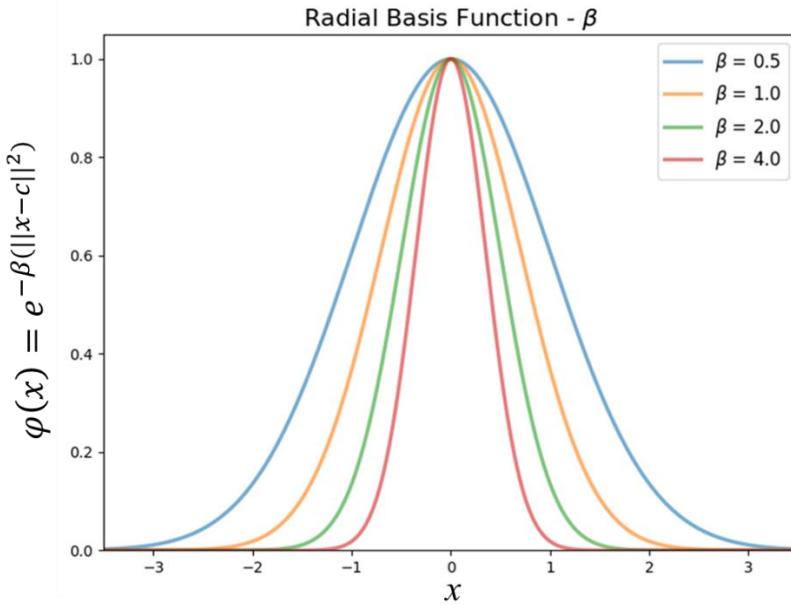
$$\tilde{y}_i = \sum_{j=0}^M w_{ij} \cdot \varphi_j (\| u - c_j \|) \quad \text{with } \varphi_0 = 1 \quad (4.1)$$

where  $\tilde{y}_i$  is the  $i$ -th element of the output vector,  $\mathbf{W}$  a matrix containing linear weights  $w_{ij}$ ,  $\mathbf{u}$  the input vector and  $c_j$  the center vector affiliated to neuron  $j$ . For the  $M$  basis function  $\varphi_j$  in the hidden layer, the typical approach of using the Gaussian RBF in combination with the Euclidean distance norm is inserted:

$$\varphi_j (\| u - c_j \|) = \exp \left( -\frac{\sum_{k=1}^N (u_k - c_{jk})^2}{2\sigma_j^2} \right) \quad (4.2)$$

$$\beta_j = \frac{1}{2\sigma_j^2} \quad (4.3)$$

Due to the characteristics of RBFs, the output of a neuron becomes smaller when the distance between the input and the neuron's center increases. The spread parameter  $\sigma_j$  determines the sphere of influence of neuron  $j$  and, network performance. With larger spreads, the width of the Gaussian RBF increases and therefore covers a greater regime of the input space. This is demonstrated with different  $\beta$  values in Fig. 4.3. Recalling that only the input and output vectors are known for a given training data-set, the centers, weights, and spreads have to be trained to realize an RBF-NN based model.



**Figure 4.3: Different *beta* values depicting differences in the spread of gaussian-type function**

### Training of Radial Basis Function Neural Networks

The RBF-NN training process mainly implies optimizing the basis function parameters, i.e. the centers,  $c_j$ , and spread of the activation function,  $\beta_j$ , in the neurons and the associated weights. The basis function parameters can be chosen via many different methods such as choosing random data points from the training data, or using *K-Means* clustering [79] approach and use the cluster centers as centers of neurons.

The *beta* coefficient in the activation function of RBF-NN controls the width of the Gaussian curve. This must be determined such that it optimizes the mapping accuracy of the activation function and the training data set. Choosing *beta* coefficient depends on the method of choosing the neuron centers. Finally, the weights between the hidden layer and the output layer constitute the third parameter to be optimized. This can be done by using the *Gradient descent* method [80] which trains the weights by comparing the target outputs with the outputs of the RBF neurons. Some of the aforementioned methods are discussed as follows:

- (i) K-means clustering: This is one of the most common approach of assigning centers to RBF neurons. Some of the advantages of using K-means clustering approaches are using a small

number of RBF neurons, where each neuron represents a cluster in the data. Furthermore, having a relatively smaller number of neurons does not affect the complexity, nor the accuracy of prediction for the RBF network [81].

(ii) Gradient Descent: Gradient Descent algorithm, widely used for optimization problems, utilizes first-order derivative calculations to search for local minima of the objective function. The RBF-NN predicted output, represented by  $\tilde{y}$  in Eqn. (4.4a) and Eqn. (4.4b) where  $W$  indicates the weights matrix and  $\Phi$  represents the radial basis function values.

$$\tilde{y} = (\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_M) = \begin{bmatrix} w_{11} & w_{11} & \cdots & w_{1m} \\ w_{21} & w_{21} & \cdots & w_{2m} \\ \cdot & \cdot & \cdots & \cdot \\ w_{I1} & w_{I2} & \cdots & w_{Im} \end{bmatrix} \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_m(x) \end{bmatrix} \quad (4.4a)$$

$$Y = W \cdot \Phi \quad (4.4b)$$

The prediction error  $E$  is proportional to the difference between the network output and the target output, computed as follows,

$$E = \sqrt{\sum_{i=1}^M (y_i^{true} - \tilde{y}_i)^2} \quad (4.5)$$

Based on the errors, the network weights as adjusted according to,

$$w_{ij} = w_{ij} - \eta \frac{\partial E}{\partial w} \quad (4.6)$$

where the parameter  $\eta$  represents the learning rate.

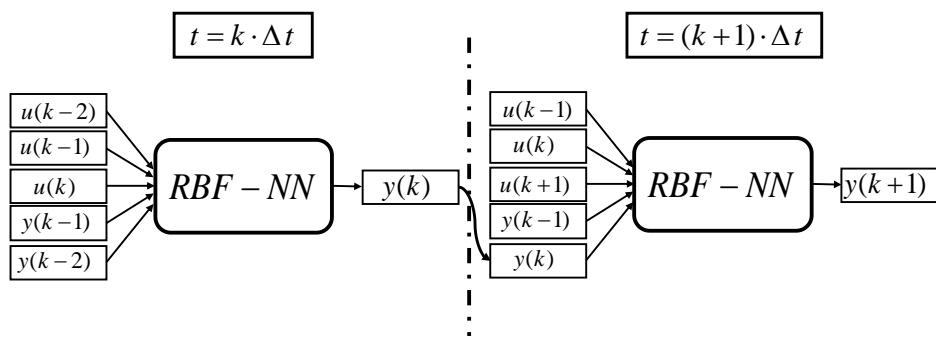
### Implementation of RBF-NN for Sloshing Problem

Sloshing fluid is a transient phenomenon and therefore previous structural inputs, sloshing loads, as well as current structural input, are required for the prediction of the loads for the next time step. As fuel sloshing is a transient phenomenon, previous structural state inputs, sloshing

loads, as well as the current structural state inputs are required for the prediction of the loads for the next time step (the future state). One widely used system identification technique is the Autoregressive technique with Exogenous inputs (ARX) as outlined in Billings [63] which assumes that the known relationship between a finite series of former inputs and previous outputs is sufficient to predict system response to subsequent inputs. Such dynamical systems can be written as

$$y(t) = f \begin{bmatrix} u(t), u(t-1), \dots, u(t-m), \\ y(t-1), \dots, y(t-n) \end{bmatrix} \quad (4.7)$$

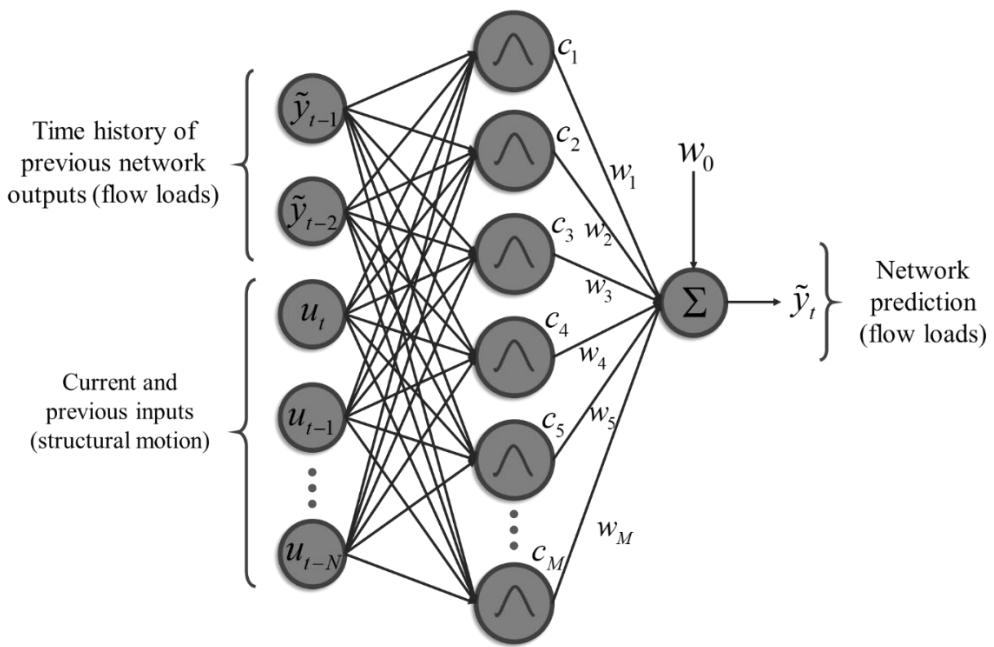
where  $y(k)$ ,  $y(k-1)$  ... are the predicted outputs i.e. sloshing loads, for the current and previous time steps while  $u(k)$ ,  $u(k-1)$ ... represent the current and previous structural states. This process is shown in Fig. 4.4 Here, it is assumed that  $f$  is stationary for the system. Such dynamical systems which make use of sequential data and require some ‘*memory*’ of previous states of the systems can be modeled by ARX-type formulation in which delays must be optimized iteratively and the values of  $n$  and  $m$  are determined by hit and trial method and theoretically as there are no constraints on limits of their values. For the current study, the inputs and output delay-orders are 5 and 3 respectively.



**Figure 4.4: ARX based network with input delay-order of 2 and output delay-order of 1**

The delays are added in the RBF-NN as inputs to account for the ‘*memory*’ effect. Essentially it acts as the information of previous states of the physical system, in this case, a sloshing tank. The methodology shown in Fig. 4.4 is implemented in the RBF-NN architecture as shown in

Fig. 4.5. The predicted outputs are represented by vector  $\tilde{y}$ , which in this case are vertical forces (in lift direction of the airfoil) and moment (in pitching moment direction of the airfoil). These loads are computed by integrating the pressure field at the tank boundary at every time step. These predicted loads are used as inputs for subsequent iterations in time marching with a defined delay. In addition to the time history of previous network outputs, the input vector contains current and previous information of structural motion that the tank undergoes.



**Figure 4.5: Radial basis function neural network with ARX-type architecture containing the time history of previous inputs and outputs of the prediction network**

The objective of this study is to model the sloshing forces as an unknown quantity. The approach to handle them should be the same as that of a stochastic system. The sloshing forces can become highly nonlinear depending upon the motion on containing structure and hence localized pressure prediction may be difficult, but it is the integrated force/moment on the tank that affects the motion of the container. Therefore, a limited CFD -based data is exploited to develop and calibrate the surrogate model which predicts integrated forces and moment on the fuel tank for subsequent structural motions fed in as inputs. The neural-network-based surrogate modeling approach is apt for the given problem. Although this approach requires a

large quantity of training data, the computational cost of a wisely chosen training data set is still very small as compared to the high fidelity model.

The generation of the training data by structural excitation for the development of surrogate model is outlined briefly. Before an input signal is selected, it is important to identify the operating range of the system. Special care must be taken not to excite the dynamics that must not be incorporated in the model. For example, the training data must not include system response near their natural frequencies if the system is not projected be excited by those frequencies during prediction run. For identification of linear systems, it is customary to apply signals consisting of sinusoids of different amplitudes or impulse inputs. However, for nonlinear model structures, it is important that all amplitudes and frequencies are represented. For the current study, the amplitude-modulated pseudo-random binary signal (APRBS), also referred to as *N-samples-constant*, is chosen for forced structural excitation to develop an input-output relation for surrogate modeling. This signal can be generated from the frequently used pseudo-random binary signal (PRBS) by assigning random amplitudes to each plateau level. If  $e(t)$  is a white noise signal with variance  $\sigma_e^2$ , the signal defined by

$$u(t) = e \left( \text{int} \left[ \frac{t-1}{N} \right] + 1 \right) \quad t = 1, 2, \dots \quad (4.8)$$

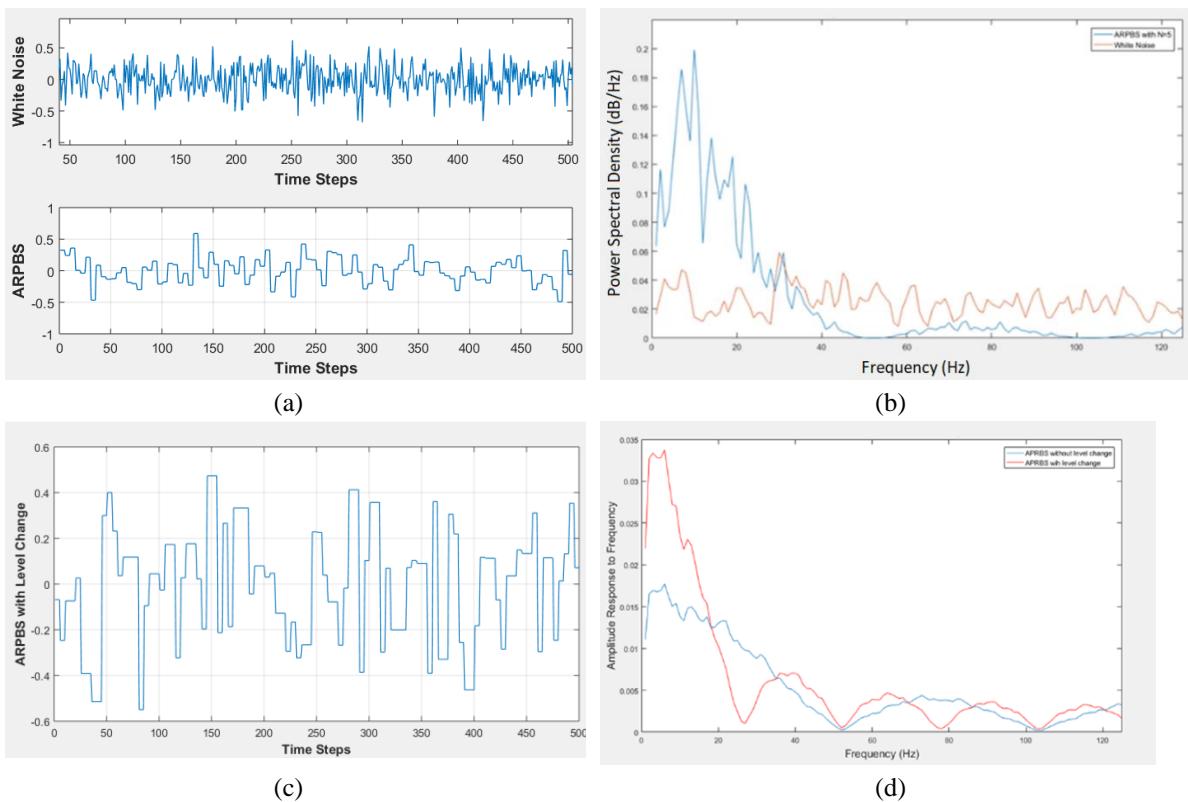
will jump to a new level at each  $N^{th}$  sampling instant (int denotes the integer part) and its covariance function is defined as,

$$R_u(\tau) = \frac{N - \tau}{N} \sigma_e^2 \quad (4.9)$$

This signal is further modified by introducing a level change parameter with probability,  $\alpha = 0.5$ , for deciding when to change level as follows:

$$u(t) = \begin{cases} u(t-1) & \text{with probability } \alpha \\ e(t) & \text{with probability } 1-\alpha \end{cases} \quad (4.10)$$

This modification may be considered as some type of low pass filtering. Normalized white noise signal shown in Fig. 4.6(a) is utilized for constructing ARPBS signal which contains frequencies relevant to the system as shown in Fig. 4.6(b). The APRBS signal is further modulated with level change modification is shown in Fig. 4.6(c). The amplitude response to frequency as shown in Fig. 4.6(d) shows that the level change parameter can be tuned to excite the frequencies of interest.



**Figure 4.6:** (a) White noise signal used to generate APRBS and cut off higher frequencies, (b) Power spectral density comparison of APRBS and white noise signals, (c) APRBS with level change modulation, and (d) Amplitude response to excitation frequencies comparison between APRBS with and without level change parameters

The main advantage of utilizing the APRBS is the large spectrum of frequencies and amplitudes. This property is of paramount importance for the nonlinear system identification task. Furthermore, only a short excitation time-series is needed, limiting the computational

cost. Markov chain-based approach is used for training the dynamic system behavior using finite setoff input-output data samples.

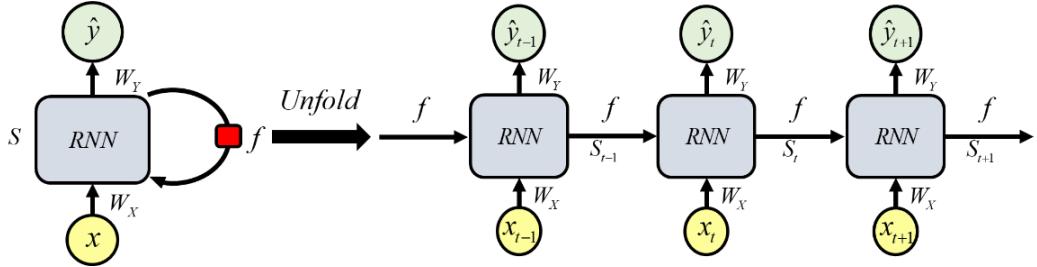
### 4.3 Recurrent Neural Network

Conventional Neural networks offer powerful tools for system identification and data-driven modeling for both supervised and unsupervised learning where the underlying features are not distinctly interpretable. However, despite their advantages, conventional feedforward neural networks have limitations when they deal with sequential data. Although as it was shown in the previous section that sequential data can be concatenated and fed into the network through architectural manipulations, it can be a tedious task that is prone to alterations for different physical systems. This hinders the scalability and reusability of feedforward models such as RBF-NNs and other conventional artificial neural networks (ANNs) type architectures which assume a black-box approach for input-to-output mapping. This shortcoming is overcome by deploying a Recurrent Neural Network (RNN) which not only allows nonlinear modeling of training data but also processes temporal data for predictive modeling. RNNs are designed to process sequences of data across multiple time steps while processing each individual data at a time. This means that RNN learns the dependence of a data point on the preceding sequence of data provided. This property conforms to the problem statement of this research as the unsteady motion of the airfoil and the internal fuel sloshing is influenced not only by the current state of the system but also by its previous states. Hence RNN based surrogates are chosen for both aeroelastic and sloshing models. The details of RNN architecture, model training, hyperparameter tuning, and testing are provided in the Results and Discussions chapter of this thesis.

RNNs are designed natively to allow deep learning networks to process sequences of data as shown in Fig. 4.7. The state,  $S_t$ , is an indicator of the temporal history, i.e. it represents the

effects of all the previous inputs and predicted outputs on the current time step. This is an elegant and compact way of expressing the effect of multiple time steps of processes in a single parameter. A single time step of the input is supplied to the network and the current state is calculated by a nonlinear transformation,  $g$ , which is a function of the present input and the previous state i.e.,

$$S_t = g(x_t, S_{t-1}) \quad (4.11)$$



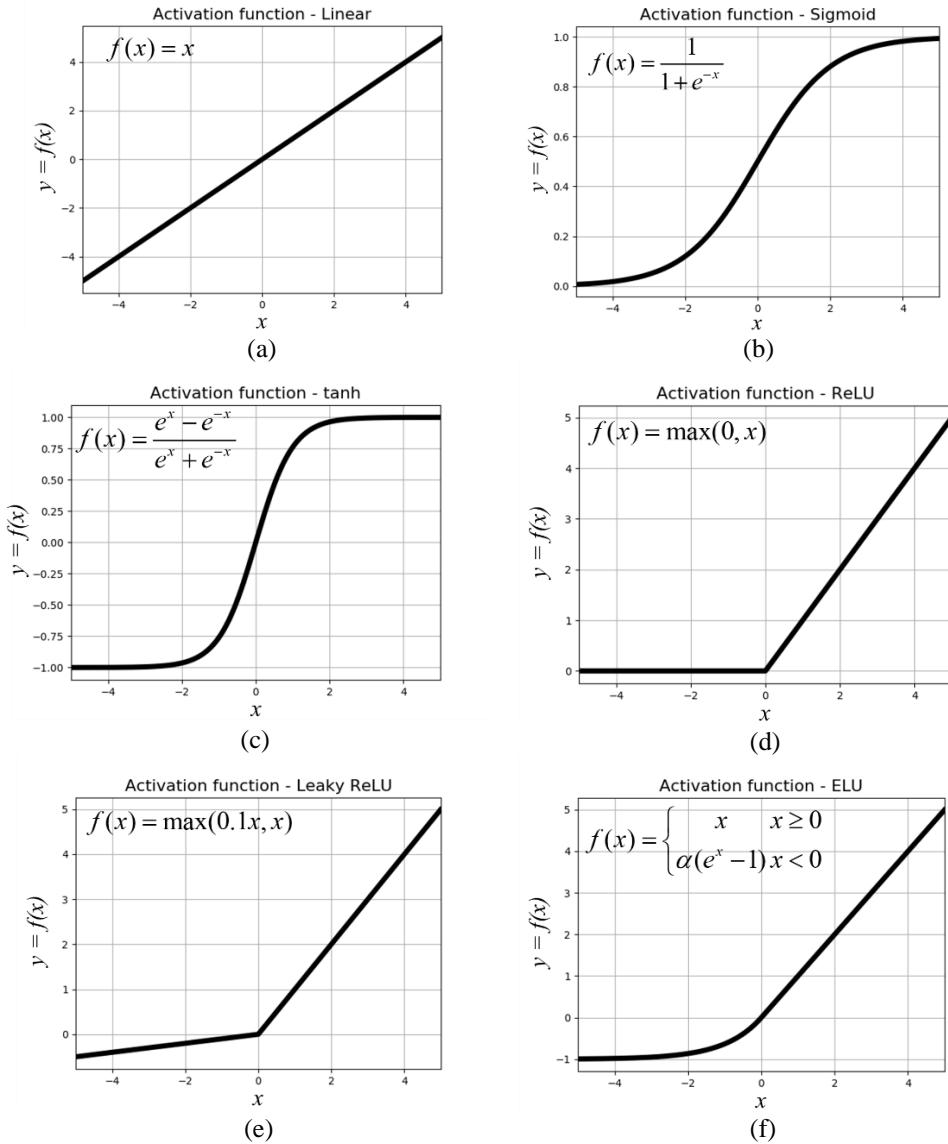
**Figure 4.7: Recurrent neural network cells in abstract and unfolded representation**

The current state,  $S_t$  becomes the previous state for the next step and the process is repeated until the desired number of steps is reached. The final current state is then used to calculate the network output,  $\hat{y}_t$  as follows

$$\hat{y}_t = F(x_t, S_t) = F(x_t, x_{t-1}, S_{t-1}) \quad (4.12)$$

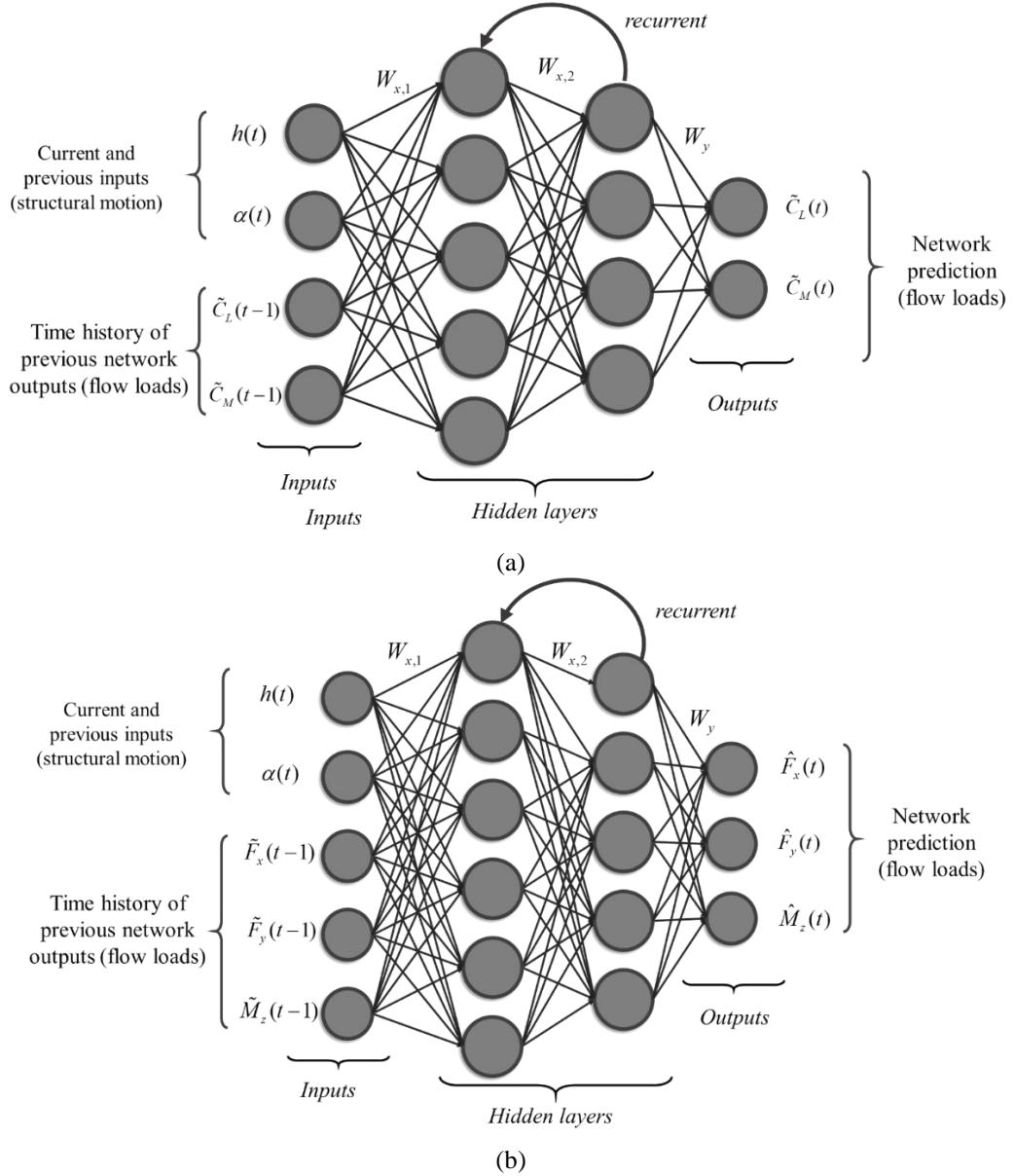
where  $F$  is a nonlinear function that encodes the underlying dynamics of the system learned from the data and can make further predictions. The choice of activation function in the neuron is not only critical for the performance of the model, but also for optimal training. The activation function essentially acts as a transfer function which can be linear or nonlinear. Some common activation functions used in neural networks are shown in Fig. 4.8 (a) – (f). The linear activation function is avoided in neural networks in general not only due to its lack of capacity to model nonlinearities, but also because it maps the output unbounded from negative infinity to positive infinity. Functions such as *sigmoid* and *tanh* shown, respectively in Fig. 4.8 (b) and

Fig. 4.8 (c), provide efficient alternatives that can capture system nonlinearities as well as keep its outputs bound between  $[0,1]$  and  $[-1,1]$ , respectively. Although the squashing property of *sigmoid* function to a value between 0 and 1 is favorable, the function quickly becomes passive to values much smaller than or larger than 0, saturating the neuron's activation to 0 and 1, respectively. The gradients in regions away from 0 are very small which are not favorable for efficient training of network as low gradients render saturation in learning. Furthermore, *sigmoid* function outputs are not zero-centered which are not favorable in case of deep networks. The outputs of neurons are fed as inputs to the next layer, which are always positive in case of *sigmoid* function. This problem can be mitigated by using *tanh* as activation function which is zero-centered, which in recent times has almost replaced *sigmoid* function as the activation function in neural networks. However, *tanh* also suffers from saturated gradients and hinders effective learning of weights during network training. Another family of activation functions based on semi-linear units has gained popularity in the machine learning community that mitigates some of the problems associated with *sigmoid* and *tanh* functions. *Rectified linear unit* (ReLU), *leaky ReLU* and *exponential linear unit* (ELU) are shown in Fig. 4.8 (d), (e) and (f) respectively. These functions have found to greatly accelerate the convergence of stochastic gradient descent as compared to *sigmoid* or *tanh* activations due to their linear, non-saturating form. They are less expensive operations and have easy implementation. The saturation tendency of *ReLU* is overcome by using *leaky ReLU* and *ELU*, which produce non-zero outputs for negative inputs. The activation function used in the hidden neurons for this study is *ReLU*, defined by  $f(x) = \max(0, x)$ , and outlined in Ramachandran et al. [82]. The linear unit overcomes the problem of vanishing gradient [65] because of its partial linear nature and overall it is a good candidate for training networks with more than one hidden layer. It also performs a nonlinear operation on the input and state vectors, and its combined output is reflected in the encoding function  $F$ .



**Figure 4.8: Activation functions (a) linear, (b) sigmoid, (c) tanh, (d) ReLU, (e) leaky ReLU, and (f) ELU**

The inputs and outputs of the sequential data through the hidden units in an RNN architecture at different time steps are considered as if they are discrete inputs and outputs of different neurons with the same weights and biases. This is evident from Fig. 4.6 where the same weight,  $W_x$ , is used for all the sequential outputs of the representative RNN cell facilitating sequential learning. A detailed representation of the densely connected multi-layer (deep) recurrent neural network in the context of the problem of interest is shown in Fig. 4.9(a) for the overall aero-structural system and in Fig. 4.9(b) for the sloshing loads in the fuel tank.



**Figure 4.9: Densely connected deep recurrent neural network in multiple-input multiple-output configuration for (a) the overall aeroelastic model, and (b) sloshing loads in the fuel tank model**

The computational nodes are connected through weights that are collectively represented by  $W_{x,1}$  and  $W_{x,2}$  for the first and second hidden layers, respectively, in Fig 4.9 (a) and Fig. 4.9 (b). The weights are initialized with random values and then adapted during the training process to improve network performance. For the aeroelastic prediction, the inputs are given in the form of temporal sequences of airfoil plunge and pitch displacements i.e.  $h$  and  $\alpha$  and also the network outputs, i.e. lift coefficient,  $C_L$  and pitching moment coefficient  $C_M$  at the previous instant. The sequential input data at the present time step, represented by  $h(t)$  and  $\alpha(t)$  and load

coefficients sequences offset by one time step represented by  $C_L(t-1)$  and  $C_M(t-1)$  in Fig 4.8 (a) and the same notation is followed for sloshing loads shown in Fig 4.8 (b). These inputs are passed through the activation functions in the neurons in each hidden layer and the outputs obtained are the lift and moment coefficients at the present time step. It is important to note that for all temporal inputs in a batch of training data, the weights of the neurons remain the same.

The network is *trained* by minimizing *loss functions* which are essentially the mean squared error (*mse*) of the difference between the output predicted by the network  $\hat{y}$  and the target value of output  $y$  i.e.,

$$Loss, \quad \hat{L} = mse(y_{true} - \hat{y}) = \frac{1}{N_{batch}} \sqrt{\sum_{n=1}^{N_{batch}} (y_{true,n} - \hat{y}_n)^2} \quad (4.13)$$

For the aeroelastic and sloshing problems, the loss function computes the squared differences between the network projected outputs and expected output in a *MIMO* (multiple-input multiple-outputs), respectively, as follows.

$$\hat{L} = mse(y_{true} - \hat{y}) = \frac{1}{N_{batch}} \sqrt{\sum_{n=1}^{N_{batch}} [(C_{L,n} - \hat{C}_{L,n})^2 + (C_{M,n} - \hat{C}_{M,n})^2]} \quad (4.14)$$

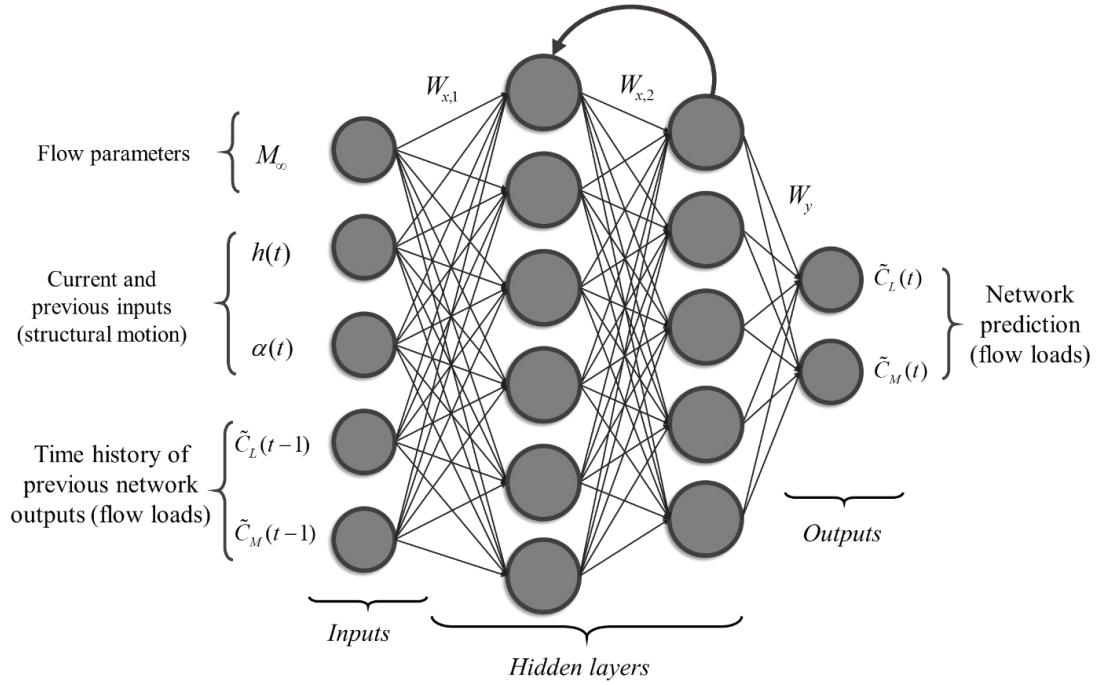
$$\hat{L} = mse(y_{true} - \hat{y}) = \frac{1}{N_{batch}} \sqrt{\sum_{n=1}^{N_{batch}} [(F_{x,n} - \hat{F}_{x,n})^2 + (F_{y,n} - \hat{F}_{y,n})^2 + (M_{z,n} - \hat{M}_{z,n})^2]} \quad (4.15)$$

The model is trained for temporal sequences of inputs using the *backpropagation* algorithm [83] by minimizing the losses using the *Adam Optimization* algorithm [84]. The network is trained using data from the desired physical systems. Once the network weights and biases are established from the training, these parameters are stored and used subsequently to form a surrogate predictive model for various arbitrary future test data that are not present in the initial training data set. In this way, the computational cost for generating CFD data for training the

network itself is a one-time process and need not be repeated often. The reliability of the prediction of arbitrary input data can be assessed by comparing it with CFD computations. In this work, the open-source *Tensorflow* [85] libraries (version 1.10) are used to implement and train the RNN architecture. The training and testing data obtained from high-fidelity CFD simulations are imported and pre-processed for facilitating tensor manipulations in *Tensorflow*. The hyperparameters of the RNN which include the number of epochs, training batch length, and number of neurons in each hidden layer are optimized based on the mean squared error of the predicted outputs for both networks. A more detailed study of hyperparameter tuning for the surrogate models is provided in the Results and Discussions chapter of this thesis.

The surrogate model developed for aeroelastic model shown in Fig. 4.9 (a) is designed to predict the lift coefficient and moment coefficient of the airfoil upon motion of the airfoil in terms of plunging and pitching. This model is extended for 3D flowfields to study the effect of sloshing fuel on aeroelastic motion of a transonic wing. Another feature is added to the prediction machinery in the form of an external flowfield. In the transonic flow regime, the shock strength and location on the wing are dependent on the far-field Mach number, which in turn affects the aerodynamic and aeroelastic behavior. The RNN surrogate is modified to incorporate this feature as shown in Fig. 4.10. This model is trained using aerodynamic data of NASA BSCW immersed in flowfields corresponding to Mach numbers between 0.70 and 0.80 and tested in the same range for intermediate Mach numbers.

The inputs to RNN surrogates are in the form of sequences. Hence, the flow parameter  $M_\infty$ , is also cast in a sequence containing the Mach number values throughout. The surrogate model for sloshing in a 3D remains the same for the 2-DOF wing-tank configuration with external flow parameterization since the sloshing fluid never directly interacts with the external flow-field. However, the training data for sloshing flows must be corresponding to all amplitudes and frequencies experienced by the wing in the given Mach number range.



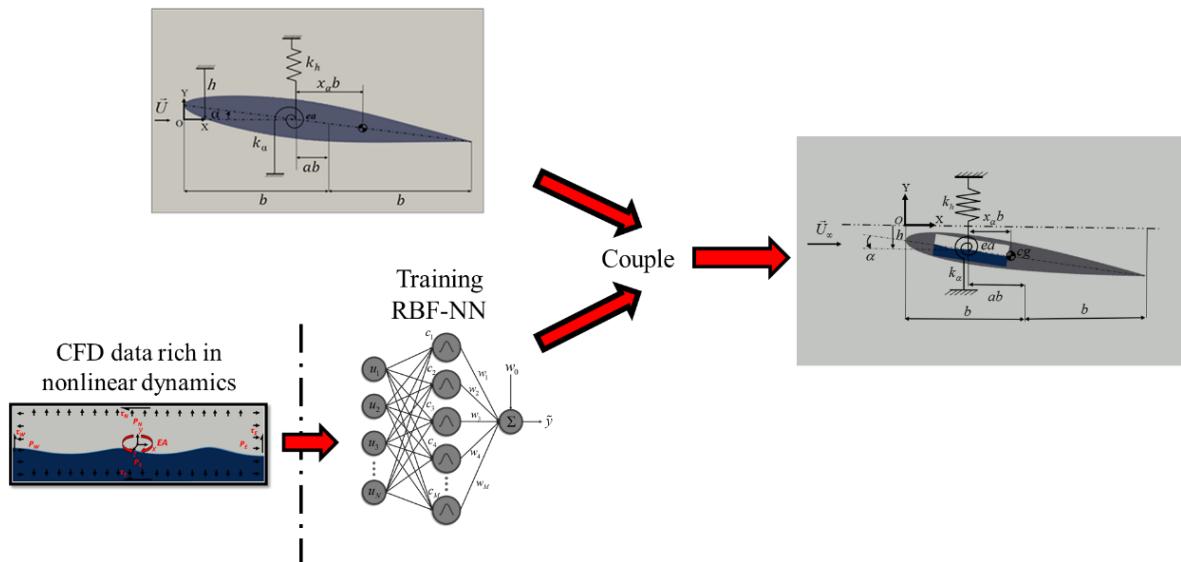
**Figure 4.10: Densely connected deep recurrent neural network in multiple-input multiple-output configuration for the aeroelastic model with parameterized flow conditions**

#### 4.4 Integrated Frameworks

A total of four machine learning based surrogate models have been developed thus far: an RBF-NN and RNN based surrogate models for sloshing and two RNN based surrogates for external flow over an airfoil and a wing in transonic flow regime. These models are trained using data generated from high fidelity CFD solvers and tested against a validation CFD dataset. Once the models prove to be efficient and accurate for standalone applications, they are integrated into a numerical platform to mimic the coupled aeroelastic-fuel sloshing system. Three such frameworks are developed in this research to study (i) flutter boundary of NACA64A010 airfoil with and without a partially filled fuel tank using RBF-NN model as sloshing surrogate and high-fidelity CFD for external flow solutions, (ii) free motion characteristics of NACA0012 airfoil with and without a partially filled fuel tank using RNN based surrogate models for both external aerodynamics and internal fluid sloshing, and (iii) free motion characteristics of NASA

BSCW wing with and without sloshing fuel tank using RNN surrogate with external flow parametrization and RNN surrogate for sloshing in a 3D tank.

The first computational framework involves the coupling of 2-DOF plunge and pitch NACA64A010 airfoil with an embedded fuel tank simulated by RNF-NN based surrogate model as shown in Fig. 4.11. The training data for the RBF-NN model is generated by exciting the partially filled fuel tank with plunging and pitching motion of relevant amplitudes and frequencies. The resulting fluid pressures are integrated along the tank walls at every time step of the CFD solution to compute effective forces and moments corresponding to aerodynamic lift and moment in the aeroelastic equation of motion as detailed in the previous chapters. This forms the training dataset and it is considered as ground truth for all future references in this study.



**Figure 4.11: Computational framework with radial basis function neural network based sloshing surrogate for studying effects of sloshing on NACA64A010 flutter boundary**

Training data in the form of a time-series is split into *train* and *test* datasets with 90% of data utilized for training and 10% for testing the trained model. The training process begins with establishing the number of inputs, number of neurons in the hidden layer and number of outputs. Since the size of the input vector depends upon the *signal delay* size, i.e. number of

elements in the temporal sequence, the RBF-NN's performance is optimized with a few trial runs. Once trained, the model is tested for accuracy in prediction with the *test* dataset. If the prediction error of the test dataset is satisfactory, the model is saved and can be used for multiple runs.

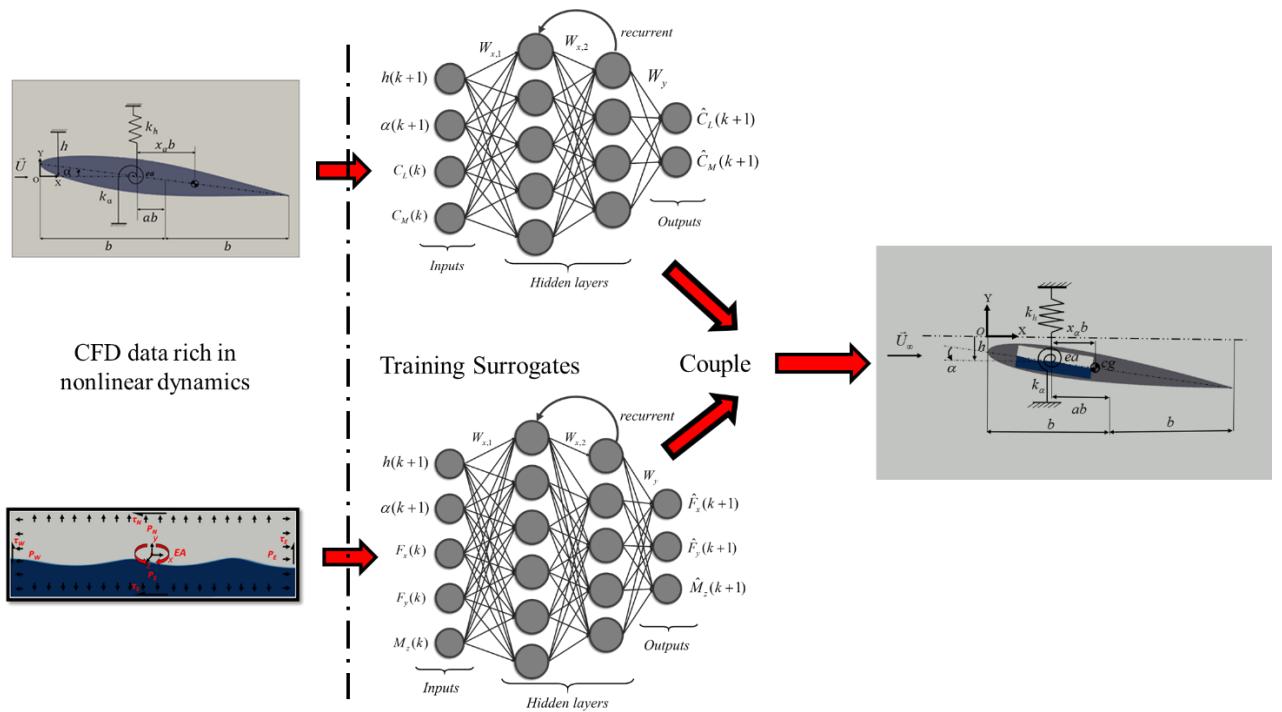
The external compressible flow solver *SU2* requires a problem configuration file and a mesh file for running aeroelastic cases. Computational domain for external flow consists of NACA64A010 airfoil immersed in a circular far-field which has a radius of 100 chord lengths centered at the tip of the airfoil at  $0^\circ$  angle of attack. The domain consists of unstructured mesh that has been tested for grid convergence and validated with McMullen's [86] numerical studies for both steady and unsteady solutions. The free motion of the airfoil is initiated by first computing the steady flowfield, followed by a few cycles of forced pitching with a small amplitude to generate a small perturbation in the flowfield and then setting the airfoil free to move governed by Eqn. 4.16 which was derived in the previous chapter.

$$\frac{m_{tot}}{m} \begin{bmatrix} 1 & x_\alpha \\ x_\alpha & r_\alpha^2 \end{bmatrix} \begin{Bmatrix} \ddot{h} \\ \ddot{\alpha} \end{Bmatrix} + \begin{bmatrix} \left(\frac{\omega_h}{\omega_\alpha}\right)^2 & 0 \\ 0 & r_\alpha^2 \end{bmatrix} \begin{Bmatrix} \bar{h} \\ \alpha \end{Bmatrix} = \frac{V^{*2}}{\pi} \begin{Bmatrix} -C_L \\ 2C_M \end{Bmatrix} + \frac{r^3}{mb\omega_\alpha^2} \begin{Bmatrix} F_{Y,ea} \\ 2rM_{OZ,ea} \end{Bmatrix} \quad (4.16)$$

The sloshing loads that appear on the right-hand side of the equation are predicted by the RBF-NN surrogate while the rest of the equation is solved as usual by *SU2* aeroelastic solver. The loads due to sloshing are added to the equation at the nondimensional structural time step given by the product of pitching frequency and physical time. This framework is utilized for computation of the flutter boundary by subsequently increasing the flutter speed index for each Mach number until a divergent response is achieved. Few cycles of free structural motion are observed to accurately determine the nature of response i.e., damped, neutral or flutter. Since all the computations are done in time-domain, the framework is still expensive due to high-fidelity computations of the external flowfield using CFD and hence the development of the

next computational framework that is completely data-driven and does not involve CFD runs to time marching solutions.

Recurrent neural network based surrogates are utilized to develop a computational framework to predict the aeroelastic motion of NACA0012 airfoil embedded with a partially filled fuel tank. The RNN to predict flow loads due to external flowfield, namely lift coefficient and moment coefficient is trained by feeding in sequences of motion data in form of plunging and pitching motion, as well as the temporal sequences of lift coefficients and moment coefficients corresponding to that motion. Similarly, the RNN to predict integrated pressure loads due to sloshing is trained by feeding in the structural motion sequence to the fuel tank along with corresponding loads. The working of this computational framework is represented by Fig. 4.12.



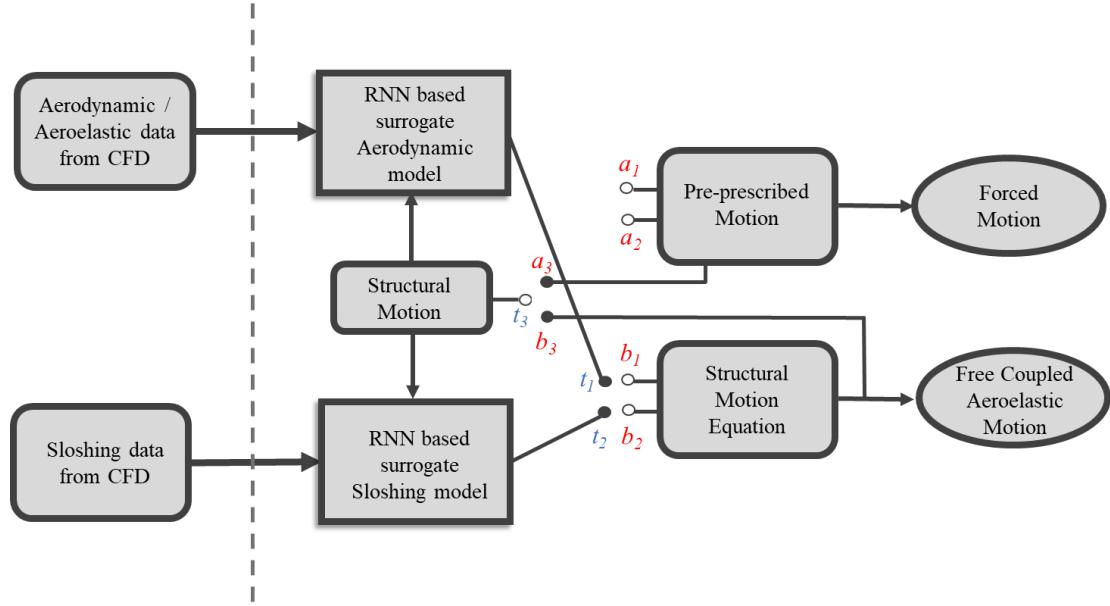
**Figure 4.12: Computational framework with recurrent neural network based surrogates for studying effects of sloshing on aeroelastic airfoil in transonic flow regime**

The surrogate model for external compressible flow is trained for both forced motion as well as free motion of the airfoil. The training data is generated by high fidelity CFD using *SU2* codes. The computational domain includes NACA0012 airfoil immersed in a circular flowfield

with a radius of 100 chord lengths centered at the leading edge of the airfoil with a  $0^\circ$  angle of attack. Grid convergence studies have been done for the unstructured mesh generated in the computational domain and numerical results are validated with McMullen's [86] data for both steady and unsteady flows. The structural displacements for forced motion of the airfoil is known beforehand to the user, which can be fed as inputs to aero-surrogate to produce aerodynamic loads as outputs. Hence, this framework can be used as a standalone model for aerodynamic load prediction. Similarly, the sloshing surrogate model is also trained and tested to predict the corresponding loads at each structural time step for pre-prescribed motion, as well as free motion, both corresponding to the airfoil motion. When the airfoil is set free, the aerodynamic and sloshing load interaction with the structural motion is governed by Eqn. 4.16. The displacements computed from the solution of Eqn. 4.16 are fed back into the surrogate models as new inputs and the prediction marches on in time to yield the plunge and pitch displacements. These functionalities are carefully and neatly implemented in the framework and are explained with the help of Fig. 4.12.

The motion of aero-structural system must be selected to either forced predefined motion or free aeroelastic motion as shown in Fig. 4.13 by connecting surrogate toggles  $t_1$  and  $t_2$  to toggles  $a_1$  and  $a_2$  or  $b_1$  and  $b_2$ , respectively. The motion parameters,  $h$  and  $\alpha$ , are supplied to the surrogates through  $t_3$  via  $a_3$  or  $b_3$ , depending on the choice of model. This determines the usage of either forced motion surrogates or free motion surrogates, as well as, the source of aero-structural motion parameters, i.e.  $h$  and  $\alpha$ . For forced motion,  $h$  and  $\alpha$  are predetermined and the surrogate models essentially predict the aerodynamic and sloshing loads respectively. For free motion, the structural motion parameters are computed by solving the aero-structural motion Eqn. 4.16 using the load predicted by the surrogate models. The structural motion sequences combined with loads predicted by the models are fed back in the surrogate model as inputs to produce subsequent predictions and to march the prediction in time. This machine

learning based computational framework can be made more robust and richer for parametric variation with more training data over different flow conditions and motion amplitudes and frequencies.

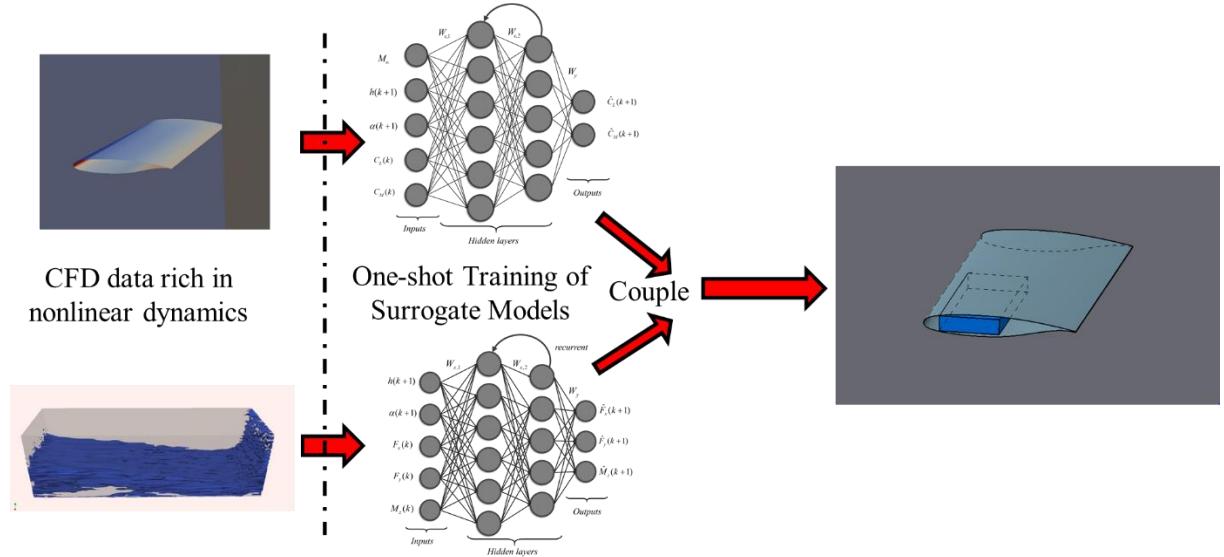


**Figure 4.13: Flowchart showing the coupling of aero-structural model for both forced and free aeroelastic motions**

The surrogate models for external flow and internal sloshing are first tested individually for both motions: forced motion with constant amplitude, and free aeroelastic motion with changing amplitude. The details of RNN architecture, model training, hyperparameter tuning, and testing are provided in the Results and Discussions chapter of this thesis.

The RNN surrogates have been tested for 2-DOF NACA0012 airfoil with an embedded tank. However, the framework is flexible and hence expandable to any 2-DOF system with a sloshing fuel tank as long as it is governed by Eqn. 4.16. This property of the framework is exploited for extending to the current problem to a 3D flow-field. NASA Benchmark Supercritical Wing (BSCW) is chosen for this study as it is a 2-DOF plunge and pitch wing designed for transonic flow conditions. The wing geometry and some experimental results for steady and unsteady flows are readily available in the opensource to validate the study. The RNN based framework

is improved by adding another flow parameterization feature to the external flow RNN surrogate as previously discussed and shown in Fig. 4.9. The working of this computational framework is represented by Fig. 4.14.



**Figure 4.14: Computational framework with recurrent neural network based surrogates for studying effects of sloshing on aeroelastic motion of BSCW in transonic flow regime**

The aerodynamic response in terms of lift and moment coefficient of BSCW is computed using CFD for forced motion with freestream Mach number varying from 0.70 to 0.80. This data is used for training of RNN based surrogate model. Similarly, the sloshing loads' response of a partially filled 3D fuel tank is generated using multiphase flow solver for tank motion excited with amplitudes and frequencies of the wing motion and used for training the sloshing surrogate. The free motion of the wing is computed by solving the aeroelastic structural equation with aerodynamic and sloshing loads predicted from surrogate models. Prediction results show good agreement for flow conditions used in training data, as well as for intermediate flow conditions that the model had not encountered during training. The details of network architecture, training approach, testing results, and prediction capability are provided in the Results and Discussions chapter of this thesis.

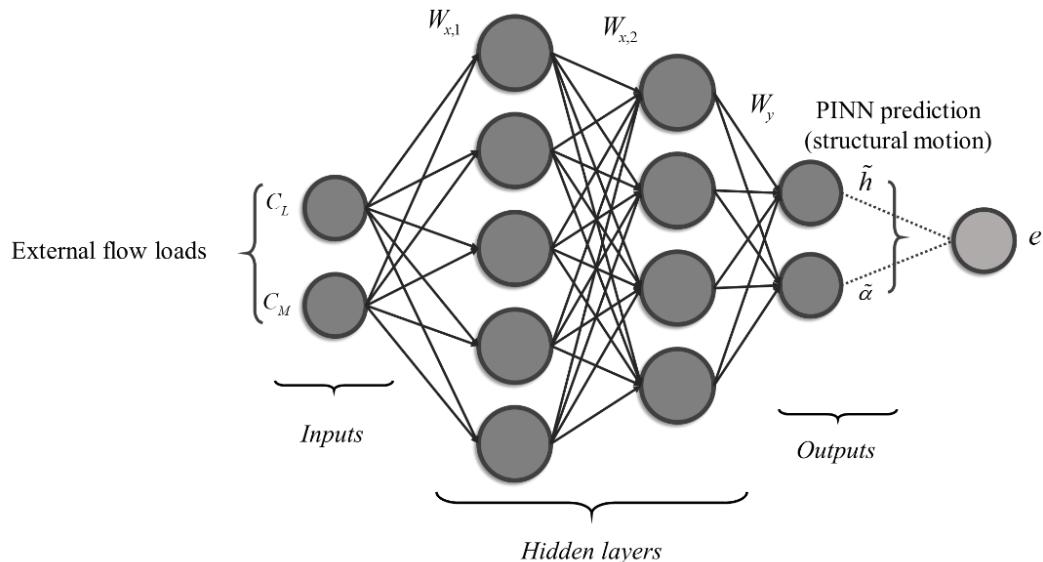
The computational savings gained from the RBF-NN and RNN based computational framework for 2D and 3D flowfields are noteworthy and justifies the development of machine learning surrogates. Several orders ( $\sim 10\text{-}100$ ) of computational savings are achieved using machine learning surrogates as compared to high fidelity simulations. The nature of structural motions and corresponding loads encountered in this problem are mostly harmonic in nature. However, the framework developed is robust to highly nonlinear datasets as well as parametric variations as long as appropriate samples of data are utilized for training.

#### 4.5 Physics-Informed Neural Network

The machine learning surrogates and computational frameworks discussed in this chapter thus far utilizes high-fidelity computations for the generation of training data for machine learning surrogates for assessing the effects of fuel sloshing on aeroelastic characteristics of airfoils and wings. These frameworks use integrated pressure loads to compute the free aeroelastic motion of 2-DOF system using the structural equation of motion. The developed surrogate models in the present work use CFD data to train the models, essentially the weights and biases of the neural networks. Although the performance of these data-driven models is excellent, it is dependent on the training data used for the learning. Hence, the performance of these models outside of the learning parameters domain cannot be relied upon. This shortcoming of these data-driven model is alleviated by using Physics-Informed Neural Networks (PINN) [66], that incorporate the governing equations of the underlying physical systems in the learning process. This physics-based surrogate neural network model is applied to the structural solver of the aeroelastic motion without sloshing given in Eqn. 4.17 as a proof of concept.

$$\begin{bmatrix} 1 & x_\alpha \\ x_\alpha & r_\alpha^2 \end{bmatrix} \begin{Bmatrix} \ddot{\bar{h}} \\ \ddot{\alpha} \end{Bmatrix} + \begin{bmatrix} \left(\frac{\omega_h}{\omega_\alpha}\right)^2 & 0 \\ 0 & r_\alpha^2 \end{bmatrix} \begin{Bmatrix} \bar{h} \\ \alpha \end{Bmatrix} = \frac{V^{*2}}{\pi} \begin{Bmatrix} -C_L \\ 2C_M \end{Bmatrix} \quad (4.17)$$

PINN effectively ‘*teaches*’ the dynamics of aeroelastic system to a neural network. Essentially the PINN learns the effect of mass and spring matrix in the equation of motion and mimics the structural equation for solution of displacement parameters when fed with the forcing term on the right-hand side of the equation. The training methodology and loss terms of the conventional neural network is modified in this approach. The loss function is modified by adding residual terms to benefit from the governing equation when training the weights. Hence a scarce set of training data can be used since now the network not only depends on the dataset but also the governing residuals. A detailed representation of the densely connected multi-layer physics-informed neural network for aeroelastic equation of motion is shown in Fig. 4.15.



**Figure 4.15: Physics-informed neural network for 2-DOF aeroelastic equation**

The residuals are calculated during the training of PINN and are represented by  $e$  in Fig. 4.15. This residual is computed from the governing equation of the system and added to the *mean squared error* (mse) to construct the loss function. This composite loss function is minimized during the training of the neural network using the backpropagation method. The mathematical equations for residual and loss function are given as follows:

$$e = e_1^2 + e_2^2 \quad (4.18a)$$

$$e_1 = \left( \ddot{h} + x_\alpha \ddot{\alpha} \right) + \left( \frac{\omega_h}{\omega_\alpha} \right)^2 h - C_L \quad (4.18b)$$

$$e_2 = \left( x_\alpha \ddot{h} + r_\alpha^2 \ddot{\alpha} \right) + \alpha + 2C_M \quad (4.18c)$$

$$Loss = mse + e \quad (4.18d)$$

The modified loss function based on the equation of motion is implemented in the neural network. The model is trained by free aeroelastic motion of NACA0012 immersed in lower transonic flow. The training details and results of PINN are detailed in the Results and Discussions chapter of this thesis.

## Chapter 5

### Results and Discussions

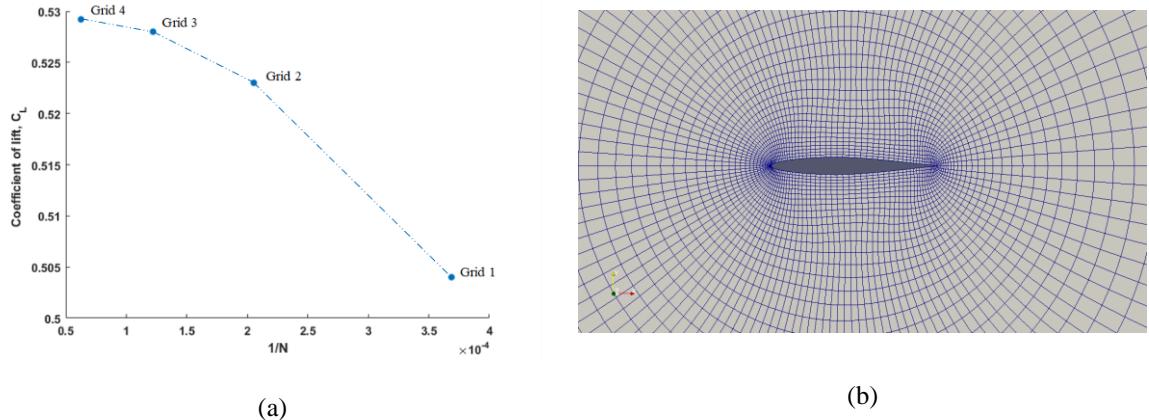
This chapter is divided into different studies undertaken in this thesis. It begins with the assessment of fuel sloshing effects on the flutter boundary of NACA64A010 airfoil in the transonic flow range. This study utilizes high-fidelity CFD solutions for external compressible flow. The sloshing is modeled by three different approaches: high-fidelity CFD, RBF-NN based surrogate model and EMS model. First, the RBF-NN based surrogate model for sloshing is used and evaluated for flutter boundary computations, followed by the EMS model. The next part of this chapter details the aeroelastic study of NACA0012 airfoil with an embedded tank. Machine learning based surrogate models are developed for both aerodynamics and sloshing. Surrogate model results of diverging aeroelastic motion are compared with those obtained from high-fidelity CFD. The aerodynamics surrogate model is further developed to incorporate parametric flow variation and implemented in a 3-D aeroelastic setting. The prediction accuracy and computational benefits of the developed framework are discussed. Finally, the chapter is closed with some discussion on Physics-Informed Neural Network, which is developed to replace the aeroelastic structural equation with a data-driven model that attempts to *learn* the physics behind the data. This work acts as the predecessor to the future works which intends to extend the applications of PINN for complete aeroelastic analysis.

#### **5.1 Flutter Boundary of NACA64A010 with RBF-NN Based Sloshing Model**

In order to demonstrate the feasibility of the proposed RBF-NN based surrogate model and EMS model for sloshing, the NACA64A010 wing section with an embedded rectangular fuel tank is considered. External flow is governed by Euler equations cast in ALE formulation to accommodate structural motion. The fuel tank is considered half-filled for this study.

## A. Grid Dependence Studies for the Inviscid Aeroelastic Solver within SU2

A NACA64A010 wing section is immersed in an external inviscid flow governed by Euler equations. The flow is transonic with Mach number 0.70 and the wing section is fixed at a 3° angle of attack. The freestream temperature and pressure are 288.15 K and 101325.0 Pa, respectively. The variation of coefficient of lift  $C_L$  is for 4 grids with the inverse of the mesh size  $N$  corresponding to Grid 1, Grid 2, Grid 3 and Grid 4 consisting of about 2300, 4000, 8100 and 16000 elements respectively as shown in Fig. 5.1(a). Fig. 5.1(b) shows mesh structure in the vicinity of the wing section for Grid 3.

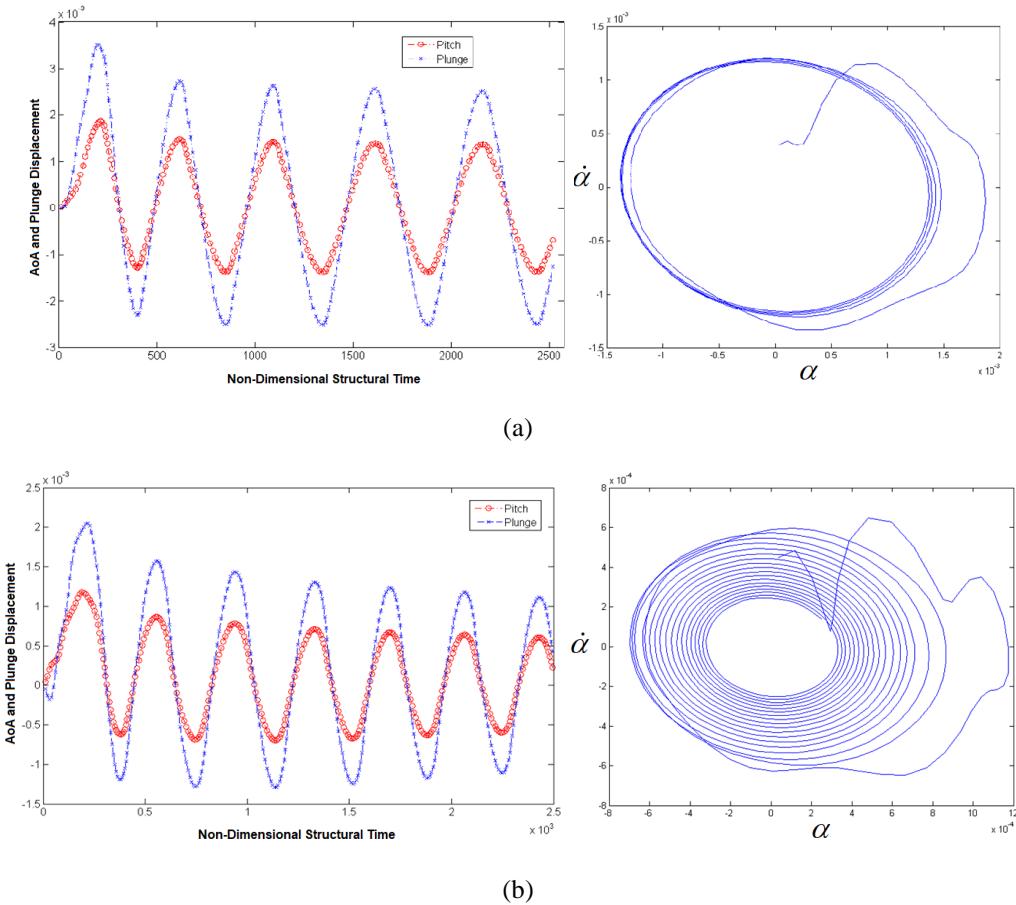


**Figure 5.1: (a) CL vs. 1/N Variation, and (b) Grid 3 mesh in the vicinity of the wing section**

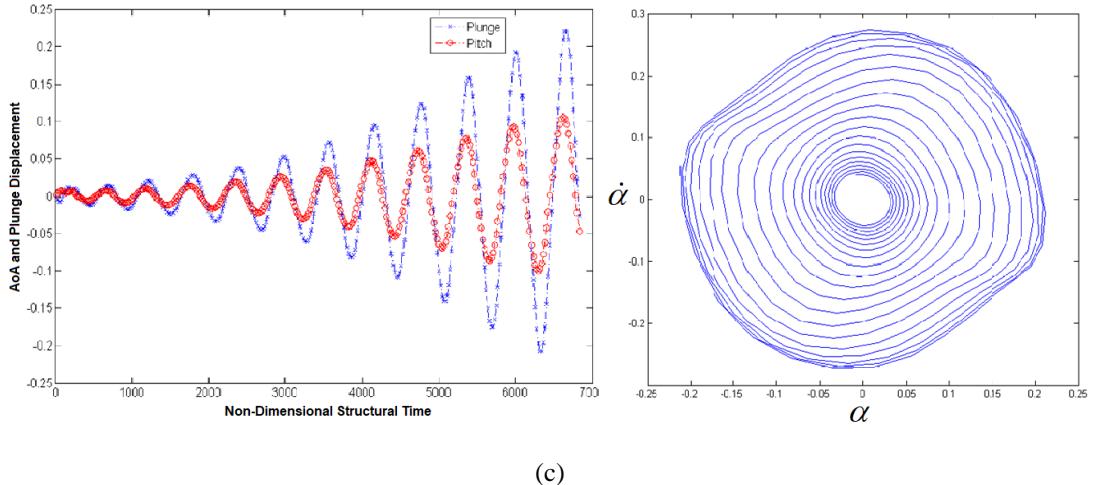
The grid convergence index is estimated as outlined in Roache [87]. The coefficient of lift is tracked to evaluate the order of convergence  $p = \left( \ln \left[ (f_3 - f_2)(f_2 - f_1)^{-1} \right] \right) (\ln(r))^{-1}$  with a refinement ratio,  $r = 1.5$  for the grids used in the current study, i.e. grids 2, 3 and 4 and where  $f_1$ ,  $f_2$ , and  $f_3$  are the values of CL computed on the finest, medium, and coarsest mesh respectively. For Grid 3 with nearly 8100 elements, the grid convergence index  $GCI = F_s |(f_2 f_1 - 1)| r^{-p} - 1$  works out to 0.0017 for a factor of safety FS of 1.25. This lies in the acceptable range of uncertainty and hence medium grid is used to compute the aeroelastic behavior.

## B. Validation of Compressible Flow Solver SU2 for the Inviscid Flutter Analysis of NACA64A010

The compressible flow solver *SU2* is validated against numerical results presented by Alonso [17] for the case of the flutter of the NACA64A010 wing section which is forced sinusoidally in pitch for 2 complete cycles at a frequency close to the natural pitching frequency of the structure and then released for free motion. The three typical aeroelastic responses i.e., damping response, neutrally stable response and diverging response at different Mach number values and different values of speed index and  $\Delta\alpha = 1^\circ$ ,  $\omega_f = 100 \text{ rad/sec}$  have been replicated in the current study as shown in Fig. 5.2 (a) – (c).



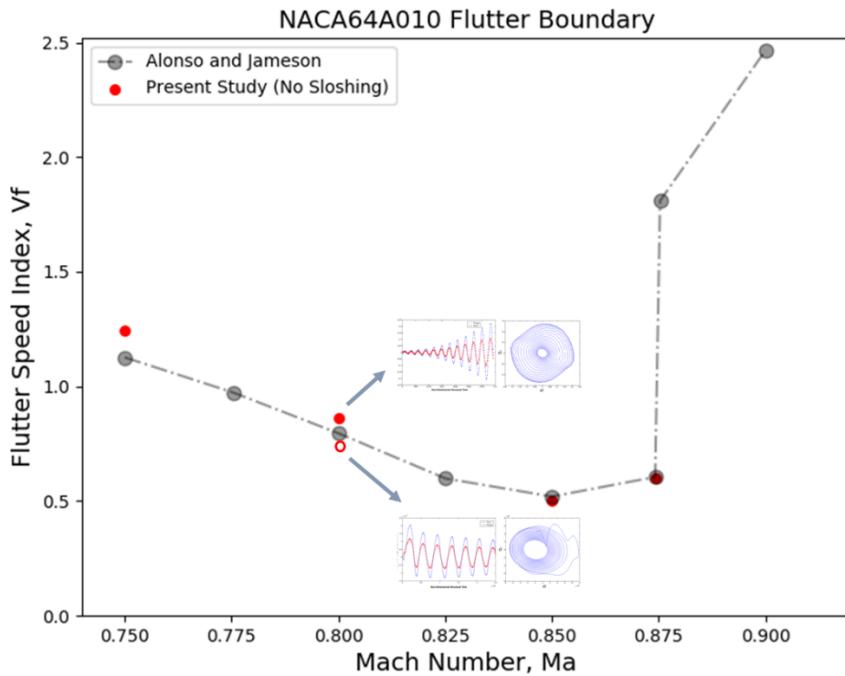
**Figure 5.2: (a) Neutral response  $M_\infty = 0.825$ ,  $V_f = 0.612$ , (b) Damped response  $M_\infty = 0.85$ ,  $V_f = 0.439$ , (Continued)**



(c)

**Figure 5.2: (c) Diverging (flutter) response  $M_\infty = 0.875$ ,  $V_f = 1.420$**

Figure 5.3 compares the computed flutter boundary at four different Mach numbers using the time-domain solver *SU2*. At each Mach number, the flutter speed index is fixed and the wing section is forcibly pitched for cycles with  $1^\circ$  amplitude at its natural frequency, i.e.  $\omega_\alpha = 100 \text{ rad/sec}$ . This generates an initial flowfield for aeroelastic computations and the wing section is set free to move for five cycles. The resulting motion is checked for damped, neutral or diverging response. The pitch and plunge motion and the phase plot is used to identify the onset of flutter. In Fig. 5.3, for a Mach number,  $M_\infty$  of 0.80, the flutter speed index is varied starting from flutter speed index,  $V_f$  of 0.78, until 0.86, where the onset of flutter is observed. The pitch and plunge plots and phase plots are shown for damping aeroelastic response at a flutter speed index,  $V_f$  of 0.78 and flutter response at  $V_f$  of 0.86. The flutter boundary of NACA64A010 is verified at four points spread across the transonic range. The obtained points will later be used for comparison of the effect of sloshing on the flutter boundary.

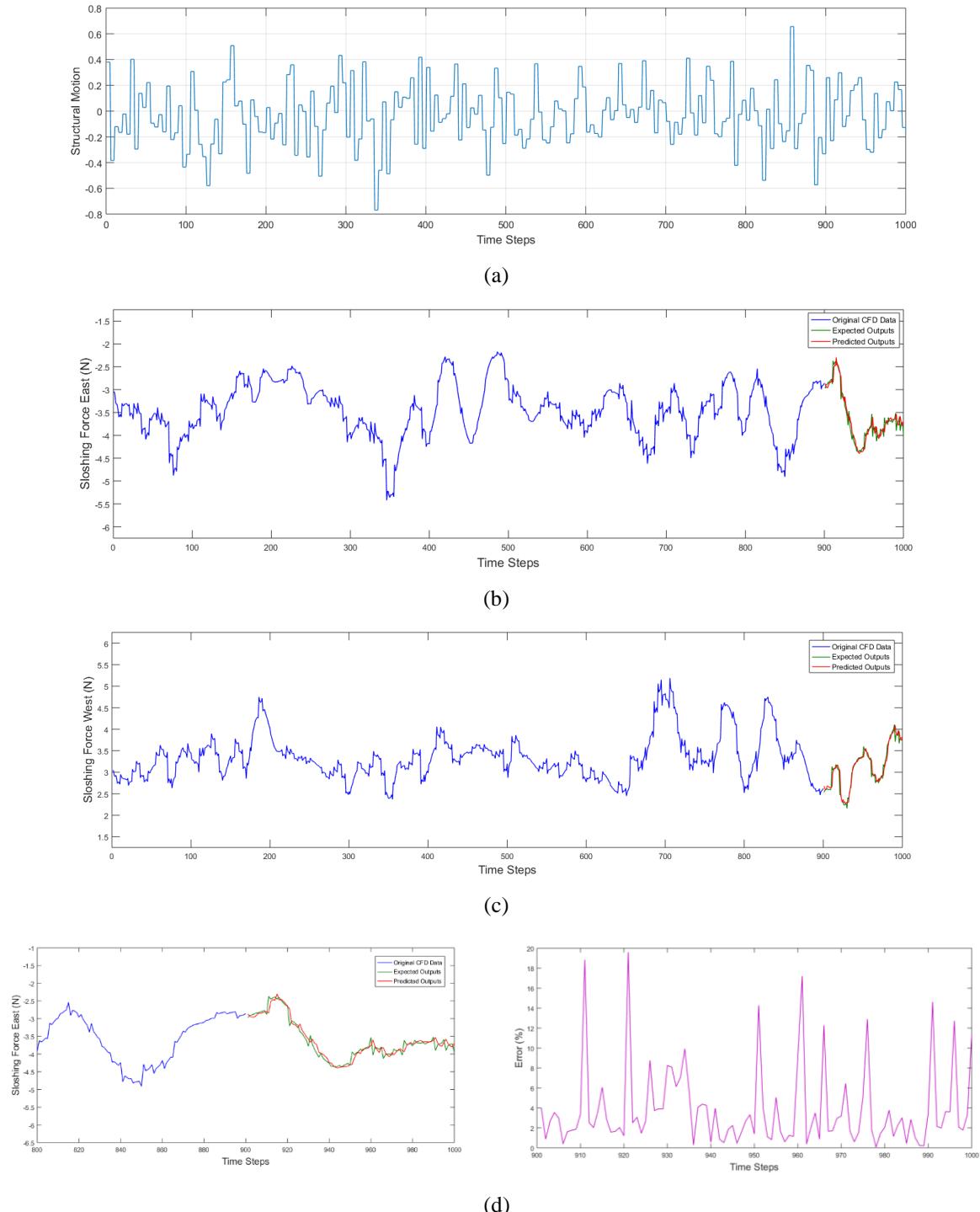


**Figure 5.3: Flutter boundary computation and comparison with Alonso and Jameson [17]**

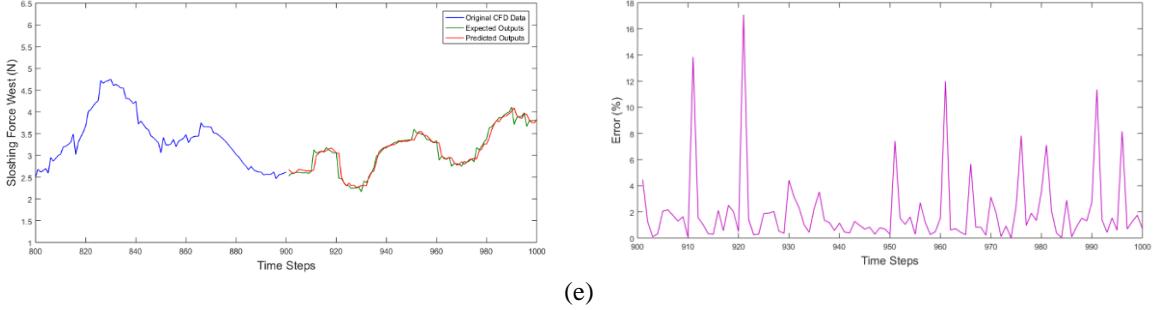
### C. Unsteady Sloshing Force Predictions Using Surrogate Model and Validation Using CFD

The constructed neural network (RBF-NN) outlined in Section 4.2, is trained using pitching and plunging data of a rigid fuel tank using N-sample constant structural excitation inputs as shown in Fig. 4.6, Chapter 4 of this thesis, which are designed by exciting natural pitching frequency of the wing section i.e.  $\omega_a$  of 100 rad/sec and average motion amplitude corresponding to the first 5 cycles from the onset of flutter. Other forms of signals can also be used for this purpose. The sloshing forces and moments on the tank wall obtained from CFD data as output corresponding to structural excitation input are divided into training data and testing data. About 90% of these data points are utilized in training of the neural network, i.e. calculation of weights and centers. The remaining 10% of the data are used for model validation. The structural input motion and corresponding prediction from RBF-NN is

compared against CFD data as shown in Fig. 5.4 (a) – (e) and the prediction error is found to be less than 8%.



**Figure 5.4:** (a) The structural input provided as pitching angle to fuel tank, (b) comparison of forces on east wall using CFD output and surrogate model prediction, (c) comparison of forces on west wall using CFD output and surrogate model prediction, (d) zoomed-in comparison of forces on east wall and corresponding errors, (Continued)



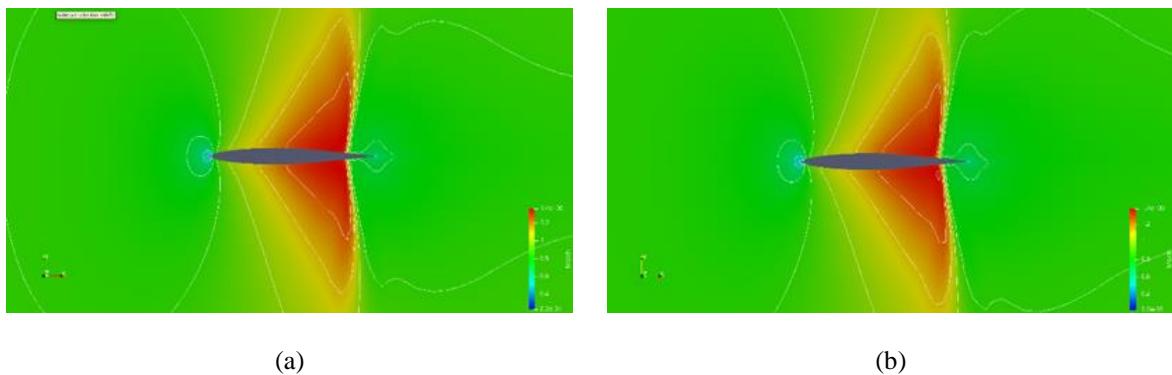
**Figure 5.4: (e) zoomed-in comparison of forces on west wall and corresponding errors**

Structural motion to the tank is given in the form of a pitching excitation for validating the predictions from the proposed RBF-NN surrogate model approach. The APRBS signal shown in Fig. 5.4 (a) is used as structural input to the half-filled fuel tank and the corresponding response forces obtained from CFD on the lateral walls of the tank are shown in Fig. 5.4 (b) and Fig. 5.4 (c) (in green line). The forces predicted from the RBF-NN are shown in red lines on the same plots with CFD force shown in green for the same structural motion. Fig. 5.4 (d) and Fig. 5.4 (e) shows zoomed-in comparison and corresponding errors of the same comparisons, respectively. It can be seen that the force predictions from the RBF-NN are in good agreement with those from CFD predictions.

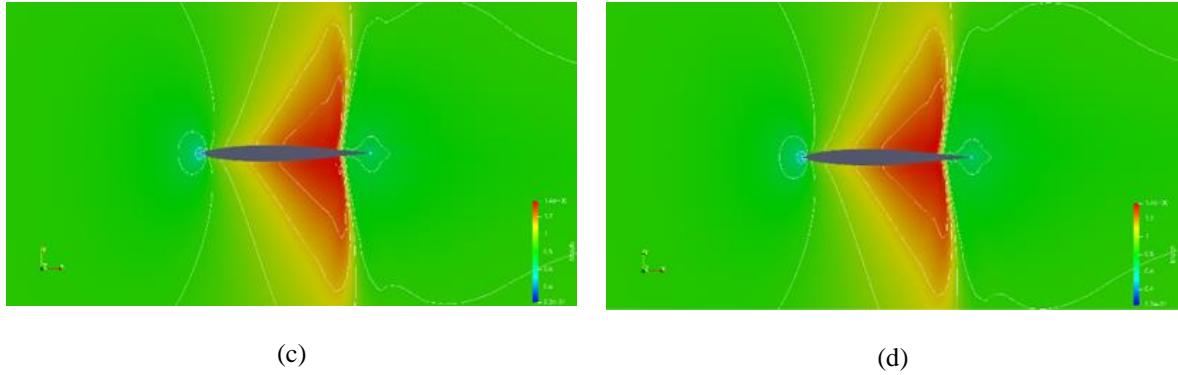
#### D. Airfoil Flutter Analysis Using the Computational Framework

The full CFD simulation of multiphase sloshing flow coupled with aeroelastic solver of *SU2* using modified *preCICE* adapters for both solvers and flutter boundary computations are done for different Mach numbers by varying the flutter speed index for each Mach number and analyzing the aeroelastic response. The fill level of the tank is fixed to 50% and the flow field is initiated with the developed flow for pitching excitation at the natural frequency and forced excitation amplitude of the containing wing section. The computed external aerodynamic flowfield is initialized by forced pitching the wing section immersed at the required flow conditions as specified earlier. This provides an initial disturbance in external flowfields, which is required for initiating the aeroelastic computation of a symmetric wing section. The free

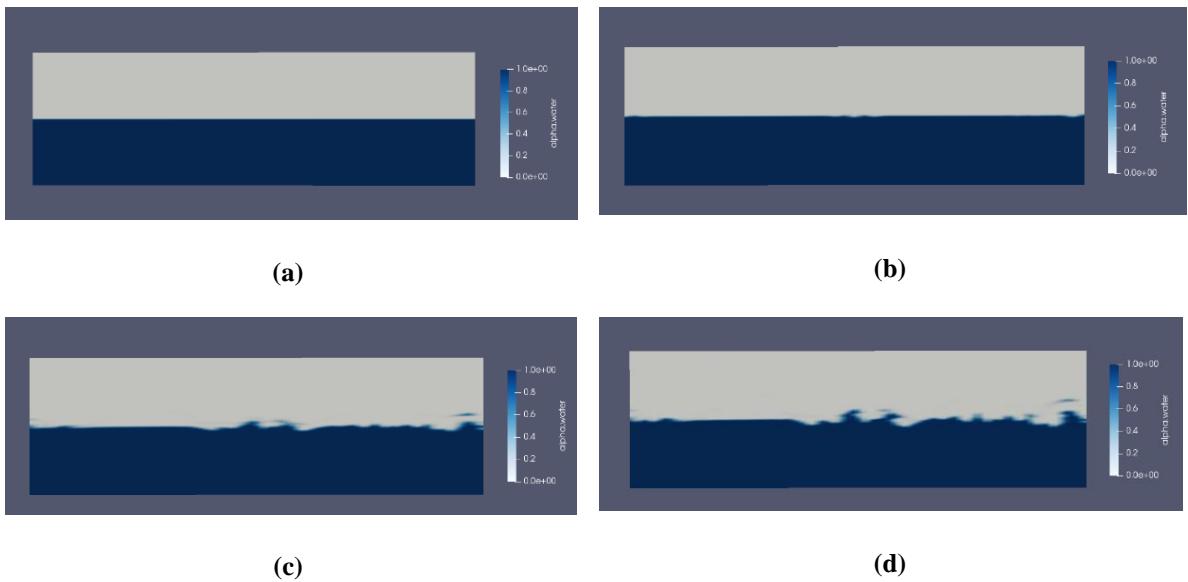
motion of the wing section is captured for the first five cycles and the aeroelastic motion is determined thereafter. The computed Mach contours for the external flowfield at selected instants of time during the unsteady motion starting from the initial state are shown in Fig. 5.5 (a) - (d). The steady-state flow-field is computed first as shown in Fig. 5.5 (a) and it is used to initialize forced pitching motion. Similarly, the flowfield obtained at the end of the forced motion shown in Fig. 5.5 (b) is utilized to describe initial flow conditions for free motion of the wing section. This process not only provides an initial perturbation to the wing-section but also speeds up the computations. Finally, Fig. 5.5 (c) and Fig. 5.5 (d) shows the flowfield around the wing section during its free motion, which in this case is diverging motion in plunge and pitch. Similarly, the fuel tank is forcefully pitched for 2 cycles and the flow-field is initialized for free motion of tank attached with wing section. The initial state of the fluid in the tank is shown in Fig. 5.6 (a), which has a 50% fill level. The perturbed fuel-vapor interface location after 2 forced pitching cycles is shown in Fig. 5.6 (b). This is followed by the free motion of the fuel tank (rigidly attached to and moving with the wing section) and fuel volume fraction is shown for 2 instances in the 5th cycle of motion in Fig. 5.6 (c) - (d).



**Figure 5.5: (a) Computed Mach contours for steady-state flow past NACA64A010 at  $M_{\infty} = 0.80$  and  $V_f = 0.70$ , (b) Unsteady flowfield at the end of 2 forced pitching cycles, (Continued)**



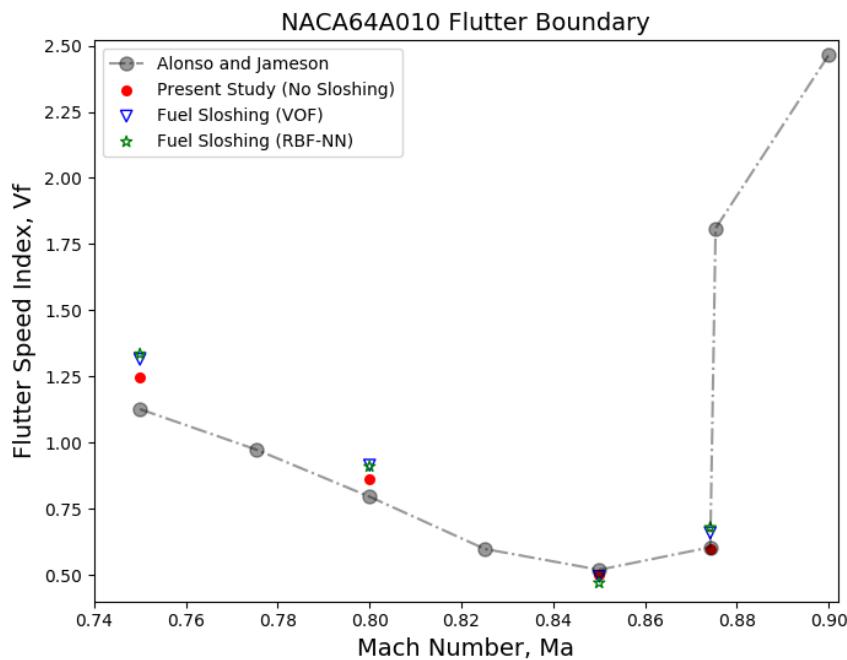
**Figure 5.5:** (c) Wing Section in  $\pi/4$  position in the 5th free cycle going into flutter, and (d) Wing Section at the end of 5th free cycle



**Figure 5.6:** (a) Volume fraction of fuel in rectangular container (50% fill level), (b) Free surface at the end of 2 forced pitching cycles, and (c) Tank at  $\pi/4$  position in the 5th free cycle, and (d) Tank at the end of 5<sup>th</sup> free cycle

The surrogate model is trained by providing structural motion to the fuel tank in the form of a combined pitching and plunging based on the APRBS signal, as described in Section 4.2 of this thesis. This signal is used as the input motion to the embedded tank inside the wing section and corresponding sloshing loads in terms of lateral forces, vertical forces and moment about pitching axis are the target outputs. As these outputs are very similar to those shown in Fig. 5.4 (b) – (c), the plots corresponding to this combined pitch plunge motion of the fuel tank are not shown here. The predicted sloshing forces and moments from the RBF-NN based surrogate model for the combined pitching and plunging motion of the fuel tank are used to compute the

aeroelastic response of the wing section due to external aerodynamic loads and sloshing loads quickly. Flutter boundary points are computed by fixing the Mach number and incrementally increasing the flutter speed index. The aeroelastic response is analyzed in time-domain and the first sign of divergence is marked as the onset of flutter. Fig. 5.7 compares the flutter boundary predicted using the full CFD model and the RBF-NN model considering the effects of fuel sloshing in the embedded fuel tanks with that without sloshing effects.



**Figure 5.7: Flutter boundary of NACA64A010 with and without fuel sloshing effects modeled by CFD and RBF-NN based surrogate model**

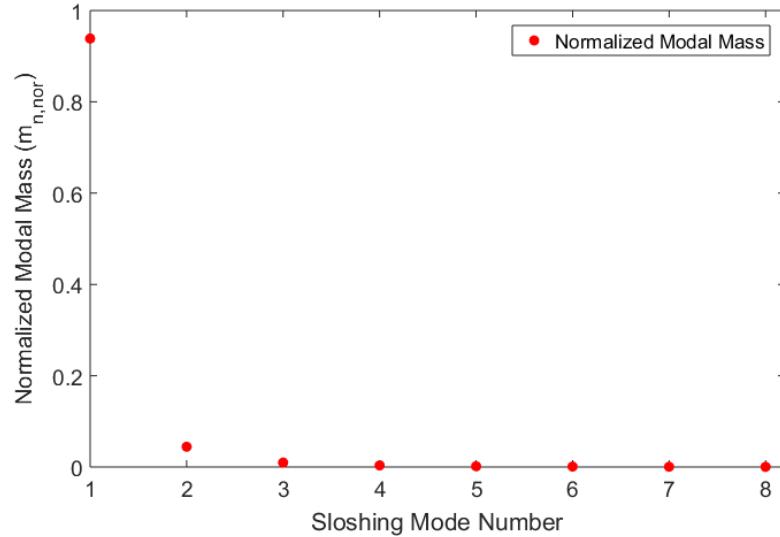
The primary focus of this work is to develop a framework to study the effects of sloshing fuel in an internal tank on the flutter characteristics of aeroelastic systems. Keeping in mind the nonlinear nature of external aerodynamics as well as sloshing of fluid, two approaches of different fidelities are adopted to simulate the effects of internal sloshing in the time-domain. The aerodynamics, however, has been computed using CFD in time domain for all cases. The coupling of *SU2* and *OpenFOAM* solvers are accomplished using the modified *preCICE* libraries. However, this method is computationally expensive, and it is not very practical for parametric variations of fuel levels, wing section geometry, and flow conditions. Hence, a

novel surrogate modeling approach is developed using radial basis function based neural networks (RBF-NN) to integrate VOF based sloshing loads with the aeroelastic system. The surrogate model successfully predicts the sloshing loads acting on the tank walls when subjected to white-noise based N-sample constant inputs. The flutter boundary is computed for NACA64A010 with and without the sloshing forces. The flutter boundary is modified by the sloshing effects of fluid inside the fuel tank. These results correspond to a particular fill level (50%) in the tank, the density and viscosity of the fluid are also fixed as well as the tank size and geometry. Each of the aforementioned factors will affect the flutter boundary and at this stage, their sensitivities to flutter computations are not known. The present framework can be utilized to study each factor individually. The RBF-NN based surrogate model for multiphase sloshing flows developed in this work provides a basis for fast prediction of flutter boundary and the onset of flutter.

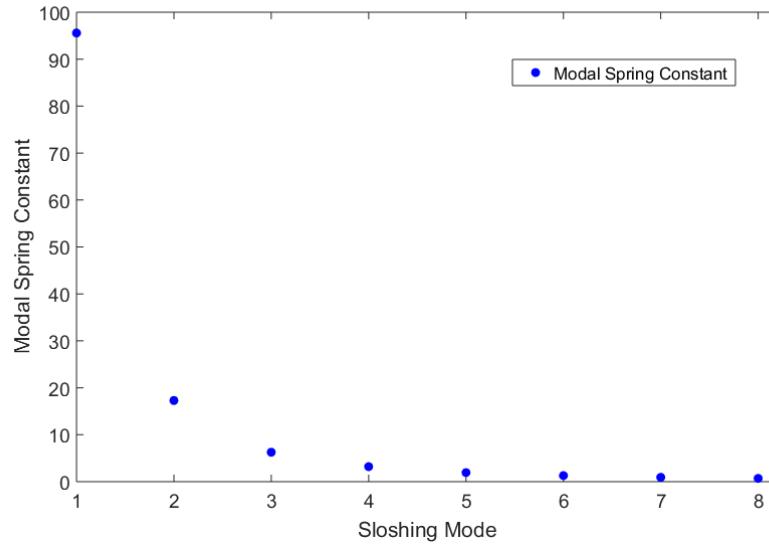
## **5.2 Flutter Boundary of NACA64A010 with EMS Model for Sloshing**

The fundamental problems of sloshing require approximation of hydrodynamic pressure distribution, integrated pressures to compute forces and moments generated, location of free surface and response frequencies of the bulk liquid. Broadly, there are two components of hydrodynamic pressure; an inertial component as a direct consequence of container acceleration, and a convective component representing free-surface liquid motions. They are modeled using a configuration of mass-spring based Equivalent Mechanical System (EMS) to represent *sloshing modes* as discussed in Section 2.2 of this thesis. Theoretically, there are an infinite number of modes and corresponding natural frequencies. However, the lowest few modes are likely to be excited and contribute to the global response of the fluid. These modes are computed using linear potential flow theory. This can be verified by computing the EMS parameters of a partially-filled fuel tank and sloshing loads by cumulatively adding higher sloshing modes. The tank width is 0.5 m and the height is 0.15 m with a water level of 0.06 m.

The contained fluid is water with a density,  $\rho_w$  of  $1000 \text{ kg/m}^3$ . The parameters are computed for the first 8 sloshing modes. The computed modal masses are normalized to 1 to observe the contribution of each modal mass. The normalized modal masses, spring constants and, natural frequencies are shown in Fig. 5.8 (a) – (c).

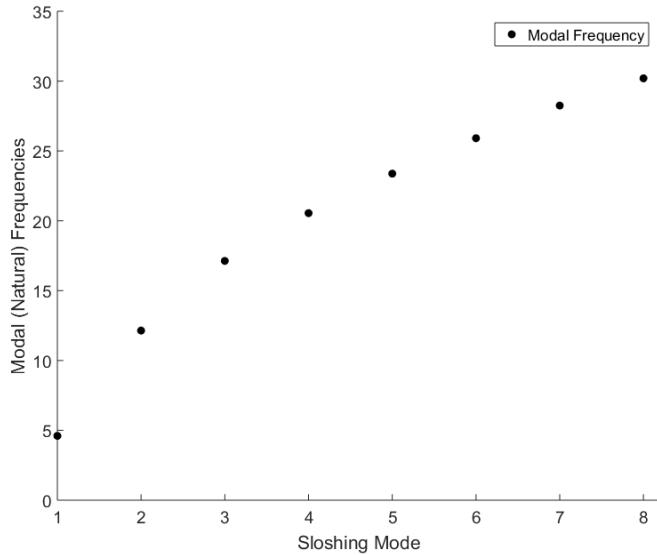


(a)



(b)

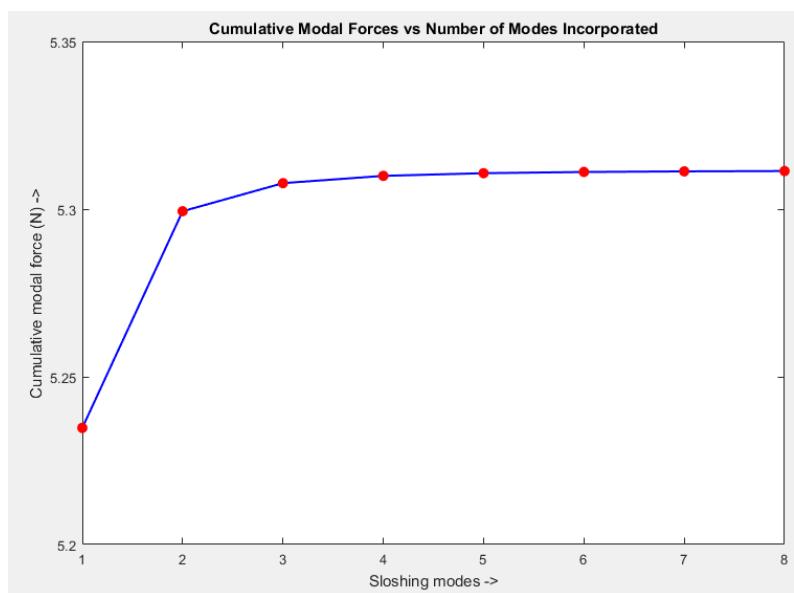
**Figure 5.8:** (a) Normalized sloshing modal mass for first 8 sloshing modes (b) Modal spring constants, (Continued)



(c)

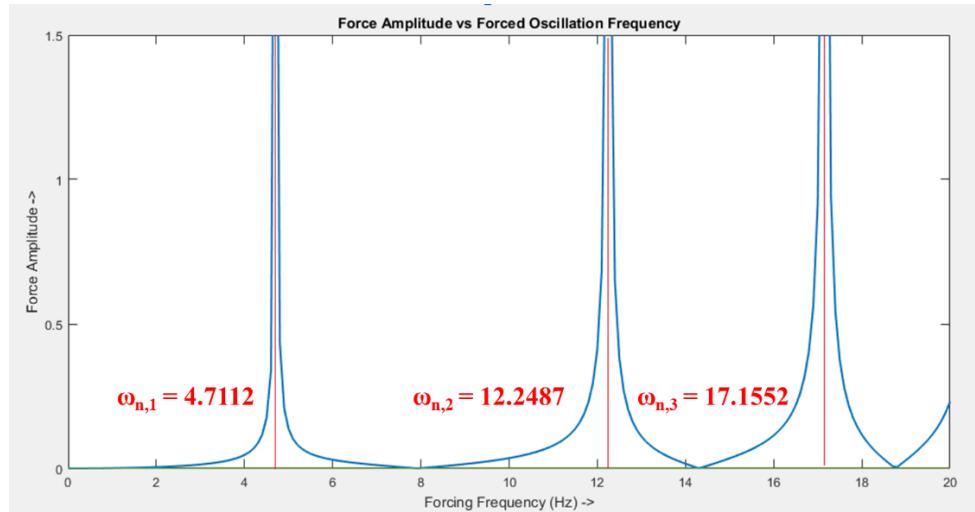
**Figure 5.8: (c) Modal (natural) frequencies of sloshing modes**

As apparent from Fig. 5.8, the first few modes play a dominant part in computing the sloshing loads in terms of forces and moments. This is confirmed by plotting the cumulative modal forces with an increasing number of modes used as shown in Fig. 5.9. The force amplitude saturates with the subsequent addition of sloshing modes. The addition of the 4<sup>th</sup> mode increases the force amplitude by just 1.14%. For the present study, only the first three sloshing modes are chosen for the EMS model.



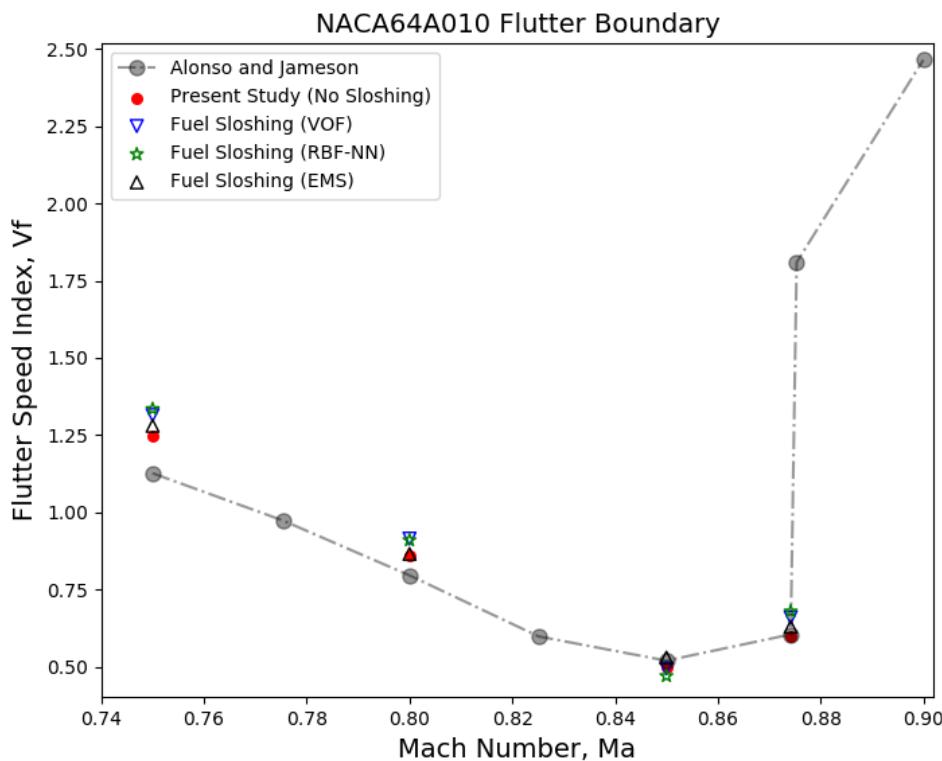
**Figure 5.9: Cumulative modal forces with increasing number of sloshing modes**

As previously discussed in Section 2.2 of this thesis, the sloshing loads are linear with forcing amplitudes. However, this holds when excitation frequencies are not close to the natural frequencies of the system. Near natural frequencies, the liquid motion exhibits nonlinear behavior in terms of sudden hike in motion amplitude, chaotic motion of fluid, wave-breaking and nonlinear mode interactions as shown in Fig. 5.10. The natural frequencies are marked in red.



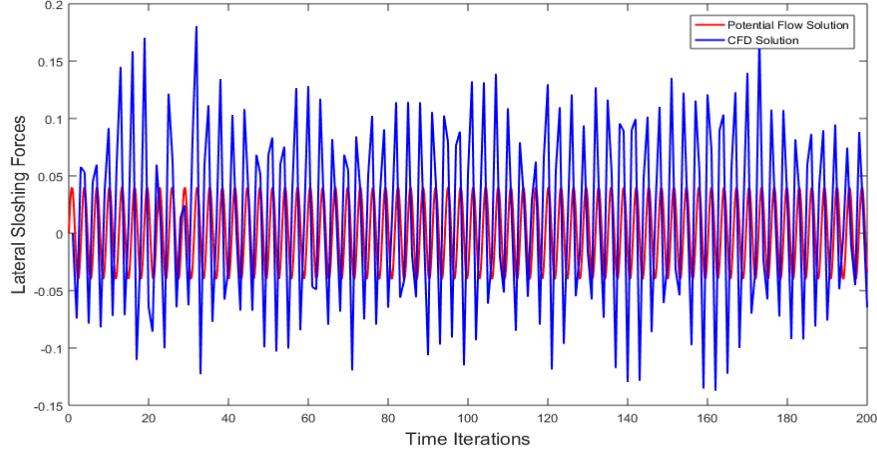
**Figure 5.10: Force amplitude response to forcing frequency**

With the above understanding, the flutter boundary of NACA64A010 is computed with sloshing modeled by the EMS model. The flutter boundary is shown in Fig. 5.11, along with the flutter boundary obtained using high fidelity CFD and RBF-NN based surrogate model.



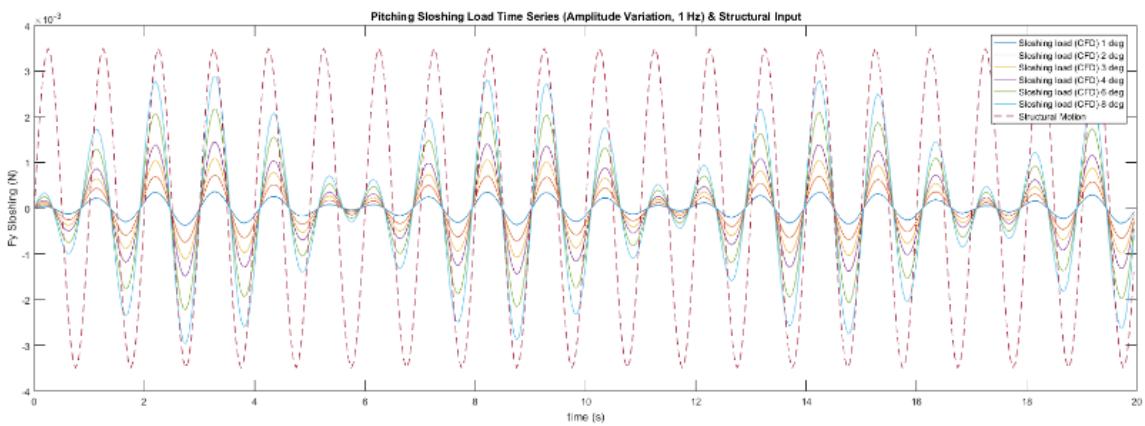
**Figure 5.11: Flutter boundary of NACA64A010 with and without the effects of fuel sloshing**

The flutter boundary obtained from the EMS model for sloshing is slightly deviated from the flutter boundary obtained using high-fidelity model. This can be attributed to many factors such as absence on nonlinear effects in sloshing and near natural forcing frequencies. It is worthwhile to compare the nature of forces obtained from CFD and EMS model based on potential flow due to the pitching motion of the tank attached to the wing section at its natural pitching frequency, i.e.  $100 \text{ rad/sec}$ . The amplitude of pitching angle is set as  $1^\circ$  and the resulting forces in the lateral direction are plotted as a time series. The forces due to sloshing are estimated at every  $t=0.02 \text{ s}$  physical time for a total time of  $T = 4 \text{ seconds}$ . A comparison of lateral forces from sloshing computed using CFD and the potential flow formulation for pitching motion is shown in Fig. 5.12 which shows that the sloshing forces at high frequencies are highly nonlinear and that the linear potential theory cannot correctly predict these loads.



**Figure 5.12: Lateral forces acting on the tank walls as a result of fuel sloshing due to pitching motion of the fuel tank at  $f = 15.91$  Hz using CFD and potential flow solution**

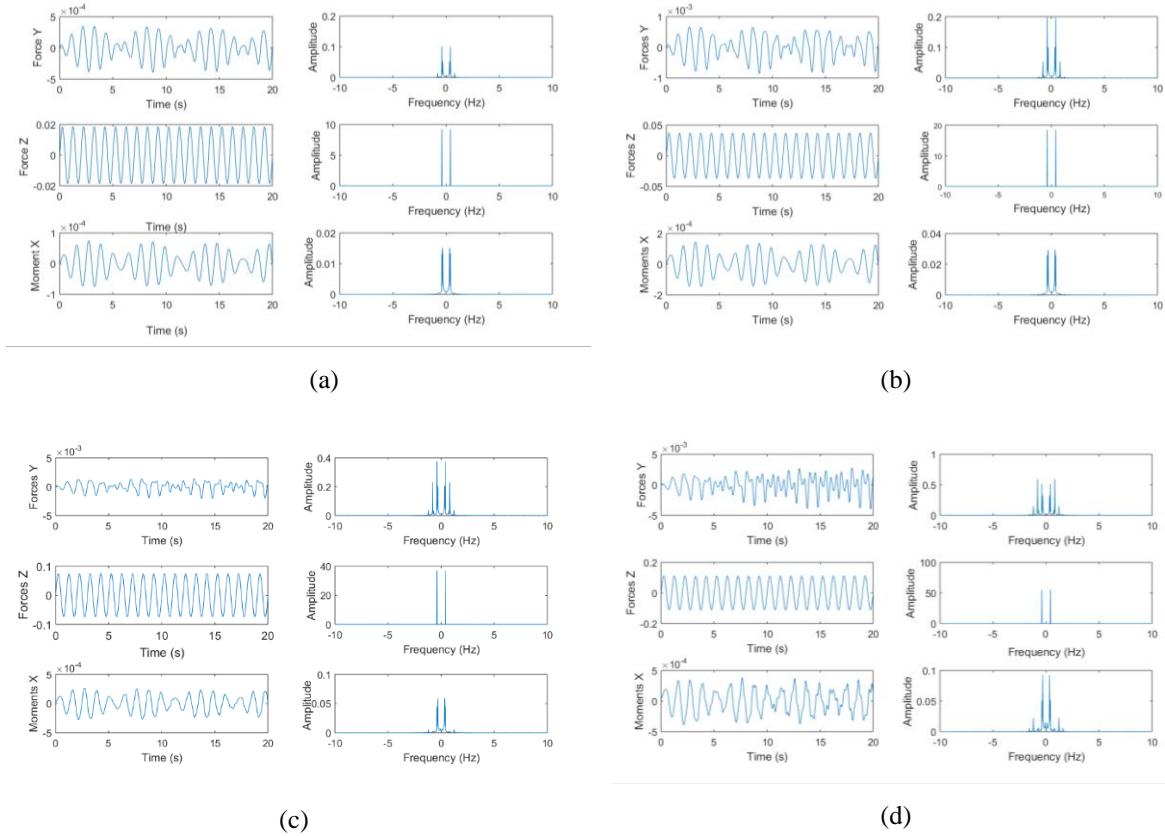
Since the correlation between sloshing loads obtained from high-fidelity CFD and EMS model is not good, it is worthwhile to analyze the limits of comparability of the EMS model with CFD. First, the linear variation of sloshing loads with amplitude is verified. This is important because EMS models can only be used for aeroelastic motions if the linear relation holds. Lateral sloshing forces on the tank wall are compared for forced pitching motion of the tank with increasing pitching amplitude ( $1^\circ$  -  $8^\circ$ ) at a constant frequency as shown in Fig. 5.13.



**Figure 5.13: Lateral sloshing forces on tank walls for forced pitching with increasing amplitude**

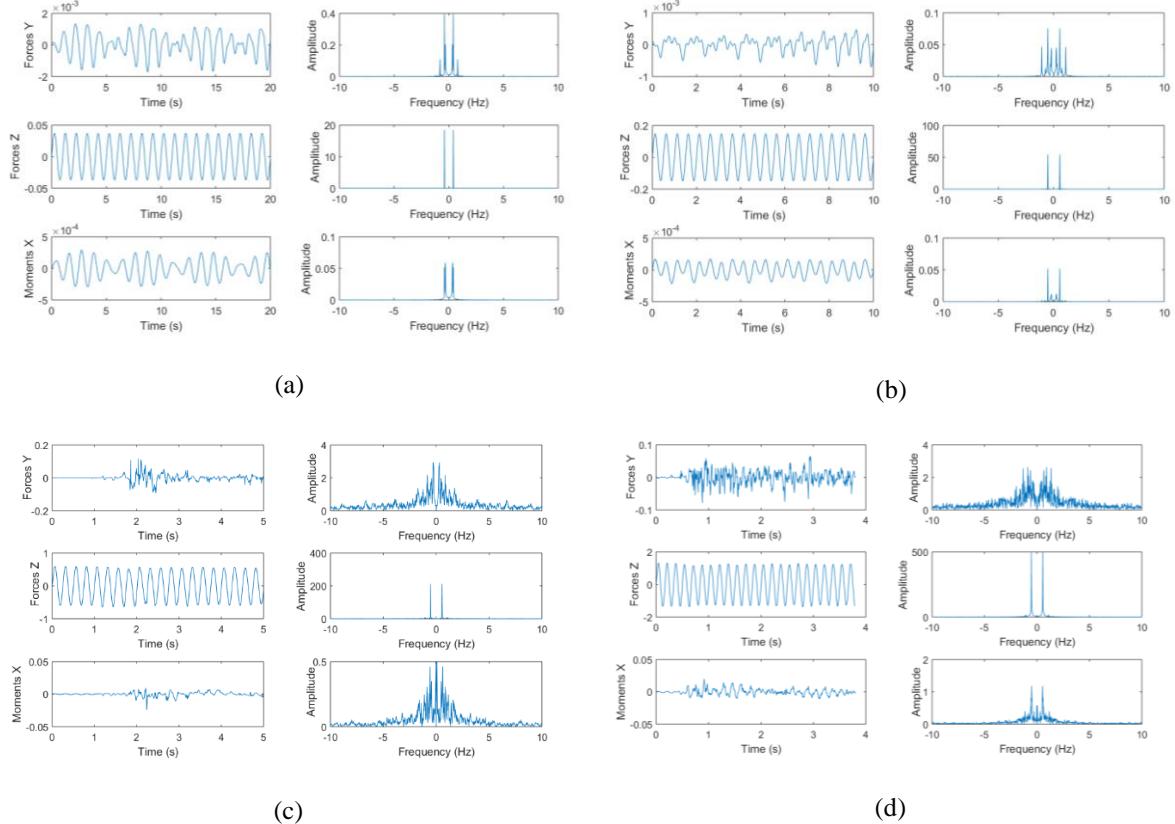
As apparent from the figure, the sloshing loads obtained from CFD are linear with forcing amplitude. However, since the aeroelastic motion is a combination of plunging and pitching motion, the parametric study with amplitude variation is done for combined motion. Figure

5.14 (a) – (e) shows the vertical sloshing force, lateral sloshing force and normal moment on the fuel tank walls and corresponding FFT plot for different values of motion amplitudes plunging and pitching at a fixed frequency of 3.14 rad/sec.



**Figure 5.14: Sloshing forces and moment time-history obtained from CFD and Fourier analysis for combined pitch-plunge motion with  $\omega=3.14$  rad/sec and (a)  $\alpha = 1^\circ, h = 2.5\%$ , (b)  $\alpha = 2^\circ, h = 5.0\%$ , (c)  $\alpha = 4^\circ, h = 10.0\%$ , and (d)  $\alpha = 6^\circ, h = 15.0\%$  of tank height**

Although the individual force or moment profile may not be important, the appearance of fewer peak, maximum of 2 in Fig. 5.14 (d) confirms that the sloshing loads obtained from CFD can be represented by a few fundamental modes, and hence comparison of CFD and EMS for amplitude variation is valid. Similarly, a parametric study of sloshing loads response using CFD with varying frequency for a fixed amplitude of motion of plunge and pitch is done. The vertical and lateral forces and normal moment to the tank wall is plotted along with their respective FFTs in Fig. 5.15 (a) – (d).



**Figure 5.15: Sloshing forces and moment time-history obtained from CFD and Fourier analysis for combined pitch-plunge motion with plunging amplitude  $h = 5.0\%$  of tank height, pitching amplitude  $\alpha = 4^\circ$  and forcing frequency of (a)  $\omega=3.14$  rad/sec, (b)  $\omega=6.28$  rad/sec, (c)  $\omega=12.56$  rad/sec, and (d)  $\omega=25.12$  rad/sec**

From Fig. 5.14 it can be inferred that the sloshing loads obtained from CFD are sensitive to excitation frequency. The response becomes more and more nonlinear with higher frequencies. More peaks in the FFT indicates that a large number of sloshing modes will be required to represent the sloshing loads obtained from CFD. However, the loads cannot be accurately captured by the EMS model due to its inherently linear nature. Hence, from the above analysis, it can be concluded that the sloshing loads obtained from CFD and EMS cannot be directly compared and the difference in the flutter boundary of NACA64A010 using CFD and EMS model for sloshing can be explained. Although the limitations of the EMS model is apparent from the last study, it still remains a useful and computationally inexpensive model for sloshing.

### **5.3 Sloshing Effects on Aeroelastic Motion of NACA0012**

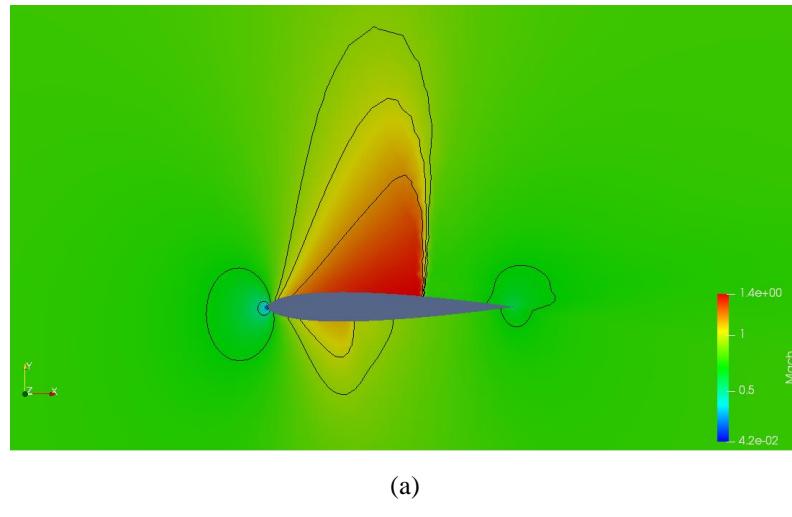
Recurrent Neural Network based surrogate models are used to predict aerodynamic loads on an airfoil and sloshing loads on a fuel tank attached rigidly to the airfoil under free motion. This framework is used to study the effects of fluid motion in the partially filled tank on the 2-DOF pitch-plunge NACA0012 airfoil. The aerodynamic loads in the form of coefficient of lift and coefficient of moments are predicted from the surrogate model based on its motion history. Similarly, sloshing loads in the form of lateral and vertical forces and normal moment are predicted by the surrogate model due to the motion of containing vessel.

An important aspect of the present study is the reduction of computational cost for investigating the multi-physics problem of fluid sloshing in a container attached to a 2-DOF airfoil while preserving the nonlinearities of flow. The choice of training data and training methodology is a critical step for this problem as it must efficiently capture the dominant dynamics of the underlying physical system. Open-source CFD code *SU2* is used for generating training data for aeroelastic prediction model, whereas *OpenFOAM* is used for generating training data for the multiphase sloshing prediction model. Data generated from these solvers are validated against results in existing literature before training samples are computed.

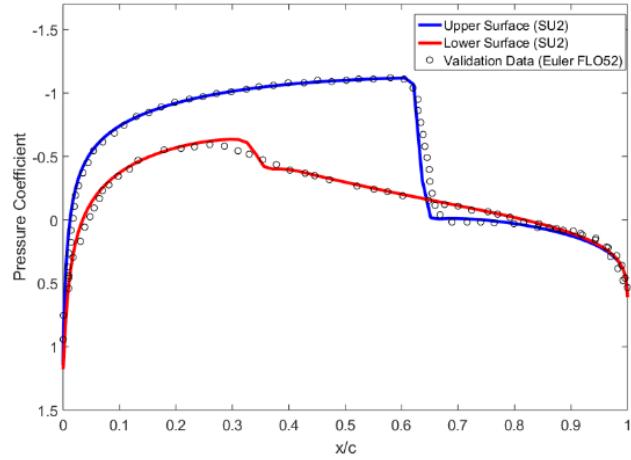
#### **A. Validation of Compressible Flow Solver for Steady and Unsteady Flow for NACA0012**

The high-fidelity solutions computed using *SU2* are validated for steady and unsteady computations. The computational domain consists of an O-type grid with far-field at 25 chord lengths from the tip of the airfoil. The upper and lower surfaces of the airfoil are divided into 100 non-uniformly spaced grids with a total of about 11,000 mesh elements in the computational domain. Steady inviscid transonic flow field around a NACA0012 airfoil immersed in a free-stream Mach number  $M_\infty$  of 0.80 and at an angle of attack,  $\alpha$  of  $1.25^\circ$  is

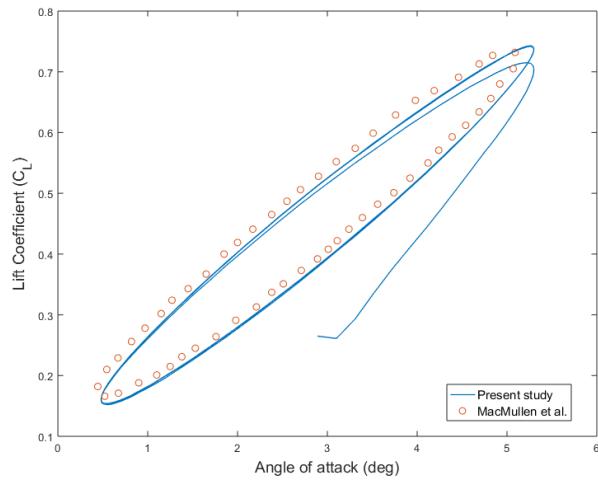
computed. The Mach contours and the coefficient of pressure,  $c_p$ , distribution on the lower and upper airfoil surfaces are shown in Fig. 5.16 (a) and Fig.5.16 (b), respectively, and validated against Jameson's data [67]. For unsteady flow validation, the airfoil is immersed in a flow field with freestream Mach number  $M_\infty$  of 0.60 and then from the steady-state flow, it is forced to pitch according to  $\alpha = \alpha_{mean} + \alpha_0 \sin(\omega t)$  where,  $\alpha_{mean}$  is  $2.89^\circ$ ,  $\alpha_0$  is  $2.41^\circ$  and  $\omega$  is  $32.9952$  (rad/s). The variation of the computed lift and moment coefficients with the pitching angle are shown in Fig. 5.16 (c) and Fig. 5.16 (d), respectively. These results are validated against McMullen's data [86]. As the computed results show good agreement with the published results, the reliability of the flow solver in generating steady and unsteady inviscid external transonic flow field data for training samples and for unsteady aeroelastic computations is thus validated.



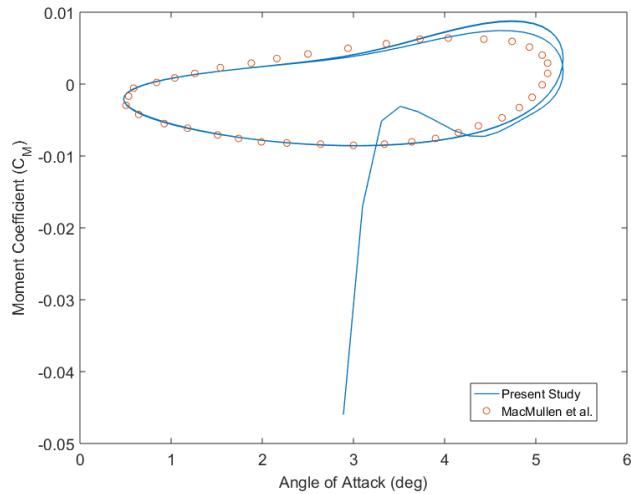
**Figure 5.16: (a) Mach contours of NACA0012 airfoil at  $M_\infty=0.80$  and  $\alpha = 1.25^\circ$ , (b) Coefficient of pressure distribution of the upper and lower airfoil surfaces, (Continued)**



(b)



(c)

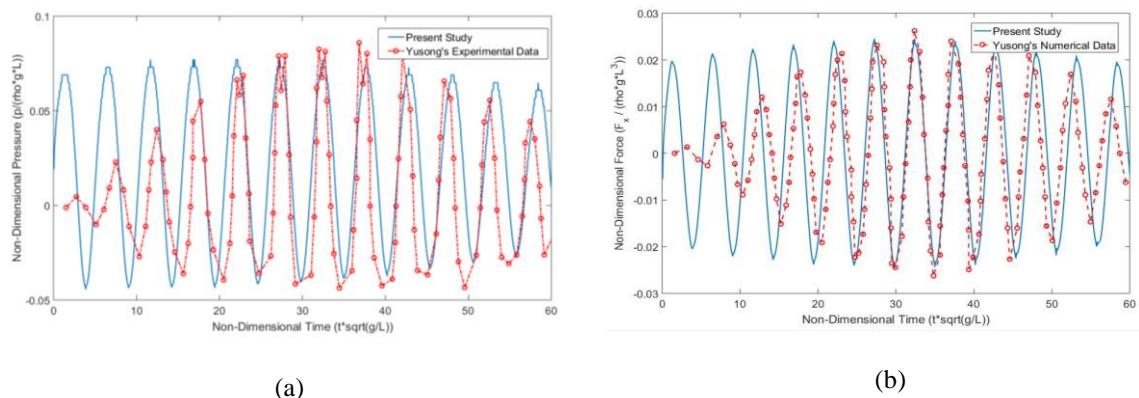


(d)

**Figure 5.16:** (b) Coefficient of pressure distribution of the upper and lower airfoil surfaces, (c) Lift coefficient variation with angle of attack for harmonic pitching comparison of present study with published results, and (d) Moment coefficient variation comparison with published data for the same motion

## B. Validation of Multiphase Flow Solver for Sloshing Flow in Fuel Tank

The *interDyMFoam* library within *OpenFOAM* is used for computing the two-phase fluid sloshing in a rectangular tank with internal dimensions of lateral length ( $L$ ), height ( $H$ ) and breadth ( $B$ ) of 1000 mm by 980 mm by 100 mm respectively with a 40% fill-level. The computational domain for the validation is the moving tank, having 1000 uniformly divided mesh cells in the lateral direction, 400 cells in the vertical direction and 100 cells in the normal direction. The tank is set to a transverse motion about the  $x$ -axis with a frequency of 1.0059 Hz and an amplitude of 0.0145 m. The dynamic pressure and forces computed from the VOF model are compared with experimental and numerical data of Yusong et al. [88] in Fig. 5.17 (a) and Fig. 5.17 (b), respectively.



**Figure 5.17: Comparison of (a) dynamic pressure, and (b) transverse sloshing forces obtained from the present study with experimental and numerical measurements of Yusong et al. [88]**

It can be seen that the computed dynamic pressure and sloshing forces estimated from the current numerical model follow the trends shown in the experimental data for the traverse motion of the partially filled tank in the latter part of the time series. The mismatch in the initial pressure and force response can be attributed to the presence of transient noise in the data. Furthermore, it must be noted that the data from literature has been manually digitized from the published figures and hence may have some measurement errors at certain points. Within

these limits, the validation of the computed sloshing loads can be concluded as being acceptable.

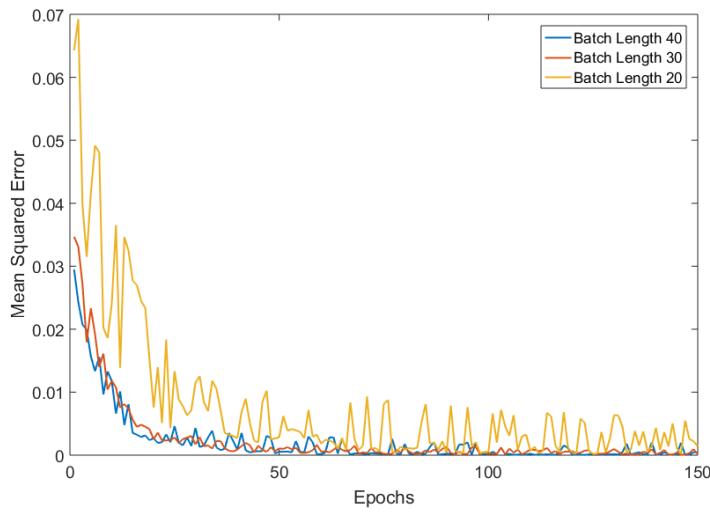
### C. Pertinent Parameters of Interest in the Machine Learning Surrogates

The input data for the RNN surrogate predicting aero-structural loads are the sequential temporal history of airfoil displacements in plunge  $h$  and pitch  $\alpha$ , and the aerodynamic lift  $C_L$  and moment  $C_M$  coefficients. The outputs produced by the network are the aerodynamic lift  $C_L$  and moment  $C_M$  coefficients at the current time step of interest. Choosing an optimum *batch length* which is one of the hyperparameters of the network is critical for the effective training and accurate prediction of the neural network. Each batch must be long enough to contain the dominant dynamic characteristics of the underlying system, but at the same time, not too long to dilute the training process as RNNs compute the loss function based on the complete batch. The inputs are weighted and passed through the activation functions in the neurons in the hidden layers to produce the outputs of the network. The outputs are compared against the ground truth data from CFD for accessing the accuracy of the surrogate model. The RNN architecture used for developing the machine learning surrogate is implemented in the *Anaconda* Python development environment [89]. Some aspects of the RNN architecture such as the effect of batch length and number of neurons on prediction accuracy are discussed in this section.

#### C.1. Choosing batch length for training and prediction

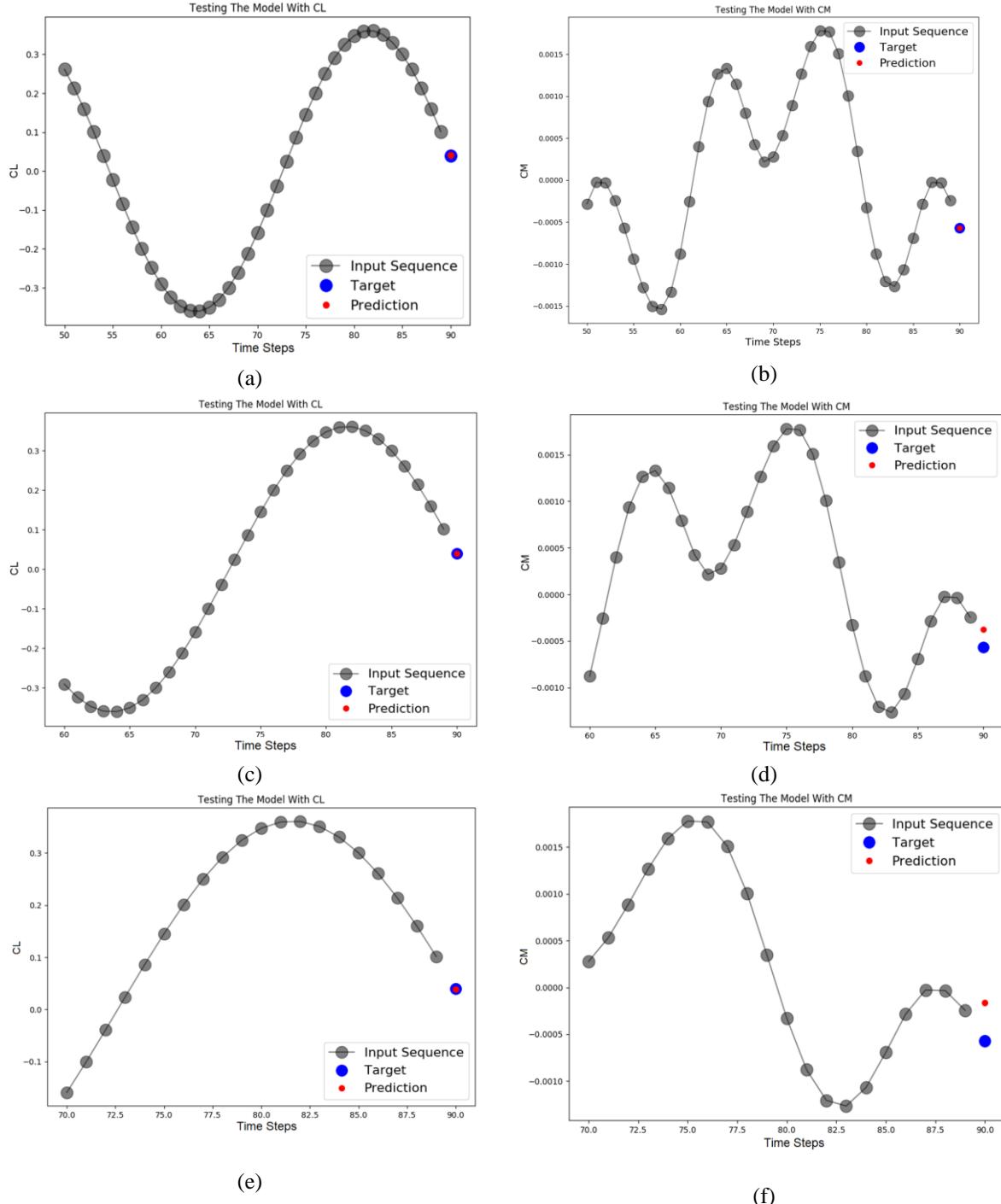
The batch length refers to the number of previous time steps chosen for the input sequence in the recurrent neural network. The choice of the batch length is critical for a recurrent neural network architecture with a given number of neurons in hidden layers as pointed out in Section 4.3 of this thesis. The RNN used here for demonstration of the effect of batch length on prediction accuracy, keeping the remaining neural network architecture parameters constant,

consists of 4 inputs namely  $h$ ,  $\alpha$ ,  $C_L$  and  $C_M$ , is trained on forced pitching motion of the airfoil. The network consists of 2 hidden layers containing 120 and 80 neurons, respectively, which predicts  $C_L$  and  $C_M$  as outputs for the next time step in the sequence. A batch length of 40 is used for the single-step prediction test. The batch length is shortened to 30 and 20 keeping the same architecture and the mean squared error during the training process is compared for the first 150 epochs in Fig. 5.18.



**Figure 5.18: Mean squared error variation with training epochs**

The mean squared error (*mse*) during training shows a descending trend for all the choices of batch length. However, it appears the convergence rate of the *mse* using longer batch lengths is better than and shows lesser oscillatory tendencies than that using shorter batch length. This is because the shorter batch length may not contain sufficient information on the dynamics of aerodynamic loads response. Now, the three networks trained using 40, 30 and 20 batch lengths are compared for prediction accuracy when fed with a sequence of aerodynamic load coefficients,  $C_L$  and  $C_M$ , and pitching motion parameter,  $\alpha$  of respective batch lengths. The three sets of input sequences for each of the networks are fed in a way so as to predict the 90<sup>th</sup> time step of the time series corresponding to Section 5.3 D of this chapter. The model predictions for  $C_L$  and  $C_M$  are shown in Fig. 5.19 (a) – (f), respectively.



**Figure 5.19:** (a) One step prediction of  $C_L$  based on input sequence for a batch length of 40, (b) Prediction of  $C_M$  for a batch length of 40, (c)  $C_L$  prediction for a batch length of 30, (d)  $C_M$  prediction for a batch length of 30, (e)  $C_L$  prediction for a batch length of 20, and (f)  $C_M$  prediction for a batch length of 20

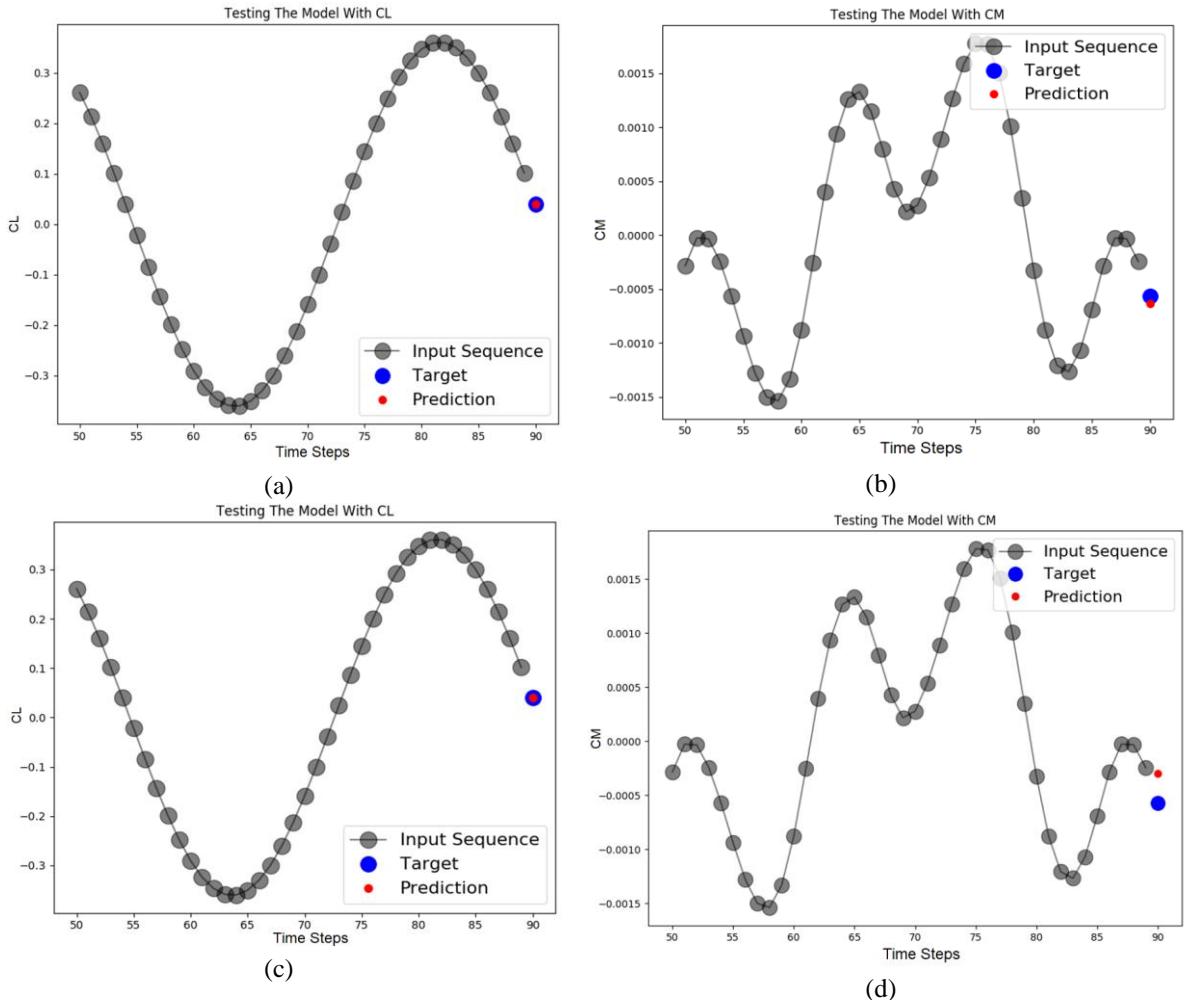
The input temporal sequences fed in the RNN are represented by black dots in Fig. 5.18 (a) – (f). The target values for the next time step are shown with a blue dot, and the actual prediction made by the model is shown with a smaller red dot. The RNN prediction accuracy deteriorates

with the reduction in batch length given the same number of neurons in the hidden layers are used. The prediction accuracy of  $C_M$  is particularly poor as compared to  $C_L$  as it has more features and lower batch lengths cannot incorporate that. This methodology of optimizing the batch length for all the other RNN based surrogates developed in the main paper is followed to maximize the accuracy of the model.

### **C.2. Choosing the number of neurons:**

The RNN has two hidden layers with 120 and 80 neurons in each layer respectively. Lesser number of neurons may underfit the problem, whereas a large number of neurons may lead to overfitting and longer training. One time step prediction accuracy of the aerodynamic surrogate model is tested by reducing the number of neurons in each layer to three-quarters, and then half, keeping a fixed batch length of 40. Single time step prediction for the 90<sup>th</sup> time step for the forced airfoil motion time series is shown in Fig. 5.20 (a) – (d).

The input temporal sequences fed in the RNN are represented by black dots in Fig. 5.20 (a) – (d). The target values for the next time step are shown with a blue dot, and the actual prediction made by the model is shown with a smaller red dot. The prediction accuracy decreases with the number of neurons, especially for the moment coefficient. The combination of 120 and 80 neurons in two hidden layers produces optimum accuracy in prediction without overfitting or underfitting the model. The same methodology of optimizing the number of neurons in both hidden layers for all other RNN based surrogates developed in this study is followed. The number of hidden layers also has a significant effect on the training of the neural network, and its performance. However, this discussion is out of scope for the present study.

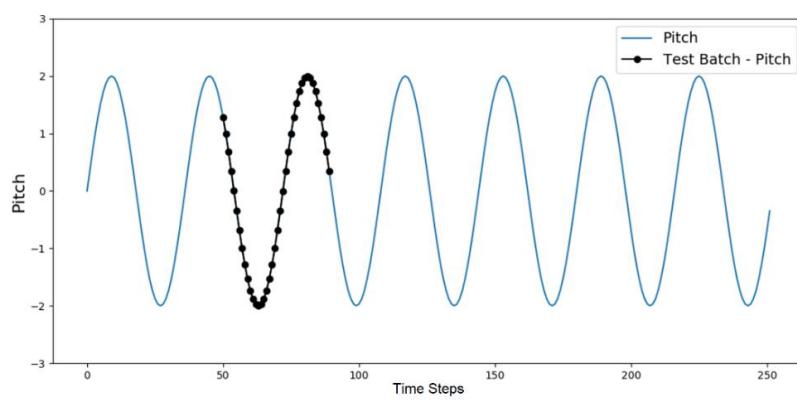


**Figure 5.20:** (a) One step prediction of  $C_L$  for RNN with 92 and 60 neurons in 2 hidden layers, (b) Corresponding  $C_M$  prediction, (c) One step prediction of  $C_L$  for RNN with 62 and 40 neurons in 2 hidden layers, and (d) Corresponding  $C_M$  prediction

#### D. Single Step Prediction Test – Aero-Structural Surrogate Model

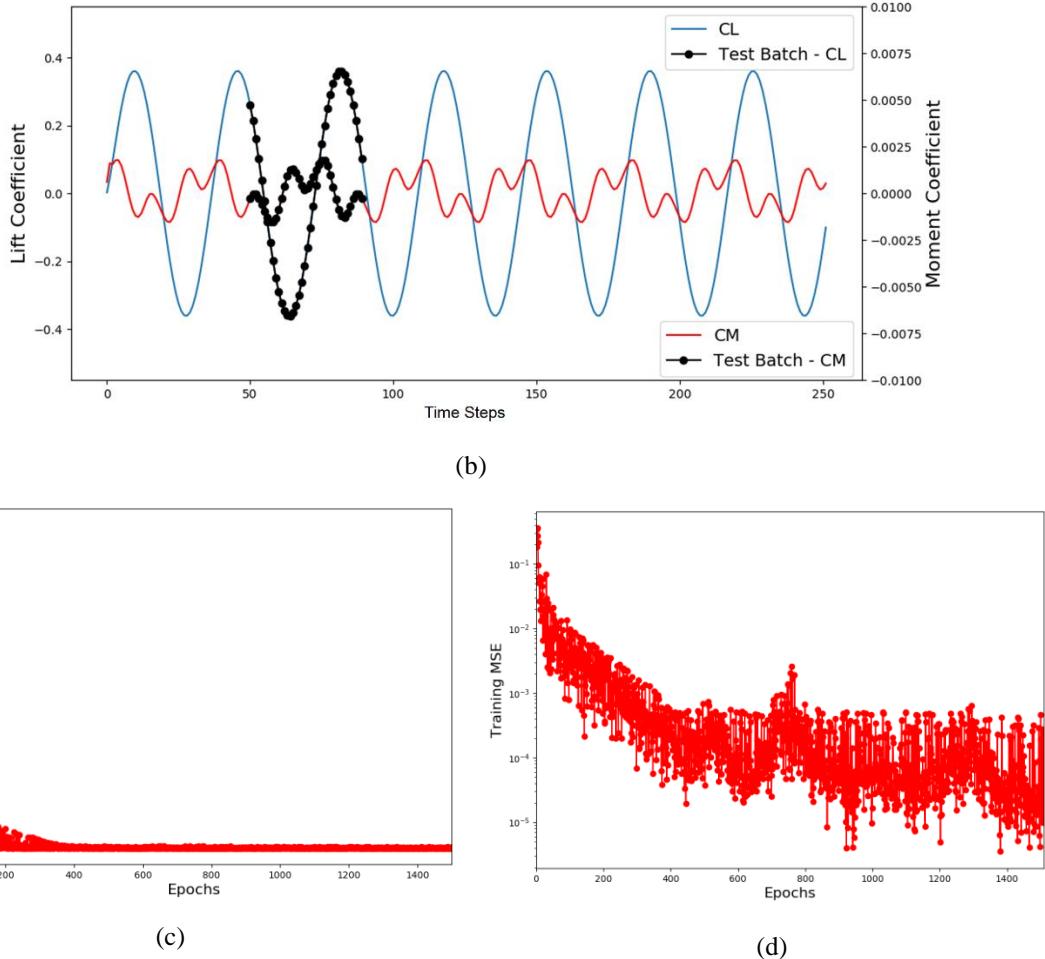
The aero-structural surrogate model is required to predict  $C_L$  and  $C_M$  at the current time step given the information of the  $h$  and  $\alpha$ , as well as the aerodynamic loads and structural displacements information from previous time steps. The unsteady airfoil motion is initiated with a forced pitching of the airfoil immersed in an external steady transonic flow field and after a few cycles of periodic variation of the aerodynamic loads, the airfoil is set to free unsteady motion which is driven by the interaction between the free structural motion and the ambient external flow field. To encompass all the possible motions of the airfoil, the prediction network is tested for both unsteady forced and free airfoil motion.

**(a) Forced Pitching of the Airfoil:** The airfoil is immersed in a uniform free stream flow at  $M_\infty$  of 0.70 and is forced to pitch according to  $\alpha = \alpha_{mean} + \alpha_0 \sin(\omega t)$  with an angular frequency  $\omega$  of  $2\pi$  (rad/s) about the angle of attack  $\alpha_{mean}$  which is assumed  $0^\circ$  and the pitching amplitude  $\alpha_0$  is  $2^\circ$ . Corresponding to the forced pitching shown in Fig. 5.21 (a), the computed time variation of lift ( $C_L$ ) and moment ( $C_M$ ) coefficients and from CFD simulation are shown in Fig. 5.21 (b). The network is trained using randomly selected input data consisting of a matrix of pitch angle, plunging displacement (all zeros in this case) and lift and moment coefficient variation with time with a batch length of 40 for each epoch. An example of one such training instance is shown in Fig. 5.21 (a) - (b) with black dots (i.e. corresponding to the range of 50 to 89 time steps in the training data). The network consists of two hidden layers consisting of 120 and 80 neurons, respectively. The training errors are estimated using the mean squared error ( $mse$ ) of predicted and expected values for the batch sequences of outputs, in this case, lift and moment coefficients. The network weights and biases are learned during the network training in order to minimize the loss function, the convergence of which is shown in Fig. 5.21 (c). For the training instance considered the convergence of the  $mse$  is of the order  $O(10^{-5})$ , which is achieved after 1500 epochs as shown in Fig. 5.21 (d). Once trained, the network can be tested to assess its predictive reliability for one time step ahead for any sequence of inputs.



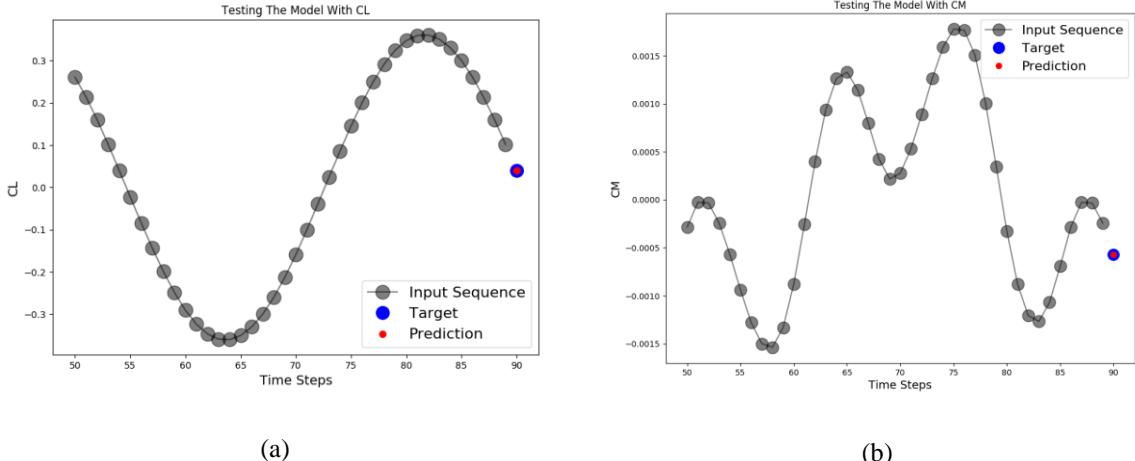
(a)

**Figure 5.21: (a) Pitching displacements of the airfoil in free motion with a random batch of inputs for testing the RNN, (Continued)**



**Figure 5.21: (b) Corresponding variation of lift and moment coefficients; Variation of (c) MSE, and (d) MSE (in log scale) vs. epochs during network training for aerodynamic data.**

Once trained, the RNN is tested with a batch of input sequence for demonstration of prediction accuracy. The input sequences shown in Fig. 5.21 (a) - (b) with black dots, corresponding to the range of 50 to 89 time steps in the training data is selected. These input sequences are fed into the RNN to predict the lift and moment coefficient at the next time step, i.e., the 90<sup>th</sup> time step in the training data. Fig. 5.22 (a) - (b) show an enlarged view of the training instances (labeled as grey points) corresponding to the batch length shown as black dots in Fig. 5.21 (a) - (b) (i.e. input sequence), expected output (i.e. target) and actual network outputs (i.e. predicted) respectively for both aerodynamic lift and moment coefficients. The target prediction at the 90<sup>th</sup> time step is shown with a blue dot and the actual network prediction is shown with a red dot.

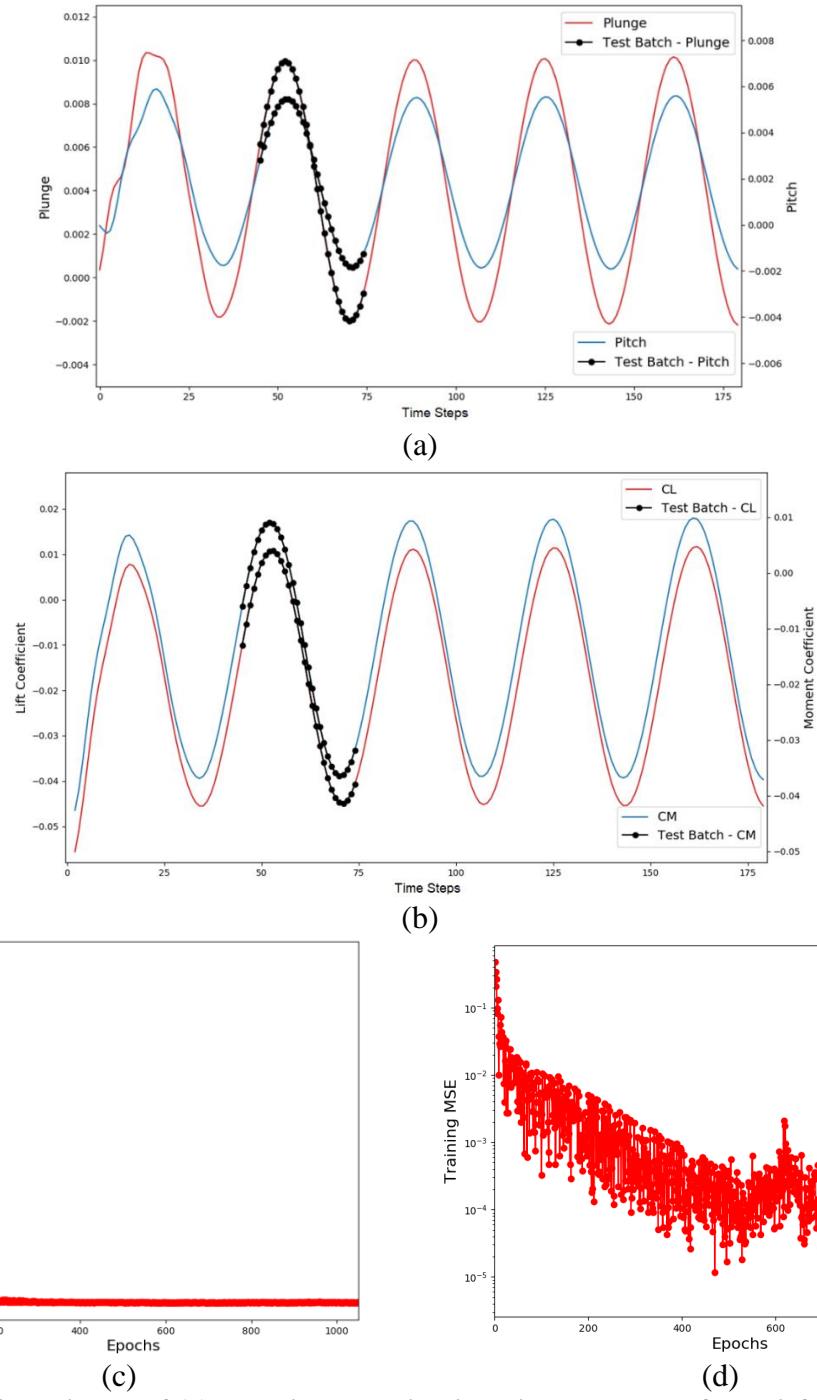


**Figure 5.22:** The training instance, the expected output and the network prediction for lift coefficient using randomly chosen batch for (a) lift, and (b) moment coefficients

The lift and moment coefficients at one time step ahead of the input series (give value of  $t$ ) predicted by the RNN (labeled as red dot in Fig. 5.22 (a) - (b)) is compared with the target values computed from CFD simulation (labelled as blue dot in Fig. 5.22 (a) - (b)). The defined acceptable error for single-step prediction is of the order  $O(10^{-5})$ . The accuracy of prediction is measured by computing squared errors for  $C_L$  and  $C_M$  at the time step of interest, which is achieved. This is expected as the RNN has been trained to reduce  $mse$  to this order, shown in Fig. 5.21 (d). The prediction accuracy of the RNN can be improved arbitrarily further by tuning the hyperparameters.

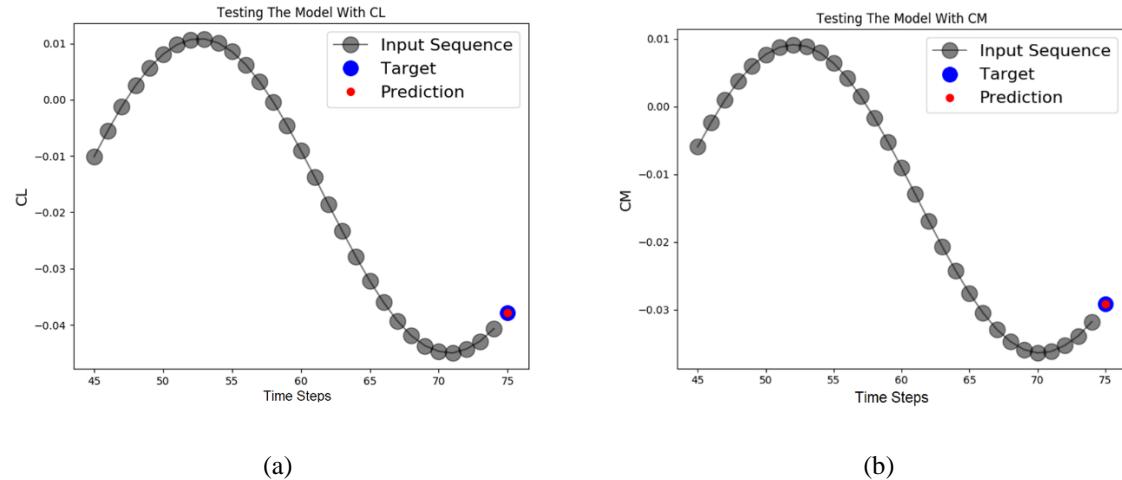
**(b) Free Plunging and Pitching of NACA0012:** After a few cycles of forced pitching and plunging of the airfoil initialized with the flow field at free stream  $M_\infty$  of 0.70 at  $0^\circ$  angle of attack, the airfoil is set free to move so that the pitch and plunge displacements are determined via the interaction with the structural dynamics and ambient unsteady aerodynamic flow. Fig. 5.23 (a) shows the time history of plunging and pitching displacements of the airfoil and Fig. 5.23 (b) shows the corresponding lift and moment coefficients. Fig. 5.23 (c) shows the convergence of weights and biases for the present training instance in form of reduction in  $mse$ , which converges to the order  $O(10^{-5})$  after 1000 epochs, as shown in Fig. 5.23 (d). This airfoil

displacement and aerodynamic loads data are used as the training set for one-step aeroelastic motion prediction model.



**Figure 5.23:** Time history of (a) plunging and pitching displacements of the airfoil in free motion with a random batch of inputs for testing the RNN, (b) Corresponding lift and moment coefficients; Variation of (c) MSE, and (d) MSE (in log scale) vs. epochs during network training for aerodynamic data.

The RNN architecture is trained on free plunging and pitching data and the learned weights and biases are used for testing its prediction accuracy. The training inputs are sequences of data from airfoil motion series in the form of plunging and pitching and the corresponding aerodynamic loads  $C_L$  and  $C_M$ . The input sequences are selected randomly for each training epoch; an instance corresponding to time steps ranging from 45 to 74 is labeled with black dots in Fig. 5.23 (a) and Fig. 5.23 (b). These inputs sequences are fed into the RNN to check the prediction accuracy for one time-step in future. Enlarged views of the training instance, expected output and actual network outputs are shown in Fig. 5.24 (a) and Fig. 5.24 (b), respectively. The trained network is tested with a batch of input sequences for lift and moment coefficients shown as grey dots in Fig. 5.24 (a) - (b) corresponding to time steps ranging from 45 to 74 of the training data.



**Figure 5.24: (a) The training instance, the expected output and the network prediction for lift coefficient using randomly chosen batch, and (b) Same parameters shown for moment coefficient**

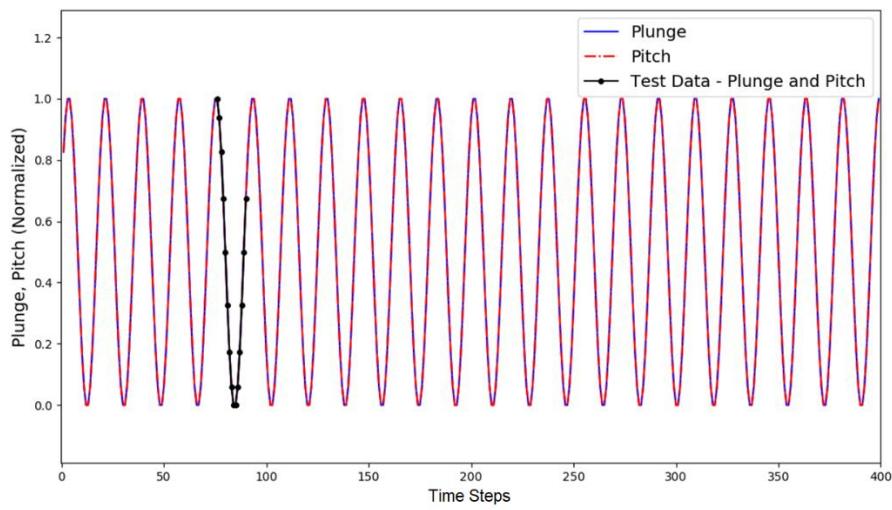
The target prediction at the 75<sup>th</sup> time step is shown with a blue dot, and the actual network prediction is shown with a red dot. The prediction accuracy is estimated by computing the squared error of the outputs at the time step of interest. The error is found to be of the order  $O(10^{-4})$  for the aforementioned testing, which is expected as the  $mse$  is reduced to order between  $O(10^{-4})$  and  $O(10^{-5})$  during training, as shown in Fig. 5.23 (d).

## E. Single Step Prediction Test for RNN Surrogate Model for Tank Fuel Sloshing Loads

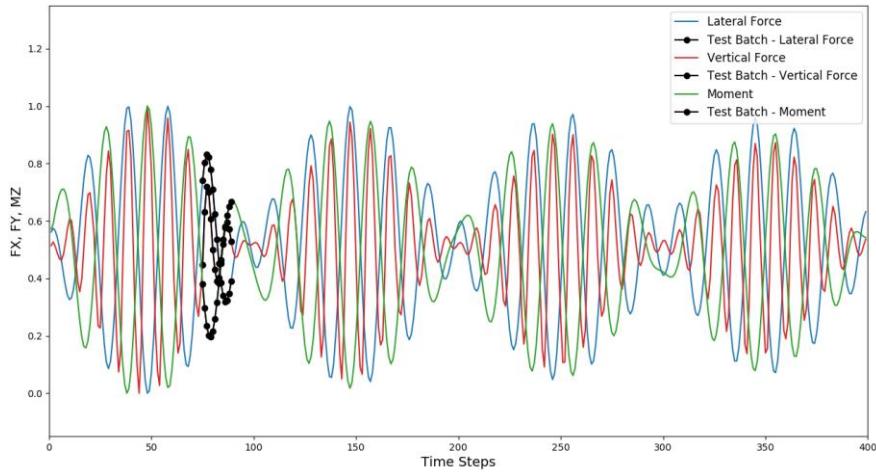
### Prediction

The RNN surrogate prediction model for fuel sloshing in the tank aims to predict *lateral* and *vertical* forces on the tank walls arising from fuel sloshing and the *moment* generated by these forces about the geometric center of the tank at the present time step. As the tank undergoes forced pitching and plunging, the inputs for the model consists of the information of plunge,  $h$  and pitch,  $\alpha$ , at the previous and current time step and also the information of  $F_X$ ,  $F_Y$  and  $M_Z$  at the previous time step. The data used for training and testing the RNN are scaled to values between 0 and 1, as input data sequences of the same order of magnitude facilitates efficient training of weights and biases. Figure 5.25 (a) shows the normalized temporal history of the pitch and plunge motion and Fig. 5.25 (b) shows the corresponding sloshing forces and moments.

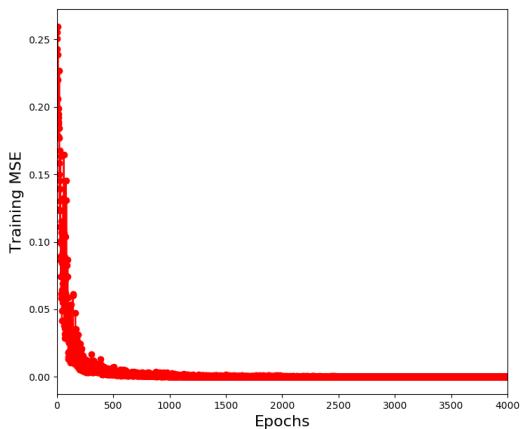
The network is trained where inputs are the temporal sequences of tank motion in form of plunging ( $h$ ) and pitching ( $\alpha$ ) at the present and previous steps, and time history of previous sloshing loads  $F_X$ ,  $F_Y$  and  $M_Z$ , respectively. The network outputs are  $F_X$ ,  $F_Y$  and  $M_Z$  at the present time step. The network architecture contains two hidden layers, containing 170 and 120 neurons each. The total number of neurons used for this case is more than that used for the pitching and plunging airfoil aerodynamics in Section 5.3 D of this chapter because there are a greater number of inputs and outputs and the outputs have more features than those of pitching and plunging airfoil aerodynamics. A batch length of 15 previous time step data is used for the inputs and *loss* is computed for subsequent epochs during the training process. Fig. 5.25 (c) and Fig. 5.25 (d) show the convergence of weights and biases for the present training instance in form of reduction in *mse*, which converges to the order  $O(10^{-5})$  after 4000 epochs, as apparent from Fig. 5.25 (c) - (d).



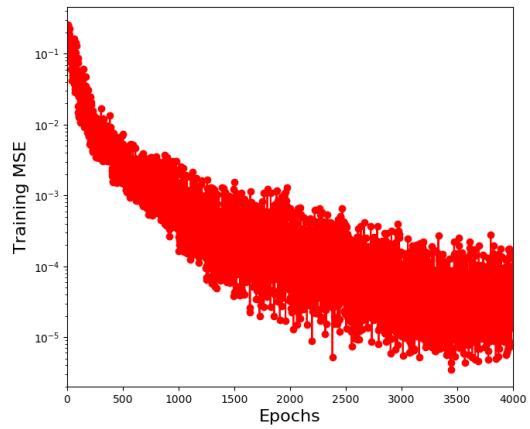
(a)



(b)



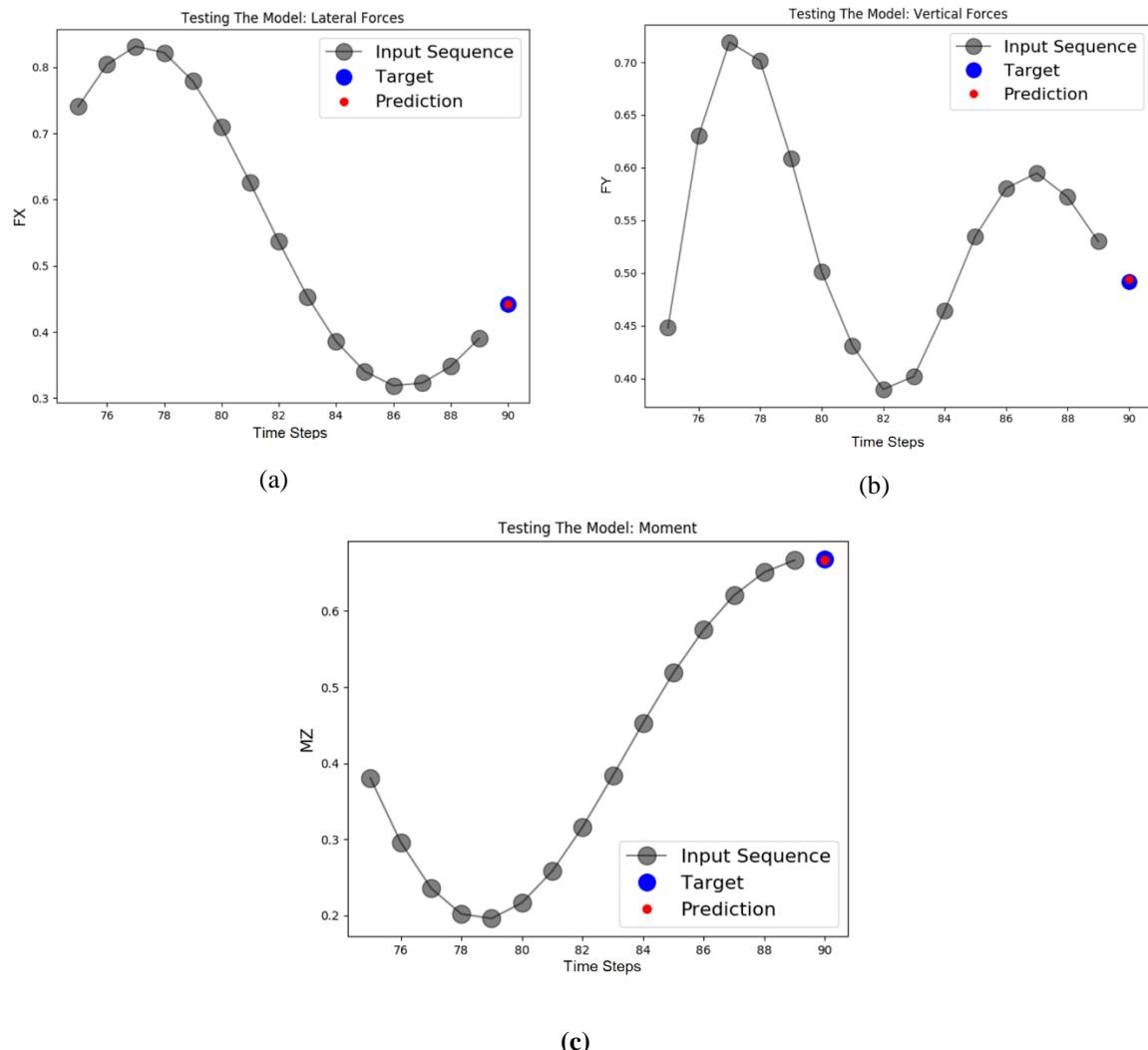
(c)



(d)

**Figure 5.25:** (a) Pitch and plunge motion time series structural inputs for tank motion, and (b) Sloshing loads in response to tank motion; Variation of (c) MSE, and (d) MSE (in log scale) vs. epochs during network training for aerodynamic data.

To evaluate and demonstrate the prediction accuracy of the trained RNN, input sequences of plunging and pitching motions shown with black dots in Fig. 5.25 (a), and corresponding  $F_x$ ,  $F_y$  and  $M_z$ , shown in Fig. 5.25 (b) are selected from the training data and is fed into the RNN. The input sequences range from 75 to 89 time steps of the training data to evaluate the sloshing loads prediction at the 90<sup>th</sup> time step. Zoomed-in comparison of training data expected values and predicted values model outputs namely, *lateral forces*, *vertical forces* and *moment* respectively represented by grey dots, blue dot, and red dot, respectively, are shown in Fig. 5.26 (a) - (c).



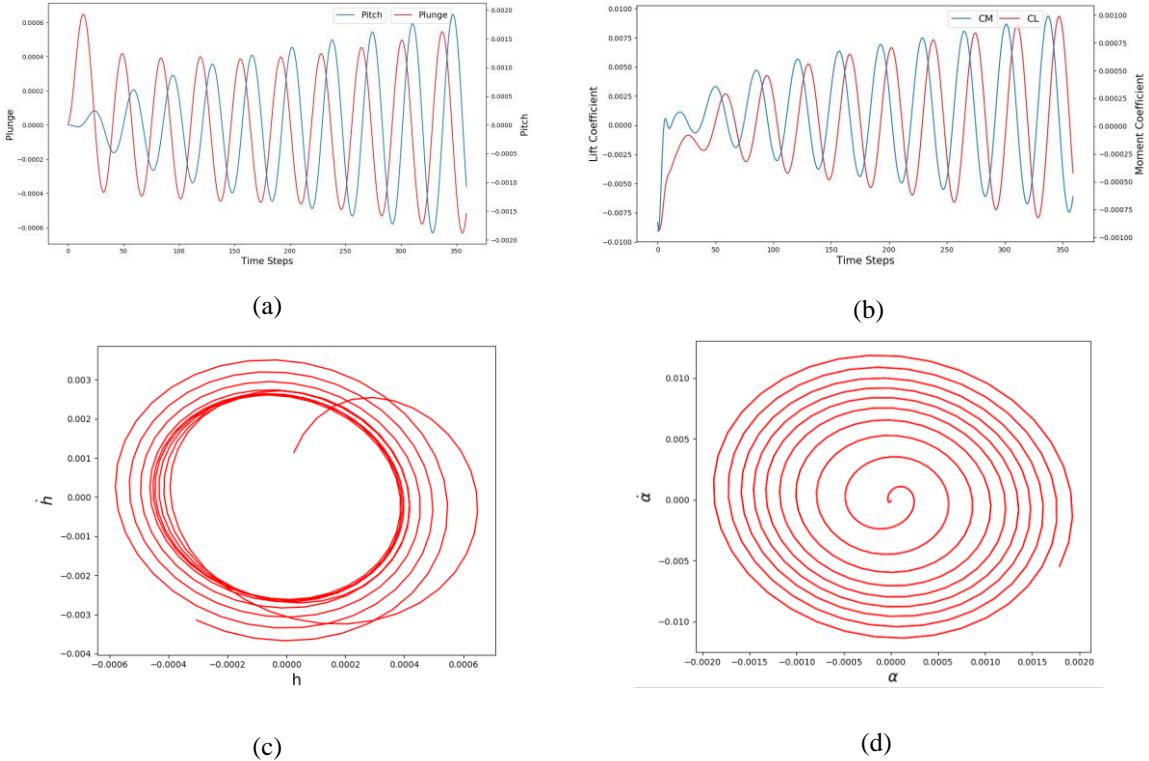
**Figure 5.26: The training sample, expected values and prediction values of surrogate model for, (a) lateral force, (b) vertical force, and (c) moment**

The prediction error is computed by taking the squared difference of the predicted value and the CFD data for  $F_X$ ,  $F_Y$  and  $M_Z$  at the time step of interest. The squared error is found to be of the order  $O(10^{-4})$ . This is expected as the RNN has been trained to reduce the *mse* between the order  $O(10^{-4})$  and  $O(10^{-5})$ , as apparent from the *mse* shown in Fig. 5.25 (d). This range of prediction error is deemed acceptable for the present study. However, the prediction accuracy of the RNN can be improved further by tuning the hyperparameters of the network.

## **F. Prediction of Aeroelastic Responses Using RNN**

On the basis of the accuracy of RNN surrogate prediction of aerodynamic load coefficients and sloshing loads at one time step ahead of the input sequences, in this section, the surrogate model is used to predict the system responses for a longer time by considering a case of a diverging aeroelastic response of the wing section. This is facilitated by appending the network output obtained at the current time step to the sequence of its previous values and then using it as inputs for prediction at the next time step as outlined in Section 4.3 of this thesis.

**(a) Aeroelastic Response Prediction:** For this case, a higher freestream Mach number is chosen for free motion, which in turn has a more diverging aeroelastic response for testing the robustness of the RNN prediction. The surrogate prediction model is built on the RNN architecture for the aeroelastic response of a NACA0012 airfoil which is forced to pitch and plunge for two cycles from a freestream flow at Mach number,  $M_\infty$ , of 0.80 and then set free to pitch and plunge. The flutter speed index,  $V_f$ , for this case is 0.425. The pitching and plunging motion, lift and moment coefficients and the corresponding phase plots for pitching and plunging are computed on the basis of free pitching and plunging motion of the airfoil and are shown in Fig. 5.27 (a) - (d),

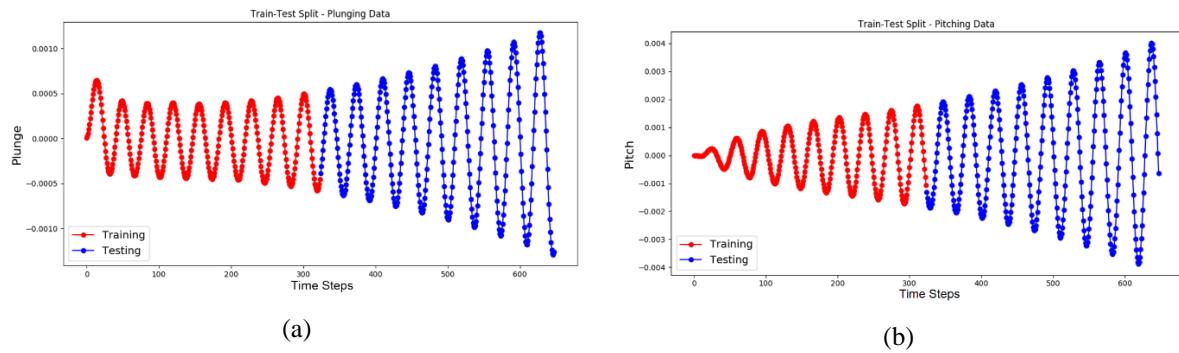


**Figure 5.27:** (a) Plunging and pitching displacements of the airfoil in free motion, (b) corresponding lift and moment coefficients obtained for the free motion of the airfoil, (c) phase plot for plunging, and (d) phase plot for pitching

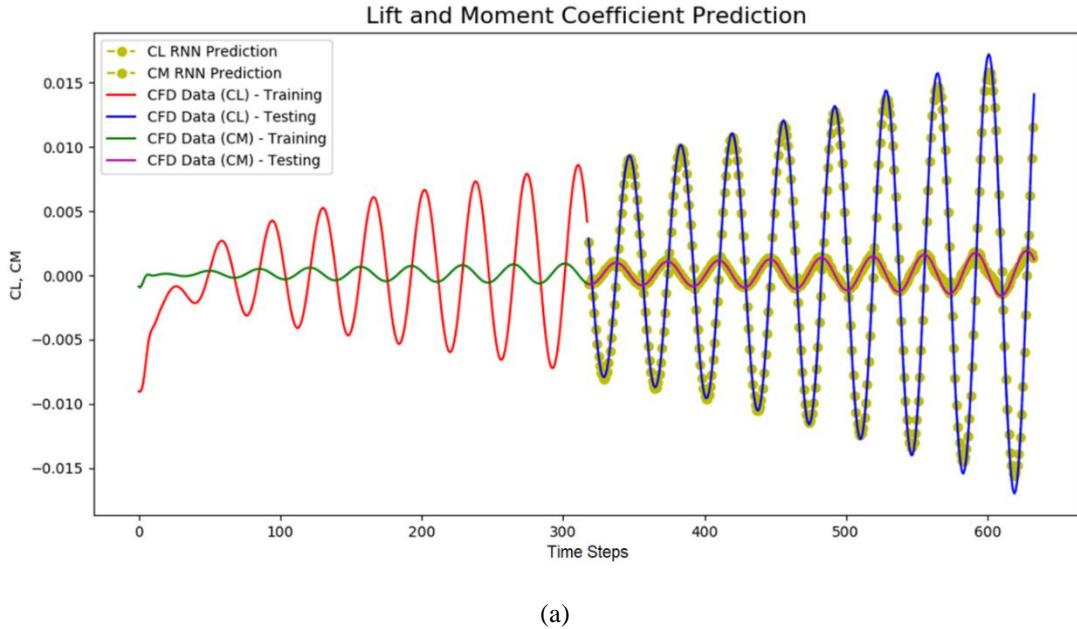
The computed high-fidelity CFD dataset is first divided into training (red color) and testing (blue color) data sets as shown in Fig. 5.28 (a) - (b) which show, respectively, the plunging and pitching data sets. A train-test data ratio of 1:1 is maintained for all other inputs as well, namely,  $C_L$  and  $C_M$ . The RNNs are trained using the data shown in red for a combined plunging and pitching motions and their corresponding lift and moment coefficients. Once trained, the surrogate model will be tested for prediction accuracy using the testing dataset shown in blue.

The RNN architecture with four input sequences feeding through two hidden layers of neurons using *ReLU* activation function generating two outputs, as detailed in Section 4.3 of this thesis, is trained with input sequences consisting of plunge and pitch displacement data and the corresponding lift coefficient ( $C_L$ ) and moment coefficient ( $C_M$ ) data with batch length of 30. These training data for lift and moment coefficients are shown in red and green lines, respectively, in Fig. 5.29 (a). The predicted variation of lift and moment coefficients from the

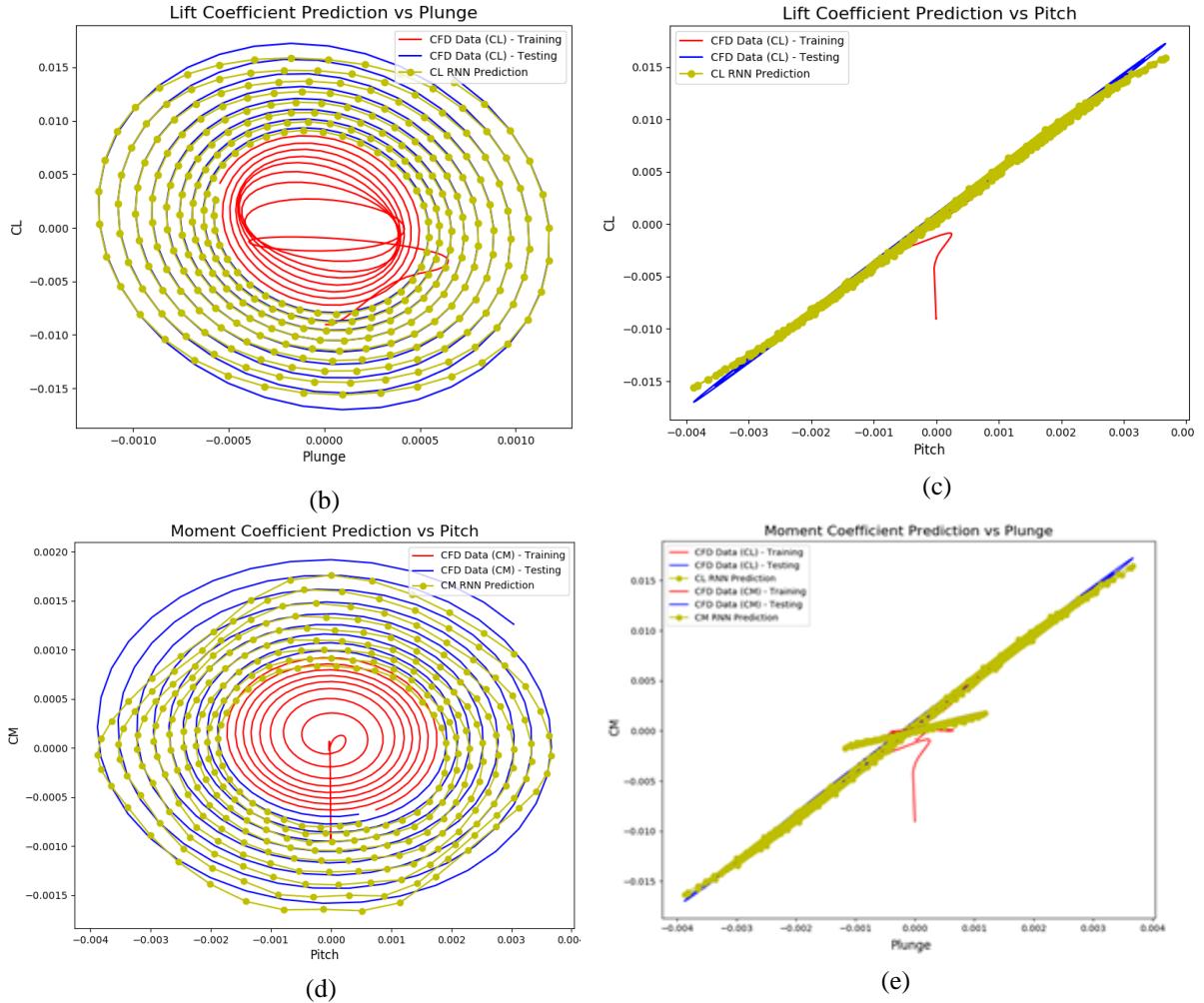
trained model when temporal sequences of displacement parameters  $h$  and  $\alpha$  and  $C_L$  and  $C_M$  from the testing data set are fed into the network as shown as blue and purple color respectively in Fig. 5.29 (a). The predicted  $C_L$  and  $C_M$  from the RNN, represented by yellow dots, are overlaid on the ground truth CFD data for comparison. The predicted variation of the aerodynamic lift and moment coefficients vs. plunging and pitching amplitudes are compared against CFD data in Fig. 5.29 (b) – (e).



**Figure 5.28:** (a) Plunging data divided into test-train sets for free-aeroelastic motion of NACA0012 airfoil, and (b) pitching data divided into test-train sets for the same motion



**Figure 5.29:** (a) Airfoil displacements and lift and moment coefficients as training data inputs, testing targets, and surrogate model prediction outputs, (Continued)



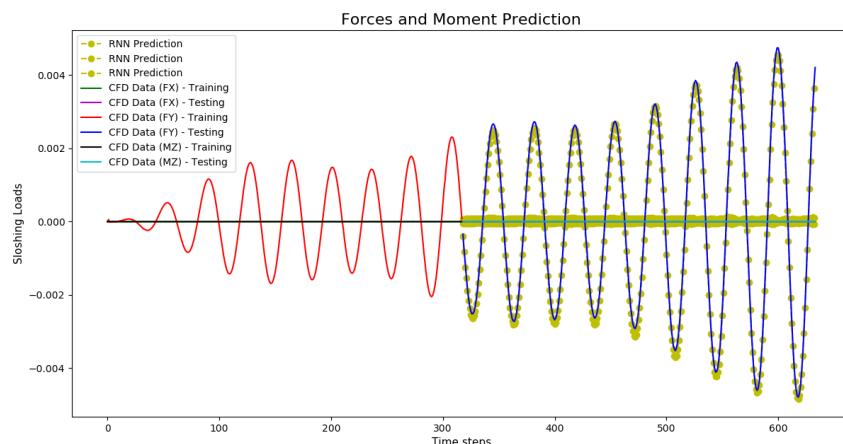
**Figure 5.29:** (b) phase plot of comparison of lift coefficient against plunge, (c) phase plot comparison of moment coefficient against pitch (d) phase plot comparison of moment coefficient against pitch, and (e) phase plot comparison of moment coefficient against plunge

The surrogate model for aeroelastic load prediction based on motion parameters,  $h$  and  $\alpha$ , predicts the lift and moment coefficients for testing data set to an acceptable accuracy. The model slightly underpredicts the lift coefficient after the airfoil goes into violent pitching motion around the 500<sup>th</sup> time step as apparent from Fig. 5.29 (a). This is because the model is trained only using the first half of the aeroelastic motion time series and it had not encountered higher values of pitching and plunging data to accurately predict corresponding lift and moment coefficients. The variation of  $C_L$  and  $C_M$  with  $h$  and  $\alpha$  is shown in Fig. 5.29 (b) - (e) which clearly shows the underprediction of aerodynamic load coefficients as the aeroelastic motion

diverges. Such inaccuracies in prediction can be eliminated by using a richer training dataset containing dynamics of aeroelastic motion or finely tuning the hyperparameters of the RNN.

**(b) Sloshing Loads Prediction:** The partially filled fuel tank moves in synchronization with the airfoil and hence the sloshing loads are computed using structural inputs provided by the aeroelastic equations. The current state of the fluid in the tank also depends on the previous states, which are represented by temporal sequences of loads due to fluid sloshing on the tank walls. The complete time series of the sloshing loads can be generated by appending one-step predictions from the predictive model over the length of the complete time series.

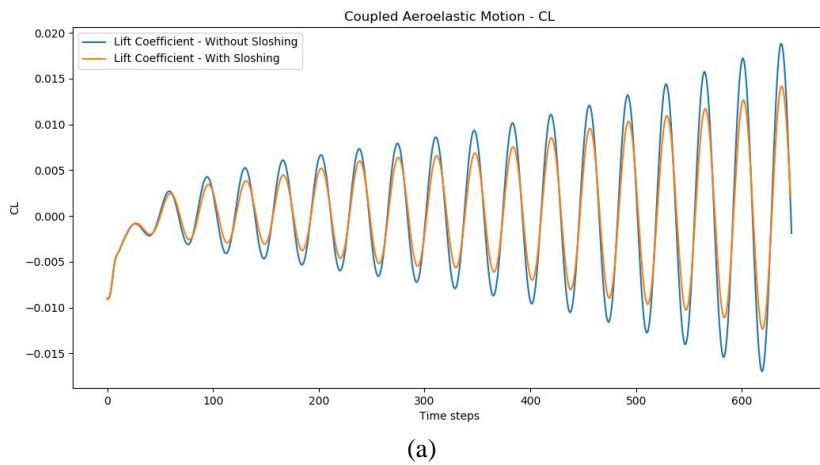
The tank is excited with structural motion obtained from free motion of NACA0012 airfoil experienced after forcing the airfoil for 2 cycles of pitching motion and then letting it move freely with freestream Mach number,  $M_\infty$ , of 0.60 and speed index,  $V_f$ , of 0.425. The free motion of the airfoil is the same as in full time series generation of aeroelastic case, shown in Fig. 5.28 (a) - (b). The time series is divided into training and testing data with a 1:1 ratio, the latter part being utilized for testing. The RNN architecture is the same as chosen in Section 4.3 of this thesis and shown in Fig. 4.9 (b). The corresponding horizontal and vertical forces on tank walls, and the corresponding moments are the outputs of the predictive model, are shown in Fig. 5.30.



**Figure 5.30: Sloshing loads due to tank motion in the form of forces and moments as training data inputs, testing targets and surrogate model prediction outputs**

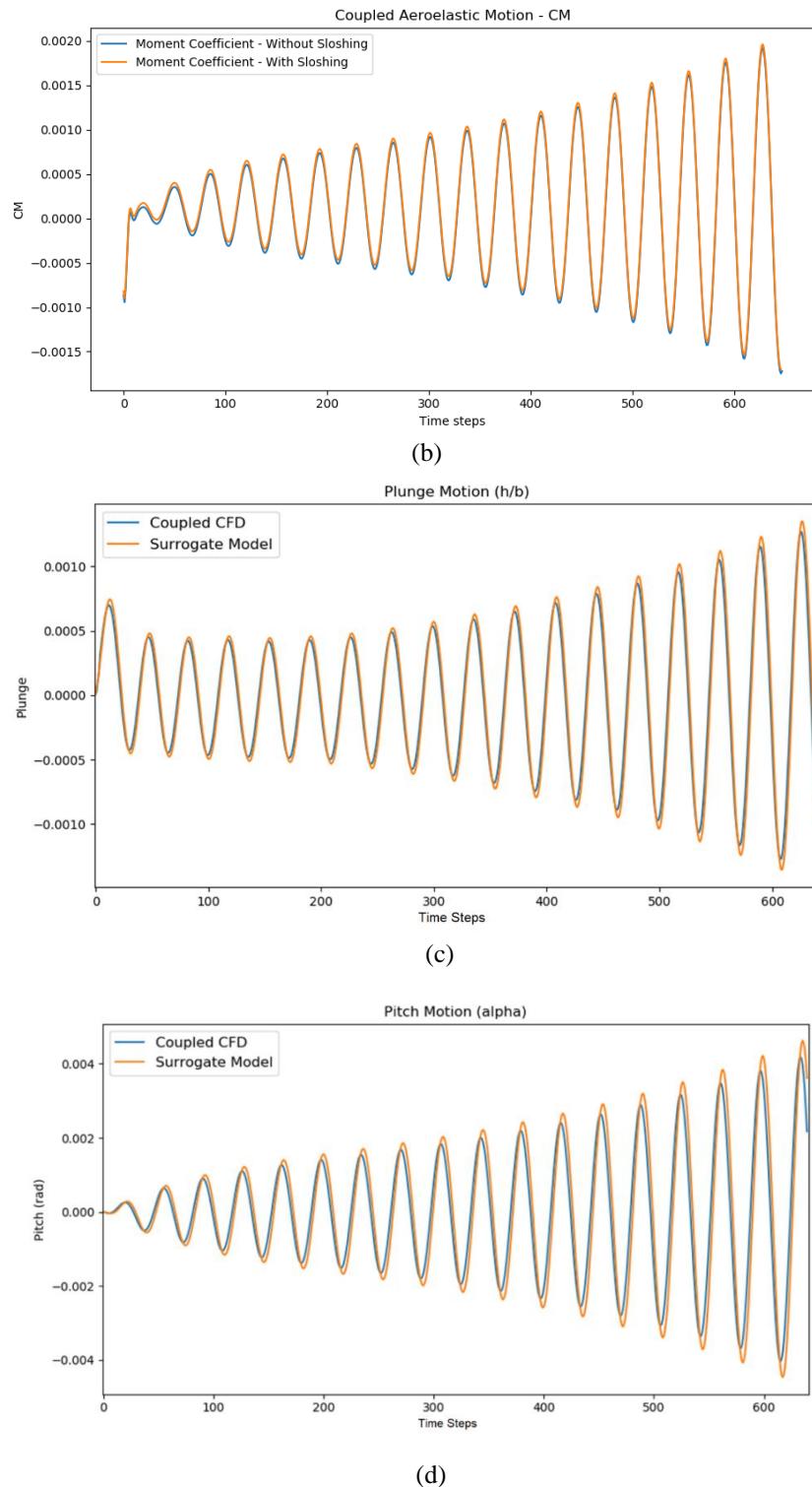
The surrogate model for sloshing loads prediction based on  $h$  and  $\alpha$ , predicts the vertical and horizontal forces and moment for testing data set to an acceptable accuracy. The RNN predictions represented by yellow dots in Fig. 5.30 are superimposed on the CFD data used for testing for all outputs, namely  $F_x$ ,  $F_y$  and  $M_z$ . The model accuracy can be improved arbitrarily by tuning the hyperparameters of the RNN.

**(c) Coupled Aeroelastic Motion with Sloshing Tank:** The aerodynamic and sloshing loads on the wing section are predicted by the surrogate model and the resulting aeroelastic motion is computed using the aeroelastic equation derived in Section 2.3 of this thesis given in Eqn. 2.73. The computed structural motion is fed back into the surrogate models to generate loads for the next time step, as elaborated in the flowchart in Section 4.4 of this thesis and shown in Fig. 4.13. The effect of sloshing on effective lift and moment coefficient of the coupled aero-structural system is depicted in Fig. 5.31 (a) - (b), respectively. The structural motion of the airfoil as computed using predictive surrogate models is compared with the structural motion computed using CFD in Fig. 5.31 (c) - (d).



(a)

**Figure 5.31: (a) The effective lift obtained by coupling sloshing tank with the aeroelastic wing section using RNN based predictive model, (Continued)**



**Figure 5.31: (b) Corresponding moment coefficient, (c) Comparison of plunge motion of airfoil computed from surrogate model and high-fidelity CFD, and (d) Comparison of pitching motion of airfoil computed from surrogate model and high-fidelity CFD**

There is a significantly greater effect of sloshing on the combined vertical loads due to aerodynamics and sloshing than normal moment. This is attributed to the fact that the center of

gravity of sloshing fluid does not move enough to cause effective moment on the elastic axis of the wing section. The structural displacements computed from the surrogate model and CFD are in good agreement for the initial free motion of the airfoil. The surrogate model slightly over-predicts the plunging and pitching motion when compared with CFD data. This mismatch can be attributed to the fact that training data for surrogate models do not contain larger airfoil displacements and hence lower prediction accuracy in that region.

**(d) Computational Cost Analysis of High-Fidelity and Surrogate Models:** The computational cost is measured in terms of CPU time of simulation running on an *Intel® Xeon® CPU E5-1650 V3 @ 3.50 GHz* processor running on single core and is summarized in Table 5.1.

**Table 5.1: Computational Cost Analysis of High Fidelity CFD and Surrogate Predictive Model**

	Time (in seconds)	Total Time (in seconds)
<b>CFD Simulations</b>		
Aeroelastic	558.01 (forced) + 1875.30 (free)	3895.76
Sloshing	1462.43	
<b>Surrogate Predictive Model</b>		
Aeroelastic	67.41	147.47
Sloshing	80.06	
<b>Computational Savings</b>	$(3895.75 - 147.46) / 3895.75 * 100 = 96.21\%$	

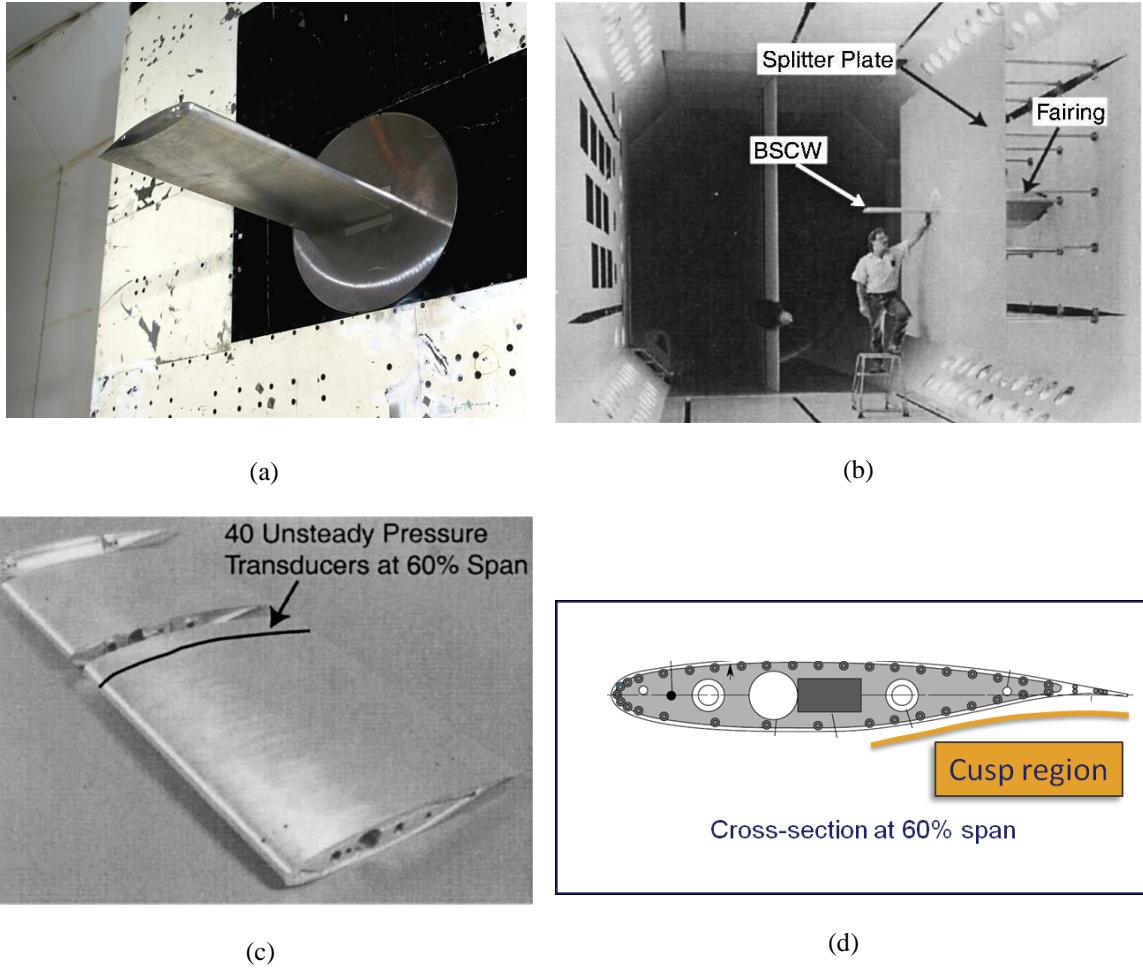
The RNN-based surrogate models for predicting aero-structural motion and sloshing loads in a partially filled fuel tank rigidly attached to the moving airfoil require one-time training using data obtained from CFD simulations. The computational time required to generate the training data set is not considered for the calculation of computational savings. The surrogate model developed in this work can be trained using rich data set containing multiple flow conditions and can be used to predict aero-structural loads for a wider variety of motions, flow conditions, and different fill-levels of fuel tank, thus further enhancing the utility and computational efficiency of surrogate model over high-fidelity CFD simulations.

## **5.4 Sloshing Effects on Aeroelastic Motion of NASA BSCW**

The present study focuses on the development of a computational framework to study the aeroelastic motion of NASA Benchmark Supercritical Wing embedded with a partially filled fuel tank with sloshing fuel. The aim of the surrogate model is to economically and accurately predict the aeroelastic motion of the wing with an embedded fuel tank exposed to fluid loadings of external flow-field as well as internal sloshing. A recurrent neural network (RNN) architecture is chosen to efficiently learn the dominant static and dynamic characteristics of the underlying physical systems using a limited set of training data rich in nonlinear dynamics generated from high fidelity CFD solvers. The computational framework is capable of coupling aerodynamic and sloshing loads in runtime to predict the coupled aero-structural motion of the system as developed for the previous study of NACA0012. The coupled motion is compared with the free motion of the wing with an empty fuel tank to study the effects of sloshing on aeroelastic characteristics.

### **A. Validation of Inviscid Compressible Solver for NASA Benchmark Supercritical Wing**

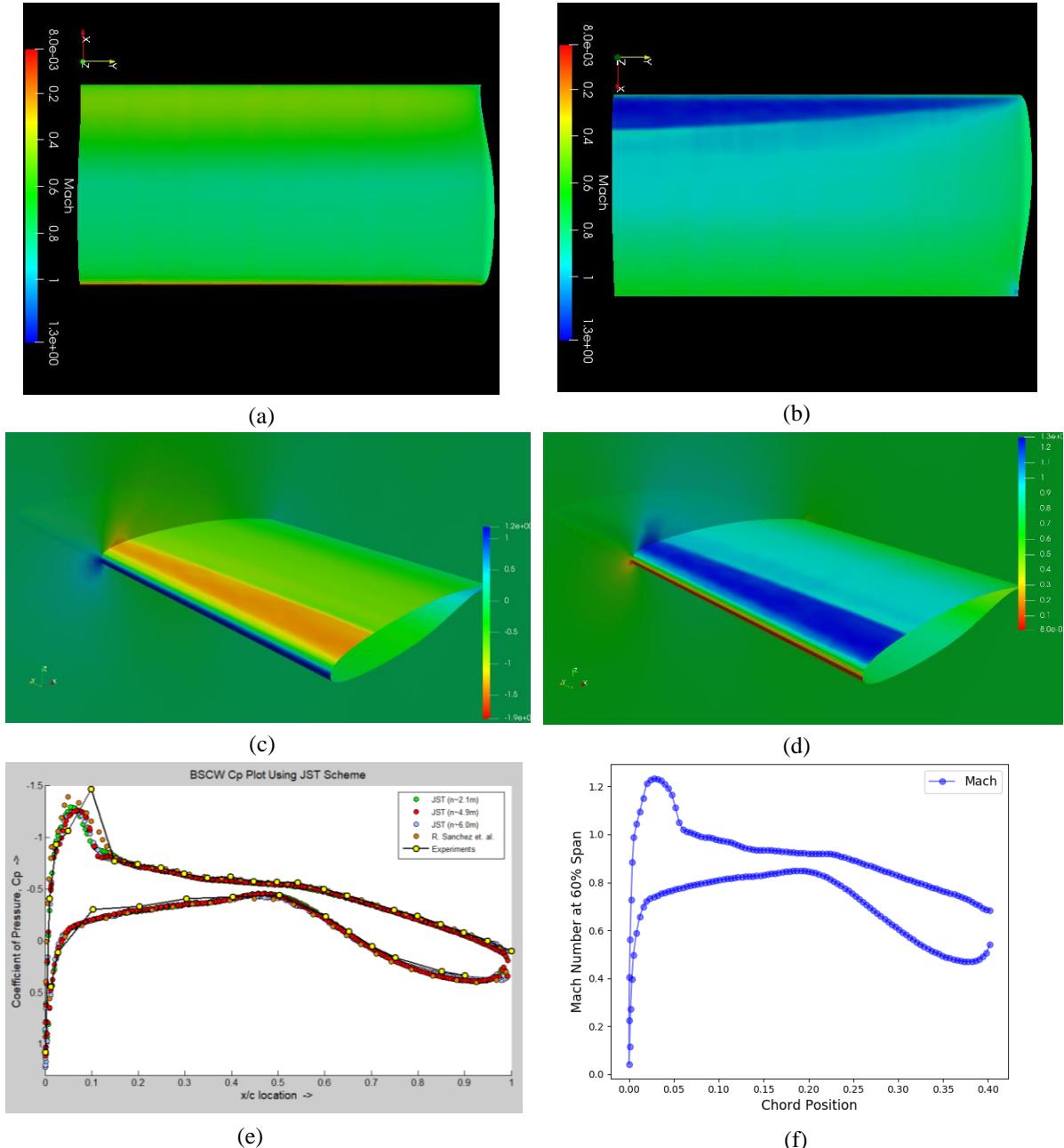
The NASA BSCW is installed at NASA Langley Research Centre's Transonic Dynamic Tunnel as shown in Fig. 5.32 (a) – (b). Two testing rigs are available to collect experimental data: Oscillating Turn-table (OTT), and Plunging and Pitching Apparatus (PAPA). There are 40 pressure transducers available at 60% span length of the wing as shown in Fig. 5.32 (c) – (d). The test rigs allow wing movement in the plunging and pitching direction and the data is collected for both steady and unsteady flows. The wing structure is considered rigid and hence the 2-DOF aeroelastic equation of motion is applicable for this configuration.



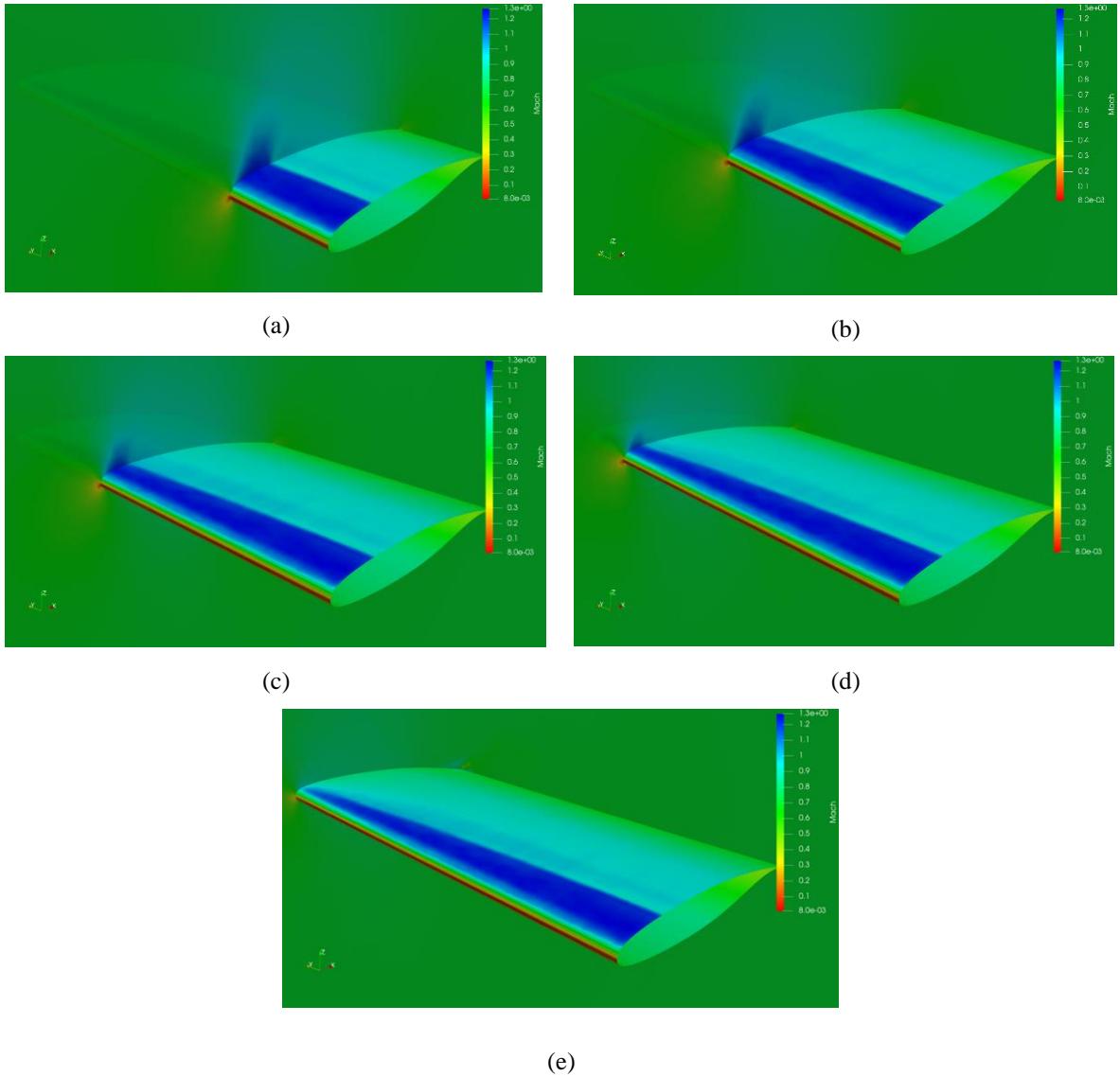
**Figure 5.32:** (a) NASA Benchmark Supercritical Wing as 2-DOF plunge and pitch wing, (b) BSCW in wind tunnel, (c) Pressure transducers at 60% span length of the wing, and (d) Cross-section at 60% span length

The high-fidelity solutions computed using *SU2* are validated for steady and unsteady computations against experimental data provided by the *2<sup>nd</sup> AIAA Aeroelastic Prediction Workshop* [90, 91] and numerical data by Sanchez [69]. Steady inviscid transonic flowfield around BSCW immersed in free stream Mach number,  $M_\infty$ , of 0.70 and at an angle of attack,  $\alpha$ , of 3° is computed. The Mach number contours on the upper and lower surface are shown in Fig. 5.33 (a) and (b), respectively. The  $c_p$  contours and Mach number contours are shown in Fig. 5.33 (c) and (d), respectively, at 60% span length and wing surface, followed by  $c_p$  distribution and Mach number distribution at the same location in Fig. 5.33 (e) and (f), respectively. The comparison and validation of steady-state results with experimental data and

numerical results is shown in Fig. 5.33 (e). As apparent from Fig. 5.33 (e), the  $c_p$  distribution obtained in this study closely resembles the experimental and numerical data shown by yellow and orange dots, respectively. Fig. 5.34 (a) – (e) shows the steady-state shock pocket by showing the Mach number contours at different span locations on the wing.

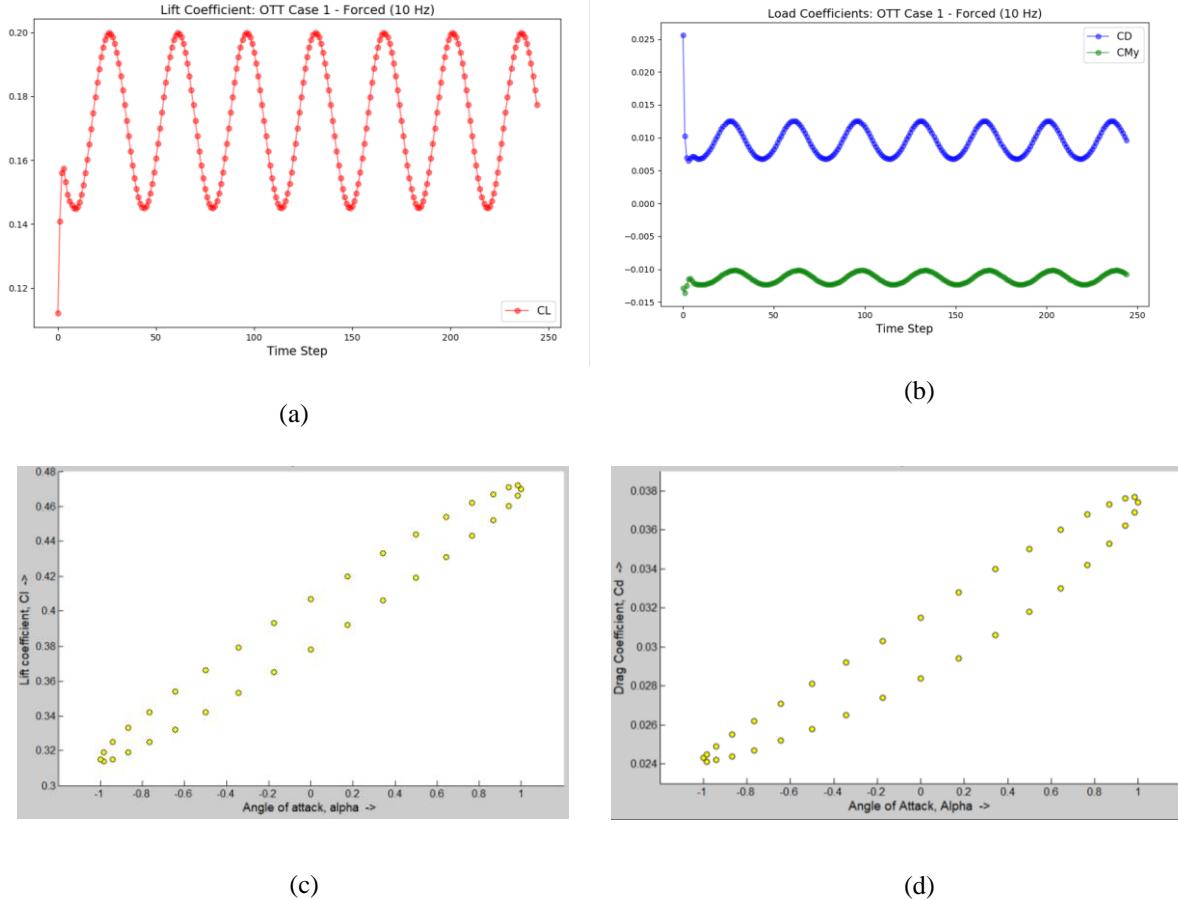


**Figure 5.33:** (a) Mach number contours on upper surface of BSCW, (b) Mach number contours on lower surface, (c) Coefficient of pressure contours on wing surface and 60% span, (d) Mach number contours on wing surface and 60% span, (e) Comparison of  $c_p$  at 60% span with numerical and experimental data, and (f) Mach number distribution at 60% span length



**Figure 5.34: Steady-state Mach number contours on BSCW surface and cross-section at span length (a) 20%, (b) 40%, (c) 60%, (d) 80%, and (e) 100%**

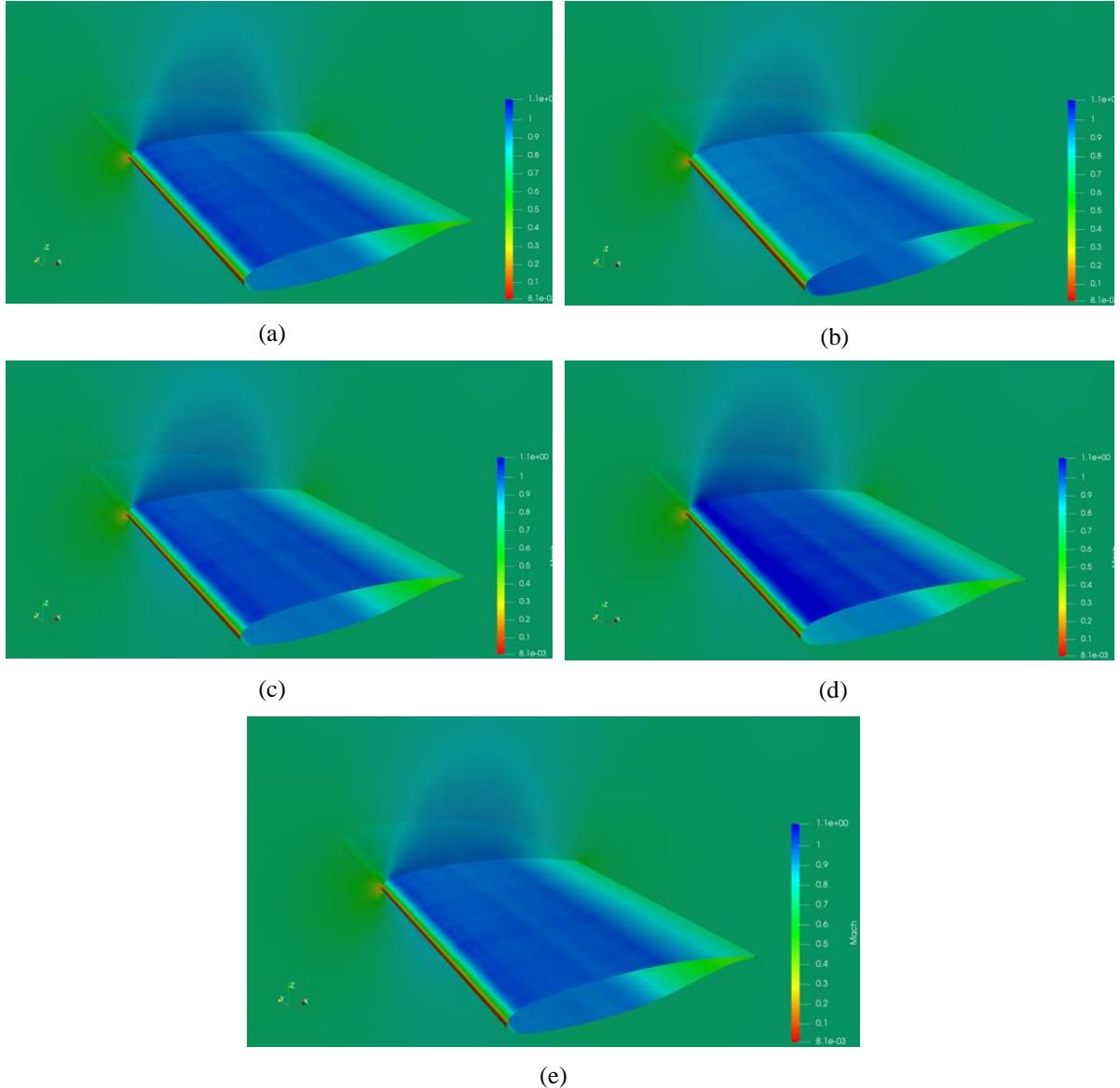
For unsteady validations, BSCW is immersed in a flowfield of free-stream Mach number,  $M_\infty$ , of 0.70 and then from steady-state it is forced to pitch according to  $\alpha = \alpha_{mean} + \alpha_0 \sin(\bar{f}t)$  where  $\alpha_{mean}$  is  $3^\circ$ ,  $\alpha_0$  is  $1^\circ$  and the forcing frequency,  $f$ , is 10 Hz. The time history of lift coefficient,  $C_L$ , and drag and moment coefficients,  $C_D$  and  $C_M$ , are plotted in Fig. 5.35 (a) and (b), respectively. The variations of the same are plotted against the angle of attack about its mean position in Fig. 5.35 (c) and (d).



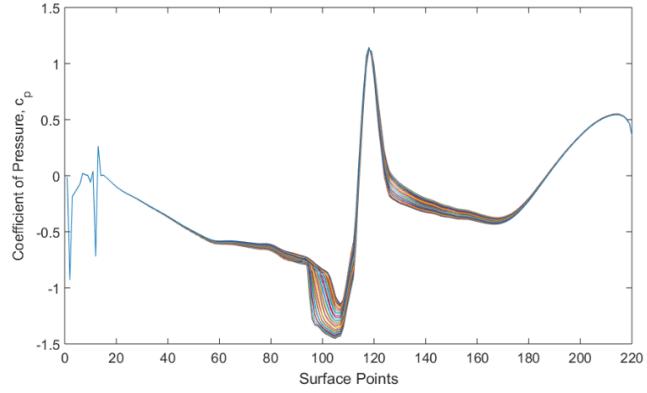
**Figure 5.35: (a) Lift coefficient, and (b) Drag coefficient and moment coefficient for BSCW OTT forced pitching at 10 Hz, (c) Lift coefficient vs angle of attack for forced pitching motion, and (d) Drag coefficient vs angle of attack for forced pitching motion**

The flowfield for the OTT forced oscillations about the mean angle of attack is shown in Fig. 5.36 (a) – (e) in form of Mach contours on the wing surface, as well as at 60% span length of the wing, which is a critical location for this wing for solution validation. An oscillating shock can be observed upon closer examination of Mach contours. The unsteady flow solution is verified by plotting the frequency response function (FRF) of  $c_p$  distribution for oscillating wing (OTT forced oscillations at 10 Hz). The 60% span wing section is divided into 220 points in total for both the upper and lower surface. The  $c_p$  distribution of 1 forced oscillation at 10 Hz is plotted, each line representing one instance of time, and the x-axis represents grid points on the section spatially, as shown in Fig. 5.37 (a). The FRF is computed casting the temporal response of  $c_p$  against the angular position of the airfoil into a Fourier series for one converged

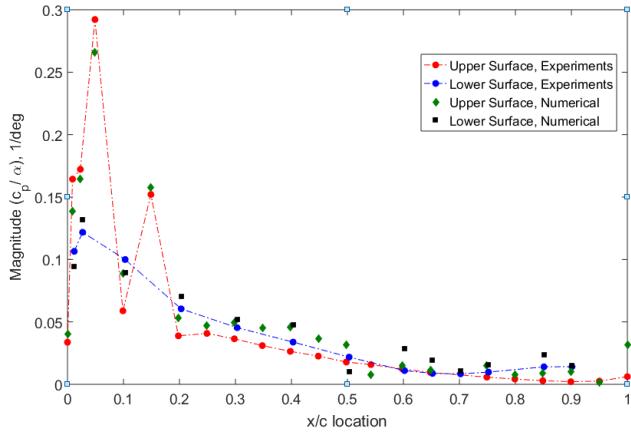
cycle of motion of BSCW. The magnitude and phase response of real and imaginary parts are validated against the experimental results [90] in Fig. 5.37 (b) – (c).



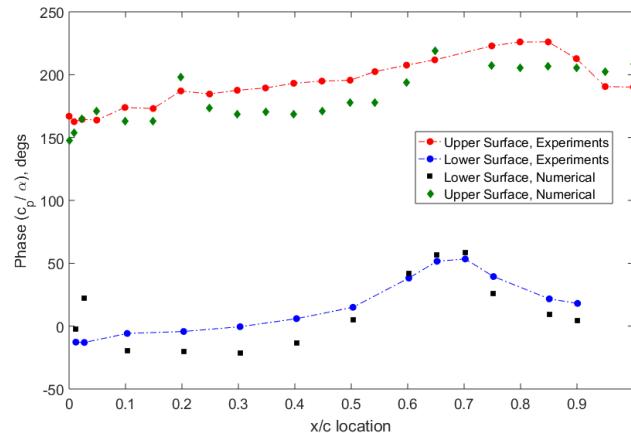
**Figure 5.36: Mach number contours for BSCW OTT forced pitching at 10 Hz at (a) 0 deg, (b) 90 deg, (c) 180 deg, (d) 270 deg, and (e) 360 deg**



(a)



(b)



(c)

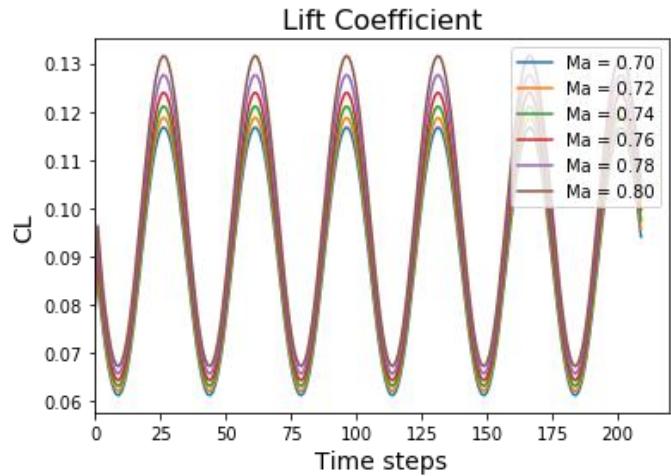
**Figure 5.37: (a)  $c_p$  vs spatial grids at 60% span of BSCW for 1 cycle of motion divided into 36 steps, (b) FRF Magnitude OTT at  $M_\infty = 0.70$ , and (c) FRF Phase OTT at  $M_\infty = 0.70$**

The FRF magnitude and phase are computed at locations where the experimental results are available. The agreement between the numerical and experimental data is within an acceptable range.

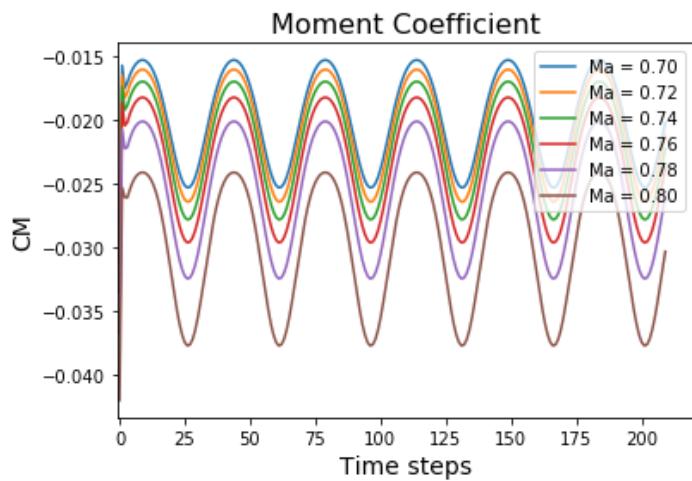
The validations for CFD solver for sloshing is not provided in this study as validation given in Section 5.3 B of this chapter holds true. Similarly, the RNN based surrogate model used in this study has the same architecture and working as the model used in Section 5.3 E and Section 5.3 F of this chapter and is not required to be demonstrated here.

## **B. RNN Surrogate Model Validation for External Flow with Parametric Variation**

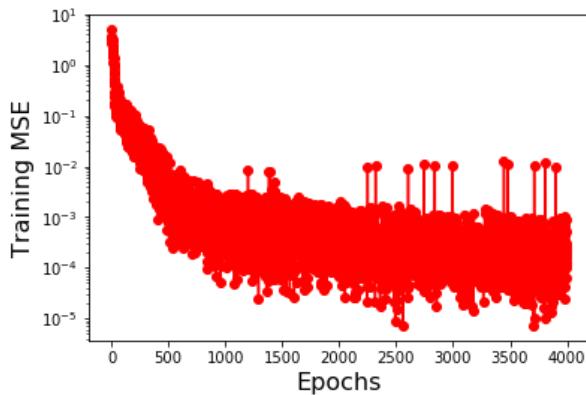
The external flow surrogate model for BSCW is required to predict  $C_L$  and  $C_M$  at the current time step given the information of the  $h$  and  $\alpha$ , as well as the aerodynamic loads and structural displacements information from previous time steps, along with the free-stream Mach number. Hence, the surrogate model can accommodate parametric flow variation in prediction of aerodynamic load coefficients as detailed in Section 4.3 and in Fig. 4.10 of this thesis. The training data for testing and validation of the model is generated by immersing BSCW in flow-field with different free-stream Mach numbers. The wing is harmonically pitched and load coefficients corresponding to the motion are obtained. The datasets consist of time series of structural motions, corresponding load coefficients, and free-stream Mach number is utilized for training. The RNN consists of three hidden layers containing 500, 400 and 300 neurons respectively and the data is fed with a batch length of 30. The aerodynamic lift and moment coefficients obtained corresponding to free-stream Mach numbers,  $M_\infty$  of 0.70, 0.72, 0.74, 0.76, 0.78 and 0.80 are utilized for training the RNN and are shown in Fig. 5.38 (a) – (b). Figure 5.38 (c) and Fig. 5.38 (d) show the convergence of weights and biases for the present training instance in the form of a reduction in  $mse$ , which converges to the order  $O(10^{-4})$  after 4000 epochs, as apparent from Fig. 5.38 (c) - (d). As apparent from Fig. 5.38 (a) - (b), the numerical values of lift and moment coefficients are not in the same order. Hence, before training the RNN, the load coefficients are scaled so that all numerical values are in the same order and the network can be efficiently trained.



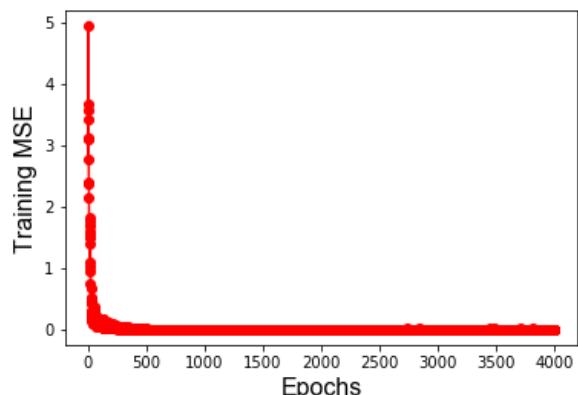
(a)



(b)



(c)

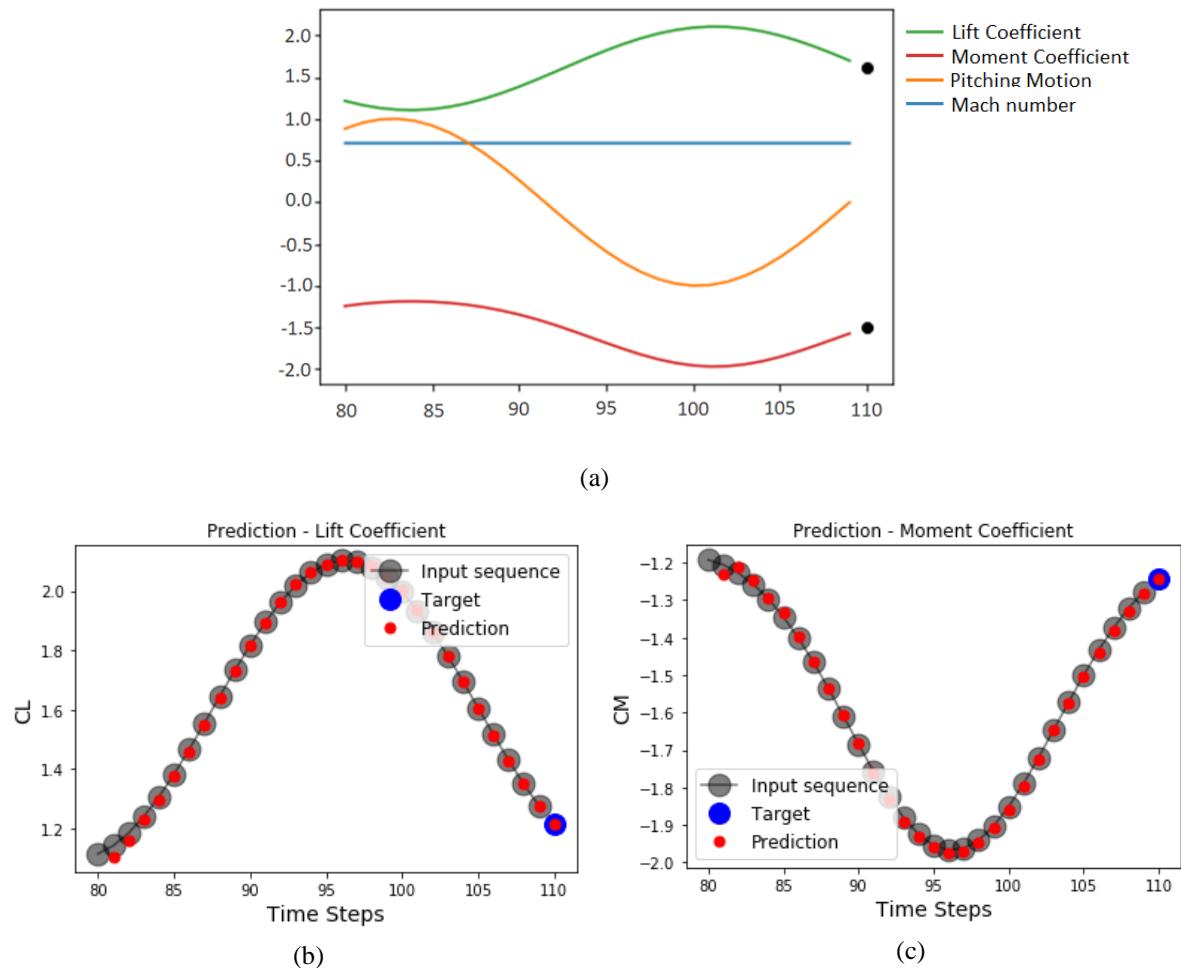


(d)

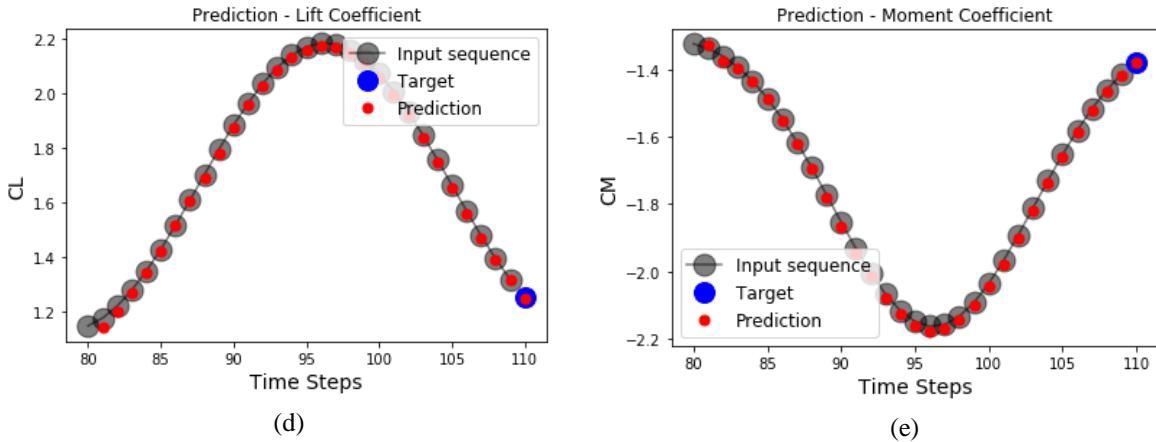
**Figure 5.38:** (a) Lift coefficient of BSCW for pitching motion, (b) Moment coefficient of BSCW for pitching motion; Variation of (c) MSE, and (d) MSE (in log scale) vs. epochs during network training for aerodynamic data

The prediction accuracy of RNN is tested with a single time step prediction. The RNN is with sequences of free-stream Mach number, structural motion, and corresponding  $C_L$  and  $C_M$  values

at free-stream Mach number of 0.70 as shown in Fig. 5.39 (a). The black dots in Fig. 5.39 (a) indicates the  $C_L$  and  $C_M$  values to be predicted by the surrogate model. The single-step predictions of RNN is shown in Fig. 5.39 (b) – (c). Since the network was trained with data corresponding to Mach number 0.70, it is worthwhile to test the prediction capability of the network to unseen inputs. The network is tested with the motion corresponding to free-stream Mach number of 0.73, and the prediction comparison is shown in Fig. 5.39 (d) – (e).

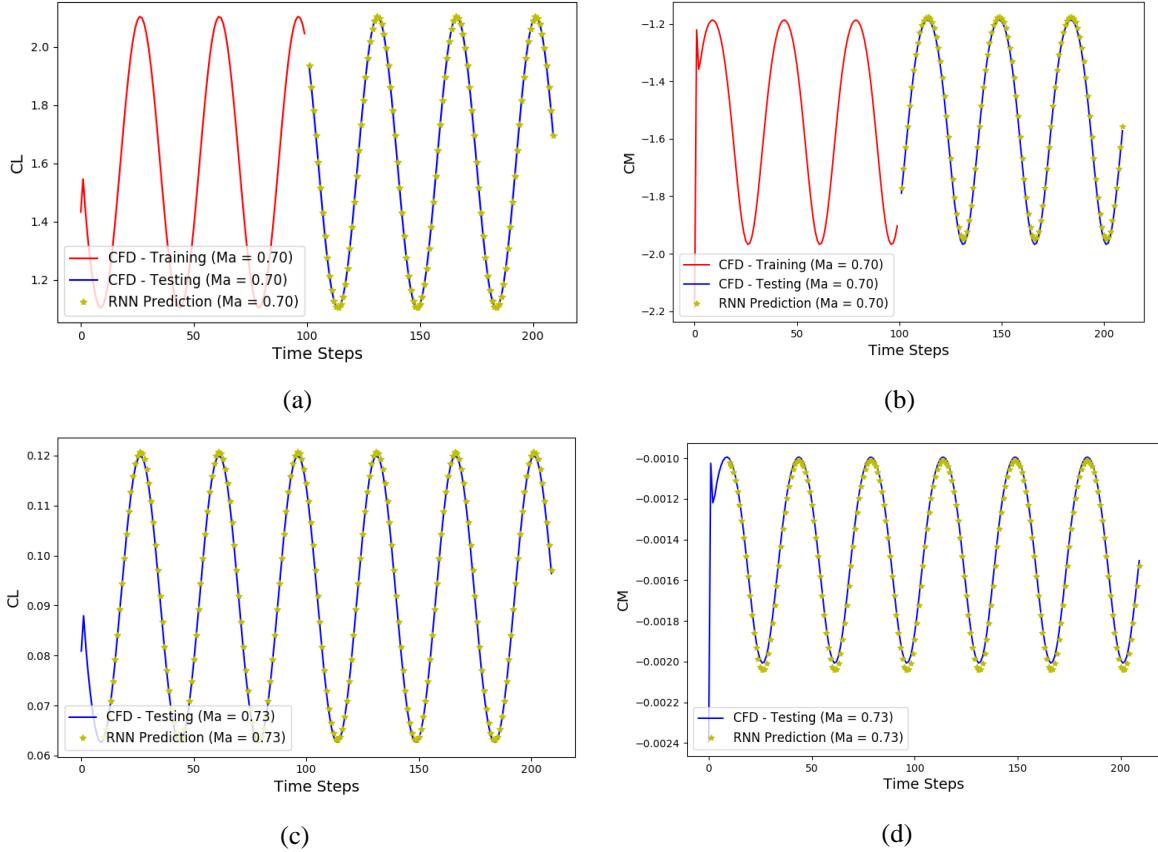


**Figure 5.39:** (a) Sample data fed to RNN at Mach 0.70, (b) The training instance, the expected output and the network prediction for lift coefficient at  $M_\infty = 0.70$ , (c) The training instance, the expected output and the network prediction for moment coefficient at  $M_\infty = 0.70$ , (Continued)



**Fig. 5.39: (d) The training instance, the expected output and the network prediction for lift coefficient at  $M_\infty = 0.73$ , and (e) The training instance, the expected output and the network prediction for moment coefficient at  $M_\infty = 0.73$**

The target for prediction at the 110<sup>th</sup> time step is shown in blue, the input sequence, which is also the time history of load coefficients is shown in grey and the actual network prediction is shown with red. It should be noted that the RNN is developed to predict the complete time series as shown with red dots. However, only the last point, in this case, the 110<sup>th</sup> time step is of relevance. The preceding predicted value in Fig. 5.39 (b) – (e) is shown for the purpose of visualization only. The network prediction seems very accurate for both training data and unseen new input sequence fed to the network. As done in previous sections, a longer time series prediction of  $C_L$  and  $C_M$  by the surrogate model for Mach number 0.70 is shown in Fig. 5.40 (a) – (b) and for unseen Mach number 0.73, as shown in Fig. 5.40 (c) – (d). The RNN prediction aerodynamic loads for a longer time series agrees very well with the CFD data for both Mach number flows. Slight deviations from CFD data can be noticed in RNN prediction of moment coefficient  $C_M$  with free-stream Mach number of 0.73. This deviation, however small, can still be improved by training the RNN with more samples of intermittent Mach numbers. However, this increases the computational cost of the framework. The RNN can be fine-tuned by playing with the hyperparameters of the network. However, for the present study, the deviations seen in Fig. 5.40 (d) are acceptable.



**Figure 5.40: Training data, testing data and RNN prediction of (a) Lift coefficient at  $M_\infty = 0.70$ , (b) Moment coefficient at  $M_\infty = 0.70$ ; Testing data and RNN prediction of (c) Lift coefficient at  $M_\infty = 0.73$ , and (d) Moment coefficient at  $M_\infty = 0.73$**

The validation and testing of RNN surrogates for the sloshing model is omitted since the same RNN architecture developed and tested in Section 5.3 E of this chapter is used in the present study.

### C. Prediction of Coupled Aeroelastic Motion of BSCW with Sloshing Tank

The RNN based surrogate models for external flow and internal sloshing are coupled using the aeroelastic equation of motion as detailed in the previous study. The free aeroelastic motion of BSCW without sloshing is compared to motion with a half-filled fuel tank. BSCW is immersed in free stream flow with Mach number,  $M_\infty$ , of 0.72. The wing is set free to plunge and pitch after two cycles of forced pitching. During aeroelastic motion, the wing retains its shape and moves as a rigid body. The structural parameters of BSCW provided by the *Aeroelastic Prediction Workshop* are given in Table 5.2.

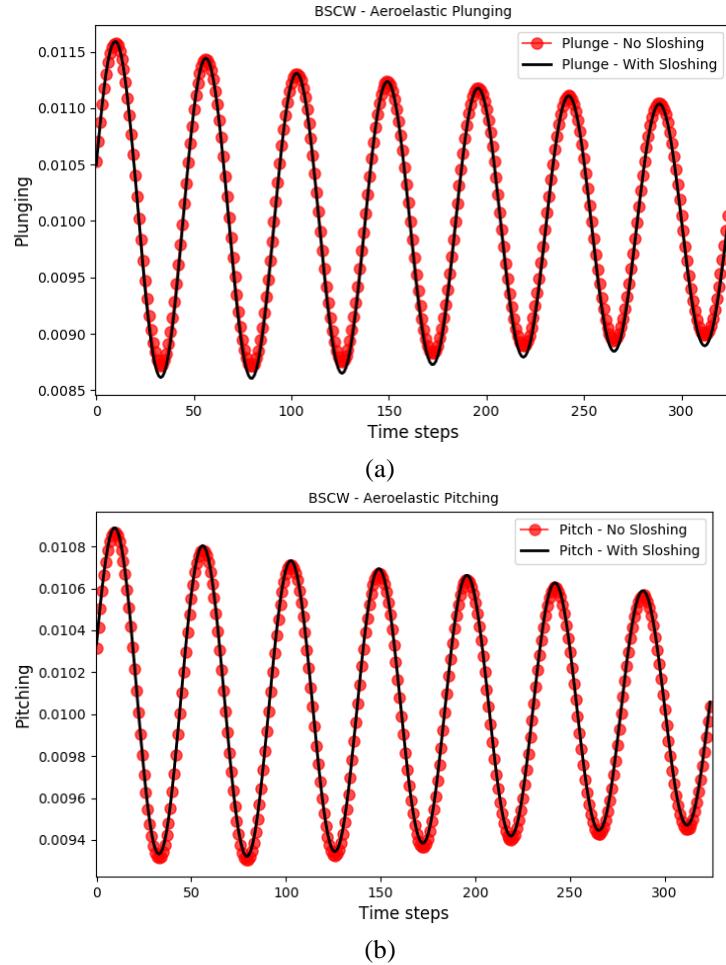
**Table 5.2: Structural Properties of NASA Benchmark Supercritical Wing**

<b>chord, c</b>	406.4 mm	<b>Moment of Inertia, <math>I_\alpha</math></b>	3.76510 kg-m <sup>2</sup>
<b>span, s</b>	812.8 mm	<b>Radius of Gyration, <math>r_\alpha</math></b>	0.4591 m
<b>Area, A</b>	330322 mm <sup>2</sup>	<b>Mass, m</b>	87.913671 kg
<b>Pitching Frequency, <math>f_\alpha</math></b>	3.33 Hz	<b>Pitching Stiffness, <math>k_\alpha</math></b>	4018.64 kg-m <sup>2</sup> /s <sup>2</sup> -rad <sup>2</sup>
<b>Plunging Frequency, <math>f_h</math></b>	5.20 Hz	<b>Plunging Stiffness, <math>k_h</math></b>	38484.12 kg/s <sup>2</sup>

The structural properties for this study are slightly modified to attain natural pitching and plunging frequencies of 1 Hz. This is done to attain sloshing characteristics similar to those obtained in the previous study in Section 5.3 of this chapter. The spring stiffness of plunging and pitching are modified as per the following relations:

$$k_h = m\omega_h^2 \quad \text{and} \quad k_\alpha = I_\alpha \omega_\alpha^2 \quad (5.1)$$

Hence,  $k_h = 3469.35 \text{ kg/s}^2$  and  $k_\alpha = 148.6787 \text{ kg-m}^2/\text{s}^2\text{-rad}^2$  is used for computation of aeroelastic motion of BSCW. The comparison between free aeroelastic motion of BSCW with and without the effects of sloshing is shown in Fig. 5.41 (a) – (b). A downward deflection in the plunging motion of the wing is observed when coupled with sloshing loads, as seen in Fig. 5.41 (a). The pitching motion is shown in Fig. 5.41 (b), however, remains nearly unaffected by sloshing. This can be attributed to two factors: small amplitude of wing motion that does not excite large fraction of fluid in the tank for convective sloshing, and high structural mass and inertia of the wing. Hence, the relatively feeble sloshing loads in the form of force and moment do not severely affect the wing motion.



**Figure 5.41: (a) Aeroelastic plunging motion of BSCW with and without sloshing at  $M_\infty = 0.72$ , and (b) Aeroelastic pitching motion of BSCW with and without sloshing at  $M_\infty = 0.72$**

#### D. Computational Cost Analysis of High Fidelity CFD and Surrogate Predictive Model

The prediction accuracy of RNN based surrogate models with parametric flow variation is demonstrated with aeroelastic motion of BSCW embedded with a half-filled fuel tank. The average time of CFD simulation of 5 cycles of pitching motion for both aerodynamics and sloshing solvers is considered for comparison. The computational time required to generate training data set for the surrogate model is not considered for calculation of computational savings. The computational cost is measured in terms of CPU time of simulation running on an *Intel® Xeon® CPU E5-1650 V3 @ 3.50 GHz* processor running on single core and is summarized in Table 5.3.

**Table 5.3: Computational Cost Analysis of High Fidelity CFD and Surrogate Predictive Model**

	Time (in seconds)	Total Time (in seconds)
<b>CFD Simulations</b>		
Aeroelastic	94371.68	119236.63
Sloshing	21864.95	
<b>Surrogate Predictive Model</b>		
Aeroelastic	667.53	1042.15
Sloshing	149.62	
<b>Computational Savings</b>	$(119236.63 - 1042.15) / 119236.63 * 100 = 99.12\%$	

The computational advantage of using RNN based surrogate models is exceptionally high. The prime reason is the high computational cost of aeroelastic motion solution using CFD with grid deformation. Furthermore, the computational time required to generate the training data set is not considered for the calculation of computational savings. The developed computational framework is robust to flow parameter variation and hence is the high computational advantage can be justified.

## 5.5 Physics Informed Neural Network for Aeroelastic Equation of Motion

A physics informed neural network model is attempted to replace the aeroelastic equation of motion with a model that *learns* the physics of the underlying system. Although the machine learning based surrogate models developed in the previous sections perform accurate predictions as a low computational cost, the robustness and reliability of the performance of these models depend largely on the data utilized to train them. Their performance outside the training set may be acceptable in the near vicinity of training data space, but cannot be relied upon outside that domain. Hence, an attempt has been made to use PINN to replace aeroelastic equation of motion. NACA0012 airfoil without sloshing fuel is used for demonstration of the methodology. The airfoil is immersed in a flow with a Mach number of 0.60. The aeroelastic

parameters and computational meshes are the same as in Section 5.3 of this chapter, hence solver validation is not necessary.

Aeroelastic PINN is different from a traditional data-driven feedforward neural network in the sense that the model parameters such as weights and biases are learned by modifying the loss function to be minimized. The loss function is a composite of the *mean squared error* and a residue which is obtained by solving the aeroelastic equation of motion as detailed in Section 4.5 of this thesis and given in Eqn. 4.18 (a) – (d). Therefore, PINN contains the information of parameters contained in the equation of motion. The model, when fed with aeroelastic loads in the form of  $C_L$  and  $C_M$ , is supposed to predict structural displacements in the form of plunging,  $h$ , and pitching,  $\alpha$ , for a 2-DOF system.

#### **A. Aeroelastic Equation Parameters in Loss Function**

The structural parameters that constitute the element of aeroelastic equation of motion are embedded in the loss function of PINN. These parameters for NACA0012 airfoil are given in Table 5.4. These are the same parameters used in Section 5.3 of this chapter.

**Table 5.4: Structural Parameters of NACA0012**

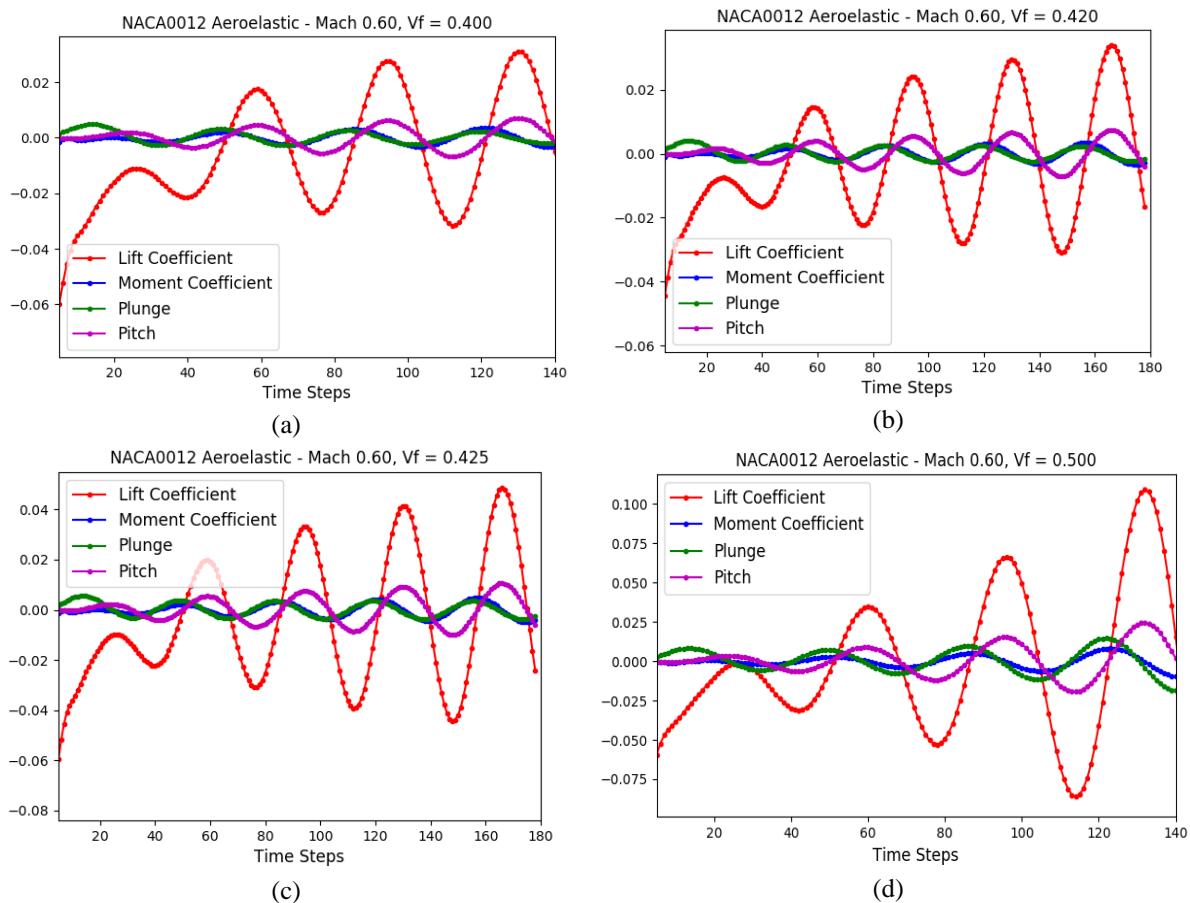
<b>Airfoil Mass Ratio, <math>\mu_{mass}</math></b>	75	<b>Plunge and Pitch Frequency, <math>\omega_h</math> &amp; <math>\omega_\alpha</math></b>	6.2832 rad/s
<b>CG Location, <math>x_\alpha</math></b>	0.25	<b>Radius of Gyration, <math>r_\alpha</math></b>	8.660

The loss function given in Eqn. 4.18 (a) – (d) is stated in condensed form in Eqn. 5.2.

$$Loss = mse + \sqrt{\left( (\ddot{h} + x_\alpha \ddot{\alpha}) + \left( \frac{\omega_h}{\omega_\alpha} \right)^2 h - C_L \right)^2 + \left( (x_\alpha \ddot{h} + r_\alpha^2 \ddot{\alpha}) + \alpha + 2C_M \right)^2} \quad (5.2)$$

## B. Data Samples for Training Structural PINN

The time derivatives of structural motions,  $\ddot{h}$  and  $\ddot{\alpha}$ , are computed using displacement information of previous time steps. The PINN utilized for the present study contains 5 hidden layers with 30, 40, 50, 40 and 30 neurons, respectively, with  $tanh$  as the activation function. The PINN is trained with aeroelastic motion profiles and corresponding load coefficients of NACA0012. Fig. 5.42 (a) – (d) shows some typical profiles utilized in the training process.

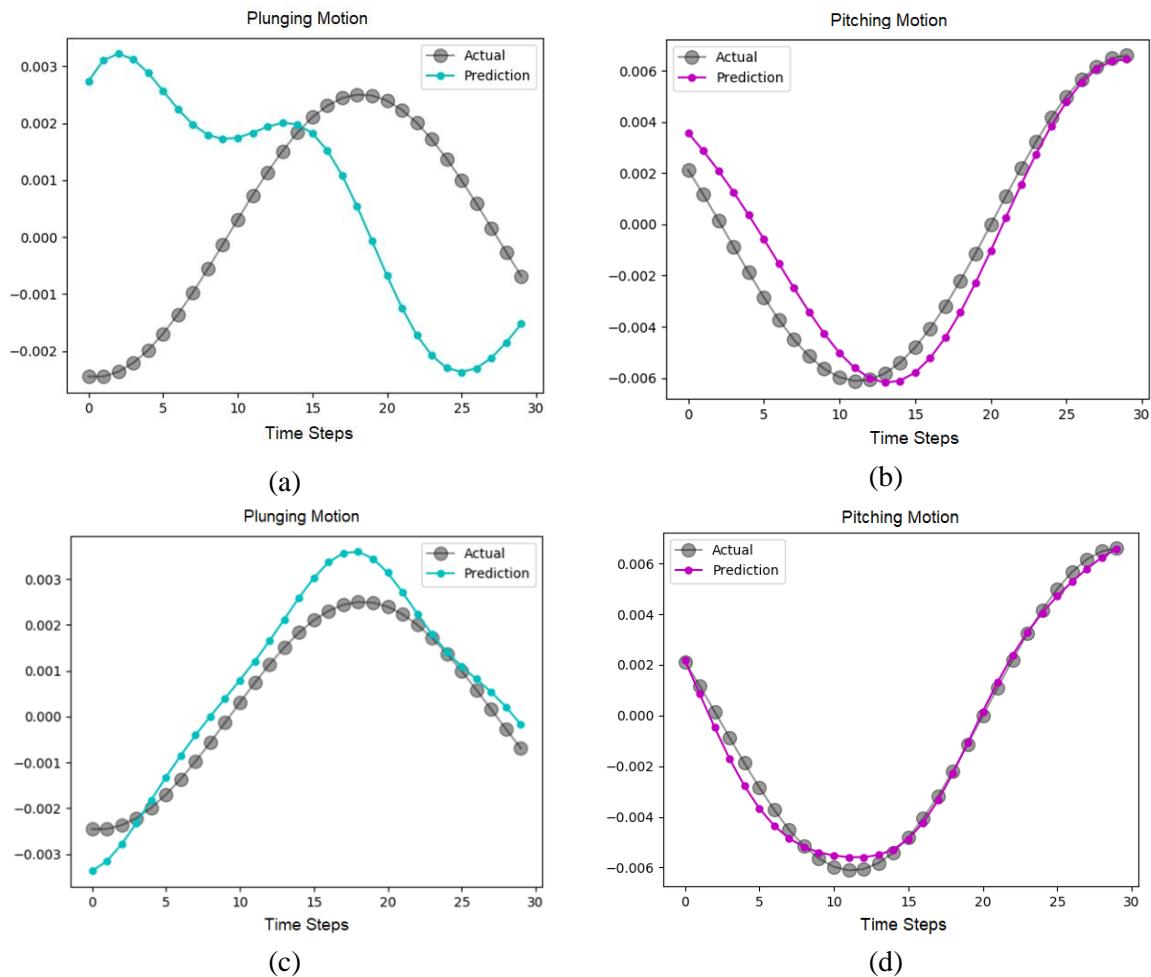


**Figure 5.42: Aeroelastic motion profiles NACA0012 for training structural PINN at  $M_\infty = 0.60$  and flutter speed index (a)  $V_f = 0.400$ , (b)  $V_f = 0.420$ , (c)  $V_f = 0.425$ , and (d)  $V_f = 0.500$**

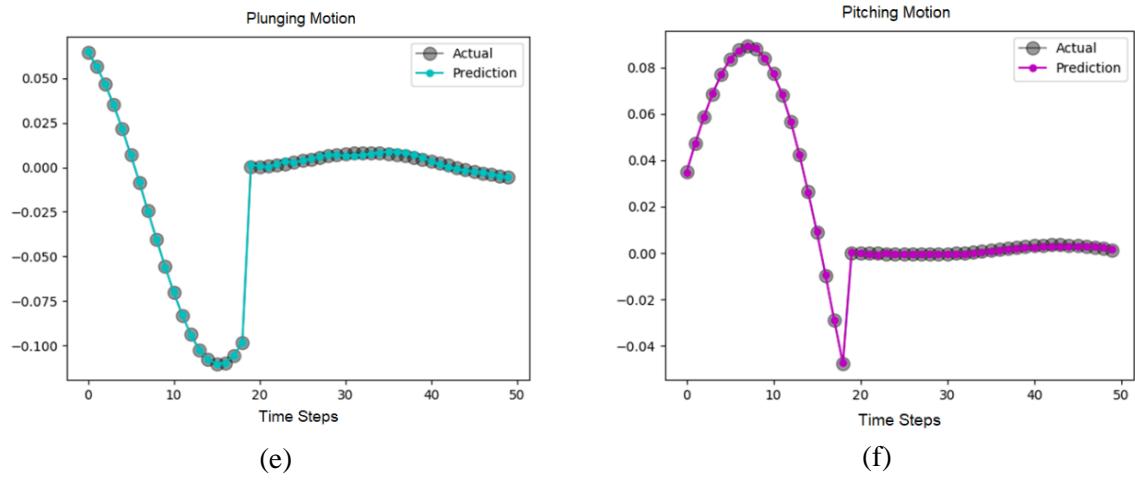
The information of flow parameters such as Mach number and flutter speed index is intrinsically encoded in the aerodynamic load coefficients  $C_L$  and  $C_M$ . All the motion data is appended in a single file and batches of data are fed to the neural network for training.

### C. Structural Displacements Prediction Results of PINN

The neural network is tested for prediction accuracy by feeding the aerodynamic load coefficients in batches. The PINN prediction is compared with data obtained from aeroelastic solver for plunging and pitching motions, as shown in Fig. 5.43 (a) – (f). PINN is tested with aeroelastic motion profile of NACA0012 immersed in a flow field with free stream Mach number,  $M_\infty$ , of 0.60 and flutter speed index,  $V_f$ , of 0.450.



**Figure 5.43: Comparison of aeroelastic motion of NACA0012 computed by CFD and predicted by PINN for (a) plunging motion, and (b) corresponding pitching motion; (c) plunging motion, and (d) corresponding pitching motion; (Continued)**



**Figure 5.43: (e) plunging motion, and (f) corresponding pitching motion**

Three sets of PINN predictions are shown in Fig. 5.43. The first set in Fig. 5.43 (a) – (b) shows poor agreement with aeroelastic solver data. The second set, Fig. 5.43 (c) – (d), shows better agreement and the last set in Fig. 5.43 (e) - (f) shows extremely good agreement with aeroelastic data. The poor prediction can be attributed to the presence of transients near the onset of motion in the training data shown in Fig. 5.42 (a) – (d). Fig. 5.43 (e) – (f) shows the transition between two aeroelastic motion profiles. However, PINN is able to capture the transition with high accuracy. With some fine-tuning of hyperparameters of the neural network and careful selection of training data sets, the prediction accuracy of PINN can be improved many folds.

## Chapter 6

### Conclusions and Future Works

This work addresses the lack of computational tools to study the effects of fuel sloshing on aeroelastic characteristics of airfoil and wing in transonic flights. Computational frameworks comprising of various models for aerodynamics and sloshing, ranging from high-fidelity CFD models, linear EMS model and data-driven machine learning surrogate models are explored. Firstly, a methodology is developed to integrate open-source CFD solvers *SU2* and *OpenFOAM* for solution of the coupled aeroelastic problem with sloshing in an embedded fuel tank. High-fidelity numerical solutions obtained from this method are considered ground truth for comparison and evaluation of other models and frameworks in this work.

Radial Basis Function Neural Network is utilized for the development of the surrogate model to simulate sloshing dynamics. The RBF-NN architecture is modified to incorporate temporal histories of previous inputs to the network, as well as, its previous outputs as network inputs, to account for the memory effect of dynamic sloshing. The loads' prediction by RBF-NN based surrogate model is represented as a time series problem, with structural motion of the tank as inputs and corresponding sloshing loads as outputs. Structural input for the generation of training data designed to excite the dynamics of fuel tank around the natural frequency of NACA64A010, i.e. 15.91 Hz. This is done to capture the dominant dynamics of the sloshing fluid in the relevant frequency range of expected motion. The amplitude of motion of structural inputs is modulated according to the aeroelastic response of the wing section.

Good prediction accuracy of sloshing loads is obtained by RBF-NN based surrogate model. A prediction error of 8% is incurred using N-sample constant inputs for structural excitation of the fuel tank. The flutter boundary of NACA64A010 with an embedded fuel tank is computed using CFD and RBF-NN to simulate sloshing. A remarkable agreement in the flutter boundary

for both models is observed. Although N-sample constant based training data performed very well, a more realistic input signal based on harmonic motions will be tested in the future to reduce the computation cost of generating training data.

The Equivalent Mechanical System is used to demonstrate the effect of fuel sloshing on the flutter boundary of NACA64A010 airfoil in transonic regime. The EMS parameters are derived by comparing its force and moment equation with those obtained from potential flow formulations for sloshing in rectangular tanks. The EMS model simulates inertial and convective components of sloshing liquids by linear sloshing modes. The flutter boundary of NACA64A010 is computed by simulating sloshing loads with the EMS model. Aeroelastic motion with EMS representation of sloshing shows a more stable aeroelastic response than with no sloshing, in general. However, the correlation between the sloshing affected flutter boundary of NACA64A010 with the EMS model and CFD is poor. A parametric study of frequency response of sloshing using CFD shows nonlinearities at higher frequencies of excitation, which cannot be captured by the EMS model. The EMS formulation is still a reliable and computationally inexpensive option for modeling sloshing in the linear regime. However, the applicability of EMS and other analytical methods are limited by geometric shapes of the vessel and structural excitation imposed on the system.

In the next study, recurrent neural network (RNN) based surrogate models are developed to efficiently and accurately predict the unsteady aerodynamic loads of a free pitching and plunging airfoil in transonic flow field and also to predict the fuel sloshing loads in a partially filled rectangular tank subjected to the same displacements of the airfoil. The study also demonstrates the feasibility of coupling these two RNN surrogate models to predict the aeroelastic response of a two-DOF airfoil in which a partially filled rectangular fuel tank is embedded. These RNN networks are trained once with the high-fidelity CFD training sample data and the trained network parameters are stored for future reuse, thus eliminating the need

for training. The training of the surrogates for external airfoil aerodynamics are computationally marginally cheaper than the surrogates for fuel tank sloshing loads owing to the increased number of outputs. The coupled RNN surrogates are tuned and used to predict both forced and free motion of the coupled aero-structural-fuel tank system for routine prediction of subsequent motion at a much cheaper computational expense with minimal compromise of prediction accuracy. The training methodology and prediction accuracy of the surrogate models are tested with forced motion of the coupled aero-structural system where training and testing data have the same amplitude of motion. The models are then trained on limited data set of free diverging aero-structural motion and tested on data set with more intense divergence. The unsteady aerodynamic and sloshing loads predicted by the surrogate models, coupled with the structural equation of motion, are utilized to predict the free motion of the aero-structural system. The resulting motion and corresponding loads agree well with the CFD data, with a significant reduction of 96.21% in the computational cost. Such RNN surrogates show enormous potential for routine use for predicting the onset of flutter, limit cycle oscillations and for flutter mitigation directed aerodynamic wing design.

The RNN based surrogate model for unsteady aerodynamic loads developed in the previous study is modified to incorporate parametric flow variations of free-stream Mach number. This model is trained with flow data varying across a range of Mach numbers in the transonic flow regime using a wing geometry. The RNN based surrogate model for unsteady aerodynamic loads and RNN based surrogate for fuel sloshing loads are used to predict the aeroelastic motion of the coupled aero-structural-fuel sloshing system. The resulting motion is compared with the aeroelastic motion of the wing with an empty fuel tank to study the effects of sloshing. A significant reduction of 99.12% in computational cost is achieved with the RNN surrogates. With the flow parametrization capability of the RNN surrogate model, they show great potential for usage in design optimization problems entailing aeroelastic stability.

In the last section, a Physics-Informed Neural Network is developed which attempts to emulate the governing equation of a 2-DOF aeroelastic system. The network parameters of PINN are learned by minimizing a modified loss function containing a residue term computed using aeroelastic equation of motion. The model is trained with aeroelastic motion profiles and corresponding load coefficients from different flow conditions and tested for prediction accuracy by feeding the aerodynamic load coefficients in batches. Except for a few patches, a good agreement between the PINN and aeroelastic solver data is observed. With network hyperparameters fine-tuning and selection of appropriate training data, PINN promises immense potential to replace conventional data-driven surrogate models for physical systems.

Naturally, the next step forward is the implementation of PINN for Euler flow equations for external compressible flows. In spite of the excellent performance of RBF-NN and RNN based surrogates for sloshing flows and aerodynamics, these models are constrained by the parametric richness of the training samples. PINN, on the other hand, requires much less data for training and calibration. Structural PINN for aeroelastic equation of motion, coupled with Euler PINN for aerodynamics has the potential to simulate aeroelastic motion for a wide range of flow conditions and has the potential to replace conventional block-box surrogates for physical systems.

In this thesis, all the aeroelastic systems are restricted to 2-DOF motion. In the future, higher structural mode shapes will be considered with an external fuel tank. Furthermore, viscous external flows will be incorporated in the aerodynamics models to emulate a more realistic aero-structural-fuel tank system. Finally, the flutter boundary optimization of transonic airfoil will be done with respect to the fill level of the embedded fuel tank.

## Bibliography

- [1] Lancaster, F.W., “Torsional Vibrations of the Tail of an Aeroplane”, Reports and Memoranda, no. 276, July 1916, AIAA Selected Reprint Series, Volume V, pp. 12-15, Aerodynamic Flutter, Garrick, I.E., ed. March 1969
- [2] Collar, A. R., “The First Fifty Years of Aeroelasticity”, Aerospace, Vol 5, No. 2, pp.12-20, Royal Aeronautical Society, Feb 1978
- [3] Tolve, L. A., “History of Flight Flutter Testing”, Proceedings of the 1958 Flight Flutter Testing Symposium, NASA SP-385, pp. 159-166, 1958
- [4] Garrick, I. E., and Reed, W. H., “Historical Development of Aircraft Flutter”, AIAA 81-0491, Journal of Aircraft, Vol. 18, No. 11, pp. 897-912, November 1981
- [5] Goldman, R. L., “Investigation of Transonic Control Surface Instabilities of a Lifting Body Configuration”, Technical Report AFFDL-TR-67- 139, Air Force Flight Dynamics Laboratory, 1968.
- [6] Norton, W. J., “Limit Cycle Oscillations and Flight Flutter Testing”, Proceedings, Society of Flight Test Engineers, 21<sup>st</sup> Annual Symposium, pp. 3.4.1-3.4.12, August 1990
- [7] Champion, L. S., and Cabrera, E. A., “F16 C/D Block 40 with Advanced Medium-Range Air-to-Air Missile (AMRAAM) Flutter Flight Test Evaluation”, AFFTC TR-92-19, December 1992
- [8] Karnick, P.T. and Venkatraman, K., “Shock Boundary Layer Interaction and Energetics in Transonic Flutter”, Journal of Fluid Mechanics, 832(12), pp. 212-240, December 2017.
- [9] Cox D., Curtiss Jr, H.C., “A modern course in aeroelasticity”, 4<sup>th</sup> Edition, Netherlands: Sijthoff and Noordhoff, 2004.
- [10] Tang, D., Attar, P., and Dowell, E.H., “Flutter/Limit Cycle Oscillations Analysis and Experiments for Wing-Store Model”, AIAA Journal, Vol. 44, No. 7, pp. 1662-1676, July 2006
- [11] Conyers, H.J., Dowell, E.H., and Hall, K.C., “Aeroelastic Studies of a Rectangular Wing with a Hole: Correlation of Theory and Experiment”, NSBE Aerospace Systems Conference, Los Angeles, CA, Feb 1-4, 2010

- [12] Bendiksen, O. O., and Kousen, K. A., "Transonic Flutter Analysis Using the Euler Equations," AIAA Paper 87-0911-CP, AIAA Dynamics Specialists Conference, Monterey, California, April 1987
- [13] Kousen, K. A., "Non-Linear Phenomena in Computational Transonic Aeroelasticity," Ph.D. Dissertation, Dept. of Mechanical and Aerospace Engineering, Princeton University, January 1989
- [14] Guruswamy, G. P., "Unsteady Aerodynamic and Aeroelastic Calculations for Wings Using Euler Equations," AIAA Journal, Vol. 28, No. 3, March 1990
- [15] Guruswamy, G. P., "Vortical Flow Computations on Swept Flexible Wings Using Navier-Stokes Equations," AIAA Journal, Vol. 28, No. 12, December 1990
- [16] Jameson, A., "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," AIAA Paper 91-1596, June 1991
- [17] Alonso, J.J., and Jameson, A., "Fully-Implicit Time Marching Aeroelastic Solutions", 32<sup>nd</sup> Aerospace Sciences Meeting and Exhibit, AIAA-94-0056, Reno, NV, January 10-13, 1994
- [18] Economou, T. D., Palacios, F., Copeland, S. R., Lukaczyk, T. W. and Alonso, J. J., "SU2: An Open-Source Suite for Multiphysics Simulation and Design," *AIAA Journal*, Vol. 54, No. 3, 2016, pp. 828-846. <https://su2code.github.io/>
- [19] Nasar, T., Sannasiraj, S. and Sundar, V., "Experimental Study of Liquid Sloshing Dynamics in a Barge Carrying Tank," Fluid Dynamics Research, Vol. 44, Issue 6, pp. 427-458, 2008
- [20] Nam, B.W., Kim., Y., "Effects of Sloshing on the Motion on Response of LNG-FPSO in Waves," 22<sup>nd</sup> Workshop on Water Waves and Floating Bodies, Plitvice, Croatia, 2007
- [21] Kim, Y., "A Numerical Study on Sloshing Flows Coupled with Ship Motion – The Anti-Rolling Tank Problem," Journal of Ship Research, Vol. 46, pp. 52-62, 2002
- [22] Kim, Y., Nam, B., Kim, D., and Kim, Y., "Study on Coupling Effects of Ship Motion and Sloshing," Ocean Engineering, Vol. 34, Issue 16, pp. 2176-2187, 2007
- [23] Abramson, H.N., "The Dynamic Behavior of Liquids in Moving Containers with Applications to Space Vehicle Technology," Report SP-106, NASA, 1966

- [24] Luskin, H., and Lapin, E., "An Analytical Approach to Fuel Sloshing and Buffeting Problems in Aircraft", Journal of the Aeronautical Sciences, Vol. 19, No. 4, pp. 217-228, April 1952
- [25] Cazier, F. W. Jr., Doggett, R. V., Jr., and Ricketts, R. H., "Structural Dynamic and Aeroelastic Considerations for Hypersonic Vehicles," NASA TM-104110, 1991
- [26] Farhat, C., Chiu, E.K., Amsallem, D., Schotte, J.S. and Ohayon, R., "Modeling of Fuel Sloshing and its Physical Effects on Flutter," AIAA Journal, Vol. 51, No. 9, pp - 2252-2265, 2013 DOI: 10.2514/1.J052299
- [27] Chiu, E.K. and Farhat, C., "Effects of Fuel Slosh on Flutter Prediction," AIAA 2009-2682, 50th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 4-7 May 2009, Palm Springs, CA, USA
- [28] Firouz-Abadi, R. D., Zarifian, P., Haddadpour, H., "Effect of fuel sloshing in the external tank on the flutter of subsonic wings", Journal of Aerospace Engineering, Vol. 27, 2014
- [29] Hall, J. C., Rendall, T. C. S., Allen, C. B., and Peel, H., "A Multi-Physics Computational Model of Fuel Sloshing ", Journal of Fluids and Structures, Vol. 56, pp. 11-32, July 2015
- [30] Liu, G. R., and Liu, M. B., "Smoothed Particle Hydrodynamics: A Meshfree Particle Method", World Scientific Publishing Co., pp. 472, 2003
- [31] Banim, R., Lamb, R., and Bergeon, M., "Smoothed particle hydrodynamics simulation of fuel tank sloshing", 25th International Congress of the Aeronautical Sciences, 2006
- [32] Graham, E. W., "The Forces Produced by Fuel Oscillation in a Rectangular Tank", Report No. SM-13748, Douglas Aircraft Company Inc., Santa Monica, Calif., April 13, 1950 (revised April 16, 1951)
- [33] Ibrahim, R.A., "Liquid Sloshing Dynamics: Theory and Application," Cambridge Press, 2005
- [34] Feng, G. C., "Dynamic Loads Due to Moving Liquid", AIAA Paper No. 73:409, 1974
- [35] Faltinsen, O. M., "A Nonlinear Theory of Sloshing in Rectangular Tanks", Journal of Ship Research, Vol. 18, No. 4, pp. 224-241, 1974
- [36] Haroun, M. A., "Dynamic Analysis of Liquid Storage Tanks", Report EERL, No. 80-4, California Institute of Technology, Pasadena, CA, 1980

- [37] Haroun, M. A., “Vibration Studies and Tests of Liquid Storage Tanks”, *Earthquake Engineering and Structural Dynamics*, Vol. 11, No. 2, pp. 179-206, 1983
- [38] Hirt, C.W. and Nicholas, B.D., “Volume of Fluid Method for the Dynamics of Free Boundaries,” *Journal of Computational Physics*, Vol 39, pp. 201-225, 1981
- [39] Partom, L. S., “Application of the VOF Method to the Sloshing of a Fluid in a Partially Filled Cylindrical Container”, *International Journal of Numerical Methods in Fluids*, Vol. 7, No. 6, pp. 535-550, 1987
- [40] The OpenFOAM Foundation, “OpenFoam V.5 User Guide”, (2011-2018) <https://cfd.direct/openfoam/user-guide/>
- [41] Joppich, W., and Kurschner, M., “MpCCI – A Tool for the Simulation of Coupled Applications”, Special Issue: Computational Frameworks, Vol/ 18, Issue 2, pp. 183-192, February 2006
- [42] Kataoka, S., Minami, S., Kawai, H., Yamada, T., and Yoshimura, S., “A Parallel Iterative Partitioned Coupling Analysis System for Large-Scale Acoustic Fluid-Structure interactions”, *Computers and Fluids*, Vol. 141, pp. 259-268, 2014
- [43] Morel, T., Duchaine, F., Thévenin. A., Piacentini, A., Kirmse, M., and Qumerais, E., “Open-PALM coupler; version 4.1.4. User guide and training manual”, Centre Européen de Recherche et de Formation Avancéen Calcul Scientifique (CERFACS), Toulouse, France. 2013
- [44] Thévenin, A., “OASIS3-MCT & open-PALM: 2 Open Source Codes Couplers”, Presentation at Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS), Toulouse, France, 2012
- [45] Larson, J., Jacob, R., and Ong, E., “The model coupling toolkit: a new Fortran90 toolkit for building multiphysics parallel coupled models”, *International Journal of High Performance Computer Application*, Vol. 19, pp. 277–92, 2005
- [46] Pawłowski, R., “Physics Integration Kernels (PIKE)”, Computer software, Version 00, July 31, 2014. <https://www.osti.gov//servlets/purl/1253615>
- [47] Bungartz, H., Lindner, F., Gatzhammer, B., Mehl, M., Scheufele, K., Shukaev, A., and Uekermann, B., “preCICE – A Fully Parallel Library for Multi-Physics Surface Coupling”, *Computers and Fluids*, Vol. 141, pp. 250-258, 2016

- [48] Weller, H.G., Tabor, G., Jasak, H., and Fureby, C., "A Tensorial Approach to Computational Continuum Mechanics using Object-Oriented Techniques", *Journal of Computational Physics*, Vol. 12, Issue 6, pp. 620-631, 1998. DOI: 10.1063/1.168744
- [49] Dhondt, G., "The Finite Element Method for Three-Dimensional Thermomechanical Applications", *Wiley*, June 2004. ISBN: 978-0-470-85752-6
- [50] de France, E., "Finite Element Analysis of Structures and Thermomechanics for Studies and Research", Open Source on [www.cide-aster.org](http://www.cide-aster.org), 1989 – 2017
- [51] DeSalvo, G. J., and Swanson, J. A., "ANSYS Engineering Analysis System User's Manual", Houston, Swanson Analysis Systems, 1985.
- [52] COMSOL Multiphysics Reference Manual, version 5.3", COMSOL, Inc, [www.comsol.com](http://www.comsol.com)
- [53] Lucia, D.J., Beran, P.S. and Silva, W.A., "Reduced-Order Modeling: New Approaches for Computational Physics," *Progress in Aerospace Sciences*, Vol. 40, pp. 51-117, 2004.
- [54] Dowell, E.H. and Hall, K.C., "Modeling of Fluid-Structure Interaction," *Annual Review of Fluid Mechanics*, Vol. 33, pp. 445-490, 2001
- [55] Silva, W. A., Hajj, M. R., and Prazenica, R. J., "Recent Applications of the Volterra Theory to Aeroelastic Phenomena", IMAC XXIII: Conference and Exposition on Structural Dynamics, Orlando, FL, Jan 31 – Feb 02, 2005
- [56] Faller, W.E. and Schreck, S. J., "Unsteady Fluid Mechanics Applications of Neural Networks," AIAA-95-0529, 33rd Aerospace Sciences Meeting and Exhibit, 9-12 January 1995, Reno, NV
- [57] Voitcu, O. and Wong, Y.S., "A Neural Network Approach for Nonlinear Aeroelastic Analysis," 43rd AIAA/ASME/ASCE/AHS,ASC Structures, Structural Dynamics, and Materials Conference, 22-25 April 2002, Denver, Colorado, USA, 2002
- [58] Mannarino, A. and Mantegazza, P., "Nonlinear Aeroelastic Reduced Order Modeling by Recurrent Neural Networks," *Journal of Fluids and Structures*, Vol. 48, pp. 103-121, 2014.
- [59] Narendra, K.S. and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. Neural Networks*. 1(1990) 4–26

- [60] Elanayar, S. and Shin, Y., "Radial Basis Function Neural Network for Approximation and Estimation of Nonlinear Stochastic Dynamic Systems," IEEE Trans. Neural Networks.(1994)594–603
- [61] Lowe, D., & Broomhead, D., "Multivariable Functional Interpolation and Adaptive Networks Complex Systems", Complex Systems Publications Inc., 2(3), pp. 321–355, 1988
- [62] Leonard, J. A.,& Kramer, M. A., "Radial basis function networks for classifying process faults", IEEE Control Systems, 11(3), pp. 31–38, 1991
- [63] Billings, S.A., "Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains, Wiley, ISBN 978-1-1199-4359-4, 2013
- [64] Winter, M., and Breitsamter, C., "Reduced-Order Modeling of Unsteady Aerodynamic Loads Using Radial Basis Function Neural Networks", Deutsche Gesellschaft für Luft- und Raumfahrt-Lilienthal-Oberth eV, 2014
- [65] Mandic, D.P. and Chambers, J.A., "Recurrent Neural Networks for Prediction – Learning Algorithms, Architectures and Stability", John Wiley and Sons, LTD, ISBN 0-471-49517-4
- [66] Raissi, M., Perdikaris, P., Karniadakis, G.E., "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", Journal of Computational Physics, Vol 378, pp. 66-707, 2019
- [67] Jameson, A., Schimidt, W., Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Range-Kutta Time-Stepping Schemes", AIAA, pp. 81-1259, June 1981
- [68] Sanchez, R., Palacios, R., Economou, T.D., Kline, H.L., Alonso, J.J., and Palacios, F., "Towards a Fluid-Structure Interaction Solver for Problems with Large Deformations within the Open-source SU2 Suite", American Institute of Aeronautics and Astronautics
- [69] Sanchez, R., Kline, H.L., Thomas, D., Variyar, A., Righi, M., Economou, T.D., Alonso, J.J., Palacios, R., Dimitriadis, G., and Terrapon, V., "Assessment of the Fluid-Structure Interaction Capabilities for Aeronautical Applications of the Open-Source SU2", European Congress on Computational Methods in Applied Sciences and Engineering, Crete Island, Greece, 5-10 June, 2016

[70] Le Tallec, P., and Mouro, J., "Fluid structure interaction with large structural displacements", Computer Methods in Applied Mechanics and Engineering, Vol. 190, No. 24-25, 2001, pp. 3039-3067.

[71] Received from www.precice.org/

[72] Chourdakis, G., "A General OpenFOAM Adapter for the Coupling Library preCICE", Master's Thesis, Technische Universität München, October 2017

[73] Sjöberg J., Zhang, Q., Ljung , L., Benveniste, A., Delyon, B., Glorennec, P-Y., Hjalmarsson, H, and Juditsky, A., "Nonlinear Black-box Modeling in System Identification: a Unified Approach", Automatica, Vol. 31, Issue 12, pp. 1691-1724, December 1995.  
[https://doi.org/10.1016/0005-1098\(95\)00120-8](https://doi.org/10.1016/0005-1098(95)00120-8)

[74] Ljung, L., and Glad, T., "Modeling of Dynamic Systems", Prentice-Hall, 1994. ISBN:0-13-597097-0

[75] Du, K. L., & Swamy, M. N., "Neural Networks in a Softcomputing Framework", Springer Science & Business Media, 2006

[76] Mohaghegi, S., del Valle, Y., Venayagamoorthy, G. K., & Harley, R. G., "A comparison of PSO and backpropagation for training RBF neural networks for identification of a power system with STATCOM", In Proceedings 2005 IEEE of the Swarm Intelligence Symposium SIS 2005, pp. 381–384, IEEE, 2005

[77] Chng, E. S., Chen, S., & Mulgrew, B., "Gradient Radial Basis Function Networks for Nonlinear and Nonstationary Time Series Prediction", IEEE Transactions on Neural Networks, 7(1), pp. 190–194, 1996

[78] Lee, M. J., & Choi, Y. K., "An Adaptive Neuro Controller Using RBFN for Robot Manipulators", IEEE Transactions on Industrial Electronics, 51(3), pp. 711–717, 2004

[79] Lloyd, Stuart P. "Least Squares Quantization in PCM", Information Theory, IEEE Transactions, 28(2), pp. 129-137, 1982

[80] Kiefer, J., and Wolfowitz, J., "Stochastic Estimation of the Maximum of a Regression Function", Annals of Mathematical Statistics, Vol 23, No. 3, pp. 462-466 (1952)

[81] Mirjalili, S., "Evolutionary Algorithms and Neural Networks – Theory and Applications", Studies in Computational Intelligence, Springer, Vol 780

- [82] Ramachandran, P., Zoph, B., Le., Q.V., “Searching for Activation Functions,” <https://arxiv.org/abs/1710.05941>
- [83] P. Werbos, “Backpropagation Through Time: What it Does and How to do it”, Proceedings of the IEEE, 78 (1990), pp. 1550-1560, <https://doi.org/10.1109/5.58337>
- [84] Kingma, D.P., Ba, J., “Adam: A Method for Stochastic Optimization”, 3rd International Conference for Learning Representations, San Diego, CA, USA, 2015, (arXiv:1412.6980[cs.LG])
- [85] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al., “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”, arXiv preprint arXiv:1603.04467, 2016
- [86] McMullen, M.S., Jameson, A. and Alonso, J.J., "Application of a Non-Linear Frequency Domain Solver to the Euler and Navier-Stokes Equations", AIAA-2002-0120 in Proceedings of AIAA 40th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 14-17, 2002, USA.
- [87] Roache, P.J., “Verification and Validation in Computational Science and Engineering,” Hermosa Publishers, Albuquerque, New Mexico, 1998
- [88] Yusong, C., Graczyk, M., Pakozdi, C., Lu, H., Huang, F., and Yang, C., “Sloshing Load Due to Liquid Motion in a Tank (Comparison of Potential Flow, CFD and Experiments)”, International Society of Offshore and Polar Engineers, Beijing, China, June 2010
- [89] Anaconda Software Distribution. Python Software versions 2.7-3.7, Anaconda, Inc., Austin, TX, USA, 2019 (since 2012) , URL: <https://www.anaconda.com>.
- [90] Heeg, J., Chwalowski, P., Schuster, D. M., Raveh, D., Jirasek, A., and Dalenbring, M., “Plans and Example Results for the 2nd AIAA Aeroelastic Prediction Workshop”, 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Jan 2015 (DOI: 10.2514/6.2015-0437)
- [91] Piatak, D. J., and Cleckner, C. S., “Oscillating Turntable for the Measurement of Unsteady Aerodynamic Phenomena”, Journal of Aircraft, Vol. 40, No. 1, Jan-Feb 2003