Goal-Oriented Mesh Adaptation for High-Fidelity Aeroelastic Simulations

by

Vivek Ojha

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2022

Doctoral Committee:

Prof. Carlos E. S. Cesnik, Co-Chair
Prof. Krzysztof J. Fidkowski, Co-Chair
Dr. Philip Beran
Prof. Yin Lu Young

Vivek Ojha

vojha@umich.edu

ORCID iD: 0000-0002-0528-8520

For my parents

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my co-advisor Prof. Krzysztof Fidkowski. Chris has been an incredible mentor and I am grateful for his continuous support and invaluable advice throughout graduate school. I value the flexibility that he gave me to pursue different ideas in my research. His immense knowledge and passion for teaching will always inspire me. I would also like to thank my co-advisor Prof. Carlos Cesnik, whose insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. I would also like to thank Dr. Philip Beran for agreeing to be an external member of the committee. The questions and comments by Dr. Beran, during our biweekly meetings, helped shape this work. I also thank Prof. Julie Young for agreeing to be a cognate committee member. I am also grateful for the support given by Dr. Nathan Wukie from the Air Force Research Laboratory.

This work would not be possible without the support that I received from many friends and colleagues in the aerospace department. I would like to thank my current and former colleagues in Prof. Fidkowksi's and Prof.Cesnik's research group for being an inspiration to do better work and also for providing me with a great environment to be in for many years. Several lab members and alumni helped me learn everything that went into this thesis: Braden Frigoletto, Gary Collins, Matteo Franciolini, Guodong Chen, Qingzhao Wang, Devina Sanjaya, Yukiko Shimizu, Kaihua Ding, Kevin Doetsch, Divya Sanghi, Cristina Riso, and Christopher Lupp. Thank you for all the countless hours you helped me in my research. I would also

like to acknowledge the Department of Aerospace Engineering at the University of Michigan. My graduate experience benefited greatly from the courses I took and the high-quality seminars that the department organized. I would be remiss if I did not thank Denise Phelps, Ruthie Freeman, and the International Center who deserve credit for providing much-needed assistance with administrative tasks.

I value the friendship of Harsh, Vishnu, and Gurmeet who helped me feel at home in Ann Arbor. I will always cherish our cookouts and late-night discussions. Pawel, Gary, Matteo, and Qingzhao, were incredible officemates and I will miss our mid-day coffee breaks. I would also like to thank Nishchint and Satyam for being there for me since undergrad. Thank you Mom and Dad for being by my side and believing in me. Without you, this wouldn't be possible. Finally, I would like to thank Aiswarya for being my partner in this journey. Thank you for giving me the reason to be better.

This work was supported by the U.S. Air Force Research Laboratory (AFRL) under the Michigan-AFRL Collaborative Center in Aerospace Vehicle Design (CCAVD), with Dr. Philip Beran as the task Technical Monitor.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

This work demonstrates a solution-adaptive approach for solving fluid-structure interaction problems using high-fidelity numerical methods along with a detailed analysis of mesh-motion errors. A high-order partitioned approach is applied to couple the fluid and the structural subsystems, where the fluid subsystem is discretized using a discontinuous Galerkin finite-element method, while the structural solver uses a continuous Galerkin discretization. High-order time integration schemes are used by the coupled solver to march forward in time, and the spatial meshes of the fluid and the structural subsystems are adapted using output-based methods. The error estimates for the unsteady outputs are evaluated by calculating the unsteady adjoint of the coupled problem. Adaptive meshing is used to demonstrate the importance of mesh-motion errors on output convergence and a comprehensive analysis is conducted to control such errors arising from the mesh deformation algorithm. The adapted meshes converge at a faster rate with fewer degrees of freedom, thereby increasing accuracy and reducing computational cost. The benefits of adaptive meshing are demonstrated on two-dimensional and three-dimensional aeroelastic problems for a variety of coupled outputs. As an alternative to the adjoint-based mesh adaptation process, a data-driven method is also developed to improve the efficiency of non-uniform anisotropic mesh generation.

CHAPTER I

Introduction

The interaction between an elastic structure and an internal or surrounding fluid flow is a phenomenon observed in many areas of engineering. These problems, which fall under the umbrella of Fluid-Structure Interaction (FSI), play a prominent role in many scientific fields. Design of wings in modern aerospace vehicles [26, 27], wind turbines blades [133], cable-stayed bridges [109, 108], energy harvesting devices [4, 115], and prosthetic heart valves [57] are a few examples of such problems. The application of interest of this thesis is in the field of aerospace engineering. Historically, design cycles of aerospace vehicles relied primarily on wind tunnel experiments to characterise newer designs. However, with increasing cost of wind tunnel tests and an inability to simulate realistic flow conditions as seen by aircraft, there has been a shift towards incorporating numerical simulation in vehicle design [1]. Advancements in computer hardware and developments in the fields of numerical methods, computational fluid dynamics (CFD) [51], computational structural dynamics (CSD) [81], thermal simulations [64], computational acoustics [116] have helped accelerate the design and analysis of newer vehicles.

In modern aerospace vehicles, the relevant physical phenomena are quite complicated. These vehicles consist of flexible structural parts that are under stresses induced by

high temperature environments and the fluid flow. The dynamics of the guidance and control system also play a significant role in the physics. Design for space also involves the aerothermodynamic coupling of spacecraft surface chemistry, which is important for analysis of plume and aerodynamic heating. As the various subsystems involved in the design are inherently coupled, a single subsystem analysis is unable to capture the non-linearities and instabilities that can occur due to the coupling of such subsystems. In order to design vehicles that are safe in the entire flight envelope, being able to accurately capture the coupled phenomena has become paramount. The need to accurately capture coupled physics has resulted in advancements in the fields of multi-physical system analysis including aeroelasticity, aeroservoelasticity, aerothermoelasticity, aeroacoustics, etc. The focus of this work is the field of aeroelasticity, which comes under the umbrella of FSI problems.

Based on the seminal book by Bisplinghoff et al.[18] - *"Aeroelasticity is defined as a science which studies the mutual interaction between aerodynamic forces and elastic forces, and the influence of this interaction on airplane design. Aeroelastic problems would not exist if airplane structures were perfectly rigid. Modern airplane structures are very flexible, and this flexibility is fundamentally responsible for the various types of aeroelastic phenomena."* Aeroelastic phenomena in modern high-speed aircraft have profound effects on the design of structural members and somewhat lesser but nonetheless important effects on mass distribution, lifting surface planform, and control systems. Accurate prediction of aeroelastic phenomena, such as bifurcations, limit-cycle oscillations, flutter, control divergence, control effectiveness, buffet, control system reversal, and load distribution, has become important for newer, more efficient designs.

Work in the field of FSI began in the late 1970s and early 1980s. Since the ear-

lier works of Belytschko [16], Bathe and Hahn [12] and Donea et al. [34], the field of computational FSI has reached a significant level of maturity. The numerical simulation of FSI started to takeoff in the 1990s and has progressed to such an extent that flutter points of a complete F-16 fighter jet can be numerically obtained [37]. Advances in the power of computers and the development of new numerical techniques have made it feasible to effectively address complex industrial problems by means of a purely computational FSI analysis. However, computational costs of these multi-physics simulations are still a bottleneck [37] in the design process. There exists a growing need for the application of high-fidelity tools for analysing FSI systems at low computational costs. Therefore, the aim of this thesis is to develop an efficient and accurate high-fidelity FSI solver for aeroelastic applications.

1.1 Computational FSI

Numerical simulation of fluid-structure interaction is a complex and challenging problem. The interaction between the two systems is non-linear and involves multiple scales, thereby making the coupled system challenging to solve. Many approaches have been suggested for simulating fluid-structure interaction [62]. Numerical approaches for solving the coupled fluid-structure system based on meshes can be broadly divided into two types: monolithic and partitioned. The monolithic approach [82] is a fully-coupled approach in which the two systems are solved simultaneously. The interfacial conditions are implicit in the solution procedure. This approach gives rise to strongly coupled, usually very large and nonlinear, algebraic systems of equations. Although there has been substantial progress in designing efficient numerical schemes for solving the nonlinear and associated linear systems [8, 70], the computational effort is still immense and numerically accurate results for three-dimensional problems are still rare [105]. The second approach, generally referred to as the partitioned approach [40], uses two separate solvers to solve the fluid and

structural subsystems and then couples them through a communication between the solvers. The interfacial conditions are used explicitly to communicate information between the fluid and structure solutions. The separation of the solid from the fluid part in the solution process usually yields a loss in efficiency and robustness mainly due to the so-called added-mass effects [21]. These effects become especially relevant for hydroelastic problems and lightweight structures, since the mass of the entrained fluid by the dynamical structure is a significant part of the total mass. However, for aeroelastic applications in compressible flows, the partitioned approach is stable and convergent for sufficiently small time steps [119, 101].

Another general classification of the FSI solution procedure is based on the treatment of the interface. In an FSI problem, both the fluid and the structural domain deform. Structural equations are usually solved in a Lagrangian formulation, which means that the grid nodes move at the same velocity as the material. Thus, dealing with deformations in the structural simulation normally does not cause difficulties. On the other hand, flow equations are traditionally solved in a fixed domain that does not deform, using an Eulerian formulation. A variety of techniques [32] have been developed to to take into account the deforming fluid domain and these can be classified as conforming mesh methods and non-conforming mesh methods. Conforming mesh methods track the moving interface by deforming the fluid mesh such that the resulting mesh conforms to the interface. This helps in controlling the mesh resolution near the interface and maintaining accuracy in such critical flow regions. Arbitrary Lagrangian–Eulerian (ALE) schemes [34, 35, 98], space-time methods [117] and local and global remeshing [85] are some of the most common techniques used to track the interface. Non-conforming mesh methods are preferred in FSI problems where the geometric complexity of the interface may require a fluid mesh that is not affordable or not manageable via a conformal approach. This approach can be seen

as a special case of interface representation techniques where the interface geometry is represented over a stationary-fluid mesh. The key point is that the fluid mesh does not move to track the interface. Immersed-boundary [83], fictitious-domain [54] and cut-cell methods [102] are a few examples of the non-conforming approaches. The hallmark of these methods is their inherent ability to handle arbitrarily complex domains with arbitrarily complex deformable boundaries without the need to construct grids that conform to and deform with solid boundaries. However, their inability to anisotropically cluster grid nodes in the vicinity of solid boundaries creates difficulties in the simulations of high Reynolds number turbulent flows [68].

The numerical approaches for computational FSI mentioned above rely on the generation of a topological map of the geometry, i.e. a grid or mesh for the discretization of the problem. Despite the robustness of such approaches, mesh-based methods also suffer from certain deficiencies. Mesh-based methods face issues from pre-defined connectivity constraints over the nodes, the maintenance of which can become tedious and time consuming in the case of deforming meshes. Mesh-based methods are also challenging to implement for problems involving cracks in solids, free surface flows and multiphase flows, where it becomes difficult to maintain connectivity between grid elements. As a result of these challenges, mesh-free and meshless particle methods (MPM) have been developed in recent years [72]. These methods refer to a class of computational techniques in which the structure of the mesh is eliminated, and in which the solution is approximated over a set of arbitrarily distributed data points or nodes. Smooth-particle hydrodynamics [84], the diffuse-element method [89], and the finite-point method [95], are a few examples of particle methods. MPMs are a promising alternative to mesh-based methods that can easily handle flow problems around moving boundaries and deforming structures. However, the high computational resource requirement and lack of commercial MPM-based solvers for FSI [80]

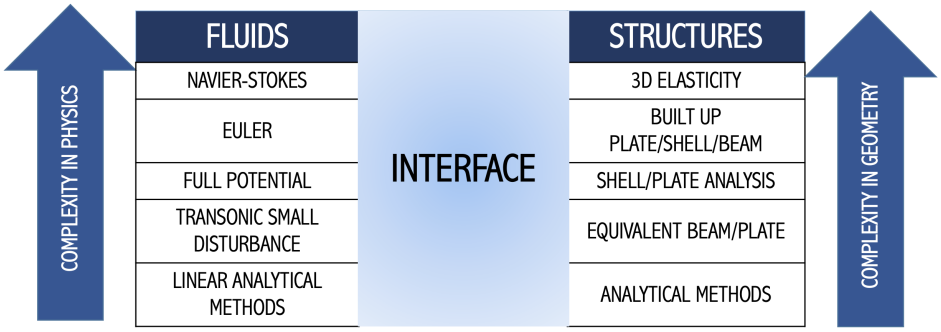


Figure 1.1: Varying levels of fidelity in modeling for fluids and structures. The figure has been adapted from a similar figure in the work by Guruswamy [58].

limit their use for such applications.

1.2 Computational Aeroelasticity

Computational aeroelasticity (CAE) refers to the coupling of high-level CFD methods with structural dynamic tools to perform aeroelastic analysis. The fluid and structural subsystems can be modeled at various levels of fidelity/complexity in physics. From the fluids perspective, models ranging from low-fidelity look-up tables to the high fidelity Navier–Stokes equations can be used. Similarly to structures, the data can be obtained starting from the low fidelity modal approach to detailed three-dimensional finite elements, as shown in Figure 1.1. Various levels of fidelity are used in different stages of the design process [5]. For many years, linear models have been used by aeroelasticians to understand aeroelastic phenomena such as flutter, divergence, control surface reversal, and gust response. These models have been described in the classic texts of Rosembaum [110], Fung [52], and Bisplinghoff [18]. However, with increasing aspect-ratios [118] in modern aircraft designed for transonic flow regimes, the study of non-linear aeroelastic phenomena has become essential. The non-linearities in aeroelastic systems are observed due to three main sources: structures, aerodynamics, and control systems. The most common type of non-linearity occurring in structures is due to large deformation, also referred to as geometric non-

linearity. Dissipative forces (non-linear damping) and material properties (non-linear stress-strain relationship) are other sources of structural non-linearities. Aerodynamic non-linearities arise from unsteady separated flow, oscillating shocks in the transonic regime, shock-boundary layer interaction, and wake roll-up. Control non-linearities include geometric non-linearities, where large deformations affect control effectiveness, and reversal. Understanding the physics behind these non-linearties has necessitated the development of nonlinear aeroelasticity.

Apart from modelling the fluid and the structural subsystems, CAE also requires the development of an interfacing technique in the case when the computational grids of the individual subsystems do not match at the boundary, as shown in Figure 1.2. The interface technique is required to exchange information such as stresses and displacements at the interface. Conservation of virtual work over the interface and low interpolation errors are important criteria for choosing the interface technique for

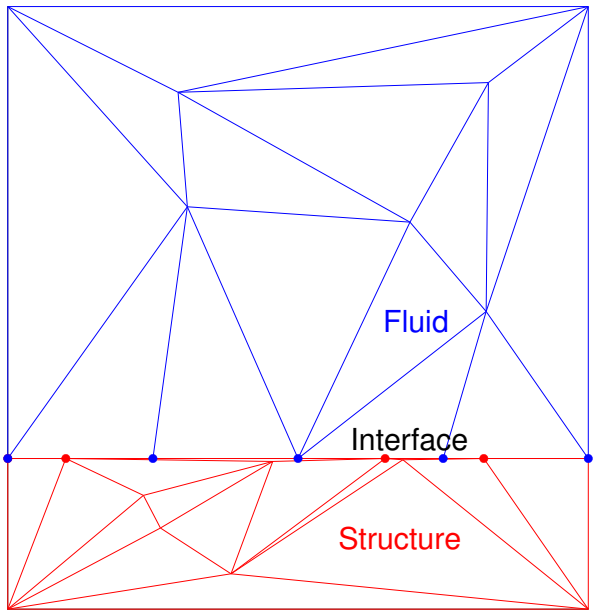


Figure 1.2: Non-matching grids at the interface of an FSI problem.

high-fidelity CAE. A variety of consistent and conservative methods exist to transfer information between nonlinear CFD and CSD grids [58, 31, 113]. Radial basis functions (RBF), thin plate splines, nearest-neighbour interpolation, and weighted-residual methods [30] are a few example of commonly used interface methods.

1.3 Motivation and Approach

CAE continues to play a critical role in the development of modern air vehicles. The current suite of linear aeroelastic models is unable to capture the structural and aerodynamic non-linearties which need to predicted/captured/modelled for newer designs. Linearization fails in the presence of strong non-linearities and therefore high-fidelity tools are necessary to capture such phenomena [5]. The need to capture such non-linearities necessitates the development of high-fidelity tools for CAE that can model the flow around complex deforming geometries in compressible flows in a robust, accurate, and efficient way.

Under mesh-based methods, the partitioned approach of solving computational FSI problems offers a number of advantages over the monolithic approach, such as customization, independent modeling, software reuse, and modularity. Being able to utilize the existing high-fidelity solvers of CFD and CSD and coupling the solvers tightly in space and time can provide provide robust, and accurate solutions for CAE. Farhat et al. [39, 38] were the first to develop a provably second-order implicit, time-accurate, and staggered procedure for the solution of an FSI problem. Second-order in time methods have been combined with first and second order accurate in space discretization of the Navier-Stokes for aeroelastic applications [56]. Second-order in time methods (with subiterations) have also been combined with a sixth-order Navier–Stokes solver in the work by Gordnier [55] to simulate a membrane wing airfoil. However, second-order methods face the drawback of long computational times

for simulations of long-term dynamic behavior due to the constrain on the time step size. Higher-order time integration schemes are often more computationally efficient than popular lower-order schemes, and this observation has driven the development of higher-order time integration schemes for FSI simulations. Following the definition of high-order from Wang et al. [128], high-order numerical methods are defined as third order or higher. One example is a partitioned scheme based on Implicit–Explicit (IMEX) Runge–Kutta methods [122]. This scheme offers arbitrarily high orders of accuracy in time and the ability to use implicit solvers for both the fluid and the structure.

High-order time integration methods have been coupled with the high-order spatial discretization such as Discontinuous Galerkin (DG) discretization of the Navier-Stokes equations for a high-fidelity (both in space and time) FSI solver by Persson et al. using monolithic [100] and partitioned [50] approach. However, in the high-fidelity solver, the fluid-structure interface is assumed to be face matching and the interface elements of the fluid and structure are discretized using the same polynomial order. Other higher-order discretizations of the Navier-Stokes equations, such as the Hybridizable discontinuous Galerkin (HDG) [112, 69] methods have found success in the monolithic approach and are an active area of research. No work exists on maintaining high spatial order in high-fidelity FSI solvers in the case of non-matching meshes and varying discretization of the fluid and structural elements at the interface. This observation motivates the first part of the thesis, which builds on the work by Persson et al. [50] to develop a high-order (both in space and time) partitioned FSI solver for aeroelastic application with non-matching grids at the interface.

High-order numerical methods provide greater levels of accuracy for the same number of degrees of freedom when compared against a low-order method. However, this

comes at a higher computational cost due to denser linear systems that need to be solved [128]. The DG discretization of the Navier-Stokes equations, an example of a high-order method, provides higher accuracy by using a high-degree local polynomial basis on each element in the mesh. With increase in the spatial order, the total number of degrees of freedom per element increases, resulting in higher computational cost per element. Solution-based *hp*-adaptive methods have been shown to be effective in minimizing the computational cost of high-order methods to achieve a given level of accuracy [46].

Goal-oriented adaptive strategies rely on local refinement indicators to guide the adaptive procedure. These refinement indicators are obtained from duality-based a posteriori error estimates for the goal functional of interest. To compute these goal-oriented error estimates, one requires the solution of an adjoint problem. For nonlinear problems, this adjoint problem is based on the linearized-adjoint operator. Pioneering work in the field of goal-oriented error estimation and adaptivity has been performed by Becker and Rannacher [14] and Prudhomme and Oden [93]. These methods have been applied to the fields of CFD [22, 74] and extended to multi-physical problems such as FSI [36, 120]. Adaptive meshing based on current position of the structure [121], elemental residuals [3], physical indicators [124] and conditions numbers [79] have also been applied for FSI problems. However, these mesh adaption criteria do not directly target output accuracy and are not as efficient as goal-oriented techniques. The benefits of goal-oriented mesh adaption using the adjoint weighted residual for FSI has been showcased primarily with low-order monolithic solvers [104, 130, 53, 6]. However, there exists a gap in the literature in developing goal-oriented coupled mesh adaptation for high-order partitioned solvers, and this observation motivates the second part of this thesis.

Solution-adaptive methods based on adjoint-weighted residual have dramatically improved the accuracy and efficiency of aerodynamic simulations [60, 125, 96, 45, 127, 74, 91, 132]. Despite their great success in aerospace applications, the additional computational cost and implementation complexity associated with adjoint-based methods cannot be neglected. First, adjoint-based methods require solving a dual linear system, i.e. , the adjoint equation set, which is of the same size as the primal system or even larger when solving on an enriched space. Second, the implementation of adjoint methods often requires the transpose of the residual Jacobian matrix, which is not always available in explicit solvers or Jacobian-free methods. In these circumstances, either the continuous adjoint equations could be derived and directly discretized [86], or special implementation efforts are required [67], adding considerable costs and effort in the development. The additional computational costs associated with the adjoint solutions, in addition to the implementation efforts, have largely hindered more widespread use of adjoint-based adaptation techniques in practice.

In order to simplify and accelerate the mesh adaptation procedure, several attempts have been explored recently using machine learning techniques, chosen mainly for their non-intrusive nature and fast online evaluations. Manevitz et al. [78] used feed-forward neural networks to predict the solution gradients in time-dependent problems, which then provided an indicator to drive mesh adaptation. Zhang et al. proposed MeshingNet [134] which takes in a chosen set of parameters to predict the local mesh density, through feed-forward networks with residual connections. Huang et al. [63] treated the computational mesh with various element sizes as a gray-scale image such that convolutional neural network architectures can be applied in mesh generation. Chen and Fidkowski [25] presented a convolutional neural network using a physical-reference mapping to avoid direct treatments of physical quantities as images. Most of these studies only focused on the element sizing and largely ignored mesh anisotropy,

i.e., element stretching and orientation. However, anisotropic meshes are important in practice for efficiently resolving certain flow features, such as boundary layers, wakes, and shocks in aerodynamic simulations. For optimal anisotropic mesh generation, Fidkowski and Chen [44] designed feed-forward networks to predict the mesh anisotropy information using the flow primal and adjoint features, while still requiring appropriate sizing information from standard adjoint-based error estimates. The lack of literature in using a neural network to predict the optimal computational mesh, including both the element sizing and stretching motivates the final part of the thesis.

1.4 Thesis Overview

This thesis addresses the development of a high-order FSI solver and a goal-oriented mesh adaptation framework that directly targets output error in aeroelastic problems. The specific contributions of the dissertation are as follows:

- Developed a high-order partitioned FSI solver by combining high-order spatial and temporal discretizations techniques.
- Developed a spatial coupling algorithm to maintain the high-order nature of the FSI solver with non-matching nodes at the FSI interface.
- Developed guidelines for generating initial meshes with low mesh-motion errors and better output convergence for CFD simulations with deforming domains.
- Developed goal-oriented mesh adaptation for high-order FSI and demonstrated the advantages of adaptive meshing for two-dimensional and three-dimensional aeroelastic problems.
- Explored the feasibility of a convolutional neural network (CNN) for generating optimal initial meshes for CFD problems without solving for the adjoint variables.

Chapter 2 describes the high-order spatial and temporal discretization used for the fluid and structural subsystem. The spatial and temporal coupling between the two subsystems is presented in Chapter 3. Chapter 4 outlines the output-based error estimation procedure and the mesh adaptation strategy. Chapter 5 describes the generation and control of errors during mesh-deformation in fluid simulations with deforming domains. Chapters 6, 7 and 8 present the application and benefits of coupled mesh adaptation in one, two, and three dimensional aeroelastic problems respectively. The three chapters contain results demonstrating the benefits of high-order in computational aeroelasticity and the efficiency gained by using goal-oriented mesh adaptation in both the fluid and structural discretization. Chapter 9 presents an efficient way to predict optimal computational fluid meshes using machine learning. Finally, conclusions and ideas for future work are given in Chapter 10

CHAPTER II

Governing Equations and Discretization

The FSI system is governed by the Navier-Stokes equations and the elasticity equations. This chapter reviews the governing equations, and the high-order spatial and temporal discretizations used in the fluid and the structural subsystems. The chapter begins with a review of the flow equations, the augmentations necessary to the Navier-Stokes equations to deal with deforming domains, and a high-order discontinuous Galerkin discretization of these equations in space. The structural governing equations are reviewed thereafter, along with the high-order continuous Galerkin (CG) discretization. The chapter ends with a review of the temporal discretization scheme used in the individual subsystems to march in time.

2.1 Fluid Subsystem

A general system of convection-diffusion equations can be written in conservation form as

$$\left.\frac{\partial \mathbf{u}}{\partial t}\right|_x + \nabla \cdot \vec{\mathbf{F}}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{0}, \quad \vec{\mathbf{F}} = \vec{\mathbf{F}}^i(\mathbf{u}) - \vec{\mathbf{F}}^\nu(\mathbf{u}, \nabla \mathbf{u}), \tag{2.1}$$

where $\mathbf{u}(\vec{x}, t) \in \mathbb{R}^{\hat{s}}$ is the conservative state vector, $\vec{x} \in \mathbb{R}^d$ is the spatial coordinate, $t \in \mathbb{R}$, $\hat{s}$ is the state rank, and d is the spatial dimension. For systems involving both convection and diffusion, the total flux can be decomposed into the inviscid, $\vec{\mathbf{F}}^i$, and

viscous fluxes, $\vec{\mathbf{F}}^{\nu}$, respectively. The viscous flux often depends linearly on the state gradient. In this work, we focus on physical conservation in fluid flow, specifically the compressible Navier-Stokes equations.

2.1.1 Compressible Navier-Stokes Equations

The fluid subsystem is governed by the compressible Navier-Stokes equations. The Navier-Stokes equations describe the motion of a viscous, Newtonian fluid. By solving these equations, we can study the fluid property that is transported by the ordered motion of the flow (i.e., convection) and by the random motion of the fluid molecules (i.e., diffusion). The governing equations of the fluid arise from the conservation of mass, momentum and energy and can be written in the PDE form as Eq 2.1.

The system of governing equations for the compressible Navier-Stokes equations consists of $(d + 2)$ equations for laminar flow in d dimensions. The conservative state vector is given as, $\mathbf{u} = [\rho, \rho v_i, \rho E]$, where ρ is the density, v_i are the d components of the velocity, and E is the total energy per unit mass. The conservation equations of the fluid state are given as

$$\begin{aligned} \partial_t \rho + \partial_j \left(\rho v_j\right) &= 0, \\ \partial_t \left(\rho v_i\right) + \partial_j \left(\rho v_i v_j + p\delta_{ij}\right) - \partial_j \tau_{ij} &= 0, \\ \partial_t (\rho E) + \partial_j \left(\rho H v_j\right) - \partial_j \left(\tau_{ij} v_i - q_j\right) &= 0. \end{aligned} \tag{2.2}$$

In the above equations, i, j indexe the spatial dimension, δ_{ij} is the Kronecker delta function, with $\delta_{ij} = 1$ if $i = j$, and 0 if $i \neq j$, q_i is the heat flux and τ_{ij} are the viscous shear and normal stresses for a Newtonian fluid,

$$\begin{aligned} \tau_{ij} &= \mu \left(\partial_i v_j + \partial_j v_i\right) + \delta_{ij} \lambda \partial_m v_m, \\ q_i &= -\kappa \partial_i T, \end{aligned} \tag{2.3}$$

where m indexes the spatial dimension and κ is the thermal conductivity. The pressure, p, total enthalpy, H, and temperature, T, are related via

$$\begin{aligned} p &= (\gamma - 1)\left(\rho E - \frac{1}{2}\rho\left(v_i v_i\right)\right), \\ H &= E + p/\rho, \\ T &= \frac{p}{\rho R}. \end{aligned} \tag{2.4}$$

The relevant physical quantities for air are,

$$\begin{aligned} \text{Dynamic viscosity:} \quad \mu &= \mu_{\text{ref}}\left(\frac{T}{T_{\text{ref}}}\right)^{1.5}\left(\frac{T_{\text{ref}} + T_{\text{s}}}{T + T_{\text{s}}}\right), \\ T_{\text{ref}} &= 188.15K, T_{\text{s}} = 110K, \\ \mu_{\text{ref}} &= 1.789 \times 10^{-5} N \cdot s/m^2, \\ \text{Bulk viscosity coefficient:} \quad \lambda &= -\frac{2}{3}\mu, \\ \text{Thermal conductivity:} \quad \kappa_T &= \frac{\gamma \mu R}{(\gamma - 1)\text{Pr}}, \\ \text{Specific-heat ratio:} \quad \gamma &= 1.4, \\ \text{Prandtl number:} \quad \text{Pr} &= 0.7, \\ \text{Gas constant:} \quad R &= 287 \text{ J}/(\text{kg} \cdot \text{K}). \end{aligned} \tag{2.5}$$

In aerodynamic analysis, we are interested in non-dimensionalized quantities, i.e., drag and lift coefficients, such that CFD codes do not necessary adopt the physical units. To establish a unit-independent system, two more non-dimensional quantities are defined, namely the Reynolds number (Re) and the Mach number (M),

$$\begin{aligned} Re &= \frac{\rho |\vec{v}| L}{\mu}, \\ M &= \frac{|\vec{v}|}{a}, \quad a = \sqrt{\gamma R T}, \end{aligned} \tag{2.6}$$

where L is the characteristic length scale of the system.

2.1.2 Reynolds-Averaged Navier-Stokes Equations

At high Reynolds number, the viscous effects in the fluid flow become smaller compared to inertial forces, and small perturbations cannot be damped out effectively and may get amplified, leading to unstable flow and eventually transition to turbulent flow. Turbulence or turbulent flow is often described as a spatially varying mean flow with superimposed three-dimensional random fluctuations that are self-sustaining and enhance mixing, entertainment and dissipation [129]. The fluctuations in the velocity and pressure field caused by these eddies are inherently unsteady, and any initial condition perturbations can lead to chaotic results. There are various way of simulating turbulence in CFD, such as Large Eddy Simulation (LES), Reynolds-Averaged Naiver-Stokes (RANS) and Direct Numerical Simualtion (DNS). In this work, RANS equations are used for modeling turbulence. The RANS equation can be obtained by applying an averaging operation to the Navier-Stokes equations. These equations are very similar to the original equations but contain some additional terms in the momentum equations called Reynolds stress terms that are unknown and need to be modelled.

In this work we use the compressible Navier-Stokes equations, Reynolds-averaged with a version of the Spalart-Allmaras (SA) turbulence model that is modified for improved stability for negative values of the turbulence working variable, $\tilde{\nu}$ [7]. These modifications to the original SA turbulence model are adopted in this work as they are specifically suited for a DG discretization [24, 28]. The model involves a single governing kinematic equation to describe the viscous eddy current flow. The model was specifically derived for use in aerodynamic applications involving wall-bounded systems as well as in turbomachinery applications. The model includes multiple parameters and coefficients as part of a simulation, but because there is only one additional dynamic differential equation to solve, it is computationally efficient. The

governing equation for the kinematic viscosity is given by,

$$\underbrace{\partial_t(\rho\tilde{\nu})}_{\text{unsteady}} + \underbrace{\partial_j\left(\rho u_j\tilde{\nu}\right)}_{\text{convective}} - \underbrace{\partial_j\left[\frac{1}{\sigma}\rho\left(\nu+\tilde{\nu}f_n\right)\partial_j\tilde{\nu}\right]}_{\text{diffusive}} + \underbrace{S_{\tilde{\nu}}}_{source} = 0, \tag{2.7}$$

where the source term for the $\tilde{\nu}$ equation is given by,

$$S_{\tilde{\nu}} = \frac{1}{\sigma}\left(\nu+\tilde{\nu}f_n\right)\partial_j\rho\partial_j\tilde{\nu} - \frac{c_{b2}\rho}{\sigma}\partial_j\tilde{\nu}\partial_j\tilde{\nu} - P + D. \tag{2.8}$$

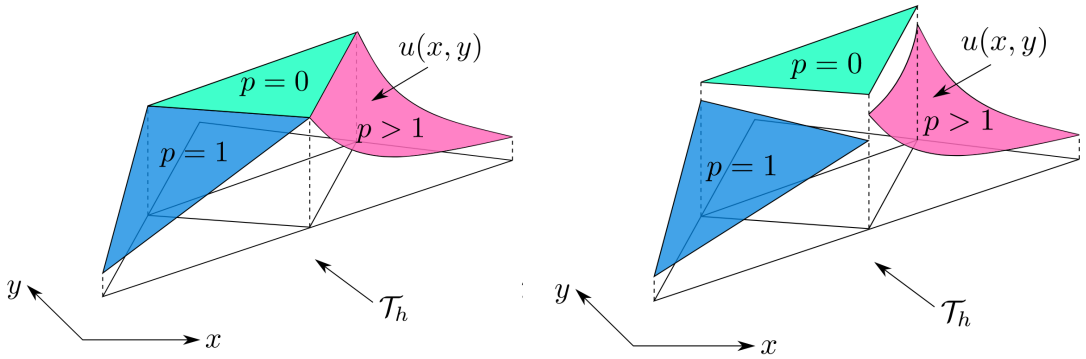
where P is the turbulence production, D is the turbulence destruction, and i, j index the spatial dimension. Further details about the production term, destruction term and closure coefficients can be found in the work by Fidkowski [42].

2.1.3 Discontinuous Galerkin Discretization

To discretize the state equations (Eq 2.1), a DG finite-element method is used in space. As a finite-element method, DG approximates the state $\mathbf{u}$ in functional form using linear combinations of basis functions on each element. No continuity constraints are imposed between adjacent elements. Denoting by T_h the set of N_e elements in a non-overlapping tessellation of the domain Ω, the state on element e, Ω_e, is approximated as

$$\mathbf{u}_h(\vec{x}(\vec{\xi}))\Big|_{\Omega_e} = \sum_{n=1}^{N_p}\mathbf{U}_{en}\phi_{en}(\vec{x}(\vec{\xi})). \tag{2.9}$$

In this equation, N_p is the number of basis functions per element, $\mathbf{U}_{en}$ is the vector of q coefficients for the n^{th} basis function on element e: $\phi_{en}(\vec{x}(\vec{\xi}))$, and $\hat{s}$ is the state rank. $\vec{x}$ denotes the global coordinates, and $\vec{\xi}$ denotes the reference-space coordinates in a master element. Formally, $\mathbf{u}_h \in \mathcal{V}_h = [\mathcal{V}_h]^q$, where, if the elements are not curved, $\mathcal{V}_h = \left\{u \in L_2(\Omega) : u|_{\Omega_e} \in \mathcal{P}^p \ \forall\Omega_e \in T_h\right\}$, and $\mathcal{P}^p$ denotes polynomials of order p on each element. Multiplying Eq 2.1 by test functions in $\mathcal{V}_h$, which are the same as the



(a) Continuous Galerkin (b) Discontinuous Galerkin

Figure 2.1: Solution approximation using the CG and DG methods. Though no continuity constraints are enforced at adjacent element boundaries, the inter-element flux is uniquely defined, as in finite volume methods.

basis functions for DG, integrating by parts on each element, and using the Roe [106] convective flux and the second form of Bassi and Rebay (BR2) [11] for the viscous treatment, we obtain the following system of nonlinear equations,

$$\mathbf{R}^f \equiv \mathbf{M}^f \frac{\mathrm{d}\mathbf{U}^f}{\mathrm{d}t} - \mathbf{r}^f = \mathbf{0}, \tag{2.10}$$

where $\mathbf{r}^f$ is the (negative) discrete spatial residual vector, $\mathbf{R}^f$ is the temporally strong-form unsteady residual, and the f superscript denotes that these equations apply to the fluids subsystem.

2.1.4 Arbitrary Lagrangian-Eulerian Formulation

In the case of a stationary domain, the fluid equations (Eq 2.1) are solved numerically in the Eulerian frame of reference, where the computational grid is fixed and the fluid moves with respect to the grid. However, numerical simulations of fluid dynamics involving a moving and/or deforming domain, such as in the case of FSI, face issues due to the lack of precise interface definition and under-resolved flow features when solved in the Eulerian frame of reference. The Lagrangian approach, on

the other hand, faces problems dealing with large distortions of the computational domain. To resolve these issues, an alternate method, the Arbitrary Lagrangian Eulerian (ALE) approach, has been introduced and is applied in the present work. A simple and effective ALE method for DG was introduced by Persson et al. [98] and a similar approach is followed in this work [65].

Let $\vec{x} = \mathscr{G}(\vec{X}, t)$ represent the one-to-one time-dependent mapping between the physical volume and the reference volume. Each point $\vec{X}$ in the static reference domain is mapped to a corresponding point $\vec{x}$ in the physical domain, based on the desired deformation of the mesh. The spatial Jacobian of the mapping, represented by $\underline{G}$, and the mapping velocity, $\vec{v}_X$, are given by

$$\underline{G} = \nabla_X \mathscr{G}, \quad \vec{v}_X = \frac{\partial \mathscr{G}}{\partial t}\Big|_X. \tag{2.11}$$

Let g=det($\underline{\mathrm{G}}$). The corresponding Navier-Stokes equations in the reference frame can be written as

$$\frac{\partial \mathbf{u}_X}{\partial t}\Big|_X + \nabla_X \cdot \vec{\mathbf{F}}_X(\mathbf{u}_X, \nabla_X \mathbf{u}_X) = \mathbf{0}, \quad \vec{\mathbf{F}}_X = \vec{\mathbf{F}}^i_X(\mathbf{u}_X) - \vec{\mathbf{F}}^\nu_X(\mathbf{u}_X, \nabla_X \mathbf{u}_X), \tag{2.12}$$

where the transformed vectors, derivatives, and fluxes in the reference frame are given by

$$\mathbf{u}_X = g\mathbf{u}, \tag{2.13}$$

$$\nabla_x \mathbf{u} = \nabla_X (g^{-1}\mathbf{u}_X)\underline{G}^{-T} = (g^{-1}\nabla_X \mathbf{u}_X - \mathbf{u}_X \nabla_X (g^{-1}))\underline{G}^{-T}, \tag{2.14}$$

$$\vec{\mathbf{F}}^i_X = g\underline{G}^{-1}\vec{\mathbf{F}}^i - \mathbf{u}_X \underline{G}^{-1}\vec{v}_X, \qquad \vec{\mathbf{F}}^\nu_X = g\underline{G}^{-1}\vec{\mathbf{F}}^\nu. \tag{2.15}$$

The explicit definition of the grid velocities requires knowledge of the mapping in advance, which is not available when solving for the coupled system in a partitioned approach. This issue is circumvented by assuming the structural subsystem to be

stationary when solving for the unsteady fluid subsystem. This assumption helps in providing the mapping, used to define the grid position and velocities, which is lagging in time and is corrected using the predictor-corrector step, as described in the temporal coupling section. To simplify the notation, in the following sections, the X subscript is dropped from the state and fluxes when referring to the fluid system, which is solved on a reference static domain but is mapped to the physical domain for coupling with the structural solver.

2.1.5 Mesh Deformation

For deforming domains, the ALE formulation of the Navier-Stokes equations requires a mapping between the reference and the deformed physical mesh. The mapping interpolates the boundary displacements to the interior of the fluid mesh. Several mesh deformation methods exist in the literature [111], and these can be classified into two main categories, 1) physical-analogy based techniques and 2) interpolation based techniques. Physical analogy methods [13] typically model each edge of the mesh as a spring with an individual stiffness value. On the other hand, interpolation based methods compute the movement of grid nodes as a function of boundary nodes, with no attached physical meaning. Radial basis function interpolation [29] and inverse distance methods [131] are some examples of interpolation-based techniques. An alternative to these methods is the use of an explicit expression for the mapping between the reference and physical domains, as introduced by Persson et al. [100]. Two different mesh deformation approaches:1) An explicit mapping approach and an 2) interpolation based approach have been used in this work, which are described below.

2.1.5.1 Explicit Mapping

The explicit mapping approach, as presented by Persson et al. [100], does not require solving a system of equations for deforming the volume. It uses explicit expressions for the mapping that blend the motion, smoothly reducing it to the identity mapping away from the boundary. In this work, explicit mapping is used as the mesh-deformation algorithm for two-dimensional aeroelastic cases undergoing rigid body deformation. For any rigid body deformation, the mesh motion algorithm divides the entire spatial domain into two regions based on inner and outer radii of the blending region. The region extending up to the inner radius from the center of the deformation marks the region of rigid deformation. Within this region, any deformation provided by the user is applied to all of the mesh elements without any blending. The presence of the rigid region prevents errors such as mesh element inversion in highly-stretched elements, which face such errors when placed in a blending region. The blending region, which exists between the inner and outer radii from the center of the motion, uses a polynomial function to blend the deformation radially

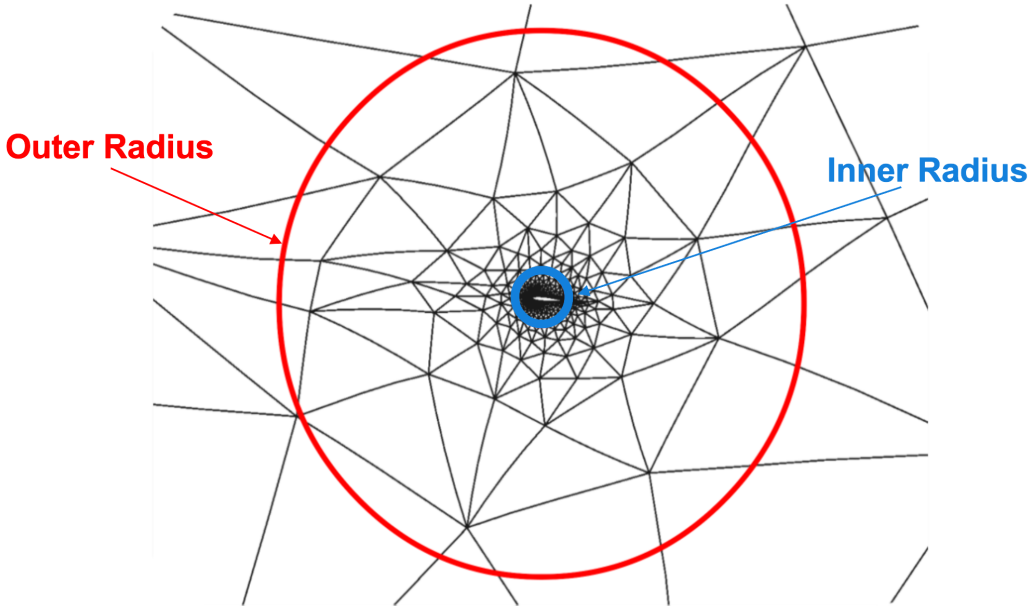


Figure 2.2: Example of inner and outer radius of the blending region for the explicit mapping mesh motion.

such that the deformation goes to zero at the outer radius. To achieve blending of the motion, polynomial blending functions $P(r)$ of odd degrees are used. The three different blending polynomials analyzed in this study are cubic, $P(r) = 3r^2 - 2r^3$, quintic, $P(r) = 10r^3 - 15r^4 + 6r^5$, and septic, $P(r) = -20r^7 + 70r^6 - 84r^5 + 35r^4$, where r is the normalized radial distance from the inner radius. Figure 2.2 shows the deformed mesh for an airfoil undergoing rigid-body pitch motion. The inner and outer radii are placed at one and ten chords away from the center of motion, which is at the quarter chord of the airfoil.

2.1.5.2 Radial Basis Function

Aeroelastic simulations in which an explicit mapping of the displacements is not readily available, benefit from an interpolation based method to deform the fluid mesh. Mesh motion using RBF's, an example of interpolation based methods, have been previously demonstrated to produce high quality meshes suitable for use within unsteady and aeroelastic FSI codes [29]. In this work, RBF's based on Wendland's C_0 continuous basis function [15] are used to deform fluid meshes in both two and three-dimensional aeroelastic problems. A more detailed description on RBF's can be found in Section 3.1.1.

2.1.6 Errors Generated by Mesh Deformation

Accommodating a prescribed mesh deformation by moving mesh nodes introduces errors into a fluid simulation. Mesh deformation algorithms generate spatial and temporal errors in a fluid simulation by two main mechanisms. First, the spatial dependence of the motion required to accommodate the prescribed mesh deformation, provided by the structural subsystem, causes the elements in the fluid mesh to distort, which reduces their spatial approximation capability [48]. This is due to the reference-space polynomial approximation space no longer being polynomial after a

nonlinear mapping into global space. These errors exist for both steady and unsteady simulations as they exist in the presence of any non-trivial deformation of the mesh. Second, for unsteady simulations involving mesh deformation, grid velocities are also introduced in the mesh, which are arbitrary in spatial variation and depend on the mesh motion parameters. Under-integration of terms in the weak form involving mesh motion, due to the use of finite quadrature and increased complexity of the integrands affected by mesh motion derivatives and velocities, introduces spatial and temporal errors in the simulation.

A solution for this inaccuracy, introduced by Lesoinne and Farhat [71], is known as enforcement of the Geometric Conservation Law (GCL), which conserves uniform flow by linearizing the mapping from the reference to the physical domain. However, GCL is not explicitly enforced in this work. Previous research [65, 98] discussing the relevance of the GCL has shown that for outputs on adapted meshes and even for outputs corrected by error estimates, the GCL is not critical. Even though the GCL explicitly enforces conservation, the errors made without it are on the order of discretization errors, which reduce with further refinement, making the ALE scheme asymptotically conservative. As the complexity of its implementation (particularly in the adjoint discretization) outweigh its benefits, it is thus not implemented in this work.

2.2 Structural Subsystem

The structural subsystem is governed by a set of partial differential equations derived on the basis of continuum modeling with a general non-isotropic constitutive law. In this work, deformations occurring in the structures are assumed to be small and a linear isotropic structure model is used. Based on Newton's second law, the

equations of linear elasticity are given as,

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}. \tag{2.16}$$

where ρ is the density of the material, $\mathbf{u}$ is the displacement vector, $\boldsymbol{\sigma}$ represents the stresses, and $\mathbf{f}$ represents the external forces acting on the structure. For linear elastic solids, the stress-strain relation is

$$\boldsymbol{\sigma} = \lambda(\nabla \cdot \mathbf{u})\mathbf{I} + 2\mu\varepsilon \tag{2.17}$$

where λ and μ are Lame parameters, and

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}}\right) \tag{2.18}$$

2.2.1 Continuous Galerkin Discretization

To discretize the governing equation (Eq 2.16), a CG finite-element method is used. The finite element approximation to a problem starts by dividing the domain of interest, Ω, into a set of subdomains (called elements), Ω_e, such that

$$\Omega \approx \hat{\Omega} = \sum \Omega_e \tag{2.19}$$

The governing equations can be presented in a strong form as a set of partial differential equations (Eq 2.16) or alternatively in terms of a variational principle or weak form expressed as an integral over the domain of interest. The weak form can be used to construct approximate solutions based on the finite-element method. As a finite-element method, CG approximates the state $\mathbf{u}$ in a functional form by using

linear combinations of basis functions on each element which is given by,

$$\mathbf{u}(\mathbf{x},t) = \sum_{b} N_b(\mathbf{x})\tilde{\mathbf{u}}_b(t) = \mathbf{N}(\mathbf{x})\tilde{\mathbf{u}}(t) \tag{2.20}$$

where b is the number of nodes per element, N_b are element shape functions, and $\tilde{\mathbf{u}}_b(t)$ are time dependent nodal displacements. The strain–displacement equation (Eq 2.18) can be approximated as

$$\boldsymbol{\varepsilon} = \sum \mathbf{B}_b\tilde{\mathbf{u}}_b = \mathbf{B}\tilde{\mathbf{u}}. \tag{2.21}$$

Discretizing the governing equation using CG requires that the shape functions within an element be continuous up through the highest derivative occurring in the functional used in variational principle and are continuous up through one order less than highest derivative appearing in functional across element boundaries. Using CG to discretize in space, the semi discrete form of the equation is given by,

$$\mathbf{M}\ddot{\mathbf{u}}^s + \mathbf{C}\dot{\mathbf{u}}^s + \mathbf{K}\mathbf{u}^s = \mathbf{F}^s, \tag{2.22}$$

where $\mathbf{u}^s$ represents the vector of displacements, $\mathbf{M}$ is the mass matrix, and $\mathbf{K}$ is the stiffness matrix of the structure. $\mathbf{C}$ denotes the internal or external damping in the structure, and $\mathbf{F}^s$ represents the vector of external forces acting on the structure. A classic Rayleigh damping, where the damping matrix is a linear combination of the mass and stiffness matrices $\mathbf{M}$ and $\mathbf{K}$, is used in this work. These vectors and matrices in the semi-discrete form are defined as

$$\mathbf{M} = \sum_{e} \mathbf{M}^{(e)}; \quad \mathbf{K} = \sum_{e} \mathbf{K}^{(e)} \quad \text{and} \quad \mathbf{F}^s = \sum_{e} \mathbf{F}^{s(e)} \tag{2.23}$$

where

$$\begin{aligned} \mathbf{M}^{(e)} &= \int_{\Omega_e} \mathbf{N}^T \rho \mathbf{N} \mathrm{d}\Omega \\ \mathbf{K}^{(e)} &= \int_{\Omega_e} \mathbf{B}^T \boldsymbol{\sigma} \mathrm{d}\Omega \\ \mathbf{F}^{s(e)} &= \int_{\Omega_e} \mathbf{N}^T \mathbf{f} \mathrm{d}\Omega \end{aligned} \tag{2.24}$$

The semi-discrete form can be re-written as a system of two first-order ordinary differential equations

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \dot{\mathbf{U}}^s + \begin{bmatrix} \mathbf{C} & \mathbf{K} \\ -\mathbf{1} & \mathbf{0} \end{bmatrix} \mathbf{U}^s = \begin{bmatrix} \mathbf{F}^s \\ \mathbf{0} \end{bmatrix} \quad \text{where} \quad \mathbf{U}^s = \begin{bmatrix} \dot{\mathbf{u}}^s \\ \mathbf{u}^s \end{bmatrix}, \tag{2.25}$$

where $\mathbf{1}$ denotes the identity matrix. The governing equations can be written in an abbreviated form as:

$$\mathbf{R}^s \equiv \mathbf{M}^s \frac{\mathrm{d}\mathbf{U}^s}{\mathrm{d}t} - \mathbf{r}^s = \mathbf{0}, \tag{2.26}$$

where $\mathbf{r}^s$ is the (negative) discrete spatial residual vector, and $\mathbf{R}^s$ is the strong-form unsteady residual. The superscript s denotes that these equations apply to the structures subsystem.

2.3 Temporal Discretization

The method of lines is used to evolve the fluid and structure partial differential equations in time, by first discretizing in space as described in the previous sections. This results in the semi-discrete form of the governing equation, as shown in Eq 2.10 and Eq 2.26, which are then integrated using standard temporal schemes. To maintain the high-order accuracy of the individual subsystems in the FSI problem, the fluid and structural governing equations are discretized with a high-order temporal scheme. Following the work of Bijl and Carpenter [17], a stiffly accurate explicit, singly diagonally implicit Runge–Kutta (ESDIRK) scheme [66], for which the solution at the next time step can be made of arbitrary high order by cancellation of the

lower order errors, is used in this work. The ESDIRK scheme is an L-stable, implicit scheme with an explicit first stage, which allows the implicit stages to be second order accurate. Consider a first-order semi-discrete system describing the fluid/structural system,

$$\dot{\mathbf{u}} = \mathbf{F}(\mathbf{u}, t), \tag{2.27}$$

where $\mathbf{u}$ is the state and $\mathbf{F}$ is the flux. For a p stage Runge-Kutta scheme, at every stage k,

$$\mathbf{u}^k = \mathbf{u}^n + \Delta t \sum_{i=1}^{k} a_{ki} \mathbf{F}^i, \quad k = 1 \dots p, \tag{2.28}$$

where $\mathbf{u}^n \simeq \mathbf{u}(t^n)$, $\mathbf{u}^k \simeq \mathbf{u}(t^n + c_k \Delta t)$ is the value of the state $\mathbf{u}$-vector on the k th-stage, $\mathbf{u}^{n+1} \simeq \mathbf{u}(t^n + \Delta t)$ and $\mathbf{F}^i$ is the flux at stage i. After computing p stages, the solution at the next time level is found by

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \sum_{i=1}^{p} b_i \mathbf{F}^{(i)}. \tag{2.29}$$

Runge-Kutta methods are generally written as a Butcher tableau [20]:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1p} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{p1} & a_{p2} & \cdots & a_{pp} \\ \hline & b_1 & b_2 & \cdots & b_p \end{array} . \tag{2.30}$$

Each of the respective Butcher coefficients a_{ij}, b_i and c_i are constrained, at a minimum, by certain order of accuracy and stability considerations. In this work, a fourth-order time-accurate ESDIRK scheme is used for which the Butcher coefficients are shown in Table 2.1. Both the fluid and structural subsystems are marched forward in time using the high-order ESDIRK4 scheme. Due to the multiple linear solves needed

per time step, the computational cost of ESDIRK methods is higher per time step compared to an explicit method. However, this stability of implicit methods makes it ideal for handling the stiffness of a high-order discretizations.

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
$\frac{83}{250}$	$\frac{8611}{62500}$	$\frac{-1743}{31250}$	$\frac{1}{4}$	0	0	0
$\frac{31}{50}$	$\frac{5012029}{34652500}$	$\frac{-654441}{2922500}$	$\frac{174375}{388108}$	$\frac{1}{4}$	0	0
$\frac{17}{20}$	$\frac{15267082809}{155376265600}$	$\frac{-71443401}{120774400}$	$\frac{730878875}{902184768}$	$\frac{2285395}{8070912}$	$\frac{1}{4}$	0
1	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$\frac{-2260}{8211}$	$\frac{1}{4}$
b_i	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$\frac{-2260}{8211}$	$\frac{1}{4}$

Table 2.1: Butcher coefficients for the fourth-order ESDIRK scheme.

CHAPTER III

Fluid-Structure Interaction

The development of a high-fidelity partitioned FSI solver depends not only on the spatial and temporal discretization of the individual subsystems but also on the spatial and temporal coupling of the two subsystems. The coupling between the fluid and the structural subsystems plays an important role in the stability and accuracy of FSI simulations. This chapter firstly reviews the spatial coupling between the two subsystems i.e. the transfer of the high-fidelity pressure at the fluid nodes at the interface, from the fluid to the structural solver and the nodal displacements and velocities at the structural nodes at the interface, from the structural to the fluid solver. Tight coupling techniques that require multiple flow and structure solutions per time step are used in this work to couple the two solver temporally. The high-order temporal coupling between the two subsystems is discussed next where the Implicit-Explicit Runge-Kutta schemes are reviewed.

3.1 Spatial Coupling

In a partitioned FSI solver, where the fluid and the structural subsystems are solved using separate solvers, the fluid and the structural meshes at the interface may or may not align. In the case that the fluid and the structure mesh, along the fluid-structure interface, are face-wise matching, spatially coupling the two subsystems is

simpler. This is because of two main reasons: First, boundary data are easily passed between the fluid and structure meshes, at either the solution nodes or Gauss integration points, by a simple pre-computed lookup table and without any interpolation. Second, the deformed fluid mesh is easily constructed to be exactly conformal to a given deformed structure position. The major drawback of this approach is that we are no longer free to choose the fluid and structural meshes independently. This also adds constrains on the element size on the boundary. For example, if the discretized fluid requires many small elements near the boundary to resolve fine features, the structure may be required to have an undesirably large number of elements. Secondly, such an approach requires the use of similar polynomial order to discretize the fluid and structural subsystems at the interface, which is not suitable in an *hp* adaptive framework. The focus of this work is to spatially couple the fluid and structural meshes at the interface when the faces are non-matching and dissimilar polynomial orders are used to discretize the fluid and structural subsystem.

Consider the fluid and the structural meshes having a common continuous interface, Γ, with non-matching nodes at the interface. The partitioned FSI problem can be viewed as a two-field problem with jump conditions along the interface which need to be satisfied in the form of the kinematic and dynamic continuity conditions. The continuity conditions at the interface are given by,

$$\mathbf{u}^f = \mathbf{u}^s \quad \text{on} \quad \Gamma, \tag{3.1}$$

$$p^s \mathbf{n}^s = p^f \mathbf{n}^f \quad \text{on} \quad \Gamma, \tag{3.2}$$

where $\mathbf{u}^{f,s}$ is the displacement vector, $p^{f,s}$ is the pressure or stress tensor and $\mathbf{n}^{f,s}$ the outward normal of the flow and structure interface, respectively. These two conditions ensure the location of the interface boundary to be the same for both subsystems and

the forces on the wetted surface of the structure to be in equilibrium with those on the fluid side.

3.1.1 Displacement Transfer

A variety of coupling methods [113] exist in the literature which can be utilized to satisfy the continuity of the displacement, as shown in Eq. 3.1. These coupling methods can be generalised in a discrete form as

$$\mathbf{U}^f = H^{fs}\mathbf{U}^s, \tag{3.3}$$

where $\mathbf{U}^s$ and $\mathbf{U}^f$ are the vectors of displacements at the interface of the structural and fluid mesh and H^{fs} is a $(n^f \times n^s)$ transformation matrix between the flow and structure interface. n^f and n^s represent the number of nodes at the interface of the fluid and structural meshes, respectively. In this work, the class of multivariate interpolation based on the idea of using a global interpolation function is used to transfer information. Multivariate interpolation requires no connectivity information and is therefore well-suited to the transfer of information between interface meshes with arbitrary geometric mismatches. Radial basis function (RBF), an example of multivariate interpolation is used in this work. RBF's are based on fitting a series of splines, or basis functions to interpolate information from one point cloud to another and are easy to implement. The displacement at both the fluid and structure interface is approximated by a sum of basis functions:

$$\mathbf{u}^j(x) = \sum_{i=1}^{N} \alpha_i \phi(||x - x_i||) \quad j = f, s \tag{3.4}$$

where α_i are RBF coefficients, ϕ is a basis function, x is position in the interface of the fluid mesh where the displacement is to be evaluated and x_i is position in the interface of the structural mesh where the displacement is known. The coefficients α_i

are found by requiring exact recovery of the original function at these points x_i. The discrete structural displacement, as shown in Eq 3.4, can be written as,

$$\mathbf{u}^s = M^{ss}\boldsymbol{\alpha}, \tag{3.5}$$

where $\boldsymbol{\alpha}$ the vector containing the coefficients α_i and M^{ss} is an $n_s \times n_s$ matrix containing the evaluation of the basis function $\phi^{s_i s_j} = \phi\left(\left\|\mathbf{x}_{s_i} - \mathbf{x}_{s_j}\right\|\right)$. The fluid displacements can be evaluated in a similar fashion:

$$\begin{aligned} \mathbf{u}^f &= M^{fs}\boldsymbol{\alpha} \\ \mathbf{u}^f &= \underbrace{M^{fs}[M^{ss}]^{-1}}_{H^{fs}}\mathbf{u}^s \end{aligned} \tag{3.6}$$

Solving the above linear system provides the fluid displacements. Many different radial basis functions have been proposed and analyzed in the literature [19]. Some of the most commonly used function are listed below

- Gaussian spline: $\phi(|\mathbf{x}|) = e^{-|\mathbf{x}|^2/r^2}$,
- Thin Plate spline: $\phi(|\mathbf{x}|) = |\mathbf{x}/r|^2 \ln|\mathbf{x}/r|$,
- Inverted Multi-Quadric biharmonic spline $\phi(|\mathbf{x}|) = \frac{1}{\sqrt{|\mathbf{x}|^2+r^2}}$,
- Beckert and Wendland C^0 spline: $\phi(|\mathbf{x}|) = ((1 - |\mathbf{x}|/, r))^2$

where r is the distance of a point on the interface of the fluid mesh to the center of the RBF. The first three bases have global support, while the last one has compact support. The advantage of using a basis with compact support is that the transformation matrix, H^{fs}, is sparse and a considerable speed-up may be obtained in both the matrix–vector multiplication and the solution of the linear system (Eq.3.6). Wendland's C^0 continuous basis function [15] based on polynomials with a local support radius are used in this work.

3.1.2 Force Transfer

The interpolation methods discussed above for transferring displacements can also be applied to transfer stresses (Eq 3.2). Conservation of energy at the interface has made interpolation methods, such as RBF, a popular coupling approach [73, 103]. However, such methods are not well-suited for transferring discontinuous fields. Consider an adaptive FSI solver, which uses the DG discretization of the Navier-Stokes equations. As continuity constraints are not enforced at adjacent element boundaries on the fluid mesh, the corresponding fluid states are discontinuous. Additionally, in a *hp*-adaptive framework, the state within a fluid element can be approximated using different spatial order polynomials. Thus, for an adaptive FSI solver, the pressure distribution on the FSI interface will not only be discontinuous but may also have locally varying polynomial representations. Approximating such a pressure distribution using global interpolation methods will degrade the local high-order nature of the states in the elements approximated using high-order polynomials. This is due to the discretization errors in the lower-order elements polluting the solution in the high-order elements via interpolation. In order to accurately transfer the locally varying pressure distribution on the interface without violating conservation of energy, a local interpolation strategy is used. The discontinuous pressure field is transferred to the structural solver in two steps. Firstly, for every overlapping fluid element at the interface, the pressure is evaluated at the Gauss-quadrature points on the fluid edge/face that is located at the interface. The pressure at the Gauss-quadrature points is then passed to the structural solver, which reconstructs the local pressure distribution within a fluid element using spline interpolation. Once the pressure is reconstructed locally in a fluid element, its contribution to the overlapping element is evaluated using the principle of virtual work.

After the transfer of the discontinuous pressure distribution at the interface, the

overlapping fluid elements are identified for every structural element at the interface. Due to non-matching faces, more than one fluid element can contribute to a single structural element. Consider a one-dimensional structural element s_e at the FSI interface, which extends from $[x^{s_e}, x^{s_{e+1}}]$. The pressure distribution over the structural element is defined as:

$$p(x)^{s_e} = \begin{cases} p(x)^{f_1} & \text{for x} \in [x^{s_e}, x^{f_2}] \\ p(x)^{f_2} & \text{for x} \in [x^{f_2}, x^{f_3}] \\ \dots & \\ p(x)^{f_n} & \text{for x} \in [x^{f_n}, x^{s_{e+1}}] \end{cases} \tag{3.7}$$

where n is the number of overlapping fluid elements. Each overlapping fluid element, f_e extends in the one-dimensional space in $[x^{f_e}, x^{f_{e+1}}]$. The pressure distributions on overlapping fluid elements are lumped onto the structural nodes in a conservative manner, by using the principle of virtual work. Thus, the lumped force vector on the structural element is given as,

$$\vec{f}^{e}_{N^s \times 1} = \int_{x^{s_e}}^{x^{s_{e+1}}} p(x)^{s_e} \left(\mathbb{N}^e(\vec{x})\right)^T_{N^s \times 1} \, dx, \tag{3.8}$$

where $\vec{f}^{e}_{N^s \times 1}$ is the force vector obtained on the element after lumping, N^s is the number of shape function per element, and $\mathbb{N}^e$ are the shape functions used to approximate the state within a structural element. Due to the discontinuous nature of the pressure distribution, the force vector can be re-written as,

$$\vec{f}^{e}_{N^s \times 1} = \sum_{i=1}^{n} \int_{\max(x^{s_e}, x^{f_i})}^{\min(x^{s_{e+1}}, x^{f_{i+1}})} p(x)^{f_i} \left(\mathbb{N}^e(\vec{x})\right)^T_{N^s \times 1} \, dx. \tag{3.9}$$

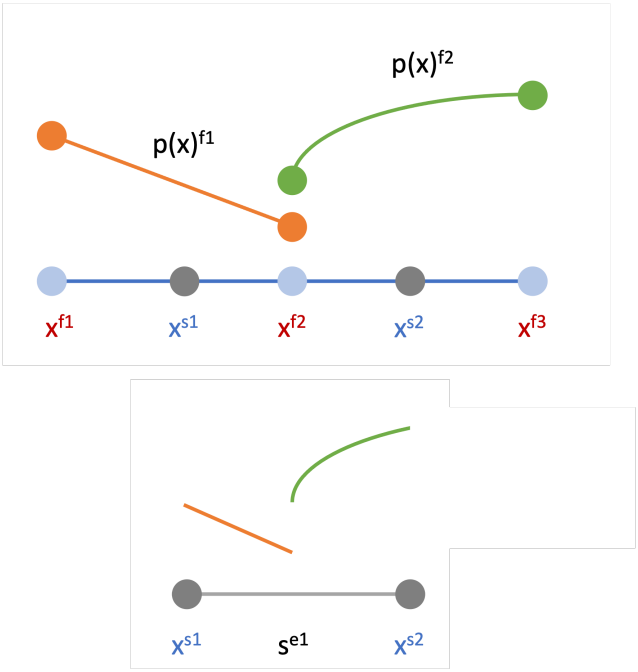


Figure 3.1: Consider a structural element at the FSI interface on which the force vector needs to be evaluated. The overlapping fluid elements are firstly identified and projected to the structural element. The overlapping pressure distributions are then lumped to create the force vector for the element by integrating with the elemental basis functions.

This procedure is repeated for every structural element at the interface to get the exact distribution of forces on the structure. A visual representation of the methodology is shown in Figure 3.1 where two fluid elements with dissimilar spatial order approximations of the pressure, $p(x)^{f1}$ and $p(x)^{f2}$, overlap a structural element, s^{e1}.

3.2 Implicit-Explicit (IMEX) Runge-Kutta Schemes

The time integration method used to couple the fluid and structural subsystems temporally in this work is the IMEX scheme [122]. IMEX is a high-order time integration scheme developed for stiff time marching problems in which the rate of change of the state can be split into stiff and non-stiff components. The stiff component of the state time derivative, a term that can lead to rapid variation in the solution, is then solved implicitly in time, while the non-stiff component is solved explicitly. To

demonstrate the scheme, consider a semi-discretized multiphysics equation,

$$\mathbf{M}\dot{\mathbf{u}} = \mathbf{r}(\mathbf{u}, t) = \mathbf{f}(\mathbf{u}, t) + \mathbf{g}(\mathbf{u}, t), \tag{3.10}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{u}$ is the state vector, and $\mathbf{r}$ is the residual, which is split into $\mathbf{f}$ and $\mathbf{g}$, the non-stiff and stiff component respectively. $\mathbf{f}$ is integrated with an n-stage explicit Runge-Kutta scheme and $\mathbf{g}$ is integrated with an n-stage diagonally-implicit Runge-Kutta scheme. The equations for an update in one time step are

$$\mathbf{u}_{n,j} = \mathbf{u}_{n-1} + \sum_{p=1}^{j-1} \hat{a}_{jp}\hat{k}_{n,p} + \sum_{p=1}^{j} a_{jp}k_{n,p} \tag{3.11}$$

$$\mathbf{M}\hat{k}_{n,j} = \Delta t_n \mathbf{f}(\mathbf{u}_{n,j}, t_{n-1} + \hat{c}_j \Delta t_n) \tag{3.12}$$

$$\mathbf{M}k_{n,j} = \Delta t_n \mathbf{g}(\mathbf{u}_{n,j}, t_{n-1} + c_j \Delta t_n) \tag{3.13}$$

$$\mathbf{u}_n = \mathbf{u}_{n-1} + \sum_{p=1}^{s} \hat{b}_p\hat{k}_{n,p} + \sum_{p=1}^{s} b_p k_{n,p} \tag{3.14}$$

The IMEX scheme considered in this work consists of a fourth-order explicit Runge-Kutta (ERK) and a stiffly-accurate fourth-order, ESDIRK scheme. The Butcher tableaus for the ERK and the ESDIRK schemes are shown in Table 3.1 and Table 2.1, respectively.

3.3 Temporal Coupling

In a partitioned procedure for FSI or aeroelastic computations, the fluid and structure subsystems are time-integrated by different schemes that are tailored to their different mathematical models, and solved in a staggered numerical algorithm. Conventional serial staggered and improved serial partitioned solution procedure [38] are examples of partitioned algorithms widely used for solving FSI problems. The staggered solution algorithm supporting this partitioned procedure can also be described

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0
$\frac{83}{250}$	$\frac{13861}{62500}$	$\frac{6889}{62500}$	0	0	0	0
$\frac{31}{50}$	$\frac{-116923316275}{2393684061468}$	$\frac{-2731218467317}{15368042101831}$	$\frac{9408046702089}{11113171139209}$	0	0	0
$\frac{17}{20}$	$\frac{-451086348788}{2902428689909}$	$\frac{-2682348792572}{7519795681897}$	$\frac{12662868775082}{11960479115383}$	$\frac{3355817975965}{11060851509271}$	0	0
1	$\frac{647845179188}{3216320057751}$	$\frac{73281519250}{8382639484533}$	$\frac{552539513391}{3454668386233}$	$\frac{3354512671639}{8306763924573}$	$\frac{4040}{17871}$	0
b_i	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$\frac{-2260}{8211}$	$\frac{1}{4}$

Table 3.1: Butcher coefficients for the fourth-order explicit Runge-Kutta scheme.

as a loosely-coupled solution algorithm due to the information exchange occurring only once every few time steps. Farhat et. al [39] were the first to develop a second-order time accurate loosely coupled partitioned scheme. Despite the simplicity of their implementation, loosely-coupled solution algorithms are often criticized in the literature for their lack of sufficient time-accuracy and sufficient numerical stability [38]. As the objective of this work is to develop a high-fidelity FSI solver, high-order coupling schemes which maintain the temporal accuracy of the coupled solver are required. Implicit–Explicit (IMEX) Runge–Kutta scheme [122] offers arbitrarily high orders of temporal accuracy for multi-physics simulations and the ability to use implicit solvers for both the fluid and the structural subsystems. This scheme couples the two subsystems tightly in time, where information exchange occurs multiple times every time step and is used in the high-fidelity FSI solver.

To showcase the application of IMEX on an FSI system, consider a coupled FSI problem in semi-discrete form as

$$\mathbf{M}\dot{\mathbf{u}} = \mathbf{r}(\mathbf{u}), \tag{3.15}$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}^f \\ \mathbf{u}^s \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} \mathbf{r}^f(\mathbf{u}^f; \mathbf{z}(\mathbf{u}^s)) \\ \mathbf{r}^s(\mathbf{u}^s; \mathbf{t}(\mathbf{u}^f)) \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}^f & 0 \\ 0 & \mathbf{M}^s \end{bmatrix}, \tag{3.16}$$

where $\mathbf{u}$ is a combined vector of the fluid and structural states, and $\mathbf{z}$ and $\mathbf{t}$ denote the terms in the residual responsible for the coupling of the two subsystems. The unsteady residual of the structural subsystem, given in Eq 2.26, may be separated as

$$\mathbf{r}^s(\mathbf{u}^s; \mathbf{t}(\mathbf{u}^f)) = \mathbf{r}^{ss}(\mathbf{u}^s) + \mathbf{r}^{sf}(\mathbf{t}(\mathbf{u}^f)). \tag{3.17}$$

The first term represents the effect of the current structural state, while the second term represents the effect of the traction, $\mathbf{t}$, from the fluid. Since the second term is linear in $\mathbf{t}$, a predicted value of the traction is introduced, $\tilde{\mathbf{t}}$, as presented in Froehle and Persson [50]. Thus, the residual can be re-written as

$$\mathbf{r}^s(\mathbf{u}^s; \mathbf{t}(\mathbf{u}^s)) = \mathbf{r}^s(\mathbf{u}^s; \tilde{\mathbf{t}}) + \mathbf{r}^{sf}(\mathbf{t}(\mathbf{u}^f) - \tilde{\mathbf{t}}). \tag{3.18}$$

Using the above formulation, Eq 3.15 can be split as

$$\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \begin{bmatrix} 0 \\ \mathbf{r}^{sf}(\mathbf{t}(\mathbf{u}^f) - \tilde{\mathbf{t}}) \end{bmatrix} + \begin{bmatrix} \mathbf{r}^f(\mathbf{u}^f; \mathbf{z}(\mathbf{u}^s)) \\ \mathbf{r}^s(\mathbf{u}^s; \tilde{\mathbf{t}}) \end{bmatrix}. \tag{3.19}$$

With the introduction of a predictor, the coupled problem has been suitably modified into a form where the high-order IMEX scheme can be applied. As already mentioned, in the IMEX scheme the non-stiff (first) term is integrated explicitly and the stiff (second) term is integrated implicitly in time. However, the scheme differs slightly from IMEX as the evaluation of the explicit terms $\mathbf{r}^{sf}$ is avoided and instead the stage flux is updated for the structure equation using the corrected value of the coupling $\mathbf{t}(\mathbf{u}^f)$. To solve the subsystems implicitly, ESDIRK4, a fourth-order time accurate implicit scheme, is employed. An explicit traction predictor proposed by Van Zuijlen

et al. [123] for the structural subsystem, at an implicit stage i, is given by

$$\tilde{\mathbf{t}} = \sum_{j=1}^{i-1} \frac{\hat{a}_{ij} - a_{ij}}{a_{ii}} \mathbf{t}_j, \tag{3.20}$$

where $\hat{a}_{ij}$ and a_{ij} are the coefficients of the explicit and implicit Runge-Kutta integration schemes, respectively.

Figure 3.2 summarizes a single time step of the coupled solver. The coupled FSI problem uses a block Gauss-Seidel partitioning, in which the structural subsystem uses the explicit traction predictor (Step 1) and is integrated first, in an implicit manner (Step 2). The spatial coupling algorithm transfers the predicted interface data to the fluid subsystem (Step 3). The predicted interface data are used by the mesh-motion algorithm to deform the fluid mesh and are then implicitly integrated (Step 4). The corrected traction obtained from the fluid solution is lastly used to correct the structural subsystem (Steps 5 and 6).

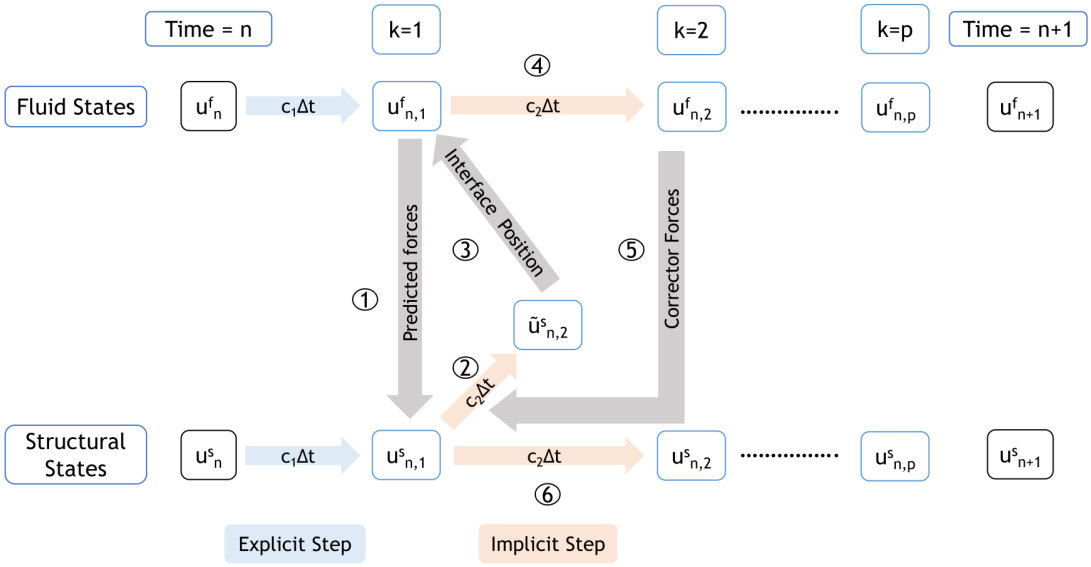


Figure 3.2: IMEX algorithm for one coupled time step.

CHAPTER IV

Output-Based Mesh Adaptation

This chapter presents a method for coupled mesh adaptation in FSI with the objective of direct minimization of output error in an FSI simulation. Output-based error estimation using discrete in space but continuous in time coupled adjoints are firstly reviewed. The chapter ends with a review of output-based mesh adaptation, where localized error estimates guide the mesh adaptation in both the fluid and structural meshes.

4.1 Output-Based Error Estimation

In the coupled FSI problem, the numerical error in an output of interest results from discretization errors generated in both the fluid and structural subsystems. These discretization errors can be reduced by uniformly refining the spatial and temporal mesh. However, resolving all regions of the fluid flow along with the the structure with equal precision is both unnecessary and inefficient. A better strategy is to (i) estimate the amount of error in the output, (ii) determine where this error originates from, and (iii) drive that error down by targeting for refinement the regions of the mesh responsible for it. Output-based error estimation techniques [93, 114, 74] identify all areas of the domain that are important for the accurate prediction of an engineering output.

The success of an output-based mesh adaptation hinges on the quality of the output error estimate. Accurate a-posteriori error estimates can be obtained by solving an adjoint problem [43] for the outputs of interest, and this is the strategy we adopt here. An adjoint provides the sensitivity of an output to perturbations in the residuals of the governing equations, which in the context of error estimation can identify regions of the domain contributing most to the output error. These regions can then be adapted to reduce the error and to obtain a more accurate output. This concept has been utilized in CFD [46, 47, 65] and monolithic FSI [104, 130, 6, 36] problems and is extended for high-order partitioned FSI solvers in this work.

4.1.1 Continuous-in-Time Adjoint Evaluation

Consider an unsteady output of the form

$$\bar{J} = \int_0^T J(\mathbf{U}^f, \mathbf{U}^s)\ dt, \tag{4.1}$$

where J is a functional of the fluid, T is final time of the simulation, $\mathbf{U}^f$, and structural, $\mathbf{U}^s$, states. The time-continuous adjoints, $\mathbf{\Psi}^f$ and $\mathbf{\Psi}^s$, represent the sensitivity of the output to perturbations in the unsteady residuals $\mathbf{R}^f$ (Eq 2.10) and $\mathbf{R}^s$ (Eq 2.26), respectively. To derive the adjoint equations, a Lagrangian is defined as

$$\mathcal{L} = \bar{J} + \int_0^T \mathbf{\Psi}^{f\top} \mathbf{R}^f dt + \int_0^T \mathbf{\Psi}^{s\top} \mathbf{R}^s dt. \tag{4.2}$$

Substituting Eq 4.1 into Eq 4.2, integrating the second term by parts, and requiring stationarity of the Lagrangian with respect to the perturbation in the state variables,

$\delta \mathbf{U}^f$, gives

$$\begin{aligned} &\frac{\partial J_T}{\partial \mathbf{U}^f}\delta \mathbf{U}^f\Big|_{t=T} + {\boldsymbol{\Psi}^f}^\top \mathbf{M}^f \delta \mathbf{U}^f\Big|_{t=T} - {\boldsymbol{\Psi}^f}^\top \mathbf{M}^f \delta \mathbf{U}^f\Big|_{t=0} + \\ &\int_0^T \left[\frac{\partial J}{\partial \mathbf{U}^f} - \frac{\mathrm{d}{\boldsymbol{\Psi}^f}^\top}{\mathrm{d}t}\mathbf{M}^f - {\boldsymbol{\Psi}^f}^\top \frac{\partial \mathbf{r}^f}{\partial \mathbf{U}^f} - {\boldsymbol{\Psi}^s}^\top \frac{\partial \mathbf{r}^s}{\partial \mathbf{U}^f}\right]\delta \mathbf{U}^f dt = \mathbf{0}. \end{aligned} \tag{4.3}$$

The middle term at $t = 0$ drops out since the initial condition on the primal fully specifies the state there, so $\delta \mathbf{U}^f = \mathbf{0}$ at $t = 0$. By transposing the time integrand and noting that the mass matrix is symmetric, the remaining terms yield the adjoint differential equation

$$\mathbf{M}^f \frac{\mathrm{d}\boldsymbol{\Psi}^f}{\mathrm{d}t} + \frac{\partial {\mathbf{r}^f}^\top}{\partial \mathbf{U}^f}\boldsymbol{\Psi}^f + \frac{\partial {\mathbf{r}^s}^\top}{\partial \mathbf{U}^f}\boldsymbol{\Psi}^f = \frac{\partial J}{\partial \mathbf{U}^f}^\top, \tag{4.4}$$

and the terminal condition

$$\boldsymbol{\Psi}^f(T) = \mathbf{0}. \tag{4.5}$$

Note that the mass matrix is non-singular. The governing equation for the structural adjoint can also be obtained in a similar fashion by requiring stationarity of the Lagrangian with respect to the perturbation in the state variables, $\delta \mathbf{U}^s$. As both the adjoint equations of the individual subsystems are coupled, they can be written in a matrix form as

$$\begin{bmatrix} \mathbf{M}^f & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^s \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\Psi}}^f \\ \dot{\boldsymbol{\Psi}}^s \end{bmatrix} + \begin{bmatrix} \frac{\partial {\mathbf{r}^f}^\top}{\partial \mathbf{U}^f} & \frac{\partial {\mathbf{r}^s}^\top}{\partial \mathbf{U}^f} \\ \frac{\partial {\mathbf{r}^f}^\top}{\partial \mathbf{U}^s} & \frac{\partial {\mathbf{r}^s}^\top}{\partial \mathbf{U}^s} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}^f \\ \boldsymbol{\Psi}^s \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial \mathbf{U}^f}^\top \\ \frac{\partial J}{\partial \mathbf{U}^s}^\top \end{bmatrix}. \tag{4.6}$$

The coupled adjoint equations require the evaluation of the linearization of the fluid and structural spatial residuals with respect to both the fluid and structural state, as shown in Eq 4.6. The on-diagonal terms $\partial \mathbf{r}^s / \partial \mathbf{U}^s$ and $\partial \mathbf{r}^f / \partial \mathbf{U}^f$, which represent the change in the structural residual due to perturbation in the structural state and

the change in the fluid residual due to perturbation in the fluid state, are evaluated using the Jacobian of the individual subsystems. The off-diagonal terms $\partial \mathbf{r}^s / \partial \mathbf{U}^f$ and $\partial \mathbf{r}^f / \partial \mathbf{U}^s$, which represent the change in the structural residual due to perturbation in the fluid state and the change in the fluid residual due to perturbation in the structural state, are more challenging to compute in a two-state aerostructural formulation. A number of ways exists in the literature to evaluate these derivatives, such as finite differences, algorithmic differentiation, and complex step. The simplest method for computing derivatives uses an appropriate finite difference formula, such as a forward finite difference, in which each input of interest is perturbed and the output reevaluated to determine its new value. The derivative is then estimated by taking the difference between the output and the unperturbed value and dividing by the value of the perturbation. Although finite differences are not usually very accurate or computationally efficient, they are extremely easy to implement and therefore are widely used. The complex-step derivative approximation computes derivatives of real functions using complex variables. This method originated with the work of Lyness [76] and Lyness and Moler [77]. They developed several methods that made use of complex variables, including a reliable method for calculating the n^{th} derivative of an analytic function. The complex-step method requires access to the source code of the given computational model, and thus, it cannot be applied to black box models without additional effort in most cases. To implement the complex-step method, the source code must be modified so that all real variables and computations are replaced with complex ones. The implementation is straightforward in languages which support operator overloading. Algorithmic differentiation (AD), also known as computational or automatic differentiation, is a well-known method based on the systematic application of the differentiation chain rule to computer programs [88]. For each intermediate variable in the algorithm, a variation due to one input variable is carried through, and all the required sensitivities are computed in one program call.

The derivatives given by the chain rule can be propagated forward (forward mode) or backwards (reverse mode). For AD, the total number of operations is independent of perturbed variables but the memory requirements may be prohibitive, especially for the case of large iterative algorithms. Due to the ease of implementation, finite differences are used to evaluate the off-diagonal derivatives, in this work.

The adjoint equation is solved backward in time, starting from the given terminal condition. The time integration scheme used for both the primal and the adjoint equation for this study is ESDIRK4, but other time schemes can be used as well. For nonlinear problems, the adjoint residual depends on the primal states $\mathbf{U}^f$ and $\mathbf{U}^s$. In multi-stage time integration, the residual must be evaluated at times in between the time nodes, but $\mathbf{U}^f$ and $\mathbf{U}^s$ are not directly available at these time locations; at least not when using general non-variational time integrators, for which only the nodal states are saved. In this work we evaluate the adjoint in between time nodes using a temporal reconstruction with a prescribed order of accuracy [43]. For the fluid subsystem, a linear reconstruction is used for evaluating the fluid states between the time nodes while a quintic reconstruction is used for a structural subsystem. Higher order reconstruction of the fluid states requires the smoothness of the fluid state and it's temporal derivatives. However, the non-smooth start of the fluid simulation at initial time causes the higher-order reconstruction to fail and provide non-physical values at the first time step. Thus, a linear reconstruction is used for the fluid states in this work.

4.1.2 Error Estimation

The unsteady adjoint can be used to evaluate the error in the output of interest through the adjoint-weighted residual [46]. Let $\mathbf{U}_H^f$ and $\mathbf{U}_H^s$ be the approximate fluid and structural solutions obtained from the current space-time time, denoted by

subscript H. The error in the output is defined as:

$$\delta \bar{J} = \bar{J}(\mathbf{U}^f, \mathbf{U}^s) - \bar{J}_H(\mathbf{U}^f_H, \mathbf{U}^s_H) \tag{4.7}$$

As finding the exact solution of the fluid and structural state, $\mathbf{U}^f$ and $\mathbf{U}^s$, is not feasible, a fine-space solution is used to estimate the error in the output of interest.

$$\delta \bar{J} = \bar{J}_h(\mathbf{U}^f_h, \mathbf{U}^s_h) - \bar{J}_H(\mathbf{U}^f_H, \mathbf{U}^s_H) \tag{4.8}$$

where the subscript h, denotes the finer space. The output in the fine-space can be expanded using Taylor expansion around the coarse space solution as follows

$$\bar{J}_h(\mathbf{U}^f_h, \mathbf{U}^s_h) \approx \bar{J}_h(\mathbf{U}^{H,f}_h, \mathbf{U}^{H,s}_h) + \int_0^T \left[\left. \frac{\partial J_h}{\partial \mathbf{U}^f_h} \right|_{\mathbf{U}^{H,f}_h} \delta \mathbf{U}^f + \left. \frac{\partial J_h}{\partial \mathbf{U}^s_h} \right|_{\mathbf{U}^{H,s}_h} \delta \mathbf{U}^s \right] dt + ... \tag{4.9}$$

The unknown perturbations $\delta \mathbf{U}^f$ and $\delta \mathbf{U}^s$ can be evaluated by expanding the fine-space residuals, $\mathbf{R}^f_h(\mathbf{U}^f_h, \mathbf{U}^s_h)$ and $\mathbf{R}^s_h(\mathbf{U}^s_h, \mathbf{U}^f_h)$, around the coarse space solutions using Taylor series expansion. Assuming the fine space residuals to be exact,

$$\mathbf{R}^f_h(\mathbf{U}^f_h, \mathbf{U}^s_h) = \mathbf{R}^f_h(\mathbf{U}^{H,f}_h, \mathbf{U}^{H,s}_h) + \left. \frac{\partial \mathbf{R}^f_h}{\partial \mathbf{U}^f_h} \right|_{\mathbf{U}^{H,f}_h} \delta \mathbf{U}^f + \left. \frac{\partial \mathbf{R}^f_h}{\partial \mathbf{U}^s_h} \right|_{\mathbf{U}^{H,s}_h} \delta \mathbf{U}^s = 0, \tag{4.10}$$

$$\mathbf{R}^s_h(\mathbf{U}^f_h, \mathbf{U}^s_h) = \mathbf{R}^s_h(\mathbf{U}^{H,f}_h, \mathbf{U}^{H,s}_h) + \left. \frac{\partial \mathbf{R}^s_h}{\partial \mathbf{U}^f_h} \right|_{\mathbf{U}^{H,f}_h} \delta \mathbf{U}^f + \left. \frac{\partial \mathbf{R}^s_h}{\partial \mathbf{U}^s_h} \right|_{\mathbf{U}^{H,s}_h} \delta \mathbf{U}^s = 0. \tag{4.11}$$

Reorganizing Eq 4.10 and Eq 4.11 in the matrix form,

$$\begin{bmatrix} \dfrac{\partial \mathbf{R}^f_h}{\partial \mathbf{U}^f_h} & \dfrac{\partial \mathbf{R}^f_h}{\partial \mathbf{U}^s_h} \\ \dfrac{\partial \mathbf{R}^s_h}{\partial \mathbf{U}^f_h} & \dfrac{\partial \mathbf{R}^s_h}{\partial \mathbf{U}^s_h}. \end{bmatrix} \begin{bmatrix} \delta \mathbf{U}^f \\ \\ \delta \mathbf{U}^s \end{bmatrix} = - \begin{bmatrix} \mathbf{R}^f_h(\mathbf{U}^{H,f}_h, \mathbf{U}^{H,s}_h) \\ \\ \mathbf{R}^s_h(\mathbf{U}^{H,f}_h, \mathbf{U}^{H,s}_h) \end{bmatrix} \tag{4.12}$$

Thus, the state perturbations can be re-written as,

$$\begin{bmatrix} \delta \mathbf{U}^f \\ \\ \delta \mathbf{U}^s \end{bmatrix} = - \begin{bmatrix} \dfrac{\partial \mathbf{R}_h^f}{\partial \mathbf{U}_h^f} & \dfrac{\partial \mathbf{R}_h^f}{\partial \mathbf{U}_h^s} \\ \dfrac{\partial \mathbf{R}_h^s}{\partial \mathbf{U}_h^f} & \dfrac{\partial \mathbf{R}_h^s}{\partial \mathbf{U}_h^s} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R}_h^f(\mathbf{U}_h^{H,f}, \mathbf{U}_h^{H,s}) \\ \\ \mathbf{R}_h^s(\mathbf{U}_h^{H,f}, \mathbf{U}_h^{H,s}) \end{bmatrix} \tag{4.13}$$

Inserting the state perturbation into the approximation of the error estimate (Eq 4.9) gives

$$\begin{aligned} \bar{J}_h(\mathbf{U}_h^f, \mathbf{U}_h^s) &\approx \bar{J}_h(\mathbf{U}_h^{H,f}, \mathbf{U}_h^{H,s}) \\ - \int_0^T \begin{bmatrix} \dfrac{\partial J_h}{\partial \mathbf{U}_h^f} & \dfrac{\partial J_h}{\partial \mathbf{U}_h^s} \end{bmatrix} & \begin{bmatrix} \dfrac{\partial \mathbf{R}_h^f}{\partial \mathbf{U}_h^f} & \dfrac{\partial \mathbf{R}_h^f}{\partial \mathbf{U}_h^s} \\ \dfrac{\partial \mathbf{R}_h^s}{\partial \mathbf{U}_h^f} & \dfrac{\partial \mathbf{R}_h^s}{\partial \mathbf{U}_h^s} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R}_h^f(\mathbf{U}_h^{H,f}, \mathbf{U}_h^{H,s}) \\ \\ \mathbf{R}_h^s(\mathbf{U}_h^{H,f}, \mathbf{U}_h^{H,s}) \end{bmatrix} dt \end{aligned} \tag{4.14}$$

Without expanding the strong form unsteady residuals, the unsteady coupled adjoint equations (Eq 4.6) can be rewritten as

$$\begin{bmatrix} \boldsymbol{\Psi}^{T,f} & \boldsymbol{\Psi}^{T,s} \end{bmatrix} \begin{bmatrix} \dfrac{\partial \mathbf{R}^f}{\partial \mathbf{U}^f} & \dfrac{\partial \mathbf{R}^s}{\partial \mathbf{U}^f} \\ \dfrac{\partial \mathbf{R}^f}{\partial \mathbf{U}^s} & \dfrac{\partial \mathbf{R}^s}{\partial \mathbf{U}^s} \end{bmatrix} = - \begin{bmatrix} \dfrac{\partial J}{\partial \mathbf{U}^f} & \dfrac{\partial J}{\partial \mathbf{U}^s} \end{bmatrix}. \tag{4.15}$$

Substituting the definition of the adjoint in the finer space from Eq 4.15 into Eq 4.14 gives,

$$\bar{J}_h(\mathbf{U}_h^f, \mathbf{U}_h^s) \approx \bar{J}_h(\mathbf{U}_h^{H,f}, \mathbf{U}_h^{H,s}) + \int_0^T \begin{bmatrix} \boldsymbol{\Psi}_h^{T,f} & \boldsymbol{\Psi}_h^{T,s} \end{bmatrix} \begin{bmatrix} \mathbf{R}_h^f(\mathbf{U}_h^{H,f}, \mathbf{U}_h^{H,s}) \\ \\ \mathbf{R}_h^s(\mathbf{U}_h^{H,f}, \mathbf{U}_h^{H,s}) \end{bmatrix} dt. \tag{4.16}$$

The output evaluated by projecting the coarse space solution into the fine space will be the same as the output evaluated in the coarse space.

$$\bar{J}_H(\mathbf{U}_H^f, \mathbf{U}_H^s) = \bar{J}_h(\mathbf{U}_h^{H,f}, \mathbf{U}_h^{H,s}) \tag{4.17}$$

Substituting Eq 4.17 into Eq 4.16, the error estimate for the output of interest at a given time step is given by,

$$\delta\bar{J}_{est} \approx \int_0^T \left[\mathbf{\Psi}_h^{T,f}\mathbf{R}_h^f(\mathbf{U}_h^{H,f}, \mathbf{U}_h^{H,s}) + \mathbf{\Psi}_h^{T,s}\mathbf{R}_h^s(\mathbf{U}_h^{H,f}, \mathbf{U}_h^{H,s})\right] dt. \tag{4.18}$$

The fine space discrete unsteady adjoint, which is unavailable, is approximated in a finer space by increasing the degrees of freedom in the spatial discretization.

4.2 Mesh Adaptation

Unsteady error estimates in the space-time fluid mesh guide the adaptation process. Space-time elements selected for refinement or coarsening are chosen based on two factors: 1) the estimated error in the space-time element, and 2) the computational cost of refinement. These two aspects are combined into an adaptive indicator called the "figure of merit". The figure of merit is defined as the element error eliminated by refinement divided by the degrees of freedom introduced by the refinement. The cost is defined by the total degrees of freedom,

$$C \equiv C^{\text{space}}C^{\text{time}} \quad , \quad C^{\text{space}} \equiv \sum_{e=1}^{N_e} n(p_e) \quad , \quad C^{\text{time}} \equiv N_t n_r, \tag{4.19}$$

where $n(p_e)$ is the number of spatial degrees for an element of order p_e, N_t is the number of time steps, and n_r is the number of temporal degrees of freedom, i.e., system solves, per time step. For example, the five-stage ESDIRK4 method has $n_r = 5$.

A user-defined growth factor, f_{tot}, is used to decide the change in the number of total degrees of freedom after each adaptation cycle. In this work, adaptive mesh refinement occurs only in the spatial discretization of the FSI system. Thus, the unsteady error estimates for the fluid and structural subsystem are aggregated over all time steps for every element in the fluid and structural meshes. Once the figure of merit is evaluated for all elements, using Eq 4.19, a decreasing refinement threshold strategy is used for choosing the elements for refinement. In the decreasing refinement threshold strategy [92], elements with the highest error are targeted for refinement first so that the mesh size grows more slowly and multiple expensive solves on the finest meshes are avoided. In the fluid mesh, the adaptive strategy refines in space by hanging node refinement (h-adaptation) or spatial order refinement (p-adaptation) while in the structural mesh only h-adaptation is used.

CHAPTER V

Mesh Deformation Errors

As discussed in Section 2.1.5, mesh-motion algorithms required for the ALE formulation of the Navier-Stokes equation, introduce errors in the fluid simulation. This chapter reviews the impact of mesh-motion algorithms on high-fidelity fluid simulations with deforming domains. The existence of an optimum spatial location of the blending region in a mesh-motion algorithm is investigated for two separate cases. The goal of this chapter is to identify trade-offs and general trends in the relationship between mesh-motion parameters and output errors. The use of output-based mesh adaptation in efficiently reducing the spatial errors generated by the mesh distortion as well the spatial discretization has been investigated. Secondly, the necessity of the implementation of a geometric conservation law (GCL) for achieving high accuracy in high-order FSI simulations involving rigid-body motions is also investigated.

5.1 Optimizing Mesh-Motion Algorithms

The Arbitrary Lagrangian Eulerian (ALE) framework helps facilitate the evolution of fluid boundaries that are imposed by the motion of the free surfaces and/or the adjoining deforming structures by providing a reference frame wherein the computational mesh can move and deform independent of the fluid particle motion. However, the distortion of mesh elements in the fluid mesh and the introduction of arbitrary

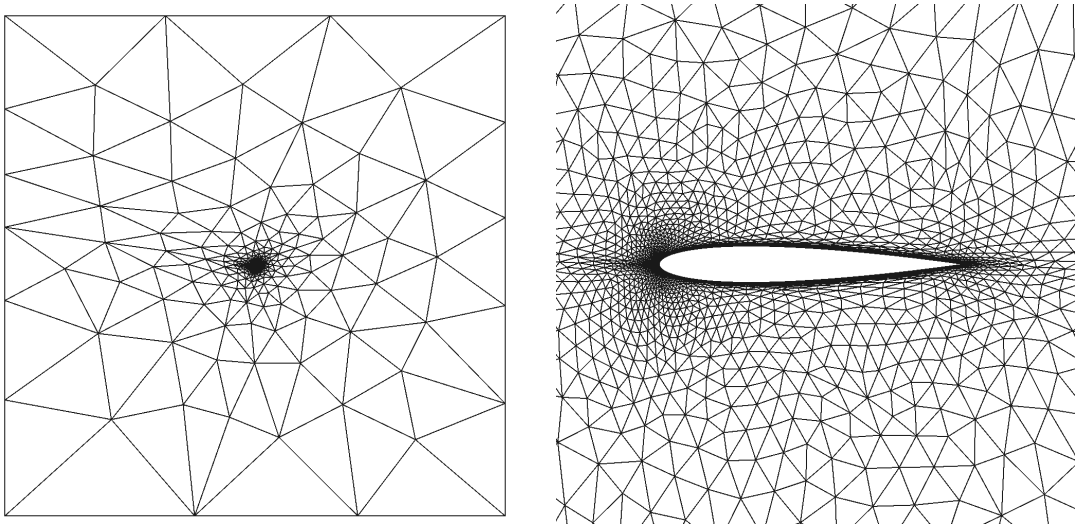
mesh velocities introduce errors in the simulation. In this section, the impact of mesh-motion algorithms on a high-fidelity fluid simulation with deforming domains and a comprehensive analysis of controlling such mesh deformation errors is studied on an unstructured mesh. Two separate cases have been designed for quantifying and analyzing the errors arising from mesh deformation. Firstly, a free-stream preservation test, which is widely used for highlighting the impact of non-linear mapping in a free-stream is modified to quantify the errors generated by the mesh deformation. In this case, the impact of the mesh deformation is characterized by entropy generation. Secondly, a more practical case of an airfoil undergoing rigid-body deformation in a steady fluid flow is analyzed for engineering outputs of interest such as lift.

5.1.1 Free-Stream Preservation

Spatial and temporal errors generated in a fluid simulation with deforming domains arise from the corresponding space-time discretization and mesh deformation. As both sources of error propagate spatially and temporally in a simulation, it is difficult to separate the errors obtained from the output-based error estimate based on the source. However, for an arbitrary mesh-motion applied to a free-stream, the states are only contaminated by the errors arising from mesh deformation, as the space-time discretization without deformation conserves the free-stream. Therefore, such a test acts as an ideal case to study the spatial and temporal distribution of mesh deformation errors. Steady and unsteady mesh-motion error quantification studies are presented for a free-stream undergoing mesh deformation. Error estimates of the entropy z are used to demonstrate the existence of optimum blending region for the mesh-motion algorithm.

Consider a uniform fluid flow around an airfoil placed centrally in square domain which spans $[-100c, 100c]$ in both dimensions, where c is the chord length of the

airfoil. To simulate a uniform fluid flow, a coarse, unstructured, triangular mesh of 5489 elements is generated, as shown in Figure 5.1. The unstructured triangular fluid meshes used in this thesis are generated using BAMG [61], an anisotropic 2D mesh generator. For fixed, user defined, degrees of freedom, BAMG is used to generate an h-adapted mesh, optimized using metric-based mesh adaptation. Free-stream boundary conditions are applied at the farfield boundaries as well as on the airfoil. The airfoil boundary acts as the set of nodes where the deformation is prescribed but as the boundary condition on the airfoil is free-stream, it does not violate the preservation phenomenon. Despite this being a free-stream preservation test, the reason for choosing a viscous mesh is because most of the instabilities that occur in FSI simulations, such as flutter, occur at high Reynolds number flows. Thus, a mesh capable of simulating such as system is used to study the impact of mesh-motion. Two degrees of freedom of the airfoil motion, the pitch, $\alpha(t)$, and plunge, $h(t)$, are prescribed using sinusoidal functions. An explicit mesh deformation algorithm, as mentioned in Section 2.1.5.1, is applied to handle the deformations occurring in the fluid domain due to the moving airfoil. The mesh deformation algorithm depends on three variables:



(a) Entire domain (b) Zoom

Figure 5.1: Unstructured viscous mesh for the free-stream preservation test.

the inner radius, the outer radius, and the polynomial blending function, which are generally user-defined. The goal of this study is to investigate the existence of an optimal blending region and blending function for a viscous mesh undergoing arbitrary motion, with the goal of minimizing the error due to mesh deformation.

The study is conducted for both steady and unsteady cases. For the steady-state deformation, an inviscid simulation is conducted with $p = 3$ order polynomials with a constant pitch deformation of 5 degrees, centered at the quarter chord of the airfoil. The error for the steady case is evaluated for an output defined as the normalized domain integral of the entropy, given by

$$\bar{J} = \frac{1}{A} \int_{\Omega} S d\Omega. \tag{5.1}$$

where S is the entropy and A is the area of the domain. Figure 5.2 compares the effect of order of blending polynomials on the entropy error estimate for inner and outer

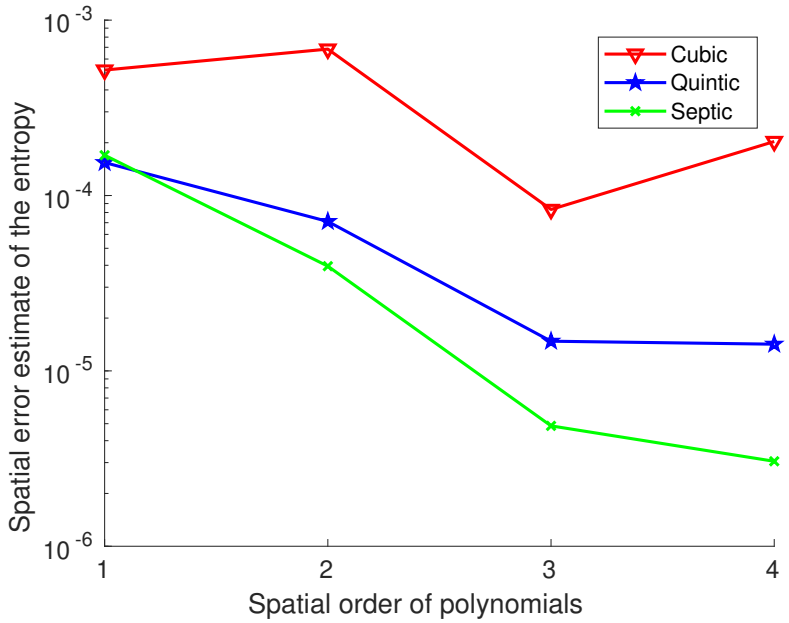
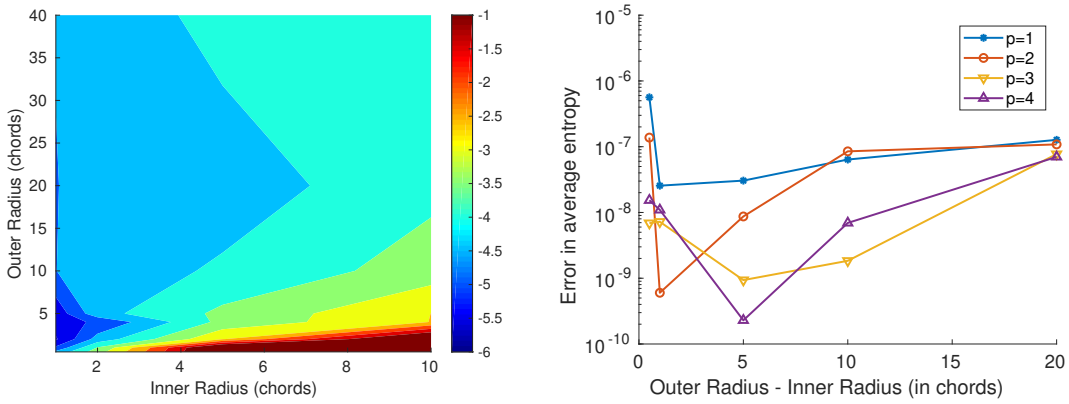


Figure 5.2: Effect of the order of the blending polynomial on the error estimate.



(a) Error estimates of the average entropy, in log scale, for varying inner and outer radii with septic blending polynomials.

(b) Error in the average entropy for varying outer radius with a fixed inner radius of $1c$.

Figure 5.3: Entropy error and error estimates.

radii of $1c$ and $5c$. It shows that across different orders of dicretizations, increasing the order of blending polynomials leads to better representation of the deformation in the elements, which in turn leads to lower errors, provided a sufficiently high quadrature rule is used for integration when mesh-motion is active. Figure 5.3(a) shows a contour representation of error estimates for varying inner and outer radii of the blending region with septic blending. The optimum inner and outer radii for the lowest error estimate come out to be $1c$ and $5c$, respectively. We can conclude that lowering the inner radius to be as low as possible while avoiding element inversion reduces the error estimate of the output due to mesh-motion, because this results in a blending region located in the domain which is finer. Lowering the outer radius causes the blending to occur in a very small domain resulting in large gradients in the determinant of the mapping, g, within the elements, thereby causing higher errors. Similarly, increasing the outer radius of the blending region also results in an increase in the error because the number of elements affected by the blending region grows, as does the size of these elements. For this case, the optimum outer radius was identified at $5c$, which lies between the two extremes and leads to the lowest

errors arising from mesh-motion. The optimum parameters are specific to the case tested. However, similar results are expected for the blending region location for viscous meshes i.e, close to the deforming domain for the lowest mesh-motion errors. A similar optimum blending region is observed for different amplitudes of the deformation and other degrees of freedom. Apart from error estimates, the absolute error in the average entropy in the entire domain is also studied for varying outer radii, as shown in Figure 5.3(b), which depicts a similar behaviour as the error estimate.

To study the propagation of spatial and temporal error arising due to mesh-motion, an unsteady pitch motion was prescribed to the fluid system given by $\alpha = \alpha_0 \sin(t)$ where $\alpha_0 = 5°$. The inviscid simulation was conducted with $p = 3$ and $p = 4$ order polynomials and an ESDIRK4 time scheme with 50 time steps for a final time of 5 time units, where one time unit is defined as the time taken for flow at free-stream speed to traverse the chord of the airfoil. The output chosen for the unsteady adjoint evaluation and error estimate is the domain integral of the entropy at the final

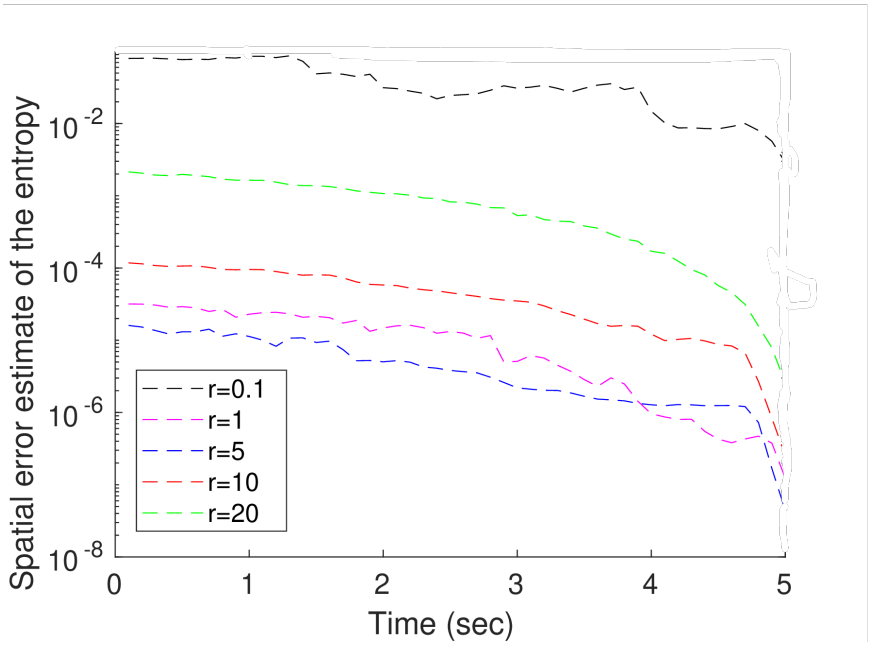
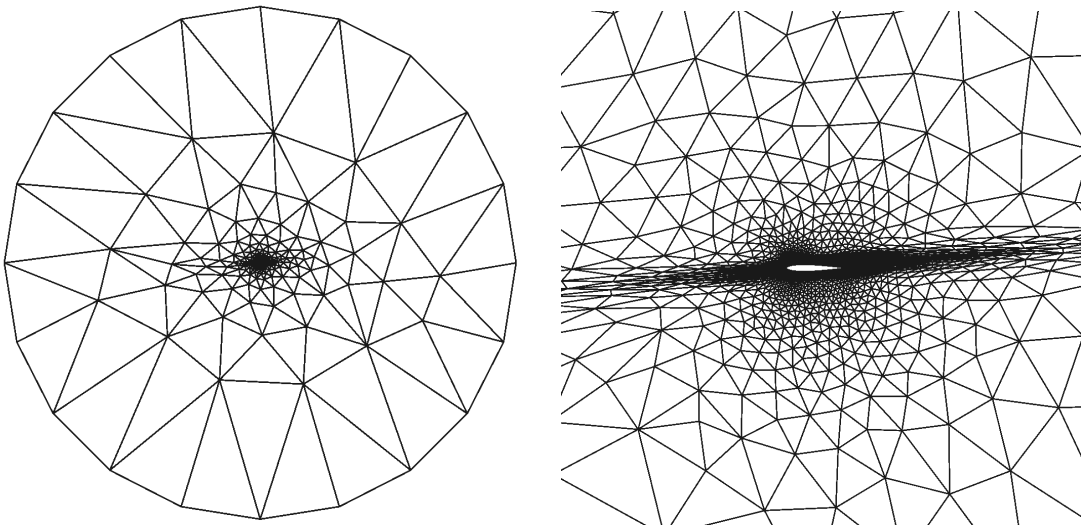


Figure 5.4: Unsteady spatial error estimate of the entropy for varying outer radii with an inner radii of $1c$ and septic blending polynomials.

time. Figure 5.4 shows the evolution of spatial error estimates over time for $p = 4$ for varying outer radii with a constant inner radius of $1c$ and septic blending. The results from steady state are corroborated with the unsteady solution where a similar optimum outer radius is observed to reduce the error estimate.

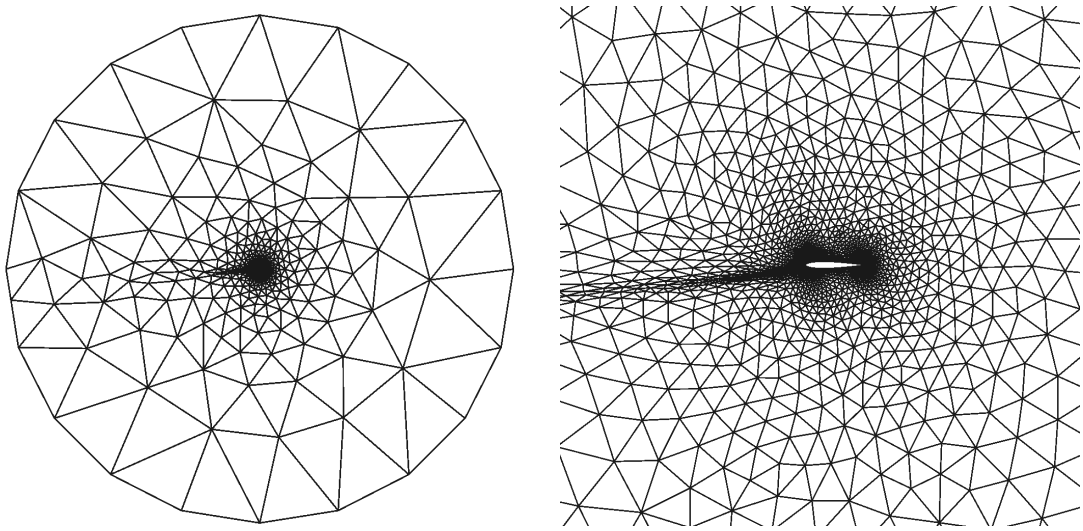
5.1.2 NACA 0012 Airfoil with Prescribed Pitch Deformation

In this test case, mesh deformation is applied to an airfoil with prescribed rigid-body motion and the output of interest chosen for the study is the lift generated by the airfoil. The investigation is conducted for both viscous and inviscid flows. The main motivation behind this problem is to investigate the effect of the position of blending regions on the convergence of the output of interest. Secondly, a comparative study based on the effect of the initial mesh used for a simulation with mesh deformation is also performed for three separate initial meshes. Finally, the effect of different definitions of error estimates used for mesh adaption on the output convergence is also investigated for simulations with mesh deformation.



(a) Entire mesh (b) Zoom

Figure 5.5: Reference mesh for lift evaluation for laminar flow.



(a) Entire mesh (b) Zoom

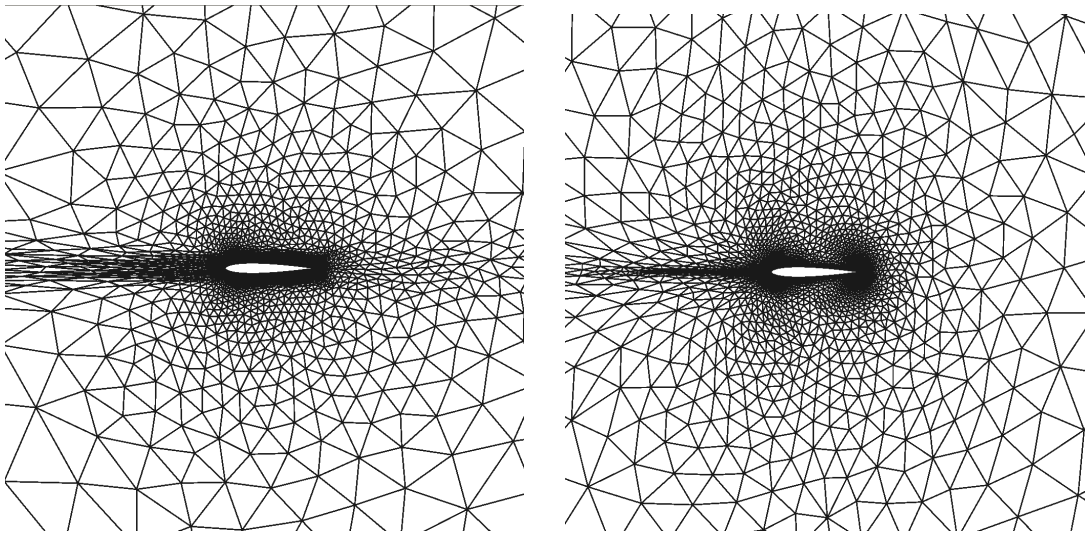
Figure 5.6: Reference mesh for lift evaluation for inviscid flow.

Consider a NACA 0012 airfoil placed centrally in a circular mesh of radius 1000 chords. A pitch deformation of five degrees about the leading edge is provided to the airfoil using the mesh-motion algorithm described in Section 2.1.5.1. The effect of viscosity on the mesh deformation error is studied by considering two flow conditions, viscous and inviscid flow. For the viscous simulations, the Reynolds number is chosen to be $Re = 1000$. The focus of this analysis is to quantify only the spatial errors generated by distortion of the mesh elements by the mesh-motion algorithm. Therefore, all the simulations are steady in nature. The meshes used for this simulation use curved elements of order three, $q = 3$, to represent the airfoil geometry. Free-stream boundary conditions are applied at the farfield boundaries and wall boundary conditions are applied at the airfoil boundary. The Mach number used for this analysis is $M = 0.345$. This subsonic Mach number is chosen based on the flutter estimation study conducted in the next section. Multiple inner and outer radii combinations are used to vary the position of the blending region for the deformation to study the effect of the position of the blending region on the output convergence. For a particular blending region, the spatial errors in the simulation are quantified by comparing

the output of interest against a reference case, which is unaffected by mesh-motion. In the reference case, the desired angle of attack is achieved by changing the flow boundary conditions at the farfield without applying any mesh deformation to the airfoil. Figure 5.5 and Figure 5.6 present the meshes used for the laminar and inviscid reference cases, respectively. The reference lift is evaluated for a spatial discretization of order five, $p = 5$ using a mesh with 5545 elements.

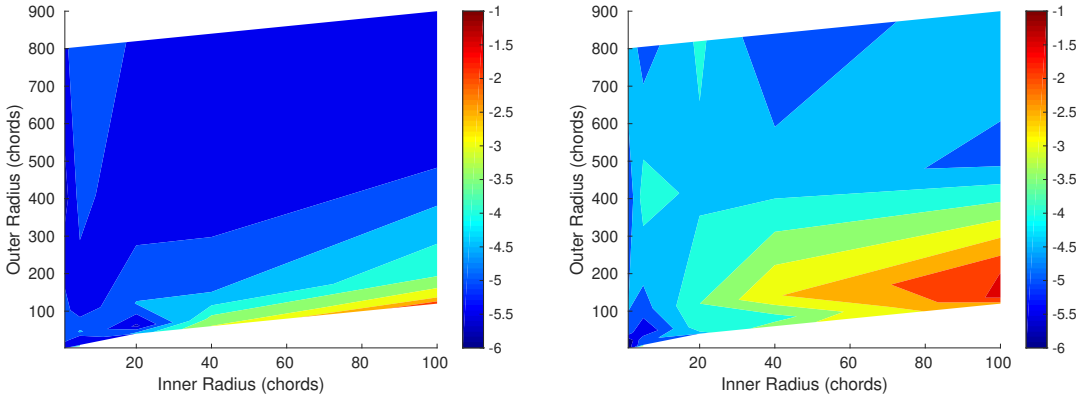
5.1.2.1 Mesh optimized for the reference position of the airfoil in ALE

Simulations involving mesh deformation generally use a mesh optimized for the reference position in ALE as the initial mesh. In simulations involving FSI, the mesh deformation is often not known a priori to the user. Therefore, this strategy is useful as it is optimized to reduce spatial discretization errors and can yield a good initial mesh. Using this strategy, a mesh optimized for the reference position of the airfoil in the ALE framework, where the airfoil is aligned with the flow is obtained, as shown in Figure 5.7. Employing such a mesh, h-optimized without mesh-motion, can lead to different output error, depending on the mesh-motion algorithm itself. Starting



(a) Initial mesh for laminar flow (b) Initial mesh for inviscid flow

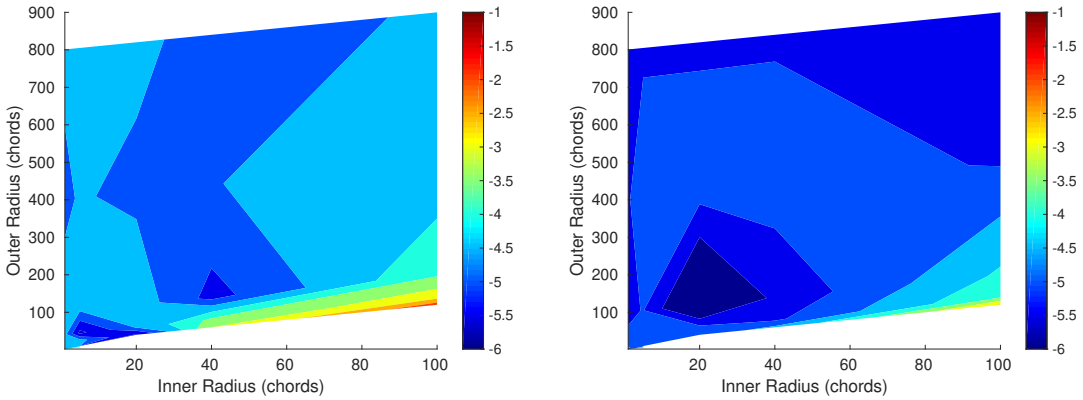
Figure 5.7: Initial mesh used in the first strategy. The mesh is generated by optimizing for the reference/undeformed mesh in ALE.



(a) Laminar (b) Inviscid

Figure 5.8: Error in lift post output-based adaptation.

with this initial mesh, the mesh-motion algorithm deforms the mesh for the various combinations of inner and outer radii. For the error analysis, discrete values of inner radius, $R_{\text{inner}} \in [1c, 5c, 20c, 40c, 100c]$ and blending distance, which is the distance between the inner and outer radius, $D_{\text{blending}} \in [1c, 5c, 20c, 40c, 100c, 400c, 800c]$, are used. As a single mesh is used for the error estimation for the various blending regions, the initial spatial error arising only from the discretization is the same for all of the blending regions. To study the effect of mesh deformation, an output convergence study is conducted using output based mesh adaptation. Starting with uniform



(a) Laminar (b) Inviscid

Figure 5.9: Error in lift post residual-based adaptation.

$p = 1$ elements in the entire domain, the initial fluid mesh is adapted in spatial order by subjecting it to six cycles of p-adaptation. The growth factor is chosen to be two for each adaptation cycle and a constraint on the total degrees of freedom post-adaptation is imposed at 5000 triangular elements. Error estimates obtained using the fluid adjoint are used to guide the mesh adaptation. Figure 5.10 and Figure 5.11 show the output convergence for the different blending regions as a function of the adaptive iteration for the laminar and inviscid flows, respectively. The various plots track the absolute error in lift for a constant inner radius and varying outer radius. As the convergence of lift is not a monotonic function of the adaptive iteration, the sudden drop in the error of the output is not an indication of convergence but a sign of the output crossing the reference lift en route to convergence. The existence of two optimum blending regions can be observed from Figure 5.8(a), which shows the absolute error in the lift at the end of the adaptive iterations for laminar flow.

The first optimum blending region is located close to the airfoil. This location of the blending region benefits from the mesh density of the initial mesh used for the simulations. The initial mesh optimized for the reference position leads to a finer mesh close to the airfoil, which effectively resolves the boundary layer. Similarly, lack of flow features far away from the airfoil results in larger element sizes there. Due to a finer mesh close to the airfoil, the blending region is well-resolved and leads to less distortion within each element, thereby resulting in better output convergence. The second optimum blending region exists where the outer radius extends far away from the airfoil. Irrespective of the inner radius, having a larger outer radius results in less distortion within an element, which in turn results in less mesh deformation error. Relatively slow convergence can be seen in the output for cases having small blending regions with larger inner radius. Pushing the inner radius away from the airfoil and keeping the blending region small results in the blending occurring primarily in

elements of larger sizes. These large elements are incapable of resolving the blending well, thus leading to high errors. Similar optimum blending regions are observed for the inviscid case as well, as shown in Figure 5.8(b). For the inviscid case, the optimum blending region close to the airfoil outperforms the optimum blending region away from the airfoil due to the initial mesh used for the inviscid flow. The initial optimized mesh for the inviscid case is more isotropic compared to the viscous flow case because of the lack of a boundary layer. The isotropic nature of the mesh combined with the radial nature of the blending leads to a better resolution for the deformations blended close to the airfoil. The output convergence study in this analysis is conducted using output-based mesh adaptation. This technique is successful in targeting the elements in the blending region for further adaptation because of the definition of the adaptive indicator. As described in Section 4.1, the adaptive indicator is a function of the adjoint and the residual evaluated by projecting the coarse space solution into the fine space. The elements inside the blending region, despite having a lower adjoint magnitude, suffer from the errors originating from mesh-motion and have a higher projected residual.

A second definition of the adaptive indicator, based solely on the residual is also tested for output convergence. Figure 5.9 shows the output convergence for the various blending regions as a function of the adaptive iteration, using residual-based adaptation. The output convergence for the various positions for the blending region in the laminar case is slower for residual-based adaptation when comparing it against output-based adaptation. The inclusion of the lift adjoint in the adaptive indicator focuses the adaptation on mesh elements important for lift evaluation irrespective of the mesh deformation errors. Residual-based adaptation, on the other hand, focuses more on the errors due to mesh deformation in the bigger elements leading to slower convergence. However, the opposite behavior is seen in the case of inviscid flow, where

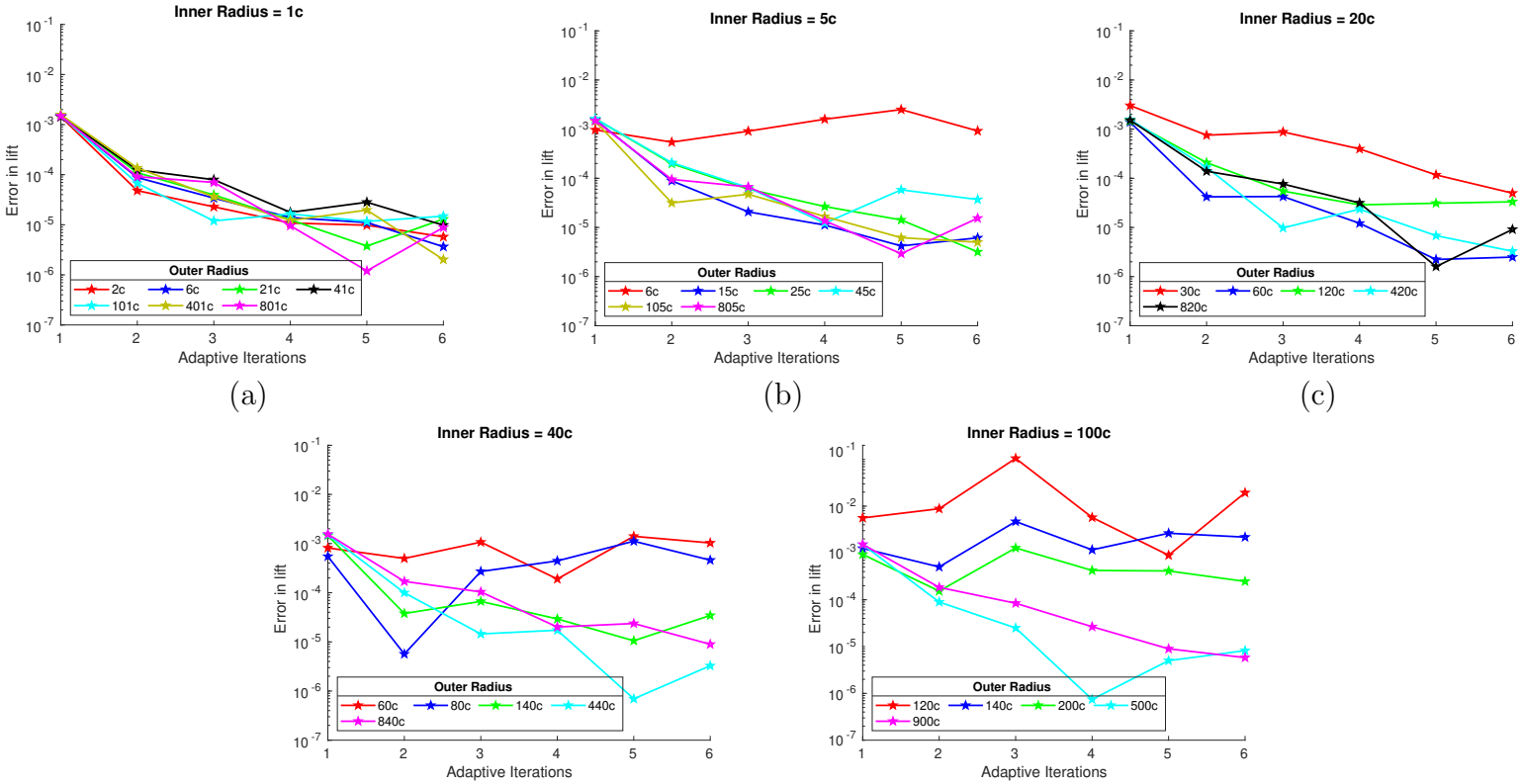


Figure 5.10: Error in lift generated by a NACA0012 airfoil in a laminar flow as a function of the adaptive iterations for the first mesh adaptation strategy, where the mesh is optimized for the reference position of the airfoil in ALE.

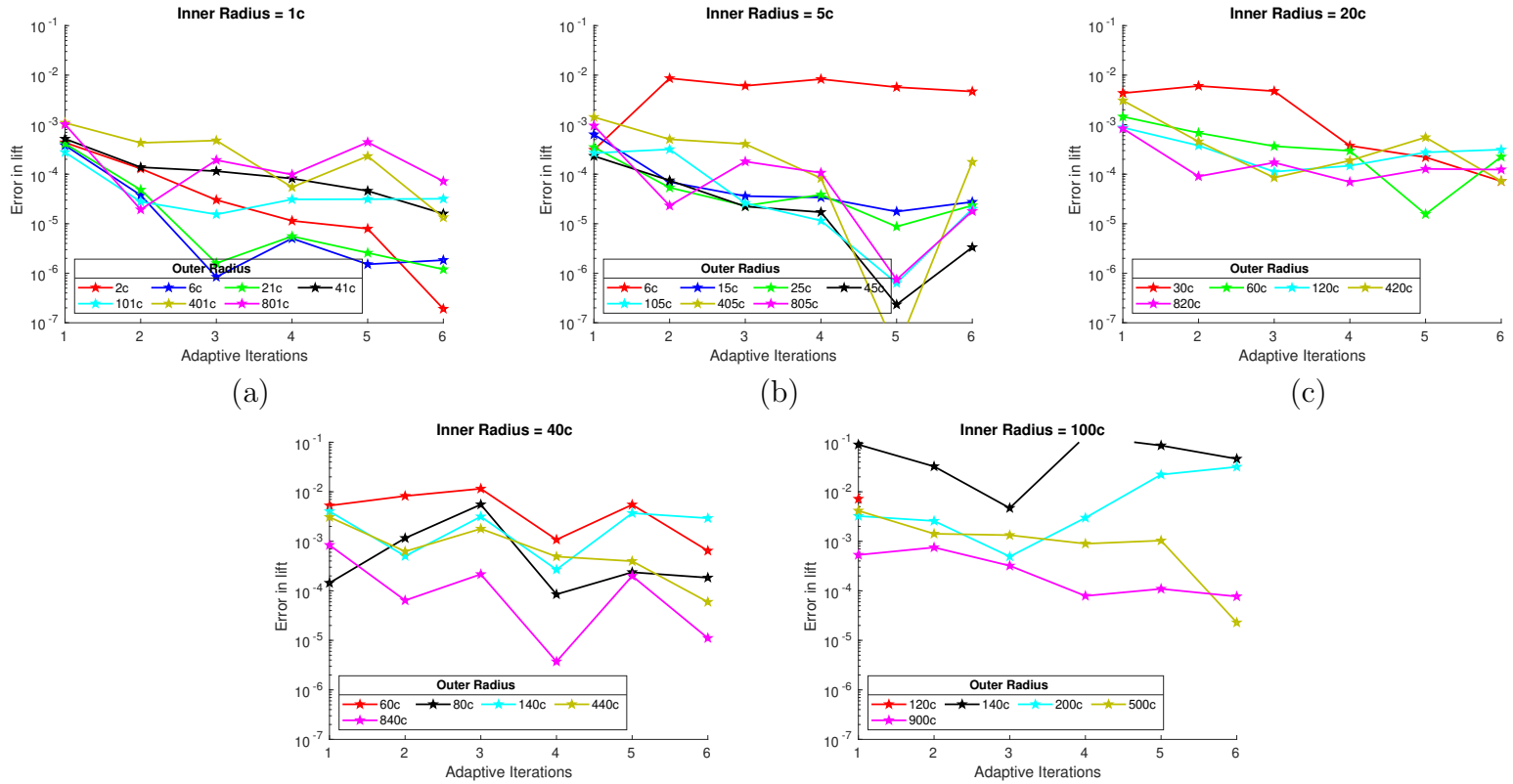
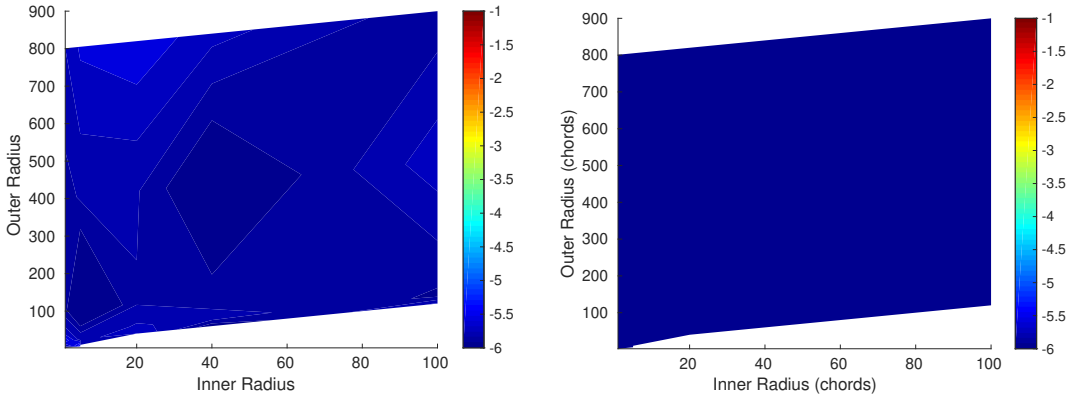


Figure 5.11: Error in lift generated by a NACA0012 airfoil in a inviscid flow as a function of the adaptive iterations for the first mesh adaptation strategy, where the mesh is optimized for the reference position of the airfoil in ALE.

better convergence rates are seen with residual-based adaptation. In output-based adaptation, the singularity of the adjoint along the stagnation streamline [126] leads to numerical noise in the adjoint evaluation. This causes excessive adaptation along the stagnation streamline, which is avoided in the case of residual-based adaptation, leading to more efficient output convergence. Mesh adaptation using residual-based adaptive indicators is comparatively much faster than output-based adaptive indicators due to the lack of the adjoint evaluation. It is also able to highlight some of the shortcomings of the output-based approach, where errors in adjoint evaluations can lead to slower output convergence for inviscid cases. However, such a definition of the error estimate is not useful for unsteady cases, where the information of characteristics, provided by the adjoint, is extremely useful.

5.1.2.2 Mesh optimized for the deformed position of the airfoil in ALE

A second strategy for the initial mesh generation can be used when the mesh deformation is known a priori to the user, such as simulations involving prescribed motion of the airfoil. In this strategy, an optimized initial mesh is generated for each specific blending region by taking the known pitch deformation of the airfoil into



(a) Laminar (b) Inviscid

Figure 5.12: Error in lift post output-based adaptation using the initial meshes proposed in the second strategy.

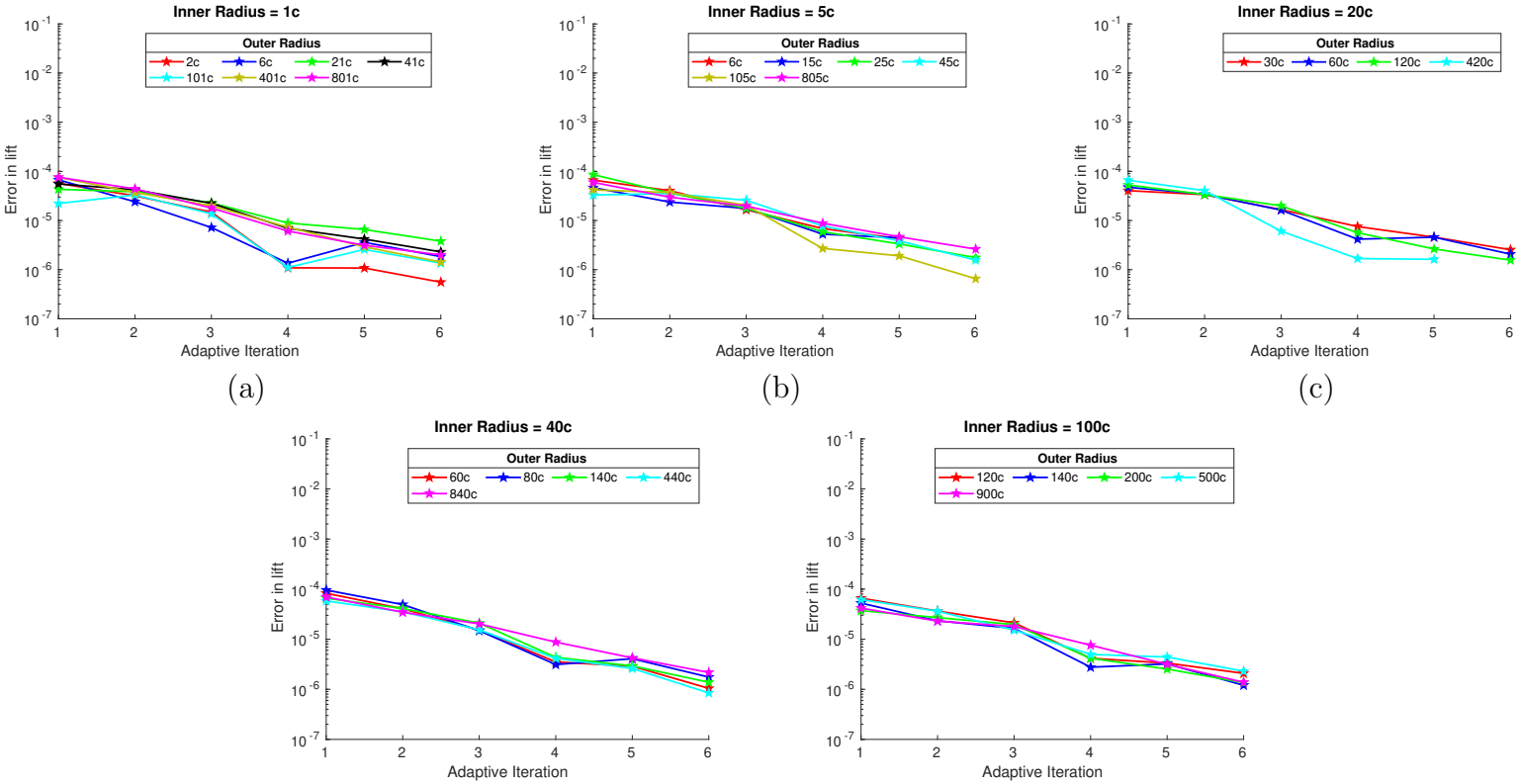


Figure 5.13: Error in lift generated by a NACA0012 airfoil in a laminar flow as a function of the adaptive iterations for the second mesh adaptation strategy, where the mesh is optimized for the deformed position of the airfoil.

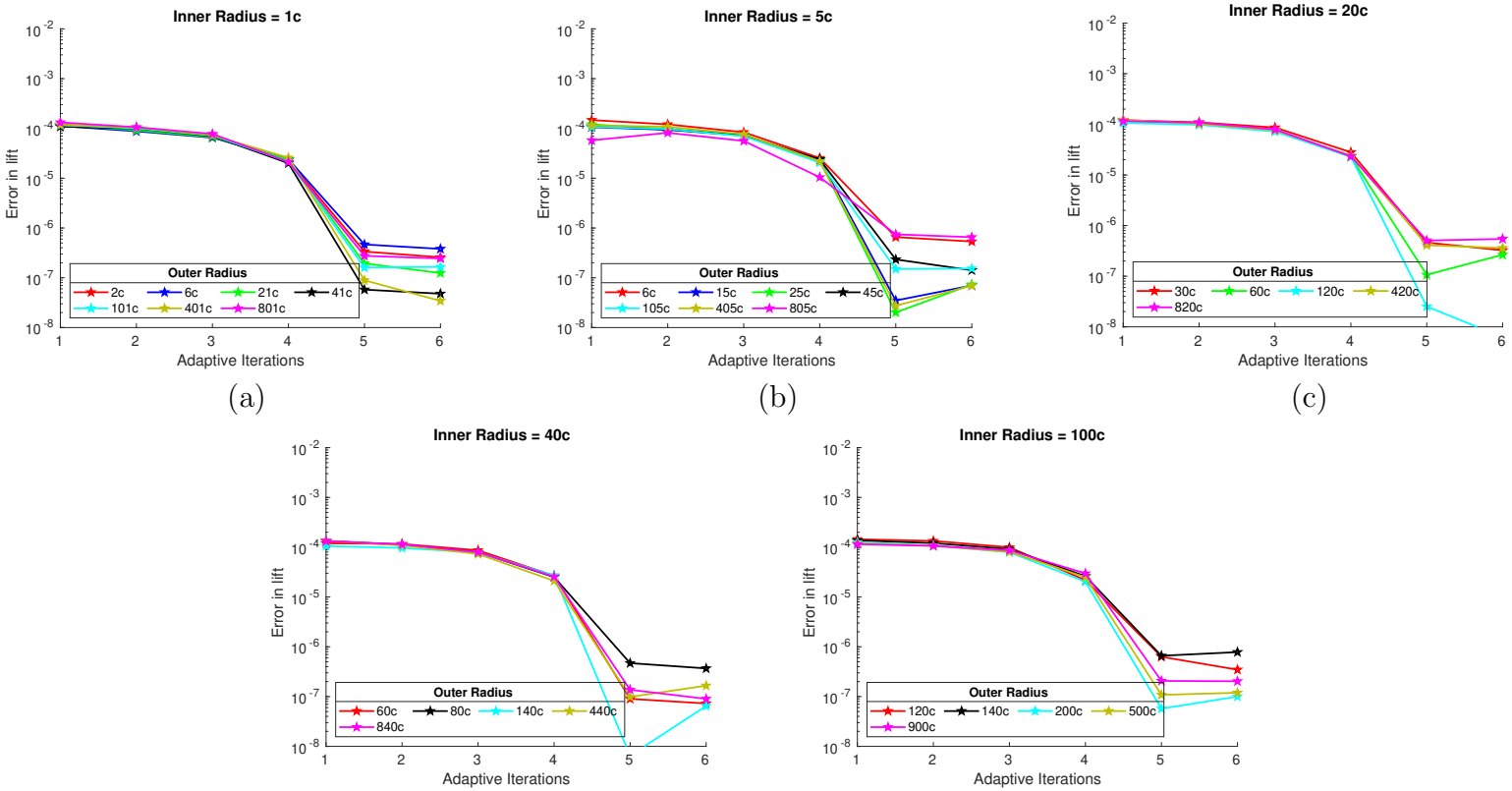
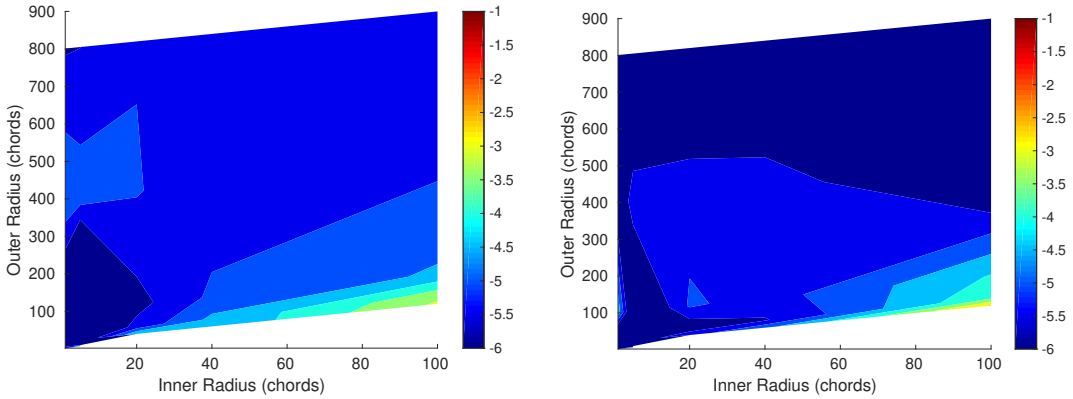


Figure 5.14: Error in lift generated by a NACA0012 airfoil in a inviscid flow as a function of the adaptive iterations for the second mesh adaptation strategy, where the mesh is optimized for the deformed position of the airfoil.



(a) Laminar (b) Inviscid

Figure 5.15: Error in lift post output-based adaptation using the initial meshes proposed in the third strategy.

optimized for the blending region leads to better convergence.

5.1.2.3 Mesh optimized for the reference lift evaluation of the airfoil

Meshes used for the reference lift evaluation, as shown in Figure 5.6, where the boundary condition at the farfield is changed to achieve the desired angle of attack, can also serve as good initial meshes for this analysis. Despite, the degrees of freedom used to resolve the wake and regions near the stagnation streamline being aligned differently than the flow direction used in the error analysis, the application of mesh deformation to these meshes re-aligns these regions. Thus, this strategy uses an initial mesh optimized for reducing the spatial discretization errors for the known deformation without knowledge of the blending region. A similar output convergence study using mesh adaptation as described in the previous subsection is conducted for the two flow regimes. Figure 5.16 and Figure 5.17 show the output convergence for the different blending regions as a function of the adaptive iteration for the laminar and inviscid flow, respectively. The existence of an optimum blending region can be observed from Figure 5.15, which shows the error in the lift at the end of the adaptive iterations, for laminar flow. The optimum blending region is located for small inner

radii with an outer radius of 100 chords. For the laminar case, the initial mesh has higher mesh density along the stagnation streamline and the boundary layer. However, the entire stagnation streamline, which extends up to the upstream farfield at 1000 chords, is not resolved because of the constraint on the total degrees of freedom. Thus, for the given constraint of 5000 elements, the mesh generator only resolves the stagnation streamline up to 100 chords, which makes the blending region with an outer radius of 100c converge more aggressively compared to other locations. Depending on the total degrees of freedom in the initial mesh, the optimum blending region may move further upstream of the airfoil if the stagnation streamline is further resolved. Therefore, the optimum blending region observed for this case is unique to this particular initial mesh, but this study highlights the importance of the initial mesh structure on output convergence for FSI simulations. The inviscid reference mesh, on the other hand, has quite a different distribution of mesh elements compared to the laminar flow mesh. The lack of a boundary layer leads to more isotropic distribution of mesh elements and leads to smaller elements compared to the laminar case in the farfield. Thus, two optimum blending regions are observed for the inviscid case, as seen in Figure 5.15b. The location of the optimum regions is similar to the first strategy, the explanation of which can be extended to this case as well.

Mesh-motion errors arise due to under-integration and loss of optimal polynomial approximation in global space due space-time deformation of the elements. Within the current scope of this chapter, where mesh deformation is applied on a single mesh without re-meshing, this effect is mitigated by changing the blending parameters, which determine the region in the mesh where the mesh-motion errors are generated. The blending parameter affects elements of varying sizes differently, which explains the sensitivity of the outputs to the position of the blending region. The blending parameter, unique to this particular deformation algorithm, is equivalent to the term

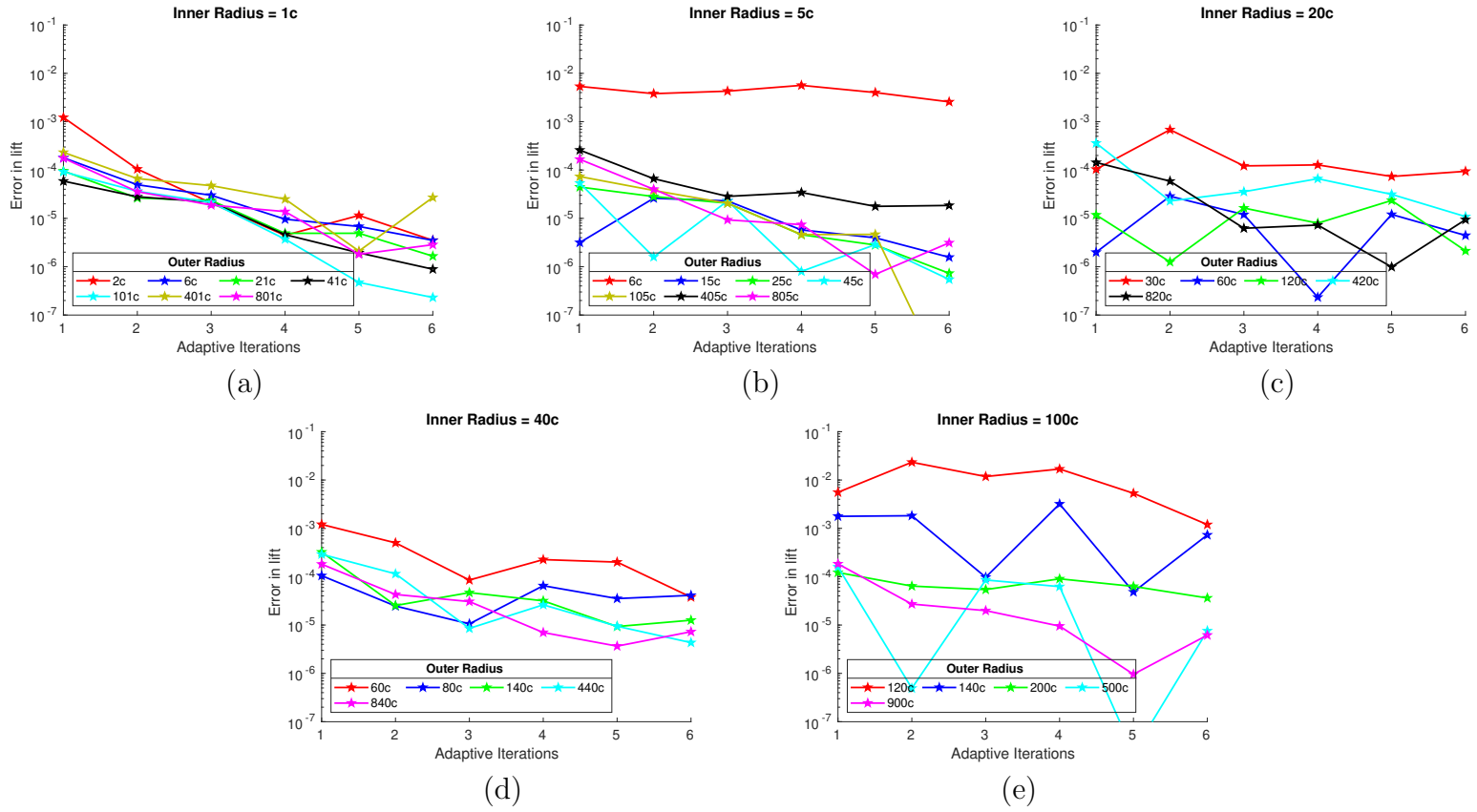


Figure 5.16: Error in lift generated by a NACA0012 airfoil in a laminar flow as a function of the adaptive iterations for the third mesh adaptation strategy, where the mesh is optimized for the reference lift evaluation of the airfoil.

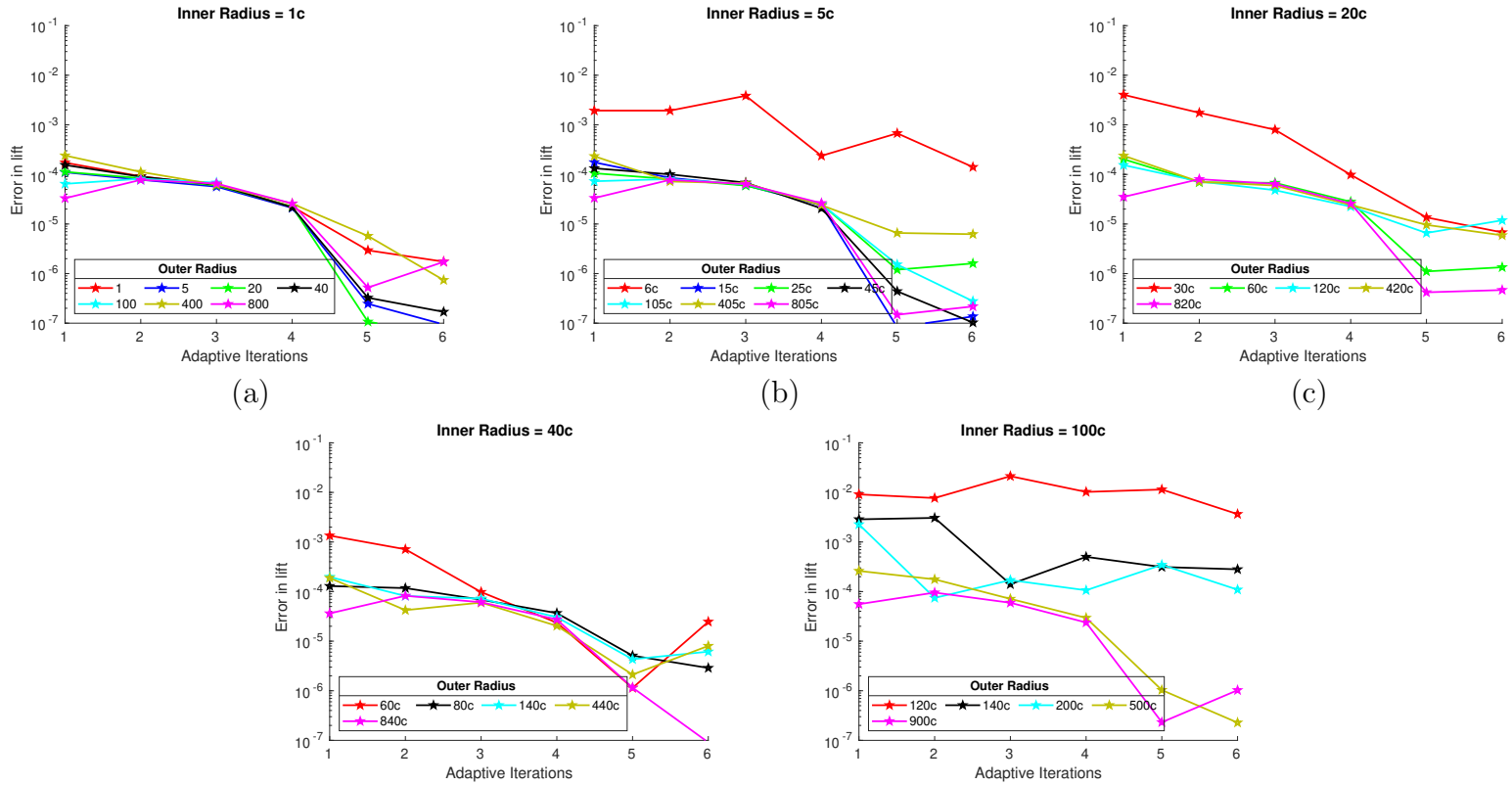


Figure 5.17: Error in lift generated by a NACA0012 airfoil in a inviscid flow as a function of the adaptive iterations for the third mesh adaptation strategy, where the mesh is optimized for the reference lift evaluation of the airfoil.

that signifies the region of influence of the displacement at a particular boundary node for any mesh deformation scheme. Other deformation approaches, including interpolation methods and PDE-based methods, also have tunable parameters and thus likely face similar issues. The output convergence study verifies that the implementation of a GCL is not necessary for achieving high accuracy in high-order FSI simulations involving rigid-body motions. Guidelines for initial mesh generation for steady FSI simulations are derived from the two test cases used in this study. When dealing with mesh deformation, the four significant conclusions from this work are:

1. Deforming the mesh in regions where the mesh density is high is favorable for output convergence.

2. Large gradients in deformation occurring within an element are difficult to resolve and should be avoided, especially if the element size is large. Thus, having a blending region extend up to the farfield promotes good convergence and lower gradients within an element.

3. Incorporating mesh deformation in the initial mesh generation process gives better output convergence.

4. Using output-based adaptation leads to balanced errors from mesh-motion and the discretization.

These guidelines can also be applied to other mesh-motion algorithms with a user defined blending region to achieve low mesh-motion errors and better output convergence. The two cases presented in this chapter have been able to demonstrate the use of output-based mesh adaptation in efficiently reducing the spatial errors generated by the mesh distortion as well the spatial discretization, thus, showing its applicability to FSI simulations. The effect of the distribution of elements in the initial mesh on the output convergence is highlighted for high-order FSI simulations.

CHAPTER VI

Pitching-Plunging Airfoil

Goal-oriented mesh adaptation has been applied extensively for computational fluid dynamics applications in aerospace engineering [47]. However, less work has been done in the application of such adaptation techniques for FSI problems, specifically in computational aeroelasticity (CAE). This chapter motivates the development of goal-oriented mesh adaptation for CAE problems by showcasing the efficiency of the adapted space-time meshes using an uncoupled adjoint on a classic aeroelastic case. The benefits of adaptivity in two-dimensional aeroelastic problems is investigated for coupled outputs on a pitching-plunging airfoil in a high Reynolds number flow. Lastly, the impact of mesh motion on FSI simulations is also highlighted in the flutter prediction study.

6.1 Case Setup

A two-dimensional pitching-plunging airfoil is a common aeroelastic model which has been studied extensively [18]. The two degree-of-freedom system, when exposed to an airstream of Mach number M_f, exhibits a dynamic instability known as flutter where the two aeroelastic modes, pitch and plunge, coalesce, resulting in a self-sustained simple harmonic motion of the structure. The Mach number M_f represents the stability boundary for the system, above which the oscillations grow in amplitude.

To study the phenomenon of flutter, consider a NACA 0012 airfoil pinned at it's elastic axis in a free-stream flow of Mach number M_∞ as shown in Figure 6.1. The two degrees of freedom of the airfoil are the plunge h, which is taken to be positive in the downward direction, and the pitch angle α, which is considered positive clockwise (pitch up). The geometric properties of the airfoil are the chord c. Measured from the nose of the airfoil, the center of mass and the elastic axis are located at x_{cg} and x_{ea} respectively. The inertia properties of the airfoil are the mass, m, and the moment of inertia about the elastic axis, I_{ea}. The airfoil is connected to two springs at the elastic axis: a plunge spring that represents the bending stiffness, K_h, of the structure, and a torsional spring that represents the torsional stiffness, K_α. The equations of motion of the pitching-plunging airfoil are given as:

$$m\ddot{h} + S\ddot{\alpha} + K_h h = -L, \tag{6.1}$$

$$S\ddot{h} + I_{ea}\ddot{\alpha} + K_\alpha \alpha = M, \tag{6.2}$$

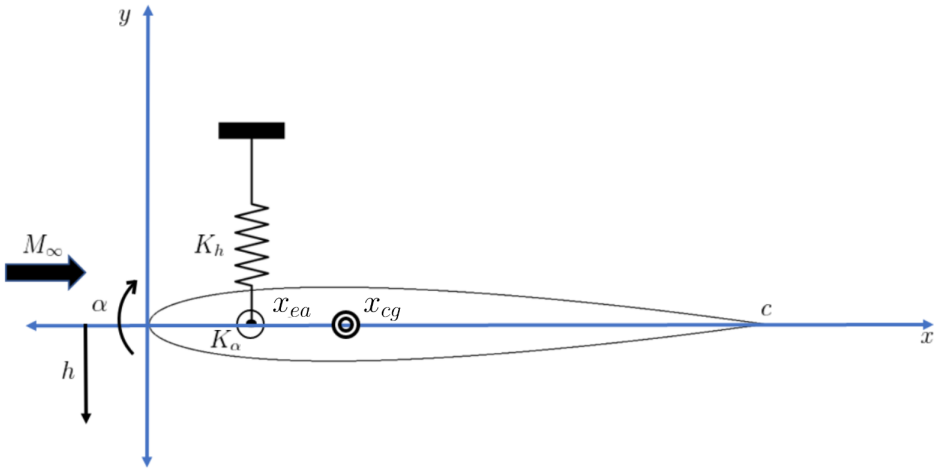
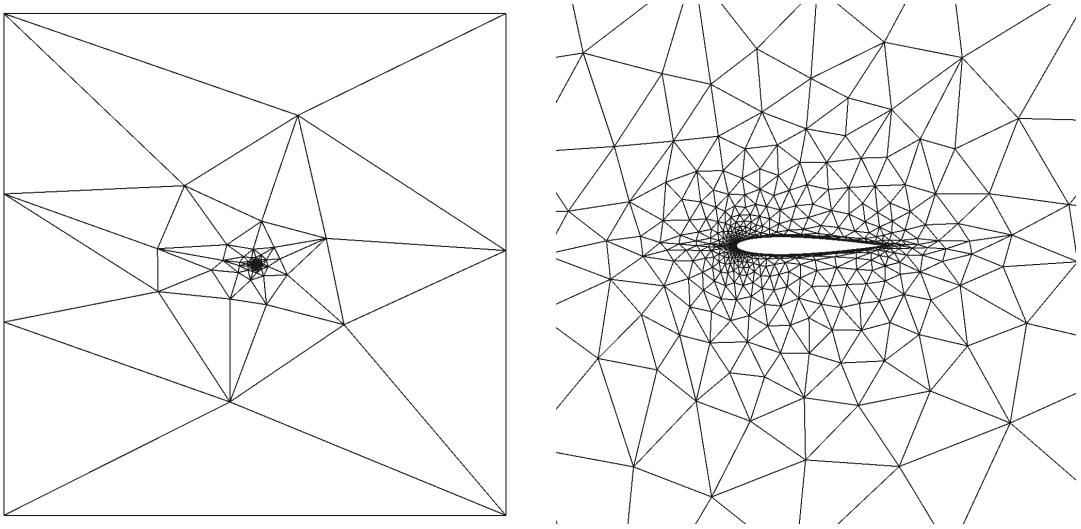


Figure 6.1: Model of a two-degree-of-freedom pitching-plunging airfoil.



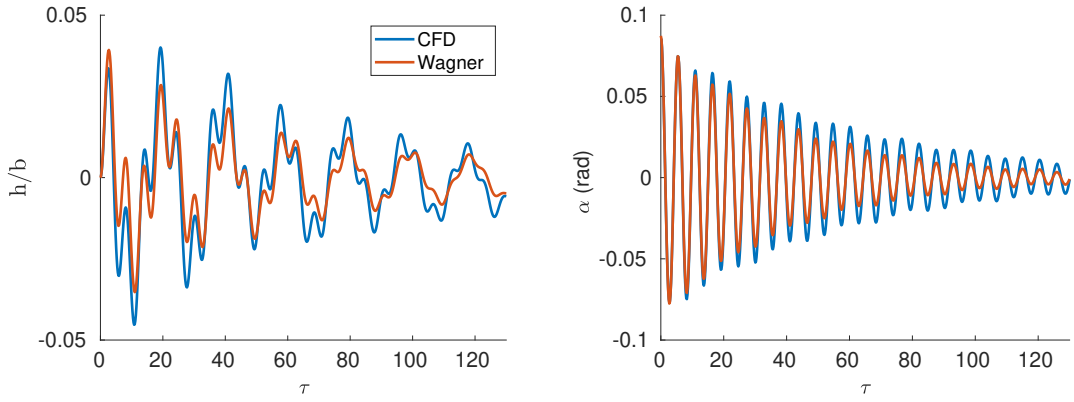
(a) (b)

Figure 6.2: Unstructured viscous mesh for the pitching plunging airfoil.

where S is the static imbalance defined by the product between $(x_{cg} - x_{ea})$ and the mass of the airfoil, L is the total lift acting on the airfoil and M is the net moment on the airfoil about the elastic axis. The non-dimensional parameters used to describe the model are defined as:

$$\mu = \frac{m}{\pi \rho_\infty b^2}, \quad \bar{\omega} = \frac{\omega_h}{\omega_\alpha}, \quad r_\alpha = \sqrt{\frac{I_f}{mb^2}}, \quad \chi = \frac{S}{mb}, \tag{6.3}$$

where μ is the mass ratio, $\bar{\omega}$ is the ratio of the uncoupled natural frequencies, and r_α and χ are the non-dimensional inertia and static balance, respectively. The uncoupled, in-vacuum, natural pitching and plunging frequencies are defined as $\omega_h = \sqrt{K_h/m}$ and $\omega_\alpha = \sqrt{K_\alpha/I_f}$, respectively. The case setup has been taken from Sanchez et al. [107] where the non-dimensional parameters are set as $\mu = 100$, $\bar{\omega} = 0.3185$, $r_\alpha = 0.5$ and $\chi = 0.25$, to ensure a subsonic flutter. The pitching frequency is set to be $\omega_\alpha = 45$ rad/s and the elastic axis is located at the quarter chord of the airfoil. The Reynolds number is set to $Re = 4 \times 10^6$ and the chord, $c = 1$ m. The airfoil is set at an initial angle of attack of five degrees to the flow.

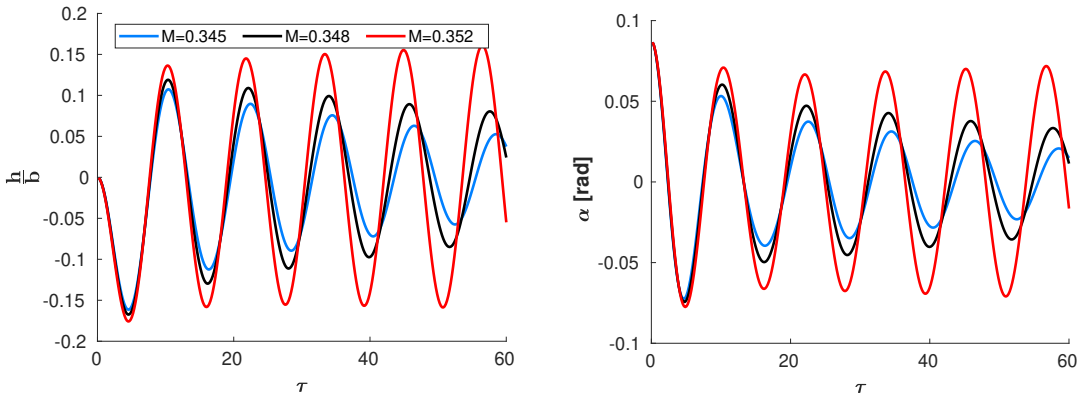


(a) Non-dimensional plunge displacement (b) Pitch angle

Figure 6.3: Aeroelastic response of the airfoil in the subcritical regime as a function of the non-dimensional time (τ).

A coarse, unstructured, triangular, viscous mesh of 2766 elements is generated for the coupled simulation, as shown in Figure 6.2. The airfoil is located centrally in the domain, the boundary of which consists of a square box which spans from $-100c$ to $100c$ in both dimensions. The fluid flow is simulated using a RANS solver with the SA turbulence model [24]. The coupled system uses the fourth-order time scheme, ESDIRK4, introduced in the Section 2.3, to march forward in time. A steady-state flow solution is used as the initial condition for the simulation. A mesh motion algorithm based on the explicit approach is applied to handle the deformations occurring in the fluid subsystem due to the moving airfoil. The mesh motion algorithm divides the spatial domain into two sub-domains. The region extending up to a radial distance of one chord away from the elastic axis is deformed rigidly according to the deformation provided by the structures solver. Following the region of rigid deformation, a septic polynomial blends the deformation smoothly in the region extending between a radial distance of one chord and ten chords resulting in zero deformation at a radial distance of ten chords.

Using the parameters defined above, the aeroelastic response of the coupled system



(a) Non-dimensional plunge displacement (b) Pitch angle

Figure 6.4: Aeroelastic response of the airfoil near the flutter boundary for different values of the uniform flow Mach number.

is verified by conducting a simulation at a subcritical Mach number, $M_\infty = 0.1$. The simulation is performed with $p = 3$ order polynomials for the fluid spatial discretization and 60 time steps per oscillation of the plunge mode of the airfoil. The response of the coupled system is compared against Wagner's incompressible aerodynamic model [52]. Figure 6.3 shows the non-dimensionalized unsteady pitch and plunge displacements as functions of the non-dimensional time, $\tau = \omega_\alpha t$. The unsteady pitch and plunge displacements are consistent with the theoretical aerodynamics model at the subcritical Mach number as well as other CFD solvers as shown in Sanchez et al. [107]. To determine the flutter boundary of the system, the aeroelastic response is recorded for varying Mach numbers. The damping coefficient of the plunge displacement in each run is evaluated by the logarithmic decrement approach and fitted by a quadratic polynomial to estimate the actual flutter Mach number. Figure 6.4 shows the aeroelastic response at three different Mach numbers close to flutter using $p = 3$ order polynomials for the spatial discretization. The responses at M_∞=0.345 and M_∞=0.348 lie in the sub-critical regime while the response at M_∞=0.352 is past the flutter boundary as the states grow without bound over time. The flutter Mach number is estimated to be at M_f=0.351 by interpolating using the damping evaluated at

these Mach numbers. The estimated flutter Mach number differs in the order of 10^{-3} from that obtained by Sanchez et al [107]. The difference in the flutter Mach number is attributed to the difference in the fidelity of the fluid solvers, where a second-order finite volume solver is used in Sanchez et al.

6.2 Coupled-Adjoint Verification

To verify the coupled adjoint, a smooth laminar flow with a Reynolds number of $Re = 1000$ is simulated for five time steps with a final time of 0.005. The unsteady simulation is only run for a few such time steps to reduce the error arising from the temporal discretization. With this case setup, the unsteady coupled adjoint is verified by comparing the sensitivities of the outputs of interest against finite differences. Two outputs of interest from both the fluid and the structural subsystem are chosen: the time-integrated lift and the pitch displacement and the final-time lift and the pitch displacement. Using the adjoint formulation, the sensitives of the outputs of interest

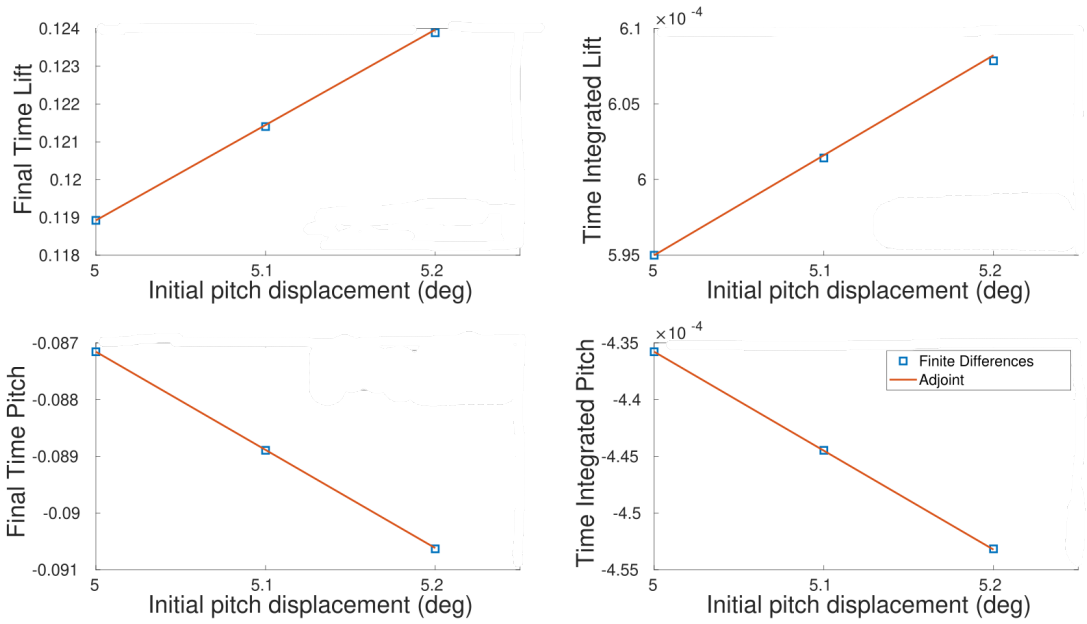


Figure 6.5: Verification of the coupled adjoint using a parameter sensitivity test.

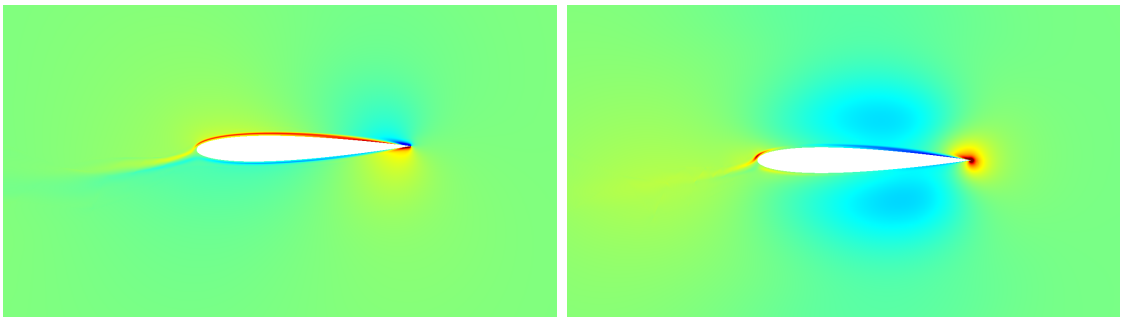
are evaluated with respect to the initial pitch angle of attack, which is set at time $t = 0$,

$$\frac{\partial J}{\partial \mathbf{U}^s}\Big|_{t=0} = \mathbf{\Psi}^{s,T}\mathbf{M}^s\Big|_{t=0}. \tag{6.4}$$

where, $\mathbf{\Psi}^s$ is the structural adjoint and $\mathbf{M}^s$ is the mass matrix of the first order semi-discrete form of the governing structural equations (Eq 2.26). To evaluate the sensitivities from finite differences, a perturbation is given to the initial pitch angle of attack and the four outputs of interest are then evaluated again. The differences between the outputs in the perturbed condition and the original condition are used to evaluate the sensitivities. For the chosen outputs of interest, the agreement is excellent for small perturbations, as shown in Figure 6.5. The spatial order of interpolation is taken to be one for verification; however, similar agreement is obtained for higher spatial orders of interpolation as well. This verifies the implementation of the coupled adjoint in the FSI solver.

6.3 Mesh Adaptation

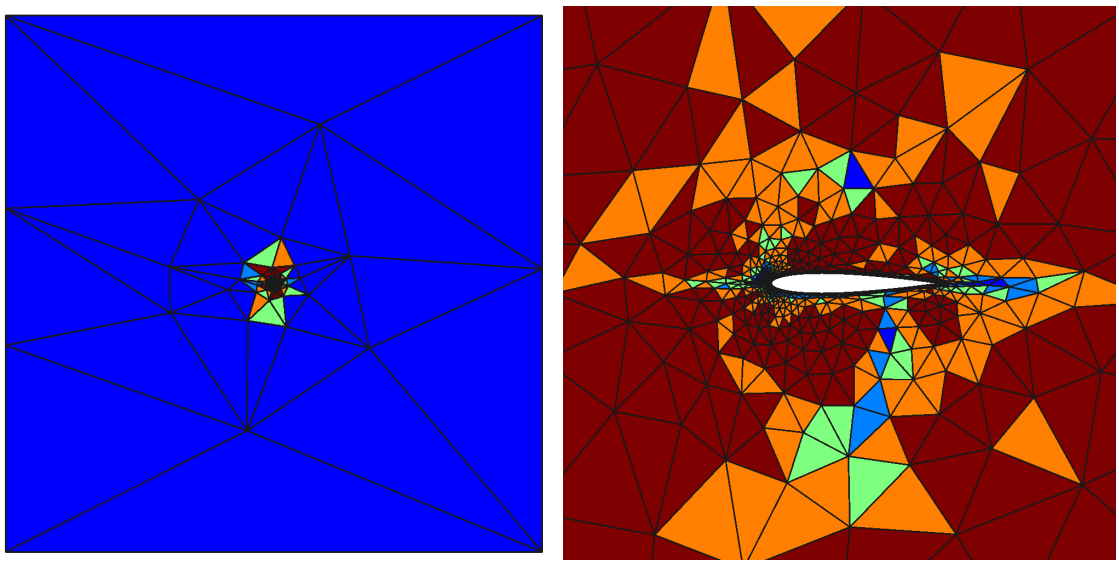
The verified coupled adjoint can be used in adapting the fluid and structural meshes for various outputs of interest, as shown in Section 4.2. In this work, the focus is on adapting fluid meshes for both fluid and structural outputs of interest.



(a) Conservation of x-momentum adjoint (b) Conservation of y-momentum adjoint

Figure 6.6: Fluid adjoint solution for time-averaged lift coefficient at the initial time.

For mesh adaptation, the case setup is slightly changed, and a turbulent flow at a Reynolds number of $Re = 4 \times 10^6$ is used. At this flow condition, the Mach number of $M = 0.345$ is slightly below the flutter boundary of the structure and large oscillations are observed in the structure which damp in time. The fluid mesh is adapted for the time-integrated lift coefficient and time-averaged pitch displacement from time $t = 0$ to $t = 30$, during which time the airfoil undergoes one complete pitch oscillation. Figure 6.6 shows the conservation of the x-momentum and y-momentum components of the fluid adjoint at the initial time. The magnitude of the fluid adjoint decreases away from the airfoil, showing that the output is most sensitive to the residuals on the elements above and below the airfoil, especially close to the trailing edge. The coupled adjoint detects the streamlines enveloping the structure as well as the regions in the mesh which are affected by the motion of the airfoil. Using the coupled adjoint, an unsteady error estimate for the output is evaluated and used to adapt the



(a) (b)

Figure 6.7: Element order distribution at the end of the adaptation process on the pitching-plunging airfoil. The elements colored dark blue denote $p = 1$, light blue denote $p = 2$, green denote $p = 3$, orange denote $p = 4$, and dark red denote $p = 5$ order elements.

fluid mesh. The errors in the coupled output arise due to the spatial discretization errors in the fluid subsystem and the temporal discretization in both the fluid and the structural subsystem. A p-refinement study of the coupled simulation corroborated the effects of spatial discretization on the output for the two-degree of freedom system. An equivalent time-refinement study did not result in a significant change due to the fourth-order order time scheme, ESDIRK4, and a small time step used in the coupled simulation. Therefore, the spatial discretization errors dominate the output evaluation and an adaptive procedure is used to obtain an optimized fluid mesh. The total number of degrees of freedom introduced after each adaptive iteration is set by the growth factor which is $f_{tot} = 1.2$. The optimized mesh is obtained by subjecting the coarse mesh to four cycles of p-adaptation.

Figure 6.7 shows the spatial order of interpolation post-adaptation for the lift output. The higher-order elements ($p = 4$ and $p = 5$) introduced in the adaptation process are those with the highest error estimates and lie primarily in the vicinity of the airfoil and inside the blending region. Adaptation occurring in the blending region seems counter-intuitive as the adjoint on these elements is not as large in magnitude as on the elements inside the rigid deformation region, as seen in Figure 6.8. The

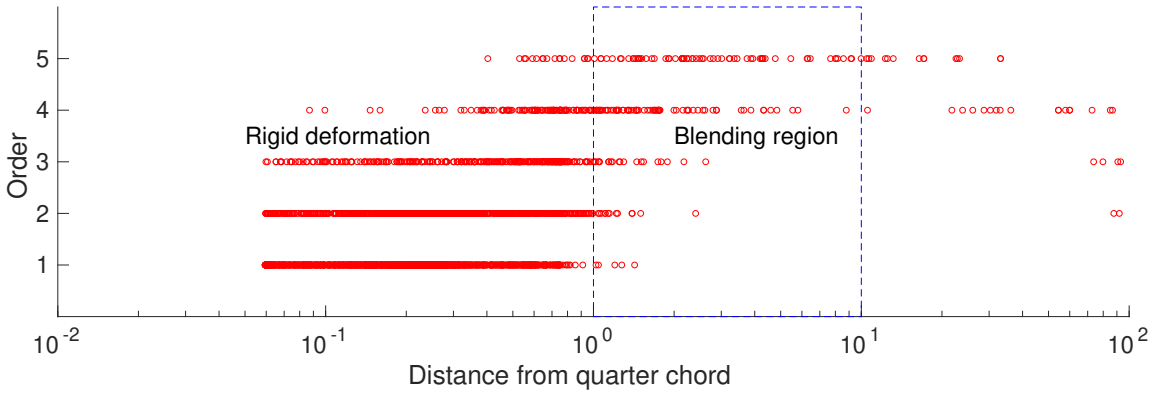
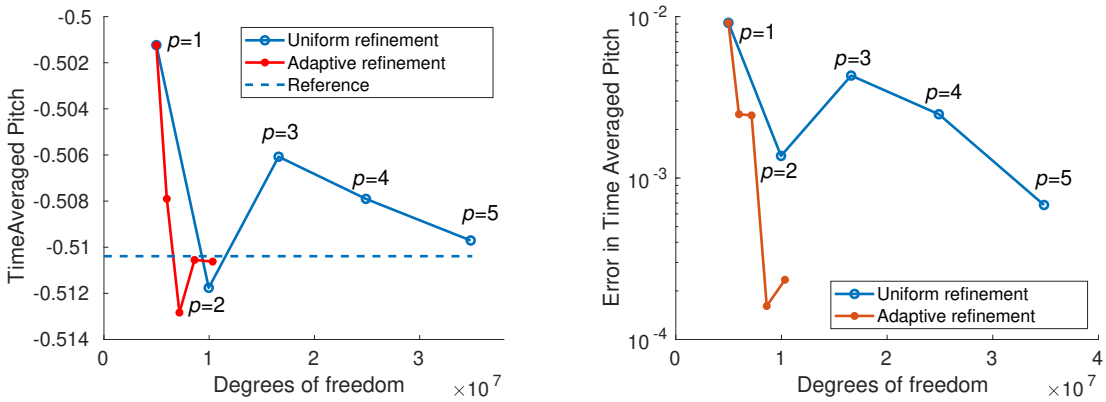


Figure 6.8: Spatial distribution of the element spatial order at the end of the adaptation process.

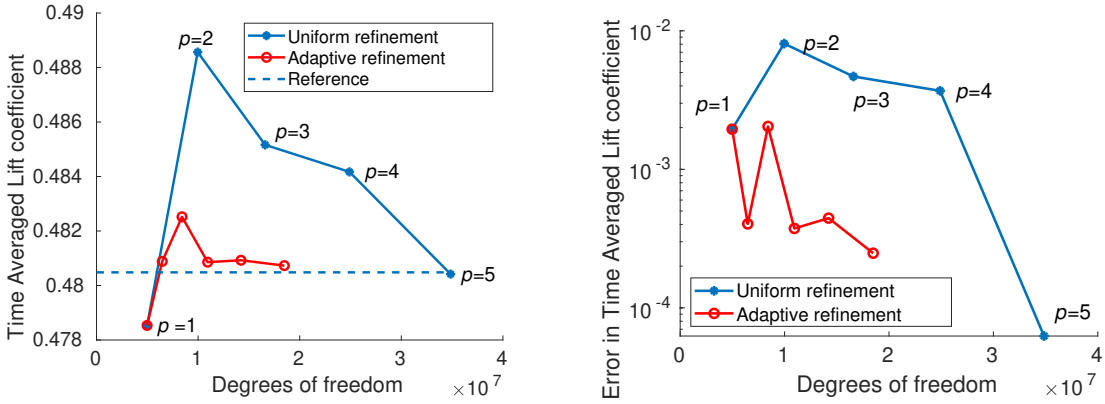


(a) Pitch convergence (b) Error convergence

Figure 6.9: Comparison of the time-averaged pitch for output-based adaptation to uniform-refinement.

reason behind elements in the blending region being chosen for adaptation also goes back to the definition of the adaptive indicator which is a function of the adjoint and the residual evaluated by projecting the coarse space solution into the fine space. The elements inside the blending region, despite having a lower adjoint magnitude, suffer from the errors originating from mesh motion algorithm. These errors result in a larger fine-space residual term in the adaptive indicator. The mesh motion errors become smaller with higher-order approximation [65] but are dominant at lower-order approximation of the state, where the adaptation begins. The noisy nature of mesh adaptation can also be attributed due to the distribution of the elements in the initial fluid mesh. The initial anisotropic fluid mesh has smaller anisotropic elements next to the airfoil, which introduce less error in the output of interest. Thus, the cells targeted for mesh adaptation are present away from the airfoil, where they are bigger and more irregular.

Goal-oriented adaptation using an uncoupled fluid adjoint for FSI problems is unable to capture the elements directly above and below the airfoil [94], which are important for the structural subsystem, thus demonstrating the importance of using



(a) Lift convergence (b) Error convergence

Figure 6.10: Comparison of the time-averaged lift coefficient for output-based adaptation to uniform-refinement.

the coupled adjoint for goal-oriented adaptation in FSI simulations. Figure 6.9 compares the convergence of the time-averaged pitch displacement for the adapted meshes against uniform p-refinement. At every stage of uniform refinement, the spatial order of the elements in the entire domain is increased by one. The reference/truth output, is evaluated with $p = 6$ elements in the entire spatial domain. The plot shows the effectiveness of the adaptation process relative to uniform refinement. Figure 6.9 also compares the error in time-averaged pitch displacement for the adapted meshes against uniform p-refinement. The adapted meshes converge at a faster rate with fewer degrees of freedom, thereby increasing accuracy and reducing computational cost. These advantages can also be seen when a similar mesh adaptation study conducted is for a fluid output of interest, the time-averaged lift coefficient as shown in Figure 6.10. As the importance of mesh motion errors arising in the blending region is evident from the spatial distribution of the higher-order elements, the blending region is thereby varied to show its effect on output evaluation in FSI simulations. Keeping the inner radius of the blending region constant at one chord, the outer radius is varied throughout the domain to show the effect of the position of the blending regions on the flutter Mach number calculations and unsteady lift evaluation. Figure 6.11

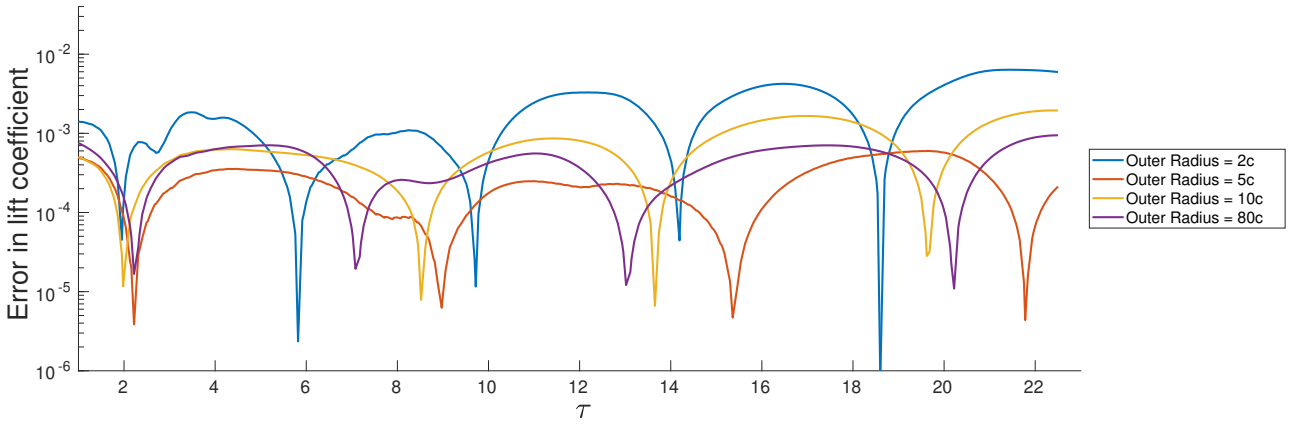


Figure 6.11: Error in the unsteady lift for blending regions with constant inner radius of 1c and varying outer radii, post-adaptation.

compares the absolute error in the time history of lift for the various blending regions, post-adaptation, against the reference unsteady lift. The results corroborate the existence of an optimum blending region close to the airfoil, seen previously in Section 5.1.

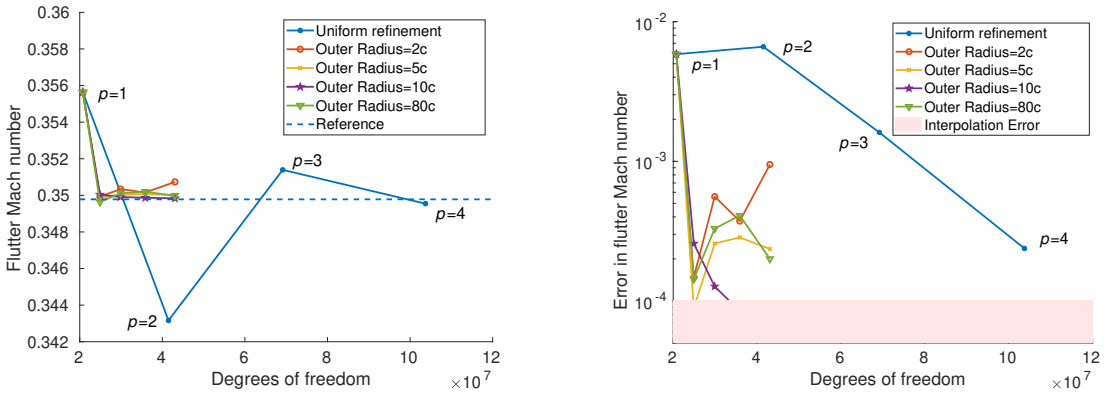
As the governing equation of the structural subsystem depends on the lift, adapting on the unsteady lift will directly affect the flutter Mach number evaluation as well. Thus, the fluid meshes adapted for the time-integrated lift output can also be reused for accurately predicting the flutter Mach number. Figure 6.12(a) compares the convergence of the flutter boundary Mach number for the first five adapted meshes against uniform p-refinement. The adapted meshes have been obtained for varying outer radii and a fixed inner radii of one chord. The reference/truth flutter Mach number, $M_f = 0.3498$, and the unsteady lift are evaluated on a highly refined viscous mesh with 10492 elements with $p = 4$ elements in the entire spatial domain. The fine mesh is generated by refining the coarse mesh anisotropically. A metric is introduced for the size of each element which is scaled by a factor to obtain the refined mesh. The plot shows the effectiveness of the adaptation process relative to uniform refinement. Figure 6.12(b) compares the absolute error in flutter Mach number for the adapted meshes against uniform p refinement. The effect of the blending region is not very

Aeroelastic simulation	For a given accuracy on a coupled output	
	Memory saving using adaptive mesh refinement	Time saving using adaptive mesh refinement
NACA0012 Airfoil	68%	63%

Table 6.1: Cost savings using goal-oriented mesh adaption in a pitching-plunging NACA0012 airfoil subjected to subsonic flow.

distinct due to the interpolation error, which dominates the evaluation of the flutter Mach number from the damping coefficient of the plunge displacements, as shown as the shaded region in the figure. The adapted meshes, irrespective of the position of the blending region, converge at a faster rate with fewer degrees of freedom, thereby increasing accuracy and reducing computational cost.

Table 6.1 summarizes the benefits of goal-oriented mesh adaptation for the two-dimensional aeroelastic case. The benefits showcased in the table are evaluated by comparing the degrees of freedom in the final adapted fluid mesh against the reference fluid mesh. Adapted meshes can provide similar or better level of accuracy in coupled outputs with almost one-third the number of degrees of freedom, when compared against uniformly refined meshes. The table doesn't take into considera-



(a) Convergence of flutter boundary for various blending regions

(b) Error in the flutter boundary for various blending regions

Figure 6.12: Comparison of the flutter boundaries for output-based adaptation to uniform mesh refinement.

tion the time involved in obtaining the adapted meshes by solving the adjoint systems.

In this chapter, an adaptive meshing procedure is applied to obtain an accurate prediction of coupled outputs for a pitching/plunging NACA0012 airfoil. The adaptive meshing procedure increases the total computational time of the coupled simulations, due to the additional adjoint evaluation, by a factor of two. However, the significant benefit of higher accuracy at lower degrees of freedom outweighs the increase in the computational expense. The impact of the mesh motion errors on FSI simulations is highlighted significantly in the adapted meshes and the existence of optimum blending region, as discussed in Chapter V, is also showcased in FSI problems. Overall, the adaptive meshing procedure provides significant increase in the accuracy of unsteady outputs for two-dimensional aeroelastic problems compared to more common adaptation methods, such as uniform refinement.

CHAPTER VII

Aeroelastic Beam

This chapter applies goal-oriented mesh adaption to a two-dimensional aeroelastic case. The high-fidelity FSI solver is first validated by conducting a convergence study. The coupled adjoint solver is then verified by conducting sensitivity perturbation tests for a variety of coupled outputs. Lastly, the resulting fluid and structural spatial error estimates, obtained using the coupled adjoint, are verified against the spatial errors obtained from finite differences. Using these error estimates, goal-oriented mesh adaptation is conducted on three coupled outputs of interest, of varying complexity. The benefits of coupled adaptive mesh refinement are demonstrated by comparing output convergence against standard mesh refinement approaches.

7.1 Case Setup

Consider the flow around an Euler-Bernoulli cantilever beam fixed at the leading edge. At the undeformed strain-free position, the beam makes an angle of attack of five degrees to the flow. The actual beam is considered two-dimensional when represented in the fluid mesh, with a length of 1m and a thickness of 0.005m. The initial coarse fluid mesh is represented by an unstructured square grid consisting of 758 nodes and 871 triangular elements. $p = 1$ order polynomials have been used for the fluid spatial discretization uniformly in the entire domain. The domain extends

up to a distance of 100 chords from the beam, which is cantilevered at the center of the domain. Free-stream boundary conditions are applied at the outer boundary of the mesh. The planar beam is represented by a coarse initial structural mesh, consisting of 16 beam finite elements of uniform size in the structural solver. The cantilevered beam is subjected to a uniform fluid flow with a Mach number of $M_\infty = 0.345$ and Reynolds number of $Re = 1000$. The Euler-Bernoulli beam has a bending stiffness $EI = 0.2Nm^2$ and density, $\rho = 1kg/m$. Each beam element has two nodes with two degrees of freedom per node. Proportional damping in the form of Rayleigh damping with parameters, $\alpha = 1$ and $\beta = 1$ is used to model the structure.

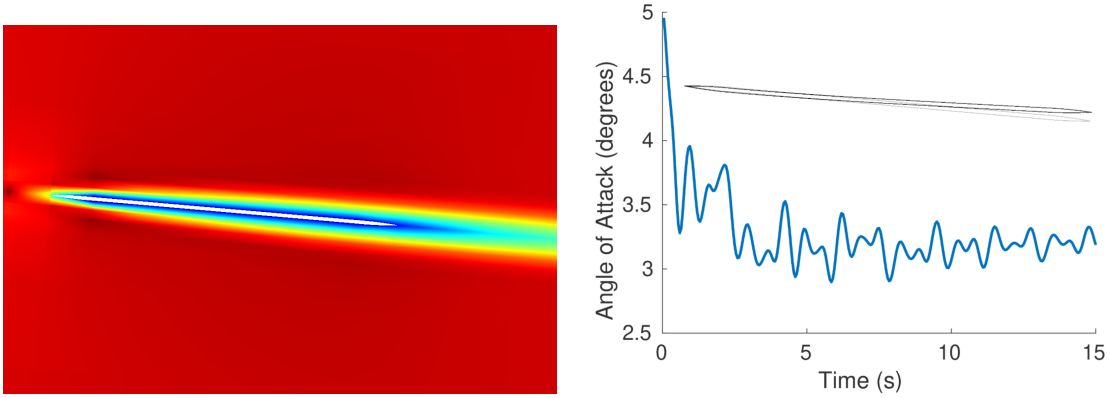
$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K} \tag{7.1}$$

The equations of motion for the cantilevered beam are given as

$$\bar{\mathbf{M}}\ddot{\mathbf{u}}^s + \mathbf{C}\dot{\mathbf{u}}^s + \mathbf{K}\mathbf{u}^s = \mathbf{F}^s, \tag{7.2}$$

where $\mathbf{u}^s$ is the displacement at a particular node, $\bar{\mathbf{M}}$ is the mass matrix, $\mathbf{K}$ is the stiffness matrix, $\mathbf{C}$ is the Rayleigh damping matrix and $\mathbf{F}^s$ is the force vector. At time $t = 0$, the structure is at rest and the fluid state is obtained from a steady solution for the steady solution. The two subsystems are coupled temporally using the IMEX scheme, discussed in Section 3.2, where each subsystem marches in time using the ESDIRK4 time scheme.

Due to matching boundaries of the structural boundary and the wetted surface, no projection is needed from the boundary of the beam and its wetted surface. Radial basis functions, as described in Section 2.1.5.2, are used to transfer the displacement from the structural subsystem to the fluid subsystem. For transferring forces between the fluid and structural subsystem, the pressure is evaluated at 10 quadrature points



(a) x momentum at final time, $T = 15$ (b) At steady state the beam develops a negative camber

Figure 7.1: Time evolution of the motion of the beam.

for a single fluid element at the interface. These data are then transferred to the structural solver, which fits a fifth-order polynomial spline to approximate the pressure distribution within each structural element. For the structural elements which overlap, partially or fully, with the fluid elements, the contribution to the force vector is evaluated by integrating the pressure over the overlapped length with the shape function of the element. The benefits of using such a high number of quadrature points is the ability to approximate the pressure distribution accurately, even for high-order elements on the fluid mesh.

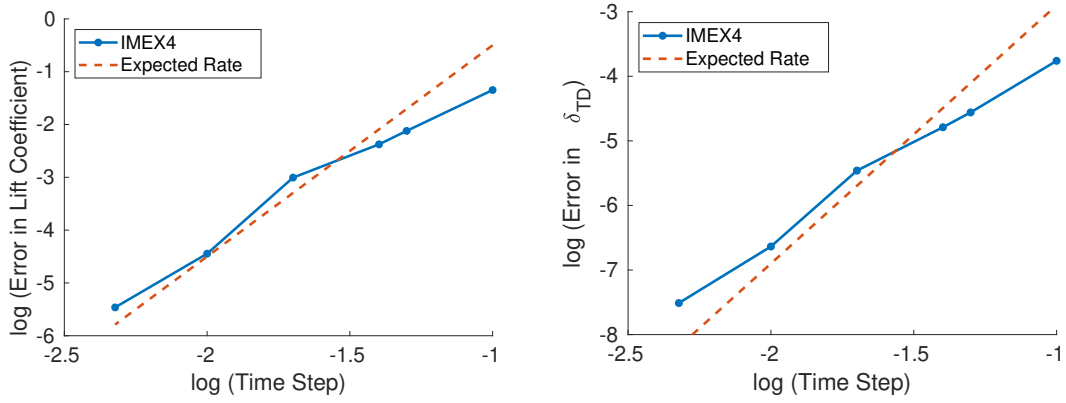
At time $t = 0$, the structure is at rest and the fluid state is obtained from a steady solution. $p = 1$ order polynomials have been used for the fluid spatial discretization and the time scheme for both subsystems is ESDIRK4. The two subsystems are coupled temporally using the IMEX scheme. For this flow condition, a convex nature in the beam shape is expected at steady state due to the lift force. This is corroborated by Figure 7.1(a), where the final time is $T = 15$. Figure 7.1(b) presents the evolution of the angle of attack on the beam over time for this flow condition. The angle of attack is measured as the angle made between the chord line and the horizontal flow

direction.

To validate the temporal convergence of the high-order coupled solver, the absolute error in the lift coefficient and non-dimensionalized tip displacement are measured at the final time, $T = 4$. A plot of the observed relative error as a function of time step for the ESDIRK4 scheme is shown in Figure 7.2. Note that in each case, the scheme exhibits convergence at the designed rate. The plot demonstrates that this partitioned approach attains up to fourth-order accuracy in time with the IMEX coupling scheme.

7.2 Coupled-Adjoint Evaluation and Verification

For the one-dimensional aeroelastic case, studied in Chapter VI, the coupled fluid adjoint was sufficient in evaluating the error estimates due to the lack of spatial discretization errors in the structural subsystem. However, for general CAE problems, where the fluid and structural subsystems are discretized both in space and time, a coupled fluid and structural adjoint provides the most accurate evaluation of the error estimates. The coupled adjoint equation is solved backwards in time, after initializing



(a) Lift (b) Tip displacement

Figure 7.2: Convergence study for the IMEX scheme.

at final time, as shown in Section 4.1.1. Just like the primal equations, the coupled adjoint equations are solved in a partitioned manner using the IMEX scheme. The off-diagonal derivatives in the coupled adjoint equations are evaluated using finite differences. The term, $\partial \mathbf{r}^s / \partial \mathbf{U}^f$, which represents the change in the fluid residual due to perturbation in the structural state, is evaluated by perturbing the position of every structural node at the interface and reevaluating the fluid residual. Due to low number of structural nodes at the interface of the cantilevered beam and the low cost of evaluating the fluid residual for a two-dimensional case, the bottleneck of the simulation still remains the non-linear solution of the fluid subsystem. The other off-diagonal term, $\partial \mathbf{r}^f / \partial \mathbf{U}^s$, which represents the change in the structural residual due to perturbation in the fluid state, is evaluated by linearizing the overall lift on the beam with respect to the fluid states.

To verify the coupled adjoint, a smooth laminar flow with a Reynolds number of $Re = 1000$ and a Mach number of $M = 0.345$ is simulated for a single time step with a final time $T = 0.0001$. The unsteady coupled adjoint is verified by comparing the sensitivities of the outputs of interest with respect to the initial shape of the beam against finite differences. Two outputs of interest from both the fluid and the structural subsystem are chosen: (i) the time-integrated lift and the tip displacement, and (ii) the final-time lift and the tip displacement. The initial shape of the beam at $t = 0$ is given by

$$u^s\Big|_{t=0} = A \sin(x), \tag{7.3}$$

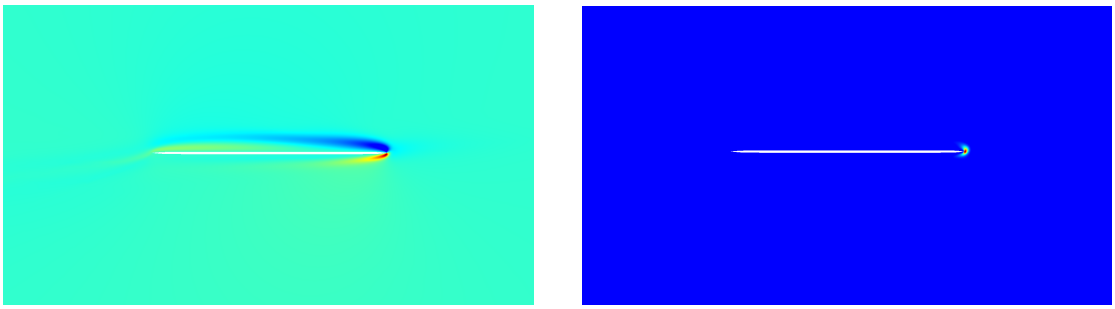
where the amplitude variable, A, is perturbed. Using the adjoint formulation, the sensitives of the outputs of interest are evaluated with respect to the initial shape of the beam, set at time $t = 0$. Using the adjoints, the sensitivities are given as

$$\frac{dJ}{d\mathbf{U}^s}\Big|_{t=0} = \mathbf{\Psi}^{s,T} \bar{\mathbf{M}}\Big|_{t=0} - \dot{\mathbf{\Psi}}^{s,T} \mathbf{C}\Big|_{t=0}. \tag{7.4}$$

$$\left.\frac{dJ}{dA}\right|_{t=0} = \left.\frac{dJ}{d\mathbf{U}^s}\right|_{t=0} \times \left.\frac{d\mathbf{U}^s}{dA}\right|_{t=0}. \tag{7.5}$$

To evaluate the sensitivities from finite differences, a perturbation is given to the amplitude variable, A, and the four outputs of interest are then evaluated again. The differences between the outputs in the perturbed condition and the original condition are used to evaluate the sensitivities. For the chosen outputs of interest, the agreement is excellent for small perturbations, as shown in Figure 7.4, and it verifies the implementation of the beam coupled adjoint.

Having verified the coupled adjoint, error estimates based on the coupled adjoint, as shown in Section 4.1.2, are used to adapt the fluid and structural meshes for coupled outputs of interest. In this work, the spatial error estimates are used to adapt in the spatial domain of the aeroelastic problem while keeping the temporal discretization fixed. The spatial error estimates of a coupled output of interest for an individual subsystem is verified by finite differences. To evaluate the error estimate using finite differences, the output of interest is reevaluated by projecting the solution to a finer space, which serves as the reference output of interest. The finer space in the fluid subsystem is obtained by increasing the spatial order of approximation on every triangular element by one. In the structural system, the finer space



(a) Conservation of x-momentum adjoint

(b) Conservation of y-momentum adjoint

Figure 7.3: Coupled adjoint solution for time-averaged lift coefficient output at the initial time.

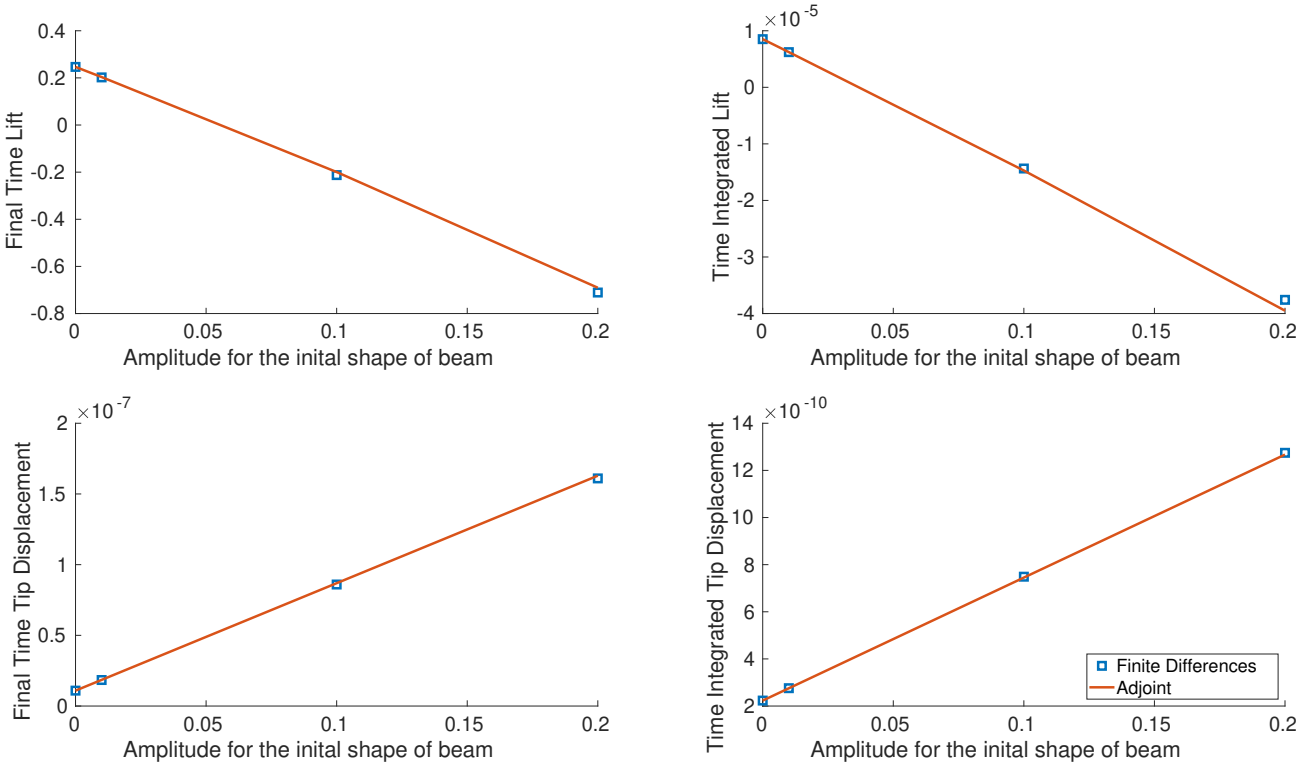


Figure 7.4: Verification of the coupled adjoint using a parameter sensitivity test.

is obtained by subdividing every beam element uniformly into two beam elements of equal lengths. For the structural error estimate, the finer space is defined by keeping the fluid spatial discretization fixed while refining the structural spatial discretization and the temporal discretization.

The difference between the outputs in the coarse and fine space will comprise of the errors due to the spatial discretization in the structural mesh and the temporal discretization. The coarse space has uniform $p = 1$ elements in the fluid mesh and 16 uniform Euler-Bernoulli elements while the finer mesh has uniform $p = 2$ elements in the fluid mesh and 32 uniform Euler-Bernoulli elements. To isolate the errors from the temporal discretization, the spatial discretization is refined while keeping the structural spatial discretization fixed which provides the structural spatial error estimate, as shown in Table 7.1. The net spatial error estimate, as shown in column 6 of Table 7.1, obtained from the adjoint based method, is able to accurately predict the relative spatial error, as shown in column 3 of Table 7.1.

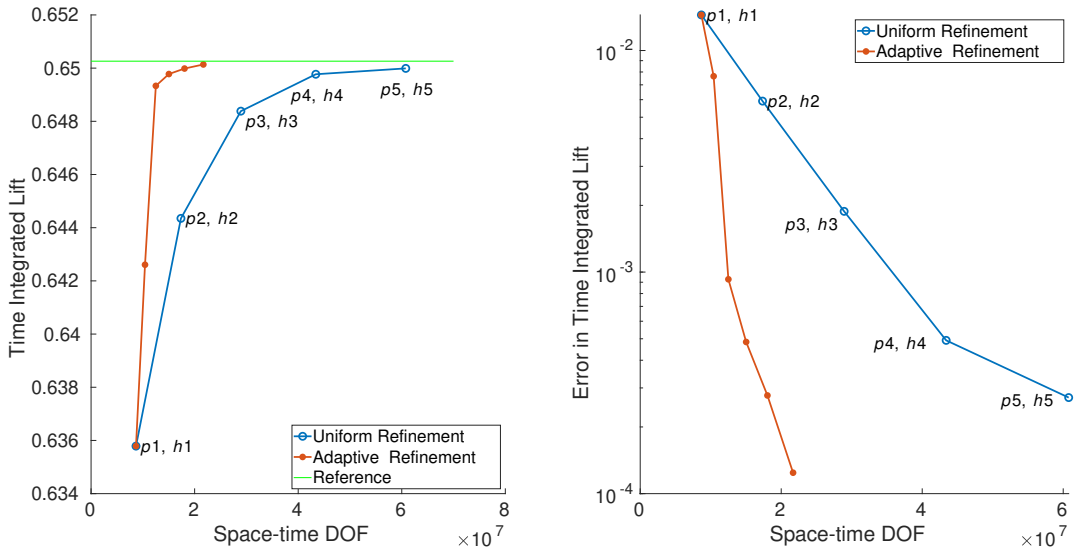
Total relative error	*Relative temporal error*	*Relative spatial error*	*Fluid spatial error estimate*	*Structure spatial error estimate*	*Total spatial error estimate*
6.77×10^{-04}	2.23×10^{-05}	6.55×10^{-04}	1.94×10^{-04}	4.87×10^{-04}	6.51×10^{-04}

Table 7.1: Comparison of the spatial error in the time-integrated tip displacement, relative to a finer approximation space and the error estimate from adjoint weighted residual.

7.3 Coupled Mesh Adaptation

Three outputs of interest are chosen for mesh adaptation for the two-dimensional aeroelastic case which are the time-averaged lift coefficient, the time-integrated tip displacement and the maximum moment developed over the beam, when subjected

to a velocity gust. For the first two outputs of interest, the case setup is the same as that described previously for the adjoint verification. The only difference is the duration of the simulation which is increased to a final time of 4 using 200 uniform timesteps. For the first output of interest, time-averaged lift coefficient output, Figure 7.3 shows the conservation of x-momentum and y-momentum components of the fluid adjoint at initial time. The sensitivity of the output by the flow at the trailing edge of the beam is highlighted by the fluid adjoint. The structural adjoint, a one dimensional field in this case, also showcases higher sensitivity of the output to the structural degrees of freedom at the trailing edge. The trailing edge is important for defining the overall camber of the beam which in turn dictates the total lift generated. Thus, the increased sensitivity of the output to the trailing edge is expected and captured well by the coupled adjoint. Error estimates using the coupled adjoint guide the mesh adaptation process. For the cantilevered beam, the initial fluid and structural mesh is subjected to five cycles of mesh adaptation, with a global growth factor of $\mathrm{f_{tot}} = 1.2$. The mesh adaptation occurs in the spatial order of mesh elements



(a) Output Convergence (b) Error in the output of interest

Figure 7.5: Comparison of the time-averaged lift coefficient for output-based adaptation to uniform-refinement.

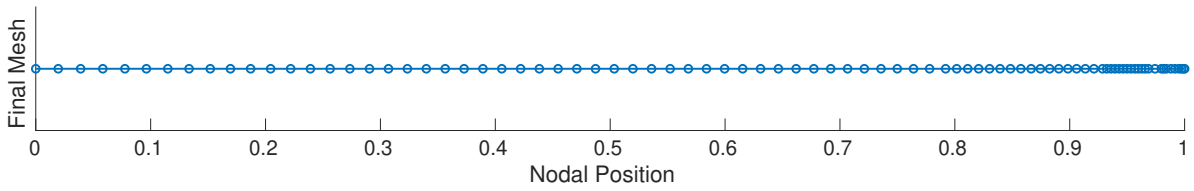
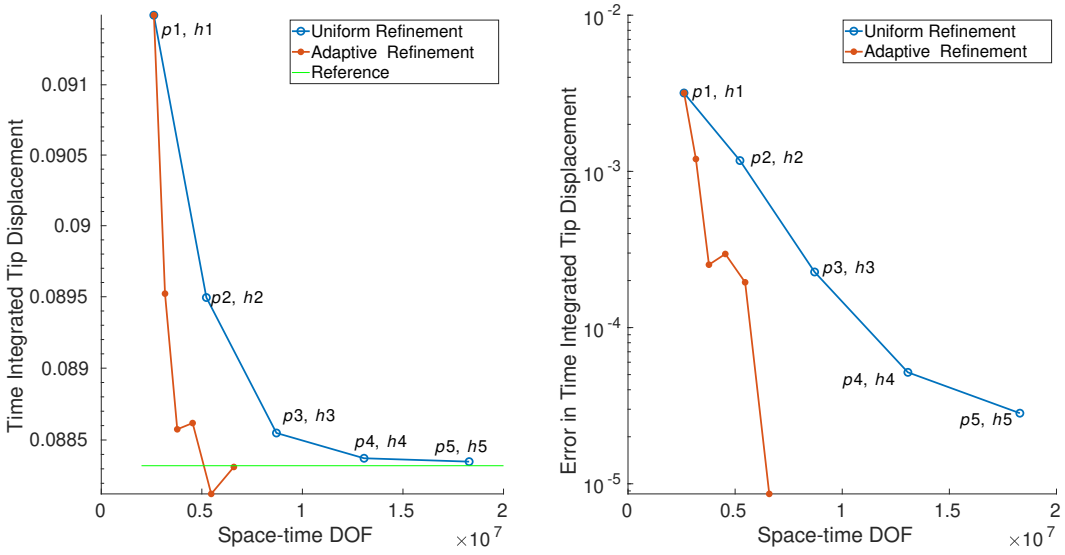


Figure 7.6: Distribution of structural nodes after mesh adaptation.

in the fluid mesh. Conversely, the beam elements in the structural mesh are refined in the element size, also referred to as $h-$refinement. The reason for choosing different strategies for mesh refinement are due to limitations in the current structural solver, developed from the ground up. The ease of implementation of h refinement in the structural solver lead to its development. For the current growth factor, the error estimates of the fluid and structural subsystem are compared and the top 20% elements with the highest figure of merit, lying either in the fluid or structural mesh are chosen for mesh adaptation. Figure 7.9 shows the spatial order distribution post adaptation. The mesh adaptation targets elements in the fluid mesh undergoing mesh deforma-



(a) Output convergence (b) Error in the output of interest

Figure 7.7: Comparison of the time-averaged tip displacement for output-based adaptation to uniform-refinement.

tion along with those elements highlighted by the fluid adjoint, i.e above and below the beam. A comparison of the output convergence between the adapted meshes against uniformly refined meshes is shown in Figure 7.5. Figure 7.6 shows the nodal distribution post adaptation in the structural mesh. Due to the coarseness of the initial structural mesh, almost all of the structural elements are marked for adaptation in the first few cycles of mesh adaptation. At the later stages, the mesh adaptation targets elements near the trailing edge which are highlighted by the structural adjoint.

Each cycle of uniform mesh refinement increases the spatial order of all elements in the fluid mesh by one and doubles the number of elements in the structural mesh. The output evaluated for a fluid mesh with a uniform $p = 6$ spatial discretization and a structural mesh consisting of 128 uniform beam elements serves as the reference

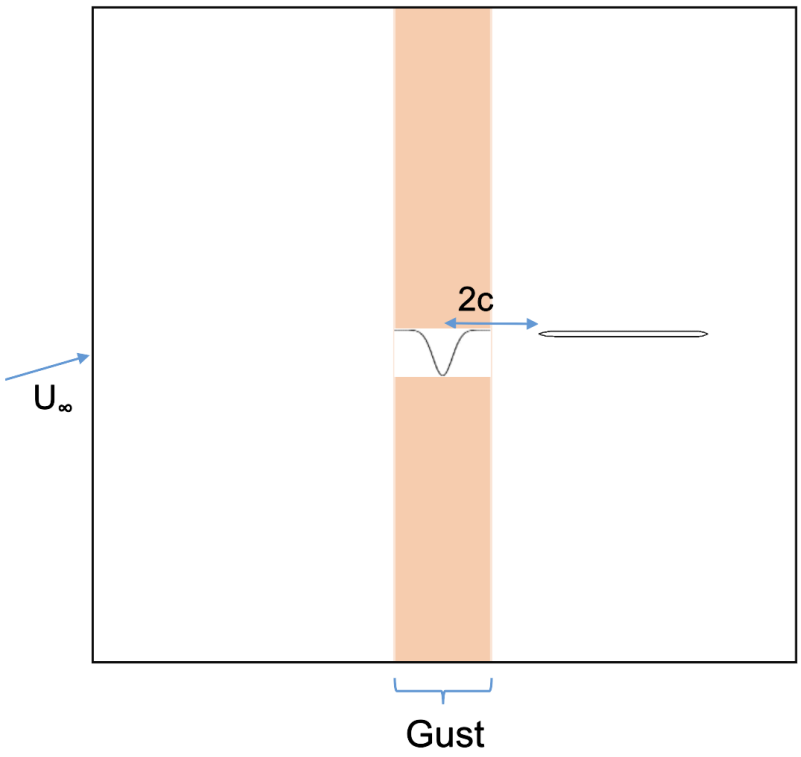
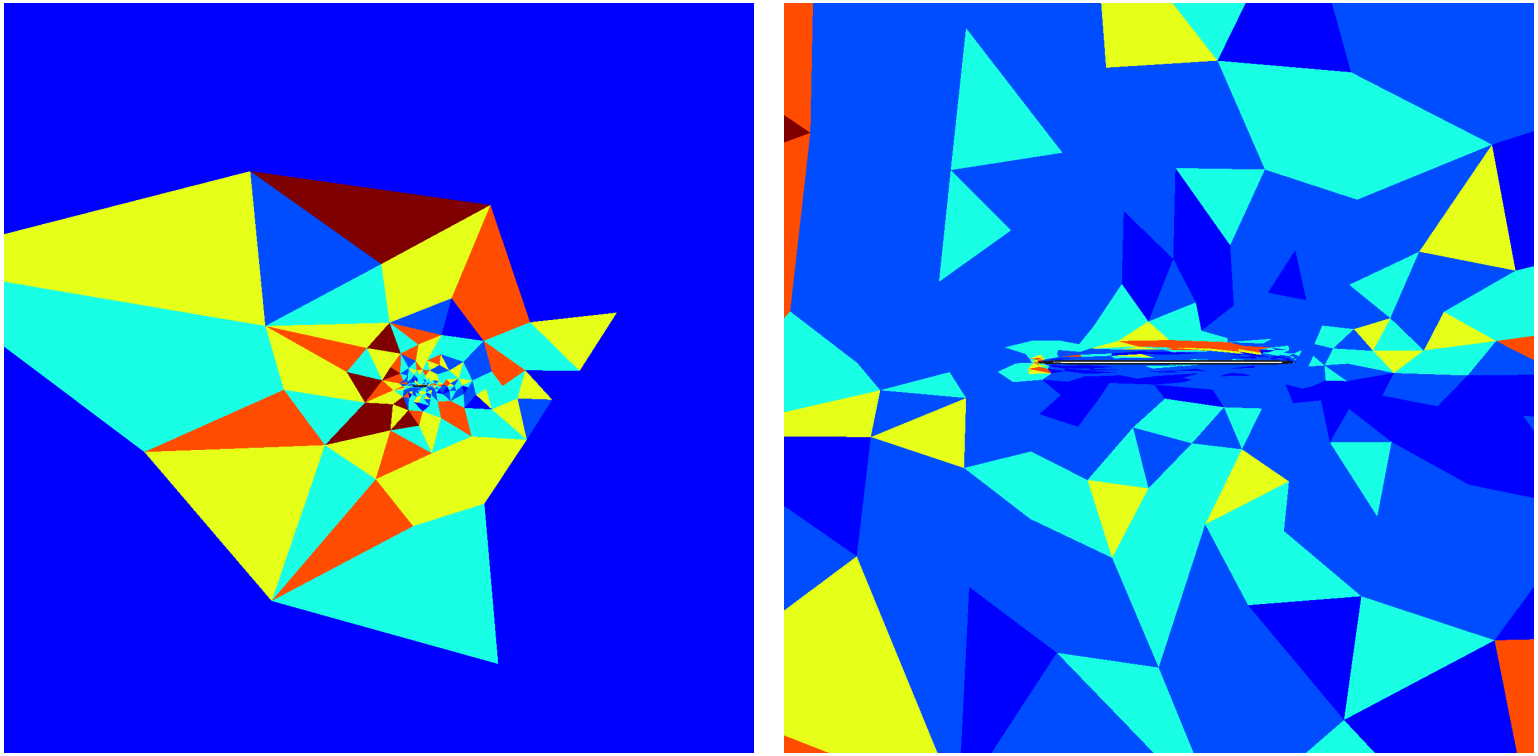


Figure 7.8: Case setup describing the initial flow conditions with the gust placed in front of the beam.



(a) (b)

Figure 7.9: Element order distribution at the end of the adaptation process for the cantilevered beam. The output chosen for mesh adaptation is the time averaged lift coefficient. Dark blue denotes $p = 1$, light blue denotes $p = 2$, green denotes $p = 3$, oranges denotes $p = 4$, and dark red denotes $p = 5$ order elements.

output and is used for evaluating the errors. A similar output convergence study is also conducted for a structural output, the time-integrated tip displacement, as shown in Figure 7.7. The advantages of adaptive meshing can be seen clearly for both outputs of interest where convergence is achieved with fewer degrees of freedom.

For the final output of interest, the uniform flow field is modified and a velocity gust is introduced in front of the structure at initial time, $t = 0$. Figure 7.8 shows the flow field at initial time $t = 0$. The velocity gust, the center of which is placed two chords in front of the cantilevered end of the beam, has a Gaussian nature and has a peak magnitude of $0.2U_\infty$. The velocity gust in front of the beam is given as,

$$V = \begin{cases} 0 & \text{for } r_d > r_{\max} \\ -0.2U_\infty \left[e^{-(r_d)^2/2} - e^{-r_{\max}^2/2} \right] & \text{for } r_d < r_{\max} \end{cases} \tag{7.6}$$

where r_d is the relative x position of any point from the center of the gust, $r_d = x + 2$, and $r_{\max}$ is the gust length given as, $r_{\max} = 0.5$. The simulation time is extended to

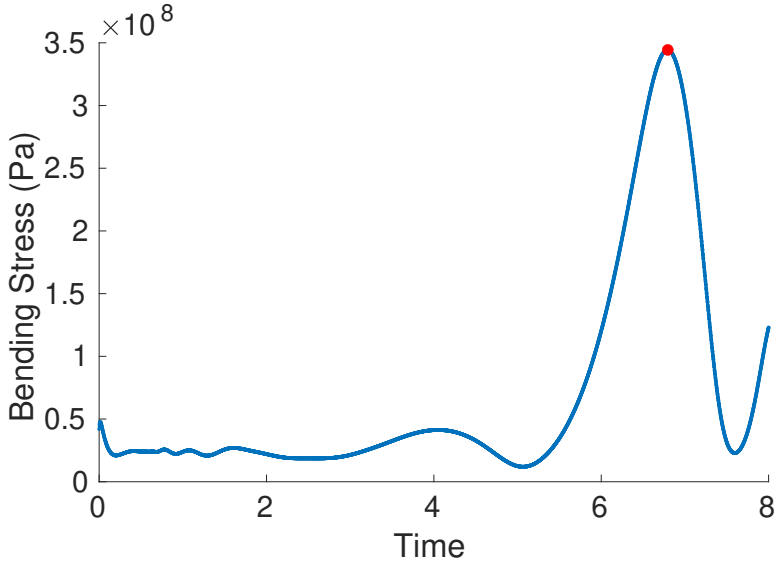
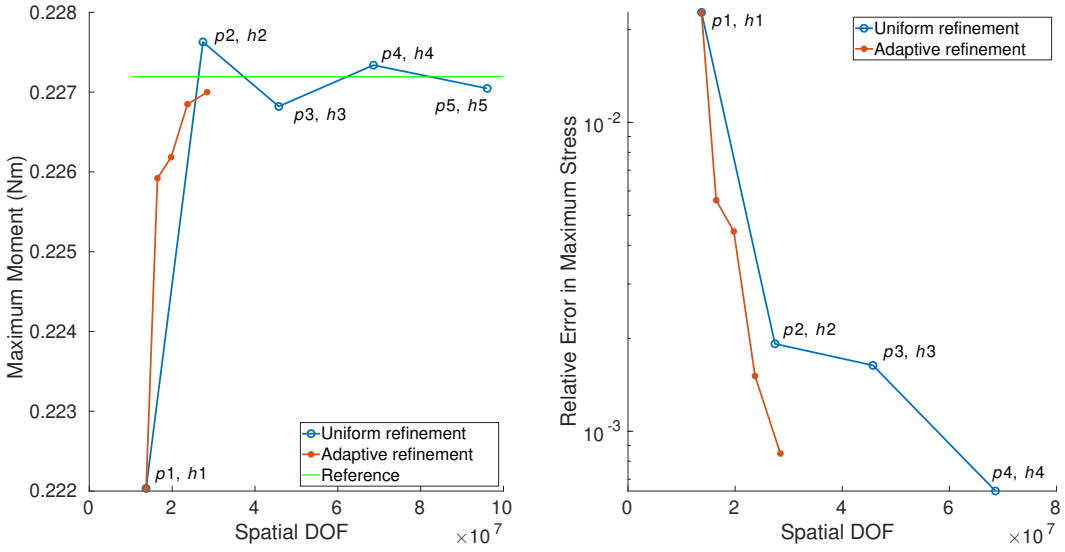


Figure 7.10: Variation of the unsteady bending stress about the cantilevered end of a cantilevered beam as a Gaussian gust flows over it. The stress is evaluated for a fluid mesh with uniform $p = 6$ order elements and a structural mesh with 128 uniform Euler-Bernoulli beam elements. The maximum moment, which is marked in red, occurs at $t = 6.36$ at the cantilevered end.

a final time of $t = 8$, in order for the gust to completely flow over the beam. The output of interest for the mesh adaptation is the maximum bending stress developed over the beam as the gust flows over it. The bending stress at a cross-section of the beam is gives as,

$$\sigma_{\text{beam}} = \frac{My}{I} \tag{7.7}$$

where M is the moment, y is the distance from the neutral axis and I is the moment of inertia. From the above definition, for a beam of uniform thickness, the maximum bending stress will always occur at the point experiencing the maximum moment. For this simulation, the maximum moment always occurs at the cantilevered end, however, the time at which the stress is maximum is not known apriori to the user. Figure 7.10 shows the unsteady bending stress about the cantilevered end of the beam. The Gaussian gust takes one second to reach the beam, after which it reduces the local angle of attack resulting in oscillations in the structure. From the figure, it can be seen that the maximum stress and thereby maximum moment is developed



(a) Output convergence (b) Error in the output of interest

Figure 7.11: Comparison of the maximum moment developed over the beam for output-based adaptation to uniform-refinement.

over the beam at the time, $t = 6.36$, at the cantilevered end. A discontinuous in time output, such as the maximum moment, can cause oscillations and convergence issues in the coupled adjoint solution. In order to circumvent this issue, the output can be redefined using a smooth Gaussian function which peaks at the time at which the maximum moment occurs such that,

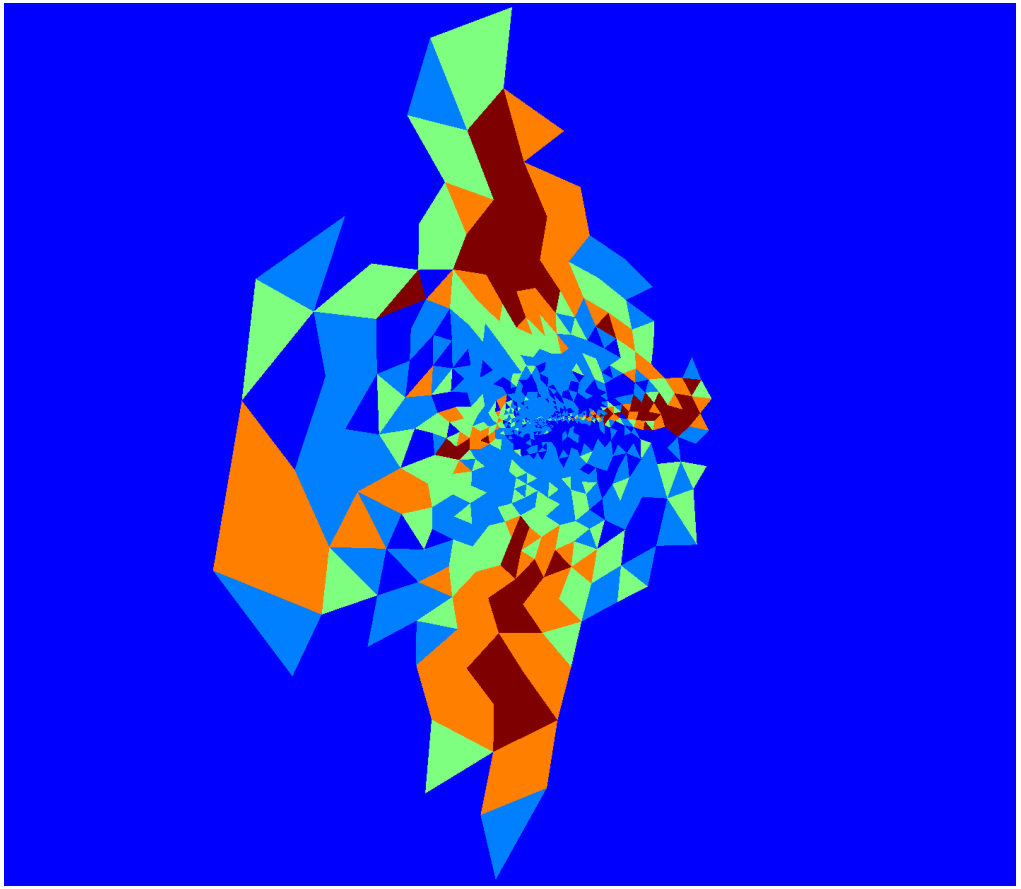
$$\bar{J} = \sigma_{\text{beam}}(\mathbf{U}^f, \mathbf{U}^s, t) \times \delta(t - t_{\max}) \approx \sigma_{\text{beam}}(\mathbf{U}^f, \mathbf{U}^s, t) \times e^{-50(t-t_{\max})^2}. \tag{7.8}$$

where, σ_{beam} is the moment about the cantilevered end of the beam, and $t_{\max}$ is the time at which the maximum moment occurs. For this output of interest, a similar output convergence study as above is conducted, as shown in Figure 7.11. Within the first few cycles of adjoint-based mesh adaptation, the meshes are able to converge to the value of the maximum stress and its location in the space-time discretization. The advantages of adaptive meshing can be seen clearly even for a complex output of interest such as the maximum moment. Figure 7.10 shows the fluid meshes post-mesh adaptation, for the stress output. Poor resolution of the gust in the larger elements away from the beam results in higher error estimates and subsequently more adaptation. Similar mesh adaptation is seen in the structural mesh for the maximum moment output as the time averaged lift output, studied previously, where a lot of adaptation occurs near the free end of the beam.

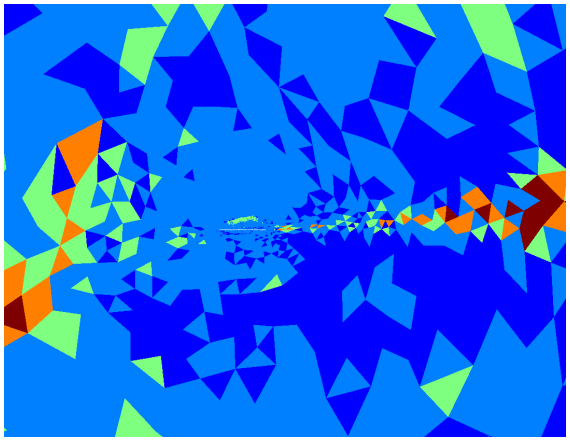
Table 7.2 summarizes the benefits of goal-oriented mesh adaptation for the two-

Aeroelastic simulation	**For a given accuracy on a coupled output**	
	Memory saving using adaptive mesh refinement	Time saving using adaptive mesh refinement
Aeroelastic Beam	66%	60%

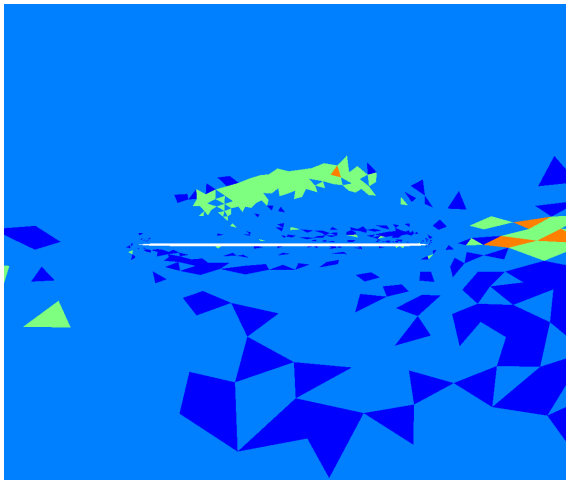
Table 7.2: Cost savings using goal-oriented mesh adaption in an aeroelastic beam subjected to subsonic flow.



(a)



(b)



(c)

Figure 7.12: Element order distribution at the end of the adaptation process for the cantilevered beam simulation. The output chosen for mesh adaptation is the maximum stress developed over the beam. Dark blue denotes $p = 1$, light blue denotes $p = 2$, green denotes $p = 3$, orange denotes $p = 4$, and dark red denotes $p = 5$ order elements.

dimensional aeroelastic case. The benefits showcased in the table are evaluated by comparing the degrees of freedom in the final adapted mesh against the reference mesh. Adapted meshes can provide similar or better level of accuracy in coupled outputs with almost one-third the number of degrees of freedom, when compared against uniformly refined meshes. Out of the total savings in the degrees of freedom, about 75% is observed in the fluid subsystem. This skewed behavior is primarily because of the higher higher number of degrees of freedom in the fluid subsystem and the simplicity of the structure. The table doesn't take into consideration the time involved in obtaining the adapted meshes by solving the adjoint systems. The adaptive meshing procedure increases the total computational time of the coupled simulations due to the additional adjoint evaluation by approximately a factor of two. However, the significant benefit of higher accuracy at fewer degrees of freedom outweighs the increase in the computational expense. Based on the existing literature about the application of goal-oriented mesh adaptation to three-dimensional CFD problems [23], higher memory savings are expected by the application of the coupled output-based mesh refinement to three-dimensional aeroelastic problems.

CHAPTER VIII

Aeroelastic Wing

This chapter applies goal-oriented mesh adaption to a three-dimensional aeroelastic wing simulated using a high-fidelity FSI solver. Modification to the evaluation of the off-diagonal derivatives in the coupled adjoint are firstly reviewed. The coupled adjoint solution along with the fluid and structural spatial error estimates, obtained using the coupled adjoint, are later verified. Using these error estimates, goal-oriented mesh adaptation is conducted on three coupled outputs of interest, of varying complexity. The benefits of coupled mesh adaptation is showcased for a subsonic and a transonic flow over the wing.

8.1 Case Setup

Consider a rectangular wing with an ONERA M6 root section [10] subjected to a uniform flow. The wing, which is cantilevered at the root, has a semi-span, $y = -2m$ and a uniform chord length, $c = 1m$. The wing acts as an enclosure of the elastic structure contained within, which is a rectangular plate of the same dimensions as the wing. The flexural rigidity of the plate D, defined as,

$$D = \frac{Eh^3}{12(1-\nu^2)}, \tag{8.1}$$

where E is the Young's modulus of the wing, h is the thickness an ν is the Poisson's ratio. The value of the flexural rigidity is set to, $D = 1000Nm^2$, such that considerable amount of the deformation in the plate is observed for the flow conditions it's subjected to. The elastic plate is modelled using four-noded quadrilateral elements, similar to the CQUAD4, isoparametric membrane-bending-shear quadrilaterals elements from NASTRAN [87].

The CQUAD4 element is based on the Discrete Kirchoff Theory which is an extension of Euler-Bernoulli beam theory and was developed by Love [75] using assumptions proposed by Kirchhoff. The theory assumes that a mid-surface plane can be used to represent a three-dimensional plate in two-dimensional form and that the planes perpendicular to the mid-surface will remain planar and perpendicular to the deformed mid-surface. Each node in the plate element has three degrees of freedom which are the out-of-plane displacement (z) and the two in-plane rotations (x and y). The planar wing is represented by a coarse initial structural mesh, consisting of 32 square elements of uniform size. The governing equations of motion of the plate are given as,

$$\bar{\mathbf{M}}\ddot{\mathbf{u}}^s + \mathbf{C}\dot{\mathbf{u}}^s + \mathbf{K}\mathbf{u}^s = \mathbf{F}^s, \tag{8.2}$$

where $\mathbf{u}^s$ is the structural state at a particular node, $\bar{\mathbf{M}}$ is the mass matrix, $\mathbf{K}$ is the stiffness matrix, $\mathbf{C}$ is the Rayleigh damping matrix and $\mathbf{F}^s$ is the force vector. Proportional damping in the form of Rayleigh damping with parameters, $\alpha = 1$ and $\beta = 0$ for the mass and stiffness matrices is added to model the damping in the structure.

Dynamic analysis of complex structures with large number of degrees of freedom are often not feasible due to their high computational cost. Thus, it is desirable to be able to transform a large degree of freedom problem to a more manageable,

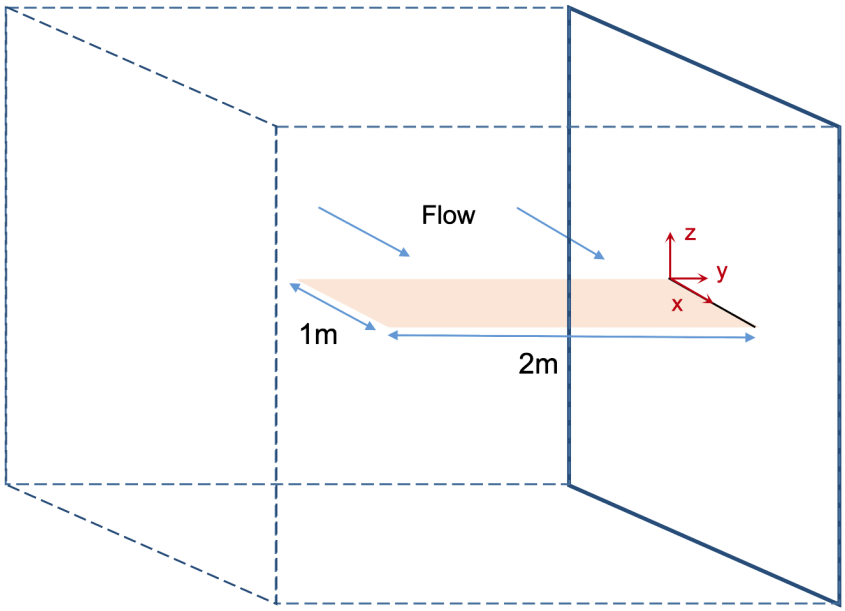


Figure 8.1: Case setup for the cantilevered wing.

smaller size for a dynamic analysis. Techniques which perform these transformations are called condensation techniques since they condense or reduce the size of the problem. The use of condensation techniques is fairly common for solving dynamics problems [90, 59, 9] and in this work, a static condensation is applied to the mass matrix. In static condensation, the rotational inertia of masses in plates is assumed to be very small in comparison with lateral inertia of the masses. Therefore, the dimensions of diagonal nonzero mass matrix will be reduced. The governing equation, Eq 8.2, can be rewritten in a partitioned manner as,

$$\begin{bmatrix} \bar{\mathbf{M}}_{aa} & \bar{\mathbf{M}}_{ab} \\ \bar{\mathbf{M}}_{ba} & \bar{\mathbf{M}}_{bb} \end{bmatrix} \ddot{\mathbf{U}}^s + \begin{bmatrix} \bar{\mathbf{C}}_{aa} & \bar{\mathbf{C}}_{ab} \\ \bar{\mathbf{C}}_{ba} & \bar{\mathbf{C}}_{bb} \end{bmatrix} \dot{\mathbf{U}}^s + \begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ab} \\ \mathbf{K}_{ba} & \mathbf{K}_{bb} \end{bmatrix} \mathbf{U}^s = \begin{bmatrix} \mathbf{F}_a^s \\ \mathbf{F}_b^s \end{bmatrix} \tag{8.3}$$

$$\text{where} \quad \mathbf{U}^s = \begin{bmatrix} \mathbf{u}_a^s \\ \mathbf{u}_b^s \end{bmatrix},$$

where, $\mathbf{u}_a^s$ represent the degrees of freedom associated with the inertial forces and $\mathbf{u}_b^s$ represent the degrees of freedom associated with the rotational forces. Using static

mass condensation, the inertia associated with the rotational degrees of freedom i.e $\bar{\mathbf{M}}_{bb}$, $\bar{\mathbf{M}}_{ab}$, and $\bar{\mathbf{M}}_{ba}$ are assumed to be zero. Assuming the damping to be linearly proportional to the mass matrix, the partitioned system can be simplified as,

$$\begin{bmatrix} \bar{\mathbf{M}}_{aa} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \ddot{\mathbf{U}}^s + \begin{bmatrix} \bar{\mathbf{C}}_{aa} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \dot{\mathbf{U}}^s + \begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ab} \\ \mathbf{K}_{ba} & \mathbf{K}_{bb} \end{bmatrix} \mathbf{U}^s = \begin{bmatrix} \mathbf{F}_a^s \\ \mathbf{F}_b^s \end{bmatrix} \quad . \tag{8.4}$$

The reduced system that needs to be solved dynamically is obtained by substituting the rotational degrees of freedom from the second equation into the first equation,

$$\begin{aligned} \bar{\mathbf{M}}_{aa}\ddot{\mathbf{u}}_a^s + \bar{\mathbf{C}}_{aa}\dot{\mathbf{u}}_a^s + \mathbf{K}_{aa}\mathbf{u}_a^s + \mathbf{K}_{ab}\mathbf{K}_{bb}^{-1}\left[\mathbf{F}_b^s - \mathbf{K}_{ba}\mathbf{u}_a^s\right] &= \mathbf{F}_a^s \\ \underbrace{\bar{\mathbf{M}}_{aa}}_{Mass}\ddot{\mathbf{u}}_a^s + \underbrace{\bar{\mathbf{C}}_{aa}}_{Damping}\dot{\mathbf{u}}_a^s + \underbrace{\left[\mathbf{K}_{aa} - \mathbf{K}_{ab}\mathbf{K}_{bb}^{-1}\mathbf{K}_{ba}\right]}_{Stiffness}\mathbf{u}_a^s &= \underbrace{\mathbf{F}_a^s - \mathbf{K}_{ab}\mathbf{K}_{bb}^{-1}\mathbf{F}_b^s}_{Force\ Vector} \end{aligned} \tag{8.5}$$

The governing equation for the condensed system, as obtained in Eq 8.5, is similar to the original governing equation, as given in Eq 8.2, but with a different definition of the mass matrix, damping matrix, stiffness matrix, and the force vector. At the undeformed, strain-free position, the wing makes an angle of attack of 5 degrees to the incoming flow. The initial coarse fluid mesh is represented using an unstructured hemispherical grid of 197011 nodes and 6864 hexahedral elements. Uniform $p = 1$ order polynomials are used to represent the fluid state in the coarse fluid mesh. The domain extends up to a radial distance of twenty chords from the cantilever end of the wing, which is placed at the symmetry plane of the grid. Free-stream boundary conditions are applied at the outer boundary of the mesh and symmetry boundary conditions are applied at the symmetry plane. The wing is subjected to a uniform fluid flow at two different flow conditions. A subsonic flow at a Mach number of $M_\infty = 0.25$ and Reynolds number of $Re = 1000$ and a transonic flow at a Mach number of $M_\infty = 0.7$ and Reynolds number of $Re = 100,000$ are used as the two different flow conditions. In the case of transonic flow over the wing, the artificial

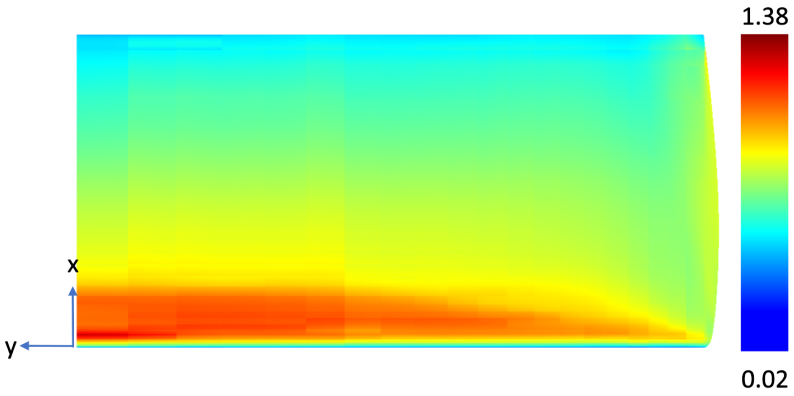
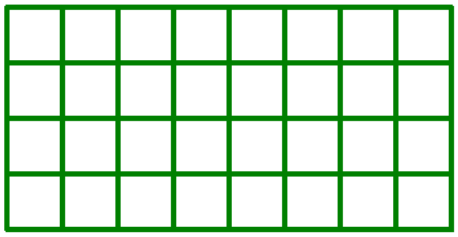
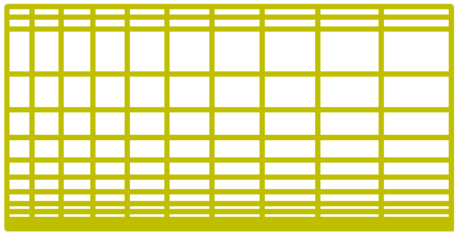


Figure 8.2: Mach number

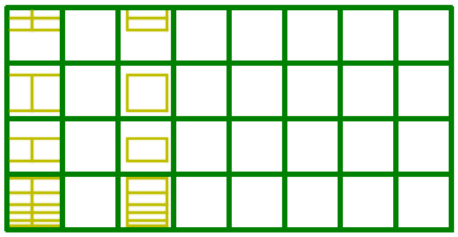
viscosity method of Persson and Peraire [99] is used for capturing shock and prevent solution oscillations at the shocks from hampering convergence. Figure 8.2 shows the Mach number variation over the surface of the wing subjected to transonic flow.



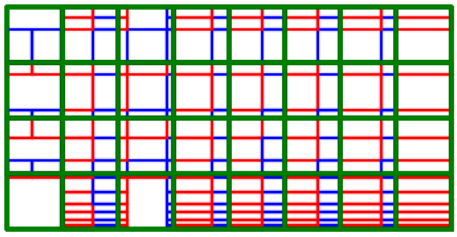
(a) Structural mesh at the interface



(b) Fluid mesh at the interface



(c) Fluid elements completely contained inside a structural element



(d) Fluid elements partially contained inside a structural element

Figure 8.3: Identification of fluid elements which overlap fully or partially over a structural element.

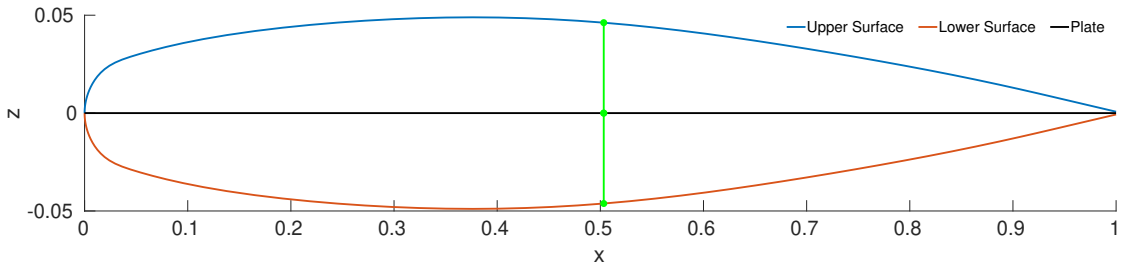


Figure 8.4: Projection of a node on the structural grid of the plate onto the upper and lower wetted surfaces of the wing cross section.

At the FSI interface of the wing, the boundaries of the elastic structure and wetted surface are non-matching and thus require special attention for transferring the forces and displacements between the two subsystems. As the thin plate is placed along the plane joining the leading edge and trailing edge, the data needs to be firstly projected from the wetted surface to the structure boundary and vice-versa. A normal projection technique is used to achieve this, where at any node on the elastic structure boundary, the point of intersection of the normal to the plate, originating from the node along $+z$ and $-z$ direction, and the wetted surface is used for projection of the data. Thus, the state at any node on the structural boundary is projected to a corresponding point on the upper and lower surface of the airfoil formed at the cut plane containing the node, as shown in Figure 8.4. Once the data is projected radial basis functions as described in Section 2.1.5.2, are used to transfer the displacement from the structural subsystem to the fluid subsystem for mesh motion. A compact support radius of one chord is used for displacement transfer using RBF's in the wing. In order to transfer forces between the fluid and structural subsystem, the pressure is evaluated at 40 two-dimensional quadrature points for a single fluid face at the interface. These data are then transferred to the structural solver, which uses a least squares method to fit a quintic polynomial spline to approximate the pressure distribution within each element. The benefits of using such a high number of quadrature points is the ability to approximate the pressure distribution accurately,

even in the case when the fluid states are approximated using high-order polynomials. In the structural elements that overlap, partially or fully, with the fluid elements, the contribution to the force vector is evaluated by integrating the pressure over the overlapped area with the shape function of the element. Figure 8.3 describes the identification of overlapping fluid elements for a structural grid.

8.2 Coupled-Adjoint Evaluation and Verification

The coupled adjoint is evaluated for the wing by solving for the adjoint equations as shown in Section 4.1.1. Similar to the two-dimensional aeroelastic case, the coupled adjoint equations are solved in a partitioned manner using the IMEX scheme. However, using finite differences for evaluating the off-diagonal derivatives for the aeroelastic wing becomes infeasible due to its high computational cost. The high cost of reevaluating the fluid residuals and the large number of structural nodes at the interface necessitate the development of an alternate and efficient strategy to evaluate the off-diagonal derivatives. Thus, a modal approximation of the structural displacement, with a fixed number of eigenmodes, is used for evaluating the off-diagonal derivatives. A detailed description of the procedure can be found in Appendix A.

A sensitivity perturbation test, as shown previously for the two-dimensional aeroelastic case, is used to verify the coupled adjoint for the three-dimensional case. The unsteady coupled adjoint is verified by comparing the sensitivities of the outputs of interest with respect to the initial shape of the plate against finite differences. Two outputs of interest from both the fluid and the structural subsystem are chosen: (i) the time-integrated lift and the trailing-edge displacement at the wingtip, and (ii) the final-time lift and the trailing edge displacement at the wingtip. The initial shape of

the wing at $t = 0$ is given by

$$u^s\Big|_{t=0} = A \sin\left(\frac{\pi y}{2}\right), \tag{8.6}$$

where the amplitude variable, A, is perturbed. The verification study for the coupled adjoint is conducted at a smooth laminar flow with a Reynolds number of $Re = 1000$ and a Mach number of $M = 0.25$. The unsteady simulation is simulated for a single time step with a final time $T = 0.0001s$. To evaluate the sensitivities from finite differences, a perturbation is given to the amplitude variable, A, and the four outputs of interest are then evaluated again. The differences between the outputs in the perturbed condition and the original condition are used to evaluate the sensitivities. For the chosen outputs of interest, the agreement is excellent for small perturbations, as shown in Figure 8.5, and it verifies the implementation of the wing coupled adjoint.

Having verified the coupled adjoint, error estimates based on the coupled adjoint, as shown in Section 4.1.2, are used to adapt the fluid and structural meshes for coupled outputs of interest. In this case, the spatial error estimates are used to adapt in the spatial domain of the aeroelastic problem while keeping the temporal discretization fixed. The spatial error estimate of a coupled output of interest for an individual subsystem is verified by finite differences. To evaluate the error estimate using finite differences, the output of interest is reevaluated by projecting the solution in a finer space, which serves as the reference output of interest. The finer space in the fluid subsystem is obtained by increasing the spatial order on an element by one. In the structural system, the finer space is obtained by subdividing every quadrilateral element uniformly into four equal quadrilateral elements. For the structural error estimate, the finer space is defined by keeping the fluid spatial discretization fixed while refining the structural spatial discretization and the temporal discretization. The dif-

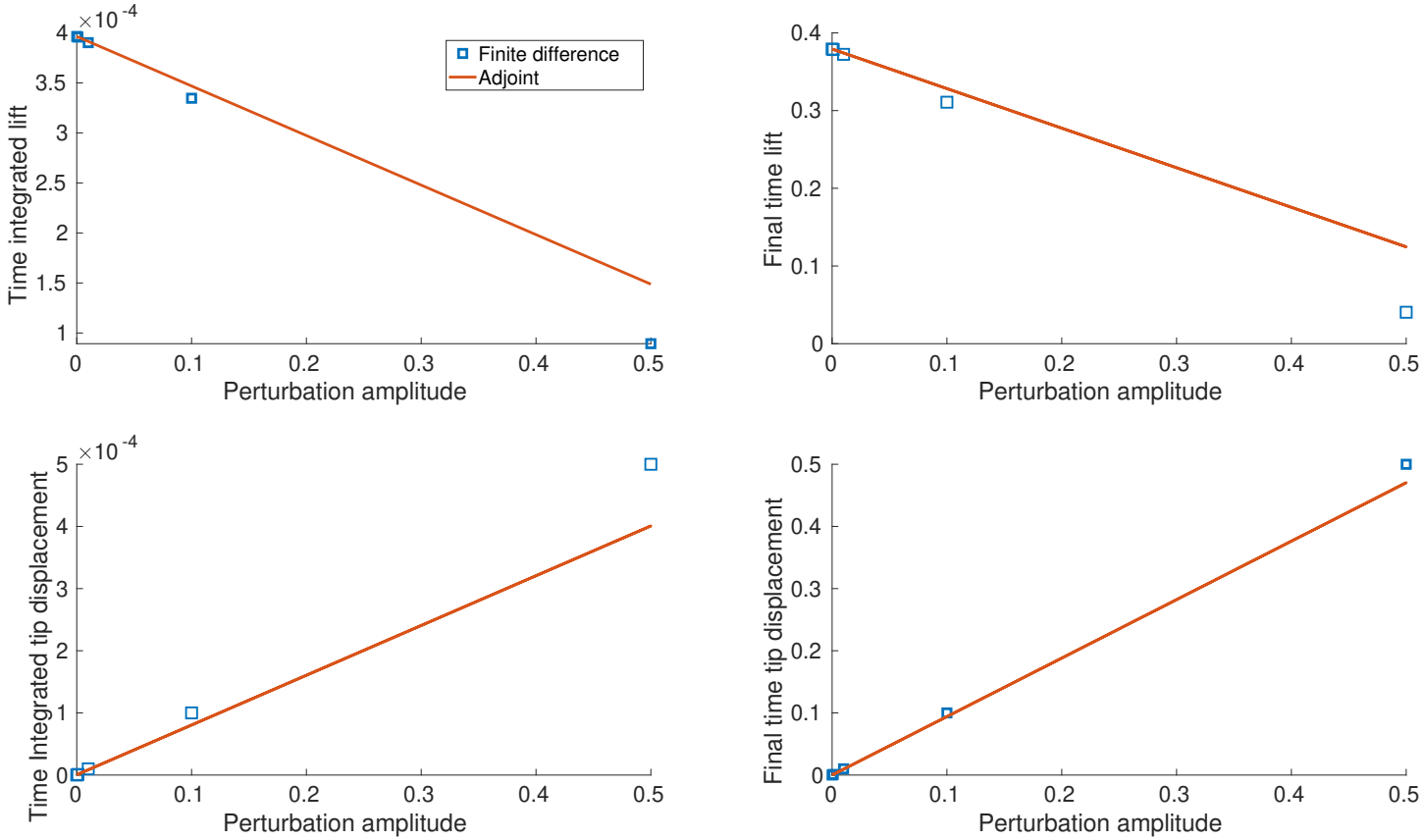


Figure 8.5: Verification of the coupled adjoint using a parameter sensitivity test.

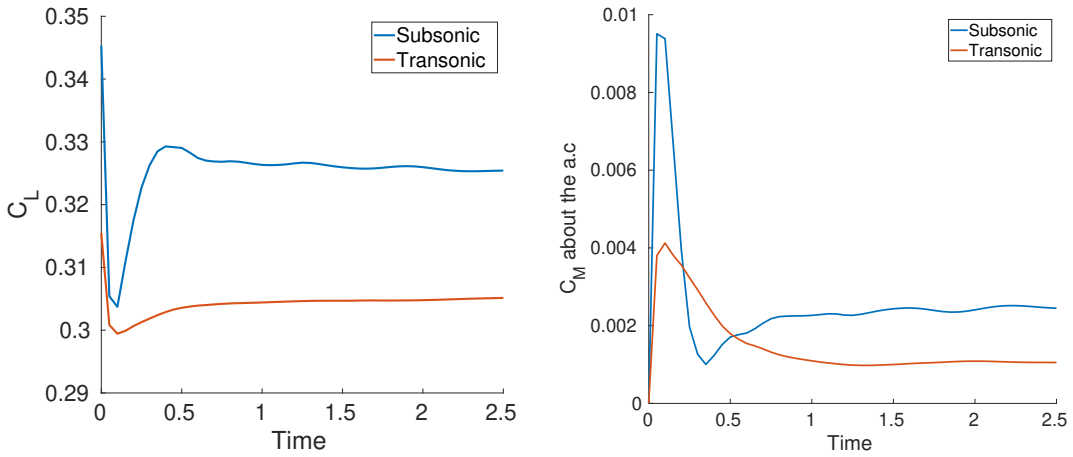
ference between the outputs in the coarse and fine space will consist of the errors due to the spatial discretization in the structural mesh and the temporal discretization. To isolate the errors from the spatial discretization, the temporal discretization is refined while keeping the structural spatial discretization fixed which provides the structural spatial error estimate, as shown in Table 7.1. The relative error is evaluated relative to a finer approximation in both space and time. The coarse space has uniform $p = 1$ elements in the fluid mesh and 32 square CQUAD4 plate elements, while the finer mesh has uniform $p = 2$ elements in the fluid mesh and 128 square CQUAD4 plate elements. The net spatial error estimate, as shown in column 6 of Table 8.1, obtained from the adjoint based method, is able to accurately predict the relative spatial error, as shown in column 3 of Table 8.1.

Total relative error	*Relative temporal error*	*Relative spatial error*	*Fluid spatial error estimate*	*Structure spatial error estimate*	*Total spatial error estimate*
9.76×10^{-08}	3.72×10^{-08}	1.42×10^{-07}	4.36×10^{-11}	1.57×10^{-07}	1.57×10^{-07}

Table 8.1: Comparison of the spatial error in the final time displacement at the trailing edge of the wing tip, relative to a finer approximation space and the error estimate from the adjoint-weighted residual.

8.3 Coupled Mesh Adaptation

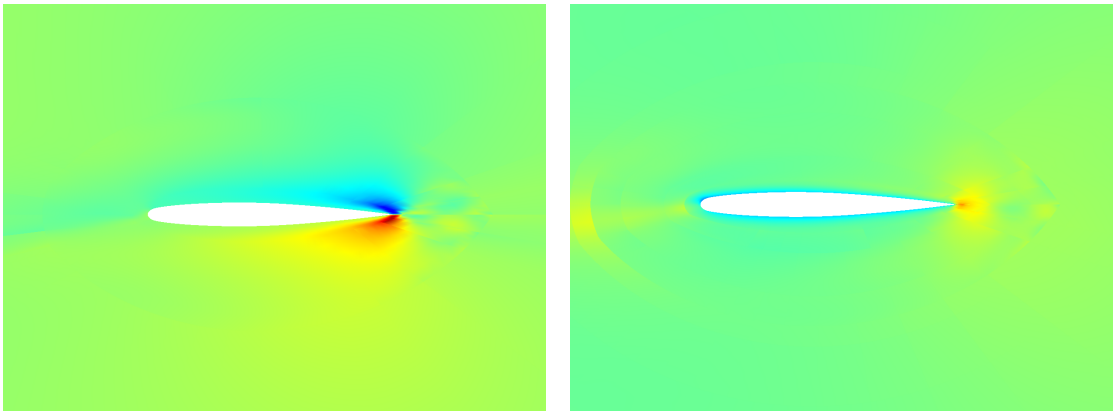
Three coupled outputs of interest are chosen for mesh adaptation for the three-dimensional aeroelastic case: the time-integrated lift coefficient, the time-integrated wingtip displacement, and the time-integrated moment developed over the plate for the subsonic and transonic flow cases. The moment is evaluated about the aerodynamic center of the wing (a.c) which lies at $x_{ac} = 0.245c$. The unsteady simulation is run up to a final time of $T_f = 2.5$ with fifty time-steps. In both flow conditions, the wing starts to bend up, in the z direction, due to the lift force generated at the initial



(a) Lift coefficient (b) Moment coefficient

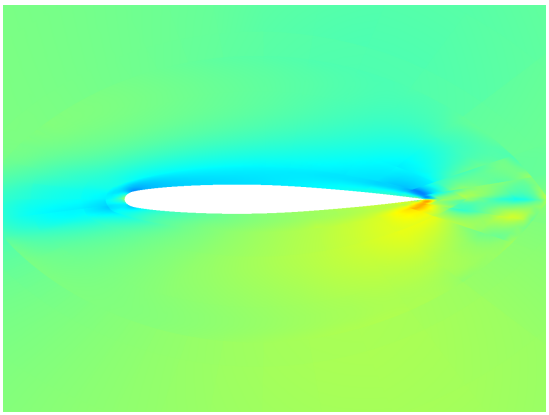
Figure 8.6: Unsteady lift coefficient for the subsonic and transonic flow.

angle of incidence. During the course of the simulation, maximum displacements of 10% and 20% of the chord are observed at the free end for the subsonic and transonic cases respectively. For the first output of interest, the time-averaged lift coefficient, Figure 8.6 shows the variation of the output for both flow conditions as the simulation progresses. The conservation of x-momentum and z-momentum components of the coupled fluid adjoint at the initial time is shown in Figure 8.7 and Figure 8.8

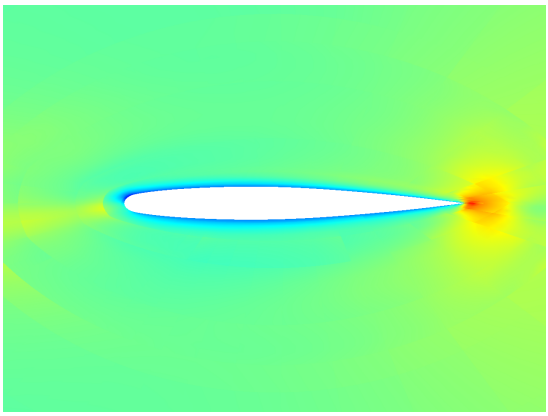


(a) Conservation of x-momentum adjoint (b) Conservation of z-momentum adjoint

Figure 8.7: Coupled-adjoint solution for the time-averaged lift coefficient output at the initial time. The wing is subject to a subsonic flow and the cut plane at which the adjoint solution is being shown is located at $y = -1$.



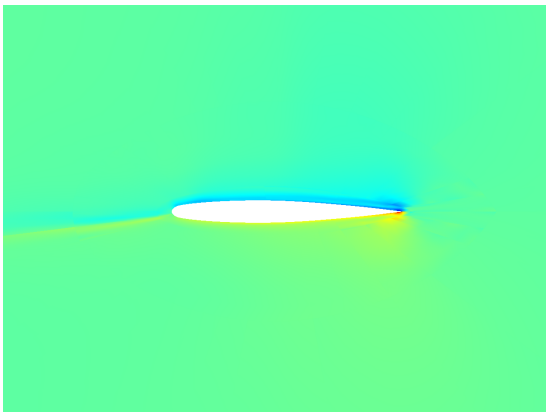
(a) Conservation of x-momentum adjoint



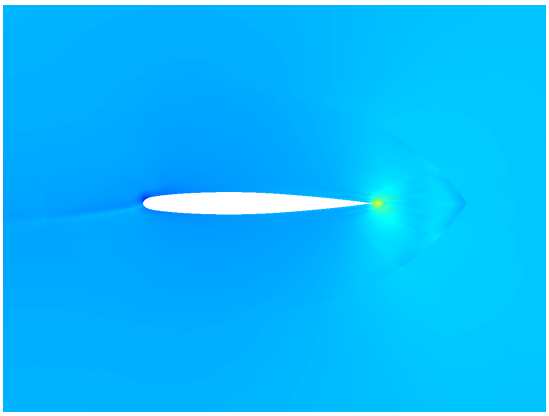
(b) Conservation of z-momentum adjoint

Figure 8.8: Coupled-adjoint solution for time-averaged lift coefficient output at the initial time. The wing is subject to a subsonic flow and the cut plane at which the adjoint solution is being shown is located at $y = -2$.

at two cut planes for the subsonic flow. The two cut planes are located at $y = -1$ and $y = -2$. The high sensitivity of the output to residual perturbations on the stagnation streamline and at the trailing edge of the wing is highlighted by the fluid adjoint. The trailing edge dictates the direction of flow coming off the wing which in turn dictates the total lift generated. Thus, the increased sensitivity of the output to

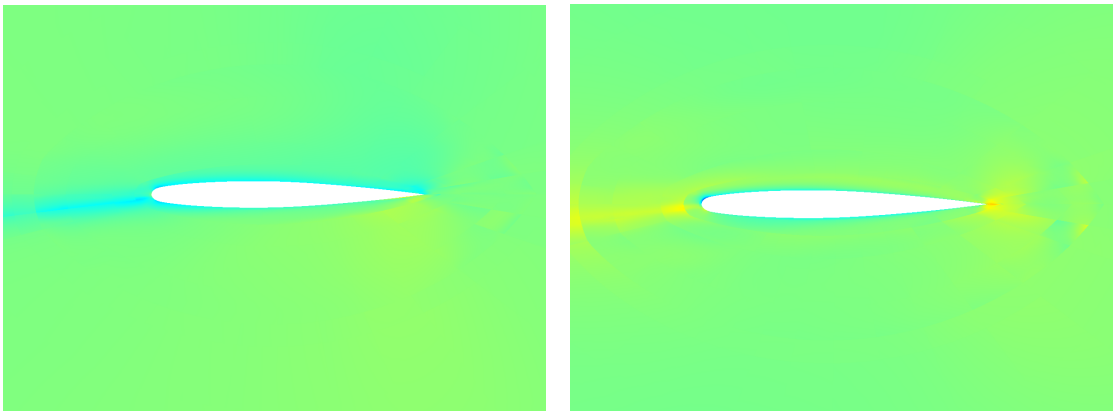


(a) Conservation of x-momentum adjoint



(b) Conservation of z-momentum adjoint

Figure 8.9: Coupled-adjoint solution for time-averaged lift coefficient output at the initial time. The wing is subject to a transonic flow and the cut plane at which the adjoint solution is being shown is located at $y = -1$.



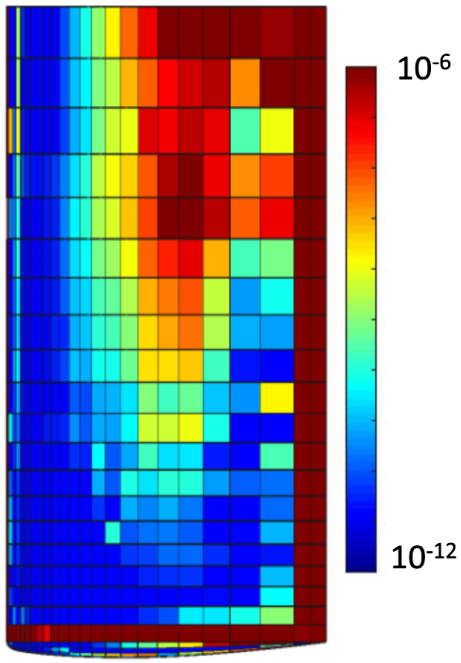
(a) Conservation of x-momentum adjoint

(b) Conservation of z-momentum adjoint

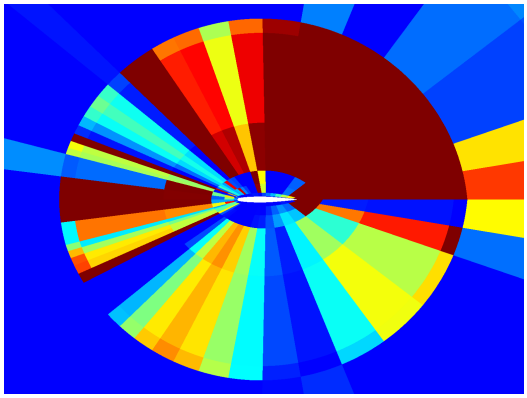
Figure 8.10: Coupled-adjoint solution for time-averaged lift coefficient output at the initial time. The wing is subject to a transonic flow and the cut plane at which the adjoint solution is being shown is located at $y = -2$.

the trailing edge is expected and captured well by the coupled adjoint. A similar fluid adjoint field is observed for the transonic flow, as shown in Figure 8.9 and Figure 8.10.

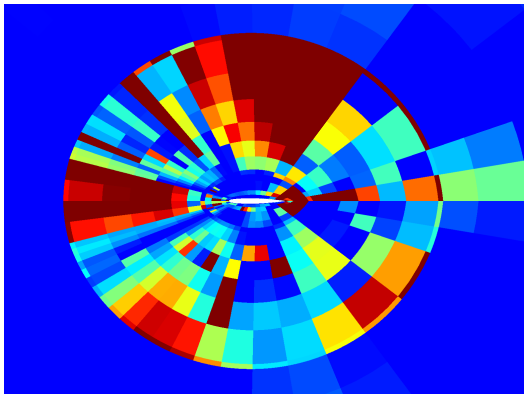
Error estimates using the coupled adjoint guide the mesh adaptation process. Figure 8.11 and 8.12 show the error estimates in the coarsest fluid mesh for the subsonic and transonic flow conditions. Due to high sensitivity of the output to the flow at the trailing edge, as observed from the fluid adjoint at the FSI interface, the elements located near the trailing edge have high spatial error estimates. The element distribution in the initial fluid mesh is such that the element sizes increase radially away from the wing. Thus, due to the unsteady projected residual component of the error estimate being larger in these bigger elements, higher error estimates are also observed in hexahedral elements away form the wing. The flow along the stagnation streamline, highlighted by the adjoint, also gets picked up for mesh adaptation. In both the subsonic and transonic flow over the wing, the lift generated near the free end of the wing affects the overall shape of the structure. Thus, higher sensitivity and subsequent higher error estimates are observed near the free end of the wing, in both



(a) Fluid spatial error estimate on the wing interface



(b) Cut plane located at $y = -1$

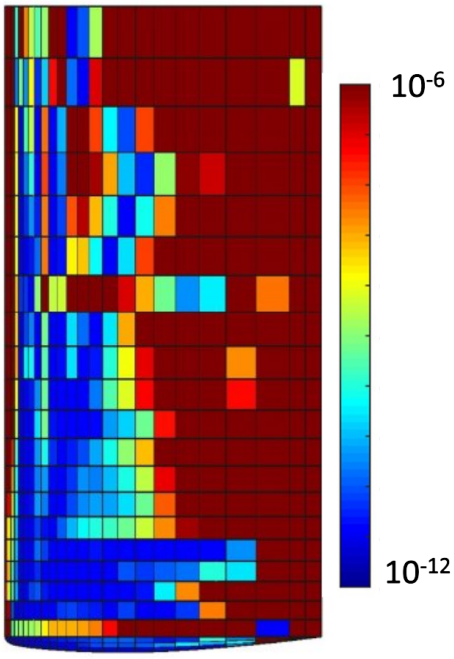


(c) Cut plane located at $y = -2$

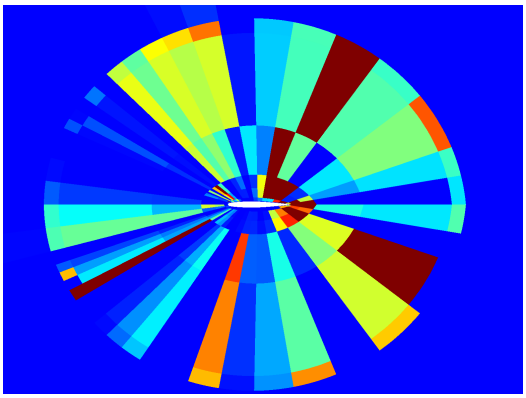
Figure 8.11: Fluid spatial error estimate for the time-integrated lift coefficient output on a wing in a subsonic flow.

flows. Additionally, in the case of transonic flow, the coupled output is also sensitive to the shock position and thus higher error estimates are observed near the leading edge of the wing where the shock is located. Figure 8.13 shows the corresponding contours of structural error estimates on the coarse structural grid. The structural spatial error estimates showcase similar behavior as the fluid spatial error estimates at the interface. Higher error estimates are observed near the free end for the subsonic flow while the presence of shock in the transonic flow, leads to higher errors being generated closer to the fixed end of the wing.

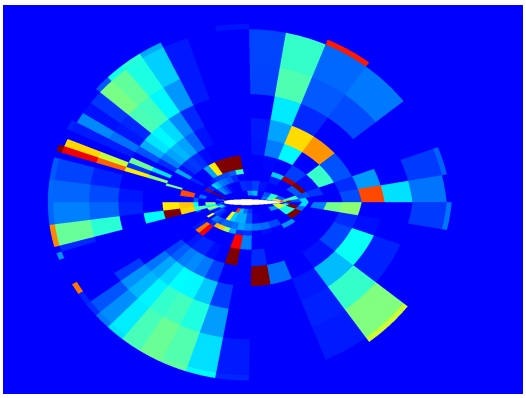
Using the error estimates, the initial fluid and structural mesh are subjected to three



(a) Fluid spatial error estimate on the wing interface



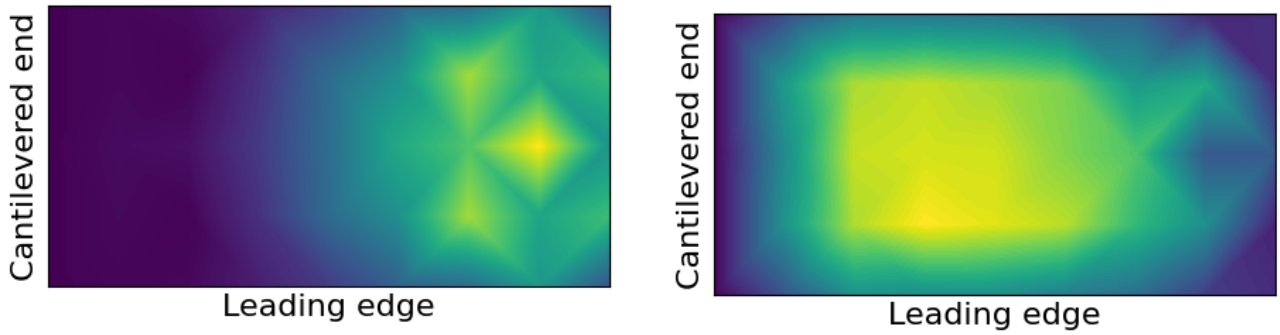
(b) Cut plane located at $y = -1$



(c) Cut plane located at $y = -2$

Figure 8.12: Fluid spatial error estimate for the time integrated lift coefficient output on a wing in a transonic flow.

cycles of mesh adaptation, with a global growth factor of $f_{tot} = 1.2$. The mesh adaptation occurs in the spatial order of mesh elements in the fluid mesh. Conversely, the plate elements in the structural mesh are refined in the element size. For the current growth factor, the error estimates of the fluid and structural subsystem are compared and the top 20% elements with the highest figure of merit, lying either in the fluid or structural mesh are chosen for mesh adaptation. Refining the structural mesh locally using h-refinement introduces hanging nodes in the structural grid. A hanging node is a node, on an edge of an element that does not belong to that element. Special consideration is required for defining the state at a hanging node in order to maintain continuity across elements. In this work, the state at a hanging node



(a) Structural spatial error estimate on the wing in a uniform subsonic flow.

(b) Structural spatial error estimate on the wing in a uniform transonic flow.

Figure 8.13: Structural spatial error estimate contours for the time integrated lift coefficient output.

is represented uniquely based on the states of the elements, with which it shares an edge. Thus, hanging nodes are treated as spurious degrees of freedom, which are not solved for. However, the hanging node approximation described above can cause a refined element surrounded by non-refined elements to not show any benefits of mesh refinement. This is because four out of the five nodes introduced during refinement of such an element are spurious. The state at the spurious nodes are constrained such that the finite element solution becomes continuous. Thus, for every element in the structural grid chosen for refinement, the neighbours of the chosen element are also refined, as shown in Figure 8.14.

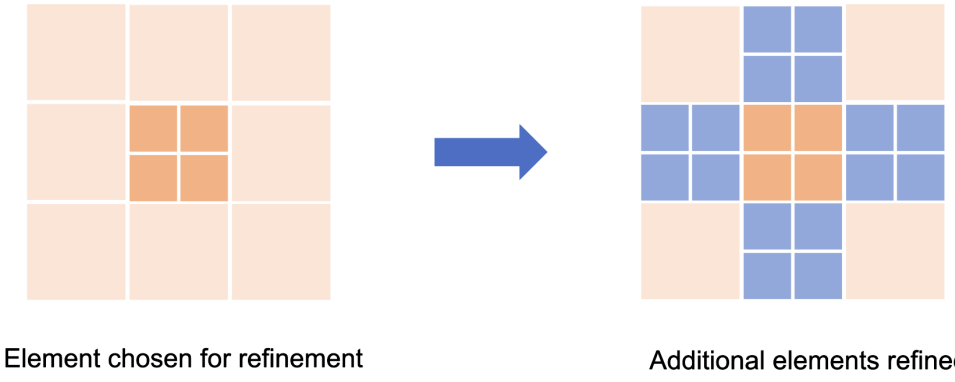
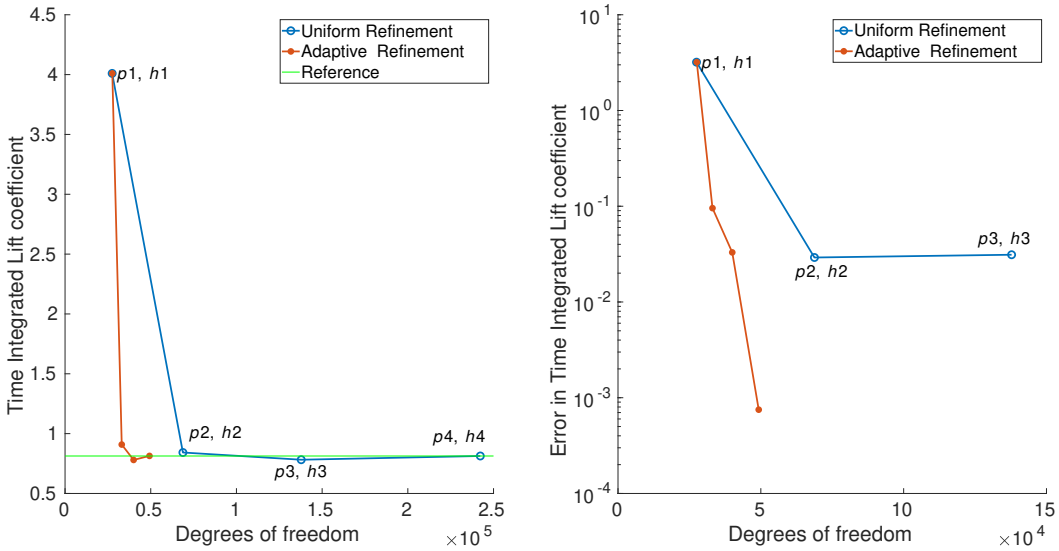
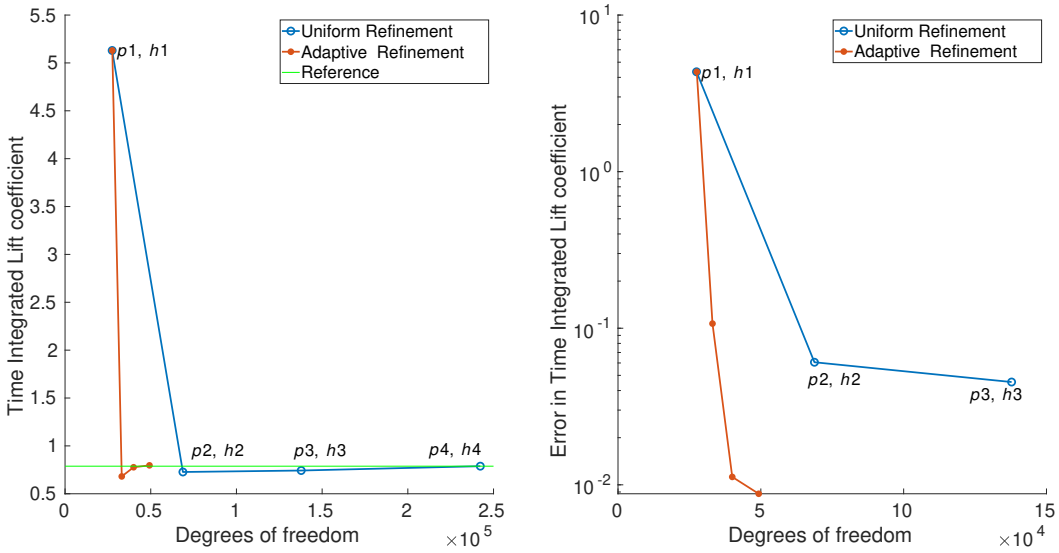


Figure 8.14: Strategy used for reducing spurious degrees of freedom when an element is chosen for mesh refinement.



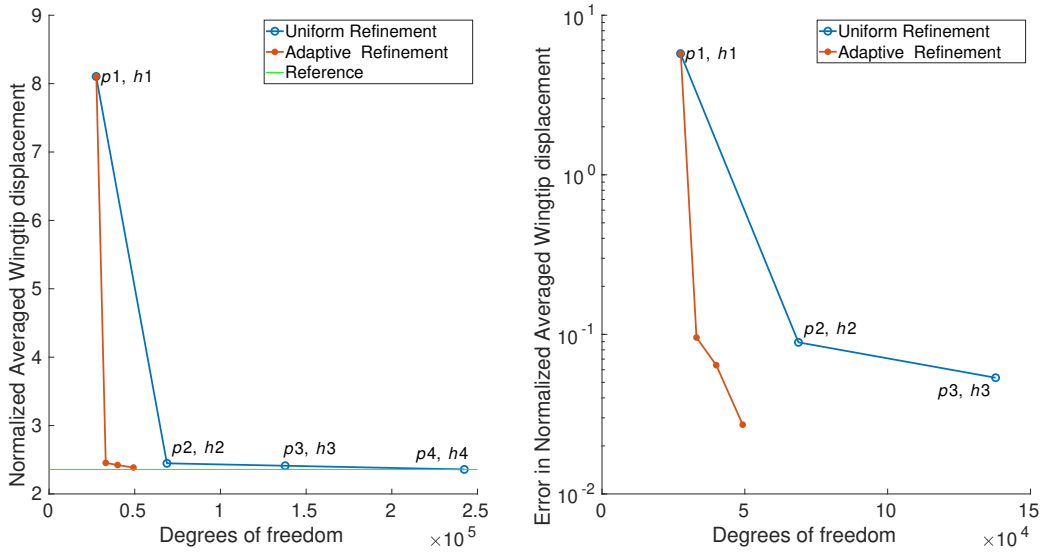
(a) Output convergence (b) Error in the output of interest

Figure 8.15: Convergence of the time-integrated lift coefficient, for an aeroelastic wing in subsonic flow, using output-based mesh refinement and uniform mesh refinement.



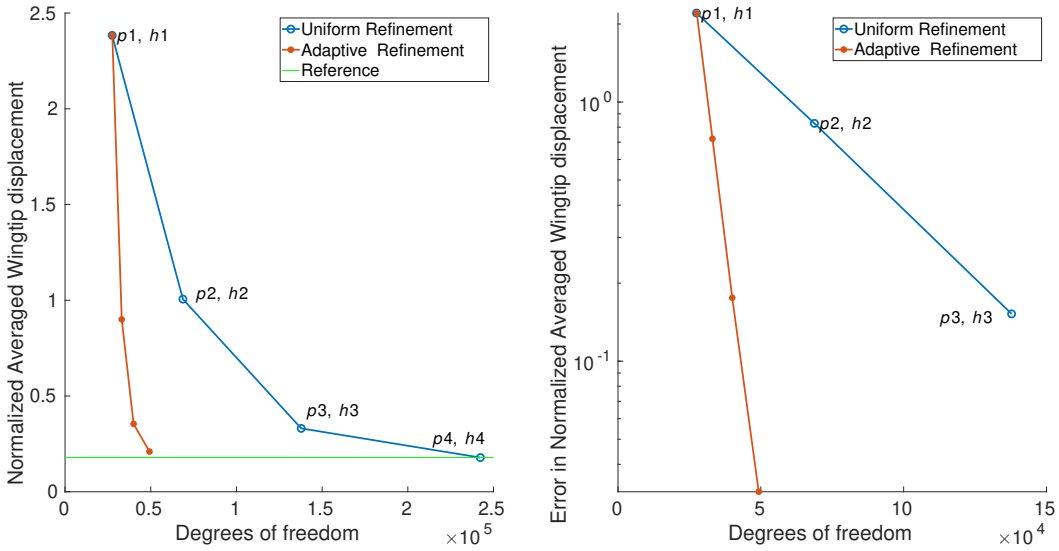
(a) Output convergence (b) Error in the output of interest

Figure 8.16: Convergence of the time-integrated lift coefficient, for an aeroelastic wing in transonic flow, using output-based mesh refinement and uniform mesh refinement.



(a) Output convergence (b) Error in the output of interest

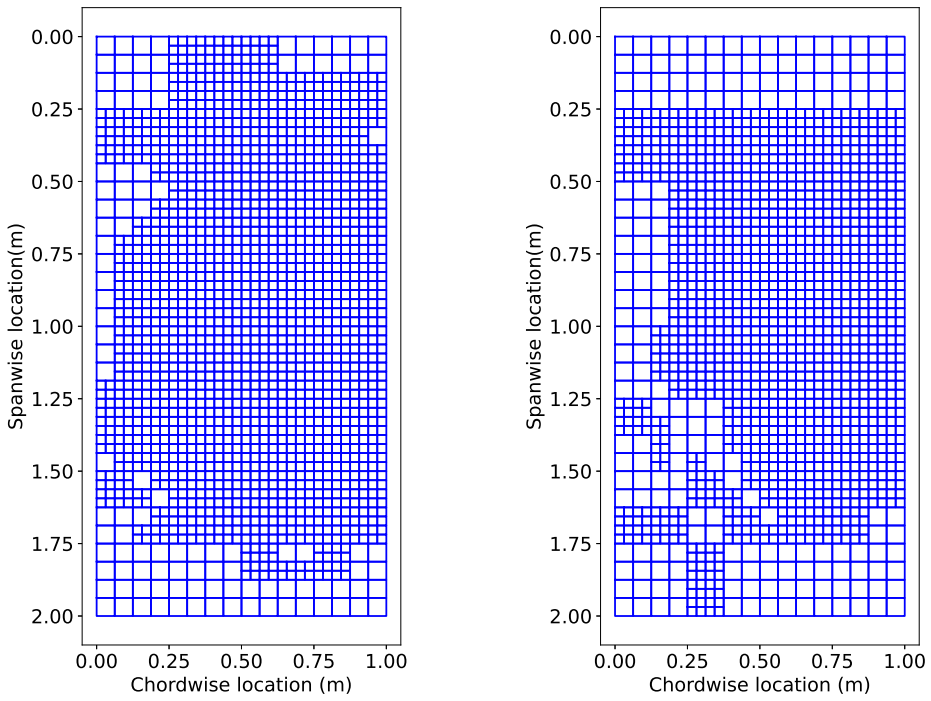
Figure 8.17: Convergence of the normalized time-averaged wingtip displacement, for an aeroelastic wing in subsonic flow, using output-based mesh refinement and uniform mesh refinement.



(a) Output convergence (b) Error in the output of interest

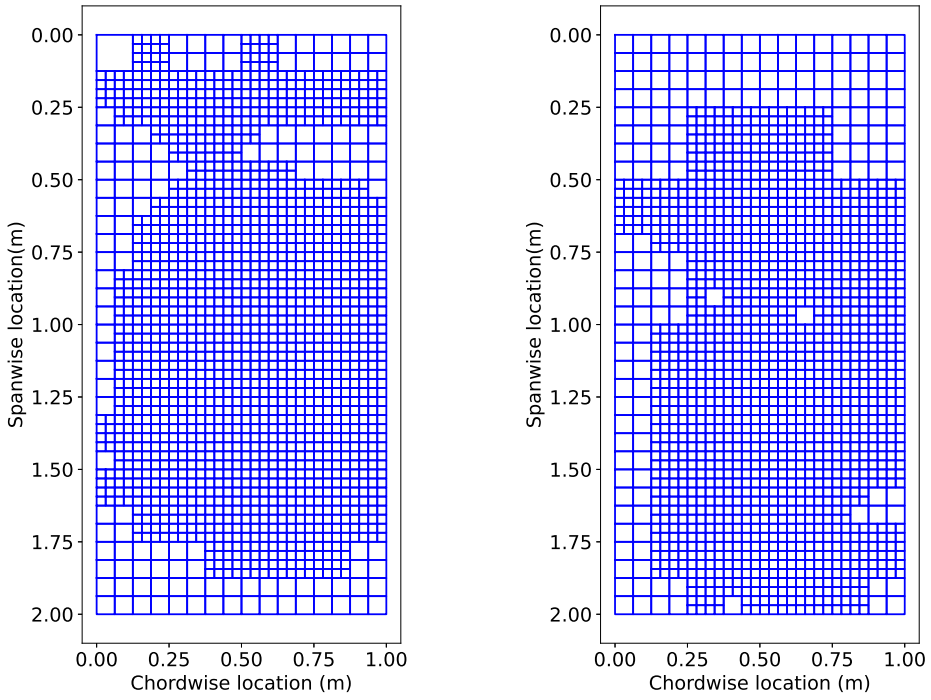
Figure 8.18: Convergence of the normalized time-averaged wingtip displacement, for an aeroelastic wing in transonic flow, using output-based mesh refinement and uniform mesh refinement.

A comparison of the output convergence between the adapted meshes against uniformly refined meshes is shown in Figure 8.15 and Figure 8.16. From the convergence plot, the two main benefits of goal-oriented mesh adaptation can be deduced, which are: 1) greater accuracy for a given number of degrees of freedom and 2) lower degrees of freedom required to reach a given level of accuracy when compared against uniform mesh refinement. The benefits of adaptive meshing in three-dimensional aeroelastic cases exceed the benefits observed in two-dimensional aeroelastic cases. The convergence plots show similar convergence behavior for both the subsonic and transonic flows, thereby showing the benefit of their applications in the presence of shock flow features. A similar behavior can be observed for other outputs such as



(a) Subsonic (b) Transonic

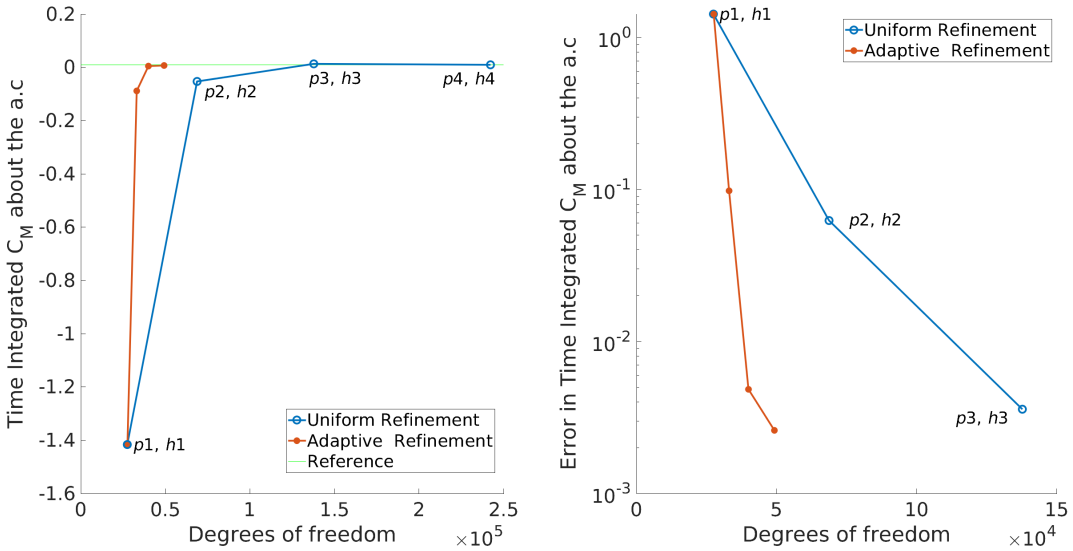
Figure 8.19: Adapted structural meshes for the time-integrated lift coefficient output.



(a) Subsonic (b) Transonic

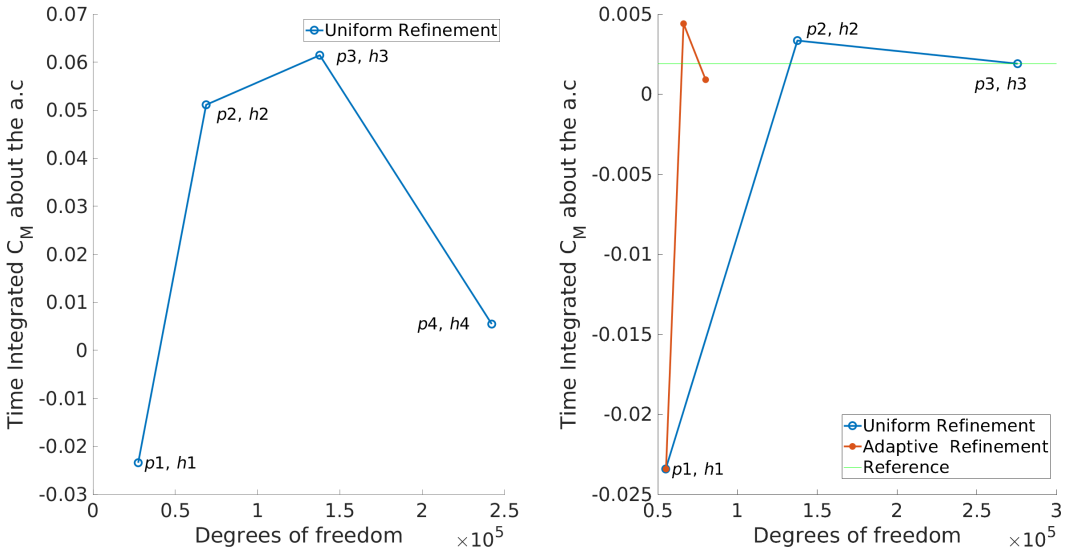
Figure 8.20: Post adapted structural meshes for the time integrated wingtip displacement output.

time-integrated wingtip displacement and time-integrated moment, as shown in Figure 8.17, 8.18, and 8.21. The lack of convergence of time-integrated moment over the wing in a transonic flow, as shown in Figure 8.22a show the importance of the initial mesh in output convergence. Due to the use of the same coarse initial fluid mesh for both flow conditions, one that lacks the spatial resolution to resolve the shock feature, and the higher sensitivity of the moment output to the shock position, convergence in the moment output is not observed in transonic flow. The output has yet to reach the asymptotic regime even with uniform mesh refinement and a better initial fluid mesh with more resolution on the upper surface is needed. Output convergence can be observed in the time-integrated moment over the wing in a transonic



(a) Output convergence (b) Error in the output of interest

Figure 8.21: Convergence of the time-integrated moment coefficient, for an aeroelastic wing in subsonic flow, using output-based mesh refinement and uniform mesh refinement.



(a) Coarse initial fluid mesh (b) Fine initial fluid mesh

Figure 8.22: Output convergence of the time-integrated moment coefficient, for an aeroelastic wing in transonic flow, using output-based mesh refinement and uniform mesh refinement for two initial fluid meshes .

flow when a finer initial fluid mesh with twice the number of elements is used as the initial mesh, as show in Figure 8.22b. In the coarse initial fluid mesh, p-refinement takes too many refinement levels or too high of a spatial order to start converging but a combined h-refinement and p-refinement helps in obtaining an accurate output of interest. Figures 8.19 and 8.20 show the nodal distribution after adaptation in the structural meshes. Due to the coarseness of the initial structural mesh, almost all of the structural elements are marked for adaptation in the first few cycles of mesh adaptation. At the later stages, the mesh adaptation targets elements near the trailing edge as well as the elements near the cantilevered end.

Table 8.2 summarizes the benefits of goal-oriented mesh adaptation for the three-dimensional aeroelastic case. The benefits shown in the table are evaluated by comparing the degrees of freedom in the final adapted mesh against the reference mesh. Adapted meshes can provide similar or better level of accuracy in coupled outputs with almost a quarter of the number of degrees of freedom, when compared to uniformly refined meshes. The lower number of degrees of freedom in the adapted meshes also lead to faster simulations. The adaptive meshing procedure increases the total computational time of the coupled simulations due to the additional adjoint evaluation by approximately a factor of two and a half, which is not reflected in Table 8.2, and is a one time cost of obtaining the adapted meshes. This factor is higher for the three-dimensional case due to the high computational cost incurred during the off-diagonal term evaluation for the coupled adjoint.

Aeroelastic simulation	**For a given accuracy on a coupled output**	
	Memory saving using adaptive mesh refinement	Time saving using adaptive mesh refinement
Aeroelastic Wing	79.5%	75%

Table 8.2: Cost savings using goal-oriented mesh adaption in an aeroelastic wing in a subsonic and transonic flow.

CHAPTER IX

Neural Network

In this chapter, the idea of using a neural network to predict the optimal computational fluid mesh, including both the element sizing and stretching is investigated. This work follows the work of Fidkowski and Chen [44], while taking the element sizing also as a network output. Furthermore, in order to enable fast mesh generation, *a priori* information, such as the geometry and the boundary conditions, are taken as the network inputs instead of the primal and adjoint solutions. The network is trained with optimal anisotropic meshes generated using an adjoint-based method that incorporates the mesh anisotropy through an expensive sampling procedure [132, 41]. The goal is to predict the optimal mesh for a given cost only using the *a priori* information. The network is trained on subsonic and transonic aerodynamic simulations over airfoils and is tested on unseen flow conditions.

9.1 Mesh Adaptation

The localized output-based error estimate developed in the Section 4.1.2 provides information about how to adapt the computational mesh. However, as just one scalar quantity per element, the error indicator is not sufficient to provide information about both the mesh sizing and mesh anisotropy. This information comes from a mesh optimization procedure, MOESS [132, 41], which solves an optimization problem to

find a mesh producing the lowest error for a given computational cost. In order to solve this optimization problem, a model for the dependence of the error on the metric is required. The approach iteratively determines the optimal mesh metric field given a prescribed metric-cost relationship and a sampling-inferred metric-error relationship.

9.1.1 Metric-Based Meshing

A Riemannian metric field, $\mathcal{M}(\vec{x})$, is a field of symmetric positive definite (SPD) tensors that can be used to encode information about the desired size and stretching of a computational mesh. At each point in physical space, $\vec{x}$, the metric tensor $\mathcal{M}(\vec{x})$ provides a "yardstick" for measuring the distance from $\vec{x}$ to another point infinitesimally far away, $\vec{x} + \delta\vec{x}$. After choosing a Cartesian coordinate system and basis for physical space, $\mathcal{M}$ can be represented as a $d \times d$ symmetric positive definite matrix. The set of points at unit metric distance from $\vec{x}$ is an ellipse (in two-dimensions): eigenvectors of $\mathcal{M}$ give directions along the principal axes, while the length of each axis (stretching) is the inverse square root of the corresponding eigenvalue. The aspect ratio is the ratio of the largest stretching magnitude to the smallest.

A mesh that conforms to a metric field is one in which each edge has the same length, to some tolerance, when measured with the metric. An example of a two-dimensional metric-conforming mesher is the Bi-dimensional Anisotropic Mesh Generator (BAMG) [61], and this is used to obtain the results in the present work. BAMG generates a mesh given a metric field, which is specified at nodes of a background mesh – the current mesh in an adaptive setting. The optimization determines changes to the current, mesh-implied, metric, $\mathcal{M}_0(\vec{x})$. Affine-invariant [97] changes to the metric field are made via a symmetric step matrix, $\mathcal{S} \in \mathbb{R}^{d\times d}$, according to

$$\mathcal{M} = \mathcal{M}_0^{\frac{1}{2}} \exp(\mathcal{S}) \mathcal{M}_0^{\frac{1}{2}}. \tag{9.1}$$

Note that $\mathcal{S} = 0$ leaves the metric unchanged, while diagonal values in $\mathcal{S}$ of $\pm$ 2 log2 halve/double the metric stretching sizes.

9.1.2 Error Convergence Model

The mesh optimization algorithm requires a model for how the error changes as the metric changes. Consider a single element, Ω_e, with a current error $\mathcal{E}_{e0}$ and a proposed metric step matrix of $\mathcal{S}_e$. The error on Ω_e following refinement with this step matrix is given by

$$\mathcal{E}_e = \mathcal{E}_{e0} \exp[\mathrm{tr}(\mathcal{R}_e \mathcal{S}_e)], \tag{9.2}$$

where $\mathcal{R}_e$ is a symmetric rate tensor. The total error over the mesh is the sum of the elemental errors, $\mathcal{E} = \Sigma_{e=1}^{N_e} \mathcal{E}_e$. The rate tensor, $\mathcal{R}_e$, is determined separately for each element through a sampling procedure [41].

9.1.3 Cost Model

Degrees of freedom, DOF, are used to measure the cost of refinement, which on each element just depends on the approximation order p, assumed constant over the elements. By (9.1) and properties of the metric tensor, when the step matrix $\mathcal{S}_e$ is applied to the metric of element e, the area of the element decreases by $\exp[\frac{1}{2}\mathrm{tr}(\mathcal{S}_e)]$. Equivalently, the number of new elements, and hence degrees of freedom, occupying the original area Ω_e increases by this factor. So the elemental cost model is

$$C_e = C_{e0} \exp\left[\frac{1}{2}\mathrm{tr}(\mathcal{S}_e)\right], \tag{9.3}$$

where C_{e0}=DOF$_{e_0}$ is the current number of degrees of freedom on element e. The total cost over the mesh is the sum of the elemental costs, $C = \Sigma_{e=1}^{N_e} C_e$.

9.1.4 Metric Optimization Algorithm

Given a current mesh with its mesh-implied metric, $\mathcal{M}_0(\vec{x})$, elemental error indicators, $\mathcal{E}_{e0}$, and elemental rate tensor estimates, $\mathcal{R}_e$, the goal of the metric optimization algorithm is to determine the step matrix field, $\mathcal{S}(\vec{x})$, that minimizes the error at a fixed cost. The step matrix field is approximated by values at the mesh vertices, $\mathcal{S}_v$, which are arithmetically-averaged to adjacent elements:

$$\mathcal{S}_e = \frac{1}{|V_e|} \sum_{v \in \mathrm{V}_e} \mathcal{S}_v, \tag{9.4}$$

where V_e is the set of vertices ($|V_e|$is the number of them) adjacent to element e. The optimization problem is to determine $\mathcal{S}_v$ such that the total error $\mathcal{E}$ is minimized at a prescribed total cost $\mathcal{C}$. First-order optimality conditions require derivatives of the error and cost with respect to $\mathcal{S}_v$. The cost only depends on the trace of the step matrix; i.e. the trace-free part of $\mathcal{S}_e$ stretches an element but does not alter its area. Therefore, the vertex step matrices are separated into trace ($s_v\mathcal{I}$) and trace-free ($\tilde{\mathcal{S}}_v$) parts, with $\mathcal{I}$ the identity tensor,

$$\mathcal{S}_v = s_v\mathcal{I} + \tilde{\mathcal{S}}_v \tag{9.5}$$

The optimization algorithm is then the same as presented by Yano [132]:

1. Given a mesh, solution, and adjoint, calculate $\mathcal{E}_e$, $\mathcal{C}_e$, $\mathcal{R}_e$ for each element e.
2. Set $\delta s = \delta s_{\max}/n_{\text{step}}$, $\mathcal{S}_v$=0.
3. Begin loop: $i = 1...n_{\text{step}}$
   (a) Calculate $\mathcal{S}_e$ from (9.4), $\frac{\partial \mathcal{E}_e}{\partial \mathcal{S}_e}$ from (9.2) and $\frac{\partial \mathcal{C}_e}{\partial \mathcal{S}_e}$ from (9.3).
   (b) Calculate derivatives of $\mathcal{E}$ and $\mathcal{C}$ with respect to s_v and $\tilde{\mathcal{S}}_v$.
   (c) At each vertex form the ratio, $\lambda_v = \frac{\partial \mathcal{E}/\partial s_v}{\partial \mathcal{C}/\partial s_v}$ and

- Refine the metric for 30% of the vertices with the largest $|\lambda_v|$:$\mathcal{S}_v = \mathcal{S}_v + \delta s \mathcal{I}$
- Coarsen the metric for 30% of the vertices with the smallest $|\lambda_v|$:$\mathcal{S}_v = \mathcal{S}_v - \delta s \mathcal{I}$

(d) Update the trace-free part of $\mathcal{S}_v$ to enforce stationarity with respect to shape changes at fixed area: $\mathcal{S}_v = \mathcal{S}_v + \delta s(\partial\mathcal{E}/\partial\tilde{\mathcal{S}}_v)/(\partial\mathcal{E}/\partial s_v)$.

(e) Rescale $\mathcal{S}_v \to \mathcal{S}_v + \beta\mathcal{I}$, where β is a global constant calculated from (9.3) to constrain the total cost to the desired dof value: $\beta = \frac{2}{d}\log\frac{C_{\text{target}}}{C}$, where C_{target} is the target cost.

Note, λ_v is a Lagrange multiplier in the optimization. It is the ratio of the marginal error to marginal cost of a step matrix trace increase (i.e. mesh refinement). The above algorithm iteratively equidistributes λ_v globally so that, at optimum, all elements have the same marginal error to cost ratio. Constant values that work generally well in the above algorithm are $n_{\text{step}} = 20$ and $\delta s_{\max} = 2\log 2$. In practice, the mesh optimization and flow/adjoint solution are performed several times at a given target cost, C_{target}, until the error stops changing. Then the target cost is increased to reduce the error further if desired.

9.1.5 Machine-Learning Based Mesh Adaptation

As an alternative to the adjoint-based mesh adaptation process, a data-driven method is developed to improve the efficiency of non-uniform anisotropic mesh generation compared with existing approaches. Optimum mesh generation for CFD simulations requires information about the elemental size as well their anisotropy, in the CFD domain. The anisotropy information becomes more relevant at higher Reynolds and Mach numbers with many directional features such as boundary layers, wakes, and shocks. In MOESS, both the primal and adjoint solutions are combined via sam-

pling of the adjoint-weighted residual to produce the most efficient element size and anisotropy distribution. This information guides the movement of mesh vertices so that each element is as close as possible to its ideal size and shape. As an alternative to determining the ideal size and anisotropy information from MOESS, an approach that uses a neural network to determine the metric from relevant features such as the flow conditions and relative position in the computational domain is presented.

9.2 Neural Network Training

The neural network is trained using meshes adapted for a variety of flow conditions and angles of attack over a NACA 0012 airfoil. All cases are two-dimensional and use the Reynolds-averaged Navier-Stokes equations. The aerodynamic lift output is considered for error estimation and mesh adaptation. Adapted meshes are generated

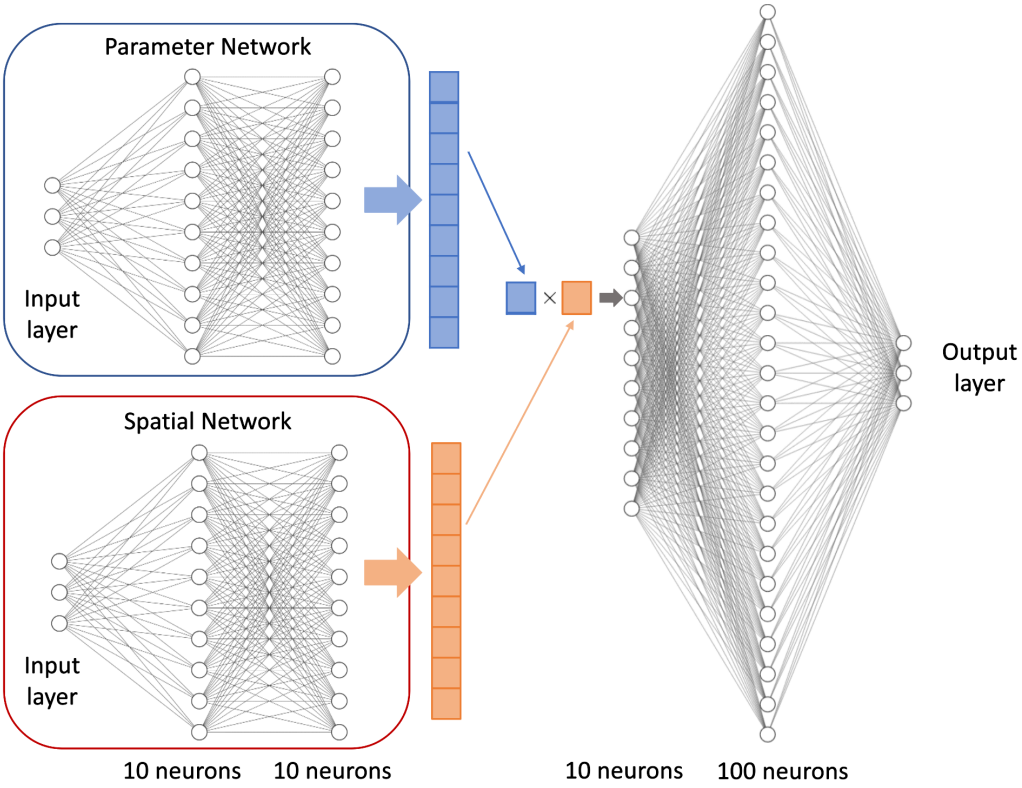


Figure 9.1: Structure of the artificial neural networks used to predict element anisotropy and sizing.

at a chosen target degree-of-freedom cost using MOESS, at a solution approximation order of $p = 2$. The Riemannian metric, described in the previous subsection, stores the ideal size, stretching and orientation of the mesh elements in a single matrix entity and is used to train the network. The metric is a general entity that can be used in any adaptation process, independent of how it is constructed and what characteristics the user wants to achieve through the adaptation process. As the metric of the adapted mesh is used to train the network, the number/distribution of elements in the initial mesh is irrelevant and any coarse mesh can be used to start the adaptation. In this work, a common initial mesh generated using BAMG [61], is used for all test cases for simplicity. The number of elements in the final adapted mesh is important as all the adapted meshes are optimized for the the final number of DOFs. Keeping the total DOFs in the optimized mesh the same helps make the training process easier but also restricts the output of the network. In the present study, networks are only trained for a constant total DOF. The neural network uses the logarithm of the Reynolds number, the angle of attack, and the Mach number as inputs to the parameter network and the normalized centroidal position of each element and the normalized wall distance as inputs to the spatial network to predict the elemental metric of the optimized mesh. The centroidal position vector of each element is normalized by the maximum distance in both dimensions. The wall distance, evaluated by averaging the wall distance of all the nodes of the element, is normalized by the maximum achievable value of the wall distance. To capture the wide range of values in the metric of the elements, the matrix logarithm of the elemental metric is used as the output of the network.

9.3 Neural Network Architecture

A fully connected multi-layer perceptron neural network, as shown in Figure 9.1 is constructed for training. The neural network consists of two individual networks, which initially process the fluid flow parameters and the spatial position of the ele-

ments separately. The parameter and spatial network have two hidden layers, consisting of 10 neurons each, between the input layer and the output layer. The outputs of the parameter and the spatial network are then combined by normalizing them and multiplying them. This serves an an input to the final network which has a single hidden layer of 100 neurons. Within each hidden layer, the input vectors are multiplied by a weight matrix before adding a bias vector. An activation function is then applied to the result to give the output of the hidden layer. The map from the input to the hidden layer and between the hidden layers involves an entry-wise rectified linear-unit (ReLU) activation function $\sigma(x) = \max(0, x)$, whereas no activation function is used for the output layer calculation. The parameters associated with the network consist of the weights and biases,

$$\mathbf{W}_i \in \mathbb{R}^{n_i \times n_{i-1}}, \quad \mathbf{b}_i \in \mathbb{R}^{n_i}, \tag{9.6}$$

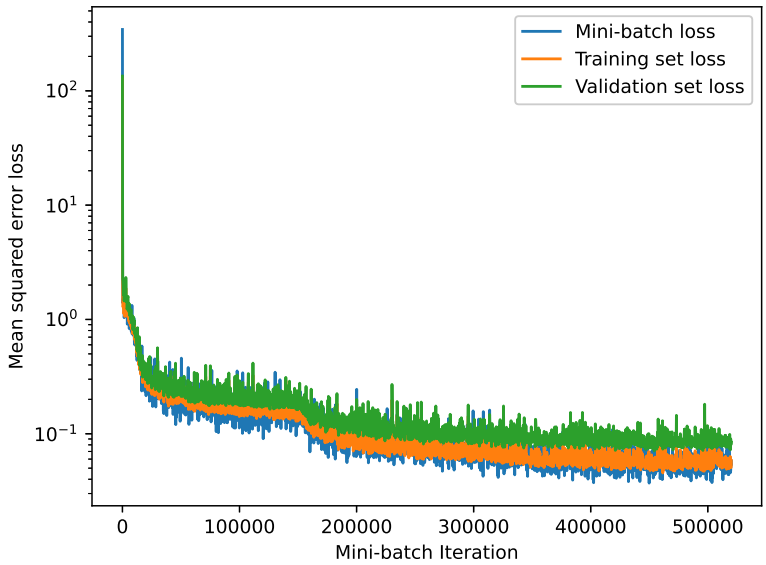


Figure 9.2: Neural network training loss history, using a 80% training, 20% validation split.

where n_i is the number of neurons in layer i. The values of these parameters are determined using an optimization procedure, the adaptive moment (Adam) estimation algorithm in TensorFlow [2], that minimizes the mean squared error loss function between predicted and actual output layer values. The actual values come from training data, which are obtained from the meshes adapted using MOESS on prototypical cases.

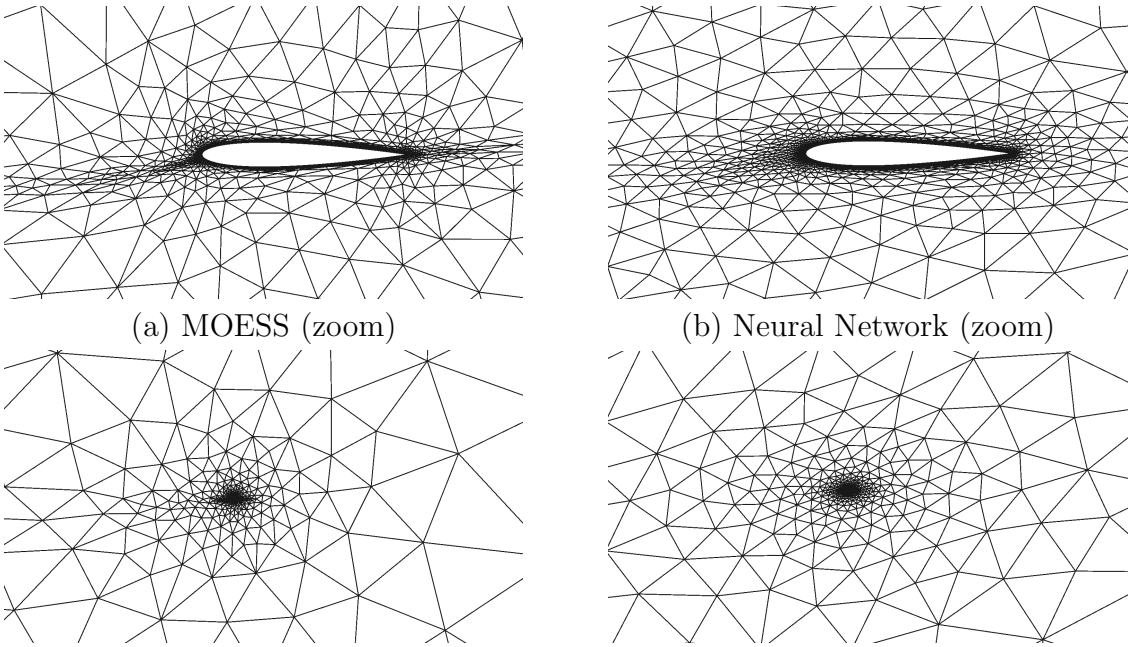
Each element in an adapted mesh serves as a training data point. These data points are split 80%/20% into training and validation categories. The training data are used to drive the optimization, whereas the validation data are used to monitor the loss on untrained data. The training data are broken into mini-batches of size 100 for the optimizer, and the learning rate is set to .005. Prior to training, the weights and biases are initialized randomly from a unit normal distribution. Several tens of thousands of optimization iterations typically lead to a stabilization of the mean-squared error, as shown in Figure 9.2. The mean squared error drops by an order of magnitude, without a significant difference between training and test data loss. This indicates that the network is not over-fitted for the training data and performs well in predicting outputs for the testing data set. Furthermore, the results of the training were not found to be overly sensitive to the choices of the mini-batch size, learning rate, or the initialization.

9.4 Results

This section presents results obtained from implementing the trained neural network to generate optimized starting meshes for various flow cases. The meshes obtained from the neural network are compared to meshes obtained using MOESS. In all cases, metric-based re-meshing is performed using BAMG.

9.4.1 Varying Reynolds number and angle of attack

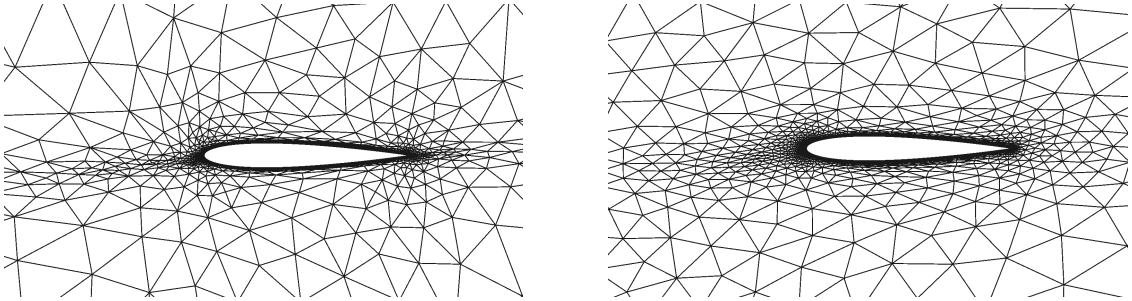
The first case that is tackled in this study is the optimal mesh generation for a two-dimensional steady-state flow over a NACA 0012 airfoil for a variety of Reynolds numbers and angles of attack at a constant Mach number. Two Mach numbers, one in the subsonic regime, $M = 0.25$, and one in the transonic regime, $M = 0.75$, are used in this case. The airfoil is located in the center of the domain, the boundary of which consists of a square box which spans from $[-100c, 100c]$ in both dimensions. The fluid flow is simulated using a RANS solver with the SA turbulence model [24]. For training and testing purposes, 100 optimized meshes are generated for various flow conditions, where $Re \in [10^6, 10^7]$ and angle of attack, $\alpha \in [0, 5]^\circ$, and these meshes are obtained using MOESS. To sample the parameter space, Latin hypercube sampling is used in this work. The data from the various flow cases are randomized and split 80%/20% into training and validation categories. For each training



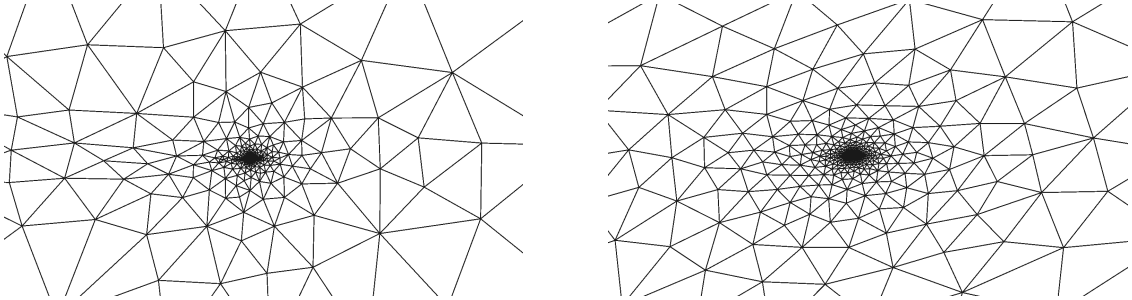
(a) MOESS (zoom) (b) Neural Network (zoom)

(c) MOESS (d) Neural Network

Figure 9.3: Comparison of element distribution in adapted meshes obtained using MOESS and the neural network for a flow at $M = 0.25$, $\alpha = 4.6^\circ$ and $Re = 3.5 \times 10^6$.

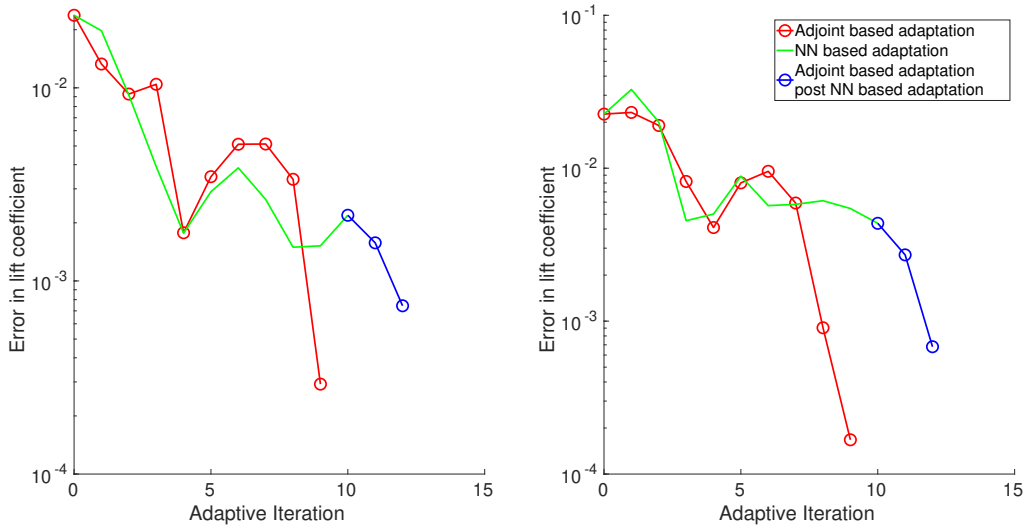


(a) MOESS (zoom) (b) Neural Network (zoom)



(c) MOESS (d) Neural Network

Figure 9.4: Comparison of element distribution in adapted meshes obtained using MOESS and the neural network for a flow at $M = 0.25$, $\alpha = 2.65°$ and $Re = 4 \times 10^6$.



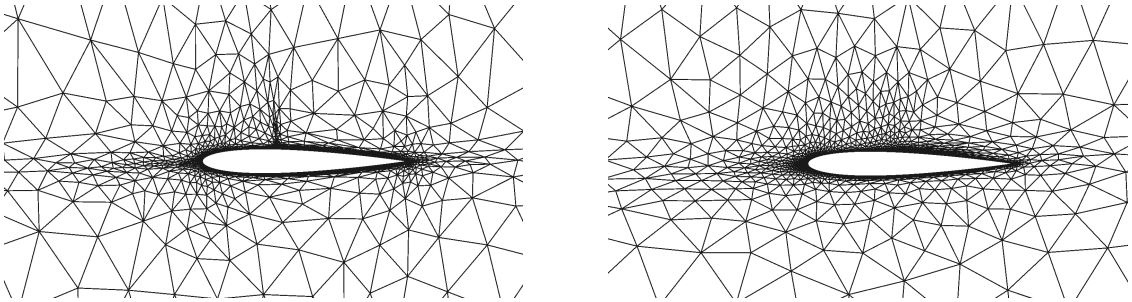
(a) $\alpha = 2.65°$ and $Re = 4 \times 10^6$ (b) $\alpha = 4.6°$ and $Re = 3.5 \times 10^6$

Figure 9.5: Comparison of output convergence in adapted meshes obtained using MOESS and the neural network for a flow at $M = 0.25$.

case, starting from a coarse initial mesh with 500 triangular elements, the meshes are subjected to 10 cycles of mesh adaptation with a growth factor of $G_f = 1.5$ and a constraint on the total degree of freedom, DOF=50,000. The total number of degree of freedom after each adaptive iteration is set by the growth factor. The resulting adapted meshes have approximately 8500 triangular elements of approximation order $p = 2$.

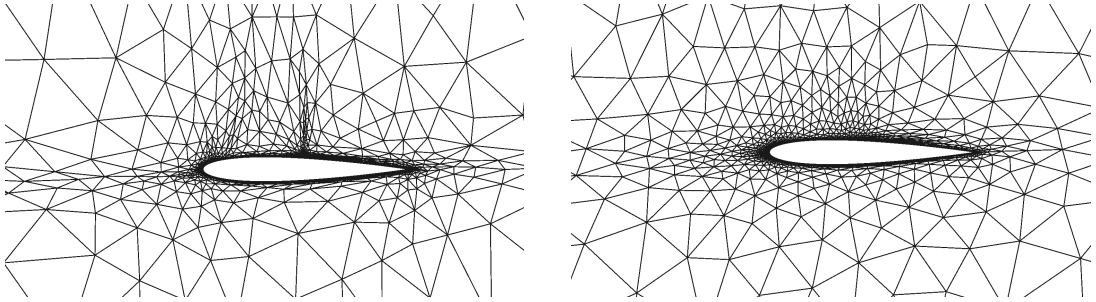
A Riemannian metric, generated using the flow and adjoint solution in MOESS or predicted using the neural network, guides the mesh adaptation process. In the neural network, the predicted metric is obtained by taking the matrix exponent of the output obtained from the network. The predicted metric is then provided to BAMG for mesh generation and using the same set of growth factor and number of adaptive iterations as MOESS, the adapted mesh is obtained. Note, in the neural-network approach, no primal or adjoint solutions are needed during the course of adaptation, as the metric only depends on the parameter and spatial network inputs shown in Figure 9.1, and not on the state or adjoint. To test the prediction capability of the neural network, adapted meshes and output convergence in the adapted meshes are compared between the neural network and MOESS for the flow conditions in the validation set. Figure 9.3 and Figure 9.4 compare adapted meshes for two flow conditions in the validation set at $M = 0.25$. For both of the flow conditions in the subsonic regime, the network is able to reproduce meshes with good detail. The network is able to successfully produce the anisotropic elements above and below the airfoil, thereby, resolving the boundary layer. The element distribution away from the airfoil also looks similar to that of the optimized meshes obtained from MOESS, where large isotropic elements are used.

Figure 9.5 compares the convergence of the lift coefficient for which both the meshes



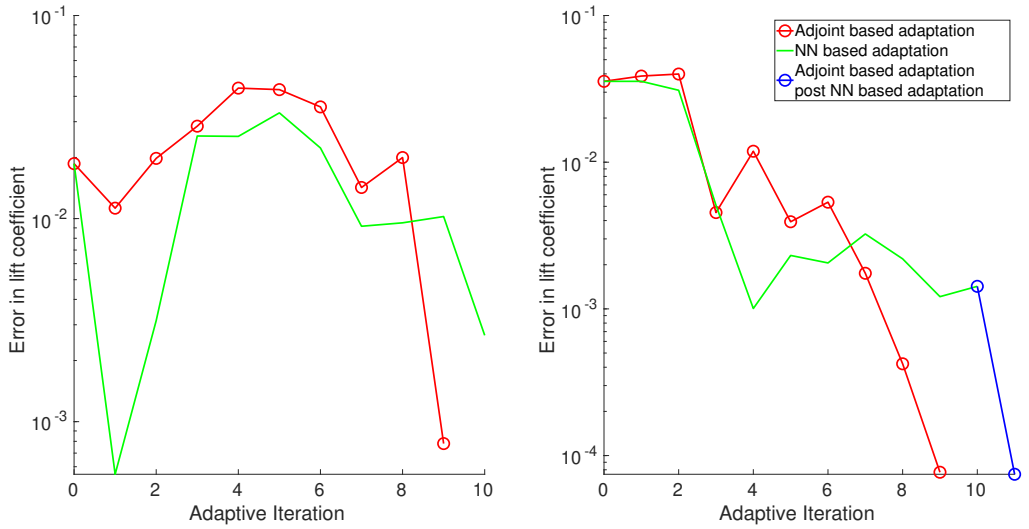
(a) MOESS (zoom) (b) Neural Network (zoom)

Figure 9.6: Comparison of element distribution in adapted meshes obtained using MOESS and the neural network for a flow at $M = 0.75$, $\alpha = 1.06°$ and $Re = 3.5 \times 10^6$.



(a) MOESS (zoom) (b) Neural Network (zoom)

Figure 9.7: Comparison of element distribution in adapted meshes obtained using MOESS and the neural network for a flow at $M = 0.75$, $\alpha = 2.73°$ and $Re = 9.47 \times 10^6$ $\alpha = 0$.



(a) $\alpha = 2.73°$ and $Re = 9.47 \times 10^6$ (b) $\alpha = 1.06°$ and $Re = 3.5 \times 10^6$

Figure 9.8: Comparison of output convergence in adapted meshes obtained using MOESS and the neural network for a flow at $M = 0.75$.

have been adapted. The final mesh obtained after 10 cycles of MOESS serves as the reference mesh. The output convergence in the meshes adapted using the network show, on average, a reduction in the output error after every adaptation cycle and give comparable accuracy relative to the meshes obtained from MOESS. The network does not outperform the meshes obtained using MOESS as the error in training also affects the quality of the output meshes, thereby affecting the output error. The meshes obtained from the network also mirror the distribution and anisotropy in elements and are close to being optimal for the output evaluation, given the constraint on the DOF. In the cases where the meshes obtained from the network are sub-optimal, a few cycles of adjoint-based mesh adaptation with the output mesh from the network serving as the starting mesh can be used to further improve the meshes, as shown by the blue curves in Figure 9.5. The meshes can be improved for the target DOF by choosing a unit growth factor, $G_f = 1$, and reshuffling the elements, or for a higher growth factor for obtaining optimal meshes for larger DOF values. The number of MOESS adaptive iterations needed to achieve higher accuracy will be fewer in the case when a network is used initially as compared to just using MOESS. This can be observed in the blue curve in Figure 9.5, where only two adaptive iterations of MOESS are required on the final mesh obtained by the neural network to produce an optimal mesh with similar errors. A growth factor of one is used for the MOESS adaptive iterations shown in the blue curve.

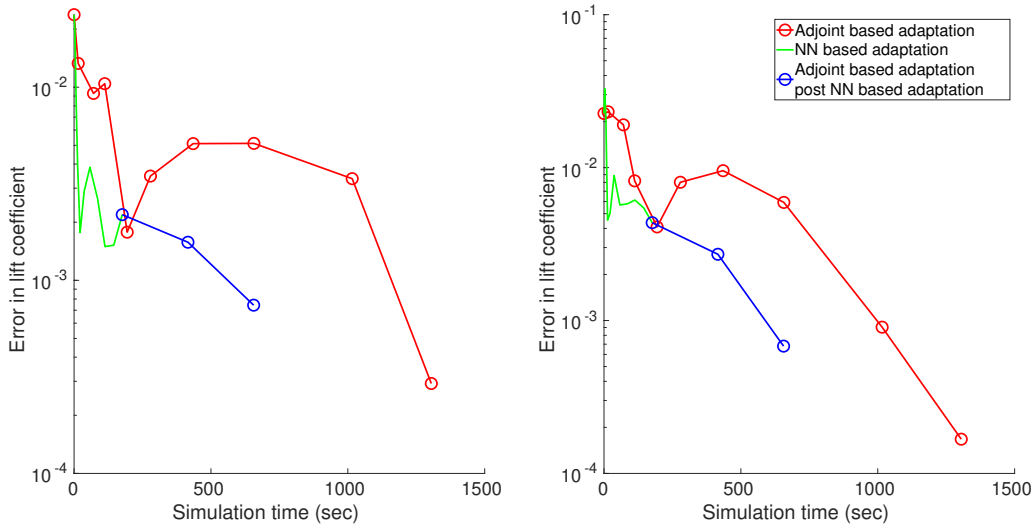
The adapted meshes for the transonic case at $M = 0.75$ show similar results, as shown in Figure 9.6 and Figure 9.7. The adapted meshes obtained from the network are able to successfully resolve the boundary layer and stagnation streamline. However, the shock is under-resolved by the network. The lack of resolution around the shock compared to the boundary layer results in the network not resolving the shock accurately. The output convergence is unaffected by the under resolution of

the shocks as the meshes obtained for the network are able to give comparable accuracy to that obtained from MOESS, as shown in Figure 9.8. Figure 9.9 compares the simulation times for the two mesh adaptation strategies for both the subsonic and transonic cases. Mesh adaptation using the network avoids the primal and adjoint solutions needed in MOESS, resulting in a 10x-15x savings in computational cost. The only time consuming component of the mesh adaption using the neural network is from the mesh generator, BAMG.

9.4.2 Varying Mach number, Reynolds number and angle of attack

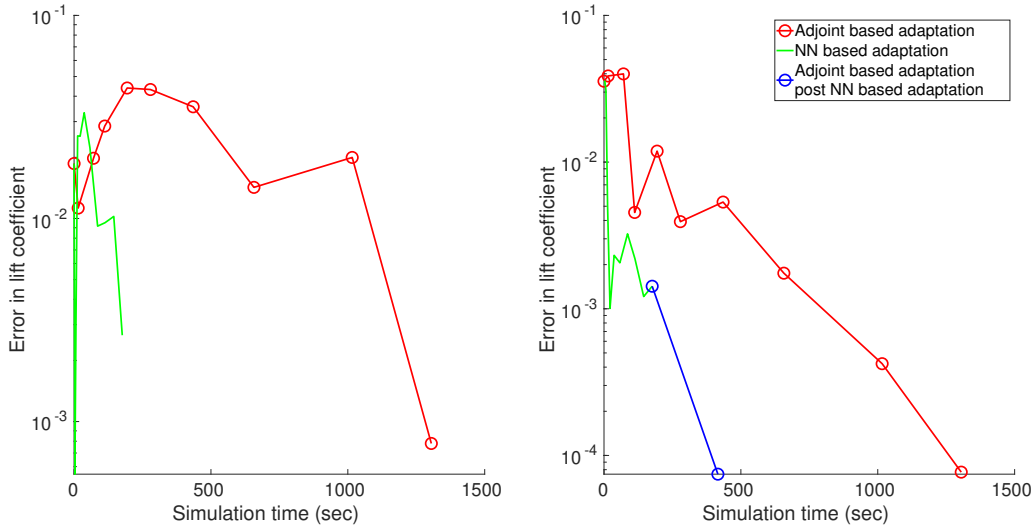
The second case that is tackled in this study is the optimal mesh generation for a two dimensional steady-state flow over a NACA 0012 airfoil for a variety of Mach numbers, Reynolds numbers and angles of attack. For training and testing purposes, 100 optimized meshes are generated for 100 different flow conditions using MOESS, where $M \in [0.25, 0.8]$, $Re \in [10^6, 10^7]$ and angle of attack, $\alpha \in [0, 5]^\circ$. The various flow conditions are decided by Latin hypercube sampling of the parameter space. Similar initial meshes and mesh adaptation parameters are used in this case as in the first case.

Figure 9.10 and Figure 9.11 compare adapted meshes for a transonic and subsonic flow condition, respectively, in the validation set. While the adapted meshes from the network perform well in the subsonic regime, they struggle to resolve the shock flow feature, stagnation streamlines, and the streamlines exiting from the trailing edge, in the transonic regime. This also results in poor output convergence for the transonic cases, as shown in Figure 9.12. The reason for the poor optimal meshes in the transonic regime is due to the lack of transonic cases used to train the network. Shock features are present in $M \in [0.5, 0.8]$ at high angles of attack, but such cases comprise only 15% of the total training data set. Secondly, in the case of shocks,



(a) $M = 0.25, \alpha = 2.65°$ and $Re = 4 \times 10^6$

(b) $M = 0.25, \alpha = 1.06°$ and $Re = 3.5 \times 10^6$



(c) $M = 0.75, \alpha = 2.73°$ and $Re = 9.47 \times 10^6$

(d) $M = 0.75, \alpha = 1.06°$ and $Re = 3.5 \times 10^6$

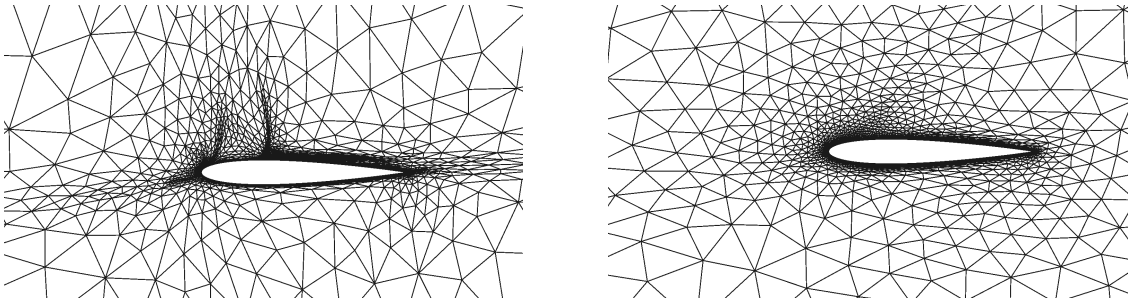
Figure 9.9: Comparison of the average time required for adapting a fluid mesh using MOESS and the neural network.

the ratio of elements used to resolve the shock compared to the boundary layer is low. This leads to less emphasis on the shock position and shock strength by the network. As the flow features targeted by mesh adaptation depend strongly on the Mach number regime, training the networks for subsonic and transonic cases separately can leads to better optimal meshes. Separating the network and training for obtaining optimal meshes only in the transonic regime where $M \in [0.5, 0.8]$ leads to much better output convergence, as shown in Figure 9.13. This better convergence in the outputs is observed due to better resolution of the stagnation streamlines and the wake but not to increased shock resolution.

The benefits of the neural network are more pronounced when the mesh adaptation process begins with coarser initial meshes. The lack of convergence in the steady state primal and adjoint solvers due to the coarseness of the mesh affects the mesh adaptation in MOESS. However, the neural network mesh adaptation process is unaffected by the initial mesh, as no solutions are required to drive the adaptation on the meshes.

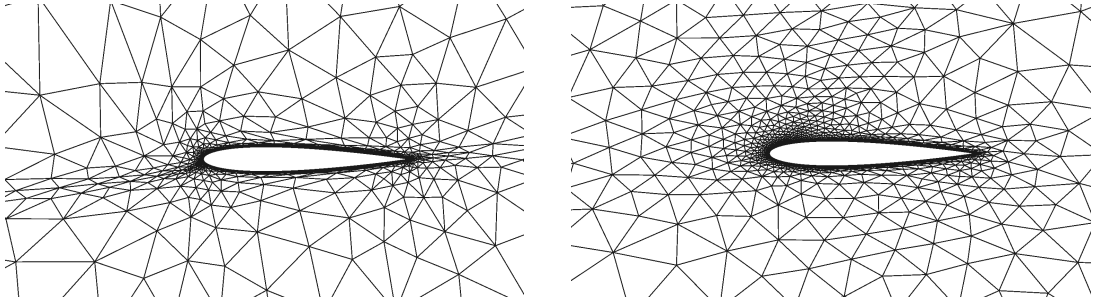
9.4.3 Entropy Adjoint - Varying Mach number, Reynolds number and angle of attack

A further test to examine the capability of the network is by training it to reproduce meshes optimized using the entropy adjoint instead of the lift adjoint. The entropy-based adjoint indicator uses entropy variables to drive the mesh adaptation [49, 33]. The areas of the mesh that are targeted by the entropy adjoint indicator are those regions that exhibit high net production of spurious entropy. Without the need for a separate adjoint solution to obtain the entropy variables, adapting using entropy variables instead of the output-based adjoint is far less computationally expensive. However, this indicator does not disregard areas of spurious entropy



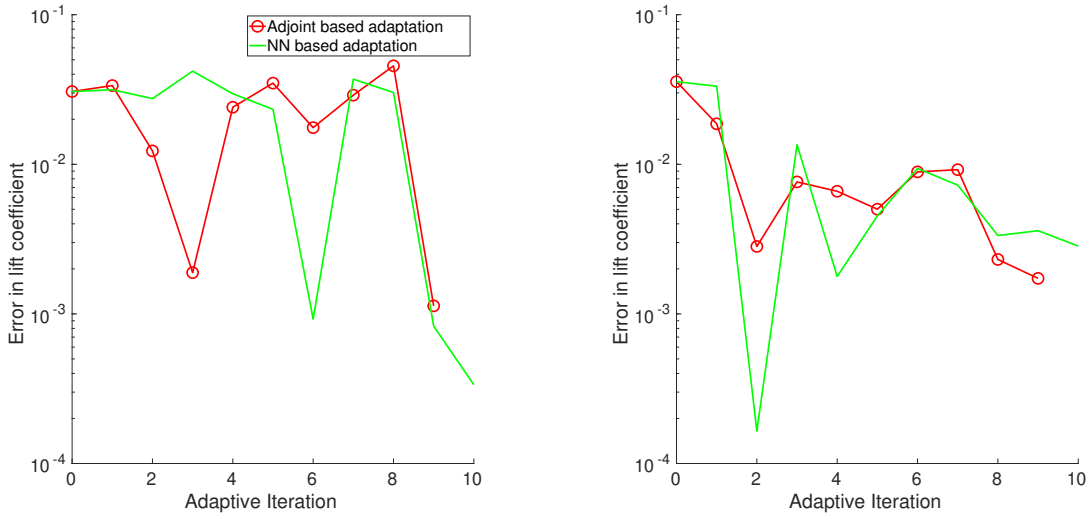
(a) MOESS (zoom) (b) Neural Network (zoom)

Figure 9.10: Comparison of element distribution in adapted meshes obtained using MOESS and the neural network for a flow at $M = 0.68$, $\alpha = 4.94°$ and $Re = 3.05\times10^6$.



(a) MOESS (zoom) (b) Neural Network (zoom)

Figure 9.11: Comparison of element distribution in adapted meshes obtained using MOESS and the neural network for a flow at $M = 0.44$, $\alpha = 4.96°$ and $Re = 7.92\times10^6$.



(a) $M = 0.68$, $\alpha = 4.94°$ and $Re = 3.05 \times 10^6$

(b) $M = 0.44$, $\alpha = 4.96°$ and $Re = 7.92 \times 10^6$

Figure 9.12: Comparison of output convergence in adapted meshes obtained using MOESS and the neural network. The network is trained using meshes adapted for varying Mach number, angle of attack and Reynolds number flows.

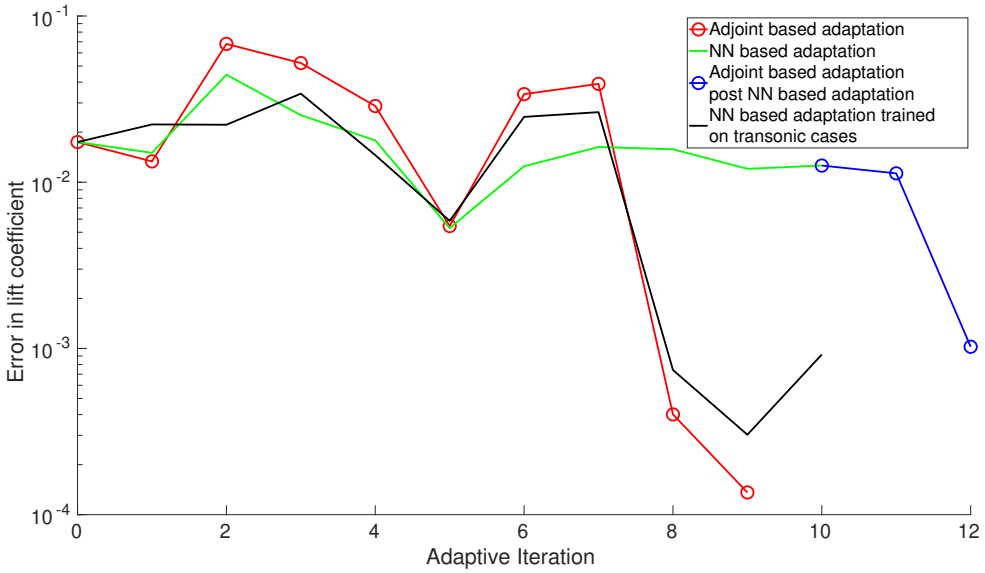
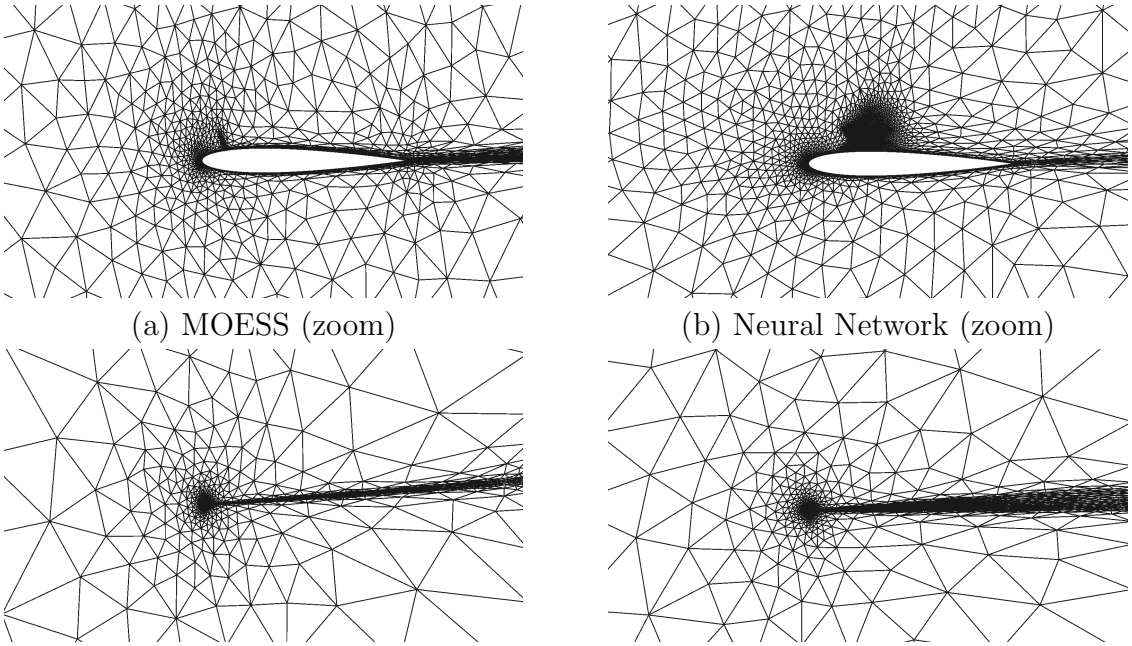


Figure 9.13: Comparison of output convergence in adapted meshes obtained using MOESS and two neural networks for a flow at $M = 0.68$, $\alpha = 4.94°$ and $Re = 3.05 \times 10^6$. The two networks are trained using meshes adapted for varying Mach number, angle of attack and Reynolds number and varying Mach number in the transonic regime, angle of attack and Reynolds number flows.

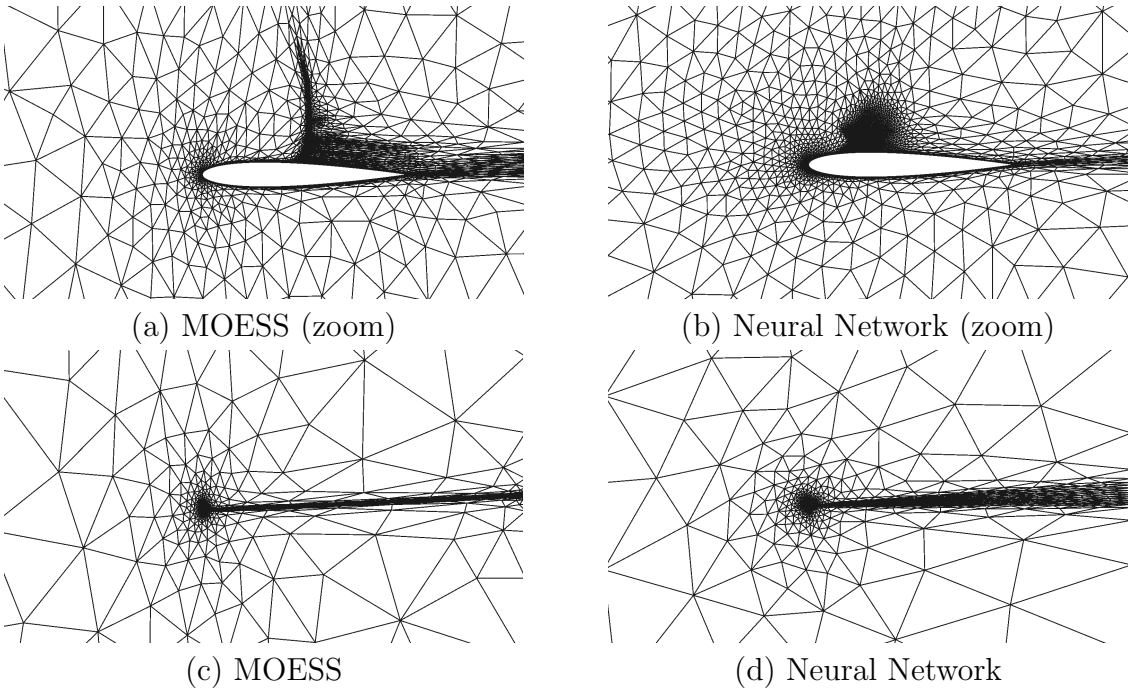
generation that have no effect on a particular engineering output. This may lead to over-refinement, particularly for cases with flow discontinuities. The entropy adjoint resolves the boundary layer, the wake (extending up the boundary of the domain) and also the shock (if present). Due to a greater number of flow features that the entropy adjoint focuses on, it acts as a good litmus test and stresses the network to the maximum. A similar network as previously discussed is used for this test case. For training and testing purposes, 100 optimized meshes are generated for various flow conditions in the transonic regime, where $M \in [0.5, 0.8]$, $Re \in [10^6, 10^7]$ and angle of attack, $\alpha \in [0, 5]°$ using MOESS. Figures 9.14 and 9.15 compare adapted meshes for two flow conditions in the transonic regime, from the validation set. The network is able to accurately resolve the wake and the boundary layer. However, the network struggles in predicting the shock position and strength. For this Mach number range, the shock traverses above the airfoil, increasing in strength while moving towards the



(a) MOESS (zoom) (b) Neural Network (zoom)

(c) MOESS (d) Neural Network

Figure 9.14: Comparison of element distributions in adapted meshes obtained using MOESS and the neural network for a flow at $M = 0.60$, $\alpha = 4.45^\circ$ and $Re = 4.54 \times 10^6$. The entropy adjoint is used to adapt the meshes obtained using MOESS.



(a) MOESS (zoom) (b) Neural Network (zoom)

(c) MOESS (d) Neural Network

Figure 9.15: Comparison of element distributions in adapted meshes obtained using MOESS and the neural network for a flow at $M = 0.76$, $\alpha = 2.8^\circ$ and $Re = 6.67 \times 10^6$. Entropy adjoint is used to adapt the meshes obtained using MOESS.

trailing edge, as the Mach number increases. Instead of refining around the shock location, the network refines the entire region where the shock traverses in an isotropic manner. As the Mach number increases, the refined region in the network adapted meshes moves along the chord, mimicking the shock traversal.

For the first two cases in this study, the lack of training cases containing shocks and less emphasis on the shock features explained the poor resolution of the shock feature in the meshes adapted by the network. However, the entropy adjoint removes both of these pitfalls and the network still struggles to resolve the shock. The numbers of elements used to resolve the shock, the boundary layer, and the wake are comparable in the case of a meshes adapted using the entropy adjoints and 60% of the training dataset has shocks present. Resolving the shock flow feature with the flow parameters needs more examination, and different network structures are being investigated to resolve this issue. A promising avenue is to use the flow features to first predict the shock position and then to combine this information with the existing network.

CHAPTER X

Conclusions and Future Work

10.1 Summary and Conclusions

The work presented in this dissertation was motivated by the need for an accurate and efficient fluid-structure interaction (FSI) solver for aeroelastic applications. The thesis presented the development of a high-order FSI solver based on the partitioned approach. The governing equations of the fluid subsystem are discretized in space using a high-order discontinuous Galerkin method while a continuous Galerkin method is used for the spatial discretization of the structural governing equations. A fourth-order implicit Runge Kutta scheme is used to discretize the individual subsystems in time. To maintain the high-order accuracy of the coupled solver, the fluid and structural subsystems are tightly coupled in both space and time. A local interpolation strategy has been developed to accurately transfer pressures at the interface of the fluid and structural subsystem and radial basis function with compact support are used to transfer structural displacements and velocities at the interface. An implicit-explicit Runge Kutta scheme is implemented to maintain the high-order temporal nature of the solver by exchanging information between the two subsystems at each stage of each step. The FSI solver is verified by predicting flutter on a pitching-plunging NACA 0012 airfoil and the high-order convergence of the FSI solver is validated for a flow over a two-dimensional cantilevered beam.

The thesis presented an analysis of the errors generated by mesh deformation algorithms, which are used to track the FSI interface by deforming the fluid mesh. Two mesh deformations algorithms are applied in this work: 1) an explicit mapping approach; and 2) an interpolation-based approach. Output-based error estimation is used to estimate and control such errors arising from mesh deformation. A two-dimensional free-stream preservation test in an inviscid flow is used to quantify error due to mesh-motion algorithms. An output-based error estimate is used to optimize the mesh motion algorithm by optimizing the variables used to blend the deformation. For an explicit mapping, an optimized inner and outer radius of the blending is obtained for a steady and unsteady deformation resulting in the least error in the output. A secondary case of an airfoil undergoing rigid body deformation in a steady fluid flow is analyzed to observe the effects of the position of the blending region on the output convergence. A better understanding of the error generated by the mesh motion algorithms is achieved and a set of guidelines have been proposed to select user-defined blending regions in mesh motion algorithms to achieve low mesh motion errors and better output convergence. The output convergence study verifies that the implementation of a geometric conservation law is not necessary for achieving high accuracy in high-order FSI simulations involving rigid body motions. The use of output-based mesh adaptation in efficiently reducing the spatial errors generated by the mesh distortion as well the spatial discretization has been demonstrated, thus, showing its applicability to FSI simulations.

The thesis presented the development and application of goal-oriented mesh adaptation in a high-fidelity FSI solver. Adaptive meshing in both the fluid and the structural subsystems is applied using a coupled adjoint for aeroelastic problems of varying complexity. The coupled adjoint provides the sensitivity of the coupled output

of interest to the space-time discretization of the aeroelastic problem and is obtained by solving the adjoint system in a similar fashion as the primal problem. Sensitivity perturbation tests are used to verify the coupled adjoint evaluation for both two-dimensional and three-dimensional aeroelastic cases. Adjoint-weighted residuals provide spatial error estimates that guide the mesh adaptation in the fluid and structural grids. The mesh adaptation occurs in the spatial order of mesh elements in the fluid mesh. Conversely, the structural elements in the structural mesh are refined in the element size. The benefits of adaptive meshing are demonstrated for three aeroelastic cases: 1) a two-dimensional pitching-plunging NACA 0012 airfoil subjected to a subsonic flow; 2) a cantilevered beam subjected to a subsonic flow; and 3) a cantilevered wing subjected to a subsonic and transonic flow. The benefits are showcased by comparing the output convergence of coupled outputs against common adaptation methods, such as uniform refinement. The adaptive meshing procedure improves in regions important to both subsystems and helps in obtaining accurate outputs at a smaller number of degrees of freedom. Flow features such as the boundary layer, wake, stagnation streamlines, and shocks along with regions in the structural and fluid mesh undergoing large deformations, which contribute largely to the error in an FSI simulation, are correctly identified by the adaptive meshing procedure and subsequently refined. Adapted meshes are able to provide a reduction of almost 60% in the total degrees of freedom in two-dimensional aeroelastic problems and 80% in three-dimensional aeroelastic cases, while providing the similar or better levels of accuracy, when compared to uniform mesh refinement. These benefits depend on the number of cycles of mesh adaptation, growth factor, and the initial fluid and structural meshes. Overall, the adaptive meshing procedure provides a significant increase in the accuracy of unsteady outputs for aeroelastic problems compared to more common adaptation methods, such as uniform refinement.

Solution-adaptive methods based on the adjoint-weighted residual have been shown to improve the accuracy and efficiency of FSI simulations. Despite their great success in aerospace applications, the additional computational cost and implementation complexity associated with adjoint-based methods cannot be neglected. As an alternative to the adjoint-based mesh adaptation process, a data-driven method is developed to improve the efficiency of non-uniform anisotropic mesh generation compared with existing approaches. A machine-learning approach is developed for determining the optimal anisotropic initial meshes in the context of an output-based adaptive solution procedure. Artificial neural networks are used to predict the desired element sizing and anisotropy from flow conditions and relative position in the fluid domain. The benefits of mesh adaptation using a neural network are demonstrated for a steady-state flow over a NACA 0012 airfoil, where the network is able to produce optimal starting meshes for a variety of Mach numbers, Reynolds numbers, and angles of attack. Meshes optimized using the lift and the entropy adjoint are used to train the network. The meshes generated from the neural network show output convergence comparable to that obtained using adjoint-based mesh adaptation. Mesh adaptation using the machine learning approach produces optimal meshes without solving for the state or the adjoint, leading to a factor 10x-15x savings in computational time. The network is able to successfully resolve flow features such as the boundary layer, wake, and stagnation streamlines. In the presence of shocks in transonic flow, the network struggles to resolve around the shock location and further investigation is needed. The network proves to be a good alternative to generating initial meshes for computational fluid dynamics and can be extended to FSI.

10.2 Research Contributions

The major contributions of this dissertation are:

- Developed a high-order partitioned FSI solver by combining high-order spatial and temporal discretization techniques. The FSI solver maintains the high-fidelity nature of the solution in the case of non-matching meshes and varying discretizations at the fluid-structure interface.

- Developed guidelines for generating initial meshes with low mesh-motion errors and better output convergence for CFD simulations with deforming domains.

- Developed goal-oriented coupled mesh adaptation for high-order partitioned FSI solvers and demonstrated the benefits of adaptive meshing in two-dimensional and three-dimensional aeroelastic problems.

- Explored the feasibility of a convolutional neural network for generating optimal initial meshes for CFD problems without solving for the adjoint variables.

10.3 Future Work

Certain topics for future work were identified during the course of this work. These topics are listed below:

- **Geometric and material non-linearity:** The benefits of high order and mesh adaptivity have been showcased in aeroelastic problems with linear structures, undergoing small deformation. In these aeroelastic cases, the non-linearities arise primarily from the fluid subsystem, however, a similar analysis can be conducted with a more sophisticated structural solver which can handle geometric and material non-linearity.

- **Design Optimization:** The adaptive high-fidelity FSI solver can be incorporated in a multidisciplinary design optimization framework to help design efficient aircraft. Coupling of the FSI solver with other subsystems such as

thermal and propulsion can help analyze and design critical subsystems that affect performance.

- **Adaptive temporal refinement:** The adaptation mechanics presented in this work have been limited to refining only the spatial discretization of the FSI system. However, further efficiency can be gained by permitting mesh adaptation in the temporal discretization. Coupled mesh adaptation in time will require the evaluation of the temporal error estimates, which need the coupled adjoint and the projected residual to be evaluated in a finer space-time grid. The existing temporal reconstruction developed for the coupled adjoint solve can be reused for evaluating the states in a finer temporal grid. The temporal error-estimates can be used to guide mesh adaptation in the temporal discretization by uniformly reducing the overall time step size or by using non-uniform time stepping. Varying time-step sizes can be used in the individual subsystems, however maintaining the high-fidelity nature of the FSI solver will require further sub-iterations.

- **Machine learning based mesh generation:** The existing machine-learning framework to generate optimal initial meshes faced issues in the presence of shock flow features. A promising avenue is to use the existing flow parameters to firstly predict the shock position, if present, by using a network and then use this information as an additional input parameter to the existing neural network for mesh generation to get better mesh refinement near the shock. Using meshes adapted for more complex outputs of interest such as fluctuating moment can be used as inputs to the network. Extending the existing machine-learning framework for unsteady simulations can be useful for multi-physics simulations such as FSI, where output convergence strongly depends on the initial mesh. Extension of the framework to unsteady CFD simulations with moving shocks is

also another promising avenue where optimal meshes generated by the network based on the current position of the shock can be used, similar to unsteady mesh refinement. Alternatively, in the case of moving/oscillating shocks, the network can be modified to create a single optimal mesh by adding two additional input parameters, which are the average position and the amplitude of oscillation of the shock. The resulting mesh will have anisotropic mesh adaptation in the region of the fluid mesh where the shock traverses.

- **Initial state prediction:** The existing machine learning framework can also be used to predict state information, instead of the element size and anisotropy, at a given point on the mesh.

- **Combined h and p refinement**: The adaptation mechanics presented in this work refine the fluid mesh using p refinement and the structural mesh using h refinement. However, solution-based combined hp-adaptations have been shown to be very effective in minimizing the computational cost to achieve a given level of accuracy [133]. The advantage is expected to be more pronounced when the solution or geometry is not smooth. Combining h refinement with p refinement in the structural discretization will require the development of a displacement transfer algorithm which preserves the high-fidelity nature of the solution at the interface. A strategy similar to the local interpolation-based approach, developed for the force transfer, can be pursued.

APPENDIX

APPENDIX A

Coupled Adjoint Evaluation

A.1 Coupled Adjoint Evaluation for the Cantilevered Plate

For an unsteady output of interest, J, the fluid and structural adjoint equations are given as,

$$\begin{bmatrix} \mathbf{M}^f & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^s \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\Psi}}^f \\ \dot{\boldsymbol{\Psi}}^s \end{bmatrix} + \begin{bmatrix} \frac{\partial \mathbf{r}^{f\top}}{\partial \mathbf{U}^f} & \frac{\partial \mathbf{r}^{s\top}}{\partial \mathbf{U}^f} \\ \frac{\partial \mathbf{r}^{f\top}}{\partial \mathbf{U}^s} & \frac{\partial \mathbf{r}^{s\top}}{\partial \mathbf{U}^s} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}^f \\ \boldsymbol{\Psi}^s \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial \mathbf{U}^f}^\top \\ \frac{\partial J}{\partial \mathbf{U}^s}^\top \end{bmatrix} \tag{A.1}$$

Off-diagonal derivative evaluation

This section discusses the evaluation of the off-diagonal derivatives needed for solving the coupled adjoint. These derivatives consist of the linearizaion of the discrete fluid spatial residuals with respect to the structural state, $\frac{\partial \mathbf{r}^s}{\partial \mathbf{U}^f}$, and the structural spatial residual with respect to the fluid state, $\frac{\partial \mathbf{r}^f}{\partial \mathbf{U}^s}$.

A.1.1 Linearization of the structural spatial residual with respect to the fluid state

The structural spatial residual, obtained from spatially discretizing the linear elasticity equations, consists of the stiffness damping, and forcing terms. The only term in the structural spatial residual dependent on the fluid state is the force vector acting on the structural nodes. Thus, the linearization of the structural spatial residual with respect to the fluid state is given as,

$$\frac{\partial \mathbf{r}^s}{\partial \mathbf{U}^f} = \frac{\partial \mathbf{F}^s}{\partial \mathbf{U}^f}. \tag{A.2}$$

The force vector, $\mathbf{F}^s$, evaluated by integrating the pressure on an element with its shape functions, is given as

$$\mathbf{F}^s = [F^{s_1}, F^{s_2}, F^{s_3}...., F^{s_n}], \tag{A.3}$$

where F^{s_i} is the force vector on the i^{th} structural element and n is the total number of structural elements. Similarly, the force vector, $\mathbf{F}^f$, evaluated by integrating the pressure on an fluid element at the interface, is given as

$$\mathbf{F}^f = [F^{f_1}, F^{f_2}, F^{f_3}...., F^{f_m}] \tag{A.4}$$

where F^{f_i} is the force vector on the i^{th} fluid element and m is the total number of structural elements. The force vector on a single structural element depends on the force acting on the overlapping fluid elements. The polynomial variation of pressure over each fluid element at the interface is transferred to the overlapping structural element resulting in a continuous/discontinuous (in the case of fluid element not matching with the structural elements at the interface) distribution of pressure over each structural element. Due to the dependence of the structural forces on the force

developed in the fluid solver, the derivative evaluation can be rewritten as,

$$\frac{\partial \mathbf{F}^s}{\partial \mathbf{U}^f} = \frac{\partial \mathbf{F}^s}{\partial \mathbf{F}^f} \times \frac{\partial \mathbf{F}^f}{\partial \mathbf{U}^f} \tag{A.5}$$

$$\mathbf{F}^s = \mathbf{M}^{fs}\mathbf{F}^f \tag{A.6}$$

The transformation of the fluid forces at the interface to the structural forces is achieved by a transformation matrix, $\mathbf{M}^{fs}$, which accounts for the first term in Eq A.6. This matrix is precomputed at the beginning of each adaptation cycle, during the primal solution, after the meshes of both the subsystems are fixed and is available to use during the adjoint solution.

$$\frac{\partial \mathbf{F}^s}{\partial \mathbf{U}^f} = \mathbf{M}^{fs}\frac{\partial \mathbf{F}^f}{\partial \mathbf{U}^f} \tag{A.7}$$

The second term, which evaluates the linearization of the fluid forces at the interface with respect to the fluid states, is readily available in the fluid solver. The linearization of the structural residual with respect to the fluid state is evaluated by combining these two derivatives using chain rule.

A.1.2 Linearization of the fluid residual with respect to the structural state

The position of the structure inside the fluid mesh dictates the application of the no-slip boundary conditions in the fluid subsystem. Thus, the fluid spatial residual depends on the position of the structure. In the case of the plate, every single node on the structural mesh of the plate has three degrees of freedom, which are the out of plane displacement (z) and the two in-plane rotations (x & y). However, the only structural degree of freedom which is used to deform the structure in the fluid mesh is the out of plane displacement. Radial basis functions based on the z-displacement are

used to deform the fluid mesh and thus affect the fluid spatial residual. Due to this approximation, the derivatives of the fluid residual with respect to the structural state are non-zero for the out of plane displacement and zero for the two in-plane rotations.

Let the fluid spatial residual, which depends on the structural state, be defined as

$$\mathbf{r}^f = \mathbf{M}\mathbf{U}^s, \tag{A.8}$$

where $\mathbf{M}$ is the derivative of interest. The derivative can be evaluated using finite differences, where the fluid residual is re-evaluated after perturbing the z-displacement of a single structural node while keeping the displacement of all the other structural nodes fixed, as follows

$$\left[\frac{\partial \mathbf{r}^f}{\partial \mathbf{U}^s}\right]_i = \frac{\mathbf{r}^f(U^{s_1}, U^{s_2}, ..., U^{s_i} + \Delta U^{s_i}, ..., U^{s_n}) - \mathbf{r}^f(U^{s_1}, U^{s_2}, ..., U^{s_i}, ..., U^{s_n})}{\Delta U^{s_i}}. \tag{A.9}$$

where U^{s_i} is the out of plane displacement of the i^{th} structural node and n is the total number of structural nodes. This process is then repeated for all of the structural nodes. This method of evaluating the linearization becomes quickly infeasible due to the large number of structural nodes and the high cost of evaluating the fluid residual for a three-dimensional case at high spatial orders. To approximate the derivative in a more efficient manner, a modal representation of the displacement is used. The modal representation can approximate the displacement over the structure using a fixed and finite number of modes in an accurate fashion. Consider the displacement of the structure at the various nodes of the structural mesh

$$\mathbf{U}^s = [U^{s_1}, U^{s_2}, U^{s_3}...., U^{s_n}] \tag{A.10}$$

where U^{s_i} represents the z displacement for the i^{th} structural node. Using the first k eigenmodes of the structures, Φ^j, the displacements over the plate can be approximated as,

$$U^{s_x} \approx \sum_{j=0}^{k} \mathrm{A}^j \Phi^j(s_x) \tag{A.11}$$

where, s_x is any point on the structure and A is the coefficient of the various eigenmodes, evaluated by making the modal representation exact at the structural nodes.

$$\mathbf{A} = \mathbf{\Phi}^{-1}\mathbf{U}^s \tag{A.12}$$

Substituting Eq A.11 into Eq A.8, the derivative can be rewritten as,

$$\mathbf{r}^f = \mathbf{M}\mathbf{U}^s = \mathbf{M}\mathbf{\Phi}\mathbf{A} \tag{A.13}$$

Linearizing the fluid residual with respect to the amplitudes of the various modes approximates the actual derivative, $\mathbf{M}$

$$\frac{\partial \mathbf{r}^f}{\partial \mathbf{A}} = \mathbf{M}\mathbf{\Phi}. \tag{A.14}$$

The linearization of the fluid residual with respect to the amplitudes of the various modes is evaluated using finite differences. This method requires reevaluating the fluid residual for only k modes and is therefore more efficient. Solving for the linear system shown in Eq A.14 provides an approximation of the derivative. In this work, $k = 9$ modes have been used. For these number of modes the error in the displacement stayed below a threshold ($10^{-4}m$) throughout the simulation and increasing the number of modes did not result in a significant change in the derivative value.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Future computer requirements for computational aerodynamics. In *NASA Conference Proceedings*, June 1968.

[2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[3] A. Abas and R. Abdul-Rahman. Adaptive FEM with domain decomposition method for partitioned-based fluid–structure interaction. *Arabian Journal for Science and Engineering*, 41(2):611–622, February 2016.

[4] A. Abdelkefi. Aeroelastic energy harvesting: A review. *International Journal of Engineering Science*, 100:112–135, March 2016.

[5] F. Afonso, J. Vale, É. Oliveira, F. Lau, and A. Suleman. A review on non-linear aeroelasticity of high aspect-ratio wings. *Progress in Aerospace Sciences*, 89:40–57, February 2017.

[6] K. Ahuja, B. Endtmayer, M. Steinbach, and T. Wick. Multigoal-oriented error estimation and mesh adaptivity for fluid-structure interaction. *arXiv preprint arXiv:2108.05654*, April 2021.

[7] S. R. Allmaras and F. T. Johnson. Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. In *Seventh international conference on computational fluid dynamics (ICCFD7)*, volume 1902. Big Island, HI, July 2012.

[8] E. Aulisa, S. Bna, and G. Bornia. A monolithic ALE Newton-Krylov solver with Multigrid-Richardson–Schwarz preconditioning for incompressible fluid-structure interaction. *Computers & Fluids*, 174:213–228, September 2018.

[9] H. Bahar and A. Bahar. A force analogy method (FAM) assessment on different static condensation procedures for frames with full Rayleigh damping. *The Structural Design of Tall and Special Buildings*, 27(9):e1468, June 2018.

[10] J. Barche et al. Experimental data base for computer program assessment. *AGARD-Report AGARD-AR-138*, pages 2002–0843, May 1979.

[11] F. Bassi and S. Rebay. Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier–Stokes equations. *International journal for numerical methods in fluids*, 40(1-2):197–207, September 2002.

[12] K. Bathe and W. Hahn. On transient analysis of fluid-structure systems. *Computers & Structures*, 10(1-2):383–391, April 1979.

[13] J. T. Batina. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA journal*, 28(8):1381–1388, August 1990.

[14] R. Becker and R. Rannacher. *A feed-back approach to error control in finite element methods: Basic analysis and examples*. Citeseer, January 1996.

[15] A. Beckert and H. Wendland. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology*, 5(2):125–134, February 2001.

[16] T. Belytschko. Fluid-structure interaction. *Computers & Structures*, 12(4):459–469, October 1980.

[17] H. Bijl, M. H. Carpenter, V. N. Vatsa, and C. A. Kennedy. Implicit time integration schemes for the unsteady compressible Navier–Stokes equations: laminar flow. *Journal of Computational Physics*, 179(1):313–329, June 2002.

[18] R. L. Bisplinghoff, H. Ashley, and R. L. Halfman. *Aeroelasticity*. Courier Corporation, June 2013.

[19] M. D. Buhmann. *Radial basis functions: theory and implementations*, volume 12. Cambridge university press, 2003.

[20] J. Butcher. Runge-Kutta methods. *Scholarpedia*, 2(9):3147, September 2007.

[21] P. Causin, J.-F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. *Computer methods in applied mechanics and engineering*, 194(42-44):4506–4527, October 2005.

[22] M. Ceze and K. J. Fidkowski. Anisotropic hp-adaptation framework for functional prediction. *AIAA journal*, 51(2):492–509, February 2013.

[23] M. Ceze and K. J. Fidkowski. Drag prediction using adaptive discontinuous finite elements. *Journal of Aircraft*, 51(4):1284–1294, July 2014.

[24] M. A. Ceze and K. J. Fidkowski. High-order output-based adaptive simulations of turbulent flow in two dimensions. *AIAA Journal*, 54(9):2611–2625, September 2016.

[25] G. Chen and K. J. Fidkowski. Output-based adaptive aerodynamic simulations using convolutional neural networks. *Computers & Fluids*, 223, April 2021.

[26] A. Collar. The expanding domain of aeroelasticity. *The Aeronautical Journal*, 50(428):613–636, August 1946.

[27] A. Collar. The first fifty years of aeroelasticity. *Aerospace (Royal Aeronautical Society Journal)*, 5(2):12–20, February 1978.

[28] M. A. de Barros Ceze. *A robust hp-adaptation method for discontinuous Galerkin discretizations applied to aerodynamic flows*. PhD thesis, University of Michigan, 2013.

[29] A. De Boer, M. S. Van der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers & structures*, 85(11-14):784–795, June 2007.

[30] A. de Boer, A. H. van Zuijlen, and H. Bijl. Review of coupling methods for non-matching meshes. *Computer methods in applied mechanics and engineering*, 196(8):1515–1525, January 2007.

[31] A. de Boer, A. H. van Zuijlen, and H. Bijl. Comparison of conservative and consistent approaches for the coupling of non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50):4284–4297, September 2008.

[32] J. Degroote. Partitioned simulation of fluid-structure interaction. *Archives of computational methods in engineering*, 20(3):185–238, September 2013.

[33] K. Doetsch and K. J. Fidkowski. Combined entropy and output-based adjoint approach for mesh refinement and error estimation. *AIAA Journal*, 57(8), August 2019.

[34] J. Donea, S. Giuliani, and J.-P. Halleux. An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *Computer methods in applied mechanics and engineering*, 33(1-3):689–723, September 1982.

[35] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian Methods. *Encyclopedia of computational mechanics*, November 2004.

[36] T. Dunne. An Eulerian approach to fluid–structure interaction and goal-oriented mesh adaptation. *International journal for numerical methods in fluids*, 51(9-10):1017–1039, July 2006.

[37] C. Farhat, P. Geuzaine, and G. Brown. Application of a three-field nonlinear fluid–structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter. *Computers & Fluids*, 32(1):3–29, January 2003.

[38] C. Farhat and M. Lesoinne. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer methods in applied mechanics and engineering*, 182(3-4):499–515, February 2000.

[39] C. Farhat, K. G. Van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer methods in applied mechanics and engineering*, 195(17-18):1973–2001, March 2006.

[40] C. A. Felippa, K.-C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer methods in applied mechanics and engineering*, 190(24-25):3247–3270, March 2001.

[41] K. Fidkowski. A local sampling approach to anisotropic metric-based mesh optimization. In *54th AIAA Aerospace Sciences Meeting*, page 0835, 2016.

[42] K. Fidkowski. Three-dimensional benchmark RANS computations using discontinuous finite elements on solution-adapted meshes. In *2018 AIAA Aerospace Sciences Meeting*, page 1104, 2018.

[43] K. J. Fidkowski. Output-based space–time mesh optimization for unsteady flows using continuous-in-time adjoints. *Journal of Computational Physics*, 341:258–277, July 2017.

[44] K. J. Fidkowski and G. Chen. Metric-based, goal-oriented mesh adaptation using machine learning. *Journal of Computational Physics*, 426, February 2021.

[45] K. J. Fidkowski and D. L. Darmofal. A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 225(2):1653–1672, August 2007.

[46] K. J. Fidkowski and D. L. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA journal*, 49(4):673–694, April 2011.

[47] K. J. Fidkowski and Y. Luo. Output-based space–time mesh adaptation for the compressible Navier-Stokes equations. *Journal of Computational Physics*, 230(14):5753–5773, June 2011.

[48] K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal. p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 207(1):92–113, July 2005.

[49] K. J. Fidkowski and P. L. Roe. An entropy adjoint approach to mesh refinement. *SIAM Journal on Scientific Computing*, 32(3):1261–1287, 2010.

[50] B. Froehle and P.-O. Persson. A high-order discontinuous Galerkin method for fluid–structure interaction with efficient implicit–explicit time stepping. *Journal of Computational Physics*, 272:455–470, September 2014.

[51] K. Fujii. Progress and future prospects of CFD in aerospace—Wind tunnel and beyond. *Progress in Aerospace Sciences*, 41(6):455–470, August 2005.

[52] Y. C. Fung. *An introduction to the theory of aeroelasticity*. Courier Dover Publications, 2008.

[53] E. Gauci, F. Alauzet, A. Loseille, and A. Dervieux. Towards goal-oriented mesh adaptation for fluid-structure interaction. In *VI International Conference on Computational Methods for Coupled Problems in Science and Engineering COUPLED PROBLEMS 2015*, March 2015.

[54] R. Glowinski, T.-W. Pan, and J. Periaux. A fictitious domain method for dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering*, 111(3-4):283–303, January 1994.

[55] R. E. Gordnier. High fidelity computational simulation of a membrane wing airfoil. *Journal of Fluids and Structures*, 25(5):897–917, July 2009.

[56] R. E. Gordnier and R. B. Melville. Transonic flutter simulations using an implicit aeroelastic solver. *Journal of Aircraft*, 37(5):872–879, September 2000.

[57] B. E. Griffith, X. Luo, D. M. McQueen, and C. S. Peskin. Simulating the fluid dynamics of natural and prosthetic heart valves using the immersed boundary method. *International Journal of Applied Mechanics*, 1(01):137–177, August 2009.

[58] G. P. Guruswamy. A review of numerical fluids/structures interface methods for computations using high-fidelity equations. *Computers & structures*, 80(1):31–41, January 2002.

[59] R. J. Guyan. Reduction of stiffness and mass matrices. *AIAA journal*, 3(2):380–380, February 1965.

[60] R. Hartmann and P. Houston. Adaptive Discontinuous Galerkin Finite Element Methods for the Compressible Euler Equations. *Journal of Computational Physics*, 183(2):508–532, December 2002.

[61] F. Hecht. BAMG: bidimensional anisotropic mesh generator. *User Guide. INRIA, Rocquencourt*, 17, October 1998.

[62] G. Hou, J. Wang, and A. Layton. Numerical methods for fluid-structure interaction - a review. *Communications in Computational Physics*, 12(2):337–377, August 2012.

[63] K. Huang, M. Krügener, A. Brown, F. Menhorn, H.-J. Bungartz, and D. Hartmann. Machine learning-based optimal mesh generation in computational fluid dynamics. *arXiv preprint arXiv:2102.12923*, February 2021.

[64] K. Kasim, A. Muley, M. Stoia, and F. Ladeinde. Advanced heat transfer devices for aerospace applications. In *ASME International Mechanical Engineering Congress and Exposition*, volume 58431, page V008T10A027. American Society of Mechanical Engineers, November 2017.

[65] S. M. Kast and K. J. Fidkowski. Output-based mesh adaptation for high order Navier-Stokes simulations on deformable domains. *Journal of Computational Physics*, 252:468–494, November 2013.

[66] C. A. Kennedy and M. H. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Applied numerical mathematics*, 44(1-2):139–181, January 2003.

[67] G. K. Kenway, C. A. Mader, P. He, and J. R. Martins. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, 110:100542, October 2019.

[68] W. Kim and H. Choi. Immersed boundary methods for fluid-structure interaction: A review. *International Journal of Heat and Fluid Flow*, 75:301–309, February 2019.

[69] A. La Spina, M. Kronbichler, M. Giacomini, W. A. Wall, and A. Huerta. A weakly compressible hybridizable discontinuous Galerkin formulation for fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 372:113392, December 2020.

[70] U. Langer and H. Yang. Recent development of robust monolithic fluid-structure interaction solvers. In *Fluid-Structure Interaction*, pages 169–192. De Gruyter, 2017.

[71] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer methods in applied mechanics and engineering*, 134(1-2):71–90, July 1996.

[72] S. Li and W. K. Liu. *Meshfree particle methods*. Springer Science & Business Media, March 2007.

[73] M. Lombardi, N. Parolini, and A. Quarteroni. Radial basis functions for inter-grid interpolation and mesh motion in FSI problems. *Computer Methods in Applied Mechanics and Engineering*, 256:117–131, April 2013.

[74] A. Loseille, A. Dervieux, and F. Alauzet. Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations. *Journal of computational physics*, 229(8):2866–2897, April 2010.

[75] A. E. H. Love. XVI. The small free vibrations and deformation of a thin elastic shell. *Philosophical Transactions of the Royal Society of London.(A.)*, (179):491–546, 1888.

[76] J. N. Lyness. Numerical algorithms based on the theory of complex variable. In *Proceedings of the 1967 22nd national conference*, pages 125–133, January 1967.

[77] J. N. Lyness and C. B. Moler. Numerical differentiation of analytic functions. *SIAM Journal on Numerical Analysis*, 4(2):202–210, December 1967.

[78] L. Manevitz, A. Bitar, and D. Givoli. Neural network time series forecasting of finite-element mesh adaptation. *Neurocomputing*, 63:447–463, January 2005.

[79] A. Masud, M. Bhanabhagvanwala, and R. A. Khurram. An adaptive mesh rezoning scheme for moving boundary flows and fluid–structure interaction. *Computers & fluids*, 36(1):77–91, January 2007.

[80] F. Mazhar, A. Javed, J. T. Xing, A. Shahzad, M. Mansoor, A. Maqsood, S. I. A. Shah, and K. Asim. On the meshfree particle methods for fluid-structure interaction problems. *Engineering Analysis with Boundary Elements*, 124:14–40, March 2021.

[81] L. Meirovitch. *Computational methods in structural dynamics*, volume 5. Springer Science & Business Media, October 1980.

[82] C. Michler, S. Hulshoff, E. Van Brummelen, and R. De Borst. A monolithic approach to fluid–structure interaction. *Computers & fluids*, 33(5-6):839–848, June 2004.

[83] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, January 2005.

[84] J. J. Monaghan. An introduction to SPH. *Computer physics communications*, 48(1):89–96, January 1988.

[85] K. R. Moyle and Y. Ventikos. Local remeshing for large amplitude grid deformations. *Journal of Computational Physics*, 227(5):2781–2793, February 2008.

[86] S. Nadarajah and A. Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. In *38th Aerospace Sciences Meeting and Exhibit*. AIAA Paper 2000-667, January 2000.

[87] M. Nastran. Reference Manual, MSC. *Software Corporation*, 2004.

[88] U. Naumann. *The art of differentiating computer programs: an introduction to algorithmic differentiation*. SIAM, January 2011.

[89] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational mechanics*, 10(5):307–318, September 1992.

[90] M. F. Nelson. The use of condensation techniques for solving dynamics problems. *SAE Transactions*, pages 1435–1444, January 1974.

[91] M. Nemec and M. Aftosmis. Adjoint error estimation and adaptive refinement for embedded-boundary cartesian meshes. In *18th AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2007-4187, June 2007.

[92] M. Nemec, M. Aftosmis, and M. Wintzer. Adjoint-based adaptive mesh refinement for complex geometries. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, page 725, 2008.

[93] J. T. Oden and S. Prudhomme. Goal-oriented error estimation and adaptivity for the finite element method. *Computers & mathematics with applications*, 41(5-6):735–756, April 2001.

[94] V. Ojha, K. Fidkowski, and C. E. S. Cesnik. High-fidelity coupled fluid-structure interaction simulations with adaptive meshing. In *AIAA Aviation 2019 Forum*, page 3056, 2019.

[95] E. Oñate, S. Idelsohn, O. Zienkiewicz, and R. Taylor. A finite point method in computational mechanics: Applications to convective transport and fluid flow. *International journal for numerical methods in engineering*, 39(22):3839–3866, November 1996.

[96] M. A. Park. Adjoint-based, three-dimensional error prediction and grid adaptation. *AIAA Journal*, 42(9):1854–1862, September 2004.

[97] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *International Journal of computer vision*, 66(1):41–66, January 2006.

[98] P.-O. Persson, J. Bonet, and J. Peraire. Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains. *Computer Methods in Applied Mechanics and Engineering*, 198(17-20):1585–1595, April 2009.

[99] P.-O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, page 112, January 2006.

[100] P.-O. Persson, J. Peraire, and J. Bonet. A high order discontinuous Galerkin method for fluid-structure interaction. In *18th AIAA Computational Fluid Dynamics Conference*, page 4327, September 2007.

[101] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aeroelastic problems Part I: Model problem, theory and two-dimensional application. *Computer methods in applied mechanics and engineering*, 124(1-2):79–112, June 1995.

[102] J. J. Quirk. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. *Computers & fluids*, 23(1):125–142, January 1994.

[103] T. C. Rendall and C. B. Allen. Unified fluid–structure interpolation and mesh motion using radial basis functions. *International journal for numerical methods in engineering*, 74(10):1519–1559, June 2008.

[104] T. Richter. Goal-oriented error estimation for fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 223:28–42, June 2012.

[105] T. Richter. A monolithic geometric multigrid solver for fluid-structure interactions in ALE formulation. *International journal for numerical methods in engineering*, 104(5):372–390, November 2015.

[106] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, October 1981.

[107] R. Sanchez, H. Kline, D. Thomas, A. Variyar, M. Righi, T. D. Economon, J. J. Alonso, R. Palacios, G. Dimitriadis, and V. Terrapon. Assessment of the fluid-structure interaction capabilities for aeronautical applications of the open-source solver SU2. 2016.

[108] P. P. Sarkar, N. P. Jones, and R. H. Scanlan. Identification of aeroelastic parameters of flexible bridges. *Journal of Engineering Mechanics*, 120(8):1718–1742, August 1994.

[109] R. H. Scanlan and N. P. Jones. Aeroelastic analysis of cable-stayed bridges. *Journal of Structural Engineering*, 116(2):279–297, February 1990.

[110] R. H. Scanlan and R. Rosenbaum. *Introduction to the study of Aircraft Vibration and Flutter*. Macmillan, 1951.

[111] M. Selim and R. Koomullil. Mesh deformation approaches–a survey. *Journal of Physical Mathematics*, 7(2):1–9, January 2016.

[112] J. P. Sheldon, S. T. Miller, and J. S. Pitt. A hybridizable discontinuous Galerkin method for modeling fluid–structure interaction. *Journal of Computational Physics*, 326:91–114, December 2016.

[113] M. J. Smith, D. H. Hodges, and C. E. S. Cesnik. Evaluation of computational algorithms suitable for fluid-structure interactions. *Journal of Aircraft*, 37(2):282–294, March 2000.

[114] P. Šolın and L. Demkowicz. Goal-oriented hp-adaptivity for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 193(6-8):449–468, February 2004.

[115] V. Sousa, M. De M Anicézio, C. De Marqui Jr, and A. Erturk. Enhanced aeroelastic energy harvesting by exploiting combined nonlinearities: theory and experiment. *Smart Materials and Structures*, 20(9):094007, August 2011.

[116] C. K. Tam and J. C. Webb. Dispersion-relation-preserving finite difference schemes for computational acoustics. *Journal of computational physics*, 107(2):262–281, August 1993.

[117] T. E. Tezduyar and S. Sathe. Modelling of fluid-structure interactions with the space-time finite elements: solution techniques. *International Journal for Numerical Methods in Fluids*, 54(6-8):855–900, June 2007.

[118] E. Torenbeek. *Advanced aircraft design: conceptual design, analysis and optimization of subsonic civil airplanes*. John Wiley & Sons, May 2013.

[119] E. H. van Brummelen. Added mass effects of compressible and incompressible flows in fluid-structure interaction. January 2009.

[120] K. G. van der Zee, E. H. van Brummelen, I. Akkerman, and R. de Borst. Goal-oriented error estimation and adaptivity for fluid–structure interaction using exact linearized adjoints. *Computer Methods in Applied Mechanics and Engineering*, 200(37-40):2738–2757, September 2011.

[121] R. Van Loon, P. D. Anderson, F. P. Baaijens, and F. N. Van de Vosse. A three-dimensional fluid-structure interaction method for heart valve modelling. *Comptes Rendus Mecanique*, 333(12):856–866, December 2005.

[122] A. van Zuijlen and H. Bijl. A higher-order time integration algorithm for the simulation of nonlinear fluid–structure interaction. *Nonlinear Analysis: Theory, Methods & Applications*, 63(5-7):e1597–e1605, November 2005.

[123] A. Van Zuijlen, A. de Boer, and H. Bijl. Higher-order time integration through smooth mesh deformation for 3D fluid-structure interaction simulations. *Journal of Computational Physics*, 224(1):414–430, May 2007.

[124] M. Vanella, P. Rabenold, and E. Balaras. A direct-forcing embedded-boundary method with adaptive mesh refinement for fluid–structure interaction problems. *Journal of Computational Physics*, 229(18):6427–6449, September 2010.

[125] D. A. Venditti and D. L. Darmofal. Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flows. *Journal of Computational Physics*, 176(1):40–69, February 2002.

[126] D. A. Venditti and D. L. Darmofal. Grid adaptation for functional outputs: application to two-dimensional inviscid flows. *Journal of Computational Physics*, 176(1):40–69, February 2002.

[127] L. Wang and D. J. Mavriplis. Adjoint-based h–p adaptive discontinuous Galerkin methods for the 2D compressible Euler equations. *Journal of Computational Physics*, 228(20):7643–7661, November 2009.

[128] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, et al. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, July 2013.

[129] F. M. White and J. Majdalani. *Viscous fluid flow*, volume 3. McGraw-Hill New York, 2006.

[130] T. Wick. *Adaptive finite element simulation of fluid-structure interaction with application to heart-valve dynamics*. PhD thesis, 2011.

[131] J. Witteveen. Explicit and robust inverse distance weighting mesh deformation for CFD. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 165, January 2010.

[132] M. Yano et al. *An optimization framework for adaptive higher-order discretizations of partial differential equations on anisotropic simplex meshes*. PhD thesis, Massachusetts Institute of Technology, 2012.

[133] P. Zhang and S. Huang. Review of aeroelasticity for wind turbine: Current status, research focus and future perspectives. *Frontiers in Energy*, 5(4):419–434, December 2011.

[134] Z. Zhang, Y. Wang, P. K. Jimack, and H. Wang. MeshingNet: A new mesh generation method based on deep learning. In *Lecture Notes in Computer Science*, pages 186–198. Springer International Publishing, 2020.