Entropy Adjoint Mesh Adaptation for the Discontinuous Galerkin Approach

Cole Bradley

Supervisor: Professor Siva Nadarajah

Department of Mechanical Engineering

McGill University

Montréal, Québec, Canada

December 5, 2023

A thesis submitted to McGill University in partial fulfilment of the requirements of the Undergraduate Honours Program.

Abstract

Goal-oriented mesh adaptation is a technique to create meshes optimized to compute accurate values for specific outputs and flow characteristics of interest in CFD problems. The goal of this thesis was to implement entropy adjoint equations for goal oriented mesh adaptation to have a low computational-cost, yet efficient indicator. This was done using the Discontinuous Galerkin method as the numerical scheme and the Dual-Weighted Residual to determine cell-wise error. It was shown that the entropy adjoint indicator succesfully adapts the mesh in areas of interest around two-dimensional NACA-0012 airfoils as well as captures entropy generating features in inviscid flow, namely shocks.

Abrégé

L'adaptation de maillage orientée vers un objectif est une technique qui permet de créer des maillages optimisés pour calculer des valeurs précises pour des sorties spécifiques et des caractéristiques d'écoulement intéressantes dans les problèmes de CFD. L'objectif de cette thèse était de mettre en œuvre les équations adjointes d'entropie pour l'adaptation de maillage orientée vers un objectif afin d'avoir un indicateur efficace à faible coût de calcul. Pour ce faire, la méthode Discontinuous Galerkin a été utilisée comme schéma numérique et le Dual-Weighted Residual pour déterminer l'erreur au niveau des cellules. Il a été démontré que l'indicateur adjoint d'entropie adapte avec succès le maillage dans les zones d'intérêt autour des profils aérodynamiques bidimensionnels NACA-0012 et capture les caractéristiques génératrices d'entropie dans les écoulements inviscides, à savoir les chocs.

Acknowledgements

I would like to express my utmost gratitude to Professor Siva Nadarajah for introducing me to computational fluid dynamics, mentoring me throughout this project, and always being available when I needed help. I also would like to thank the members of McGill's Computational Aerodynamics Research Group for helping me throughout my project, notably Pranshul Thakur for always being willing to answers my questions, work through bugs in my code, and help expand my knowledge. Additionally, I would like to thank my friends and family for always being supportive throughout the process of developing this thesis. Finally, I would like to thank my girlfriend Rylee for believing in me throughout the process and always providing support.

Contents

Chapter 1

Introduction

1.1 Motivation

Computational fluid dynamics (CFD) is a method of analysis that utilizes numerical methods and computational tools to allow researchers and industry professionals to study fluid flow. This is essential as fluid flow phenomena seen in the real world are almost always defined by highly nonlinear partial differential equations (PDEs) that cannot be solved analytically. Numerical methods can be used to discretize and solve these equations. Advancement of High Performance Computing (HPC) has allowed for more complex problems to be solved using numerical methods. However, there is still emphasis in the field on the importance of efficient methods that reduce computational costs. Additionally, due to the discretized nature of numerical solutions, inconsistencies compared to natural phenomena are developed resulting in flow characteristics and changes not being accurately captured. These problems that arise in CFD are the basis for a variety of research topics that have the goal of building robust, high-order numerical solvers that efficiently produce accurate solutions.

High-order Discontinuous Galerkin Methods satisfy the requirements of the advancement of robust and accurate solutions. These methods are a powerful combination of the finite volume and finite element methods that are commonly used to solve PDEs. This allows for them to have highly accurate and adaptive solutions that can solve conservation law problems [1]. One of the key properties of the discontinuous galerkin method is its hp-discontinuous property. Spatial discretizations of the domain are referred to as h and polynomial order for the basis functions is referred to as p. These can be adjusted to develop a more accurate solution. This adjustment is referred to as mesh adaptation and will be the focus of this paper. Mesh adaptation is essential to capturing flow characteristics that tend to increase error within numerical solutions [2]. Common examples of these are shock waves.

1.1.1 Shock Waves

Shock waves are common phenomena that occur when fluid flows over a wing. Shock waves are caused by transonic flow which is a flow speed that causes both supersonic and subsonic flow regions about the aircraft wing [3]. The sudden change from supersonic to subsonic flow develops a shock. This change in flow properties including flow speed and pressure happens over an infinitesimal distance. This introduces problems in producing accurate numerical representations of fluid flow over the shock as they are modelled as discontinuities. Figure 1.1 depicts shock waves developed by fluid flow over supersonic jets.

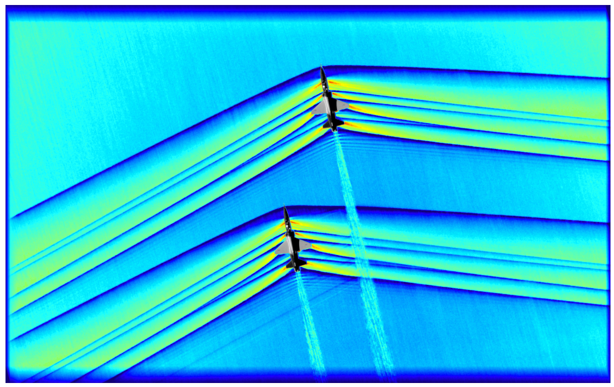


Figure 1.1: Oblique shocks from supersonic jets. Figure from [4]

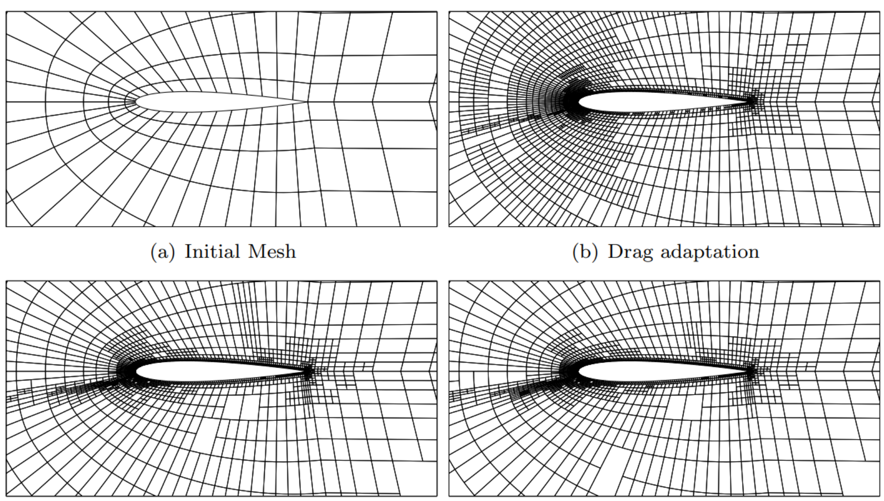
These sudden changes in flow properties caused by shocks can introduce instabilities into our solution. These instabilities can be seen by the production of non-physical solutions. It is common for methods that aren't sufficiently robust to allow shock waves to create negative density regions which are not possible in nature [5]. These instabilities and non-physical results need to be accounted for to avoid divergence and ensure an accurate solution is obtained. Limiters are commonly used to adjust the computation of the solution and stop non-physical results [6]. Artificial viscosity is also introduced as this creates dissipation around the shock allowing it to be captured in high-order schemes [7]. Complex flow features can also be captured by creating very fine grids. This however greatly increases computational costs. As a result, mesh adaptation is commonly used to build efficient grids. This allows for fine grid sections near areas of the flow that are primarily responsible for inaccuracies in the solution and coarse sections where the flow is stable and easily captured by the numerical scheme.

There are two primary forms of mesh adaptation: feature based and goal oriented. Feature based, also called hessian-based, focuses on refining based on the elements with the largest residual [2]. Goal oriented introduces a function of interest allowing for the mesh to be adapted to optimize for this function [2]. Due to entropy stability in smooth solutions and the conservation of entropy in invscid flow, entropy variables are a prime candidate to be a functional of interest. As a result, this thesis will focus on implementing entropy-adjoint mesh adaptation so that areas where spurious entropy is generated will be targeted allowing, for more efficient convergence to a numerical solution and better capturing of flow characteristics of interest.

1.2 Objective

It is essential for flow features of interest to be captured in the solution to build a strong understanding of the problems being analyzed. The objective of this thesis is to develop an entropy adjoint equation that is paired to the primal problem presented by the flows partial differential equation. This dual problem can be solved to find an adjoint solution that allows for identification of areas of the grid that have entropy generation causing increased inaccuracy within the solution. The mesh can then be adapted using a fixed fraction method to refine the mesh in areas causing instability in the solution allowing for a grid that can be solved to find a more accurate solution.

This entropy-adjoint mesh adaptation will be investigated on three cases of fluid flow over a NACA-0012 airfoil represented by Eulers equations. For transonic flow, this mesh adaptation will allow for better capturing of the shock developed on the top surface. For subsonic flow, regions where erroneous numerical entropy generation occurs will be targeted. This will then be compared with lift adaptations, drag adaptations, and feature based adaptations. An example of goal oriented mesh adaptation can be seen in figure 1.2 where various adjoint functionals were used for mesh adaptation to allow for a more accurate solution.



(a) Initial Mesh (b) Drag adaptation

(c) Lift adaptation (d) Moment adaptation

Figure 1.2: Mesh adaptation for various adjoint funtionals. Figure from [8].

1.3 Overview

This thesis is organized as follows. Chapter 2 describes the Discontinuous Galerkin method which is the numerical scheme that is used to solve the fluid dynamics problems posed. It also shows the derivation for the Euler equation in its conservative form and how it is adjusted for the DG scheme. Chapter 3 gives an extensive overview of adjoint equations, entropy adjoint variables, the Dual-Weighted Residual and how these are used for goal-oriented, fixed-fraction mesh adaptation. Chapter 4 presents and discusses the results of the test cases explored using entropy adjoint mesh adaptation. Chapter 5 is the final chapter, providing a conclusion as well as future work.

Chapter 2

The Discontinuous Galerkin Method

2.1 The Discontinuous Galerkin Method

There are various numerical high order schemes that are used for solving CFD problems. These include the Finite Difference Method (FD), the Finite Volume Method (FV), and the Finite Element Method (FE) [9]. The FD method is a very commonly used numerical method where the grid is discretized and then solutions are found for each discrete point on the resulting grid. The grid can then step through time to see how the solution evolves for each time step. This method is generally considered the simplest to implement of the three [9]. The FV method is another commonly used numerical method. This method divides the domain into cells instead of individual grid points. The solution is then calculated via dependencies such as flux between the cells. The solution for each cell is then assumed to be the average of the solution over the cell creating a constant value for the entire element [9]. This means that neighbouring cells will have different averages and as a result, the solution will be discontinuous between cells. These discontinuities are advantageous as it means that each cell is its own system and a global system doesn't need to be solved. This also allows for high performance computing clusters to be used to solve the solution as it can be divided into sub-grids that are individually solved on separate nodes and the combined once a solution for the sub-grids is found [9]. The third common method, the FE method, divides the domain into cells and uses a set of basis functions to approximate the solution in each cell, differing from the FV method as it doesn't take the average, but attempts to approximate the solution as a linear combination of the chosen basis functions. Additionally, the FEM is continuous at the cell boundaries. The numerical method used for discretizing and solving problems in this thesis is the Discontinuous Galerkin method which is a combination of the FV method and the FE method.

The Discontinuous Galerkin method (DG) was first used by Reed and Hill to solve linear hyperbolic advection equations that modelled neutron transfer [10]. The DG method is a versatile high-order numerical scheme that is highly parallelizable, helping to reduce its computational time compared to other schemes [1]. This is possible because it is discontinuous at the boundaries of elements similar to the FV Method. However, the DG method is similar to the FE method as it uses approximates the solution for each cell as a linear combination of a set of chosen basis functions. This allows the DG method to solve for high-order solutions without a large stencil and accurate approximate solutions across the entire domain that are only dependent on neighbouring cells.

The DG method has been derived to cover a significant number of differential equations such as conservation laws. Conservation laws include linear advection problems, Euler equations, and the Navier-Stokes equations [1]. This makes it ideal for solving computational fluid dynamics problems and the problems posed in this thesis. Because of this it is important to build an understanding of how the method is developed.

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \quad x \in [L, R] = \Omega \tag{2.1}$$

The simplest equation to derive the solution using the DG method is the linear advection equation given in equation 2.1. In this equation there is a linear flux term given by $f(u)$, a solution given by $u(x,t)$, and the domain is represented by the symbol Ω. As explained earlier, the domain is then split into elements.

$$\Omega \approx \Omega_h = \bigcup_{k=1}^{K} D^k \tag{2.2}$$

From equation 2.2 the domain is seen to be split into K cells. These cell-wise solutions are approximated using basis functions. This set of basis functions is labelled as ψ_n and the number of basis functions corresponds to p which is the polynomial order of the solution. The approximated solution can then be written in the form:

$$x \in D^k : \quad u_h^k(x,t) = \Sigma_{i=1}^{N_p} u_h^k \psi_n(x) \tag{2.3}$$

The flux for each element is directly dependent on the solution u_h^k so we only have u_h^k as the set of unknowns that must be solved for to find a solution for our problem. The result of this will be a set of solutions for each cell. The solution over the entire domain can then be written as the direct sum of the cell-wise solutions and is of the form:

$$u(x,t) \approx u_h(x,t) = \oplus_{k=1}^{K} u_h^k(x,t) \tag{2.4}$$

To solve for our approximate solutions u_h^k, we multiply the original PDE in equation 2.1 by a test function that is carefully selected from the set of basis functions and integrate over the element to get the following formulation:

$$\int_{D_k} \left(\frac{\partial u_h^k}{\partial t}\phi_m + \frac{\partial f_h^k}{\partial x}\phi_m\right)dx = 0 \tag{2.5}$$

After this step, we can integrate by parts, yielding the weak form of the DG method:

$$\int_{D_k} \left(\frac{\partial u_h^k}{\partial t}\phi_m - f_h^k \frac{\partial \phi_m}{\partial x}\right)dx = -[f^* \phi_m]_{x^k}^{x^{k+1}} \tag{2.6}$$

This is what can be reformulated into a flux equation that is solvable. Integrating by parts again yields the strong-form of the DG method, however in this thesis only the weak

DG form is used and so the strong form will not be considered. To simplify solving and to allow a stress and mass matrix to be built in the next steps, we can set the test and basis functions to be equivalent. There are various basis polynomials that can be used. These include Lagrange polynomials, Legendre polynomials and many more. These are chosen to make sure our matrices do not become ill-conditioned [1].

$$\psi_j(x) = \phi_j(x) = l_j^k(x) = \Pi_{i=0, i\neq j}^{p} \frac{x - x_i}{x_j - x_i} \tag{2.7}$$

After choosing an interpolating polynomial such as the lagrange polynomials seen in equation 2.7, they can be subbed into equation 2.6 to get the following important formulation which represents the weak DG solution in terms of the interpolating polynomials re-written in its operator form:

$$M^k \frac{du_h^k}{dt} + S^k f_h^k = (f_h^k - f^*) l^k \Big|_{x^k}^{x^{k+1}} \tag{2.8}$$

In this form we observe the mass and stiffness matrix. These are computed as follows:

$$M_{ij}^k = \int_k l_i^k(x) l_j^k(x) dx \tag{2.9}$$

$$S_{ij}^k = \int_k l_i^k(x) \frac{dl_j^k(x)}{dx} dx \tag{2.10}$$

In equation 2.8 we also see the flux differentials between elements. This is calculated using a chosen flux scheme. This adaptivity of the DG method allows for various types of flux schemes such as upwind or downwind schemes depending on the requirements of a given problem [1]. Equation 2.8 can be solved and then integrated through time using various classic ODE solvers. In this thesis, implicit ODE solvers are used as they are adequate enough to satisfy the requirements, but explicit solvers such as Runge-Kutta are commonly used in DG schemes such as when solving using the strong-form [11]. The main requirement of the time step size is that it must satisfy the Courant-Friedrichs-Lewy (CFL) condition or the scheme becomes unstable and will diverge [12]. The optimal rate of convergence for this scheme is $O(\Delta x^{p+1})$. DG schemes can then be extended into 3-dimensions. For conservation laws, this can be expressed in the following form:

$$\mathcal{R}(u) = \frac{\partial u}{\partial t} + \nabla \cdot F(u) - S(u) = 0 \tag{2.11}$$

In this form $\mathcal{R}$ represents the residual, $F(u)$ represents the flux, and $S(u)$ represents the source term. The next section of this thesis will review how this is applied to the Euler equations.

2.2 Euler Equations

Eulers equations are derived from the governing equations of fluid dynamics, the Navier-Stokes equations, and represents fluid flow that is inviscid and adiabatic [13]. This means that fluid viscosity and heat transfer are not accounted for in the fluid flow. The Euler equations are as follows:

$$\frac{\partial \rho}{\partial t} + \vec{u} \cdot \nabla \rho + \rho \nabla \cdot \vec{u} = 0 \tag{2.12}$$

$$\frac{\partial (\rho \vec{v})}{\partial t} + \rho \vec{u} \cdot \nabla \vec{u} + \nabla p = 0 \tag{2.13}$$

$$\frac{\partial (\rho e)}{\partial t} + \vec{u} \cdot \nabla e + p \nabla \cdot \vec{u} = 0 \tag{2.14}$$

Equations 2.12, 2.13, and 2.14 correspond respectively to the conservation of mass, conservation of momentum, and conservation of energy laws seen for fluid flow. In these equations ρ refers to the density, $\vec{u}$ is the velocity vector, p is the fluid pressure, and e is specific energy of the fluid. These can be rewritten in the conservative form seen in equation 2.11 as follows:

$$\frac{\partial W}{\partial t} + \nabla \cdot F = 0 \tag{2.15}$$

As expected, in equation 2.15 it can be seen that there are no source terms. W and F are also observed. W is a vector that represents the conservative solution set to the Euler equations at the given time step. F is a matrix that represents the convective flux at the given time step. This form allows the solution W to step through time as a function of the convective flux represented by $\nabla \cdot F$. This is how the DG method is able to numerically solve and provide accurate solutions to the Euler equations. The conservative solution W is represented as follows:

$$W = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}$$

The convective source matrix F can be separated into the following three vectors:

$$F_x = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uh \end{bmatrix}, \quad F_y = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vh \end{bmatrix}$$

This reformulation of the convective flux allows the conservative form seen in equation 2.15 to be rewritten in the following simplified form:

$$\frac{\partial W}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = \bar{0} \tag{2.16}$$

Equation 2.16 is used when solving the Euler equations with PHiLiP (Parallel High-Order Library for PDEs), the McGill Computational Aerodynamics research group code.

Chapter 3

Goal Oriented Mesh Adaptation

In mesh adaptation there are two common types of mesh adaptation: feature based and goal oriented [14]. Feature based, also known as hessian based, takes features from the primal flow solution and uses these to optimize the mesh adaptation. This works by using the residual of the solution as an error estimate and then adapting according to this using a mesh adaptation technique such as fixed-fraction. The goal of feature based mesh adaptation is to minimize interpolation error [14]. While successful at this and adapting to improve the overall solution, it doesn't adapt the mesh to better identify error in observable design variables such as lift and drag [15]. It also doesn't help adapt the mesh to accurately show the existence of flow characteristics of interest such as shocks. As a result, there has been an increase in research into goal-oriented mesh adaptation [8, 15, 16]. Goal-oriented mesh adaptation takes a function of interest such as the entropy equations and solves the adjoint problem. Methods such as the Dual-Weighted Residual are then used to adapt the mesh according to the adjoint solution. This allows for characteristics such as entropy generation around shocks to be identified in the mesh adaptation and to adapt accordingly [17]. As goal-oriented mesh adaptation is the primary method analyzed in this thesis, its formulation will be the focus of this chapter.

3.1 Adjoint Problems

3.1.1 Adjoint Equations

Adjoint equations are a dual problem that is derived from the primal flow solution. Certain design variables can be chosen and the sensitivity of these variables is used to determine an optimal mesh design [18]. These design functions are written in the following form:

$$\mathcal{J} = \int_{\Omega} g_{\Omega}(u) d\Omega + \int_{\Gamma} g_{\Gamma}(u) d\Gamma \tag{3.1}$$

The form shown in equation 3.1 shows the functional has a domain integral represented with Ω and a boundary integral represented with Γ. Different functionals of choice such as lift, drag, and entropy functionals can be implemented into this equation to create a dual problem. This results in an adjoint solution. This dual problem can be formulated as an optimization problem.

$$\begin{aligned} \min_{x} \quad & \mathcal{J}(u,x) \\ \text{s.t.} \quad & \mathcal{R}(u,x) = 0, \end{aligned} \tag{3.2}$$

In the optimization problem the functional $\mathcal{J}$ is optimized with respect to the design variable x. This optimization is constrained by the residual of the primal flow solution. This optimization problem can be re-formulated using Lagrange multipliers to give the following problem:

$$\mathcal{L}(u,x,\psi) = \mathcal{J}(u,x) - \int_{\Omega} \psi^T \mathcal{R}(u,x) d\Omega \tag{3.3}$$

We can then take the derivative of this equation with respect to the design variable resulting in the following problem:

$$\frac{\partial \mathcal{L}}{\partial x} = [\frac{\partial \mathcal{J}}{\partial x} - \psi^T \frac{\partial \mathcal{R}}{\partial x}] + (\frac{\partial u}{\partial x})^T [\frac{\partial \mathcal{J}}{\partial u} - \psi^T \frac{\partial \mathcal{R}}{\partial u}] \tag{3.4}$$

An optimal solution for this equation is when it is equal to zero. To achieve this, the first and second term need to be equal to zero. This is done by taking the second term and equating it to determine the following result

$$[\frac{\partial \mathcal{R}}{\partial u}]\psi = (\frac{\partial \mathcal{J}}{\partial u})^T \tag{3.5}$$

which is the adjoint problem with the adjoint solution represented as ψ. This is a problem that requires a system of linear equations to be solved.

3.1.2 Entropy Adjoint Equations

One choice for the adjoint functional is the entropy equation. This is because entropy is conserved across the boundaries of a system unless entropy is generated [19]. This results in the entropy adjoint equations being able to identify locations of entropy generation such as shocks. This makes the choice of entropy as a functional well-suited to mesh-adaptation that wants to capture this flow feature. The entropy function is defined as follows:

$$U = \frac{-\rho S}{\gamma - 1}, \quad S = ln(\frac{p}{\rho^{\gamma}}) \tag{3.6}$$

The derivation of the entropy functional begins by defining the entropy flux as F_i and the entropy function as $U(\vec{u})$. Entropy conservation laws state that for inviscid flow $\partial_i F_i = 0$ [19]. The entropy variables are the calculated as

$$\vec{v} \equiv U_u^T \tag{3.7}$$

where the entropy variables are the vector $\vec{v}$ and are equivalent to the derivative of the chosen entropy function with respect to the solution. This allows the entropy variables to be defined as follows for an i-dimensional grid:

$$\vec{v} = U_{\vec{w}}^T = \begin{bmatrix} \frac{\gamma-S}{\gamma-1} - \frac{1}{2}\frac{\rho V^2}{p} \\ \\ \frac{\rho v_i}{p} \\ \\ -\frac{\rho}{p} \end{bmatrix} \tag{3.8}$$

A flux Jacobian matrix can then be defined as A_i resulting in the entropy flux relation $U_{\vec{u}} A_i = (F_i)_{\vec{u}}$. The transformation matrix is then defined as the derivative of the primal solution with respect to the entropy variables and is represented as $\vec{u}_{\vec{v}}$. These satisfy the following two properties [8]:

1. The transformation matrix is symmetric and positive definite.
2. The resulting quantity $A_i \vec{u}_{\vec{v}}$ is symmetric.

This allows the conservation law to be linearized and results in the following important relations taken from [8]:

$$A_i \partial_i \vec{u} = A_i \vec{u}_{\vec{v}} \partial_i \vec{v} = (A_i \vec{u}_{\vec{v}})^T \partial_i \vec{v} = \vec{u}_{\vec{v}} A_i^T \partial_i \vec{v} = \vec{u}_{\vec{v}} A_i^T \partial_i \vec{v} = 0 \Rightarrow A_i^T \partial_i \vec{v} = 0 \tag{3.9}$$

This relation is important when substituting to see that the domain term cancels out in the functional. Fidkowski and Roe showed that using perturbation methods equation 3.3 could be transformed into the following [8]:

$$\mathcal{J}'(\delta \vec{u}) - \int_{\Omega} \psi^T \mathcal{R}'(\vec{u})(\delta \vec{u}) d\Omega = 0 \tag{3.10}$$

This equation can then be integrated by parts to recieve the following formulation where both the domain and the boundary adjoint results can be seen:

$$\mathcal{J}'(\delta\vec{u}) + \int_{\Omega} \partial_i \psi^T F_i'(\delta\vec{u}) d\Omega - \int_{\Gamma} \psi^T F_i'(\delta\vec{u}) n_i d\Gamma = 0 \tag{3.11}$$

This adjoint lagrangian then implemented using entropy variables yields the following results:

$$\mathcal{J}'(\delta\vec{u}) + \int_{\Omega} \partial_i \vec{v}^T A_i(\delta\vec{u}) d\Omega - \int_{\Gamma} \vec{v}^T A_i(\delta\vec{u}) n_i d\Gamma = 0 \tag{3.12}$$

It can then be seen from the relation yielded in 3.9 that the domain integral becomes zero and only the boundary integral must be solved for. The final step for finding the entropy functional is to transform this form into one that incorporates the entropy flux:

$$\mathcal{J}'(\delta\vec{u}) = \int_{\Gamma} \vec{v}^T A_i(\delta\vec{u}) n_i d\Gamma = \delta[\int_{\Gamma} F_i n_i d\Gamma] \tag{3.13}$$

This gives us the resulting equation for the entropy functional in terms of the entropy flux:

$$\mathcal{J} = \int_{\Omega} F_i(u_h^b) n_i d\Omega \tag{3.14}$$

It is also important to note that comparing equation 3.13 to equation 3.11 that the entropy variables are equivalent to the adjoint solution. Because of this either the entropy variables can be used as the adjoint solution or a linear solver can be used with the entropy functional to calculate the adjoint solution. The reduced computational costs of the entropy transport equation due to not having to solve a linear system at every adaptation step is where its advantage lies. After this, an error estimate for each cell needs to be determined. This is done by finding the Dual Weighted Residual.

3.2 Dual Weighted Residual

The next step to mesh adaptation is to use a method that calculates the error for each cell. The Dual-Weighted Residual (DWR) is a commonly used method for determining cell-wise error in goal-oriented mesh adaptation [20, 21]. This method begins by defining a a flow solution for a coarse grid and a fine grid which can be represented as u_H and u_h respectively. The residuals for these two grids are then calculated

$$\mathcal{R}_H(u_H) = 0, \quad \mathcal{R}_h(u_h) = 0 \tag{3.15}$$

where the fine grid is created as a more refined space using h or p adaptations. In this research a p+1 solution is used for the fine grid as this is what is currently implemented for the DWR in the group code PHiLiP. Once these two solutions are created a prolongation operator is determined which transfers the coarse grid to a fine grid. This is defined as $u_h^H = I_h^H u_H$

where I_h^H is the prolongation operator. The residual of the coarse grid projection to the fine grid is then approximated using a Taylor's series expansion that neglects higher-order terms

$$\mathcal{R}_h(u_h) = 0 \approx \mathcal{R}_h(u_h^H) + [\frac{\partial \mathcal{R}_h}{\partial u_h}\bigg|_{u_h^H}](u_h - u_h^H) \tag{3.16}$$

resulting in:

$$[\frac{\partial \mathcal{R}_h}{\partial u_h}\bigg|_{u_h^H}](u_h - u_h^H) = -\mathcal{R}_h(u_h^H) \tag{3.17}$$

The same procedure can be done for the functional of interest resulting in a similar Taylor series expansion:

$$\mathcal{J}_h(u_h) \approx \mathcal{J}_h(u_h^H) + [\frac{\partial \mathcal{J}_h}{\partial u_h}\bigg|_{u_h^H}](u_h - u_h^H) \tag{3.18}$$

The fine grid solution for the functional is then assumed to be a sufficient approximation of the true solution allowing the functional error to be expressed as the the difference between these two:

$$\mathcal{J}_h(u_h) - \mathcal{J}_h(u_h^H) \approx \mathcal{J}(u) - \mathcal{J}_h(u_h^H) = (\frac{\partial \mathcal{J}_h}{\partial u_h}\bigg|_{u_h^H})(u_h - u_h^H) \tag{3.19}$$

Analyzing equation 3.19 and 3.19 the following relation is determined:

$$[\frac{\partial \mathcal{R}_h}{\partial u_h}\bigg|_{u_h^H}]\psi_h = (\frac{\partial \mathcal{J}_h}{\partial u_h}\bigg|_{u_h^H})^T \tag{3.20}$$

In this equation ψ_h is the adjoint solution for the fine grid. Substituting 3.17 into 3.20 the error can be written as:

$$\mathcal{J}_h(u_h) - \mathcal{J}_h(u_h^H) = -\psi_h^T \mathcal{R}_h(u_h^H) \tag{3.21}$$

This allows the cell-wise error represented by η_k to be represented and bound as follows:

$$\eta_k = |(\psi_h^T)_k (\mathcal{R}_h(u_h^H))_k| \tag{3.22}$$

$$|\mathcal{J}_h(u_h) - \mathcal{J}_h(u_h^H)| \leq \Sigma_k |(\psi_h^T)_k (\mathcal{R}_h(u_h^H))_k| \tag{3.23}$$

From this it can be seen that the cell-wise error is a function of the adjoint solution on the fine grid and the residual on the coarse grid. This cell-wise error calculation is the final step prior to mesh adaptation. It is important as it allows the mesh adaptation software to identify elements of interest that are significantly affecting the sensitivities of the functional.

3.3 Grid Refinement Strategy

There are many types of mesh adaptation strategies. These include uniform, and fixed fraction grid refinement. Uniform is a simple technique where the grid is refined uniformly throughout [22]. This is effective at building a more accurate solution, however it is very inefficient. This is because elements that are contributing minimally to the error of the solution are further refined. This is unnecessary for determining flow features of interest and increases computational costs. As a result, fixed fraction is the grid refinement strategy used in this thesis.

The fixed-fraction grid refinement technique can be used for both feature-based and goal-oriented error-indicators. The requirement of it is that cell-wise error is computed. This is because the fixed-fraction method requires a fraction that indicates the percentage of elements that are to be refined. It is also possible to have a very small percentage of cells that are coarsened at each mesh adaptation iteration, however this won't be explored in this thesis [23]. The number of elements with the highest element-wise error corresponding to the fraction are chosen to be refined. These are then refined via h or p adaptation. In this thesis h adaptation will be the focus. This works by splitting the identified elements isotropically into 4 smaller elements for quad meshing [24].

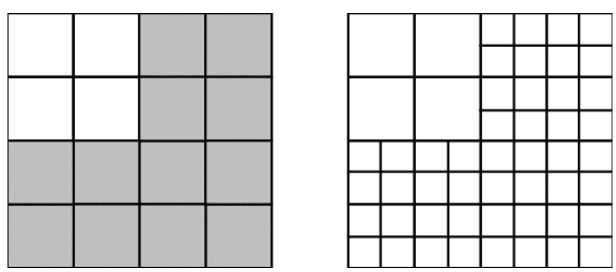


Figure 3.1: Fixed Fraction Mesh Refinement for $f = 0.75$. Figure from [25].

Chapter 4

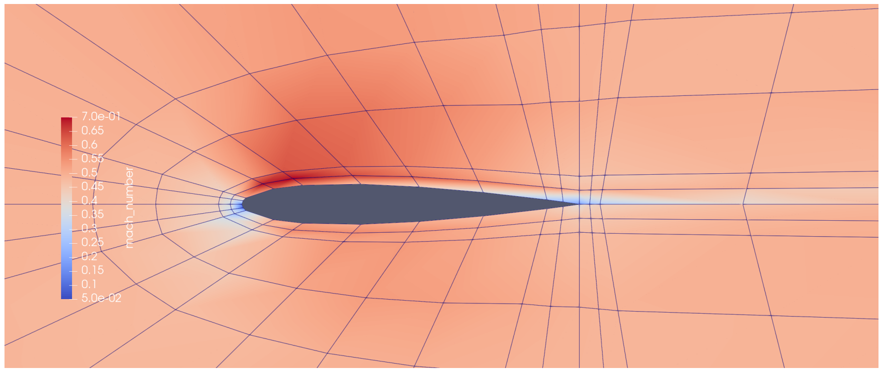
Results

To analyze the results of the entropy adjoint mesh adaptation, it was used on three test cases. The tests were steady state flow analysis using two-dimensional grids. Two subsonic test cases with different angles of attack, and a transonic test case, all using NACA-0012 airfoils were run. The cells-wise solutions were approximated with $p = 1$ order polynomials. The entropy adjoint indicator is then compared to other design indicators lift and drag, as well as feature based mesh adaptation.

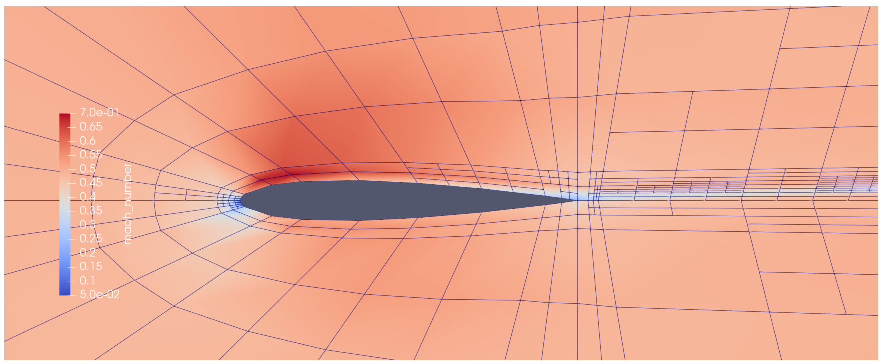
4.1 Subsonic Test 1

The first subsonic test case was steady inviscid flow over a NACA-0012 airfoil. The NACA-0012 airfoil had an angle of attack of 2.5 degrees and the mach number of the flow was $M = 0.5$. Three adaptations were run on the case using a fixed fraction of $f = 0.15$. This was done for lift adjoint, drag adjoint, entropy adjoint, and feature based adaptation. The results of these adaptations are seen in figure 4.1.

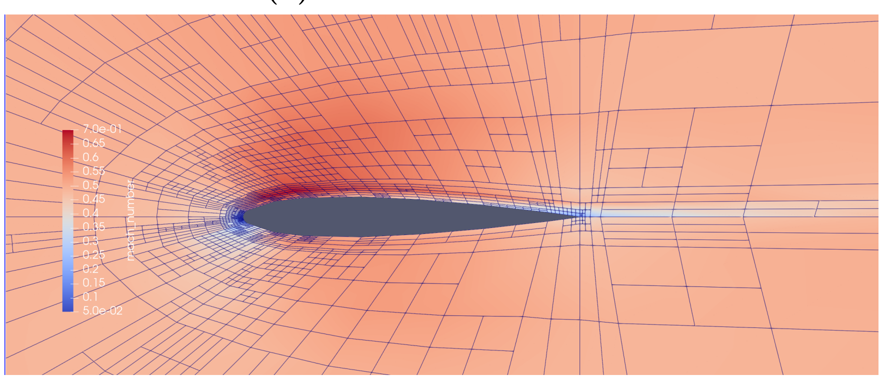
From the first test it can be seen the initial grid is very coarse. After undergoing feature-based adaptation there were very few cells around the NACA-0012 airfoil that adapted. The majority of adaptations for it appeared in the wake or the farfield of the mesh. This shows how the feature-based mesh adaptation does a poor job at adapting the mesh in the location of interest, near the NACA-0012 airfoil. This trend continued to be seen in the rest of the test cases and shows how the feature-based mesh adaptation doesn't work well for capturing flow outputs and features of interest. Following this, it can be seen that in the lift and drag oriented mesh adaptation a significant number of cell adaptations were done near the stagnation points on the leading and trailing edge of the airfoil along with their associated streamlines. This is as expected as these are the locations primarily responsible for the calculation of the lift and drag of the airfoil as well as where a significant amount of error was introduced into the numerical solution. Finally, the entropy adjoint mesh adaptation shows similar results to the drag and lift oriented mesh adaptation with the majority of mesh adaptation occuring at the leading and trailing edges of the airfoil. This indicates that the entropy-adjoint indicator is able to detect spurious entropy generation in the numerical solution at these locations.



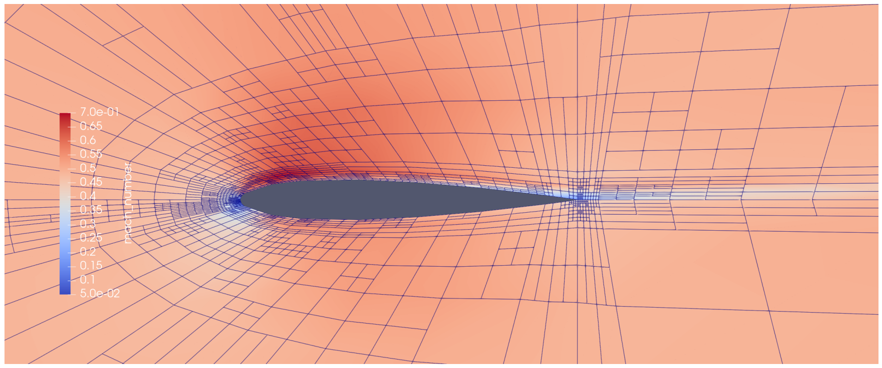
(a) Baseline Solution



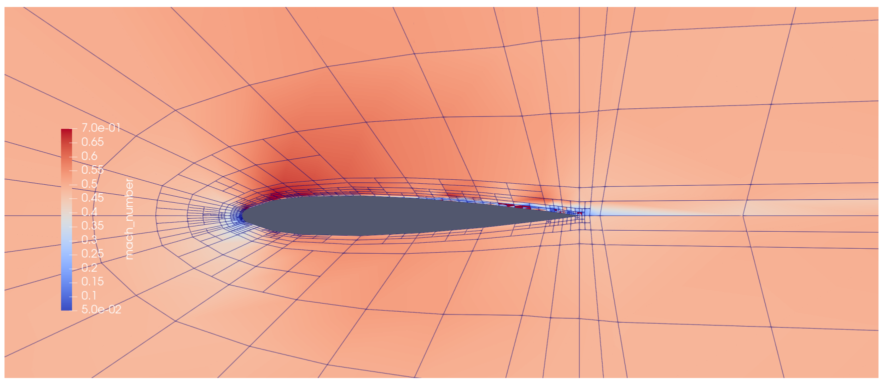
(b) Feature Based Solution



(c) Drag Adjoint Solution



(d) Lift Adjoint Solution



(e) Entropy Adjoint Solution

Figure 4.1: Subsonic Test 1, $M = 0.5$, $\alpha = 2.5$, $f = 0.15$

Analyzing the lift coefficients of the test case shows that all the different adjoints converge towards a similar value. Figure 4.2 shows how the lift adjoint seems to converge the fastest, followed by the drag adjoint beginning to converge to the same value. After this, the entropy adjoint is shown to begin converging towards a similar value. Finally, the feature-based adaptation is the furthest away from the lift-adaptation value. This is as expected as the lift indicator is tailored towards converging the coefficient of lift value the fastest, and the drag indicator will perform similarly as it will be mainly adapting elements along the boundary layer of the airfoil responsible for the drag calculation. The feature-based will also take a very long time to converge towards the coefficient of lift value as the majority of the mesh adaptation is seen in the farfield and not near the airfoil. In general, figure 4.2 doesn't show the best convergence and this is likely due to the fact that the solution is approximated with $p = 1$ polynomials and because there were not very many mesh adaptations performed. If more adaptations were done for higher order solutions it would likely show better convergence towards the coefficient of lift and it would give a better depiction of the performance of the different types of adaptation. This is discussed further in the future work section of this thesis.

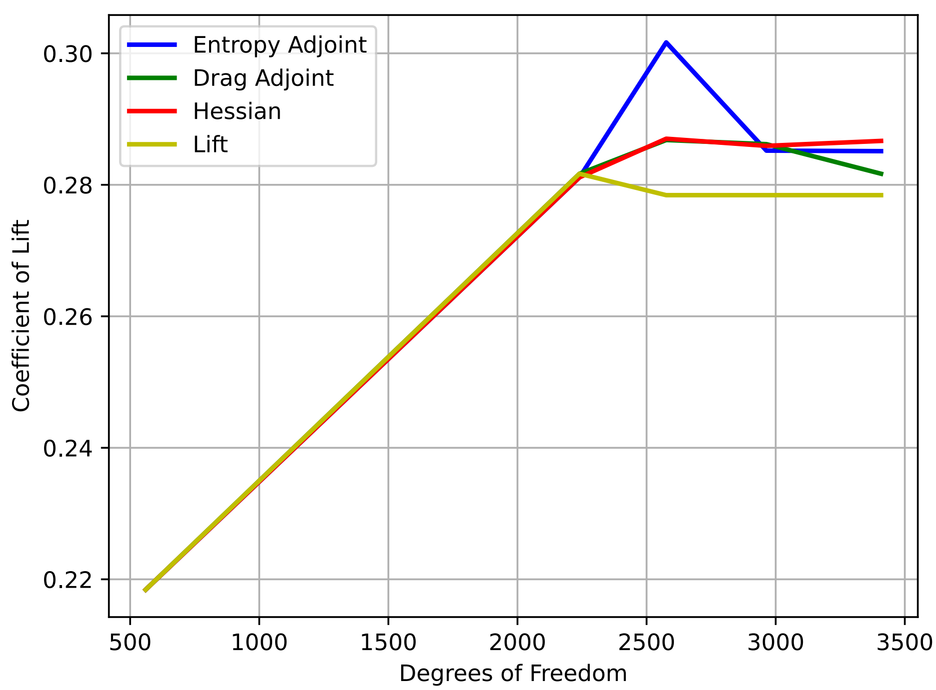
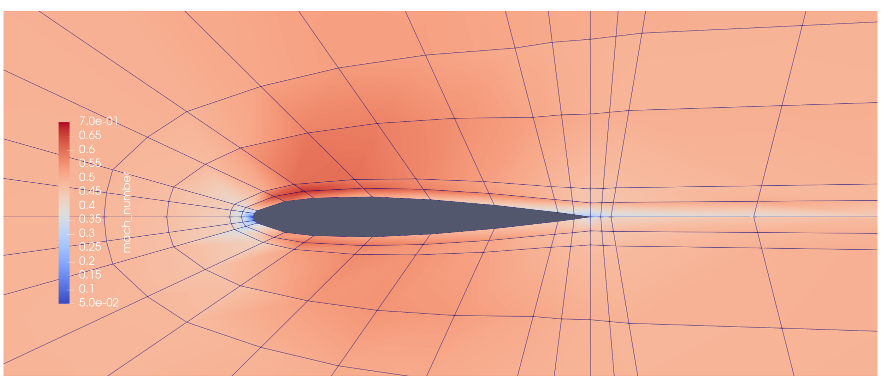


Figure 4.2: Subsonic Test 1 Coefficients of Lift during Adaptation

4.2 Subsonic Test 2

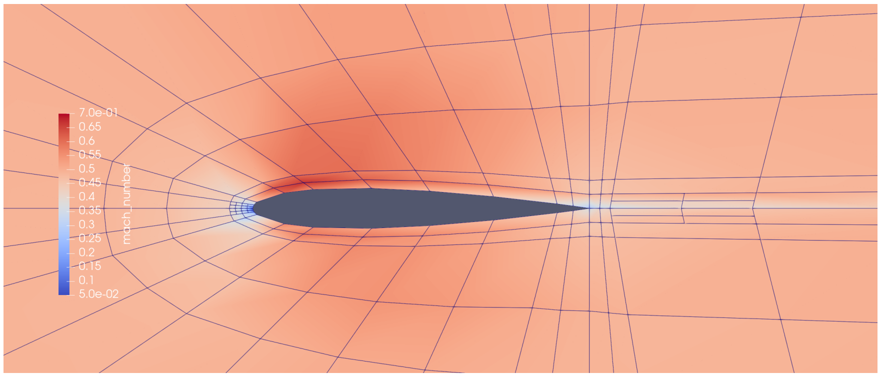
The second subsonic test case was steady inviscid flow over a NACA-0012 airfoil. The NACA-0012 airfoil had an angle of attack of 1.25 degrees and the mach number of the flow was $M = 0.5$. Three adaptations were run on the case using a fixed fraction of $f = 0.10$. This was done for lift adjoint, drag adjoint, entropy adjoint, and feature based adaptation. The results of these adaptations are seen in figure 4.3.

The mesh adaptations for subsonic test 2 yielded very similar results to the mesh adaptation for subsonic test 1. This is as a expected as very little changed with the exception of the angle of attack and the fixed fraction. Again, figure 4.3 shows how the feature-based adaptation performed very poorly at adapting the mesh in the areas of interest around the airfoil. The lift, drag, and entropy adjoint mesh adaptations all had very similar mesh adaptations, pinpointing the stagnation points as the location of interest for their indicators.

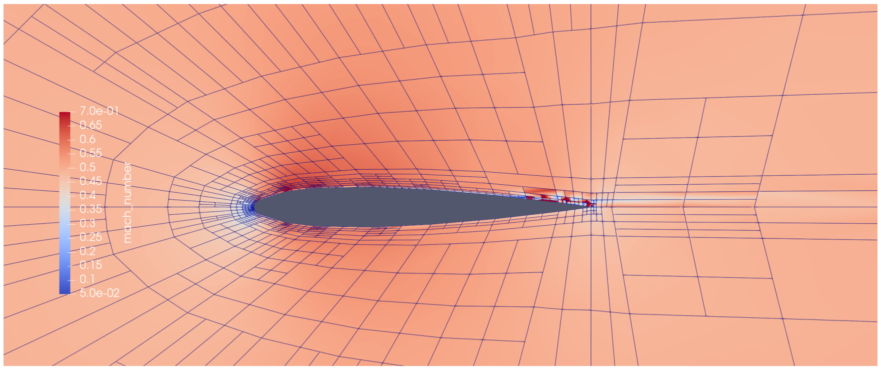


(a) Baseline Solution

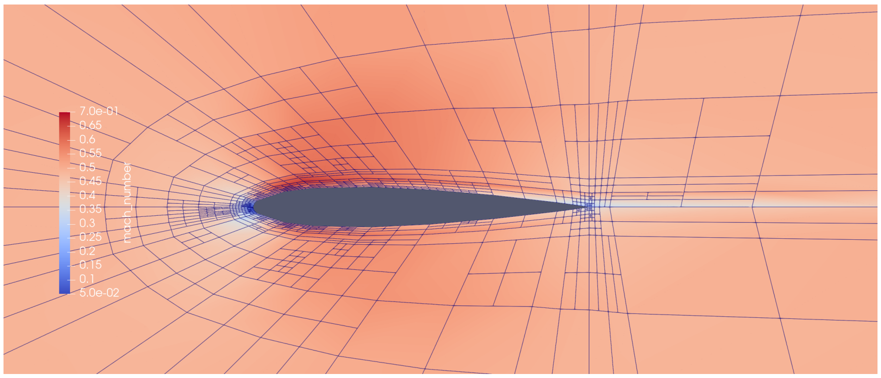
Figure 4.3: Subsonic Test 2, $M = 0.5$, $\alpha = 1.25$, $f = 0.10$



(b) Feature Based Solution

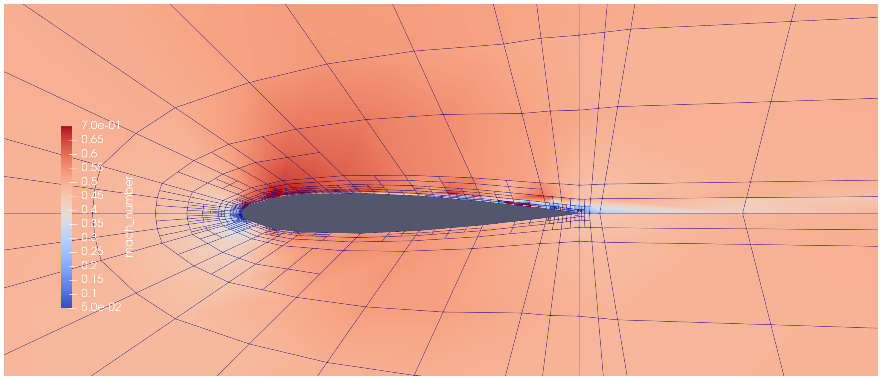


(c) Drag Adjoint Solution



(d) Lift Adjoint Solution

Figure 4.3: Subsonic Test 2, $M = 0.5$, $\alpha = 1.25$, $f = 0.10$



(e) Entropy Adjoint Solution

Figure 4.3: Subsonic Test 2, $M = 0.5$, $\alpha = 1.25$, $f = 0.10$

As expected, the lift coefficients also performed very similar to subsonic test 1, with the lift adjoint mesh adaptation converging the fastest. After this the drag adjoint appears to converge to a similar value. Following this the entropy adjoint and then the feature-based mesh adaptation slowly converge. Again, the reasoning for the lack of convergence by all adaptation types to the same value is likely due to the fact that the solution is approximated with low order polynomials and not many mesh adaptations were run.

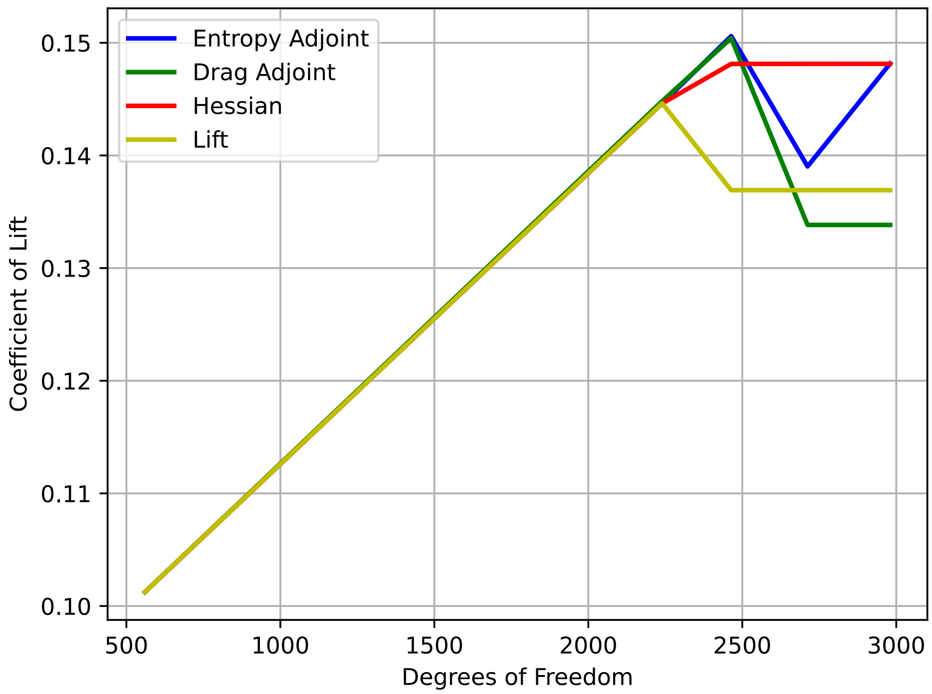
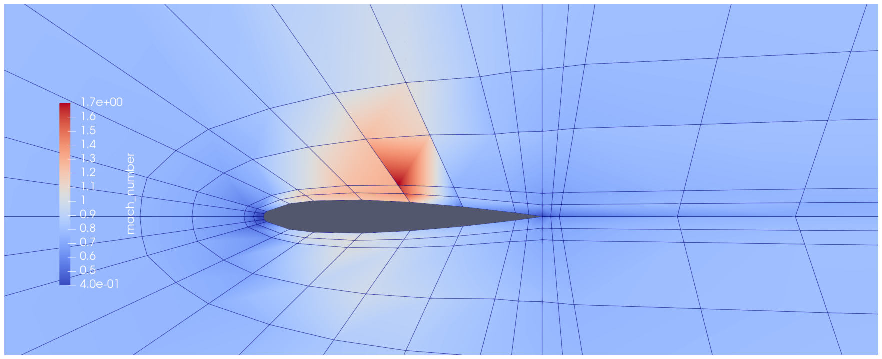


Figure 4.4: Subsonic Test 2 Coefficients of Lift during Adaptation

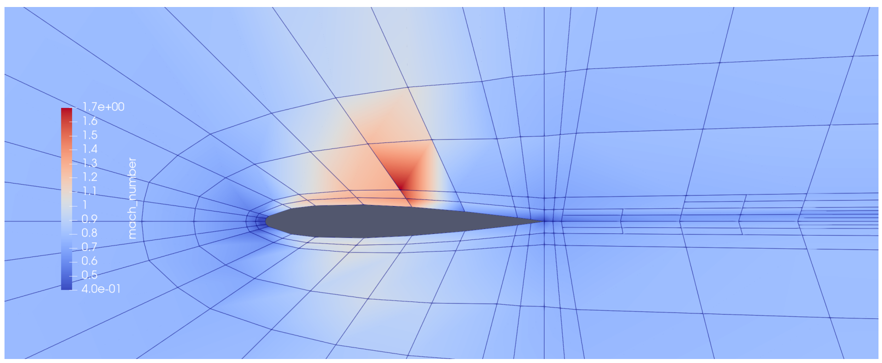
4.3 Transonic Test

The transonic test case was steady inviscid flow over a NACA-0012 airfoil. The NACA-0012 airfoil had an angle of attack of 1.25 degrees and the mach number of the flow was $M = 0.8$. Three adaptations were run on the case using a fixed fraction of $f = 0.25$. This was done for lift adjoint, drag adjoint, entropy adjoint, and feature based adaptation. The results of these adaptations are seen in figure 4.5.

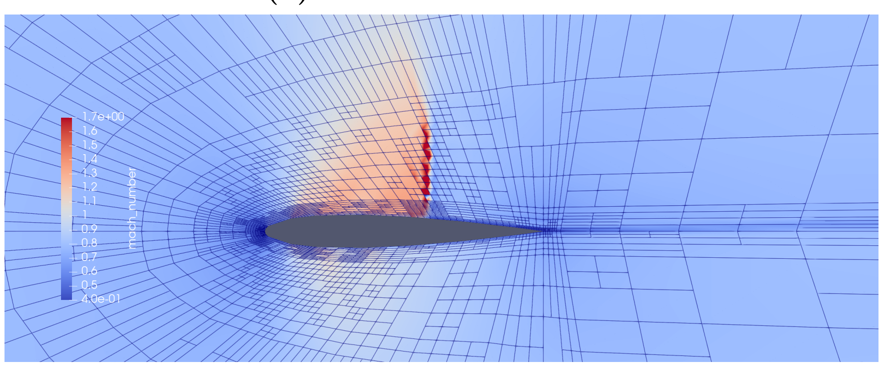
As this test case was transonic, it means that a shock will appear in the solution along the airfoil. In figure 4.5 this can be seen. From the original coarse solution, the shock is not well captured and the mesh did a poor job at capturing flow features of interest around the airfoil. The same occurred in the feature-based mesh adaptation, where cells around the airfoil didn't refine, only cells in the wake and farfield. This continues to show that feature-based mesh adaptation does a poor job at refining the mesh in areas of interest. Following this, the drag and lift oriented mesh adaptations are shown. It can be seen that there was significant mesh adaptation at the stagnation points. They also adapted the mesh around the location of the shock as it is close to the boundary layer and cells that affect the lift and drag computation. In general, these two performed very well at capturing the flow features of interest. Finally, the entropy adjoint mesh adaptation can be analyzed. As in the case of the two subsonic tests, there was significant mesh adaptation at the stagnation points on the leading and trailing edge as this is where spurious entropy generation occurs in numerical simulations over airfoils. There was also significant mesh adaptation along the shock, better capturing its effects in the solution. This is due to the fact that the shock generates entropy and so it is captured by the entropy adjoint indicator.



(a) Baseline Solution

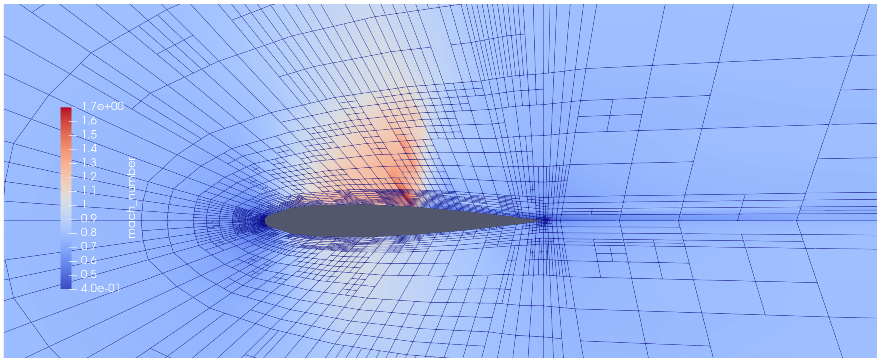


(b) Feature Based Solution

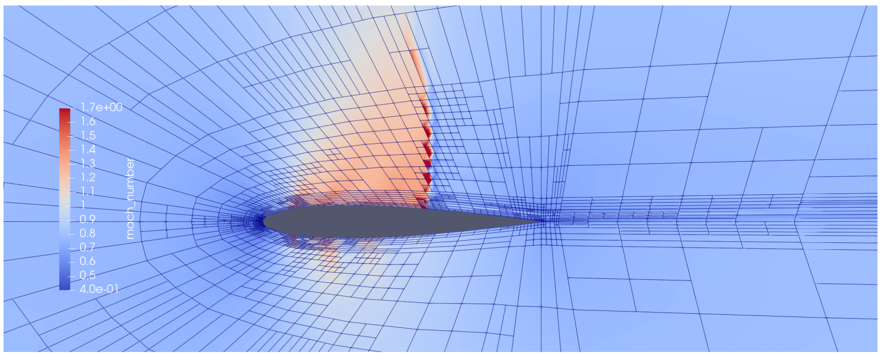


(c) Drag Adjoint Solution

Figure 4.5: Transonic Test, $M = 0.8$, $\alpha = 1.25$, $f = 0.25$



(d) Lift Adjoint Solution



(e) Entropy Adjoint Solution

Figure 4.5: Transonic Test, $M = 0.8$, $\alpha = 1.25$, $f = 0.25$

Again, figure 4.6 shows the convergence of the coefficient of lifts can be seen to have a similar trend to the previous two test cases. The lift indicator converged fastest, followed by the drag indicator. Finally, the entropy adjoint and feature-based mesh adaptation look to need more iterations for convergence to the true coefficient of lift. This is as expected due to the low order polynomials and the low number of mesh adaptations run on the original grid. With an increase of adaptations on a higher order grid, it likely would have better depicted the difference in performance of the indicators.

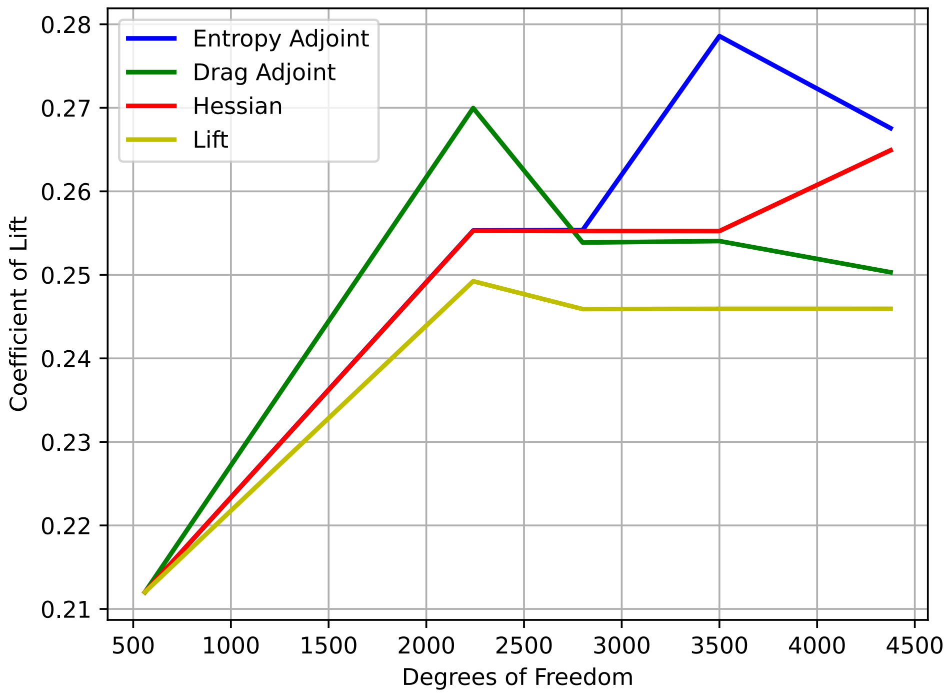


Figure 4.6: Transonic Test Coefficients of Lift during Adaptation

4.4 Summary

In summary, the entropy adjoint mesh adaptation succesfully identified locations in the mesh of spurious entropy generation, along with the shock that generated entropy. This shows its utility as an indicator for doing mesh adaptation in areas of interest in airfoil problems. It also has the advantage of not needing to solve a linear system of equations to determine the adjoint solution, reducing its computational costs compared to other goal-oriented indicators. Finally, figure 4.7 shows how as the degrees of freedom increased during adaptations, the Dual-Weighted Residual decreases for the entropy indicator. This indicates that the numerical solution has grid independence.

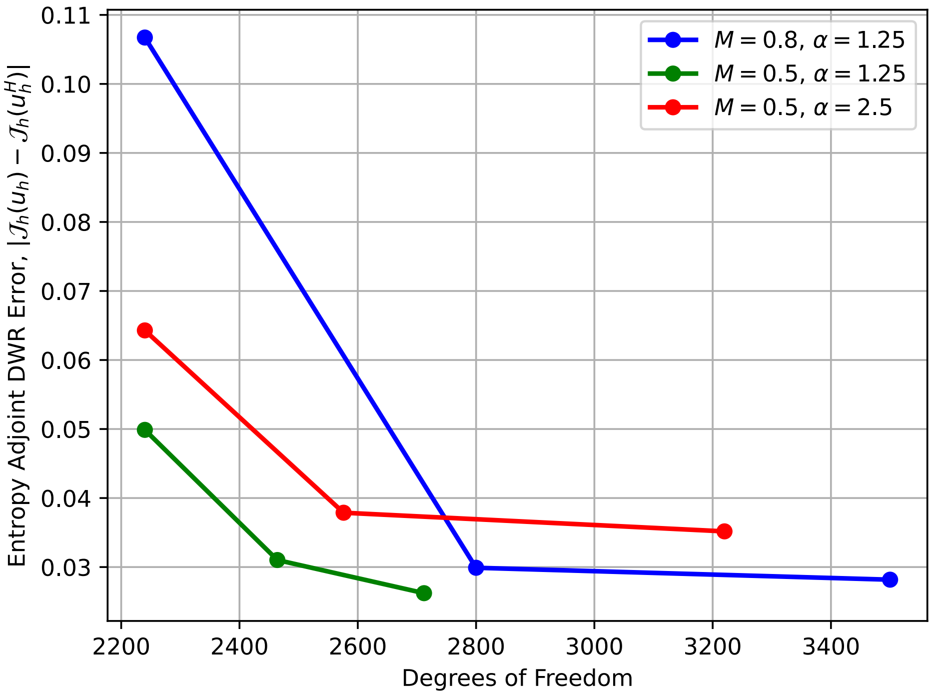


Figure 4.7: Fixed Fraction Mesh Refinement for $f = 0.75$.

Chapter 5

Conclusion and Future Work

This thesis implemented and investigated the usage of the entropy adjoint equations for mesh adaptation. Using the Discontinuous Galerkin method as a numerical solver and the Dual-Weighted Residual to compute the cell-wsie error, it was shown with three different two-dimensional test cases that the entropy adjoint functional can be used to identify cells of spurious entropy generation. It also successfully identified locations in the inviscid flow that generated entropy such as shock waves. Additionally, it was also shown that there are reduced computational costs for the entropy indicator compared to other goal-oriented indicators as a linear system of equations doesn't need to be solved to compute the adjoint solution. The two subsonic and transonic test case also underwent mesh adaptation for lift-oriented, drag-oriented and feature-based mesh adaptation for comparison. From this it was shown that feature-based mesh adaptation performed very poorly at identifying cells to be adapted that were in areas of interest of the airfoil compared to the goal-oriented indicators. Aditionally, the coefficient of lift convergence was tracked for the different adaptation types to compare their performances. From this, it was shown that the lift indicator converged the fastest followed by the other two goal-oriented indicators. Finally, the feature-based mesh adaptation didn't show signs of convergence towards the coefficient of lift.

Future work to improve the analysis of the indicator performance needs to be done. Running test cases with higher order polynomial approximations for the cell-wise solutions along with more adaptations would show more clearly the improved performance of the goal-oriented indicators and how they compare to eachother. The entropy adjoint mesh adaptation technique could also be extended to three-dimensional test cases to adapt the mesh to capture other flow characteristics of interest such as wingtip vortices [26]. Other possible future work includes adapting the entropy adjoint functional for the Navier-Stokes equations. This is to account for the introduction of viscosity into the flow meaning that entropy will be generated in the solution, changing the formulation of the entropy transport flux.

Bibliography

[1] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods*. Springer, New York, NY, 2010.

[2] Timothy J. Baker. Mesh adaptation strategies for problems in fluid dynamics. *Finite Elements in Analysis and Design*, 25(3):243–273, 1997. Adaptive Meshing, Part 2.

[3] Ülgen Gülçat. Transonic Flow. In *Fundamentals of Modern Unsteady Aerodynamics*, pages 181–204. Springer Singapore, Singapore, 2016.

[4] NASA. Air-to-air background oriented schlieren shock. https://www.nasa.gov/image-detail/f4_p3_rgb_planedrop, 2019. Accessed: 2023-11-20.

[5] Ziniu Wu, Yizhe Xu, Wenbin Wang, and Ruifeng Hu. Review of shock wave detection method in CFD post-processing. *Chinese Journal of Aeronautics*, 26(3):501–513, 2013.

[6] Zi-Fei Meng, A.-Man Zhang, Ping-Ping Wang, and Fu-Ren Ming. A shock-capturing scheme with a novel limiter for compressible flows solved by smoothed particle

hydrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 386:114082, 2021.

[7] Garrett Barter and David Darmofal. Shock Capturing with Higher-Order, PDE-Based Artificial Viscosity. In *18th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2007. _eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2007-3823.

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

[9] Sandip Mazumder. Chapter 1 - Introduction to Numerical Methods for Solving Differential Equations. In Sandip Mazumder, editor, *Numerical Methods for Partial Differential Equations*, pages 1–49. Academic Press, January 2016.

[10] W H Reed and T R Hill. Triangular mesh methods for the neutron transport equation. 10 1973.

[11] H. T. Huynh. Discontinuous Galerkin and Related Methods for ODE. *Journal of Scientific Computing*, 96(2):51, August 2023.

[12] Nickolay Y. Gnedin, Vadim A. Semenov, and Andrey V. Kravtsov. Enforcing the Courant–Friedrichs–Lewy condition in explicitly conservative local time stepping schemes. *Journal of Computational Physics*, 359:93–105, April 2018.

[13] F. Angrand and Institut National de Recherces en Informatique et Automatique Workshop. *Numerical Methods for the Euler Equations of Fluid Dynamics*. SIAM, January 1985. Google-Books-ID: RzABBbx4obMC.

[14] F. Alauzet and L. Frazza. Feature-based and goal-oriented anisotropic mesh adaptation for RANS applications in aeronautics and aerospace. *Journal of Computational Physics*, 439:110340, August 2021.

[15] Giovanni Todarello, Floris Vonck, Sébastien Bourasseau, Jacques Peter, and Jean-Antoine Désidéri. Finite-volume goal-oriented mesh adaptation for aerodynamics using functional derivative with respect to nodal coordinates. *Journal of Computational Physics*, 313, 03 2016.

[16] Carlos Lozano. Drag and Entropy Variables. July 2019.

[17] Vinh-Tan Nguyen, Hoang Huy Nguyen, Matthew Antony Price, and Jaik Kwang Tan. Shock capturing schemes with local mesh adaptation for high speed compressible flows on three dimensional unstructured grids. *Computers & Fluids*, 70:126–135, November 2012.

[18] Guglielmo Vivarelli, Ning Qin, Shahrokh Shahpar, and David Radford. Anisotropic adjoint sensitivity-based mesh movement for industrial applications. *Computers & Fluids*, 221:104929, May 2021.

[19] Robert T. Balmer. Chapter 7 - Second Law of Thermodynamics and Entropy Transport and Production Mechanisms. In Robert T. Balmer, editor, *Modern Engineering Thermodynamics*, pages 205–247. Academic Press, Boston, January 2011.

[20] Thomas Richter and Thomas Wick. Variational localizations of the dual weighted residual estimator. *Journal of Computational and Applied Mathematics*, 279:192–208, May 2015.

[21] Keigan MacLean. Unstructured anisotropic all-quad mesh adaptation based on a continuous local error model for the discontinuous Galerkin method. Publisher: McGill University.

[22] Christopher Roy. Strategies for Driving Mesh Adaptation in CFD (Invited). In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 2009. _eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2009-1302.

[23] Jing Tang, Pengcheng Cui, Bin Li, Yaobin Zhang, and Hang Si. Parallel hybrid mesh adaptation by refinement and coarsening. *Graphical Models*, 111:101084, September 2020.

[24] Keigan MacLean and Sivakumaran Nadarajah. Unstructured Anisotropic Mesh Adaptation for Quads Based on a Local Error Model. In *AIAA Scitech 2021*

Forum. American Institute of Aeronautics and Astronautics, 2021. _eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2021-1840.

[25] Alejandro Diaz-Morcillo, Luis Nuño, Juan Balbastre, and David Sánchez-Hernández. Adaptive mesh refinement in electromagnetic problems. pages 147–155, 01 2000.

[26] Sheldon I. Green. Wing Tip Vortices. In Sheldon I. Green, editor, *Fluid Vortices*, Fluid Mechanics and Its Applications, pages 427–469. Springer Netherlands, Dordrecht, 1995.

McGill

NON-EXCLUSIVE LICENSE

AUTHOR'S NAME: Bradley | Cole | CB
LAST NAME | ***FIRST NAME*** | ***INITIALS***

AUTHOR'S DATE OF BIRTH: 05 / 21 / 2001 /
MONTH / DAY / YEAR

MCGILL EMAIL ADDRESS: cole.bradley@mail.mcgill.ca **MCGILL ID NUMBER:** 260891176

PERMANENT ADDRESS: 1035 Varsity Estates Pl NW Canada
COUNTRY

MCGILL UNIT: Department of Mechanical Engineering **FACULTY:** Faculty of Engineering

TITLE OF WORK: Entropy Adjoint Mesh Adaptation for the Discontinuous Galerkin Approach

I hereby promise that I am author of the above cited Work.

I confirm that it is my original **Work**, does not infringe any rights of others, and that I have the right to make the grant conferred by this non-exclusive license. I also confirm that if third party copyrighted material was included in my **Work** for which, under the terms of the Copyright Act, written permission from the copyright owner(s) is required, I have obtained such permission from the copyright owner(s) and may grant such permission for the full term of copyright protection.

I hereby grant to McGill University a non-exclusive, worldwide, irrevocable, royalty free license, in respect of my **Work**, to reproduce, convert, archive, preserve, conserve, communicate, distribute and loan, in paper form, in microform, electronically by telecommunication or on the internet, and/or any other formats as may be adopted for such use from time to time.

I hereby grant permission and authorize McGill University to permit access to my **Work** and make it available to interested persons in paper or electronic form, through the library, interlibrary and public loan.

I understand that I retain copyright ownership and moral rights in my **Work**, and that I may deal with the copyright in my **Work** consistent with these rights. I promise to inform any person to whom I may hereafter assign or license my copyright in my Work of the rights granted by me to McGill University**.**

The authorization contained in this non-exclusive license is to have effect on the date given below (**Effective Date**)**.**

I agree to indemnify and hold McGill University harmless against any claim and any loss, damage, settlement cost or expense (including legal fees) incurred by McGill University and arising out of, or in connection with, my statements and representations or this license.

The parties hereto expressly request that this agreement and all related notices and documents be drawn up in English.
Les parties aux présentes conviennent expressément que cette convention ainsi que tous les avis et documents s'y rattachant soient rédigés en anglais.

SIGNATURE OF AUTHOR: ColeB **EFFECTIVE DATE:** 30/11/2023