AN HP-ADAPTIVE PETROV-GALERKIN METHOD FOR STEADY-STATE AND UNSTEADY FLOW PROBLEMS

By

Behzad Reza Ahrabi

W. Kyle Anderson
Professor of Computational Engineering
(Chair)

James C. Newman III
Professor of Computational Engineering
(Committee Member)

Li Wang
Research Assistant Professor of
Computational Engineering
(Committee Member)

John V. Matthews, III
Associate Professor of Mathematics
(Committee Member)

AN HP-ADAPTIVE PETROV-GALERKIN METHOD FOR STEADY-STATE AND UNSTEADY FLOW PROBLEMS

By

Behzad Reza Ahrabi

A Dissertation Submitted to the Faculty of the University
of Tennessee at Chattanooga in Partial Fulfillment of
the Requirements of the Degree of Doctor of
Philosophy in Computational Engineering

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

August 2015

ABSTRACT

After several decades of development, higher-order finite-element methods are now being considered for realistic and large scale Computational Fluid Dynamics (CFD) simulations. This necessitates further studies on utilization of mesh adaptation techniques in order to reach reliable solutions at minimal computation cost. In this study, adaptation capabilities have been developed within a Petrov-Galerkin (PG) finite-element method. The mesh modification mechanisms include h-, p-, and combined hp-adaptations which are performed in a non-conforming manner. The constrained approximation method has been utilized in order to retain the continuity of the solution space in presence of hanging nodes. Hierarchical basis functions have been employed to facilitate the implementation of the constrained approximation method. The adaptive methodology has been demonstrated on numerous cases using the Euler and Reynolds Average Navier-Stokes (RANS) equations, equipped with a modified Spalart-Allmaras (SA) turbulence model. Also, a PDE-based artificial viscosity has been added to the governing equations, to stabilize the solution in the vicinity of shock waves. For accurate representation of the geometric surfaces, high-order curved boundary meshes have been generated and the interior meshes have been deformed through the solution of a modified linear elasticity equation. A fully implicit linearization has been utilized within a Newton-type algorithm to advance each iteration or

time-step, for steady-state or unsteady simulations, respectively. In order to navigate the adaptation process, adjoint-based and feature-based techniques have been employed in the steady-state and unsteady problems, respectively. It was shown that weak implementation of the boundary conditions and the use of a modified functional are required to obtain a smooth adjoint solution where Dirichlet Boundary conditions are imposed. Failure to utilize both results in a non-smooth adjoint solution. To accelerate the error reduction, an enhanced h-refinement has been used in the vicinity of singularity points. Several numerical results illustrate consistent accuracy improvement of the functional outputs and capability enhancements in resolving complex viscous flow features such as shock boundary layer interaction, flow separation, and vortex shedding.

DEDICATION

This work is dedicated to my loving and supportive wife, Elaheh, and to our beautiful daughter, Roshana.

ACKNOWLEDGEMENTS

First and foremost, I express my sincere gratitude to my first advisor, Professor W. Kyle Anderson. Kyle, I wish I could describe in words how grateful I am to be your student. You are not only my advisor. “Cooking with gas” forever...

Next, very special thanks go to my second advisor Professor James C. Newman. Jim, without your help, motivation, encouragement, and support, I could not finish this research. Thank you for everything.

Besides my advisors, I would like to thank the rest of my thesis committee: Dr. Li Wang and Dr. John V. Matthews for sharing their experience with me and their great attention to the details during this research.

Last but not the least, I would like to thank my wife, Elaheh. Ela, I could not even imagine how supportive you are. From bottom of my heart, I thank you for helping me to pursue what I love besides you and our daughter.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

I.1 Automatic Adaptation

Most of the physical phenomena in the world around us are governed by Partial Differential Equations (PDEs) in the form of Initial Boundary Value Problems (IBVPs). Countless examples of fluid dynamics, solid mechanics, and electromagnetics are well-known cases of these type of problems. Today, there is no doubt that numerical methods have been very successful to tackle complex PDEs which are well-guarded if one tries to face them with classical mathematical methods. The key to this success is the basic idea to discretize the given continuous PDE to obtain a system of equations with finite number of unknowns which can be solved using a computer [1]. To apply the discretization, the given domain of interest is represented by a computational domain which is typically partitioned into sub-domains or elements. This partitioning is known as computational mesh and it includes the location of unknowns. Then an approximate solution is sought on the computational mesh. This is in contrast to classical methods that seek an exact solution to the PDE over the domain of interest. For a consistent discretization method, as the number of unknowns increases, the discretization error decreases, and the approximate solution converges to the exact solution to the PDE. However, there should be a tradeoff between the computation

cost and the required accuracy. Fortunately, in many real-world applications, the exact solution is not needed and instead, the solution only needs to be accurate-enough. Immediately, two fundamental questions should be answered. First, what is called an accurate-enough solution? Second, how to obtain an accurate-enough solution?

In order to answer the first question, a clear definition of the error is required. In the classical sense, any difference between the approximate solution and the exact solution to the PDE over the entire domain of interest is considered as error. Clearly, the exact solution can be utilized to obtain any desirable output or study any phenomenon which is governed by the PDE. However, in real-word applications, the computations are performed for a limited number of objectives and thus the above mentioned definition of the error can be replaced with those that target the objectives of the computation. Therefore a solution can be called accurate-enough if the error in the computed objective of interest is less than a prescribed value. Depending on the application, other definitions may serve equally. In any case, the key is that such an approach avoids the waste of computational resources for unnecessary resolutions and so it drastically reduces the required computational cost.

To answer the second question, one needs to realize the possible sources of the error. In general, the accuracy of a numerical solution depends on the discretization method, and also on the quality of the computational mesh. In engineering applications, the quality of a mesh should be simply assessed by its capability to obtain an accurate solution using the given discretization method. It is well known that the shape of the elements and their density distribution are the key factors in this regard. To follow this topic, it is useful to define the term of the *optimal mesh*. Different researchers may have different definitions for

this term. For example, Baserinia [2] calls a mesh optimal if it minimizes the discretization error for a prescribed number of elements. However, in this study, a mesh is considered as optimal if it minimizes the computational cost associated with the chosen discretization method to reach the least accurate-enough solution. Here it should be emphasized that we have related the quality of the mesh to the accuracy of the solution which in turn depends on the computation's objective. This implies that different simulation objectives for the same geometry may have different optimal meshes. The careful reader will notice that, by choosing the discretization method, the second question will change to how to obtain an optimal mesh?

Generally, the mesh generation is a pre-processing step and it is usually done without the knowledge of the exact solution. Even if the exact solution is known, each discretization method demands exclusive requirements for a high quality mesh. In simple words, it is not wrong to say that it is impossible, even for experts, to generate an optimal mesh manually. Therefore an automated algorithm is needed for this purpose. Such algorithms usually start with an initial mesh and after obtaining the solution on that mesh, an *a posteriori* error analysis is performed to determine which areas of mesh need modifications to reach an optimal mesh. This process will be repeated until an accurate-enough solution is obtained. Such algorithms are known as automatic mesh adaptation.

Based on above discussion, automatic adaptation is a necessity for all mesh-based discretization methods. However, each discretization method requires special considerations to be utilized within an adaptive algorithm.

I.2 High-Order CFD Methods

Currently, second-order finite-volume methods are the dominant methodologies in Computational Fluid Dynamics (CFD). However, after several decades of development, finite-element simulations over complex geometries can now be performed for both time-dependent and steady state applications. Although much less mature and not yet widely adopted, finite-element methodologies offer some potentially very distinct advantages over finite-volume counterparts (for example, see reference [3]). Many of these advantages stem from the fact that finite-element methods use discretization stencils that only require immediately adjacent nodes, whereas finite-volume algorithms inevitably require much larger stencils. One major benefit of the compact stencil is that higher order discretization in both space and time is possible through a clear and well-defined path without complications associated with the larger stencil in finite-volume methods. The compact stencil also presents significant advantages over finite-volume methods because an accurate linearization of the nonlinear residual is easily obtained, thereby enabling very significant benefits in the development of algorithms modeled after Newton's method and for sensitivity analysis.

In addition to the algorithmic advantages previously mentioned, the development of a comprehensive finite-element fluid-dynamic simulation capability provides several benefits for high-fidelity physical modeling. Many simulations require long-distance tracking of critical features, such as vortices, to determine the effects caused by their impingement on other geometries or structures. Flow over a helicopter is a well-known example of this type. In such cases, the use of second-order finite-volume methods requires excessively refined

meshes to resolve these features. The development of adaptive finite-element methodology addresses these difficulties in two ways. First, high-order methods simply resolve small features for much longer distances than second-order methods without dissipating their strength. For example, Anderson et al. [4] have shown an example of turbulent flow simulation in which the finite-element solution shows significantly less dissipation when compared to finite-volume solution, even if both schemes use second order accuracy for spatial discretization. Secondly, adaptive finite-element methods are more capable than finite-volume counterparts. The major reason is that in addition to local mesh refinements, significant benefits can be obtained by local enrichment of the order of accuracy. Higher-order discretization has lower truncation error, and at the same time, requires less work than that required for local spatial refinement. However, due to the large stencils, pursuing high-order adaptive grid capability with finite-volume schemes is cumbersome at best and is functionally impractical.

I.3 Motivation

Among higher order finite-element methods being developed for compressible flow problems, the Discontinuous-Galerkin (DG) schemes [5-25] have been the most utilized. However, stabilized Petrov-Galerkin (PG) schemes [25-42] are increasingly absorbing more attention from researchers in the higher order CFD community. A major difference between these schemes is that in a DG scheme, each element has its exclusive set of nodes and the solution can be discontinuous between adjacent elements whereas in a PG scheme, some nodes are shared between adjacent elements and the solution is continuous all over the

computational domain. As a result, for elements with lower polynomial degrees, a PG scheme requires significantly less Degrees of Freedom (DOFs) and non-zero matrix entries than a DG scheme for comparable accuracy [25, 38, 39, 41]. This difference is well appreciated by noting that for linear and quadratic tetrahedral elements, a DG scheme may require an order of magnitude more computational resources than a PG scheme [40]. This advantage can be further enhanced using adaptation techniques. Although in the higher order CFD community, a considerable amount of research has been conducted on adaptation techniques for DG schemes [8, 9, 12, 16, 19-21, 23, 24, 43-50], the methodology for PG schemes is still not rigorously established and this provides the principal motivation behind this dissertation. In particular, this work aims to add several adaptation capabilities to a general framework, denoted as FUNSAFE (Fully UNStructured Adaptive Finite Elements). This framework uses PG formulation for discretization and it is capable for a wide range of applications including fluid dynamics, electromagnetics, and structural analysis [25, 38, 40, 51-53]. The adaptive method should be equally applicable for all of the mentioned applications and in the same time, it should impose the least amount of effort for implementation. This is particularly beneficial for multidisciplinary applications. Also, the adaptation capability is intended to be utilized in both steady-state and unsteady applications. Note that for the unsteady applications, derefinement is crucial as the flow features travel in the computational domain and without a derefinement mechanism, the computation will be hindered by an over-refined mesh.

I.4 Scope

Adaptive higher order methods cover a broad area of research. In this section, the particular objectives of this study have been described and the extents of the framework to achieve these objectives have been determined. Although dynamic adaptation has been considered for the unsteady problems, the main emphasis has been put on the steady-state problems. In particular, a robust adaptive methodology has been sought such that it can be effectively employed in a wide range of compressible flow applications, including subsonic and transonic flows. To this end, numerous components within flow solver and adaption algorithm must work in harmony.

A common component of any adaptation algorithm is an error analysis mechanism which identifies the regions of the computational mesh that need refinement. Feature-based error indicators [45, 54-59] and output-based error estimators [19-21, 23, 48, 49, 55, 60-73] are two well-known alternatives for such a mechanism.

Feature-based methods aim to capture regions with distinguishing flow features such as shock waves, shear layers, vortices, and singularities. For this purpose, the gradients of the flow variables are typically used as error indicators. These methods have an *ad hoc* nature and generally are not considered as reliable adaptation techniques. Nevertheless, due to the simplicity and cost efficiency, they are extensively used, particularly in unsteady applications where frequent adaptations are required to trace an evolving phenomenon. With the same incentive, in this study, feature-based methods have been employed in unsteady problems.

Output-based (or adjoint-based) methods are particularly useful when a specific functional output, usually defined in an integral form, is the main objective of the simulation. Lift and drag coefficients are familiar examples in aeronautical applications. In such cases, output-based methods perhaps offer the most reliable option to navigate the adaptation algorithm as they target the chosen functional output and try to adapt the mesh such that a prescribed precision is ensured. For this purpose, the sensitivity of the functional with respect to the local residual of the flow (or primal) solution is calculated in the form of an adjoint (or dual) solution. This approach provides an estimation of local errors that directly contribute to the global error of the desired functional. Thus, regions with highest local errors are chosen for mesh refinement. Despite the benefits, applications of output-based methods have been mostly limited to the steady-state problems. This is mainly due to the computational costs and implementation complexities for the unsteady problems. Therefore, in the present work, the output-based method has been employed only for steady-state problems. Examples of output-based adaptation for unsteady problems can be found in references [66, 70-72].

In order to develop an effective output-based adaptation, the quality of the adjoint solution is of critical importance. In particular, for viscous flows, due to the presence of Dirichlet boundary conditions on the no-slip walls, special attention should be given to the implementation of the boundary conditions. In the present work, the spatial discretization is based on a Streamline-Upwind Petrov-Galerkin (SUPG) scheme [26, 27, 36]. To obtain a smooth adjoint solution near the no-slip walls, the boundary conditions have been imposed weakly [74-76]. Also, the definition of the functional outputs have been modified to be

compatible with these boundary conditions. The implementation of the boundary conditions is, in essence, based on the Nitsche's method [77] and its particular formulation has been taken from a Symmetric Interior Penalty Galerkin (SIPG) method [15, 21, 25, 78] that is commonly used in the DG discretizations.

A major concern in the solution of transonic and supersonic flows is the need to stabilize the numerical scheme in vicinity of the shock waves. Although several methods exist to address this problem [79-81], present work seeks a method that shows a consistent behavior in an output-based adaptation algorithm. To this end, a PDE-based artificial viscosity [79, 80] has been added to the governing equations.

Another aspect of this study pertains to mesh modification mechanisms. In finite-element context, mesh modifications are usually categorized into h-adaptation, p-adaptation, or hp-adaptation. For h-adaptation, the local refinement is accomplished by subdividing the elements into smaller elements of the same polynomial order. The p-adaptation, on the other hand, is obtained by local change of the polynomial degree of the element's shape functions. These two methods have been shown in figure I.1. While both h-refinement and p-enrichment produce more unknowns, p-enrichment is expected to be more effective in error reduction for the same number of unknowns. However, p-enrichment may be problematic in regions with sharp gradients in the solution, or sharp corners in the geometry. Therefore, the most effective approach is a combined hp-refinement algorithm which utilizes p-enrichment in regions with smooth solution and h-refinement everywhere else. In this study, all three cases of h-, p- and hp-adaptations have been implemented. For

the case of hp-adaptation, the categorization of elements has been done using a smoothness indicator.

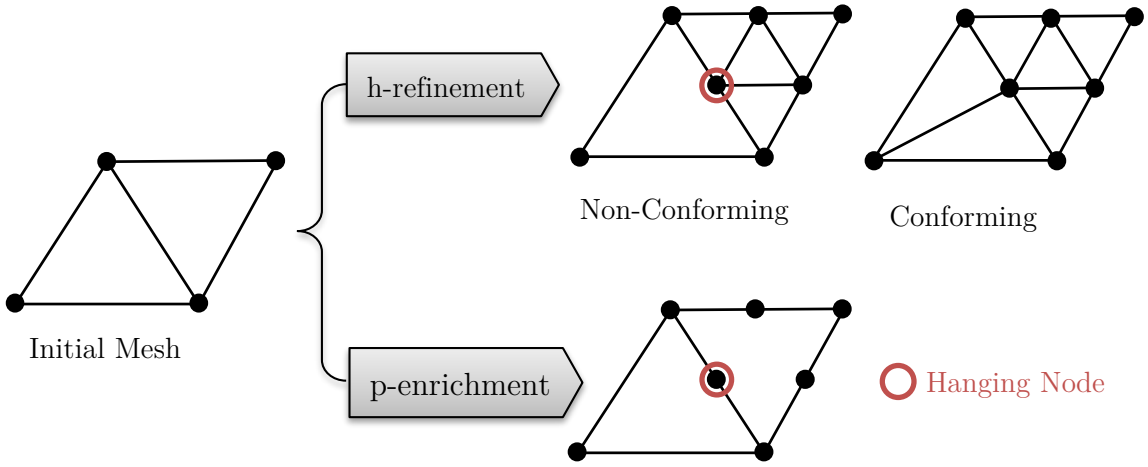


Figure I.1 Example of h- and p-adaptation

Different h-refinement methods can be categorized in several ways. One categorization is non-conforming versus conforming. Figure I.1 also demonstrates an example of each method. As seen in this figure, in non-conforming refinement, hanging nodes are generated. Such a mesh is called *irregular*. Although conforming method does not generate hanging nodes, successful implementation of that may require a great deal of effort, especially for mixed-type elements in three dimensional meshes. As discussed by Remacle et al. [9], non-conforming adaptation offers a simpler implementation for multiple levels of refinement/derefinement. Since one of the goals of the present work is an efficient dynamic hp-adaptation, the speed, versatility, and simplicity offered by a non-conforming method provides enough incentive to choose this method for development. However, as mentioned earlier, in PG schemes, the discrete solution is required to be continuous over the

computational domain and hanging nodes can violate this requirement across the interface between refined and unrefined elements. To address this problem, a technique known as constrained approximation [73, 82-88] has been employed. In this method, a function value at a hanging node is constrained by the function values at adjacent nodes such that a continuous solution is obtained across all element interfaces.

In higher order methods, high order representation of the geometry has a crucial role to reach optimal orders of accuracy. Anderson et al. [38] have shown an example in which linear representation of the surface for quadratic elements causes the order of accuracy to become less than that obtained by linear elements. In this study, to accurately represent the geometries, high-order curved boundary edges have been generated. Since curving the boundary edges can result in collapsed cells when high aspect ratio elements are present in the viscous boundary layers, the interior meshes have been deformed through a linear elasticity solver.

I.5 Overview

The contributions of this dissertation in the FUNSAFE framework include:

- Development of non-conforming dynamic hp-adaptation capabilities for both Lagrange and hierarchical basis functions. Also, elements can be non-uniformly sub- or super-parametric.
- Development of output-based adaptation for steady-state problems, and feature-based adaptation for unsteady problems.

- Development of weakly imposed boundary conditions and modified functional outputs to obtain smooth adjoint solutions in vicinity of Dirichlet-type boundaries.
- Implementation of PDE-based artificial viscosity to enhance the stability of the scheme for the shock capturing purposes.

The remainder of this text describes the methods used to accomplish these tasks. Chapter II presents the governing equations as well as the full system of equations resulting from a SUPG discretization. In chapter III, the solution expansions and the details of constrained approximations have been precisely described. Chapter IV describes the details of the adaptation methodology which includes the adjoint-based error estimation and decision making criteria for the hp-adaptation. Numerical examples are presented in chapter V to demonstrate the ability of the current adaptive methodology. Finally, Section VI offers conclusions and discusses the future work.

CHAPTER II

GOVERNING EQUATIONS AND DISCRETIZATION

II.1 Governing Equations of Fluid Flow

The governing equations consist of the compressible Reynolds Averaged Navier-Stokes (RANS) equations coupled with the one equation negative Spalart-Allmaras (SA) turbulence model. In the conservative form, these equations can be written as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} = \mathbf{S} \quad \text{in } \Omega \tag{II.1}$$

where the bold letters denote vector variables due to multiple equations, and i indexes the spatial dimension. Also, as seen in following, $\overrightarrow{()}$ denotes a vector in n_{sd} spatial dimensions (herein $n_{sd} = 2$). The vector of the conservative flow variables $\mathbf{Q}$, the source term $\mathbf{S}$, and the flux vector $\mathbf{F}_i$ which consists of inviscid and viscous parts, $\mathbf{F}_i^E$ and $\mathbf{F}_i^v$, are given by

$$\mathbf{Q} = \begin{Bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho E \\ \rho \tilde{\nu} \end{Bmatrix}, \qquad \mathbf{S} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ S_T \end{Bmatrix} \tag{II.2}$$

$$\mathbf{F}_i = \mathbf{F}_i^E - \mathbf{F}_i^v \tag{II.3}$$

$$\mathbf{F}_i^E = \begin{Bmatrix} \rho u_i \\ \rho u_1 u_i + \delta_{1i} p \\ \rho u_2 u_i + \delta_{2i} p \\ \rho H u_i \\ \rho u_i \tilde{\nu} \end{Bmatrix}, \mathbf{F}_i^v = \begin{Bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{ij} u_j + \kappa \dfrac{\partial T}{\partial x_i} \\ \dfrac{1}{\sigma} \mu (1 + \psi) \dfrac{\partial \tilde{\nu}}{\partial x_i} \end{Bmatrix}, \quad i = 1,2 \tag{II.4}$$

where ρ is the density, p is the static pressure, u_i is the velocity component in direction of the Cartesian coordinate x_i, E is the specific total energy, $H = E + \frac{p}{\rho}$ is the specific total enthalpy, and δ_{ij} is the Kronecker delta. With the assumption of the perfect gas, the pressure is related to the state variables by the constitutive relation,

$$p = (\gamma - 1) \left(\rho E - \frac{1}{2} \rho u_i u_i \right) \tag{II.5}$$

where γ is the ratio of specific heats and it is set to 1.4. Also, the shear stress tensor is given by

$$\tau_{ij} = (\mu + \mu_T) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \tag{II.6}$$

where μ is the dynamic viscosity which is obtained by the Sutherland's Law and μ_T is the turbulent eddy viscosity. Moreover, κ is the thermal conductivity, and T is the temperature which is related to the specific total energy by

$$\kappa T = \gamma \left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T} \right) \left(E - \frac{1}{2} u_i u_i \right) \tag{II.7}$$

where Pr and Pr_T are Prandtl and turbulent Prandtl numbers which are set to 0.72 and 0.9, respectively.

Also, $\tilde{\nu}$ is the working variable of the SA turbulence model. For more details on turbulence model, the reader is referred to the references [21, 23, 25, 89].

II.2 Spatial Discretization

To start, the *strong form* of the problem is written as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} - \mathbf{S} = \mathbf{0} \qquad \vec{x} \in \Omega \text{ and } t \in [0, \infty) \tag{II.8.a}$$

$$\mathbf{F}_i = \mathbf{F}_i^b \qquad \vec{x} \in \Gamma_F \text{ and } t \in [0, \infty) \tag{II.8.b}$$

$$\mathbf{Q}(\boldsymbol{x}, t) = \mathbf{Q}_D \qquad \vec{x} \in \Gamma_D \text{ and } t \in [0, \infty) \tag{II.8.c}$$

$$\mathbf{Q}(\boldsymbol{x}, 0) = \mathbf{Q}_0(\boldsymbol{x}) \qquad \vec{x} \in \Omega \tag{II.8.d}$$

where, $\Omega \subset \mathbb{R}^{n_{sd}}$ is a bounded domain with Lipschitz-continuous boundary Γ, $\mathbf{F}_i^b$ is the prescribed boundary fluxes through Γ_F portion of the boundary, and $\mathbf{Q}_D$ is the Dirichlet boundary condition on the Γ_D portion of the boundary. By defining the *solution* and *weight* spaces,

$$\boldsymbol{\mathcal{S}}_t := \{\mathbf{Q} |\ \mathbf{Q}(\cdot, t) \in [\mathcal{H}^1(\Omega)]^{n_Q}, t \in [0, \infty) \ \textit{and}\ \mathbf{Q}(\vec{x}, t) = \mathbf{Q}_D \ \textit{on}\ \Gamma_D\} \tag{II.9}$$

$$\mathcal{W} := \ \{w |\ w \in \mathcal{H}^1(\Omega) \ \textit{and}\ w(\vec{x}) = 0 \ \textit{on}\ \Gamma_D\} \tag{II.10}$$

the *weak form* of the problem is expressed as: for any $t \in [0, \infty)$ find $\mathbf{Q} \in \boldsymbol{\mathcal{S}}_t$ such that for all $w \in \mathcal{W}$,

$$\iint_\Omega w \frac{\partial \mathbf{Q}}{\partial t} - \frac{\partial w}{\partial x_i} \mathbf{F}_i - w\mathbf{S}\ d\Omega + \int_{\Gamma_F} w\ (\mathbf{F}_i^b n_i)\ d\Gamma = \mathbf{0} \tag{II.11}$$

where n_i's are the components of the unit outward-normal on the Γ. Equation (II.11) is simply obtained by: multiplying the equation (II.8.a) by the weight function w, performing

integration over Ω, applying the integration by parts, and substituting equation (II.8.b) in the resulting boundary integral. Since weight functions vanish on the Dirichlet-type boundaries, the boundary integral has been limited to Γ_F. By performing a spatial integration, a *semi-discrete* approach is pursued and thus the weight functions do not depend on the time. On the other hand, time dependency of the solution has been translated into a time-varying solution space. In equations (II.9) and (II.10), $\mathcal{H}^1$ is the usual Sobolev space of weakly differentiable functions, and $[\mathcal{H}^1]^{n_Q}$ is the corresponding space of vector functions with n_Q components.

Toward developing the discrete (finite dimension) form, the Ω is approximated by a *computational domain* Ω^h with piecewise-polynomial boundary Γ^h. Then, the *finite-element mesh* $\mathcal{T}^h = \{\Omega^1, \Omega^2, ..., \Omega^{n_{el}}\}$ is defined as the geometrical division of Ω^h into a finite number of non-overlapping elements such that

$$\Omega^h = \bigcup_{e=1}^{n_{el}} \Omega^e \tag{II.12}$$

where n_{el} denotes the number of elements. Accordingly, the boundary is partitioned as:

$$\Gamma^h = \bigcup_{e=1}^{n_{el}} \Gamma^e \cap \Gamma \tag{II.13}$$

where Γ^e denotes the boundary of the element e. A two dimensional mesh is called *regular* if for any two elements Ω^e and Ω^f, $e \neq f$, only one of the followings is true [84]:

1. $\Omega^e \cap \Omega^f$ is empty,
2. $\Omega^e \cap \Omega^f$ is a single common vertex,
3. $\Omega^e \cap \Omega^f$ is a single common edge.

With such definition, a regular mesh does not include any hanging node. For the sake of simplicity, to present the discretization, the mesh is assumed to be regular. However, in the section of constrained approximation (Section III.3), this assumption will be relaxed.

To define the *approximation spaces*, each element e is equipped with a polynomial order $1 \leq P(\Omega^e) = P^e$. In general, the distribution of polynomial orders may be non-uniform. Obviously, this is a necessity for p- and hp-adaptation. At this point, spatial approximation spaces can be precisely defined as

$$\begin{aligned} \boldsymbol{\mathcal{S}}_t^h := \{\mathbf{Q} | \mathbf{Q}(\cdot,t) \in [\mathcal{H}^1(\Omega^h)]^{n_Q}, \mathbf{Q}(\cdot,t)|_{\Omega^e} \in [\mathcal{P}_{P^e}(\Omega^e)]^{n_Q}, \\ t \in [0,\infty) \ \forall e \text{ and } \mathbf{Q}(\cdot,t) = \mathbf{Q}_D^h \text{ on } \Gamma_D^h\} \end{aligned} \tag{II.14}$$

$$\mathcal{W}^h := \ \{w |\ w \in \mathcal{H}^1(\Omega^h);\ w|_{\Omega^l} \in \mathcal{P}_{P^e}(\Omega^e)\ \forall e \text{ and } w = 0 \text{ on } \Gamma_D^h\} \tag{II.15}$$

where $\mathcal{P}_P$ is the polynomial space, complete to the order P. Now, the discrete solution to the weak form can be expressed as: for any $t \in [0,\infty)$ find $\mathbf{Q}^h \in \boldsymbol{\mathcal{S}}_t^h$ such that for all $w^h \in \mathcal{W}^h$,

$$\iint_{\Omega^h} w^h \frac{\partial \mathbf{Q}^h}{\partial t} - \frac{\partial w^h}{\partial x_i}\mathbf{F}_i - w^h \mathbf{S}\ d\Omega + \int_{\Gamma_F^h} w^h\ (\mathbf{F}_i^b n_i)\ d\Gamma = \mathbf{0} \tag{II.16}$$

provided that $\mathbf{F}_i$ and $\mathbf{S}$ are calculated based on $\mathbf{Q}^h$. This solution is expanded in a piecewise manner as

$$\mathbf{Q}^h = \sum_{i=1}^{n_{DOF}} \mathbf{Q}_i N_i \quad \text{on } \Omega^h \tag{II.17}$$

where $\mathbf{Q}_i$'s are the solution's coefficients or the Degrees of Freedom (DOFs), N_i's are the basis functions for the finite-dimension space $\boldsymbol{\mathcal{S}}_t^h$ and n_{DOF} is the dimension of that space as well as the number of DOFs. If the weight function w^h is constructed using the same

class as the solution's basis functions N, the original Bubnov-Galerkin discretization is derived. It is well-known that in situations where advection fluxes dominate diffusion fluxes, the original Galerkin method will suffer from spurious oscillations which lead to instability of the method. As Dona and Huerta [90] describe, these oscillations are due to the negative numerical diffusion produced by the original Galerkin discretization. Intuitively, the common remedy in CFD is to add numerical dissipation to the discretization. During the last three decades, several stabilization methods, for example the Streamline-Upwind Petrov-Galerkin (SUPG) [26, 27, 36], Galerkin Least Square (GLS) [91], and Variational Multi-Scale (VMS) [74, 92, 93] methods have been developed for continuous Galerkin methods. An excellent review of stabilized methods for compressible flows, including their formulation and history, can be found in reference [36]. In the developed framework, SUPG, GLS, and VMS can each be used for stabilization. In this study, however, only the results of the SUPG method have been presented. To clarify the notation used for the stabilization and the weak boundary conditions, it is also useful to rewrite the equation (II.8.a) using a quasi-linear differential operator as (see also references [94, 95])

$$\mathcal{L}(\mathbf{q}) = \mathbf{S} \tag{II.18}$$

$$\mathcal{L} := [\mathbf{A}^q]\frac{\partial}{\partial t} + [\mathbf{A}_i^E]\frac{\partial}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\mathbf{G}_{ij}\frac{\partial}{\partial x_j}\right) \tag{II.19}$$

where $[\mathbf{A}^q] = \frac{\partial \mathbf{Q}}{\partial \mathbf{q}}$ is the variable transformation matrix, $[\mathbf{A}_i^E] = \frac{\partial \mathbf{F}_i^E}{\partial \mathbf{q}}$ is the Euler flux Jacobian matrix, and $[\mathbf{G}_{ij}]$ is the diffusivity matrix which is defined such that $\mathbf{F}_i^v = [\mathbf{G}_{ij}]\frac{\partial \mathbf{q}}{\partial x_j}$. Here, $\mathbf{q}$ is the vector of the dependent variables which may be chosen over the conservative variables $\mathbf{Q}$ to facilitate the implementation. In the present work, it is the

vector of state variables $\mathbf{q} = [\rho, u_i, T, \tilde{\nu}]^T$. This choice is based on the need for modeling fluids with nonlinear equations of state that typically provide the pressure and other thermodynamic variables in terms of density and temperature. Hereafter, $\mathbf{q}^h$ is considered as the discrete solution. The stabilization is performed by adding a stabilization term to the Galerkin discretization as

$$\underbrace{\iint_{\Omega^h} N[\mathbf{A}^q]\frac{\partial \mathbf{q}^h}{\partial t} - \frac{\partial N}{\partial x_i}\mathbf{F}_i - N\mathbf{S}\; d\Omega + \int_{\Gamma_F^h} N(\mathbf{F}_i^b n_i)\; d\Gamma}_{Galerkin\ Discretization} + \underbrace{\sum_{e=1}^{n_{el}} \iint_{\Omega^e} [\mathbf{P}^e]\left([\mathbf{A}^q]\frac{\partial \mathbf{q}^h}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} - \mathbf{S}\right) d\Omega}_{Stabilization\ Term} = \mathbf{0} \tag{II.20}$$

provided that $\mathbf{F}_i$ and $\mathbf{S}$ are calculated based on $\mathbf{q}^h$**.** In above equation, N has been used instead of w to emphasize that hereafter the weight functions are chosen from the same class as the solution's basis functions. The stabilization term is calculated over all the elements in the computational domain. The value in the parentheses of this term is the residual of the original PDE and this is why these methods are known as residual-based stabilization methods [93]. The major property of such a method is the consistency, in the sense that the stabilization term goes to zero as the numerical solution approaches the exact solution of the PDE. The term [**P**] is called the perturbation to the test function space as it modifies the original Galerkin methods to a Petrov-Galerkin method with $N[\mathbf{I}] + [\mathbf{P}]$ as the weight function [36].

For SUPG method,

$$[\mathbf{P}^e] = [\mathbf{A}_i^E]\frac{\partial N}{\partial x_i}[\boldsymbol{\tau}^e] \tag{II.21}$$

where $[\boldsymbol{\tau}]$ is called the stabilization matrix. It has the dimension of time and it can be obtained based on the eigensystem decomposition of the projection of the flux Jacobian matrices onto the spatial gradients of the basis functions. However, the stabilization may also be derived from flux-vector splitting formulations. Advantages of such an approach are that differentiability, positivity, and total enthalpy conservation can be maintained [39, 96, 97]. In the present study, only the "standard" stabilization based on eigensystem decomposition is used [98] with viscous scaling as described in reference [99]. This term for element e is given by

$$[\boldsymbol{\tau}^e]^{-1} = \sum_{j=1}^{n_s^e} \left| \frac{\partial N_j^e}{\partial x_i} [\mathbf{A}_i^E] \right| + \frac{\partial N_j^e}{\partial x_i} [\mathbf{G}_{ik}] \frac{\partial N_j^e}{\partial x_k} \tag{II.22}$$

where N_j^e and n_s^e are the *shape functions* and number of modes within element e, respectively. Here, a shape function within an element is considered as the restriction of a basis function to that element (see Section III.1). In above equation,

$$\left| \frac{\partial N_j^e}{\partial x_i} [\mathbf{A}_i^E] \right| = [\mathbf{T}] |\boldsymbol{\Lambda}| [\mathbf{T}]^{-1} \tag{II.23}$$

where $[\mathbf{T}]$ and $|\boldsymbol{\Lambda}|$ denote the matrix of right eigenvectors and the diagonal matrix of absolute values of the eigenvalues of the left hand side of the above equation, respectively.

Remarks

1. In the SUPG scheme, the numerical dissipation is added in streamwise direction. The added dissipation can be interpreted as an added numerical viscous flux [33], or upwinding through giving more weight to the up-stream element [26].
2. For the GLS scheme,

$$[\mathbf{P}^e] = \mathcal{L}(N)[\boldsymbol{\tau}^e] \tag{II.24}$$

where operator $\mathcal{L}$ was previously defined in equation (II.18). If a steady-state solution is sought, for pure advection (e.g. Euler equations) or for linear elements, the GLS and SUPG schemes are equivalent.

II.3 Initial and Boundary Conditions

In this study, the initial condition is set to free-stream condition with exception of the no-slip walls, where the no-slip condition is applied. The free-stream value of the turbulence working variable $\tilde{\nu}$ is set to 3 for fully turbulent flows [100].

Regarding boundary conditions, far-filed, inviscid wall, and no-slip wall boundaries are considered. The walls are assumed to be adiabatic. For the far-field boundaries, the boundary flux vector $\mathbf{F}_i^b$ only includes the inviscid part and is constructed using the Roe scheme [101] based on the free-stream and interior state values. For the inviscid walls, the boundary flux vector only takes the pressure from interior and thus $\mathbf{F}_i^b = [0, \delta_{1i}p, \delta_{2i}p, 0]^T$.

II.3.1 Weakly Imposed Dirichlet Boundary Conditions

To apply no-slip condition on the walls, the discrete weak form in equation (II.20) may be augmented with the weak implementation of the Dirichlet boundary conditions [74]. Bazilevs and Hughes proposed such an approach in reference [75] for the advection diffusion and incompressible Navier–Stokes equations and then, they extended their method in reference [76]. It should be mentioned that this method is essentially based on the Nitsche's method [77]. Here the same idea is followed, although the utilized formulation is based on

a DG discretization for compressible flows. This formulation is obtained by ignoring interior stabilization from Symmetric Interior Penalty Galerkin (SIPG) method [15, 21, 25, 78]. As it will be shown in the numerical results, weak boundary condition is one of the essential ingredients to obtain a smooth adjoint solution near the no-slip wall boundaries. To impose weak boundary conditions, equation (II.20) is augmented as follows

$$\begin{aligned}\iint_{\Omega^h} N[\mathbf{A}_0]\frac{\partial \mathbf{q}^h}{\partial t} - \frac{\partial N}{\partial x_i}\mathbf{F}_i - N\mathbf{S}\; d\Omega + \int_{\Gamma_F^h} N(\mathbf{F}_i^b n_i)\; d\Gamma \\ + \sum_{e=1}^{n_{el}} \iint_{\Omega^e} [\mathbf{P}^e]\left([\mathbf{A}_0]\frac{\partial \mathbf{q}^h}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} - \mathbf{S}\right) d\Omega + \mathcal{N}_{\Gamma_D} = \mathbf{0}\end{aligned} \tag{II.25}$$

where

$$\begin{aligned}\mathcal{N}_{\Gamma_D} = &\int_{\Gamma\cap\Gamma_D^h} N\Big(\mathbf{F}_i^E(\mathbf{q}^b) - \mathbf{F}_i^v(\mathbf{q}^b, \nabla\mathbf{q}^h)\Big)n_i\; d\Gamma \\ &- \int_{\Gamma\cap\Gamma_D^h} \left([\mathbf{G}_{i1}(\mathbf{q}^b)]\frac{\partial N}{\partial x_i}, [\mathbf{G}_{i2}(\mathbf{q}^b)]\frac{\partial N}{\partial x_i}\right)\cdot(\mathbf{q}^h - \mathbf{q}^b)\vec{n}\; d\Gamma \\ &+ \int_{\Gamma\cap\Gamma_D^h} \eta[\mathbf{G}(\mathbf{q}^b)](\mathbf{q}^h - \mathbf{q}^b)\vec{n}\cdot N\vec{n}\; d\Gamma\end{aligned} \tag{II.26}$$

where $\mathbf{q}^b$ is a state vector which is constructed based on the Dirichlet boundary conditions and interior solution. Adiabatic and no-slip conditions yield $\mathbf{q}^b = [\rho, 0, 0, T, 0]^T$ and thus the components of the boundary viscous flux associated with the energy equation vanishes. Note that to calculate $\mathbf{q}^b$, ρ and T are calculated based on the interior solution on the element adjacent to the boundary. Also, turbulence working variable $\tilde{\nu}$ is set to zero at no-slip walls.

The first term in equation (II.26) is called the consistency term. To justify this term, note that if the Dirichlet boundary conditions are imposed weakly, unlike equations (II.10) and (II.15), weight functions do not vanish on the boundary. Thus this term results from

the integration by parts in the derivation of the weak form in equation (II.11). Note that to calculate the viscous flux, the gradients $\nabla \mathbf{q}^h$ are calculated based on the interior solution on the element adjacent to the boundary.

The second term is called the symmetry term, or adjoint-consistency term. If the discretization is adjoint-consistent, by substitution of the exact solution of the adjoint problem as the weight function in equation (II.25), this equation should be satisfied identically [15, 76, 102].

The last term is referred to as the penalty term, where the penalty parameter η is explicitly evaluated using the element geometry parameters as well as the order of discretization [21, 25]. Here the value given by reference [21] has been used

$$\eta^e = n_s^e \frac{|\Gamma^e|}{|\Omega^e|} \tag{II.27}$$

where $|\Gamma^e|$ and $|\Omega^e|$ are the perimeter and area of the element e which is adjacent to the boundary.

Remarks

Considering that the viscous flux is linear in the flow gradients, the sum of consistency and penalty terms can be written as a *modified* viscous flux. For this purpose, a *modified* gradient vector $\nabla \mathbf{q}^{\boxed{h}}$ is defined as

$$\nabla \mathbf{q}^{\boxed{h}} = \nabla \mathbf{q}^h + \eta(\mathbf{q}^h - \mathbf{q}^b)\vec{n} \tag{II.28}$$

and thus, equation (II.26) is simplified to

$$\begin{aligned}\mathcal{N}_{\Gamma_D} = & \int_{\Gamma \cap \Gamma_D^h} N \left(\mathbf{F}_i^E(\mathbf{q}^b) - \mathbf{F}_i^v(\mathbf{q}^b, \nabla \mathbf{q}^{\boxed{h}}) \right) n_i \, d\Gamma \\ & - \int_{\Gamma \cap \Gamma_D^h} \left([\mathbf{G}_{i1}(\mathbf{q}^b)] \frac{\partial N}{\partial x_i}, [\mathbf{G}_{i2}(\mathbf{q}^b)] \frac{\partial N}{\partial x_i} \right) \cdot (\mathbf{q}^h - \mathbf{q}^b) \vec{n} \, d\Gamma \end{aligned} \tag{II.29}$$

Note that the $\nabla \mathbf{q}^{\boxed{h}}$ is only used for integration of the viscous flux along the boundary. As it will be mentioned in the Section IV.1.4, $\nabla \mathbf{q}^{\boxed{h}}$ will also be used in calculation of the functional of the interest (i.e. lift or drag) when a discrete adjoint solution is sought.

II.4 Semi-discrete Formulation

Due to Galerkin-based formulation, the weight space is spanned by the same basis functions as the solution space. Therefore, in order to obtain a semi-discrete formulation, the equations (II.20) and (II.25) can be iterated for N_i, $1 \leq i \leq n_{DOF}$. This process results in n_{DOF} equations which can be compactly written as

$$[\mathbf{M}] \frac{\partial \mathbf{q}^h}{\partial t} + \mathbf{R}(\mathbf{q}^h) = \mathbf{0} \tag{II.30}$$

where $\mathbf{R}$ denotes the spatial residual, and $[\mathbf{M}]$ denotes the mass matrix.

II.4.1 Steady-State Solutions

To obtain a steady-state solution, the equation (II.30) is discretized in time using an implicit backward difference formula (BDF). This yields

$$\mathbf{Res^{n+1}}(\mathbf{q}^{h,n+1}) = \frac{[\mathbf{M}]}{\Delta t}(\mathbf{q}^{h,n+1} - \mathbf{q}^{h,n}) + \mathbf{R}(\mathbf{q}^{h,n+1}) = 0 \tag{II.31}$$

where $\mathbf{Res^{n+1}}$ represents the unsteady flow residual at time step $n+1$. The implicit system is linearized using an automatic differentiation implementation [103] and the vector of the

dependent variables is updated in a Newton-iteration algorithm similar to that in reference [104]. The linearized system and the update equation are given by

$$[\mathbf{J}^n(\mathbf{q}^{h,n})]\Delta\mathbf{q}^{h,n} = -\mathbf{R}^n(\mathbf{q}^{h,n}) \tag{II.32}$$

$$\mathbf{q}^{h,n+1} = \mathbf{q}^{h,n} + w_{opt}\Delta\mathbf{q}^{h,n} \tag{II.33}$$

where $[\mathbf{J}] = \left[\frac{\partial \mathbf{Res}}{\partial \mathbf{q}^h}\right]$ denotes the Jacobian matrix. The w_{opt} is a nominal optimum relaxation factor which is determined in a line search process. It should be mentioned that the utilized Newton algorithm has been modified to include local time-steps which are amplified by the CFL number to accelerate the global convergence. At small CFL numbers, the algorithm essentially becomes an explicit method, whereas at high CFL numbers the algorithm approaches Newton's method. To enhance robustness, a limiting relaxation factor $w_{\rho,T}$ is determined such that neither the density nor the temperature changes by more than 10 percent. $w_{\rho,T}$ serves as the maximum factor during the line search that is used to determine the optimal value w_{opt}. Here, the RMS of the unsteady residual $\mathbf{Res^{n+1}}$ is evaluated at four values of the relaxation factors: 0, w_{min},$(w_{min} + w_{\rho,T})/2$, and $w_{\rho,T}$. The optimal value is found by locating the minimum of a fitted cubic polynomial within the range of [0, $w_{\rho,T}$]. If the optimal relaxation factor falls below the minimal value w_{min}, the step is rejected and the CFL number is divided by 10. If a full step is taken, as characterized by a relaxation factor of $w_{opt} = 1.0$, the CFL is doubled. In other cases the CFL remains at the previous value. In this study w_{min} is set to 0.1. Also, the maximum value of CFL has been set to 10^6 and no minimum value has been set.

II.4.2 Unsteady Solutions

For time-dependent solutions, the second order backward difference formula (BDF2) is applied on equation (II.30) which yields

$$\mathbf{Res^{n+1}}(\mathbf{q}^{h,n+1}) = \frac{[\mathbf{M}]}{\Delta t}\left(\frac{3}{2}\mathbf{q}^{h,n+1} - 2\mathbf{q}^{h,n} + \frac{1}{2}\mathbf{q}^{h,n-1}\right) + \mathbf{R}(\mathbf{q}^{h,n+1}) = 0 \tag{II.34}$$

Similar to the case of steady-state solution, the unsteady residual is linearized and the resulting system of equations is solved using Newton's method. However, a constant time-step is used for the entire computational domain.

II.4.3 Solution of Linear System

At each iteration or time step, the linear system is solved using the GMRES [105] algorithm with a preconditioner based on incomplete LU decomposition with 0 and at most 5 levels of fill [106], for unsteady and steady state problems, respectively.

II.5 Artificial Viscosity Formulation

In this study, the artificial viscosity is added to the governing equations in order to stabilize the solution near the shock waves. To this end, the viscous flux is augmented as

$$\mathbf{F}_i^v = [\mathbf{G}_{ij}]\frac{\partial \mathbf{q}}{\partial x_j} + [\mathbf{A}_{ij}^{\epsilon}]\frac{\partial \widetilde{\mathbf{Q}}}{\partial x_j} \tag{II.35}$$

where $\widetilde{\mathbf{Q}}$ is a state vector which includes ρH instead of ρE. This choice is expected to conserve the total enthalpy across the shock which is required by Rankine-Hugoniot shock jump relations (see references [80, 81]). The artificial viscosity matrix is $[\mathbf{A}_{ij}^{\epsilon}] = \hat{\epsilon}\,[\mathbf{D}_{ij}]$, where $\hat{\epsilon}$ is the artificial viscosity, and $[\mathbf{D}_{ij}]$ is a diagonal matrix. Persson and Peraire [81]

have proposed a piecewise-constant artificial viscosity that scales with h/P in a finite-element discretization, aiming to make the shock width also be $O(h/P)$. To detect the shock, they have developed a discontinuity sensor based on the rate of decay of the expansion coefficients in a hierarchical basis used to expand the discrete solution. The beauty of their method is that by fixing the element size h, and increasing the polynomial order P, the shock can be resolved within one element. The downside, however, is that due to the element-wise design of the method, large jumps in artificial viscosity may be observed between adjacent elements which may result in spurious oscillations in the flow gradients. Thus, Barter and Darmofal [80] extended that method by developing a diffusion model, in form of a PDE, to smooth the distribution of the artificial viscosity. In their proposed model, which is referred to as artificial viscosity PDE, the above mentioned discontinuity sensor has been used in the source term, and the diffusion term has been biased by the directional mesh size metrics which is particularly useful for anisotropic meshes. Burgess and Mavriplis [107] have compared the above mentioned methods and concluded that although the PDE-based method results in a more dissipated solution, it shows a more robust and consistent convergence behavior in an adjoint-based adaptive algorithm. In this study, a similar comparison was repeated which came to the same conclusion. Thus the PDE-based method was chosen. However, it should be mentioned that the benefits of this method comes with the extra cost of the additional PDE. All the studies in references [80, 81, 107] have used the DG scheme and to our knowledge, this is the first application of the PDE-based artificial viscosity method in the PG scheme. The following formulation has been taken from reference [80]. The artificial viscosity matrix is given by,

$$[\mathbf{A}^{\epsilon}_{ij}] = \hat{\epsilon}(\epsilon)\text{diag}\left[\hat{I}\frac{h_i}{\bar{h}}\right]\delta_{ij} \tag{II.36}$$

$$\hat{\epsilon}(\epsilon) \equiv \begin{cases} 0, & \epsilon \leq \theta_L \\ \dfrac{\theta_H}{2}\left(1+\sin\left[\pi\left(\dfrac{\epsilon-\theta_L}{\theta_H-\theta_L}-\dfrac{1}{2}\right)\right]\right), & \theta_L < \epsilon < \theta_H \\ \theta_H & \epsilon \geq \theta_H \end{cases} \tag{II.37}$$

where $\bar{h}(\vec{x})$ is the arithmetic mean of the components of $\vec{h}(\vec{x}) \in \mathbb{R}^{n_{sd}}$ which is a vector of the element size metrics described in following, and $\hat{I} = [1,1]^T$. Using equation (II.37), the artificial viscosity $\hat{\epsilon}$ varies smoothly between zero and maximum value of θ_H, as ϵ, the working variable of the artificial viscosity PDE, varies between θ_L and θ_H. Here, $\theta_L = 0.01\theta_H$, and $\theta_H = \lambda_{max}\bar{h}/P$. Here, the maximum eigenvalue (or wave speed) of the system is $\lambda_{max} = \sqrt{u_i u_i} + c$, where c is the speed of sound. The artificial viscosity PDE and its boundary conditions are:

$$\frac{\partial\epsilon}{\partial t} = \nabla\cdot\left(\frac{[\boldsymbol{\eta}]}{\tau}\nabla\epsilon\right) + \frac{1}{\tau}\left(\frac{\bar{h}(\vec{x})}{P}\lambda_{max}(\mathbf{q})S_e(\mathbf{q}) - \epsilon\right), \vec{x}\in\Omega \text{ and } t\in[0,\infty) \tag{II.38}$$

$$\frac{\partial\epsilon}{\partial\vec{n}} = \frac{\epsilon-\epsilon_\infty}{L}, \qquad \vec{x}\in\Gamma \text{ and } t\in[0,\infty) \tag{II.39}$$

where τ is a time constant, and $[\boldsymbol{\eta}]$ is the diffusivity matrix. They are compactly given by

$$\frac{[\boldsymbol{\eta}]}{\tau} = \frac{C_1C_2P\lambda_{max}(\mathbf{q})}{\min\limits_i h_i}\text{diag}(h_x^2, h_y^2); \quad C_1C_2 = 15. \tag{II.40}$$

S_e is the shock indicator which forces the ϵ to be non-zero near the discontinuities. It is given by,

$$S_e(F_e,\theta_S,\psi_0,\Delta\psi) = \begin{cases} 0, & F_e \leq \psi_0 - \Delta\psi \\ \theta_S, & F_e \geq \psi_0 + \Delta\psi \\ \dfrac{\theta_S}{2}\left(1+\sin\left(\pi\dfrac{F_e-\psi_0}{2\Delta\psi}\right)\right) & |F_e-\psi_0| < \Delta\psi \end{cases} \tag{II.41}$$

where F_e is the discontinuity sensor developed by Persson and Peraire [81] which is given by,

$$F_e = \log_{10}\left(\frac{\langle q-\hat{q}, q-\hat{q}\rangle_e}{\langle q, q\rangle_e}\right) \tag{II.42}$$

where $\langle , \rangle_e$ denotes L_2 norm on element e, and q is a state variable which for a solution of order P^e is expanded as $q = \sum_{j=1}^{n_s^e(P^e)} q_j N_j^e$ within element e. $\hat{q}$ is a truncated expansion given by $\hat{q} = \sum_{j=1}^{n_s^e(P^e-1)} q_j N_j^e$. In this study, density has been used as the state variable. Since we are seeking a continuous solution, $P = 1$ is the lowest allowable polynomial order. Thus, for linear elements, $\hat{q}$ is taken to be the arithmetic average of the q_j in the element.

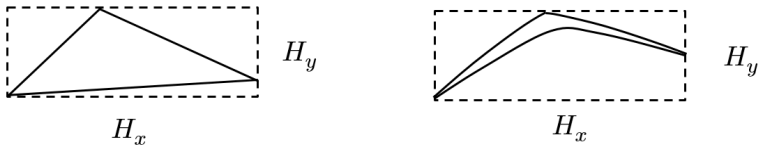
In equation (II.41), $\theta_S = 1$ is the maximum value. ψ_0 and $\Delta\psi$ are empirical constants which determine when the shock indicator should take effect. Here, $\psi_0 = -(4 + 4.25\log_{10}(P))$, and $\Delta\psi = 0.5$.

Remarks

1. Equation (II.39) denotes a Robin boundary condition and it has been designed to be effective regardless of the angle at which the shock intersects with the boundary. However, to simplify the implementation, here, a homogeneous Neumann condition $\frac{\partial \epsilon}{\partial \vec{n}} = 0$ has been used. This boundary condition implies that the shock is normal to the boundary [79].
2. In reference [80], the vector of mesh size metrics $\vec{h}(\vec{x})$ is defined such that the arithmetic mean, $\bar{h}(\vec{x})$ is a continuously varying scalar function throughout the mesh. In particular, $\vec{h}(\vec{x}) = \sum_{k=1}^{n_{sd}+1} \vec{\vec{H}}_k N_k(\vec{x})$, where $\vec{\vec{H}}_k$ is the average value of the bounding box vectors of all elements surrounding the k^{th} vertex node of an element

and N_k is the linear vertex shape function associated with the vertex node. The bounding box vectors for a triangular element with linear edges has been shown in figure II.1.a. Here we have made a modification which is particularly beneficial for curved boundary layer elements. Figure II.1.b shows a thin element that has the same mesh metrics as the triangle in Fig 1.a. However, the curved element has smaller area and accordingly it needs less artificial viscosity. Thus, we use $|\Omega|/H_y$ instead of H_x, and $|\Omega|/H_x$ instead of H_y, where $|\Omega|$ is the area of the element. Using this modification, the mesh size metrics for thin boundary layer elements reduces significantly and accordingly less artificial viscosity is added to these elements.

3. To solve the equation (II.38), the artificial viscosity $\hat{\epsilon}$ is appended to the state vector $\mathbf{q}$ in equation (II.18) and thus the artificial viscosity PDE is discretized using the same PG scheme described earlier.



(a) Linear edges (b) Curved edges

Figure II.1 Mesh size metrics

II.6 Mesh Curving Strategy

A common practice in CFD, and particularly in the second-order accurate schemes, is that the boundaries of the geometry are represented by a series of linear elements. However, to achieve higher accuracies, an increased conformity is required to properly

account for surface curvature. Therefore, a mechanism is required to project the boundary edges to the exact geometry. However, such projection may generate collapsed elements near the boundary, especially when high aspect ratio elements are used, for example in the boundary layers. Thus, a robust mesh deformation strategy must be employed to move the interior elements away from the curved boundaries. In this study, a modified form of the linear elasticity equations has been employed to deform the computational mesh. These equation are given by

$$\begin{aligned} \frac{\partial}{\partial x}\left[d_{11}\frac{\partial \delta_x}{\partial x}+d_{12}\frac{\partial \delta_y}{\partial y}\right]+\frac{\partial}{\partial y}\left[d_{44}\left(\frac{\partial \delta_x}{\partial y}+\frac{\partial \delta_y}{\partial x}\right)\right]=0 \\ \frac{\partial}{\partial y}\left[d_{21}\frac{\partial \delta_x}{\partial x}+d_{22}\frac{\partial \delta_y}{\partial y}\right]+\frac{\partial}{\partial x}\left[d_{44}\left(\frac{\partial \delta_x}{\partial y}+\frac{\partial \delta_y}{\partial x}\right)\right]=0 \end{aligned} \tag{II.43.a}$$

where $\delta=(\delta_x,\delta_y)$ denotes the displacement vector in the Cartesian coordinate directions, the coefficients $d_{11}=d_{22}=\frac{E(1-\nu)}{(1+\nu)(1-2\nu)}$, $d_{12}=d_{21}=\frac{E\nu}{(1+\nu)(1-2\nu)}$, $d_{44}=\frac{E\nu}{2(1+\nu)}$. Here, E denotes the Young's modulus, and ν is the Poisson's ratio. To improve the robustness of the method for extremely high aspect ratio elements, E can be defined as a decreasing function of the distance to the wall.

To solve the above equations, the Galerkin method with similar hierarchical basis functions as those used for flow variables are used. In the FUNSAFE framework, the polynomial orders of the flow variables and the geometry mappings can be chosen independently for each individual element. This is particularly useful to employ sub- and super-parametric elements arbitrarily. For clarification, hereafter, the polynomial order of the solution variables is denoted by P, and the polynomial order of the geometry mappings

is denoted by Q. For instance, a P1-Q3 element is a super-parametric element with linear shape functions for the flow variables, and cubic shape functions for the geometry mapping.

Typically, a mesh is first generated using linear edges. Since hierarchical shape functions are used here, the vertex nodes are directly placed on the exact geometry while the edge nodes are chosen such that the resulting computational edge passes through some desired points on the exact geometry. For example, in the case of quadratic edges, the edge node is iteratively updated such that the mid-point of the edge marches from its initial location toward the normal direction to the edge until the intersection with exact geometry is determined. Similarly, for cubic edges, the number of points matched on the exact geometry is increased to two. As it will be shown in the results, in all of the studied cases, the initial mesh is equipped with curved Q3 elements. During the adaptation, h-refined elements are, generally, embedded within the parent elements except for those near the solid boundary where they need to conform to the exact geometry. The geometry of the children elements are initialized based on the current mapping of their parents. For maximum conformity, it is useful to repeat the mesh curving process if the refinement occurs on the boundary. Due to the possible presence of the hanging nodes in the geometry mapping, once more, the constrained approximation is used to solve the linear elasticity equations on an hp-adapted mesh.

CHAPTER III

SOLUTION EXPANSION AND CONSTRAINED APPROXIMATION

III.1 Basis Functions

Considering the transformation of the conservative variables $\mathbf{Q}^h$ to the dependent variables $\mathbf{q}^h$, instead of equation (II.17), the discrete solution is expanded as,

$$\mathbf{q}^h = \sum_{i=1}^{n_{DOF}} \mathbf{q}_i N_i \quad on\ \Omega^h \tag{III.1}$$

In order to relate the above expansion to the computational mesh, it is useful to define the term *node* as any geometrical entity (vertex, edge, and element interior) that associates a solution coefficient $\mathbf{q}_i$ and its corresponding basis function N_i to the computational mesh. As will be seen in the next section, this definition is particularly beneficial when hierarchical basis functions are employed. The basis functions N_i are constructed in a piecewise manner by combining the shape functions which are defined within elements. To explain this concept, note that the discrete solution $\mathbf{q}^h$ can be expanded as

$$\mathbf{q}^h = \bigcup_{e=1}^{n_{el}} \mathbf{q}^e \quad on\ \Omega^h \tag{III.2}$$

where $\mathbf{q}^e$ is the restriction of $\mathbf{q}^h$ to the element e. Then, $\mathbf{q}^e$ is expanded as

$$\mathbf{q}^e = \sum_{j=1}^{n_s^e} \mathbf{q}_j^e N_j^e \quad \text{on } \Omega^e \tag{III.3}$$

$$N_j^e = N_{i_j(e)} \quad \text{on } \Omega^e \tag{III.4}$$

$$N_i = \bigcup_{e \in E_i} N_{j_i(e)}^e \tag{III.5}$$

where $\mathbf{q}_j^e$, N_j^e, and n_s^e are the local solution coefficients, shape functions, and the number of shape functions within element e, respectively. Note that j is a local index within element e, and i is a global index referring to the node i within the mesh. Accordingly, $j_i(e)$ is the local index in element e which points to the node i, and $i_j(e)$ is the global index associated with the local node j. Also, E_i is the set of the elements that contain the node i. In this manner, the basis function N_i is only non-zero over the elements of E_i and the shape function $N_{j_i(e)}^e$ is the restriction of N_i to the element e. This notion has been illustrated in figure III.1. In this figure, the blue umbrella is a basis function and each individual triangular facet is a local shape function.

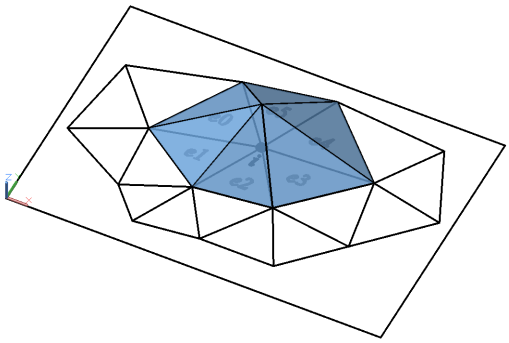


Figure III.1 Sample basis function

Due to continuity of the solution space, the basis functions need to keep the continuity across element interfaces. In other words, the trace of the shape functions of adjacent elements that contribute in forming N_i need to match on the element interfaces. As will be discussed in the section of constrained approximation (Section III.3), this is an important property which should be retained in presence of hanging nodes.

III.2 Shape Functions

In the present study, both Lagrange and hierarchical shape functions have been implemented for triangular elements. However, it should be mentioned that in spite of their popularity, the Lagrange shape functions are not the best choice when p- or hp-adaptation is desired. One reason is that by changing the polynomial order, all the shape functions in an element need to be updated, and an interpolation step is required to project the solution on the new set of nodes. Another reason is that two adjacent elements with different polynomial orders may not be able to share the nodes on the common interface (edge in 2D and edge/face in 3D). This is particularly important when a continuous solution is sought. Although, by using a technique known as *constrained approximation* [73, 82-88], the continuity can be maintained, separate sets of nodes need to be stored for different elements surrounding the mentioned interface (see reference [73]).

Above mentioned issues can be addressed by employing *hierarchical* shape functions. The notion of hierarchy comes from the fact that higher-order polynomial spaces can be incrementally built from the lower-order ones. In other words, to obtain higher order expansions, new terms, which can be viewed as corrections, are added to the old expansion.

A remarkable feature of this approach is that some shape functions are associated with the interior of the elements and the rest are associated with sub-element entities (i.e. vertices/edges in 2D). Accordingly, in addition to the element, edges are also equipped with polynomial orders. As a result, by changing the polynomial order associated with an edge, the polynomial spaces of all the elements, which share that edge, are automatically updated. Therefore, non-uniform distribution of the polynomial orders is achieved more easily than Lagrange elements. Another notable feature of hierarchical shape functions is that obtaining arbitrary higher order expansions for triangular elements can be automated in a straightforward manner. Hierarchical shape functions have been previously employed in SUPG schemes in studies by Whiting and Jansen [94] and Whiting et al. [95].

The hierarchical shape functions used in this study are similar to those presented in reference [84]. To clarify the upcoming notation, it should be mentioned that so far the symbol e has been used to denote elements. Hereafter, it is used for edges as well. However, in order to avoid confusion, when it is used for an edge, it comes with a local index. For example, e_1 denotes the first edge in an element.

To facilitate the numerical integrations, the shape functions have been be presented for a reference element which is defined as,

$$\widehat{\Omega} = \{(\xi, \eta) \in \mathbb{R}^2;\ -1 \leq \xi + \eta \leq 0\} \tag{III.6}$$

Figure III.2 shows this element and the convention used for numbering and orientation of the vertices and edges. Letting P^b and P^{e_i} denote the polynomial orders of the element's interior and the local edge e_i, respectively, the polynomial space for the reference element is defined as,

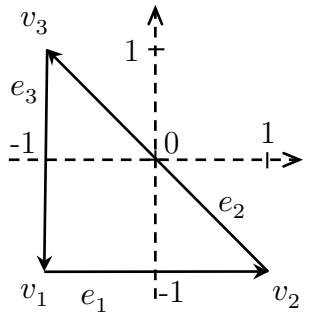


Figure III.2 The reference element

$$W_T = \Big\{ w \in \mathcal{P}_{P^b}(\widehat{\Omega}); w|_{e_j} \in \mathcal{P}_{P^{e_i}}(e_i), i = 1, \dots, 3 \Big\} \tag{III.7}$$

where

$$\mathcal{P}_P(\widehat{\Omega}) = \text{span}\Big\{ \xi^i \eta^j; (\xi, \eta) \in \widehat{\Omega}; i, j = 0, \dots, p;\ i + j \leq P \Big\} \tag{III.8}$$

To satisfy the H^1-conformity, it is assumed that

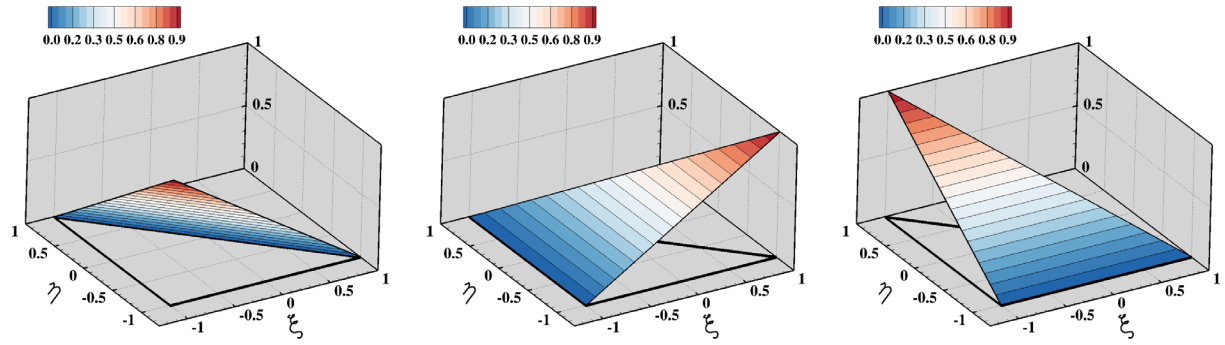
$$P^{e_i} \leq P^b, \qquad 1 \leq i \leq 3 \tag{III.9}$$

In two dimensions, the complete set of the shape functions is obtained by *vertex*, *edge*, and *bubble* functions. For a triangle element, $(P^b + 1)(P^b + 2)/2$ shape functions are required to reach a polynomial space complete to the order P^b.

The vertex functions are given by

$$\begin{aligned} \varphi^{v_1} &= -\frac{1}{2}(\xi + \eta) \\ \varphi^{v_2} &= \frac{1}{2}(\xi + 1) \\ \varphi^{v_3} &= \frac{1}{2}(\eta + 1) \end{aligned} \tag{III.10}$$

These functions, which are similar to the linear Lagrange shape functions, have been illustrated in figure III.3.



(a) φ^{v_1} (b) φ^{v_2} (c) φ^{v_3}

Figure III.3 Vertex functions for the master element $\widehat{\Omega}$

The edge functions are defined using vertex functions and a kernel function ϕ as,

$$\begin{aligned}
\varphi_k^{e_1} &= \varphi^{v_1}\varphi^{v_2}\phi_{k-2}(\varphi^{v_2} - \varphi^{v_1}), \quad 2 \le k \le P^{e_1} \\
\varphi_k^{e_2} &= \varphi^{v_2}\varphi^{v_3}\phi_{k-2}(\varphi^{v_3} - \varphi^{v_2}), \quad 2 \le k \le P^{e_2} \\
\varphi_k^{e_3} &= \varphi^{v_3}\varphi^{v_1}\phi_{k-2}(\varphi^{v_1} - \varphi^{v_3}), \quad 2 \le k \le P^{e_3}
\end{aligned} \tag{III.11}$$

where

$$\begin{aligned}
\phi_0(x) &= c_0 \\
\phi_1(x) &= c_1 x \\
\phi_2(x) &= c_2(5x^2 - 1) \\
\phi_3(x) &= c_3(7x^3 - x)
\end{aligned} \tag{III.12}$$

Note that in equation (III.11), the parentheses denote function evaluation. With such definition, the edge e_i is associated with $(P^{e_i} - 1)$ shape functions, traces of which coincide with scaled Lobatto polynomials on the edge e_i and vanish on all other edges. Lobatto polynomials can be written as

$$l_k(x) = l_0(x) l_1(x) \phi_{k-2}\ (x), \quad 2 \leq k$$
$$l_0 = \frac{1-x}{2}, \quad l_1 = \frac{1+x}{2} \tag{III.13}$$

Figure III.4 illustrates the edge function for $k = 2, \ldots, 5$.

Finally the bubble functions are defined using the same kernel functions as

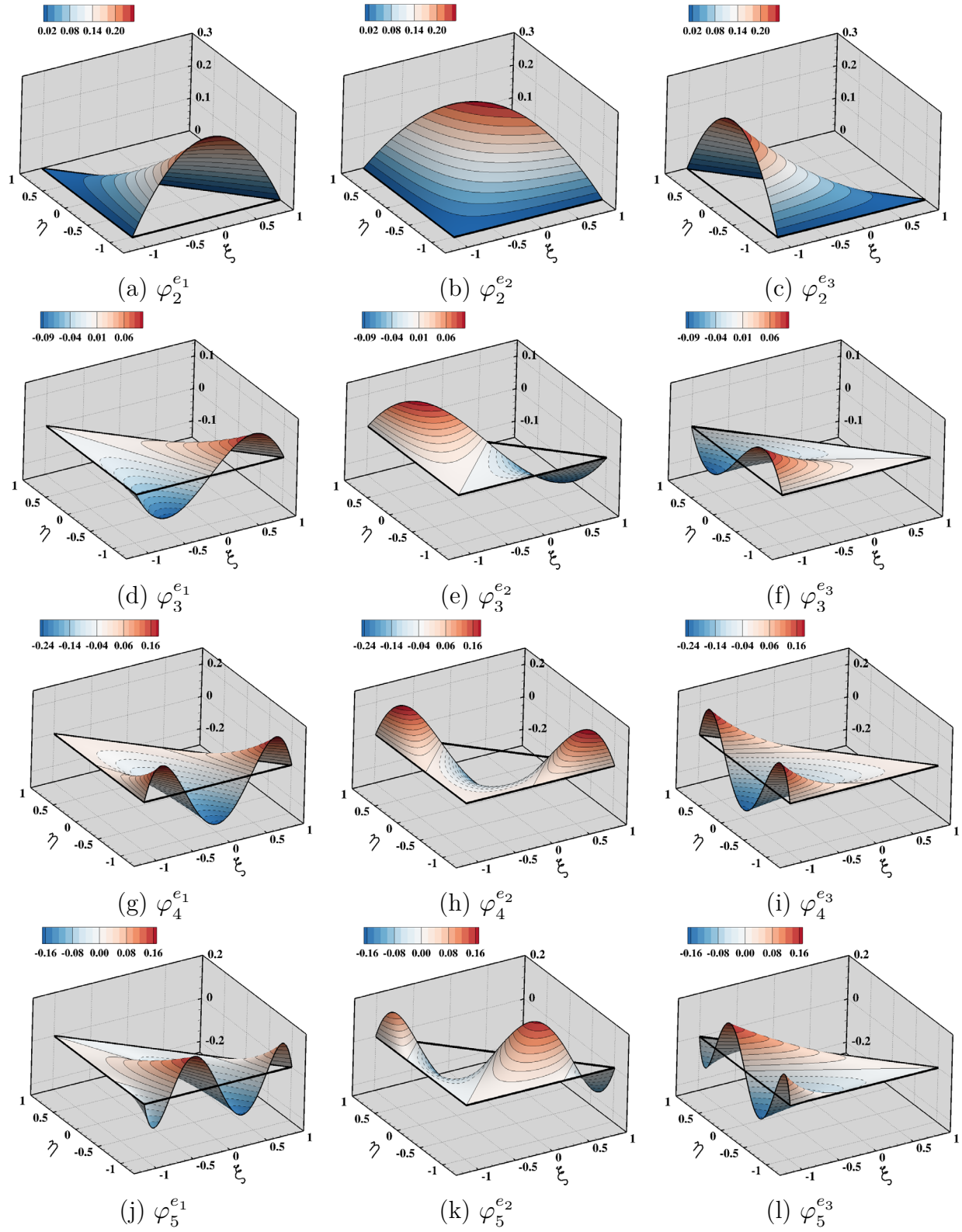
$$\varphi^b_{n_1,n_2} = \varphi^{v_1} \varphi^{v_2} \varphi^{v_3} \phi_{n_1-1}(\varphi^{v_2} - \varphi^{v_1}) \phi_{n_2-1}(\varphi^{v_1} - \varphi^{v_3}),$$
$$1 \leq n_1, n_2;\ n_1 + n_2 \leq P^b - 1 \tag{III.14}$$

Note that similar to equation (III.11), the parentheses denote function evaluation. With such definition, the bubble functions vanish on all the edges. Figure III.5 illustrates these function up to $P^b = 3, \ldots, 5$.

Remarks

1. Using kernel functions in the definition of edge and bubble functions, one only needs to extend the list of the kernel functions to reach higher order expansions.
2. The constants c_k can be utilized to condition the linear system in equation (II.32) [19].
3. Following set of c_k returns the standard Lobatto polynomials.

$$c_0 = -2\sqrt{\frac{3}{2}}, \qquad c_1 = -2\sqrt{\frac{5}{2}}, \qquad c_2 - \frac{1}{2}\sqrt{\frac{7}{2}}, \qquad c_3 = -\frac{1}{2}\sqrt{\frac{9}{2}} \tag{III.15}$$



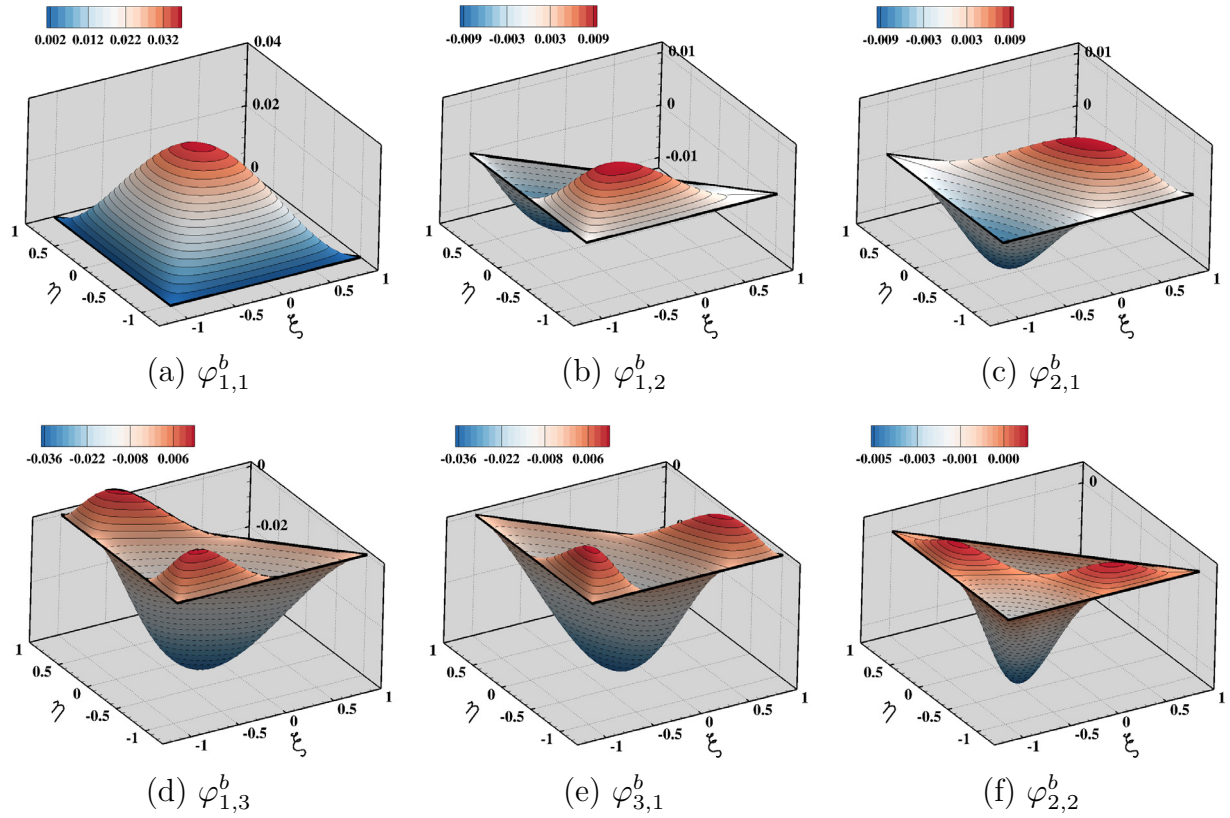
(a) $\varphi_2^{e_1}$ (b) $\varphi_2^{e_2}$ (c) $\varphi_2^{e_3}$

(d) $\varphi_3^{e_1}$ (e) $\varphi_3^{e_2}$ (f) $\varphi_3^{e_3}$

(g) $\varphi_4^{e_1}$ (h) $\varphi_4^{e_2}$ (i) $\varphi_4^{e_3}$

(j) $\varphi_5^{e_1}$ (k) $\varphi_5^{e_2}$ (l) $\varphi_5^{e_3}$

Figure III.4 Hierarchical edge functions for the master element $\widehat{\Omega}$, ($P^{e_i} = 2, \ldots, 5$)



(a) $\varphi^b_{1,1}$ (b) $\varphi^b_{1,2}$ (c) $\varphi^b_{2,1}$

(d) $\varphi^b_{1,3}$ (e) $\varphi^b_{3,1}$ (f) $\varphi^b_{2,2}$

Figure III.5 Hierarchical bubble functions for the master element $\widehat{\Omega}$, ($P^b = 3, \ldots, 5$)

III.2.1 Orientation of Element Edges

As mentioned earlier, to form continuous basis functions, the trace of the shape functions from adjacent elements, which contribute in that basis function, need to match on the interface between elements. In the case of hierarchical shape functions, the basis functions which are formed by combining odd-order edge functions from two adjacent elements need special attention. Note that if all the elements in the mesh follow the numbering and orientation convention defined by the reference element (shown in figure III.2), then two adjacent elements have opposite orientations on the shared edge. This has been shown in figure III.6. Now, based on the equation (III.12),

$$\varphi_{k-2}(-x) = (-1)^{k-2}\varphi_{k-2}(x), \quad 2 \leq k \tag{III.16}$$

Thus, considering equation (III.11), the trace of two odd-order edge functions from two adjunct elements will be opposite of each other. To solve this issue, as suggested in reference [84], each physical edge in the mesh can be assigned a unique (global) orientation. Then, elements will be equipped with an orientation sign (+1 or -1) for each of their local edges. The orientation sign which is shown by $o(e)$ indicates whether the local edge of the element has the same or opposite orientation with respect to the physical edge. It follows that one only needs to multiply each edge function by $o^k(e)$, to account for the orientation of the physical mesh. More details on this topic can be found in reference [84].

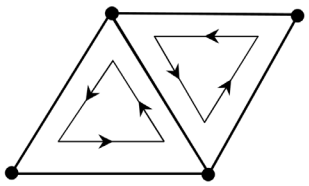
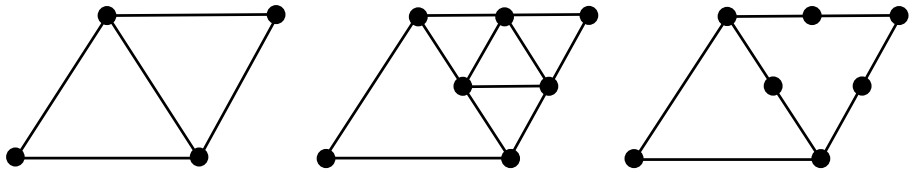


Figure III.6 Orientation of element edges

III.3 Constrained Approximation

So far, the computational mesh was assumed to be regular. Such a mesh does not include any hanging node. In the context of continuous finite elements, hanging nodes can break the H^1-conformity of the solution across the interface between refined and unrefined elements. To address this problem, a technique known as *constrained approximation* [73, 82-88] has been employed in which the function value at a hanging node is constrained by the function values at adjacent nodes such that a continuous solution is obtained across all

element interfaces. This concept can be illustrated by an example. Figure III.7.a shows two triangular elements with linear shape functions. For local h-refinement, the right triangle is subdivided into four similar elements, and for p-enrchiment, it is replaced with a quadratic element, as shown in figures III.7.b and III.7.c, repectively. To set the constraint, the function value at the hanging node is forced to follow the linear profile dictated by the left element at the common edge, for both cases.



(a) Initial Mesh (b) h-refinement (c) p-enrichment

Figure III.7 Example of non-conforming h-refinement and p-enrichment

The difficulty in implementation of the constrained approximation is that in a general *hp*-adaptation, elements with different polynomial orders and different h-refinement levels can locate next to each other, and the implementation should be able to handle all the possible configurations. To alleviate this issue, it is a common practice to limit the number of possible configurations. The limitation employed in this study is the *1-irregularity rule* [87] which limits the maximum difference of h-refinement levels of neighbor elements to one. However, such a limitation has not been imposed for the case of p-enrichment.

Another important consideration regarding the complexity of the implementation is the choice of the shape functions. As mentioned before, hierarchical shape functions, provide a more modular way to achieve non-uniform polynomial order distribution compared to the

Lagrange shape functions. By returning to the above example, the hanging node produced by p-enrichment can be avoided if hierarchical shape functions are employed. Here, the polynomial order of the common edge can be set to one in favor of the linear element on the left and this automatically keeps the continuity of the solution between two elements. By extending this example to a three dimensional case, the convenience obtained by hierarchical basis function is more appreciated considering that several elements may share a common edge. Nevertheless, in the case of non-conforming h-refinement, both Lagrange and hierarchical shape functions generate hanging nodes.

III.3.1 Finding Constraints

The constraint relations are found based on a description in reference [84], although slight modifications have been added to reflect the present implementation details and differences. Using hierarchical shape functions and 1-irregularity rule, in two dimensions, hanging nodes can only take place at the edges between elements with one h-refinement level difference as shown in figure III.8. In this figure, elements Ω^2 and Ω^3 have been generated by h-refinement of the neighbor of element Ω^1. Similarly, small edges e_2, e_3 have been generated by subdivision of edge e_1 and thus they inherit their orientations from that edge. That is if $e_1 = v_1 v_2$, then $e_2 = v_1 v_3$ and $e_3 = v_3 v_2$.

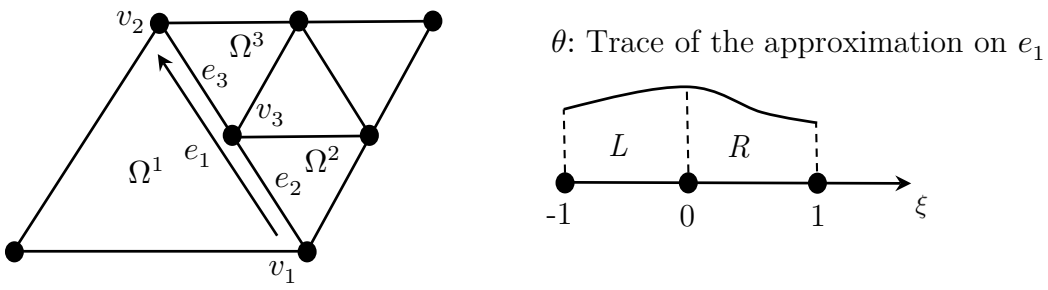


Figure III.8 Constrained approximation with 1-irregularity rule and hierarchical shape functions

In figure III.8, v_3 is assumed to be at the mid-point of e_1 and thus considering $\vec{x}^{e_1}\colon[-1,1] \to \mathbb{R}^2$ to be the map of edge e_1 to the physical space,

$$\begin{aligned} v_1 &= \vec{x}^{e_1}(-1) \\ v_2 &= \vec{x}^{e_1}(1) \\ v_3 &= \vec{x}^{e_1}(0) \end{aligned} \tag{III.17}$$

Edges e_2 and e_3 inherit their polynomial orders from edge e_1, right after their generation. However, in successive adaptation cycles, the polynomial orders of elements may change arbitrarily. Therefore, after each adaptation cycle, it is necessary to update the polynomial order of the edges. To this end, for each edge, the minimum polynomial order is picked from its adjacent elements. In the case of figure III.8,

$$P = P^{e_1} = P^{e_2} = P^{e_3} = \min\{P^b(\Omega^1), P^b(\Omega^2), P^b(\Omega^3)\} \geq 1 \tag{III.18}$$

In following, the solution coefficients or DOFs are denoted by u and those associated with edges e_1, e_2, and e_3 are labeled as:

- Constraining: the vertex coefficients u^{v_1}, u^{v_2} and edge coefficient $u_k^{e_1}$, $2 \leq k \leq P$.
- Constrained: the vertex coefficients u^{v_3} and edge coefficients $u_k^{e_2}$, $u_k^{e_3}$, $2 \leq k \leq P$.

To set a constraint, the constrained coefficients need to be expressed in terms of constraining ones. Toward this, consider $V^P_{[-1,1]}, V^P_{[-1,0]}$ and $V^P_{[0,1]}$ as the spaces of scalar polynomials of order P or lower defined on the subscribed intervals. These spaces are equipped with following polynomial bases:

$$\mathcal{B}^P_{[-1,1]} = \{l_k\}^P_{k=0} = \{l_k(\xi);\ \xi \in [-1,1]; k = 0, \dots, P\} \tag{III.19.a}$$

$$\mathcal{B}^P_{[-1,0]} = \{l^L_k\}^P_{k=0} = \{l_k(1+2\xi);\ \xi \in [-1,0]; k = 0, \dots, P\} \tag{III.19.b}$$

$$\mathcal{B}^P_{[0,1]} = \{l^R_k\}^P_{k=0} = \{l_k(-1+2\xi);\ \xi \in [0,1]; k = 0, \dots, P\} \tag{III.19.c}$$

Now, the trace of the approximate solution on edge e_1 (see figure III.8) can be expressed as,

$$\theta = (\mathbf{u}^{e_1})^T \cdot \mathbf{l}^{e_1} \quad \text{on } [-1,1] \tag{III.20}$$

where $\mathbf{u}^{e_1}$ is the vector of the coefficients associated with edge e_1,

$$\mathbf{u}^{e_1} = [u^{v_1}, u^{v_2}, u^{e_1}_2, \dots, u^{e_1}_P\ ]^T \tag{III.21}$$

and $\mathbf{l}^{e_1}$ is the vector of the functions in $\mathcal{B}^P_{[-1,1]}$,

$$\mathbf{l}^{e_1} = [l_0, l_1, l_2, \dots, l_P]^T \tag{III.22}$$

Similarly, for edges e_2 and e_3,

$$\mathbf{u}^{e_2} = [u^{v_1}, u^{v_3}, u^{e_2}_2, \dots, u^{e_2}_P\ ]^T \tag{III.23.a}$$

$$\mathbf{u}^{e_3} = [u^{v_3}, u^{v_2}, u^{e_3}_2, \dots, u^{e_3}_P\ ]^T \tag{III.23.b}$$

$$\mathbf{l}^{e_2} = [l^L_0, l^L_1, l^L_2, \dots, l^L_P]^T \tag{III.23.c}$$

$$\mathbf{l}^{e_3} = [l^R_0, l^R_1, l^R_2, \dots, l^R_P]^T \tag{III.23.d}$$

To obtain a continuous solution along edge e_1,

$$(\mathbf{u}^{e_2})^T \cdot \mathbf{l}^{e_2} = (\mathbf{u}^{e_1})^T \cdot \mathbf{l}^{e_1} \quad \text{on } [-1,0] \tag{III.24.a}$$

$$(\mathbf{u}^{e_3})^T \cdot \mathbf{l}^{e_3} = (\mathbf{u}^{e_1})^T \cdot \mathbf{l}^{e_1} \quad \text{on } [0,1] \tag{III.24.b}$$

To solve the above equations, the functions in $\mathbf{l}^{e_1}$can be expressed in terms of $\mathbf{l}^{e_2}$ and $\mathbf{l}^{e_3}$. Thus, by defining the transition matrices $[\mathbf{M}_L^P]$ and $[\mathbf{M}_R^P]$ as,

$$\mathbf{l}^{e_1} = [\mathbf{M}_L^P]^T \mathbf{l}^{e_2} \tag{III.25.a}$$

$$\mathbf{l}^{e_1} = [\mathbf{M}_R^P]^T \mathbf{l}^{e_3} \tag{III.25.b}$$

the coefficients of edges e_2 and e_3 are obtained as,

$$\mathbf{u}^{e_2} = [\mathbf{M}_L^P]\mathbf{u}^{e_1} \tag{III.26.a}$$

$$\mathbf{u}^{e_3} = [\mathbf{M}_R^P]\mathbf{u}^{e_1} \tag{III.26.b}$$

Above equations reveal how the constrained DOFs are related to constraining ones.

Transition matrices are $(P+1) \times (P+1)$ matrices. Due to hierarchy of one dimensional shape functions (l_k), these matrices are also hierarchical. That is $[\mathbf{M}_L^{P+1}]$ and $[\mathbf{M}_R^{P+1}]$ are obtained by adding one new row and one new column to the matrices $[\mathbf{M}_L^P]$ and $[\mathbf{M}_R^P]$, respectively. Therefore as suggested in reference [84], for implementation, only the matrices corresponding to the maximum required polynomial order need to be stored. Some entries of these matrices depend on the choice of the constants c_k in equation (III.12). Reference [84] has given these matrices for $P = 3$ with the assumption of $c_0 = -2$. Here, they are given for $P = 5$ and for arbitrary values of c_k, $0 \leq k \leq 3$.

$$[\mathbf{M}_L^5] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{c_0}{4} & 0 & -\frac{c_2}{4} & 0 \\ 0 & 0 & \frac{1}{4} & -\frac{3c_1}{8c_0} & \frac{3c_2}{4c_0} & -\frac{c_3}{2c_0} \\ 0 & 0 & 0 & \frac{1}{8} & -\frac{5c_2}{4c_1} & \frac{23c_3}{16c_1} \\ 0 & 0 & 0 & 0 & \frac{1}{16} & -\frac{7c_3}{32c_2} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{32} \end{bmatrix} \tag{III.27.a}$$

$$[\mathbf{M}_R^5] = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{c_0}{4} & 0 & -\frac{c_2}{4} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & \frac{3c_1}{8c_0} & \frac{3c_2}{4c_0} & \frac{c_3}{2c_0} \\ 0 & 0 & 0 & \frac{1}{8} & \frac{5c_2}{4c_1} & \frac{23c_3}{16c_1} \\ 0 & 0 & 0 & 0 & \frac{1}{16} & \frac{7c_3}{32c_2} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{32} \end{bmatrix} \tag{III.27.b}$$

The first row of $[\mathbf{M}_L^P]$ and second row of $[\mathbf{M}_R^P]$ are simply unit vectors as they point to the unconstrained vertex coefficients u^{v_1} and u^{v_2}, respectively. Also, the second row of $[\mathbf{M}_L^P]$ and the first row $[\mathbf{M}_R^P]$ are equal as both point to the constrained coefficient u^{v_3}. Equation (III.26) reveals how constrained coefficients are related to constraining ones.

III.3.2 Applying Constraints

The constraint relations established in the previous sub-section should be imposed to the linear system shown in equation (II.32). To this end, an algorithm similar to that in

reference [88] has been utilized, although slight modifications has been added to reduce the size of the final linear system.

Based on equation (III.26), constrained DOFs can be explicitly written in terms of unconstrained (or constraining) DOFs. If i is the global node index of a constrained DOF, the corresponding constraint relation can be written as

$$u_i = \sum_{j=1}^{n_c^i} \alpha_j^i u_{r_j^i} \tag{III.28}$$

where n_c^i denotes the number of constraining nodes for node i, and $\mathbf{r}^i$ and $\boldsymbol{\alpha}^i$ are two vectors with the length of n_c^i which include the global indices of constraining nodes, and corresponding factors, respectively. For implementation, the scalar n_c, and vectors $\mathbf{r}$ and $\boldsymbol{\alpha}$ are stored for each hanging node.

A direct result of equation (III.28) is that the constrained DOFs are not required to be solved for in a coupled system like equation (II.32). In other words, the unconstrained DOFs are the actual unknowns of the problem. Now the question is: in presence of hanging nodes, how the linear system of equation (II.32) should be formed? Note that this equation was previously formed with the assumption that all basis functions of the solution space are H^1-conforming. However, the basis functions associated with constrained and constraining DOFs are not conforming where the hanging nodes are present. Figure III.9 illustrates this issue for the configuration previously shown in figure III.8. To solve this problem, a new set of conforming basis functions $\tilde{N}_i$ can be constructed from existing basis functions N_i. Using this algorithm, one can generate as many conforming shape functions as there are unconstrained DOFs in the computational mesh. The main idea is to obtain a linear

combination of existing basis functions for which the trace of the shape functions from adjacent elements match along both sides of the edge with the hanging nodes (i.e. edge e_1 in the present example). Equation (III.25) provides the coefficients required for such a linear combination. In the present example, the elements are assumed to be linear ($P = 1$). Thus using the transition matrices $[\mathbf{M}_L^1]$ and $[\mathbf{M}_R^1]$, equation (III.25) results in

$$[l_0, l_1]^T = \begin{bmatrix} 1 & 1/2 \\ 0 & 1/2 \end{bmatrix} [l_0^L, l_1^L]^T \tag{III.29.a}$$

$$[l_0, l_1]^T = \begin{bmatrix} 1/2 & 0 \\ 1/2 & 1 \end{bmatrix} [l_0^R, l_1^R]^T \tag{III.29.b}$$

In other words,

$$l_0 = l_0^L + \frac{1}{2} l_1^L \quad \text{for } \xi \in [-1,0] \text{ or } \vec{x} \in [v_1, v_3] \tag{III.30,a}$$

$$l_0 = \frac{1}{2} l_0^R \quad \text{for } \xi \in [0,1] \text{ or } \vec{x} \in [v_3, v_2] \tag{III.30.b}$$

$$l_1 = \frac{1}{2} l_1^L \quad \text{for } \xi \in [-1,0] \text{ or } \vec{x} \in [v_1, v_3] \tag{III.30.c}$$

$$l_1 = \frac{1}{2} l_0^R + l_1^R \quad \text{for } \xi \in [0,1] \text{ or } \vec{x} \in [v_3, v_2] \tag{III.30.d}$$

Note that in this example,

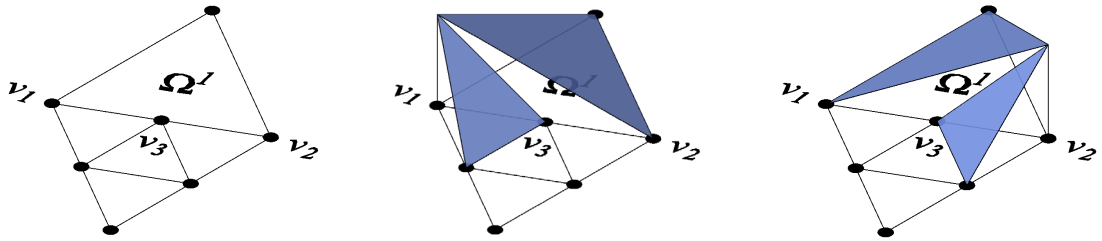
- l_0, l_0^L correspond to the basis function of vertex v_1 which is denoted by N_1,
- l_1, l_1^R correspond to the basis function of vertex v_2 which is denoted by N_2,
- and l_0^R, l_1^L correspond to the basis function of vertex v_3 which is denoted by N_3.

Also, note that vertices v_1, v_2 are constraining nodes and vertex v_3 is the constrained node. Equation (III.30) shows that for constraining nodes, the conforming basis functions can be constructed as follows:

$$\tilde{N}_1 = N_1 + \frac{1}{2} N_3 \tag{III.31.a}$$

$$\tilde{N}_2 = N_2 + \frac{1}{2} N_3 \tag{III.31.b}$$

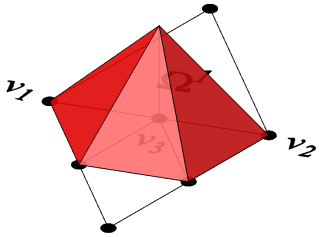
These functions have been depicted in figure III.9. It is worth to mention that similar figures for the case of non-conforming p-adaptation for Lagrange elements can be found in reference [73].



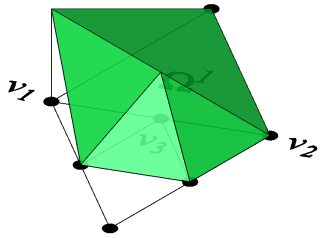
(a) No basis function shown

(b) N_1

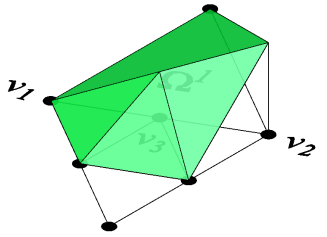
(c) N_2



(d) N_3



(e) $\tilde{N}_1 = N_1 + \frac{1}{2} N_3$



(f) $\tilde{N}_2 = N_2 + \frac{1}{2} N_3$

Figure III.9 Non-conforming and conforming basis functions.

Following reference [88], hereafter the conforming basis functions are referred to as *condensed* basis functions. In general, for the constraining node I, the condensed basis functions $\tilde{N}_I$ can be formed as,

$$\tilde{N}_I = N_I + \sum_{J=1}^{n_d^I} \beta_J^I N_{s_J^I} \tag{III.32}$$

where n_d^I is the number of hanging nodes which are constrained by the node I, and $\mathbf{s}^I$and $\boldsymbol{\beta}^I$ are two vectors with the length of n_d^I which include the global indices of constrained nodes, and corresponding factors, respectively. Careful inspection of equations (III.25) and (III.26) reveals that if $i = s_J^I$ and $I = r_j^i$, then

$$\beta_J^I = \alpha_j^i \tag{III.33}$$

This result implies that using a proper algorithm, the vectors $\mathbf{s}$ and $\boldsymbol{\beta}$ are not required to be stored. In practice, only the vectors $\mathbf{r}$ and $\boldsymbol{\alpha}$ are stored.

Regarding the implementation, the advantage of this method is that the condensed basis functions do not need to be constructed in advance. Instead, for the linear system of equation (II.32), all the original basis functions, including those of constrained and unconstrained DOFs, may be used to calculate the residual vector $\mathbf{R}$ and the Jacobian matrix $[\mathbf{J}]$. Then, the resulting system is modified to form a *condensed system* which only includes the entries corresponding to the unconstrained DOFs.

$$\left[\tilde{\mathbf{J}}^n(\mathbf{q}^{h,n})\right]\Delta\mathbf{q}^{h,n} = -\tilde{\mathbf{R}}^n(\mathbf{q}^{h,n}) \tag{III.34}$$

Such a system is supposed to be similar to that which results from direct utilization of the condensed basis functions. To explain this procedure, considering that the entries of $\mathbf{R}$ are linear in weight functions, the corresponding entries of $\tilde{\mathbf{R}}$ can be obtained by the same linear combination used to form condensed basis functions. That is, similar to equation (III.32),

$$\widetilde{R}_I = R_I + \sum_{J=1}^{n_d^I} \beta_J^I R_{s_J^I} \tag{III.35}$$

Once the condensed residuals for all constraining nodes have been evaluated, the residual of constrained nodes are set to zero. As mentioned earlier, the vectors $\boldsymbol{\beta}$ and $\mathbf{s}$ are not stored and instead the vectors $\boldsymbol{\alpha}$ and $\mathbf{r}$ are utilized in the actual implementation. This process has been shown in algorithm II.1.

Algorithm III.1

```
1: i → 1^st constrained node
2: while (i is not pointing to null)
3:     for j = 1, n_c^i
4:         I = r_j^i
5:         R_I = R_I + α_j^i R_i
6:     end for
7:     R_i = 0
8:     i → next constrained DOF
9: end while
```

Noting that $[\mathbf{J}] = \left[\frac{\partial \mathbf{Res}}{\partial \mathbf{q}^h}\right]$, a similar procedure is carried out for the evaluation of the condensed Jacobian matrix $\left[\tilde{\mathbf{J}}\right]$ by adding the non-zero entries of constrained nodes to those of constraining nodes. However, it should be noted that the connectivity pattern in $\left[\tilde{\mathbf{J}}\right]$ needs to be updated to accommodate the new dependencies that are transferred from constrained nodes to constraining nodes. Then equation (III.28) is used to determine the matrix entries for constrained nodes. Since the constrained nodes are associated with independent DOFs, they can be eliminated from $\left[\tilde{\mathbf{J}}\right]$ using a further static condensation

step. By doing so, the spatial part of $\left[\tilde{\mathbf{J}}\right]$ would be conservative. That is, if the contribution from mass matrix is ignored, due to conservation laws, all the column sums should be zero. The conservative property of the constraint approximation technique developed in the current work has been verified with the condensed Jacobian matrix in several cases which include multiple levels of h- and hp-refinements.

As should be apparent at this point, the use of constraint approximation greatly simplifies the incorporation of h- and hp-adaptive refinement into existing high-order finite-element schemes.

CHAPTER IV

ADAPTATION METHODOLOGY

In this study, output-based and feature-based adaptations have been used for steady-state and unsteady problems, respectively. In this chapter the details of these methods are described.

IV.1 Output-Based Adaptation

Output-based (or adjoint-based) adaptation methods are particularly useful when a specific functional output (e.g. lift or drag) is the main objective of the simulation. These methods target the desired functional output and iteratively adapt the mesh such that the final adapted mesh ensures a prescribed precision for the computed value of the functional output. For this purpose, the sensitivity of the functional output with respect to residual of the flow (primal) equations is calculated in the form of an adjoint (dual) variable. Then the local error is estimated by the inner product of the residual of the flow and the adjoint variable. Having the distribution of the local errors, regions with high errors are chosen for refinement. Using this method, only those regions in the mesh that directly affect the computed value of the functional output are refined. A major benefit of such an approach is that areas that are not intuitively recognizable by the user are captured by the error

estimation mechanism. This is particularly useful for hyperbolic equations for which, the propagation of information cannot be trivially anticipated. The other benefit is that the estimated error may also be used to estimate a correction to the functional value on the current mesh. It should be noted, however, that the final adapted mesh will only be suitable for the chosen functional. That is, an optimal mesh for the lift is not guaranteed to provide the prescribed precision for the drag estimation.

IV.1.1 Discrete Adjoint-Based Error Estimation

The presented formulation is an abbreviated derivation based on the approach developed by Venditti and Darmofal [61, 63, 64] which was initially utilized in finite volume schemes [65, 67]. In recent years, this approach has been extensively utilized within DG finite-element schemes (e.g. see references [16, 20, 21, 23, 46, 50]). Here, we utilize this approach in a Petrov-Galerkin discretization for compressible turbulent flows.

Consider a coarse mesh $\mathcal{T}^H$ with low polynomial order P, as an affordable finite-element mesh to start the calculations. The objective is to estimate a functional $\mathcal{J}(\mathbf{q})$, where $\mathbf{q}$ is the steady-state solution to the PDE under consideration (see equation (II.8.a)). Let $\mathbf{q}^H$ to be the discrete solution on $\mathcal{T}^H$, and $\mathcal{J}^H(\mathbf{q}^H)$ to be the corresponding approximation of the functional. Also, consider $\mathcal{T}^h$ as a globally h-refined/p-enriched finite-element mesh which is assumed to be uniformly embedded within $\mathcal{T}^H$, except at the boundaries where it needs to conform to the actual geometry. It is desired to estimate $\mathcal{J}^h(\mathbf{q}^h)$ without solving the discrete problem on $\mathcal{T}^h$. To this end, $\mathcal{J}^h(\mathbf{q}^h)$ can be expanded about the coarse mesh solution in a Taylor series as

$$\mathcal{J}^h(\mathbf{q}^h) = \mathcal{J}^h(\mathbf{q}^h_H) + \left.\frac{\partial \mathcal{J}^h}{\partial \mathbf{q}^h}\right|_{\mathbf{q}^h_H} (\mathbf{q}^h - \mathbf{q}^h_H) + \text{HOT} \tag{IV.1}$$

where $\mathbf{q}^h_H$ is a projected solution from the coarse mesh to the fine mesh via a projection operator as

$$\mathbf{q}^h_H \equiv I^h_H \mathbf{q}^H \tag{IV.2}$$

Also, $\left.\frac{\partial \mathcal{J}^h}{\partial \mathbf{q}^h}\right|_{\mathbf{q}^h_H}$ refers to the sensitivity of the functional with respect to the fine level flow variables evaluated at $\mathbf{q}^h_H$. By ignoring Higher Order Terms (HOT), all terms on the left hand side of the equation (IV.1) are explicitly computable, with the exception of the $\mathbf{q}^h$. Since $\mathbf{q}^h$ is assumed to be a steady-state solution, by ignoring time terms in equation (II.30) we have

$$\mathbf{R}^h(\mathbf{q}^h) = 0 \tag{IV.3}$$

and thus by an expansion about the coarse mesh solution as,

$$\mathbf{R}^h(\mathbf{q}^h) = \mathbf{R}^h(\mathbf{q}^h_H) + \left[\frac{\partial \mathbf{R}^h}{\partial \mathbf{q}^h}\right]_{\mathbf{q}^h_H} (\mathbf{q}^h - \mathbf{q}^h_H) + \text{HOT} = 0 \tag{IV.4}$$

the $\mathbf{q}^h - \mathbf{q}^h_H$ can be estimated as

$$\mathbf{q}^h - \mathbf{q}^h_H \approx -\left[\frac{\partial \mathbf{R}^h}{\partial \mathbf{q}^h}\right]_{\mathbf{q}^h_H} \mathbf{R}^h(\mathbf{q}^h_H) \tag{IV.5}$$

By substitution of equation (IV.5) into equation (IV.1), an approximation for functional output on the fine mesh may be expressed as

$$\mathcal{J}^h(\mathbf{q}^h) \approx \mathcal{J}^h(\mathbf{q}^h_H) - \left.\frac{\partial \mathcal{J}^h}{\partial \mathbf{q}^h}\right|_{\mathbf{q}^h_H} \left[\frac{\partial \mathbf{R}^h}{\partial \mathbf{q}^h}\right]_{\mathbf{q}^h_H} \mathbf{R}^h(\mathbf{q}^h_H) \tag{IV.6}$$

At this point, the discrete adjoint variable $\boldsymbol{\lambda}^h$ on the fine mesh can be defined as,

$$\left[\frac{\partial \mathbf{R}^h}{\partial \mathbf{q}^h}\right]^T \boldsymbol{\lambda}^h = \left(\frac{\partial \mathcal{J}^h}{\partial \mathbf{q}^h}\right)^T \tag{IV.7}$$

which expresses the adjoint variable as the sensitivity of the functional $\mathcal{J}$ with respect to the local spatial residual $\mathbf{R}$. Using this definition, equation (IV.6) is simplified to

$$\mathcal{J}^h(\mathbf{q}^h) \approx \ \mathcal{J}^h(\mathbf{q}_H^h) - \left(\boldsymbol{\lambda}^h|_{\mathbf{q}_H^h}\right)^T \mathbf{R}^h(\mathbf{q}_H^h) \tag{IV.8}$$

Since $\boldsymbol{\lambda}^h|_{\mathbf{q}_H^h}$ is defined on the fine mesh, its direct evaluation through equation (IV.7) can be as expensive as the primal solution. Alternatively, the adjoint variable can be calculated on the coarse mesh using

$$\left[\frac{\partial \mathbf{R}^H}{\partial \mathbf{q}^H}\right]^T_{\mathbf{q}^H} (\boldsymbol{\lambda}^H|_{\mathbf{q}^H}) = \left(\frac{\partial \mathcal{J}^H}{\partial \mathbf{q}^H}\right)^T_{\mathbf{q}^H} \tag{IV.9}$$

and then, like equation (IV.2), a projection operator may be employed to project the adjoint solution on the fine mesh as

$$\boldsymbol{\lambda}_H^h \equiv J_H^h(\boldsymbol{\lambda}^H|_{\mathbf{q}^H}) \tag{IV.10}$$

Using $\boldsymbol{\lambda}_H^h$, a *computable* estimate of the output function on the fine mesh is obtained as

$$\mathcal{J}^h(\mathbf{q}^h) \approx \mathcal{J}^h(\mathbf{q}_H^h) \underbrace{-(\boldsymbol{\lambda}_H^h)^T \mathbf{R}^h(\mathbf{q}_H^h)}_{\text{Computable Correction}} \tag{IV.11}$$

where the computable correction is expressed as the inner product of the adjoint solution and the primal residual error.

IV.1.2 Adaptation Criteria and Element Picking Strategy

The computable correction gives a spatial distribution of nodal error and thus may be used to navigate the adaptation process. As a possible choice, an elemental error indicator may be defined as

$$\varepsilon_e = \sum_{l(e)} c_{l(e)} \left| [(\boldsymbol{\lambda}_H^h)^T \mathbf{R}^h(\mathbf{q}_H^h)]_{l(e)} \right| \tag{IV.12}$$

In above expression, the index e refers to an element in the coarse mesh, and the index $l(e)$ refers to the local fine-mesh modes embedded in the coarse-mesh element. The coefficient c is used to determine the share of nodal errors that contribute to the elemental error. For the sake of simplicity, in this study, nodal errors are equally distributed among coarse-mesh elements which surround the node. Note that due to the use of absolute value function in equation (IV.12), the total error is a conservative measure of the computable error.

Having the elemental errors from equation (IV.12), a refinement list is formed that sorts the elements in the descending order of their contribution to the total error. Then elements are picked from the top of the list until the sum of the picked error exceeds a prescribed fraction θ_e of the total error. This approach has been used, for example, in reference [68], although here to avoid excessive refinements at final adaptation cycles, the number of picked elements at each adaptation cycle is limited by another prescribed fraction θ_n. For all numerical test cases in the present work, $\theta_e = 75\%$ and $\theta_n = 20\%$.

IV.1.3 Projection Operators

Here the fine the mesh $\mathcal{T}^h$ is formed by uniform p-enrichment of the coarse mesh $\mathcal{T}^H$. Thus, due to the hierarchy of the shape functions, I_H^h and J_H^h can be naturally obtained by initializing the new modes to zero. Although this approach is simple and computationally inexpensive, the values of adjoint variable at the higher dimension space are totally ignored and the error is obtained by the inner product of the coarse level adjoint and the fine level flow residual. A better approach is to utilize a projection method that results in a smoothed

adjoint solution on the fine mesh. A common technique is to reconstruct the solution on the fine mesh using a patch-wise least square method similar to that in references [20, 46]. However, the implementation of this method on non-conforming meshes is not trivial and thus we have reserved this practice for the future work. Nevertheless, as it will be shown in the numerical results, output-based adaptation in this study has demonstrated at least a satisfactory performance.

IV.1.4 Functional Modification for Weakly Imposed Boundary Conditions

When weak boundary conditions are used, if the functional of interest depends on the viscous forces, equation (II.28) is used to calculate the viscous forces on the boundary edges. It will be shown in results section that this approach is required to obtain a smooth adjoint solution in vicinity of Dirichlet-type boundaries.

IV.1.5 Deciding Between p-enrichment or h-refinement

For hp-adaptation, the elements identified for refinement need to be divided into two groups: First, p-enrichment group which includes elements with smooth solution, and second, h-refinement group which includes the elements with large gradients. For this grouping, the discontinuity sensor given in equation (II.42) has been employed. Such approach has been previously used by Wang and Mavriplis [20]. Note that this discontinuity sensor was initially introduced in section II.5 for the shock capturing. The grouping is done using following criterion,

$$\begin{cases} F_e \geq S_0 - \kappa, & \text{h} - \text{refinement} \\ F_e < S_0 - \kappa, & \text{p} - \text{enrichment} \end{cases} \tag{IV.13}$$

where $s_0 = 1/P^4$, and κ is a tuning parameter which is set to 6 in the present work. In addition, all the state variable are used to calculate the discontinuity sensor, and finally, the maximum values are chosen.

IV.1.6 Enhanced h-refinement (EHR)

After grouping elements, one should decide about how many levels of p-enrichment or h-refinement should be done on each element. A common approach in hp-adaptation is to use one level for both p-enrichment and h-refinement at each adaptation cycle. In the present work, we use one level for p-enrichment. However, for h-refinement, depending on the value of the discontinuity sensor F_e, we may use either one or two levels. In particular,

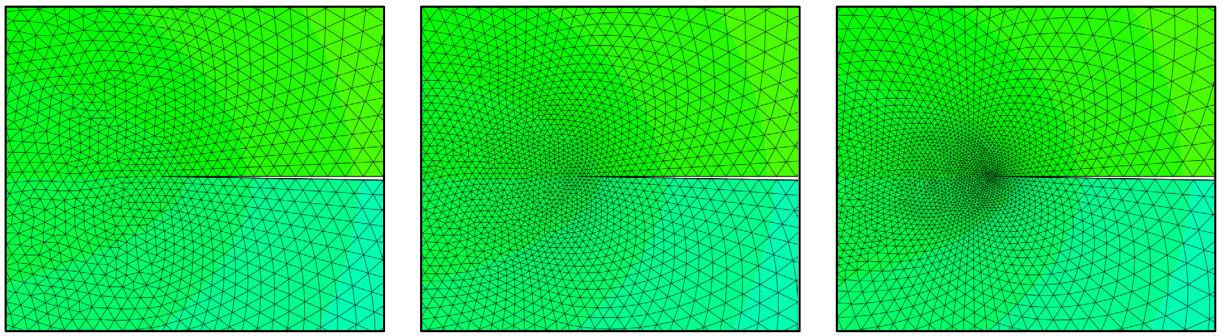
$$\begin{cases} F_e \geq \theta_s \max\limits_e F_e & 1 \text{ level of } \mathrm{h}-\text{refinement} \\ F_e < \theta_s \max\limits_e F_e & 2 \text{ levels of } \mathrm{h}-\text{refinement} \end{cases} \tag{IV.14}$$

where $\theta_s = 1.1$ is a tuning parameter. Note that F_e is a negative number and its lower values indicate smoother solutions. In most cases studied here, F_e was observed to be between -2 and -16.

We refer to 2 levels of h-refinement as *enhanced h-refinement (EHR).* As it will be shown in the numerical results, EHR is particularly effective when a singularity is present in the solution. In such a situation, the error is rapidly reduced by concentrating nodes near the singularity.

The idea behind EHR can be explained using the Reentrant corner problem. Consider to solve the Laplace equations using the Method of Manufactured Solutions (MMS) on three meshes that are shown in figure IV.1. These meshes have similar grid densities at

outer boundaries and different grid densities near the corner. Assuming that all the elements in this figure are linear, by uniform h-refinement of each mesh, the error is expected to reduce by an order of 2. However, due to the presence of singularity, the error is reduced by an order of 0.5, as shown in figure IV.2. Note that although meshes with more clustering of the nodes near the corner have lower values of the error, the slope of the error reduction is the same for all three meshes. This observation suggest that to reach the nominal order of accuracy, more than one level of refinement is required in vicinity of the singularity point, as shown by the dashed line in figure IV.2. The proposed approach in equation (IV.14) tries to detect regions with non-smooth solutions and enhance the h-refinement on those regions.



(a) Mesh 1 (b) Mesh 2 (c) Mesh 3

Figure IV.1 Three meshes for Reentrant corner problem

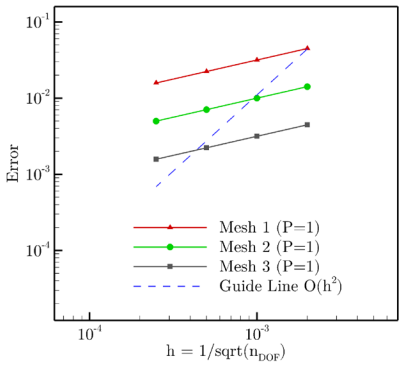


Figure IV.2 Error convergence for Reentrant corner problem

IV.2 Feature Based Adaptation

In the present work, a feature-based adaptation has been developed for unsteady problems. The main purpose is to investigate the possibility of dynamic adaptation in tracing evolving flow features. Such a problem requires a derefinement mechanism. Otherwise, the computation will be hindered by an over-refined mesh. As mentioned earlier, a non-conforming mesh adaptation method has been used in this study. Such a method can be efficiently implemented using a parent-children data structure. Specifically, a tagged element for h-refinement is stored as the parent and the newly generated elements are stored as the children. Each child can also be a parent to obtain a multi-level refinement. Using this method, derefinement can be simply done by deleting the elements at the upper level. Parent-children structure is not required for p-modifications, and thus polynomial degree of each element is changed locally.

Toward developing an adaptation criterion, the following error indicator ε_e is defined based on the magnitude of the velocity gradient:

$$\varepsilon_e = \int_{\Omega^e} \left[\sum_{i=1}^{n_{sd}} \sum_{j=1}^{n_{sd}} \left(\frac{\partial u_i}{\partial x_j} \right)^2 \right]^{\frac{1}{2}} d\Omega \tag{IV.15}$$

Then, the mesh is adapted until the following statement holds for all of the elements:

$$\varepsilon_e |\Omega^e| (P^e)^2 \leq t \frac{\varepsilon_{tot}}{n_{el}} \tag{IV.16}$$

where ε_{tot} is the total integral of adaptation parameter over the computational mesh and t is a tuning parameter. In particular, the following criterion is used to pick the elements for refinement/derefinement:

$$\begin{cases} |\Omega^e| > \Omega_D^e & \text{refinement} \\ |\Omega^e| < 4\Omega_D^e & \text{derefinement} \end{cases} \tag{IV.17}$$

where

$$\Omega_D^e = t \frac{\varepsilon_{tot}}{n_{el} \varepsilon_e (P^e)^2} \tag{IV.18}$$

CHAPTER V

NUMERICAL RESULTS

In this chapter, several numerical examples are given to demonstrate the performance of the present adaptive high-order finite-element flow solver. These examples include application of adjoint-based adaptation in steady-state problems and feature-based adaptation in an unsteady problem. In steady-state problems, only refinement mechanisms are employed while in the unsteady problem, derefinement is also used and a dynamic adaptation is presented.

V.1 Output-Based Adaptation in Steady-State Flows

The developed adjoint-based adaptive algorithm is applied to six numerical examples including four subsonic flows, and two transonic flows. For four examples, the NACA0012 airfoil is used and the exact geometry of the airfoil is calculated based on a modified formula from Turbulence Modeling Resource (TMR) website [100], which is supported by NASA Langley Research Center for verifying and validating turbulence models. This formula results in a sharp trailing edge:

$$\begin{aligned} y = \ \pm\, 0.594689181\big(&0.298222773\ \sqrt{x} - 0.127125232\ x \\ -\ &0.357907906\ x^2\ +\ 0.291984971\ x^3 - 0.105174606\ x^4\big) \end{aligned} \tag{V.1}$$

For all examples, the geometry is represented using Q3 elements. For all hp-adaptations, the computation starts with P1 elements and the maximum polynomial order is set to 3. Then, if an element requires more p-enrichment, it will be h-refined, regardless of the smoothness of the solution in the element.

V.1.1 Clarification of Figures

In order to clarify the figures presented in this section, it is useful to mention the following notes:

1- In some figures, the functional output is plotted against the number of degrees of freedom, n_{DOF}. In these figures, n_{DOF} represents the number of unconstrained nodes in the calculation; it does not represent the total number of unknowns in the computational domain, which also includes the number of variables solved for at each node.

2- In some figures, the error in the functional is plotted against a mesh spacing metric, h, determined strictly based on number of nodes.

3- In order to study the convergence behaviors, in most cases, the adaptation process has been continued to reach a precision more than that normally required for engineering applications. Thus, the final adapted mesh may be too fine to be properly depicted. In such cases, instead of the final adapted mesh, an intermediate adapted mesh may be shown. To ensure the sufficient resolution, the percentage of relative change in the functional output obtained by the shown mesh, $\Delta_{\mathcal{J}}$, is reported in the figure's caption. For example, $\Delta_{\mathcal{J}} = 0.1\%$ means

that the relative difference in the computed functional on the shown mesh and its previous one is 0.1 percent.

4- In some figures, the corrected functional output is plotted. This corrected value is calculated based on equation (IV.11).

V.1.2 Subsonic Inviscid Flow over NACA0012

The first example shows an application of adjoint-based adaptation for subsonic inviscid flow over a NACA0012 airfoil at free stream Mach number of 0.5 and the angle-of-attack of 2 degrees. The functional of interest is the drag coefficient, C_D, which is calculated based on the static pressure forces. This problem has been presented to demonstrate the convergence behavior of the proposed adaptive approach in an inviscid flow (see also reference [50]). Moreover, due to presence of the singularity at the trailing edge, the usefulness of the EHR is illustrated. The studied cases include h-refinement on uniform P1, P2, and P3 elements, and hp-adaptation using P1 to P3 elements. In all cases, results with and without EHR are compared.

Figure V.1 shows the initial computational mesh which contains 2962 triangular elements. With uniform P1 discretization, this mesh includes 1551 DOFs. The outer boundary has been placed at a distance of 100 chord lengths away from the airfoil. Figure V.2 compares regular h-refinement and EHR in vicinity of the trailing edge for the 4^{th} h-adapted mesh on P3 elements. In this figure, bold lines show the elements on the initial mesh. As seen, EHR forces significantly more refinements at the trailing edge. The actual usefulness of EHR is revealed by examining the error convergence histories. In order to

compute the error, a reference value, $C_{D,Ref}$, has been taken from an hp-adapted solution in which the absolute difference in the functional value between last two adapted mesh has been less than 10^{-9}. Figure V.3 shows the Mach number contours on the mentioned hp-adapted mesh. Also, figure V.4 shows the contours of the x- and y-velocity adjoint variables on that mesh. As seen, a smooth adjoint solution has been obtained.

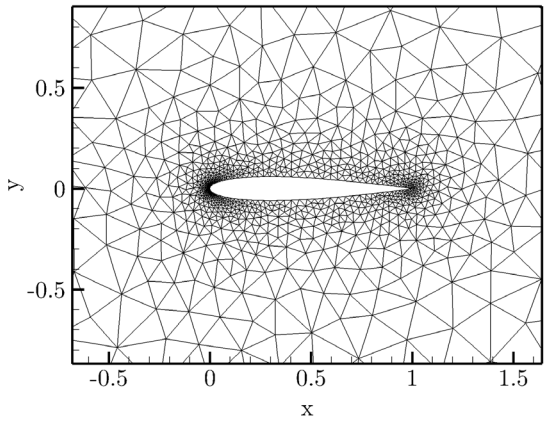
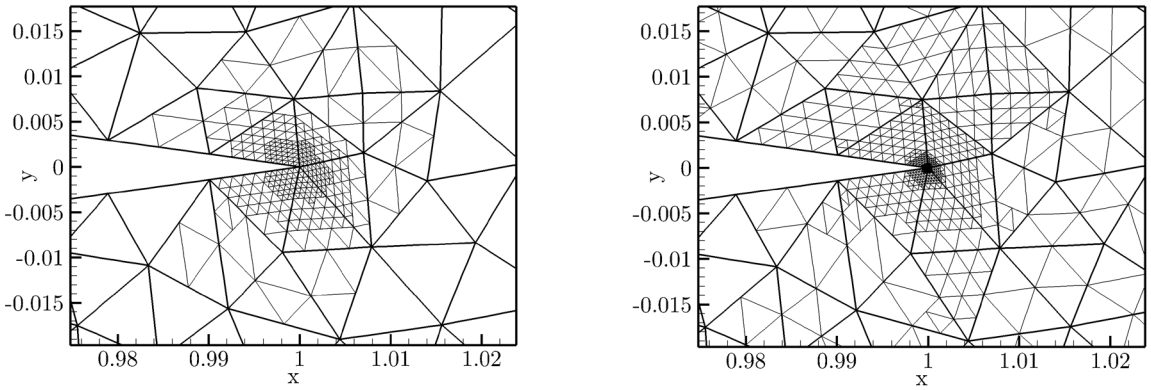


Figure V.1 Initial meshes for the subsonic inviscid flow over NACA0012



(a) Trailing Edge of 4th h-adapted mesh with P3 elements ($\Delta_{\mathcal{J}} = 2.5\%$)

(b) Trailing Edge of 4th EHR mesh with P3 elements and EHR ($\Delta_{\mathcal{J}} = 0.3\%$)

Figure V.2 Comparison of regular h-refinement and EHR in vicinity of the trailing edge for the subsonic inviscid flow over NACA0012 at $M_\infty = 0.5$ and $\alpha = 2°$, Bold lines: elements of the initial mesh

Figures V.5 and V.6 show the convergence history of the drag coefficient error for all of the mentioned adaptive refinements. In each sub-figure, the results with and without EHR have been compared and in order to show the convergence order, auxiliary guidelines have been plotted. In all cases, except for the h-adaptation on P1 elements, EHR results in a notably steeper error reduction slope. For example, in the h-adaptation case on P2 elements, by employing EHR, a convergence order of 6 has been achieved whereas without EHR the convergence order is limited to 3. This corresponds to the same order as that achieved by h-adaptation on P1 elements. This clearly shows the effect of the singularity in limiting the performance of the higher order elements.

The reason of the failure of EHR for P1 elements is that the discontinuity sensor has not been successful in detecting the trailing edge. As discussed in section II.5, the discontinuity sensor is defined based on the rate of decay of the expansion coefficients in the hierarchical basis used to expand the discrete solution. However, for a linear expansion, there are not enough modes to calculate the rate of the decay. Thus, for P1 elements, the average of state variable was used instead of $\hat{q}$ in equation (II.42). This study showed that this approach is not reliable.

Figure V.7 summarizes the error convergence histories. As expected, by increasing the polynomial order, the order of convergence increases. Also, it is seen that hp-adaptation shows the superior performance in terms of the number of DOFs. Notably, the hp-adaptation has achieved the same order of convergence as P3 elements. This is because in the final adaptation cycles, most of the computational mesh has been covered by P3 elements.

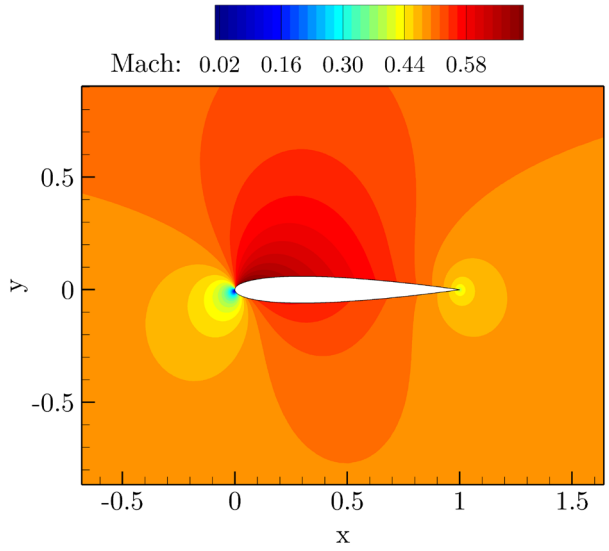
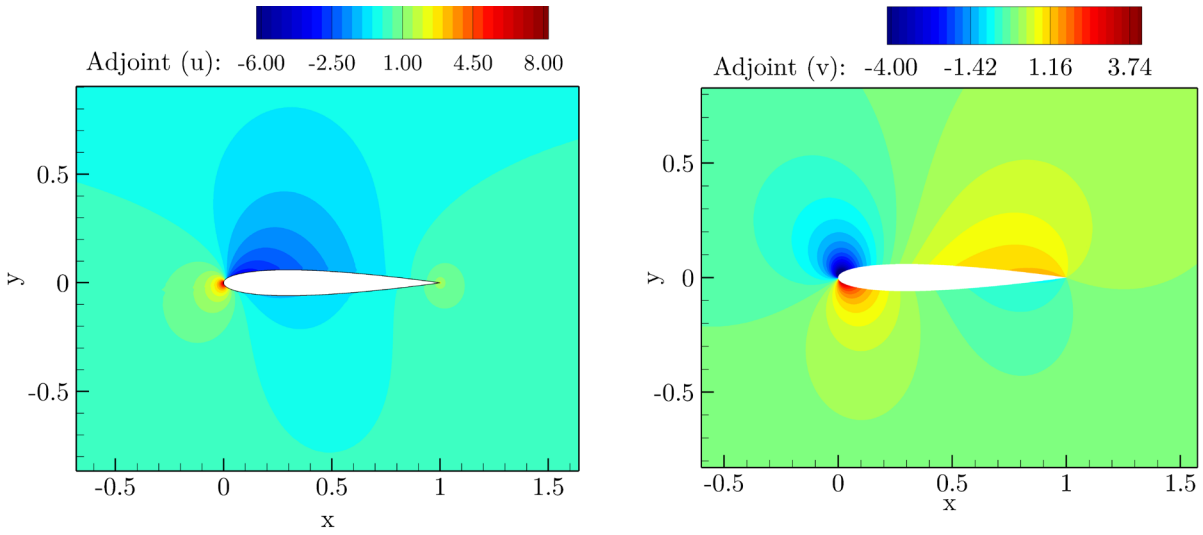


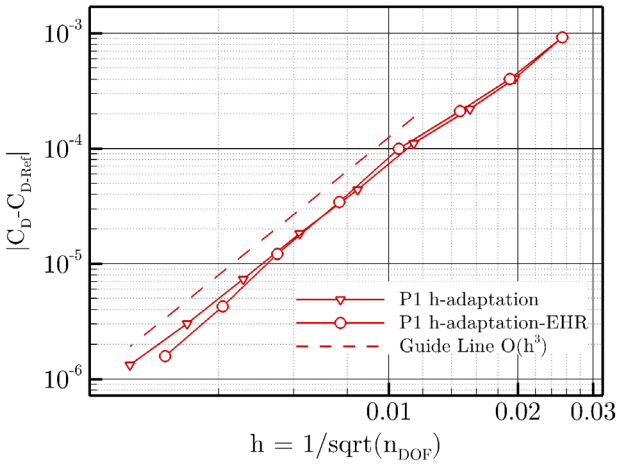
Figure V.3 Mach number contours for the subsonic inviscid flow over NACA0012 at $M_\infty = 0.5$ and $\alpha = 2°$



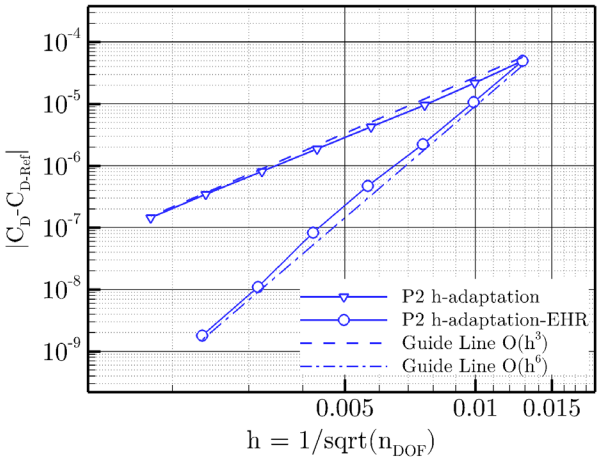
(a) x-velocity adjoint

(b) y-velocity adjoint

Figure V.4 Drag-based adjoint solution for the subsonic inviscid flow over NACA0012 at $M_\infty = 0.5$ and $\alpha = 2°$

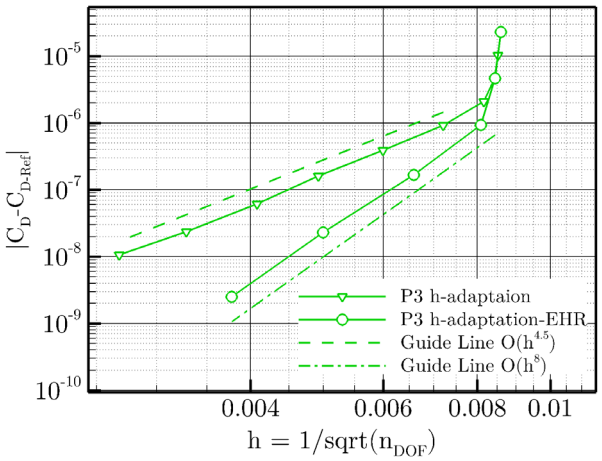


(a) h-adaptation on P1 elements

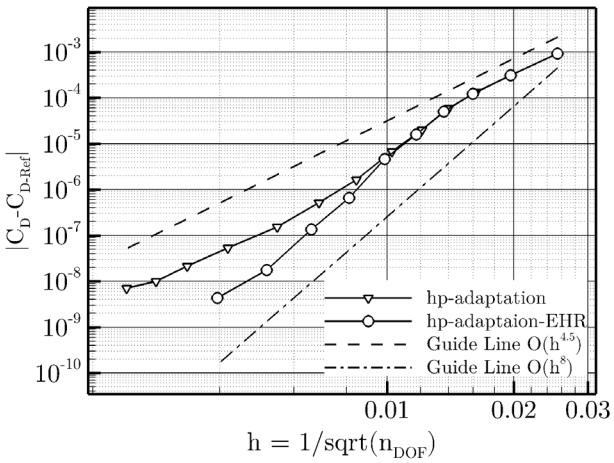


(b) h-adaptation on P2 elements

Figure V.5 Convergence of drag coefficient in drag-based h-adaptation on P1 and P2 elements for the subsonic inviscid flow over NACA0012 at $M_\infty = 0.5$ and $\alpha = 2°$

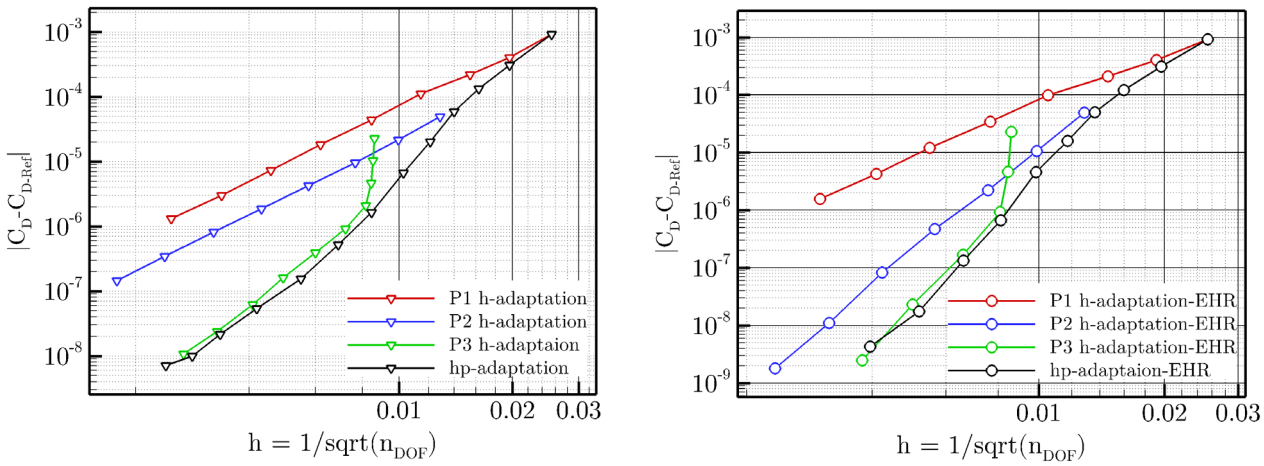


(a) h-adaptation on P3 elements



(b) hp-adaptation using P1 to P3 elements

Figure V.6 Convergence of drag coefficient in drag-based h-adaptation on P3 elements and hp-adaptation using P1 to P3 elements for the subsonic inviscid flow over NACA0012 at $M_\infty = 0.5$ and $\alpha = 2°$



(a) h- and hp-adaptation

(b) h- amd hp adaptation with EHR

Figure V.7 Comparison of convergence of drag coefficient in drag-based h- and hp-adaptations for the subsonic inviscid flow over NACA0012 at $M_\infty = 0.5$ and $\alpha = 2°$

V.1.3 Subsonic Laminar Flow over NACA0012

The second numerical example shows an application of adjoint-based adaptation for subsonic laminar flow over a NACA0012 airfoil at free stream Mach number of 0.5, angle-of-attack of 1 degree, and Reynolds number, based on the airfoil chord, of 5,000. This problem has been widely used for the similar purpose (see e.g. reference [50, 68]). Due to the no-slip condition on the airfoil surface, it provides a test case to examine the behavior of the adjoint solution in presence of Dirichlet boundary condition. This example demonstrates the importance of the weak boundary conditions and the modified functional. Here, the functional of interest is the total drag coefficient which includes both pressure and viscous stresses.

The initial computational mesh, shown in figure V.8, contains 4524 triangular elements. With uniform P1 discretization, this mesh includes 2340 DOFs. The outer

boundary has been placed at a distance of 100 chord lengths away from the airfoil, and the normal spacing to the wall is 0.0002.

Studied cases include h- and hp-adaptations which both start from initial mesh with P1 elements. Figure V.9 illustrates the 5th h-adapted mesh and the 4th hp-adapted mesh. As seen, both adaptation mechanisms resolve similar regions of the flow field to increase the predictive accuracy of the drag coefficient. In the smooth regions of the flow field, P2 and P3 elements have been almost exclusively employed. This is most evident along the stagnation streamline. In contrast, the wake region has been mostly refined by h-adaptation.

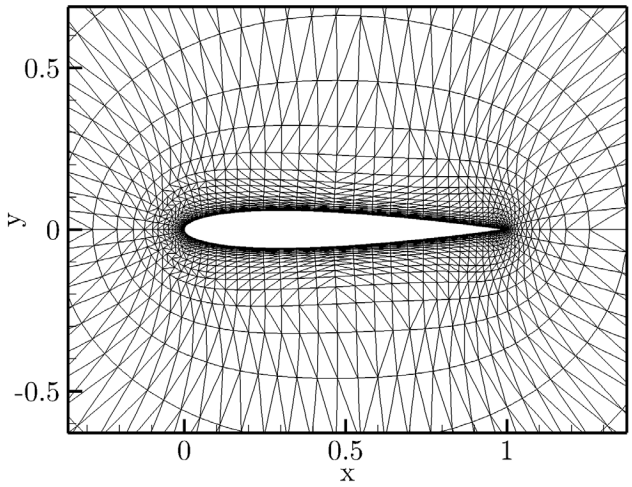
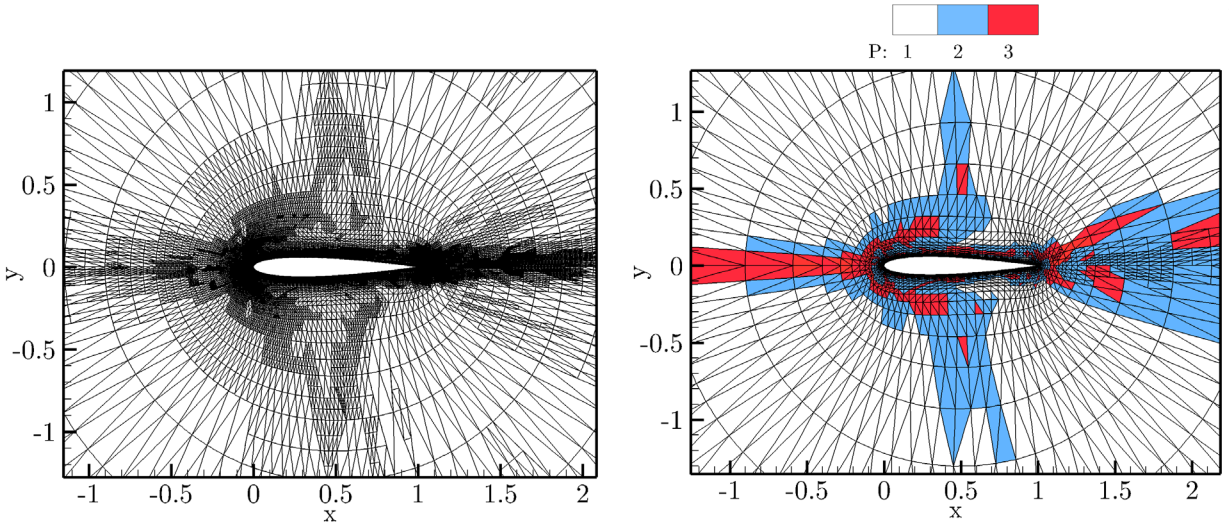


Figure V.8 Initial mesh for the subsonic laminar flow over NACA0012

Mach number contours are depicted in figure V.10, where the wake can clearly be seen. Since drag is strongly dependent on the wake, comparison of the Mach number contours with the adaptive meshes in figure V.9 demonstrates that the adjoint-based adaptive method highly resolves this region of the flow field.



(a) 5th h-adapted mesh ($\Delta_{\mathcal{J}} = 0.3\%$) (b) 4th hp-adapted mesh ($\Delta_{\mathcal{J}} = 0.2\%$)

Figure V.9 Adapted meshes for the subsonic laminar flow over NACA0012 at $M_\infty = 0.5$, $\alpha = 1°$, and $Re = 5{,}000$

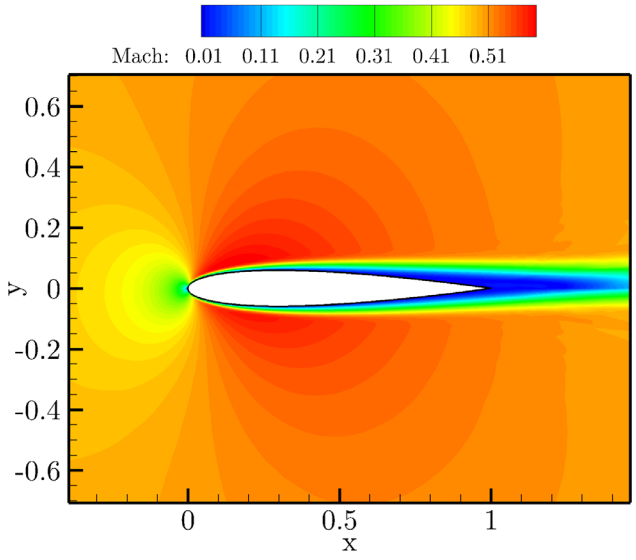
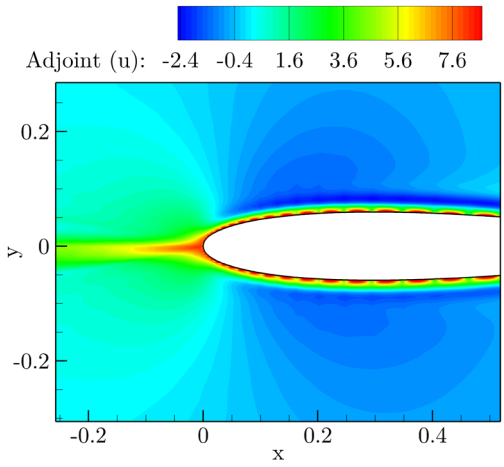


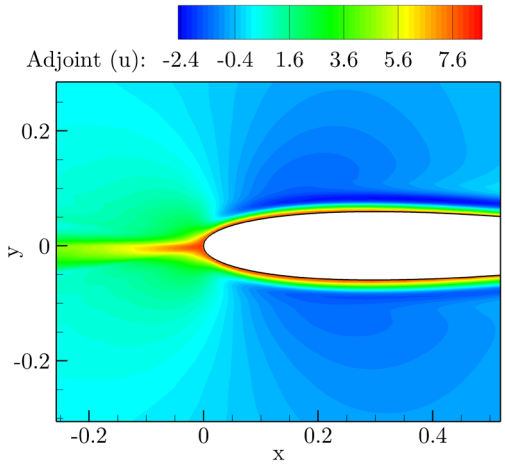
Figure V.10 Mach number contours for the subsonic laminar flow over NACA0012 at $M_\infty = 0.5$, $\alpha = 1°$, and $Re = 5{,}000$

Figure V.11 depicts the contours of the x-velocity adjoint variable obtained on the initial mesh with P2 elements. Close inspection of figure V.11.a illustrates the non-smooth behavior of the adjoint in the absence of weak boundary conditions and a modified functional. This behavior is present if either strong boundary conditions which are not dual consistent, or the original functional is used in the calculation of the adjoint. Furthermore, it should be noted that this non-smooth behavior can be readily seen for this low-Reynolds number flow. For high-Reynolds number flows, with very thin boundary layers, this behavior may not be visible on inspection, however, it would still be present. Figure V.11.b demonstrates that with the proper implementation of weak boundary conditions and modified functional, the adjoint variable is smooth along the airfoil surface.

The convergence history of the h- and hp-adaptation based on the total drag coefficient are shown in figure V.12. As seen in figure V.12.a, the drag coefficient converges faster for the hp-adaptive technique. Here again, the reference value, $C_{D,Ref}$, has been taken from an hp-adapted solution in which the absolute difference in the functional value between last two adapted mesh has been less than 10^{-9}. Moreover, figure V.12.b illustrates the rate of convergence to the reference drag for both adaptation strategies. Based on the guideline slopes, the hp-adaption has a convergence rate of nearly four orders greater than h-adaptation. Thus, in terms of number of DOFs, the hp-adaptive technique results in a more efficient means of obtaining greater accuracy.

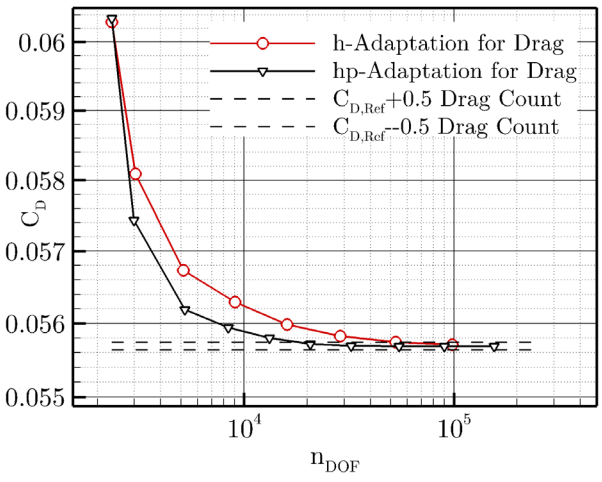


(a) Non-smooth adjoint due to lack of weak implementation of boundary condition or modified functional output

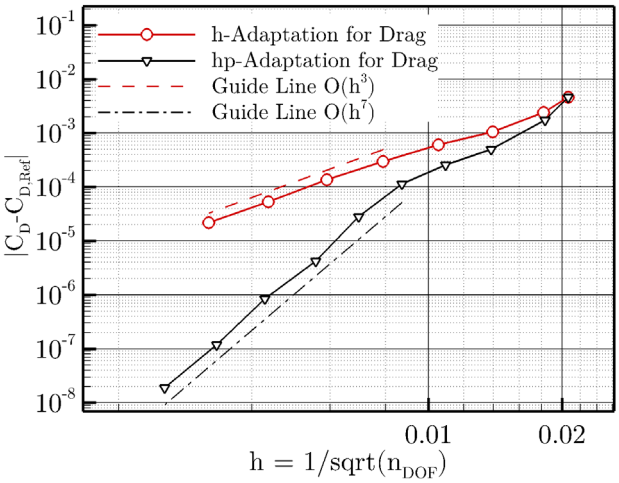


(b) Smooth adjoint obtained by simultaneous use of weak boundary conditions and modified functional output

Figure V.11 Comparison of adjoint solutions for the subsonic laminar flow over NACA0012 at $M_\infty = 0.5$, $\alpha = 1°$, and $Re = 5{,}000$



(a) Drag coefficient



(b) Error in drag coefficient

Figure V.12 Convergence of drag coefficient in drag-based h- and hp-adaptation for the subsonic laminar flow over NACA0012 at $M_\infty = 0.5$, $\alpha = 1°$, and $Re = 5{,}000$

V.1.4 Subsonic Turbulent Flow over NACA0012

The third example shows an application of adjoint-based adaptation for subsonic turbulent flow over the NACA0012 airfoil at free stream Mach number of 0.15, angle-of-attack of 10 degrees, and Reynolds number, based on the airfoil chord, of six million. These conditions have been specified by the AIAA Technical Discussion Group on Solver Technologies for Turbulent Flows. Results from the present study are compared with those from the Turbulence Modeling Resource (TMR) website [100]. In a previous study [4], we compared the forces, moments, pressure distributions, skin friction, and profiles of velocity and turbulence working variable of the Spalart-Allmaras turbulence model between the developed SUPG scheme and the finite-volume solutions obtained using FUN3D [108-110] and CFL3D flow solvers. It was demonstrated that for most of the comparisons, the proposed SUPG scheme obtains similar results as finite-volume schemes but using less DOFs, and also SUPG solutions demonstrate significantly less dissipation of the wake profiles downstream of the airfoil. In the present work, lift and drag coefficients are obtained using h- and hp-adaptive solutions. In each case, FUN3D results on meshes containing 3704, 14576, 57824, 230336, 919424, 3673856, and 14689281 DOFs, are used for comparison. It should be noted that the FUN3D results have been obtained using second-order accurate discretization of the convective terms in the turbulence model. Moreover, in order to calculate the error, the solution on the finest mesh is used as the reference solution. Note that the error of the reference solution is assumed to be zero and thus it is not shown in the error plots with logarithmic scale. Similar to previous examples, the computed errors are all

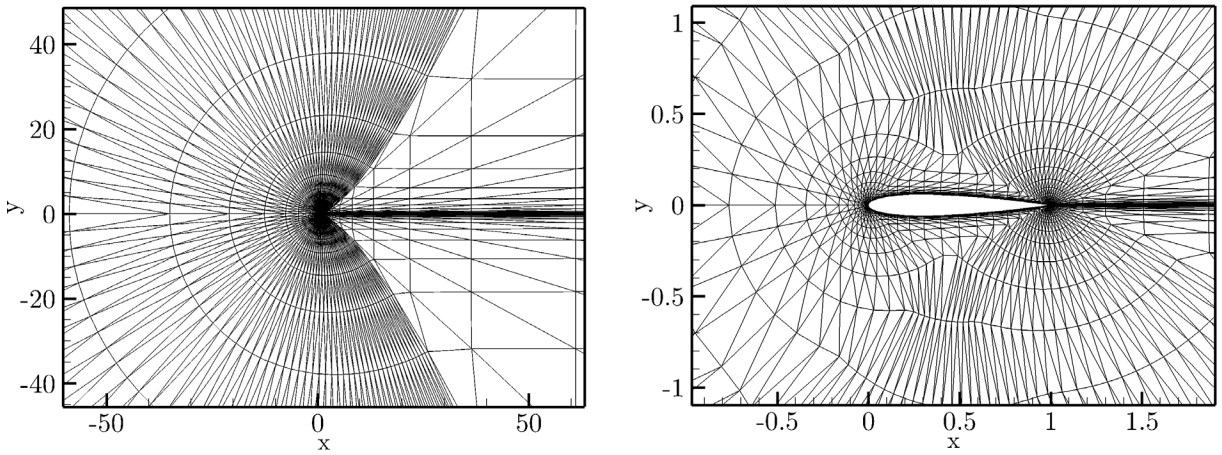
presented against a mesh spacing parameter that is determined strictly based on the number of DOFs.

Figure V.13 shows the coarsest mesh in the above mention series which is used as the initial mesh for h- and hp-adaptation. Shown in the figures V.14 and V.15 are the 6^{th} h- and hp-adapted meshes for drag prediction, respectively. Mach number contours for the final hp-adapted mesh have been shown in figure V.16. Also shown in figure V.17 are the x-velocity adjoint variable for the lift and drag, respectively. As seen in this figure, both lift and drag show strong sensitivity to the solution on the stagnation streamline, although this sensitivity is higher for the lift. The trace of this sensitivity can be clearly seen in figures V.14 and V.15.

Figures V.18 and V.19 depict the convergence of the lift and drag for h- and hp-adaptation cases. It is observed that while all the solutions converge to the reference values, adapted solutions show faster convergence. Specifically, the drag is obtained to within half of a drag count of the reference value using less than 20 thousand DOFs for hp-adaptation and less than 50 thousand DOFs for h-adaptation. Examining the pressure and skin-friction components in figures V.20 and V.21, it is seen that h- and hp-adaptive solutions also obtain values within half of a drag count with similar grid densities as the total drag. Also, shown in figures V.18 to V.21 are the corrected values of the lift and drag, which in all cases show faster convergence to the reference value.

Finally for this example, the details of the flow variables are examined using the profiles of velocity components as well as the turbulence working variable. The profiles' data are extracted at ten chord lengths downstream of the leading edge ($\frac{x}{c} = 10, 1 \leq \frac{y}{c} \leq 2$).

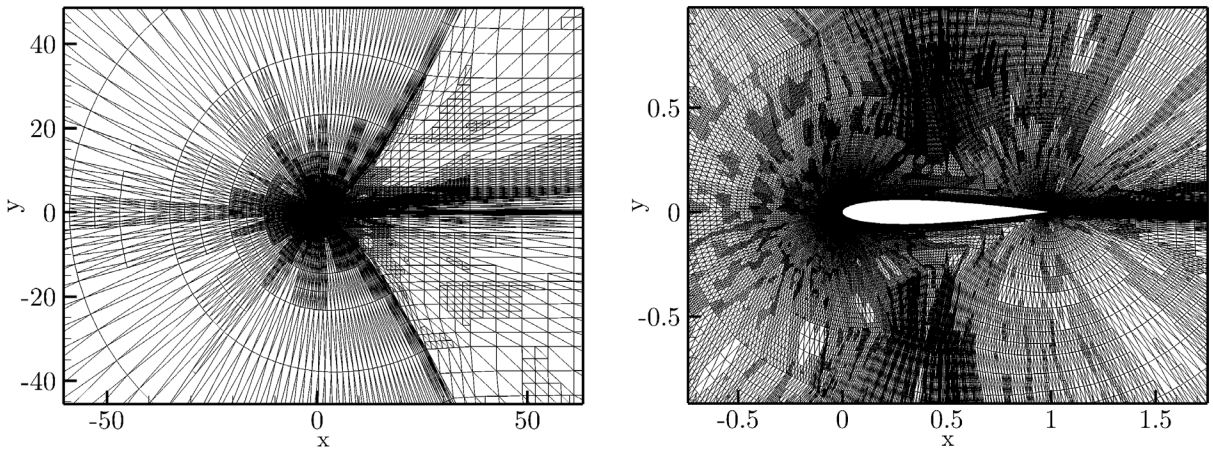
In figure V.22, the results of 8th drag-based hp-adapted mesh with 47 thousand DOFs are compared with FUN3D results on meshes with 230 thousand and 14.6 million DOFs. Also, figure V.23 shows the grid density in vicinity of the profiles' location for the compared solutions. Note that the FUN3D solution with 14.6 million DOFs is considered as the reference solution. It is observed that the FUN3D solution on the mesh with 230 thousand DOFs has been severely dissipated while the hp-adapted solution with only 47 thousand DOFs shows a fairly good agreement with the reference solution. However, it is quite apparent that there are significant oscillations and negative values of the turbulence working variable at the edge of the wake, which is not unusual on coarse meshes, and is one motivation for the development of the negative SA model [89].



(a) Far-field view

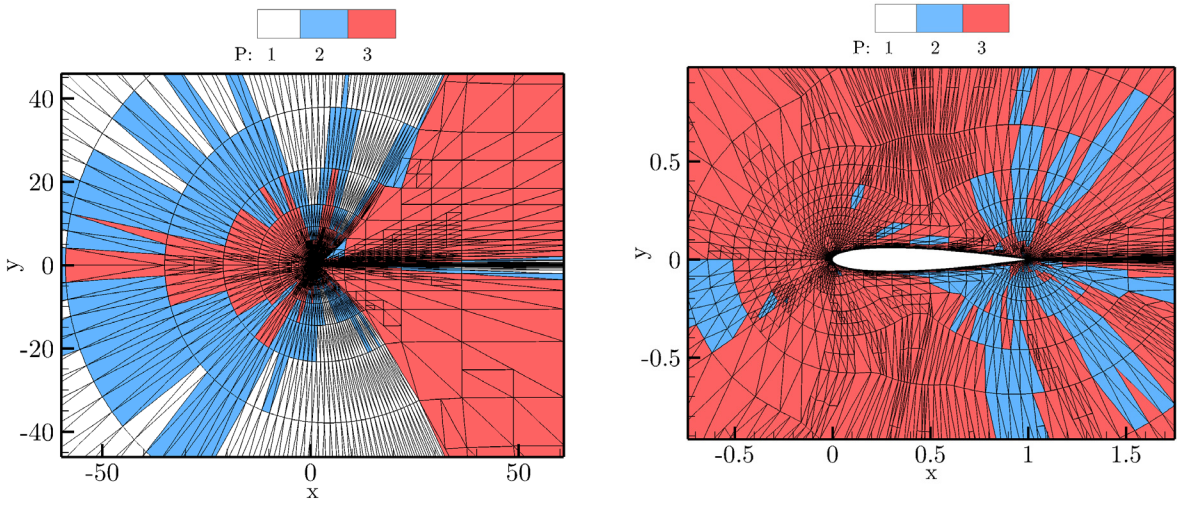
(b) Near-field view

Figure V.13 Initial meshes for the subsonic turbulent flow over NACA001 at $M_\infty = 0.15$, $\alpha = 10°$, and $Re = 6{,}000{,}000$



(a) Far-field view (b) Near-field view

Figure V.14 6th drag-based h-adapted mesh ($\Delta_{\mathcal{J}} = 0.5\%$) for the subsonic turbulent flow over NACA0012 at $M_\infty = 0.15$, $\alpha = 10°$, and $Re = 6{,}000{,}000$



(a) Far-field view (b) Near-field view

Figure V.15 6th drag-based hp-adapted mesh ($\Delta_{\mathcal{J}} = 0.2\%$) for the subsonic turbulent flow over NACA0012 at $M_\infty = 0.15$, $\alpha = 10°$, and $Re = 6{,}000{,}000$

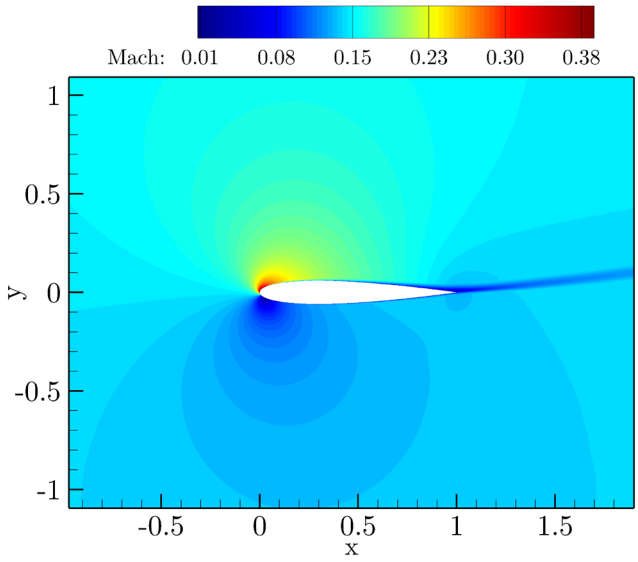
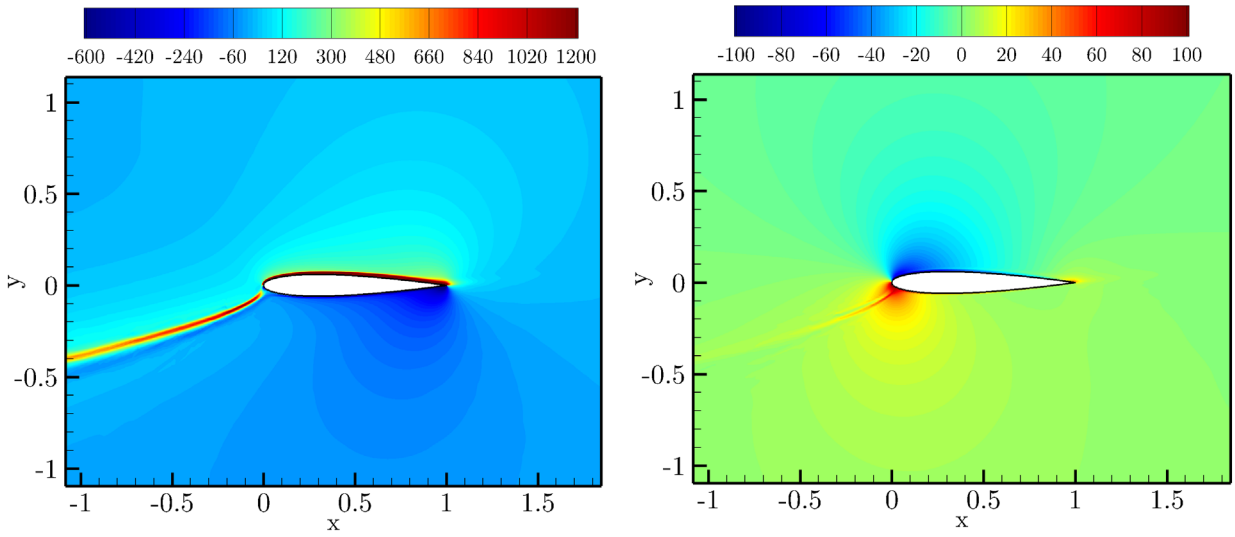


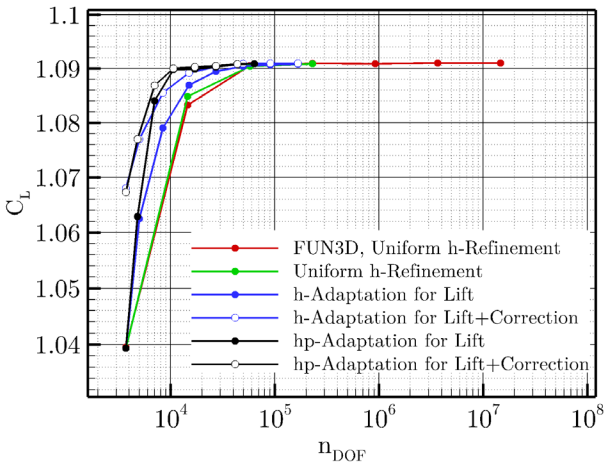
Figure V.16 Mach number contours for the subsonic turbulent flow over NACA0012 at $M_\infty = 0.15$, $\alpha = 10°$, and $Re = 6{,}000{,}000$



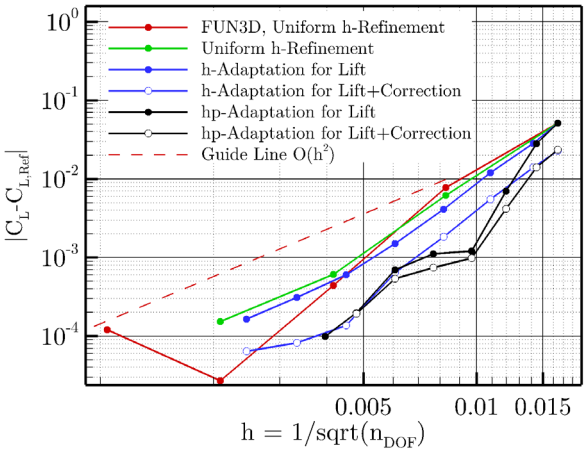
(a) Lift-based x-velocity adjoint

(b) Drag-based x-velocity adjoint

Figure V.17 Adjoint solutions for the subsonic inviscid flow over NACA0012 at $M_\infty = 0.15$, $\alpha = 10°$, and $Re = 6{,}000{,}000$

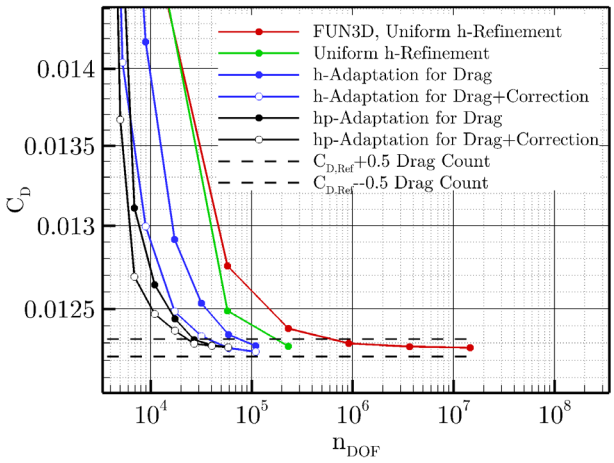


(a) Lift coefficients

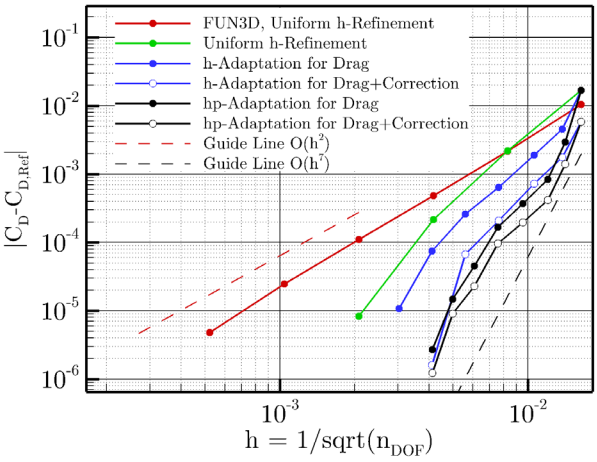


(b) Error in lift coefficients

Figure V.18 Convergence of lift coefficient in lift-based h- and hp-adaptation for the subsonic turbulent flow over NACA0012 at $M_\infty = 0.15$, $\alpha = 10°$, and $Re = 6{,}000{,}000$

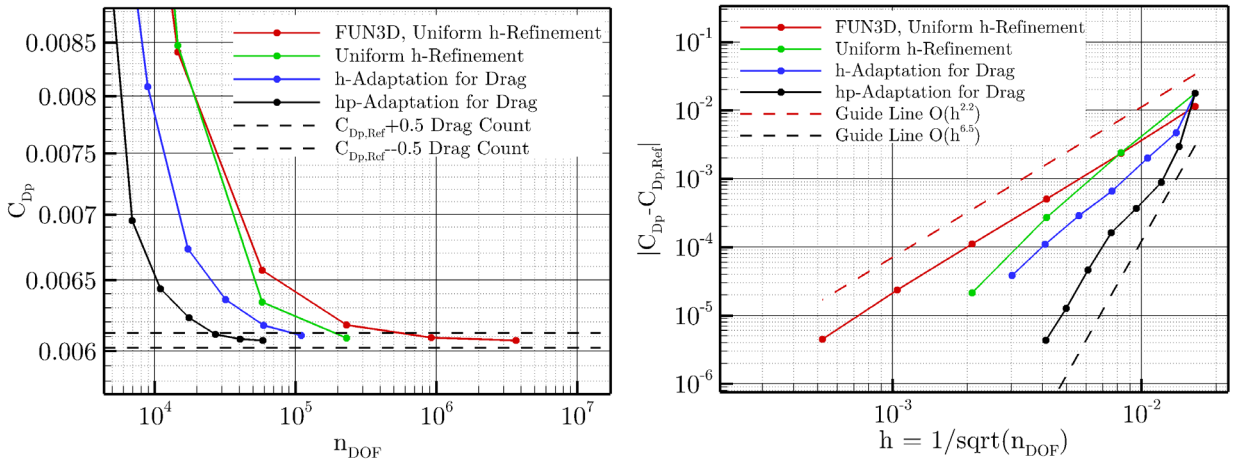


(a) Drag coefficients



(b) Error in drag coefficients

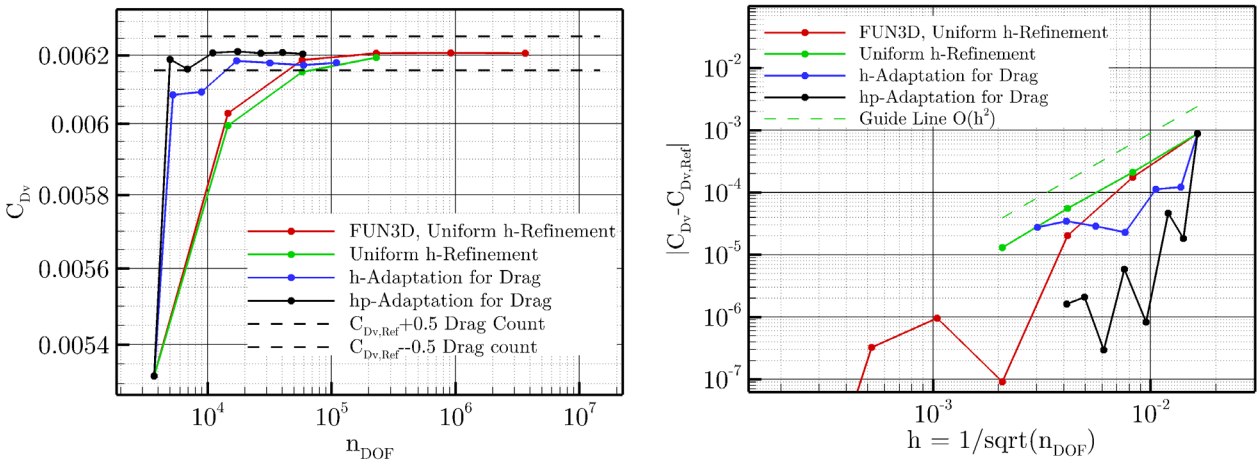
Figure V.19 Convergence of drag coefficient in drag-based h- and hp-adaptation for the subsonic turbulent flow over NACA0012 at $M_\infty = 0.15$, $\alpha = 10°$, and $Re = 6{,}000{,}000$



(a) Pressure drag coefficients

(b) Error in pressure drag coefficients

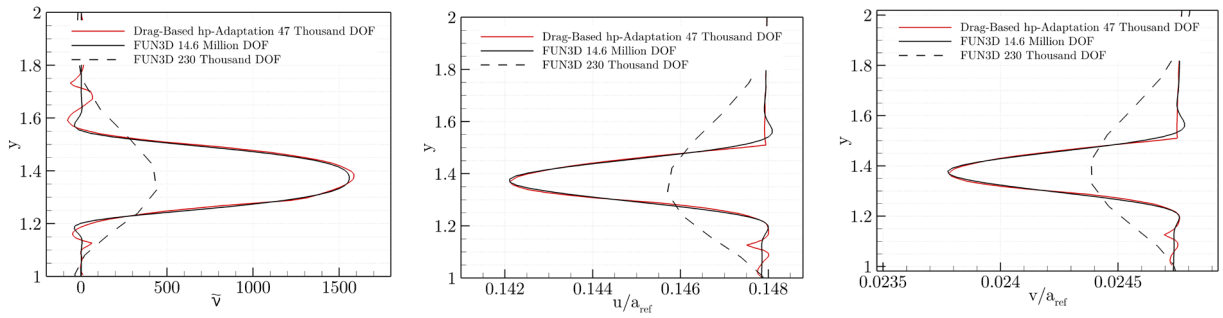
Figure V.20 Convergence of pressure drag in drag-based h- and hp-adaptation for the subsonic turbulent flow over NACA0012 at $M_\infty = 0.15$, $\alpha = 10°$, and $Re = 6{,}000{,}000$



(a) Skin friction drag coefficients

(b) Error in Skin friction drag coefficients

Figure V.21 Convergence of skin friction in drag-based h- and hp- adaptation for the subsonic turbulent flow over NACA0012 at $M_\infty = 0.15$, $\alpha = 10°$, and $Re = 6{,}000{,}000$

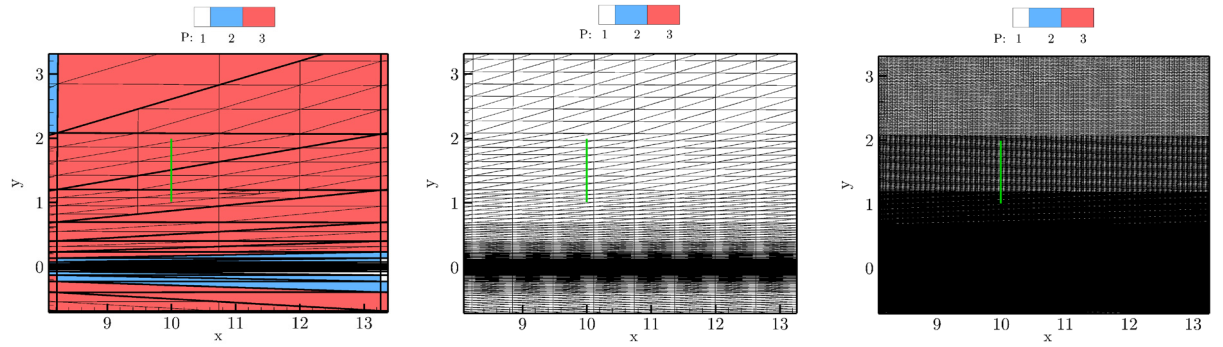


(a) Turbulence working variable

(b) x-velocity component

(c) y-velocity component

Figure V.22 Comparison of profiles of turbulence working variable and velocity components at wake region for the subsonic turbulent flow over NACA0012 at $M_\infty = 0.15$, $\alpha = 10°$, and $Re = 6{,}000{,}000$



(a) 8th drag-based hp-adapted mesh with 47 thousand DOFs

(b) FUN3D mesh with 230 thousand DOFs

(c) FUN3D mesh with 14.6 million DOFs

Figure V.23 Grid density in vicinity of profiles' location for the subsonic turbulent flow over NACA0012 at $M_\infty = 0.15$, $\alpha = 10°$, and $Re = 6{,}000{,}000$

V.1.5 Subsonic Turbulent Flow over Three Element Airfoil

The fourth numerical example shows an application of adjoint-based adaptation for subsonic turbulent flow over a multi-element airfoil. The geometry consists of a leading edge slat, a main center element and a trailing edge flap. For this case, the Mach number is 0.2, the Reynolds number is 9 million, and angle of attack is 16.2 degrees. This test case has

been widely utilized and had originally been used for a code-validation workshop [111]. The complex geometry of this test case provides sufficient challenge to assess the capability of the present adaptive method in turbulent flows with complex flow structures. For this purpose the total lift and drag coefficients has been chosen as the functional outputs, which include both pressure and viscous stresses.

The initial computational mesh, shown in figure V.24, contains 20521 triangular elements. With uniform P1 discretization, this mesh includes 10441 DOFs. The outer boundary has been placed at a distance of 100 chord lengths away from the airfoil. The spacing normal to the wall in the boundary layer mesh is 5×10^{-6}. The geometry has been represented by Q3 elements. Shown in figure V.25 are views of the 5^{th} lift-based hp-adapted mesh. As seen, large portions of the flow are relatively smooth and thus, the field is dominated by p-enrichment. Different views of the mesh at the 7^{th} cycle of the lift-based h-adaptation are shown in figure V.26. The refinement pattern in this case reveals the complex structure of the flow. Refinements have taken place on the stagnation streamline, leading edge, trailing edge, and the top surface of the elements. In addition, wake regions of the slat and the flap have been highly refined. The Mach number contours are shown in figure V.27, and illustrate that although the freestream Mach number is low, the flow accelerates significantly over the leading edge slat and upper surface of the main airfoil. Note, however, that the two-dimensional nature of the flow amplifies this effect. The x-velocity adjoint variable for the lift and drag are shown in figure V.28.a and V.28.b, respectively. As expected, the refinements observed in figures V.25 and V.26 follow the sensitivity data provided by the adjoint solution.

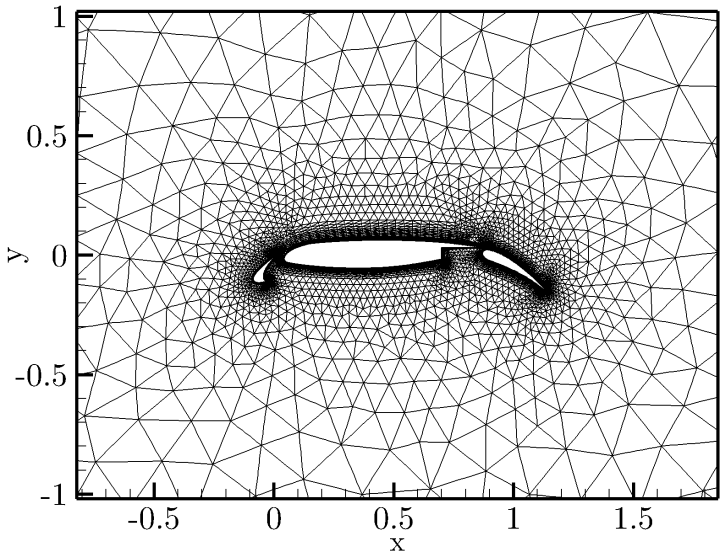
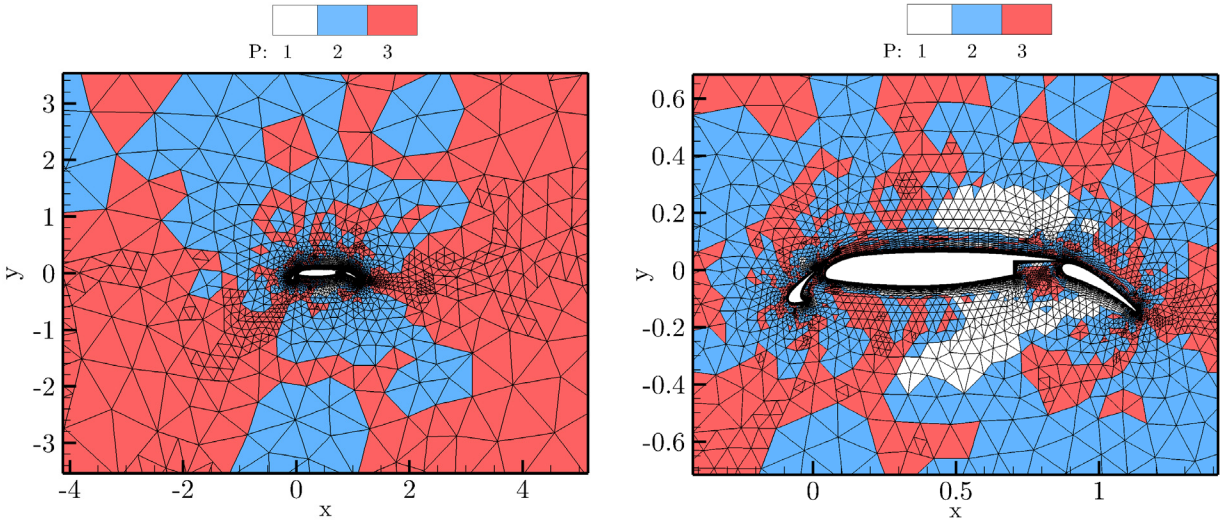


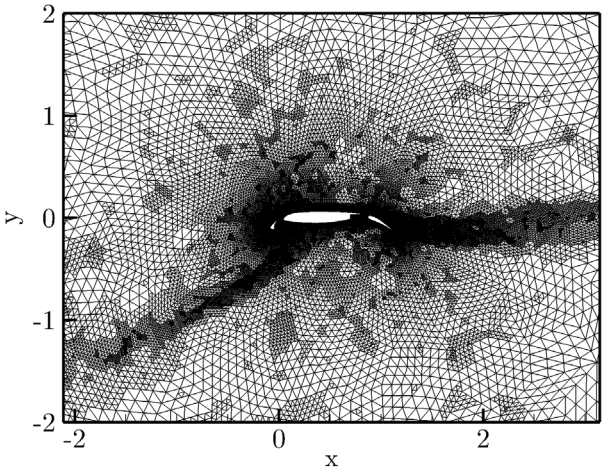
Figure V.24 Initial mesh for the subsonic turbulent flow over three-element airfoil



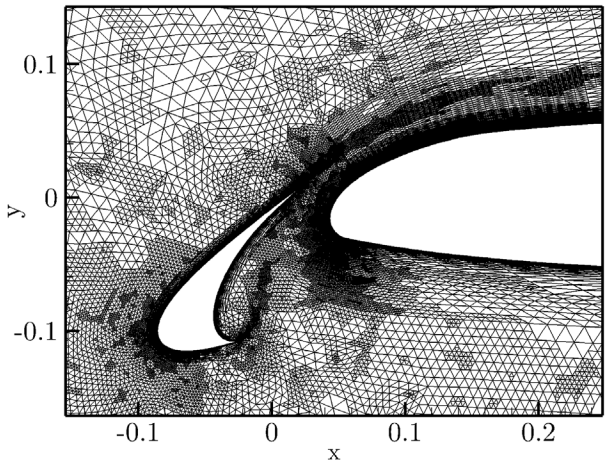
(a) Medium-field view

(b) Near-field view

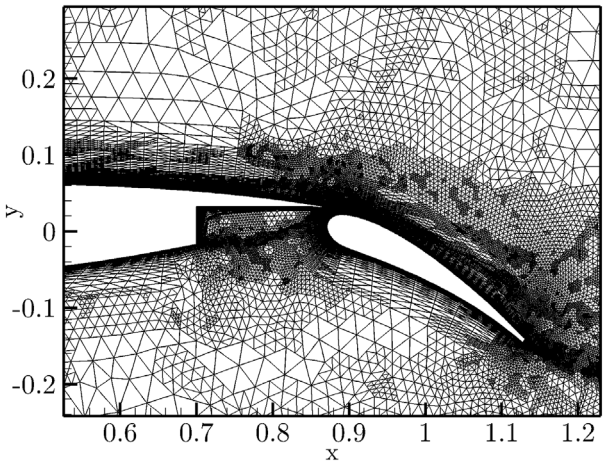
Figure V.25 5th Lift-based hp-adapted mesh ($\Delta_J = 0.02\%$) for the subsonic turbulent flow over three-element airfoil at $M_\infty = 0.2$, $\alpha = 16.2°$, and $Re = 9{,}000{,}000$



(a) Medium-field view



(a) Slat and leading edge



(b) Flap cove and flap

Figure V.26 7^{th} Lift-based h-adapted mesh ($\Delta_{\mathcal{J}} = 0.05\%$) for the subsonic turbulent flow over three-element airfoil at $M_\infty = 0.2$, $\alpha = 16.2°$, and $Re = 9{,}000{,}000$

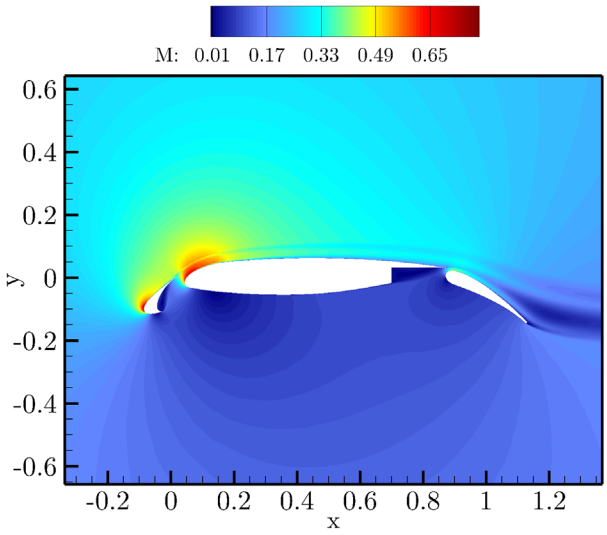
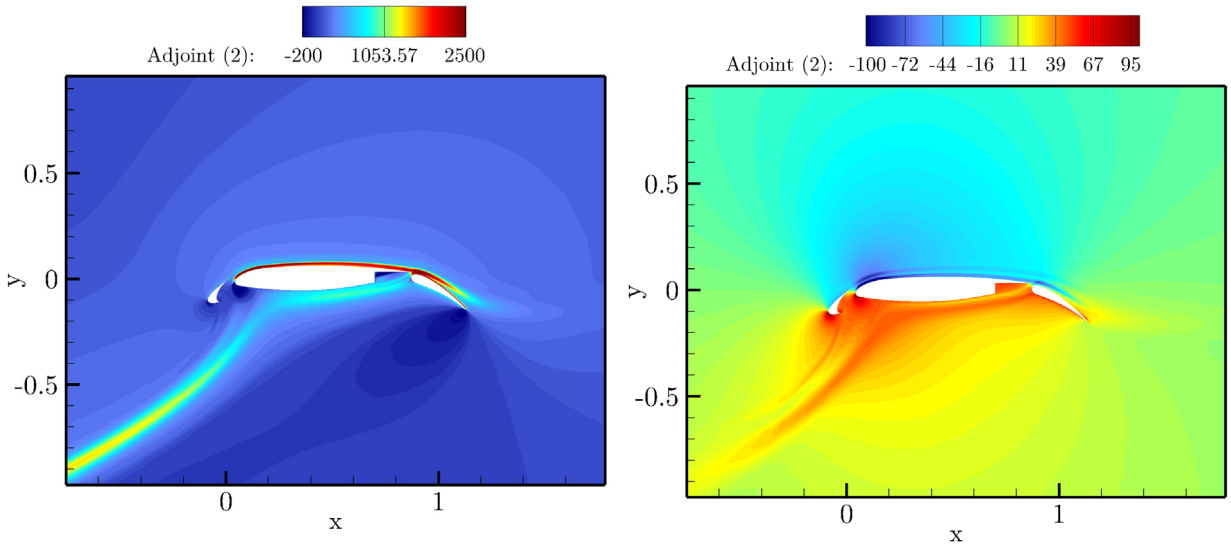


Figure V.27 Mach number contours for the subsonic turbulent flow over NACA0012 at $M_\infty = 0.2$, $\alpha = 16.2°$, and $Re = 9{,}000{,}000$

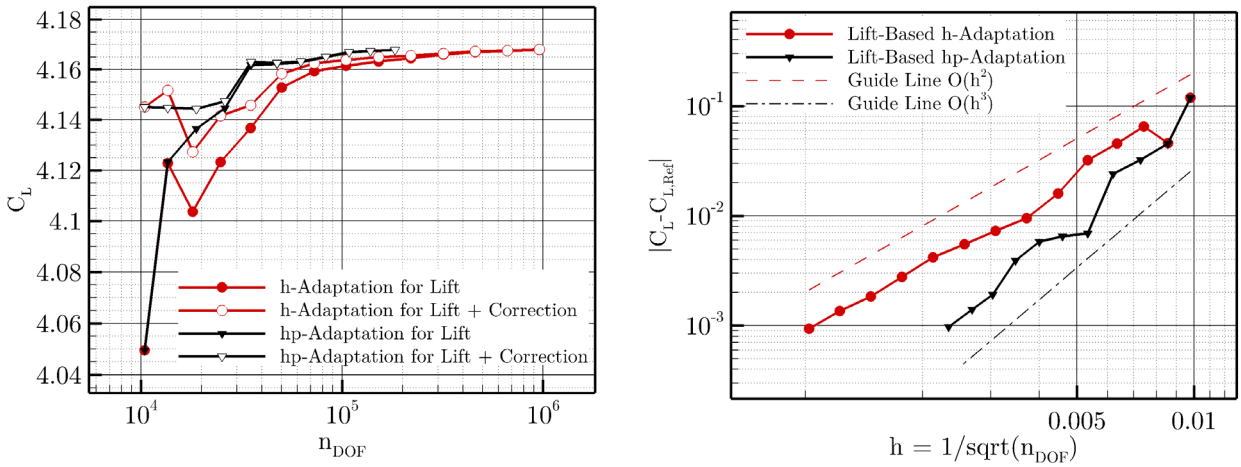


(a) Lift-based x-velocity adjoint (b) Drag-based x-velocity adjoint

Figure V.28 Adjoint solutions for the subsonic turbulent flow over three-element airfoil at $M_\infty = 0.2$, $\alpha = 16.2°$, and $Re = 9{,}000{,}000$

In order to compute the errors, for both lift and drag coefficients, the reference values has been taken from hp-adapted solutions in which the absolute difference in the functional value between last two adapted mesh have been less than 10^{-5}. Figures V.29 and V.30 show

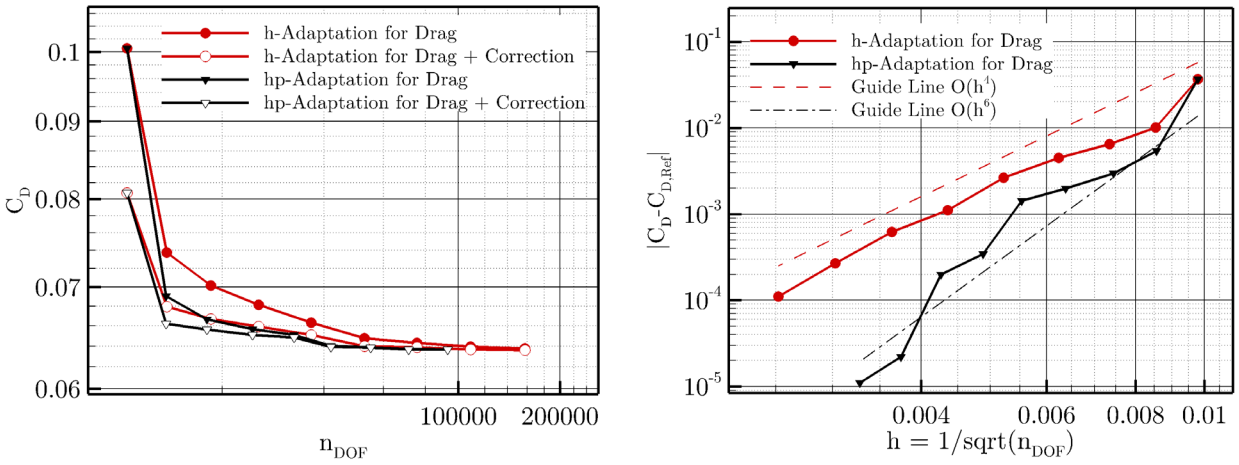
the convergence of the lift and drag coefficients, as well as their corrected values, for the h- and hp-adaptive solutions. Observe that the correct values of lift and drag in all cases are better estimates. However, the corrected drag coefficient, particularly early in the adaptation process, significantly improves the predicted values. Additionally note that the order of convergence of the lift-based h-adaptation is nearly the same as hp-adaptation. On the other hand, the convergence rate for the drag-based hp-adaptation demonstrates better performance. This potentially indicates that when drag is the primary quantity of interest, hp-adaptation should be employed. Finally, figure V.31 compares the surface pressures obtained from the final hp-adapted mesh with the experimental values [111]. Despite a small over-prediction on the upper surface of the leading edge slat, a good agreement is observed.



(a) Lift coefficients

(b) Error in lift coefficients

Figure V.29 Convergence of lift coefficient in lift-based h- and hp-adaptation for the subsonic turbulent flow over three element airfoil at $M_\infty = 0.2$, $\alpha = 16.2°$, and $Re = 9{,}000{,}000$



(a) Drag coefficients

(b) Error in drag coefficients

Figure V.30 Convergence of drag coefficient in drag-based h- and hp-adaptation for the subsonic turbulent flow over three element airfoil at $M_\infty = 0.2$, $\alpha = 16.2°$, and $Re = 9{,}000{,}000$

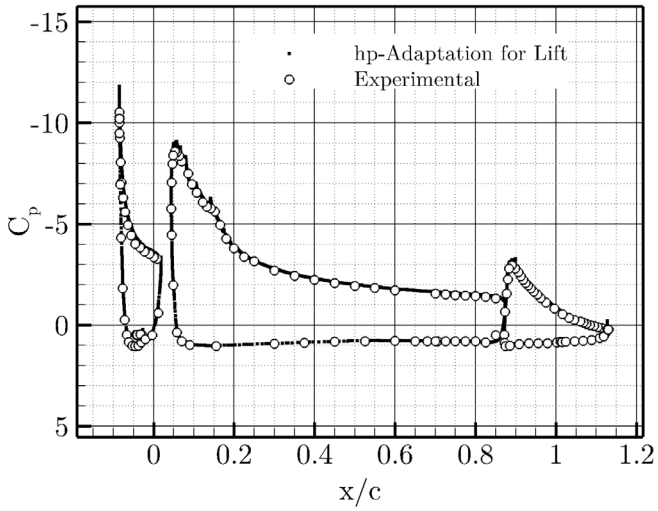


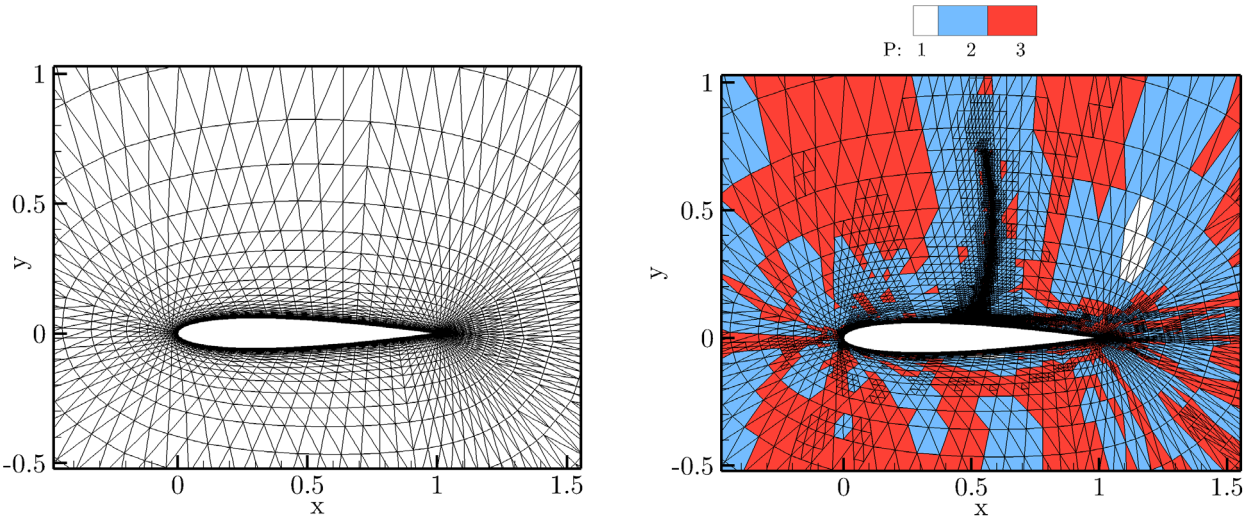
Figure V.31 Comparison of surface pressures between hp-adapted solution and experimental values for the subsonic turbulent flow over three-element airfoil at $M_\infty = 0.2$, $\alpha = 16.2°$, and $Re = 9{,}000{,}000$

V.1.6 Transonic Turbulent Flow over NACA0012

The next numerical example has been selected to show the application of the PDE-based artificial viscosity for shock capturing as well as the hp-adaptation methodology for flows with discontinuities. To this end, the transonic, turbulent flow over a NACA0012 airfoil has been examined. The flow conditions assumed are a free stream Mach number of 0.8, angle-of-attack of 2.5 degrees, and Reynolds number, based on the airfoil chord, of 3 million. The functional of interest for the adjoint-based adaptation is the lift coefficient. The advantage of the utilized artificial viscosity method is realized when higher order elements ($P > 1$) are used and thus in this test case, only hp-adaptation has been utilized.

The initial computational mesh, shown in figure V.32.a, contains 11760 triangular elements. With uniform P1 discretization, this mesh includes 5978 DOFs. The outer boundary has been placed at a distance of 200 chord lengths away from the airfoil, and the normal spacing to the wall is 8×10^{-5}. The adaptation process has been repeated until the absolute difference between lift coefficients on the last two meshes has been dropped to less than 10^{-5}. Figure V.33 shows the convergence of the lift coefficient, as well as its corrected value, for the adaptive solution. As expected the final adapted mesh, depicted in figure V.32.b, has significant h-refinement in the vicinity of the shock wave. Away from this discontinuity, where the flow is smooth, higher-order elements provide the required resolution. Shown in figures V.34.a and V.34.b are the Mach number contours for the initial and the final adapted mesh, respectively. As can be clearly seen, the initial mesh is not adequate for the region of supersonic flow on the upper surface upstream of the shock. In the final adapted mesh, all critical features of the flow field have been highly resolved.

Illustrated in figure V.35 is a close-up view of the shock near the airfoil surface. In this view the shear layer and the shock induced boundary layer separation can be clearly seen. On inspection, it can be observed that due to the separated flow, the abrupt deflection causes the upstream shock to become oblique. Furthermore, as a result of the lower pressure rise across the oblique shock, in order to recover, a weak normal shock is also formed. Since the interaction between the shock wave and the boundary layer is a critical feature in the accurate prediction of the lift, the adjoint-based adaptive methodology automatically resolved this region of the flow. In particular, close inspection of figure V.32.b shows that upper side of the recirculation region has been highly refined and the refined areas extends to the wake region.



(a) Initial Mesh

(b) Final hp-adaptaed mesh

Figure V.32 Initial and final hp-adapted mesh for the transonic turbulent flow over NACA0012 at $M_\infty = 0.8$, $\alpha = 2.5°$, and $Re = 3{,}000{,}000$

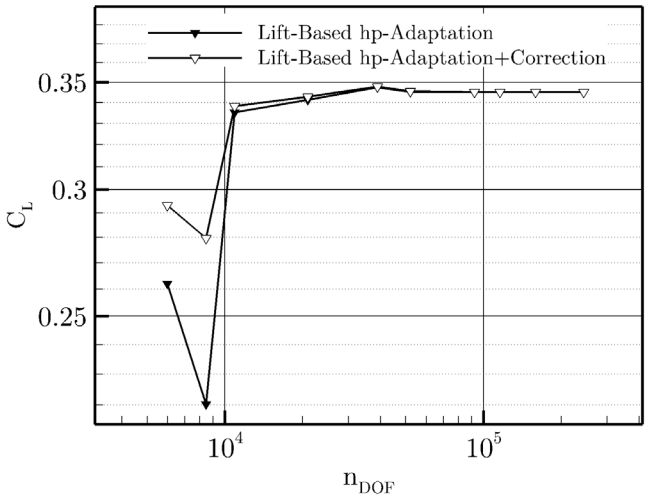
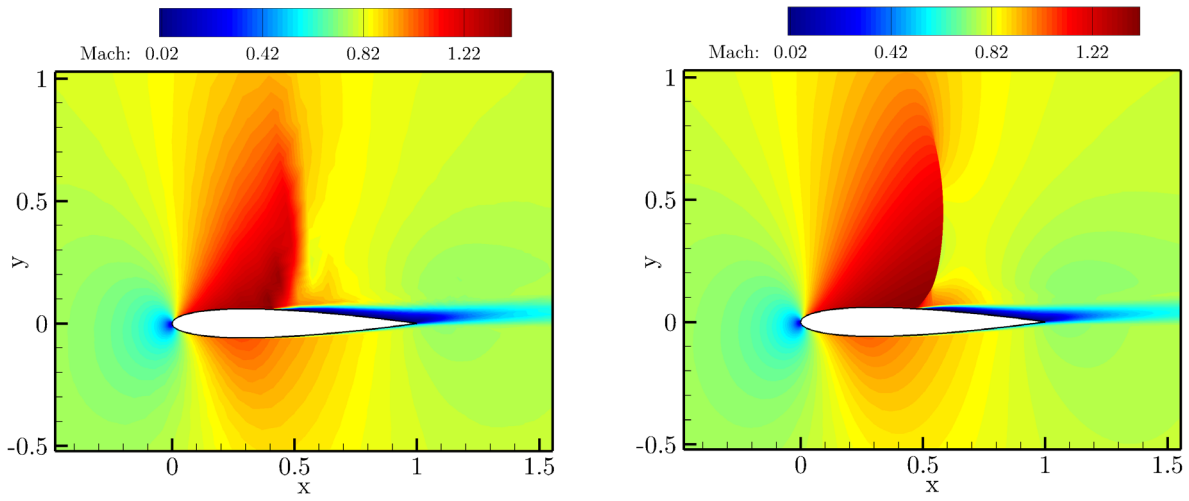


Figure V.33 Convergence of lift coefficient in lift-based hp-adaptation for the transonic turbulent flow over NACA0012 at $M_\infty = 0.8$, $\alpha = 2.5°$, and $Re = 3{,}000{,}000$



(a) Mach number on initial mesh

(b) Mach number on final hp-adapted mesh

Figure V.34 Mach number contours on initial and final hp-adapted mesh for the transonic turbulent flow over NACA0012 at $M_\infty = 0.8$, $\alpha = 2.5°$, and $Re = 3{,}000{,}000$

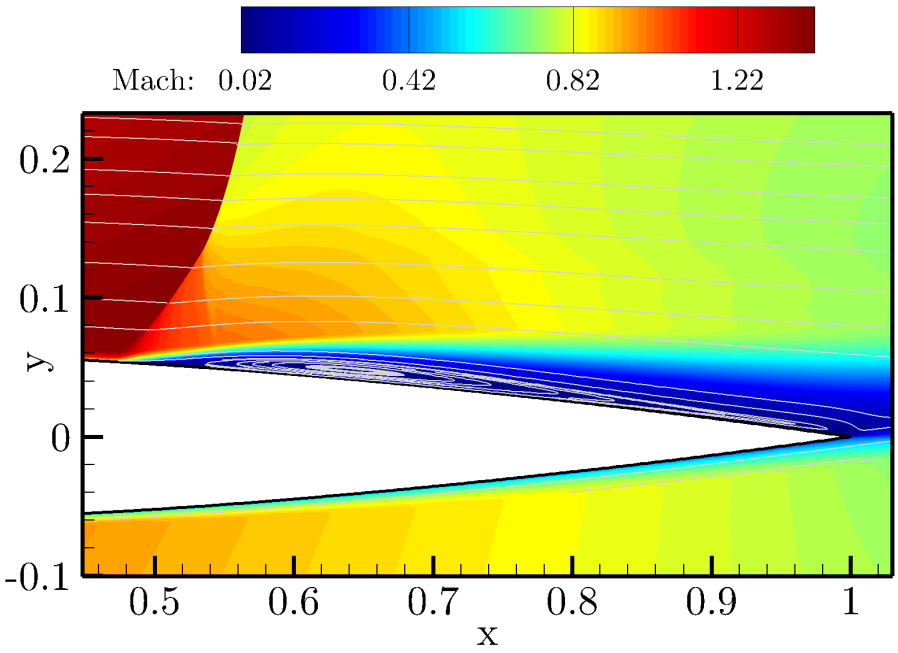
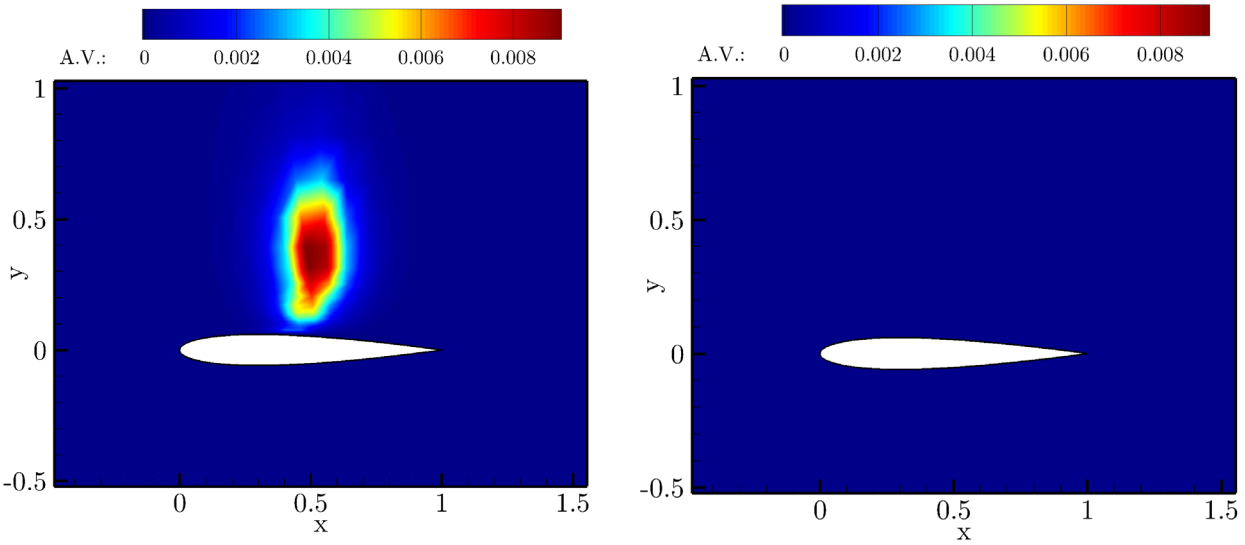


Figure V.35 Shock induced boundary layer separation in the transonic turbulent flow over NACA0012 at $M_\infty = 0.8$, $\alpha = 2.5°$, and $Re = 3{,}000{,}000$.

The added artificial viscosity in the initial and final hp-adapted meshes are shown in figure V.36. Observe that in the initial mesh, considerable artificial viscosity has been added in the vicinity of the shock wave. However, the amount of added artificial viscosity in the final adapted mesh is not visibly detectable. This is an indication that the current implementation is consistent, and that as the mesh is refined, the stabilization vanishes and the original governing equations are recovered.



(a) Artificial viscosity on initial mesh

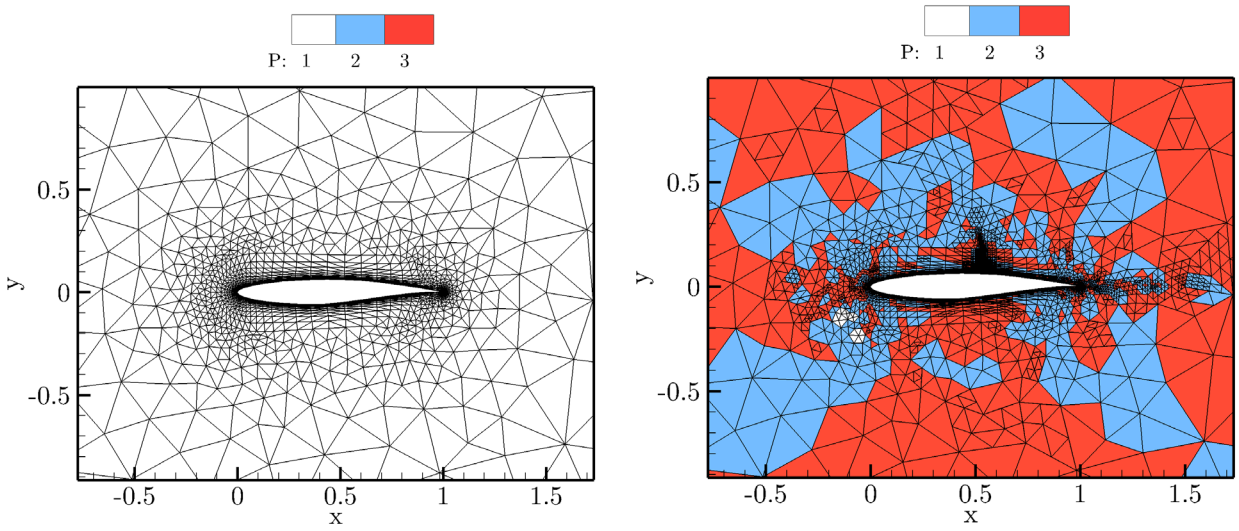
(b) Artificial viscosity on final hp-adapted mesh

Figure V.36 Artificial viscosity contours on initial and final hp-adapted mesh for the transonic turbulent flow over NACA0012 at $M_\infty = 0.8$, $\alpha = 2.5°$, and $Re = 3{,}000{,}000$

V.1.7 Transonic Turbulent Flow over RAE2822

In order to validate the numerical solutions in presence of shock waves, in this example, the subsonic turbulent flow over RAE2822 airfoil at Mach number of 0.729, angle-of-attack of 2.31 degrees, and Reynolds number, based on the airfoil chord, of 6.5 million is compared with experimental data [112]. Once more, the functional of interest for the adjoint-based adaptation is the lift coefficient. The initial computational mesh, shown in figure V.37.a, contains 5903 triangular elements. With uniform P1 discretization, this mesh includes 3019 DOFs. The outer boundary has been placed at a distance of 100 chord lengths away from the airfoil, and the normal spacing to the wall is 1×10^{-5}. The adaptation process has been repeated until the absolute difference between lift coefficients on the last two meshes has been dropped to less than 10^{-5}. Figure V.37.b shows the final hp-adapted

mesh and figure V.38 shows the convergence of the lift coefficient, as well as its corrected value. Shown in figure V.39 are the Mach number contours for the initial and the final adapted meshes, respectively. Spurious oscillations at upper surface of the airfoil can be easily observed for the solution on the initial mesh. These oscillations can also be seen in figure V.40 which compares the surface pressures obtained on the initial and final hp-adapted meshes with the experimental values. As seen, the results of final hp-adapted mesh shows an acceptable agreement with experimental data. However, in this case, the separation bubble has not been properly resolved.



(a) Initial Mesh

(b) Final hp-adaptaed mesh

Figure V.37 Initial and final hp-adapted mesh for the transonic turbulent flow over RAE2822 at $M_\infty = 0.729$, $\alpha = 2.31°$, and $Re = 6{,}500{,}000$

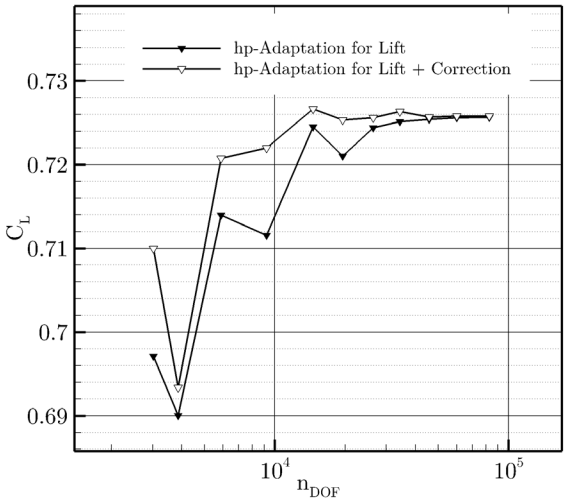
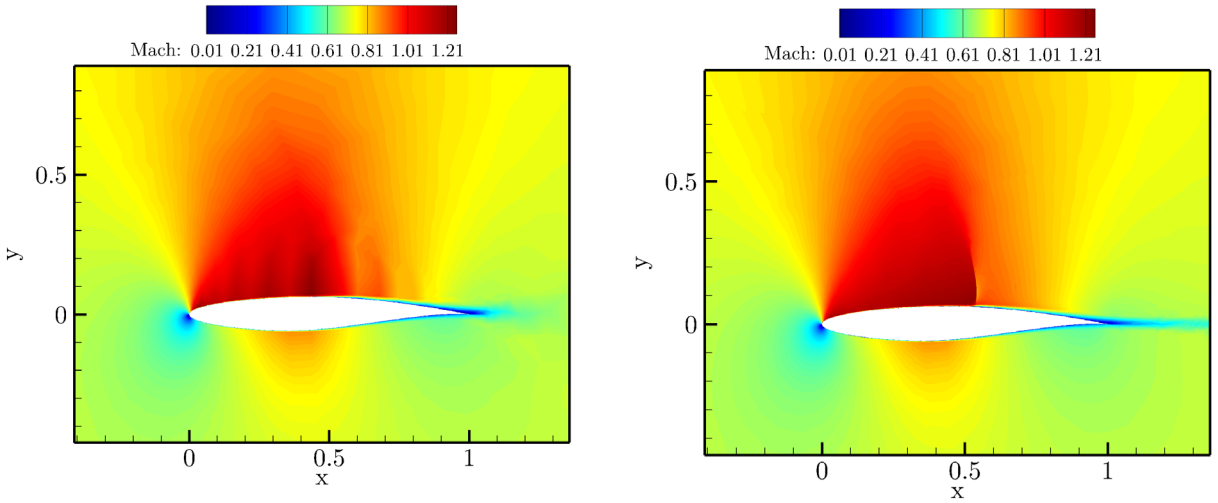


Figure V.38 Convergence of lift coefficient in lift-based hp-adaptation for the transonic turbulent flow over RAE2822 at $M_\infty = 0.729$, $\alpha = 2.31°$, and $Re = 6{,}500{,}000$



(a) Mach number on initial mesh

(b) Mach number on final hp-adapted mesh

Figure V.39 Mach number contours on initial and final hp-adapted mesh for the transonic turbulent flow over RAE2822 at $M_\infty = 0.729$, $\alpha = 2.31°$, and $Re = 6{,}500{,}000$

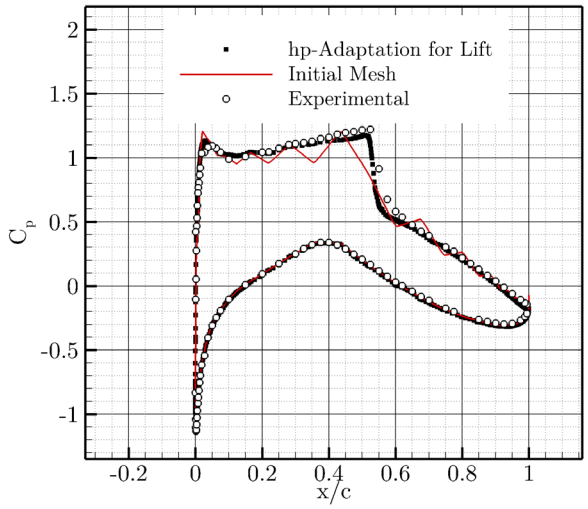


Figure V.40 Comparison of surface pressures between solutions on initial and hp-adapted meshes with experimental values for the transonic turbulent flow over RAE2822 at $M_\infty = 0.729$, $\alpha = 2.31°$, and $Re = 6{,}500{,}000$

V.2 Feature-Based Adaptation in Unsteady Flows

V.2.1 Vortex Shedding Flow over a Cylinder

The final numerical example is the problem of laminar vortex shedding over a cylinder. This case has been chosen to demonstrate the operation of the developed program in dynamic adaptation. The geometry consists of a cylinder with the diameter of 1 whose center is located at $(x = 0, y = 0)$. The computational domain extends from -8 to 8 in y-direction and -8 to 25 in x direction. Of course, to minimize the effect of outer boundaries, a larger domain is needed. However, here the performance of the adaptation is assessed by its ability to retain the vorticity strength in the wake region and not by exact quantitative comparisons. Four cases have been run which all start with a triangular mesh with 6248 elements and 3241 points. These cases are as follows:

- Case 1: whole the domain is covered with P1 elements.
- Case 2: whole the domain is covered with P2 elements.
- Case 3: h-adaptation is performed on P1 elements. Maximum refinement level is set to 3.
- Case 4: p-adaptation is performed starting with P1 elements and the maximum P is set to 3.

The free-stream Mach number is 0.2, the Reynolds number is 100, and the time step is 0.02. Each computation starts with uniform flow and continues until a periodic solution is reached. For each time step computation is continued until the residual of all equations drops to 10^{-14}. Since the problem is purely transient, error estimation and subsequent adaptation is repeatedly performed after 5 time steps for both h- and p-adaptations.

Figure V.41 shows the development of the shedding for the case of h-adaptation. The left panel shows the meshes and the right panel shows the vorticity contours. As expected from a feature-based error indicator, the refined regions follow the solution very well. Figure V.42 replicates the same pictures for the case of p-adaptation. This time the left panel shows the polynomial degree maps. As seen in this figure, after the shedding formation, the core of the wake region is covered with P3 elements while the outer parts are covered with P2 elements. Also it can be seen that in some regions P3 elements have been placed next to the P1 elements without creating any problem.

Figure V.43 shows the time variation of vorticity at $(x = 2, y = 0)$. In this figure, it is seen that for all cases, there is a transition stage until a periodic variation is reached. Also, all cases except case 1 agree on the magnitude of the vorticity in the periodic region and in fact, case 3 and case 4 almost fall on top of each other. Another point is that case 1

is the slowest one to reach the periodic state. This can be described by the fact that case 1 has the most dissipation and consequently the least capability to retain the vorticity. This point can also be seen in figure V.44 which compares the vorticity contours of all cases at a snapshot within the periodic state. It can be seen in this figure that both adapted cases have retained the vorticity in the wake region better than the other cases. Note that in this figure, the location of positive and negative vortices for different cases do not match. This is because each case has a different transition time to reach the periodic state. However, all the sub-figures show the snapshot at $t = 600$.

Finally, this test case verifies that refinement and derefinement mechanisms have been successfully implemented.

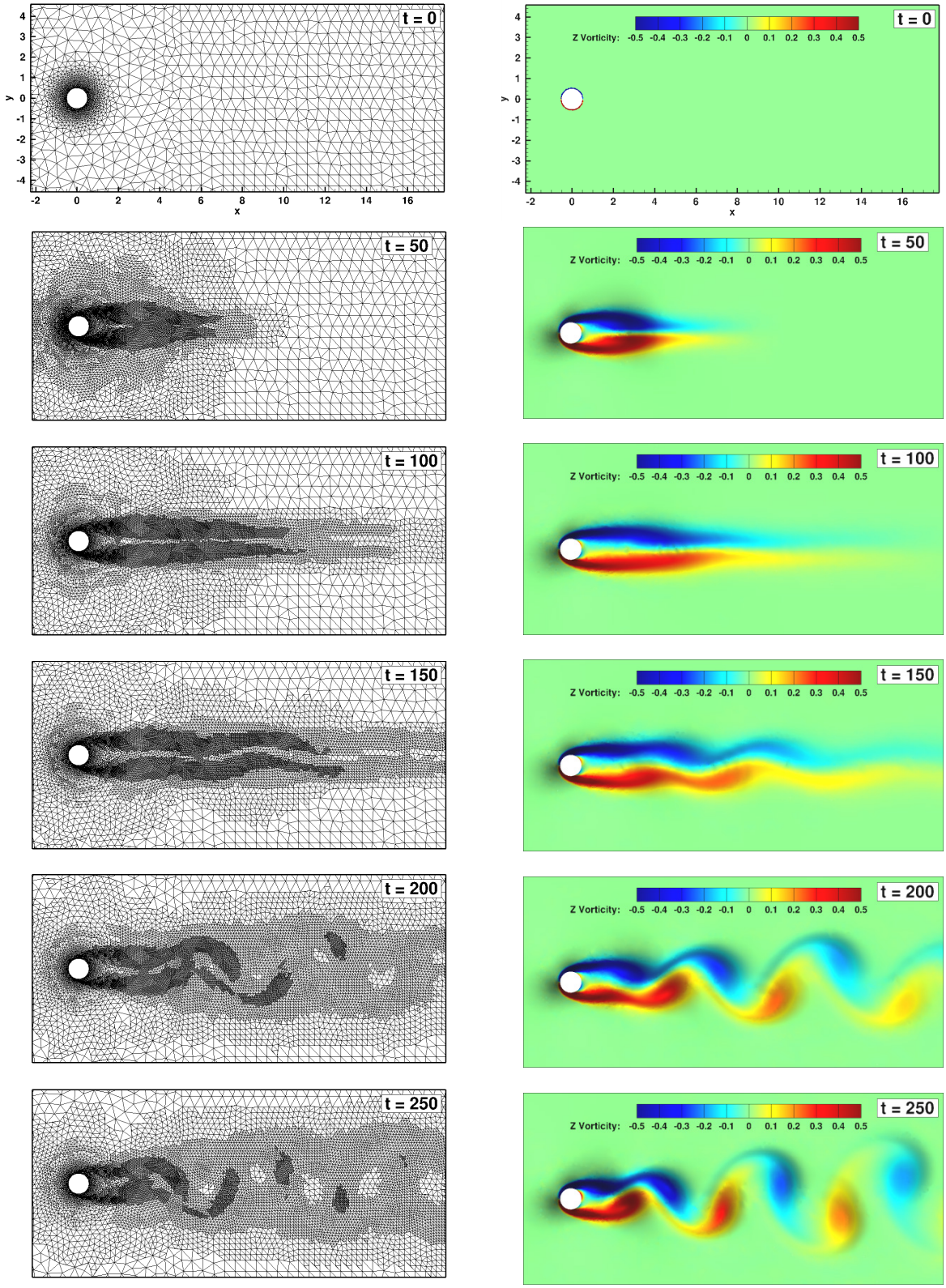


Figure V.41 Mesh and vorticity contours in dynamic h-adaptation on P1 elements for the Vortex shedding over a cylinder at $M_\infty = 0.2$ and $Re = 100$

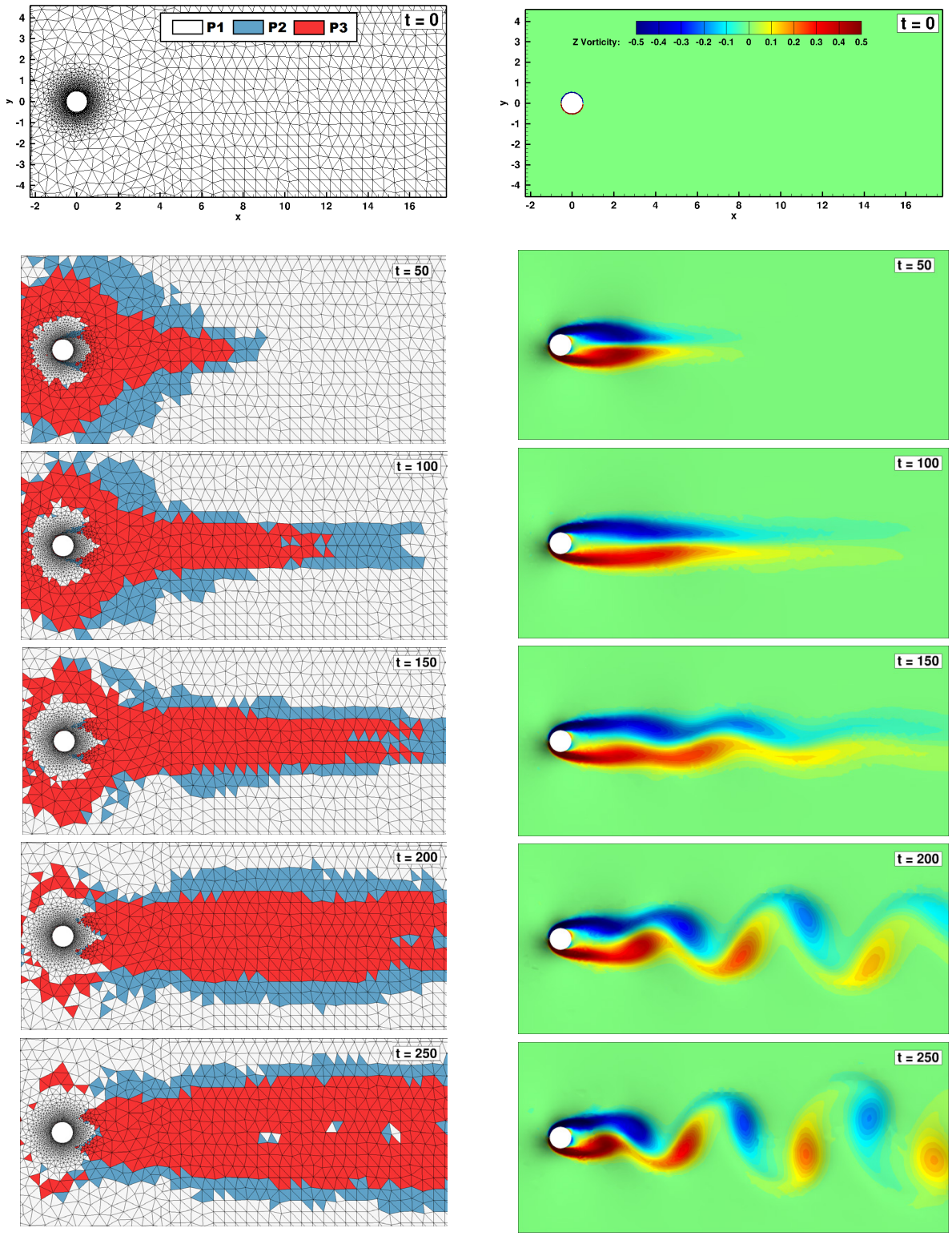


Figure V.42 Mesh and vorticity contours in dynamic p-adaptation using P1 to P3 elements for the Vortex shedding over cylinder at $M_\infty = 0.2$ and $Re = 100$

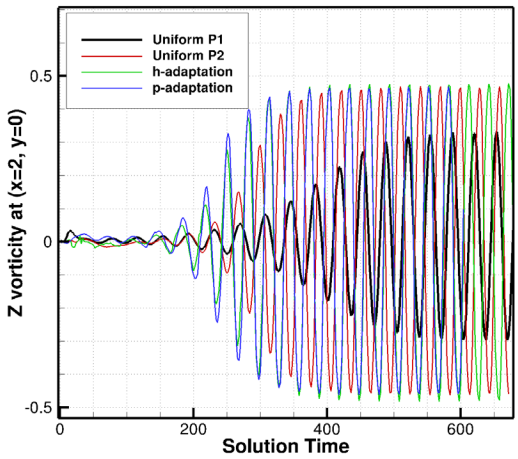
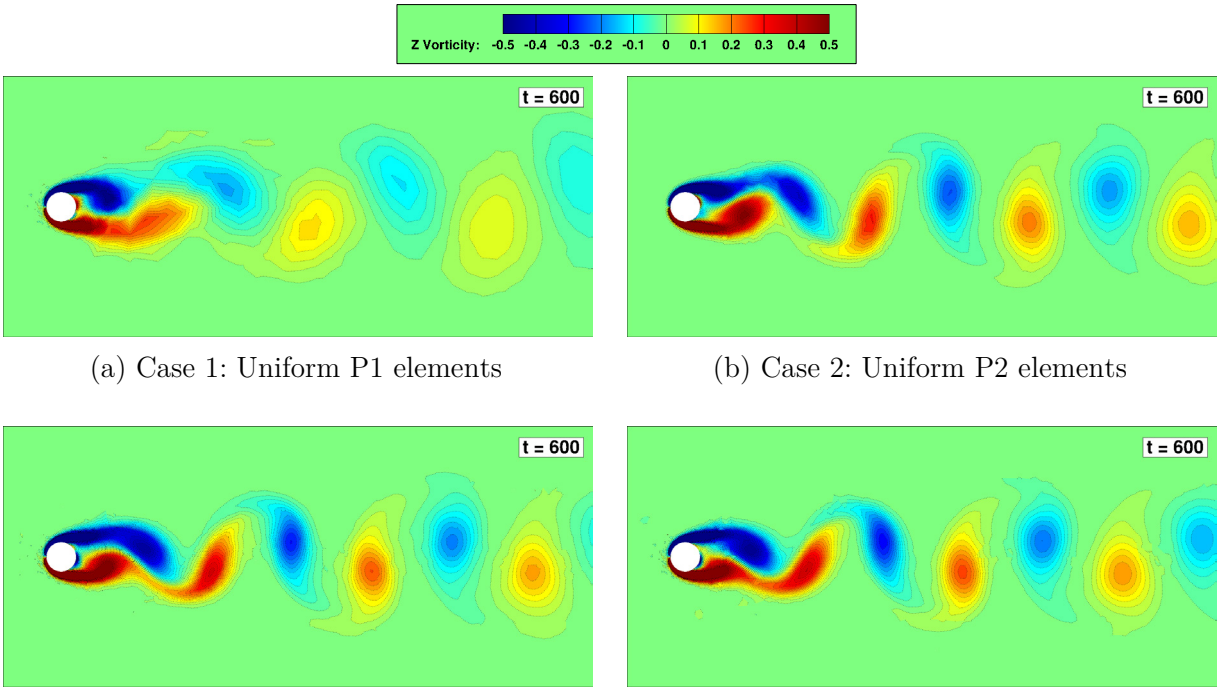


Figure V.43 Comparison of time variation of vorticity at $(x = 2, y = 0)$ for the vortex shedding flow over a cylinder at $M_\infty = 0.2$ and $Re = 100$



(a) Case 1: Uniform P1 elements

(b) Case 2: Uniform P2 elements

(c) Case 3: Dynamic h-adaptation on P1 elements

(d) Case 4: Dynamic p-adaptation using P1 to P3 elements

Figure V.44 A snapshot of periodic state for the vortex shedding flow over a cylinder at $M_\infty = 0.2$ and $Re = 100$

CHAPTER VI

CONCLUSION

In this study, output-based and feature-based adaptation algorithms were implemented within a Petrov-Galerkin finite-element method. Constrained approximation with hierarchical basis functions were employed to perform h-, p-, and combined hp-adaptations in a non-conforming manner. The resulting method is not limited to the fluid problems and it can be utilized within any continuous Galerkin method. This is particularly beneficial for multidisciplinary applications. For the geometric surfaces, high-order curved boundary meshes were generated, with the interior meshes deformed through the solution of linear elasticity equations. The methodology was demonstrated on numerous cases using the Euler and Reynolds Average Navier-Stokes (RANS) equations, equipped with a modified Spalart-Allmaras (SA) turbulence model. An enhanced h-refinement technique based on the smoothness of the solution was proposed and it was shown that employment of this technique near the geometric singularities, such as trailing edges, significantly increases the accuracy of integrated quantities. Moreover, it was shown that the implementation of weak boundary conditions and use of a modified functional are required to obtain a smooth adjoint solution where Dirichlet boundary conditions are imposed. The finite-element h- and hp-adaptive solutions were compared with finite-volume solutions in a subsonic

turbulent flow and it was shown that finite-element solutions are significantly less dissipative. Furthermore, to stabilize the solution in the vicinity of shock waves, PDE-based artificial viscosity was added to the governing equations. The adaptive algorithm was shown to automatically resolve the shock induced boundary layer separation that was present in a transonic, turbulent flow over an airfoil. Also, it was shown that the current implementation is consistent, and that as the mesh is refined, the added artificial viscosity vanishes and the original governing equations are recovered.

Topics for future work include:

1- Utilization of alternative stabilization methods such as Galerkin Least Squares (GLS) and Variation Multiscale (VMS) methods in the hp-adaptive algorithm should be explored. In particular, adjoint-consistency properties for these methods need to be studied in detail.

2- The current method should be extended to three dimensional problems with mixed-type elements.

3- Development of conforming mesh adaptation, in order to avoid the generation of hanging nodes, should be investigated.

4- Feature-based adaptation in unsteady problems should be replaced with reliable error estimation methods, like the error transport equations, for dynamic adaptation.

5- Dynamic load balancing needs to be addressed in order to obtain an efficient parallel implementation.

REFERENCES

[1] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*: Dover Publications, Incorporated, 2012.

[2] A. R. Baserinia, "Residual-based isotropic and anisotropic mesh adaptation for computational fluid dynamics," PhD, Department of Mechanical Engineering, University of Waterloo, 2008.

[3] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary*, et al.*, "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids,* vol. 72, pp. 811-845, 2013.

[4] W. K. Anderson, B. R. Ahrabi, and J. C. Newman III, "Finite-element solutions for turbulent flow over the NACA 0012 airfoil (Invited)," presented at 53rd AIAA Aerospace Sciences Meeting, AIAA Paper 2015-1531, 2015.

[5] B. Cockburn, S. Hou, and C.-W. Shu, "The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case," *Mathematics of Computation,* vol. 54, pp. 545-581, 1990.

[6] F. Bassi and S. Rebay, "A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations," *Journal of Computational Physics,* vol. 131, pp. 267-279, 1997.

[7] F. Bassi and S. Rebay, "High-order accurate discontinuous finite element solution of the 2D Euler equations," *Journal of Computational Physics,* vol. 138, pp. 251-285, 1997.

[8] R. Hartmann and P. Houston, "Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations," *Journal of Computational Physics,* vol. 183, pp. 508-532, 2002.

[9] J.-F. Remacle, J. E. Flaherty, and M. S. Shephard, "An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems," *SIAM Review,* vol. 45, pp. 53-72, 2003.

[10] K. J. Fidkowski and D. L. Darmofal, "Development of a higher-order solver for aerodynamic applications," presented at 42nd AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2004-0436, 2004.

[11] 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,* vol. 207, pp. 92-113, 2005.

[12] J. F. Remacle, X. Li, M. S. Shephard, and J. E. Flaherty, "Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods," *International Journal for Numerical Methods in Engineering,* vol. 62, pp. 899-923, 2005.

[13] C. R. Nastase and D. J. Mavriplis, "High-order discontinuous Galerkin methods using an *hp*-multigrid approach," *Journal of Computational Physics,* vol. 213, pp. 330-357, 2006.

[14] P.-O. Persson and J. Peraire, "An efficient low memory implicit DG algorithm for time dependent problems," presented at 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2006-0113, 2006.

[15] R. Hartmann, "Adjoint consistency analysis of discontinuous Galerkin discretizations," *SIAM Journal on Numerical Analysis,* vol. 45, pp. 2671-2696, 2007.

[16] T. A. Oliver, "A high-order, adaptive, discontinuous Galerkin finite element method for the Reynolds-Averaged Navier-Stokes equations," PhD, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2008.

[17] L. T. Diosady and D. L. Darmofal, "Preconditioning methods for discontinuous Galerkin solutions of the Navier–Stokes equations," *Journal of Computational Physics,* vol. 228, pp. 3917-3935, 2009.

[18] D. Mavriplis, C. Nastase, K. Shahbazi, L. Wang, and N. Burgess, "Progress in high-order discontinuous Galerkin methods for aerospace applications," presented at 47th AIAA Aerospace Science Meeting, AIAA Paper 2009-601, 2009.

[19] L. Wang, "Techniques for high-order adaptive discontinuous Galerkin discretizations in fluid dynamics," PhD, Department of Mechanical Engineering, University of Wyoming, 2009.

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

[21] N. K. Burgess, "An Adaptive Discontinuous Galerkin Solver for Aerodynamic Flows," PhD, Department of Mechanical Engineering, University of Wyoming, 2011.

[22] N. K. Burgess and D. J. Mavriplis, "Robust computation of turbulent flows using a discontinuous Galerkin method," presented at 50th AIAA Aerospcae Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA Paper 2012-0457, 2012.

[23] M. Ceze, "A robust hp-adaptation method for discontinuous Galerkin discretizations applied to aerodynamic flows," PhD, Department of Aerospace Engineering, University of Michigan, 2013.

[24] R. Hartmann, "Higher-order and adaptive discontinuous Galerkin methods applied to turbulent delta wing flow," in *New Results in Numerical and Experimental Fluid Mechanics VIII.* vol. 121, A. Dillmann, G. Heller, H.-P. Kreplin, W. Nitsche, and I. Peltzer, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 497-505.

[25] L. Wang, W. K. Anderson, J. T. Erwin, and S. Kapadia, "Discontinuous Galerkin and Petrov Galerkin methods for compressible viscous flows," *Computers & Fluids,* vol. 100, pp. 13-29, 2014.

[26] A. N. Brooks and T. J. Hughes, "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering,* vol. 32, pp. 199-259, 1982.

[27] T. Tezduyar and T. Hughes, "Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations," presented at 21st AIAA Aerospace Sciences Meeting, AIAA Paper 83-0125, 1983.

[28] T. J. Hughes and T. Tezduyar, "Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations," *Computer Methods in Applied Mechanics and Engineering,* vol. 45, pp. 217-284, 1984.

[29] T. J. Hughes, "Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier-Stokes equations," *International Journal for Numerical Methods in Fluids,* vol. 7, pp. 1261-1275, 1987.

[30] T. E. Tezduyar, "Stabilized finite element formulations for incompressible flow computations," *Advances in Applied Mechanics,* vol. 28, pp. 1-44, 1991.

[31] L. P. Franca and S. L. Frey, "Stabilized finite element methods: II. The incompressible Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering,* vol. 99, pp. 209-233, 1992.

[32] D. L. Bonhaus, "A higher order accurate finite element method for viscous compressible flows," PhD, Department of Aerospace and Ocean Engineering, Virginia Polytechnic Institute and State University, 1998.

[33] V. Venkatakrishnan, S. R. Allmaras, D. S. Kamenetskii, and F. T. Johnson, "Higher order schemes for the compressible Navier-Stokes equations," presented at 16th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2003-3987, 2003.

[34] T. J. Hughes and G. N. Wells, "Conservation properties for the Galerkin and stabilised forms of the advection–diffusion and incompressible Navier–Stokes equations," *Computer Methods in Applied Mechanics and Engineering,* vol. 194, pp. 1141-1159, 2005.

[35] B. S. Kirk and G. F. Carey, "Development and validation of a SUPG finite element scheme for the compressible Navier–Stokes equations using a modified inviscid flux discretization," *International Journal for Numerical Methods in Fluids,* vol. 57, pp. 265-293, 2008.

[36] T. J. R. Hughes, G. Scovazzi, and T. E. Tezduyar, "Stabilized methods for compressible flows," *Journal of Scientific Computing,* vol. 43, pp. 343-368, 2010.

[37] B. S. Kirk, S. W. Bova, and R. B. Bond, "The influence of stabilization parameters in the SUPG finite element method for hypersonic flows," presented at 48th AIAA

Aerospcae Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA Paper 2010-1183, 2010.

[38] W. K. Anderson, L. Wang, S. Kapadia, C. Tanis, and B. Hilbert, "Petrov–Galerkin and discontinuous-Galerkin methods for time-domain and frequency-domain electromagnetic simulations," *Journal of Computational Physics,* vol. 230, pp. 8360-8385, 2011.

[39] J. T. Erwin, W. K. Anderson, S. Kapadia, and L. Wang, "Three-dimensional stabilized finite elements for compressible Navier–Stokes," *AIAA Journal,* vol. 51, pp. 1404-1419, 2013.

[40] J. T. Erwin, W. K. Anderson, L. Wang, and S. Kapadia, "High-order finite-element method for three-dimensional turbulent Navier–Stokes," presented at 21st AIAA Computational Fluid Dynamics Conference, AIAA Paper 2013-2571, 2013.

[41] R. S. Glasby, N. Burgess, W. K. Anderson, L. Wang, D. J. Mavriplis, and S. R. Allmaras, "Comparison of SU/PG and DG finite-element techniques for the compressible Navier-Stokes equations on anisotropic unstructured meshes," presented at 51st AIAA Aerospcae Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA Paper 2013-0691, 2013.

[42] B. Kirk and T. Oliver, "Validation of SUPG finite element simulations of shockwave/turbulent boundary layer interactions in hypersonic flows," presented at 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA Paper 2013-0306, 2013.

[43] K. S. Bey, "An hp-adaptive discontinuous Galerkin method for hyperbolic conservation laws," PhD, University of Texas at Austin, 1994.

[44] K. S. Bey, A. Patra, and J. T. Oden, "hp-version discontinuous Galerkin methods for hyperbolic conservation laws: A parallel adaptive strategy," *International Journal for Numerical Methods in Engineering,* vol. 38, pp. 3889-3908, 1995.

[45] J.-F. Remacle, X. Li, N. Chevaugeon, and M. S. Shephard, "Transient mesh adaptation using conforming and non conforming mesh modifications," in *IMR*, 2002, pp. 261-272.

[46] J. C.-C. Lu, "An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method," PhD, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2005.

[47] R. Hartmann, "Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier–Stokes equations," *International Journal for Numerical Methods in Fluids,* vol. 51, pp. 1131-1156, 2006.

[48] K. J. Fidkowski and D. L. Darmofal, "An adaptive simplex cut-cell method for discontinuous Galerkin discretizations of the Navier-Stokes equations," presented at 18th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2007-3941, 2007.

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

[50] M. Woopen, G. May, and J. Schütz, "Adjoint-based error estimation and mesh adaptation for hybridized discontinuous Galerkin methods," *International Journal for Numerical Methods in Fluids,* vol. 76, pp. 811-834, 2014.

[51] L. Wang, W. K. Anderson, and J. E. Taylor, "Solutions of high-order methods for three-dimensional compressible viscous flows," presented at 42nd AIAA Fluid Dynamics Conference and exhibit, AIAA Paper 2012-2836, 2012.

[52] W. K. Anderson, L. Wang, J. C. Newman III, and S. Kapadia, "Extension of the Petrov-Galerkin time-domain algorithm for dispersive media," *IEEE Microwave and Wireless Components Letters,* vol. 23, pp. 234-236, 2013.

[53] L. Wang, W. K. Anderson, J. E. Taylor, and S. Kapadia, "High-order methods for solutions of three-dimensional turbulent flows," presented at 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA Paper 2013-856, 2013.

[54] T. J. Baker, "Mesh adaptation strategies for problems in fluid dynamics," *Finite Elements in Analysis and Design,* vol. 25, pp. 243-273, 1997.

[55] R. Balasubramanian and J. C. Newman III, "Comparison of adjoint-based and feature-based grid adaptation for functional outputs," *International Journal for Numerical Methods in Fluids,* vol. 53, pp. 1541-1569, 2007.

[56] M. Marchant and N. Weatherill, "Adaptivity techniques for compressible inviscid flows," *Computer Methods in Applied Mechanics and Engineering,* vol. 106, pp. 83-106, 1993.

[57] S. Z. Pirzadeh, "An adaptive unstructured grid method by grid Subdivision, local remeshing, and grid movement," presented at 14th AIAA Computational Fluid Dynamics Conference, AIAA Paper 99-3255, 1999.

[58] G. P. Warren, W. K. Anderson, J. L. Thomas, and S. L. Krist, "Grid convergence for adaptive methods," presented at 10th AIAA Computational Fluid Dynamics Conference, AIAA Paper-91-1592, 1991.

[59] C. Zhou and J. Ai, "Mesh adaptation for simulation of unsteady flow with moving immersed boundaries," *International Journal for Numerical Methods in Fluids,* vol. 72, pp. 453-477, 2013.

[60] N. A. Pierce and M. B. Giles, "Adjoint recovery of superconvergent functionals from PDE approximations," *SIAM review,* vol. 42, pp. 247-264, 2000.

[61] D. A. Venditti and D. L. Darmofal, "Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow," *Journal of Computational Physics,* vol. 164, pp. 204-227, 2000.

[62] R. Becker and R. Rannacher, "An optimal control approach to a posteriori error estimation in finite element methods," *Acta Numerica,* vol. 10, pp. 1-102, 2001.

[63] D. A. Venditti and D. L. Darmofal, "Grid adaptation for functional outputs: Application to two-dimensional inviscid flows," *Journal of Computational Physics,* vol. 176, pp. 40-69, 2002.

[64] D. A. Venditti and D. L. Darmofal, "Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows," *Journal of Computational Physics,* vol. 187, pp. 22-46, 2003.

[65] M. A. Park, "Adjoint-based, three-dimensional error prediction and grid adaptation," *AIAA Journal,* vol. 42, pp. 1854-1862, 2004.

[66] K. Mani and D. J. Mavriplis, "Discrete adjoint based time-step adaptation and error reduction in unsteady flow problems," presented at 18th AIAA Computational Fluid Dynamics Conference, AIAA paper 2007-3944, 2007.

[67] R. Balasubramanian and J. C. Newman III, "Adjoint-based error estimation and grid adaptation for functional outputs: Application to two-dimensional, inviscid, incompressible flows," *Computers & Fluids,* vol. 38, pp. 320-332, 2009.

[68] N. K. Burgess and D. J. Mavriplis, "An hp-adaptive discontinuous Galerkin solver for aerodynamic flows on mixed-element meshes," presented at 49th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2011-0490, 2011.

[69] K. J. Fidkowski and D. L. Darmofal, "Review of output-based error estimation and mesh adaptation in computational fluid dynamics," *AIAA journal,* vol. 49, pp. 673-694, 2011.

[70] K. J. Fidkowski and Y. Luo, "Output-based space–time mesh adaptation for the compressible Navier–Stokes equations," *Journal of Computational Physics,* vol. 230, pp. 5753-5773, 2011.

[71] B. T. Flynt and D. J. Mavriplis, "Discrete adjoint based adaptive error control in unsteady flow problems," presented at 50th AIAA Aerospcae Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA Paper 2012-0078, 2012.

[72] B. T. Flynt and D. J. Mavriplis, "Efficient adaptation using discrete adjoint error estimates in unsteady flow problems," presented at 51st AIAA Aerospcae Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA Paper 2013-0520, 2013.

[73] B. R. Ahrabi, W. K. Anderson, and J. C. Newman III, "High-order finite-element method and dynamic adaptation for two-dimensional laminar and turbulent Navier-Stokes," presented at 32nd AIAA Applied Aerodynamics Conference, AIAA Paper 2014-2983, 2014.

[74] Y. Bazilevs and I. Akkerman, "Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual-based variational multiscale method," *Journal of Computational Physics,* vol. 229, pp. 3402-3414, 2010.

[75] Y. Bazilevs and T. J. Hughes, "Weak imposition of Dirichlet boundary conditions in fluid mechanics," *Computers & Fluids,* vol. 36, pp. 12-26, 2007.

[76] Y. Bazilevs, C. Michler, V. Calo, and T. Hughes, "Weak Dirichlet boundary conditions for wall-bounded turbulent flows," *Computer Methods in Applied Mechanics and Engineering,* vol. 196, pp. 4853-4862, 2007.

[77] J. Nitsche, "Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind," *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg,* vol. 36, pp. 9-15, 1971.

[78] M. F. Wheeler, "An elliptic collocation-finite element method with interior penalties," *SIAM Journal on Numerical Analysis,* vol. 15, pp. 152-161, 1978.

[79] G. E. Barter, "Shock capturing with PDE-based artificial viscosity for an adaptive, higher-order discontinuous Galerkin finite element method," PhD, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2008.

[80] G. E. Barter and D. L. Darmofal, "Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation," *Journal of Computational Physics,* vol. 229, pp. 1810-1827, 2010.

[81] P.-O. Persson and J. Peraire, "Sub-cell shock capturing for discontinuous Galerkin methods," presented at 44$^{\text{th}}$ AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2006-0112, 2006.

[82] M. Ainsworth and B. Senior, "Aspects of an adaptive hp-finite element method: Adaptive strategy, conforming approximation and efficient solvers," *Computer Methods in Applied Mechanics and Engineering,* vol. 150, pp. 65-87, 1997.

[83] L. Demkowicz, K. Gerdes, C. Schwab, A. Bajer, and T. Walsh, "HP90: A general and flexible Fortran 90 hp-FE code," *Computing and Visualization in Science,* vol. 1, pp. 145-163, 1998.

[84] P. Solin, K. Segeth, and I. Dolezel, *Higher-Order Finite Element Methods*: CRC Press, 2003.

[85] L. Demkowicz, *Computing with hp-Adaptive Finite Elements: Volume 1 One and Two Dimensional Elliptic and Maxwell problems*: CRC Press, 2006.

[86] M. Ainsworth, L. Demkowicz, and C.-W. Kim, "Analysis of the equilibrated residual method for a posteriori error estimation on meshes with hanging nodes," *Computer Methods in Applied Mechanics and Engineering,* vol. 196, pp. 3493-3507, 2007.

[87] P. Šolín, J. Červený, and I. Doležel, "Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM," *Mathematics and Computers in Simulation,* vol. 77, pp. 117-132, 2008.

[88] W. Bangerth and O. Kayser-Herold, "Data structures and requirements for *hp* finite element software," *ACM Transactions on Mathematical Software,* vol. 36, pp. 1-31, 2009.

[89] 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)*, 2012, pp. 1-11.

[90] J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*: John Wiley & Sons, 2003.

[91] T. J. Hughes, L. P. Franca, and G. M. Hulbert, "A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations," *Computer Methods in Applied Mechanics and Engineering,* vol. 73, pp. 173-189, 1989.

[92] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy, "The variational multiscale method—a paradigm for computational mechanics," *Computer Methods in Applied Mechanics and Engineering,* vol. 166, pp. 3-24, 1998.

[93] S. Marras, J. F. Kelly, F. X. Giraldo, and M. Vázquez, "Variational multiscale stabilization of high-order spectral elements for the advection–diffusion equation," *Journal of Computational Physics,* vol. 231, pp. 7187-7213, 2012.

[94] C. H. Whiting and K. E. Jansen, "A stabilized finite element method for the incompressible Navier-Stokes equations using a hierarchical basis," *International Journal for Numerical Methods in Fluids,* vol. 35, pp. 93-116, 2001.

[95] C. H. Whiting, K. E. Jansen, and S. Dey, "Hierarchical basis for stabilized finite element methods for compressible flows," *Computer Methods in Applied Mechanics and Engineering,* vol. 192, pp. 5167-5185, 2003.

[96] J. Gressier, P. Villedieu, and J.-M. Moschetta, "Positivity of flux vector splitting schemes," *Journal of Computational Physics,* vol. 155, pp. 199-220, 1999.

[97] C. Wang and J. Liu, "Positivity property of second-order flux-splitting schemes for the compressible Euler equations," *Discrete And Continuous Dynamical Systems Series B,* vol. 3, pp. 201-228, 2003.

[98] T. J. Barth, "Numerical methods for gasdynamic systems on unstructured meshes," in *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*, ed: Springer, 1999, pp. 195-285.

[99] F. Shakib, T. J. Hughes, and Z. Johan, "A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering,* vol. 89, pp. 141-219, 1991.

[100] C. Rumsey. *Turbulence Modeling Resource Website.* Available: http://turbmodels.larc.nasa.gov

[101] P. L. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics,* vol. 43, pp. 357-372, 1981.

[102] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, "Unified analysis of discontinuous Galerkin methods for elliptic problems," *SIAM Journal on Numerical Analysis,* vol. 39, pp. 1749-1779, 2002.

[103] A. Walther, A. Griewank, and O. Vogel, "ADOL-C: Automatic differentiation using operator overloading in C++," *PAMM,* vol. 2, pp. 41-44, 2003.

[104] N. K. Burgess and R. S. Glasby, "Advances in numerical methods for CREATETM-AV analysis tools," presented at 52nd AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2014-0417, 2014.

[105] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing,* vol. 7, pp. 856-869, 1986.

[106] Y. Saad, *Iterative Methods for Sparse Linear Systems*: Siam, 2003.

[107] N. K. Burgess and D. J. Mavriplis, "Computing shocked flows with high-order accurate discontinuous Galerkin methods," presented at 42nd AIAA Fluid Dynamics Conference and Exhibit, AIAA Paper 2012-2715, 2012.

[108] W. K. Anderson and D. L. Bonhaus, "An implicit upwind algorithm for computing turbulent flows on unstructured grids," *Computers & Fluids,* vol. 23, pp. 1-21, 1994.

[109] W. K. Anderson, R. D. Rausch, and D. L. Bonhaus, "Implicit/multigrid algorithms for incompressible turbulent flows on unstructured grids," *Journal of Computational Physics,* vol. 128, pp. 391-408, 1996.

[110] E. J. Nielsen and W. K. Anderson, "Recent improvements in aerodynamic design optimization on unstructured meshes," *AIAA Journal,* vol. 40, pp. 1155-1163, 2002.

[111] S. M. Klausmeyer and J. C. Lin, "Comparative results from a CFD challenge over a 2D three-element high-lift airfoil," *NASA Technical Memo,* vol. 112858, 1997.

[112] P. H. Cook, M. A. McDonald, and M. C. P. Firmin, "Aerofoil RAE 2822 - pressure distributions, and boundary layer and wake measurements," in *Experimental Data Base for Computer Program Assessment*, ed: AGARD Report AR 138, 1979.

VITA

Behzad Reza Ahrabi was born in September of 1983 in Ahwaz, Iran. He graduated from Dr. Hesabi High School in 2001. He earned a Bachelor of Science in Mechanical Engineering from Isfahan University of Technology in September 2005. In February 2008, he earned a Master of Science in Mechanical Engineering from Amirkabir University of Technology. In August 2011, he started his PhD in Computational Engineering at the UTC Simcenter: the National Center for Computational Engineering. In August 2015, he earned his PhD and was employed in Mechanical Engineering department of University of Wyoming to follow his research in computational methods as a post-doctoral researcher.