Impact of Triangle Shapes Using High-Order Discretizations and Direct Mesh Adaptation for Output Error

by

Huafei Sun

B.A.Sc., Engineering Science, Aerospace Option (2007)
University of Toronto

Submitted to the School of Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Computation for Design and Optimization

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2009

Author ....
School of Engineering
September, 2009

Certified by ....
David L. Darmofal
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ....
Jaime Peraire
Professor of Aeronautics and Astronautics
Director, Program in Computation for Design and Optimization

Impact of Triangle Shapes Using High-Order Discretizations and Direct Mesh Adaptation for Output Error

by
Huafei Sun

Submitted to the School of Engineering
on September, 2009, in partial fulfillment of the
requirements for the degree of
Master of Science in Computation for Design and Optimization

Abstract

The impact of triangle shapes, including angle sizes and aspect ratios, on accuracy and stiffness is investigated for simulations of highly anisotropic problems. The results indicate that for high-order discretizations, large angles do not have an adverse impact on solution accuracy. However, a correct aspect ratio is critical for accuracy for both linear and high-order discretizations. In addition, large angles are not problematic for the conditioning of the linear systems arising from discretization. They can be overcome through small increases in preconditioning costs.

A direct adaptation scheme that controls the output error via mesh operations and mesh smoothing is also developed. The decision of mesh operations is solely based on output error distribution without any *a priori* assumption on error convergence rate. Anisotropy is introduced by evaluating the error changes due to potential edge split, and thus the anisotropies of both primal and dual solutions are taken into account. This scheme is demonstrated to produce grids with fewer degrees of freedom for a specified error level than the existing metric-based approach.

Thesis Supervisor: David L. Darmofal
Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

I would like to express my gratitude to the many people who have made this thesis possible.

First and foremost, I would like to thank my advisor, Prof. David Darmofal, for giving me the opportunity to work with him on this interesting yet challenging research project. I am grateful for his insights, guidance, and inspiration throughout this research, and look forward to continuing our work together.

I would also like to thank the Project X team (Julie Andren, Garret Barter, Laslo Diosady, Krzysztof Fidkowski, Bob Haimes, Josh Krakos, Eric Liu, JM Modisette, Todd Oliver, and Masa Yano). This work would not have been possible without the numerous insightful discussions with them and their tireless effort involved in the Project X codes. Special thanks go to Laslo Diosady, Xun Huan, JM Modisette, and Masa Yano for their patient help during the drafting of this thesis. Many discussions with Bob Haimes have also proven very useful. I would also like to thank everyone at ACDL for making the last two years a lot of fun.

Last but not least, I would like to thank my parents for their constant love, support, and encouragement, without which I am sure I would not have gotten this far.

This work was partially supported by funding from The Boeing Company with technical monitor of Dr. Mori Mani.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Over the last several decades, Computational Fluid Dynamics (CFD) technology has become an indispensable component for the analysis and design of aerospace vehicles. Due to both algorithm development and increasing computational power, the complexity of problems - either from geometry, physics, or both - that can be simulated has increased dramatically. Despite the wide use of CFD, lack of automation still hinders the analysis process. In particular, the process of mesh generation still requires heavy "person-in-the-loop" involvement and is often the bottleneck in CFD applications.

The results from the AIAA Drag Prediction Workshops illustrate the difficulties in generating grid converged results even by experts utilizing very fine meshes that are carefully hand-crafted [50]. More specifically, Mavriplis demonstrates that even for very fine meshes, asymptotic results appear different for different families of self-similar grids [50]. He also concludes that fully resolving all features in computational aerodynamics problems is infeasible via successive global refinements of an arbitrary initial mesh. As such, grid adaptation techniques have significant promise to lead to grid converged results and to reduce the amount of manual intervention involved in the grid generation process. Additionally, as high-order methods have significant potential in reducing the time to achieve a specified accuracy level, adaptation schemes should be made compatible with high-order discretizations.

Many physical problems exhibit highly anisotropic behaviors, where the length scale in one direction can be several orders of magnitude smaller than in the other. To resolve these features, which often occur in thin boundary layers, wakes, or shocks, highly anisotropic elements need to be employed to improve computational efficiency. In turbulence applications, aspect ratios on the order of 100,000 are often appropriate. Therefore, generating highly anisotropic meshes during adaptation becomes a key step in the efficient simulation of such applications.

While an equilateral element is ideal for an isotropic problem, one important question regarding mesh generation is how element shapes affect solution accuracy in an anisotropic context. This question is addressed in this thesis for high-order simulation of highly anisotropic problems. Furthermore, for engineering applications, accuracy in an integral output is often of more interest than the overall solution. In aerodynamic applications, for example, such outputs include the lift, drag, and moment coefficients on an aircraft. This thesis presents an adaptation scheme that has a direct control over the error of a specified output.

1.2 Background

1.2.1 Impact of Triangle Shapes

Understanding the impact of triangle shapes provides an important guideline for mesh generation and adaptation. While high aspect ratio meshes are important for efficiency to resolve anisotropic features, triangular elements with large angles are known to deteriorate solution quality. Babuška and Aziz demonstrate that the accuracy of a finite element approximation on triangular meshes degrades as the maximum angle of elements approaches 180° asymptotically [1]. Although little can be said about elements of intermediate quality, avoiding large angles leads to the development of several mesh generation algorithms, including minimum-maximum triangulations [3, 24], and hybrid methods with structured meshes in the regions where high anisotropy is needed [43, 48]. Both of these methods have different robustness problems, especially for complex geometries, and avoiding large angles in mesh generation still remains a difficult problem.

Shewchuk conducts a comprehensive study on triangle shapes for linear finite element [72]. He concludes that the triangle shape that leads to the best accuracy in terms of the H^1 error is one with an aspect ratio equal to the ratio κ of the maximum and minimum singular values of the solution Hessian but without large angles. However, he also demonstrates that this so-called "superaccuracy" is very fragile as the elements have to align *precisely* with the singular vectors of the Hessian, and so he suggests that generating triangles with an intermediate aspect ratio of $\sqrt{\kappa}$ but possibly with large angles is a more realistic goal for anisotropic mesh generation. Also, Rippa shows in his work that this aspect ratio is optimal for the L^p-norm of the interpolation error for $1 \leq p < \infty$ [65]. Although these work provide good insights into what an optimal triangle shape should be, their analysis are limited to linear elements, and the impact of large angles and aspect ratios has not been studied for high-order discretizations of highly anisotropic problems. If high-order discretizations are not sensitive to element angles, then large angles introduced by a Delaunay triangulation in a stretched space [9, 19] would not be problematic.

In this thesis, the impact of large angles and element aspect ratios will be studied for boundary layer problems using high-order discretizations. Both solution accuracy and conditioning of the discretized system will be investigated when large angles are present.

1.2.2 Mesh Adaptation

Given an error indicator, the goal of adaptation is to decrease the error by modifying the solution space. For finite element methods, this can be done in one of three ways: p-adaptation, where the order of interpolation is changed on a fixed mesh; h-adaptation, where the mesh is modified but the interpolation order is constant; or hp-adaptation, where p- and h-adaptations are combined. While p-adaptation works efficiently for solutions with sufficient regularity, h-adaptation allows for the generation of anisotropic elements, which are necessary for efficiently simulating anisotropic problems. hp-adaptation strives to combine the best of the two strategies, but making the decision between p- and h-adaptation requires an estimate of solution regularity, which is often difficult. Houston and Süli present a review of common methods for making this decision [40], and Georgoulis *et al.* present an hp-scheme for anisotropic adaptation [32]. In this work, the adaptation strategy is h-adaptation at a constant p.

Two popular approaches can be found in literature for h-adaptation. The first is metric-based adaptation, where a field of metric tensor is computed from an *a priori* or *a posteriori* error estimate and the adapted mesh is generated according to the requested metric field (e.g. [26, 74, 75]). The second one is direct adaptation, which directly modifies an existing mesh based on an error estimate, without computing any metric request (e.g. [31, 52, 61]).

In the metric-based adaptation, a field of metric tensor is first computed, which contains the information of size and orientation of an element, for example, see [37, 75]. One common method for generating the size request is based on the principle of error equidistribution [58, 77]. Together with an *a priori* assumption, this principle has been applied in output-based adaptation (e.g. [74]). Anisotropy detection is often achieved through estimating the derivatives of a selected scalar field. For linear interpolation, the solution Hessian is computed [62]. This idea is introduced into output-based adaptation by Venditti and Darmofal [75], and Formaggia *et al.* combine this idea with output-based error analysis and derive an output-based anisotropic error estimate [29]. Fidkowski extends this idea for high-order interpolations [26], and Pagnutti and Ollivier-Gooch provide a similar method based on estimating truncation errors [60]. The common scalar field used is Mach number for aerodynamic simulations [26, 37, 74]. Also, Castro-Díaz *et al.* propose to take into account the metric ellipse of every variable of the PDE system of interest [13].

After the metric tensor field is obtained in the metric-based adaptation, a new mesh is generated according to this field. For isotropic meshing, the standard Delaunay triangulation is an easy and prevalent method. Shewchuk provides a comprehensive review on this subject [71]. For anisotropic meshing from a given metric field, one popular approach is the anisotropic Delaunay triangulation, which constructs a Delaunay triangulation in the stretched space of the metric tensor [9, 59]. Another approach is the metric-driven optimization, which applies mesh operations to achieve best metric conformity [11, 44]. The mesh generation can also be done by a pliant method, which adjusts node positions according to attraction/repulsion with neighbors [10, 64].

For the direct adaptation, mesh operations are directly applied to the existing mesh according to the error distribution. No intermediate size-request is computed. This approach favors both the primal and the dual solutions based on their impact on the error. Georgoulis *et al.* employ this method with anisotropic refinement on Cartesian meshes [31, 32]. The anisotropy is introduced based on local solves on the new configuration due to a potential split. Leicht and Hartmann apply a similar method except that the anisotropy is determined based on the inter-element jumps in the discretized solution from a discontinuous Galerkin method [45]. In Park's work, all possible mesh operations are applied in a greedy manner to decrease output error [61], and this method has shown superiority over the metric-based approach in terms of error per degree of freedom (DOF).

In this thesis, a direct adaptation scheme that controls the output error is developed. Delaunay-type mesh smoothing techniques are also applied. Mesh smoothing is often designed to improve a certain mesh quality, either in terms of geometric properties or some error measures. The simplest approach is Laplacian smoothing [28], and more sophisticated approaches involve optimization problems with basic mesh operations, for example, see [30]. These optimization schemes can be Delaunay-type, involving equidistribution of edge length under a desired metric field [14].

1.3 Outline of Thesis

This thesis is organized as follows. The discretization strategy and solution method are presented in Chapter 2. Chapter 3 studies the impact of element shapes on solution accuracy and conditioning of the linear system arising from the discretization. In Chapter 4, the output error estimation using a dual-weighted-residual method and its implementation are described. A direct adaptation scheme to control the output error is developed in Chapter 5, and numerical results are shown.

Chapter 2

Discretization and Solution Method

This chapter describes the governing equations considered in this work, and presents a high-order discontinuous Galerkin finite element discretization for a general system of conservation laws. The linear solution method for the induced discretized system is also discussed.

2.1 Governing Equations

Let $\Omega \in \mathbb{R}^d$ be an open, bounded domain in a $d-$dimensional space. A general time-dependent conservation law in the domain, Ω, expressed in the strong form is given by

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathcal{F}^i(\mathbf{u}) - \nabla \cdot \mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u}) = \mathcal{S}(\mathbf{x}, t), \text{ in } \Omega, \tag{2.1}$$

where $\mathbf{u}(\mathbf{x}, t) : \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}^m$ is the $m-$state solution vector. The inviscid flux $\mathcal{F}^i(\mathbf{u}) : \mathbb{R}^m \rightarrow \mathbb{R}^{m\times d}$, the viscous flux $\mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u}) : \mathbb{R}^m \times \mathbb{R}^{m\times d} \rightarrow \mathbb{R}^{m\times d}$, and the source term $S(\mathbf{x}, t) : \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}^m$ characterize the governing equations to be solved. In this work, the governing equations considered include the advection-diffusion equation and the Navier-Stokes equations, both of which have the form of the conservation law given in Eqn. (2.1).

The advection-diffusion equation has its inviscid and viscous fluxes defined as

$$\mathcal{F}^i \equiv \vec{\beta} u, \qquad \mathcal{F}^v \equiv \bar{\bar{\mu}} \cdot \nabla u, \tag{2.2}$$

where $\vec{\beta}$ is the advection velocity and $\bar{\bar{\mu}}$ is the diffusivity tensor. In this work, $\bar{\bar{\mu}}$ is isotropic, and its notation is replaced by a scalar μ unless otherwise stated.

For the compressible Navier-Stokes equations, the conservative state vector is $\mathbf{u} = [\rho, \rho\vec{v}, \rho E]^T$, where ρ is the density, $\vec{v}$ is the velocity, and E is the specific total internal energy. The inviscid and viscous fluxes are given by

$$\mathcal{F}^i \equiv \begin{bmatrix} \rho\vec{v} \\ \rho\vec{v} \otimes \vec{v} + p\bar{\bar{I}} \\ \rho\vec{v}H \end{bmatrix}, \qquad \mathcal{F}^v \equiv \begin{bmatrix} 0 \\ \bar{\bar{\tau}} \\ \bar{\bar{\tau}} \cdot \vec{v} + \kappa_T \nabla T \end{bmatrix}, \tag{2.3}$$

where p is the static pressure, $H = E + p/\rho$ is the specific total enthalpy, T is the temperature calculated from the ideal gas law, κ_T is the thermal conductivity, and τ is the shear stress, for which a Newtonian fluid is assumed. The pressure is related to the state vector by

$$p = (\gamma - 1)\rho \left(E - \frac{1}{2} \vec{v} \cdot \vec{v} \right),$$

where γ is the ratio of specific heats.

2.2 Discontinuous Galerkin Discretization

Let $\mathcal{T}_H$ be a triangulation of the domain Ω with non-overlapping elements, κ, such that $\bar{\Omega} = \bigcup_{\kappa \in \mathcal{T}_H} \bar{\kappa}$. Also, define a function space $\mathcal{V}_H^p$ by

$$\mathcal{V}_H^p \equiv \{\mathbf{v} \in \left(L^2(\Omega)\right)^m : \mathbf{v}|_\kappa \in (\mathcal{P}^p(\kappa))^m, \forall \kappa \in \mathcal{T}_H\}, \tag{2.4}$$

where $\mathcal{P}^p(\kappa)$ denotes the space of the p-th order polynomials on κ. Multiplying Eqn. (2.1) by a test function, $\mathbf{v}_H \in \mathcal{V}_H^p$, and integrating by parts over all elements leads to the weak formulation of the conservation law, which reads as follows: find $\mathbf{u}_H(\cdot, t) \in \mathcal{V}_H^p$ such that

$$\sum_{\kappa \in \mathcal{T}_H} \int_\kappa \mathbf{v}_H \frac{\partial \mathbf{u}_H}{\partial t} + \mathcal{R}_H(\mathbf{u}_H, \mathbf{v}_H) = 0, \ \forall \mathbf{v}_H \in \mathcal{V}_H^p, \tag{2.5}$$

where the semi-linear weighted residual (linear in the second argument) is given by

$$\mathcal{R}_H(\mathbf{u}_H, \mathbf{v}_H) = \sum_\kappa \left[\mathbb{E}_\kappa(\mathbf{u}_H, \mathbf{v}_H) + \mathbb{V}_\kappa(\mathbf{u}_H, \mathbf{v}_H) + \mathbb{S}_\kappa(\mathbf{v}_H) \right], \tag{2.6}$$

and $\mathbb{E}_\kappa(\mathbf{u}_H, \mathbf{v}_H)$, $\mathbb{V}_\kappa(\mathbf{u}_H, \mathbf{v}_H)$, and $\mathbb{S}_\kappa(\mathbf{v}_H)$ denote the contributions of the inviscid, viscous, and source terms, respectively. Specifically,

$$\mathbb{E}_\kappa(\mathbf{u}_H, \mathbf{v}_H) \equiv - \int_\kappa \nabla \mathbf{v}_H^T \cdot \mathcal{F}^i(\mathbf{u}_H) d\mathbf{x} + \int_{\partial\kappa} (\mathbf{v}_H^+)^T \hat{\mathbf{F}}(\mathbf{u}_H^+, \mathbf{u}_H^-, \hat{\mathbf{n}}) ds,$$

where $\hat{\mathbf{F}}$ is a numerical flux function, $\hat{\mathbf{n}}$ is the outward pointing normal, and $(\cdot)^+$ and $(\cdot)^-$ denote trace values taken from opposite sides of a face. In this work, $\hat{\mathbf{F}}$ is the Roe numerical flux function [66]. Boundary conditions are enforced weakly by appropriately setting the numerical flux on the domain boundaries. The boundary treatment for the Navier-Stokes equations can be found in Oliver [55]. The viscous flux contribution, $\mathbb{V}_\kappa(\mathbf{u}_H, \mathbf{v}_H)$, is discretized according to the second form of Bassi and Rebay (BR2) [5], which is dual consistent and requires only the nearest neighbor coupling. Further details on the discretization of the viscous term can be found in Fidkowski *et al.* [27]. The discretization of the source term is given by

$$\mathbb{S}_\kappa(\mathbf{v}_H) \equiv - \int_\kappa \mathbf{v}_H^T \mathcal{S}(\mathbf{x}, t) d\mathbf{x}.$$

To obtain the semi-discrete form of the governing equation, a basis for the function space, $\mathcal{V}_H^p$, is chosen. In particular, let $\{\boldsymbol{\phi}_i\}$ for $i = 1, ..., N$ be a basis of the space

$\mathcal{V}_H^p$. Then any function $\mathbf{w}_H \in \mathcal{V}_H^p$ can be expressed as

$$\mathbf{w}_H = \sum_{i=1}^{N} \mathbf{W}_i \boldsymbol{\phi}_i(\mathbf{x}),$$

where $\mathbf{W} \equiv \{\mathbf{W}_i\} \in \mathbb{R}^N$ is the vector of expansion coefficients. Let $\mathbf{U}_H$ be the vector of expansion coefficients for the solution $\mathbf{u}_H$, then the semi-discrete form of the weak formulation in Eqn. (2.5) can be written as a system of ODEs: given $\mathbf{U}_H(0)$, find $\mathbf{U}_H(t)$ such that

$$\mathbf{M}_H \frac{d\mathbf{U}_H}{dt} + \mathbf{R}_H(\mathbf{U}_H) = 0, \tag{2.7}$$

where $\mathbf{R}_H(\mathbf{U}_H)$ is the discrete residual vector such that

$$\mathbf{R}_H(\mathbf{U}_H)_i = \mathcal{R}_H(\mathbf{u}_H, \boldsymbol{\phi}_i),$$

and $\mathbf{M}_H$ is the mass matrix given by

$$\mathbf{M}_{H_{ij}} = \int_\Omega \boldsymbol{\phi}_i^T \boldsymbol{\phi}_j d\mathbf{x}.$$

2.3 Temporal Discretization

Although the focus of this work is only on steady-state solutions corresponding to $\mathbf{R}_H(\mathbf{U}_H) = 0$, a pseudo-time stepping is applied to improve the robustness of the solver for nonlinear equations. The solution is time marched to a steady state using a backward Euler discretization in which the timestep is raised to essentially infinite values as the solution converges. In particular, the solution vector at a time interval, $n+1$, is given by

$$\mathbf{U}_H^{n+1} = \mathbf{U}_H^n - \left(\frac{\partial \mathbf{R}_H}{\partial \mathbf{U}_H} + \frac{1}{\Delta t} \mathbf{M}_H \right)^{-1} \mathbf{R}_H(\mathbf{U}_H^n). \tag{2.8}$$

2.4 Linear System

The time-marching scheme given by Eqn. (2.8) requires at each time step the solution to the linear system $\mathbf{Ax} = \mathbf{b}$, where

$$\mathbf{A} \equiv \mathbf{K}_H + \frac{1}{\Delta t}\mathbf{M}_H, \quad \mathbf{x} \equiv \Delta \mathbf{U}_H^n, \quad \mathbf{b} \equiv -\mathbf{R}_H(\mathbf{U}_H^n), \tag{2.9}$$

and $\mathbf{K}_H \equiv \frac{\partial \mathbf{R}_H}{\partial \mathbf{U}_H}$. The matrix $\mathbf{A}$ is often referred to as the Jacobian matrix. When the time step, Δt, is small, the Jacobian matrix is easy to solve because it is dominated by the mass matrix, $\mathbf{M}_H$, which is a block diagonal matrix for the DG discretization presented. On the other hand, as Δt increases, the Jacobian matrix becomes more dominated by $\mathbf{K}_H$, which takes into account the coupling between neighboring elements that can be strong in general, and as such, the Jacobian matrix becomes more difficult to solve.

For the DG discretization presented, the Jacobian matrix has a sparse block-structure with N_e block rows, where N_e is number of elements in the computational domain. Each block has a size of n_b, which is the number of degrees of freedom in one element. Each block row has a non-zero diagonal block, which corresponds to the coupling between degrees of freedom within one element. In addition, because the BR2 scheme requires only the nearest neighbor coupling [5], each block row of the Jacobian matrix also involves n_f off-diagonal blocks, where n_f is the number of faces per element ($n_f = 3$ for 2D, and $n_f = 4$ for 3D).

2.5 Linear Solution Method

Knowing the large size of the Jacobian matrix and its block-sparse structure, iterative methods are used to solve the linear system given in Eqn. (2.9). More specifically, a restarted GMRES algorithm is applied [68, 70]. To expedite the convergence of the GMRES algorithm, a preconditioner must be applied in general to improve the conditioning of the linear system. In this work, the preconditioner considered is a dual threshold incomplete LU factorization [69], ILUT(p, τ), where p is the number of allowed fill-in's per row, and τ is the fill-in drop threshold. Since the Jacobian matrix has a block structure, ILUT(p, τ) is implemented in a block form, where the Frobenius norm is used to measure the contribution of a block. The standard ILU(0), being a particular case of ILUT(p, τ) with $p = 0$ and $\tau \to \infty$, is also considered in this study and implemented in a block form.

The efficiency of incomplete factorization is highly dependent on the ordering of unknowns, especially for a non-symmetric system [8]. The reordering scheme used in this work is the Minimum Discarded Fill (MDF) method, where the element that produces the least discarded fill-in is ordered first and the process is repeated in a greedy manner. This method was presented by D'Azevdo *et al.* [15], and modified for a block matrix and reported to work well with ILU(0) in a DG context [63]. For the block form of the MDF algorithm, each block of the matrix is reduced to a scalar using the Frobenius norm, and the reordering is based on the reduced scalar matrix. For efficiency comparison, a reordering of elements based on lines of maximum coupling is also considered in this work [18, 27].

Chapter 3

Impact of Triangle Shapes

In this chapter, the impact of triangle shapes, including the impact of large angles and aspect ratios, on accuracy and stiffness is investigated for simulations of highly anisotropic problems. Sections 3.1 and 3.2 present the model problems and meshes considered, including large-angle elements and right-angle elements. Section 3.3 studies the discretization errors on these meshes, and Section 3.4 studies the conditioning of the linear systems arising from these meshes.

3.1 Model Problems

To study the impact of large angles and aspect ratios, a physical problem with a highly anisotropic behavior is needed. For such a problem, large-angle elements may be generated from an anisotropic mesh generation scheme, such as a mapped Delaunay triangulation. Thus, a thin boundary layer case is used for this study, and the governing equations considered include the advection-diffusion equation and the Navier-Stokes equations.

3.1.1 Advection-Diffusion Equation

The advection-diffusion equation is described in Section 2.1. For the anisotropic problem considered in this chapter, the convection velocity is uniform and horizontal with a unit magnitude, and a diffusivity of $\mu = 10^{-8}$ is introduced. The computational domain is a rectangular box of $[0.05, 1.05] \times [0, 0.001]$. A source term is added such that the exact solution to this problem has a form of

$$u = 1 - e^{\frac{-y}{\sqrt{c\mu x}}}, \tag{3.1}$$

with $c = 0.59$. This solution is shown in Figure 3-1 and resembles a thin boundary layer growing with $\sqrt{x}$ along the bottom wall. It has a thickness of $\delta_{0.99} = 8 \times 10^{-5}$ at the inflow and 3.6×10^{-4} at the outflow. Note that the leading edge of the boundary layer is not included in the computational domain.

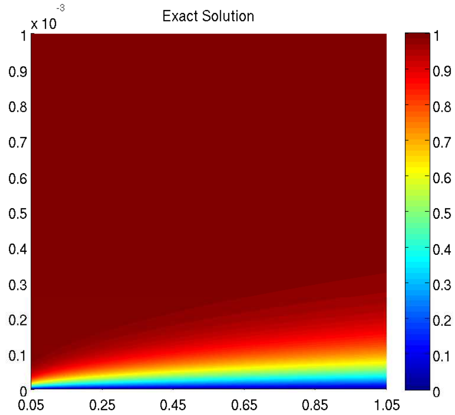


Figure 3-1: Exact solution of u for triangle shape study

3.1.2 Navier-Stokes Equations

The compressible Navier-Stokes equations are given in Eqn. (2.3), and the ideal gas law is assumed. A source term is created to make an exact solution of $\rho, P = \text{const}$, $v = 0$, and u given by (3.1), with $Re = 10^8$ and $M_\infty = 0.1$. The computational domain is the same as for the case of the advection-diffusion equation.

3.2 Meshes

As previously cited in Section 1.2.1, a realistic goal for anisotropic meshing for linear elements is to generate triangles with an aspect ratio of $\sqrt{\sigma_{\max}/\sigma_{\min}}$, where the σ's are the singular values of the solution Hessian [72]. Also, at this aspect ratio, angle size is not critical for solution accuracy [72]. In fact, this value is the target aspect ratio in many anisotropic adaptation schemes [19, 37, 62, 75], in which interpolation errors are equidistributed in all directions. In this chapter, this aspect ratio is used as a guideline for the meshes considered, and the element orientation is decided based on the angle θ of the singular vectors of the Hessian matrix.

Figure 3-2 shows the ratio $\sigma_{\max}/\sigma_{\min}$ of the Hessian of the solution considered in Eqn. (3.1). Along the bottom wall, which will be the main source of discretization error, the ratio of $\sigma_{\max}/\sigma_{\min}$ is about 10^8, so an aspect ratio of $\mathcal{AR} \approx 10^4$ might be expected to result in a high-quality solution for linear elements. Three different aspect ratios are considered in this work: 10^3, 10^4, and 10^5. In addition, for the solution considered, θ is essentially zero everywhere in the computational domain, and has a maximum value of 0.14°. This indicates that the major axes of elements should be aligned horizontally.

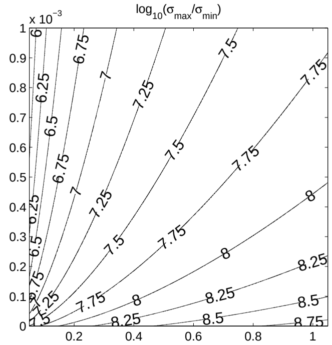
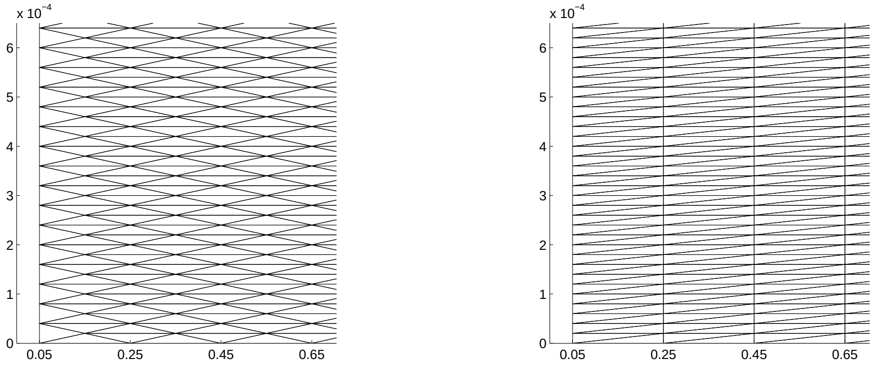


Figure 3-2: Contour of $\log_{10}(\sigma_{\max}/\sigma_{\min})$ of the Hessian for u in Eqn. (3.1)

To study the impact of large angles, elements with angles close to 180° (obtuse triangles) are compared with right-angle elements (acute triangles), which were believed to be of high-quality for a boundary layer problem. Figure 3-3 shows the meshes of these two kinds of triangles. Note the y-axis is rescaled only for the convenience of visualization, so in the physical domain, each of the obtuse triangles has one angle of about 180°, and each of the acute triangles has essentially two angles of about 90°. For the aspect ratio of 10^4, the large angle in the obtuse triangles is 179.9771°.



(a) Obtuse, one 180°-angle

(b) Acute, two 90°-angles

Figure 3-3: Obtuse and acute meshes

3.3 Discretization Error

An error convergence study is conducted for the advection-diffusion equation and the Navier-Stokes equations on the meshes of all aspect ratios considered. The L^2 error and the error of a selected output are used to assess the solution quality on these meshes. In engineering applications, the error of a functional output often draws more interests than the global error. In this section, the L^2 errors are normalized by the L^2 norm of the true solution, that is,

$$\mathcal{E} \equiv \frac{\|u - u_e\|_{L^2(\Omega)}}{\|u_e\|_{L^2(\Omega)}},$$

where u_e is the exact solution. The output errors are normalized by the true output value. For all cases, solution orders of $p = 1$ and $p = 3$ are considered.

3.3.1 Advection-Diffusion Equation

Figure 3-4 shows the normalized L^2 error for the acute and obtuse meshes of different aspect ratios. The mesh size h is defined as $1/\sqrt{N}$, where N is the total number of elements. For the linear solutions, an aspect ratio of $\mathit{AR} = 10^4$ generally leads to the lowest error at a given DOF for both shapes. This result is consistent with the conclusion that an aspect ratio of $\sqrt{\sigma_{\max}/\sigma_{\min}}$ leads to the lowest L^2 error for linear elements regardless of angle sizes [12, 65, 72]. While the solution quality degrades slightly when the aspect ratio changes from 10^4 to 10^3, the degradation is more severe for $\mathit{AR} = 10^5$. In addition, the adverse impact of large angles is apparent only for $\mathit{AR} = 10^5$. For the high-order solutions, the aspect ratios of 10^4 and 10^3 lead to very similar errors, but $\mathit{AR} = 10^5$ results in a significantly higher error. However, the difference between the acute and obtuse meshes is minor for any aspect ratio. In other words, while the impact of aspect ratios on solution accuracy is critical, the impact of large angles is minor for high-order schemes.

The error of the heat transfer across the bottom wall is also measured. This output is given by

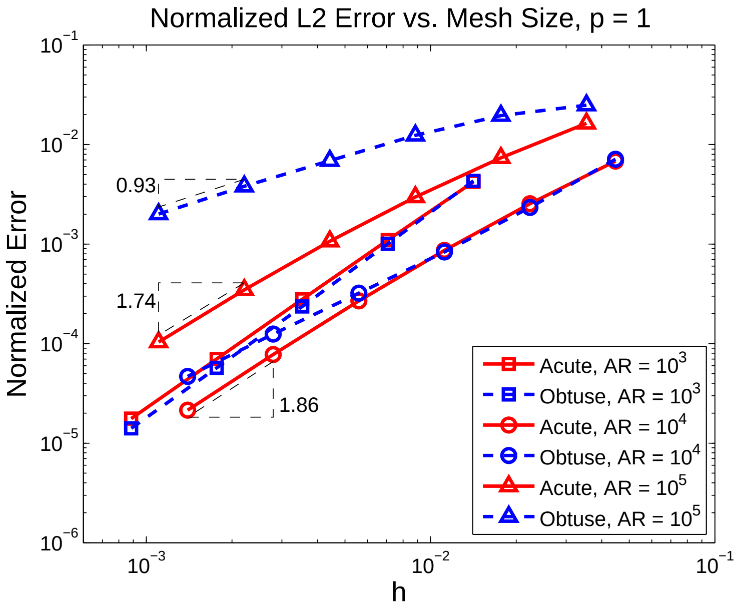
$$J = \int_{\text{bottom wall}} \mu \frac{\partial u}{\partial y} dx. \tag{3.2}$$

Figure 3-5 shows the normalized heat error for the acute and obtuse meshes. The results lead to a similar conclusion to what was observed for the L^2 error, where only the linear solutions are susceptible to large angles; that is, the difference between the two element shapes is less pronounced for high-order solutions than linear solutions. However, a correct aspect ratio is critical for the error level. While the best aspect ratio is $\mathit{AR} = 10^4$ for both element shapes and for both interpolation orders, one important difference compared to the results of L^2 error is that $\mathit{AR} = 10^4$ is considerably better than both $A\!\!\!R = 10^3$ and 10^5 for linear elements. Also, note that the error reaches machine precision as the meshes are refined for the high-order solutions.

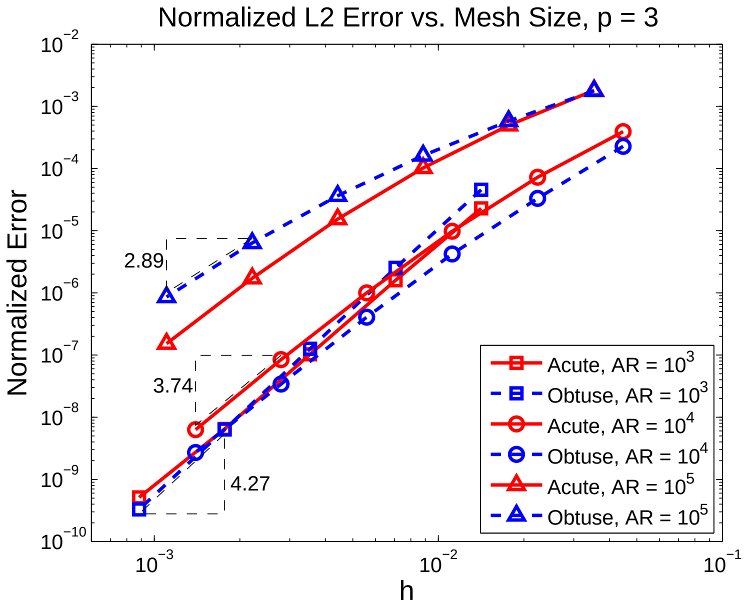
3.3.2 Navier-Stokes Equations

Figure 3-6 shows the convergence of the normalized L^2 error of ρu for the Navier-Stokes equations. The conclusion drawn is very similar to that of the L^2 error for the advection-diffusion case. For the linear solutions, an aspect ratio of $A\!\!\!R = 10^4$ generally leads to the lowest error at a given DOF for both element shapes, and the degradation of solution quality is more severe for $A\!\!\!R = 10^5$ than for $A\!\!\!R = 10^3$. Also, the impact of large angles is more pronounced for $A\!\!\!R = 10^5$ than other aspect ratios. For the high-order solutions, the aspect ratios of 10^4 and 10^3 lead to very similar errors, but $A\!\!\!R = 10^5$ results in a significantly higher error. However, the difference between the two triangle shapes is minor for any aspect ratio.

Figure 3-7 shows the normalized output error for the acute and obtuse meshes, where the output is selected to be drag on the bottom wall. For linear solutions, at the aspect ratios of 10^3 and 10^4, the obtuse and acute elements lead to similar error levels. However, at $A\!\!\!R = 10^5$, the acute elements outperform the obtuse ones significantly, and in fact, this shape leads to the lowest error for linear solutions. This optimality of acute triangles is very fragile in the sense that the error significantly increases when the aspect ratio deviates from 10^5. In fact, as it decreases to 10^3, the acute element becomes the shape with the highest error for linear solutions. This agrees with Shewchuk's conclusion about the fragile "superaccuracy" of H^1 error for acute linear elements [72], which should happen at an aspect ratio of about 10^8 in the case studied. On the other hand, for high-order solutions, the obtuse elements lead to a similar error compared to the acute ones for any aspect ratio, and the best triangle shape is the obtuse ones with $A\!\!\!R = 10^4$. Furthermore, for drag error, the impact of aspect ratio is again very pronounced for both linear and high-order solutions and for both element shapes.

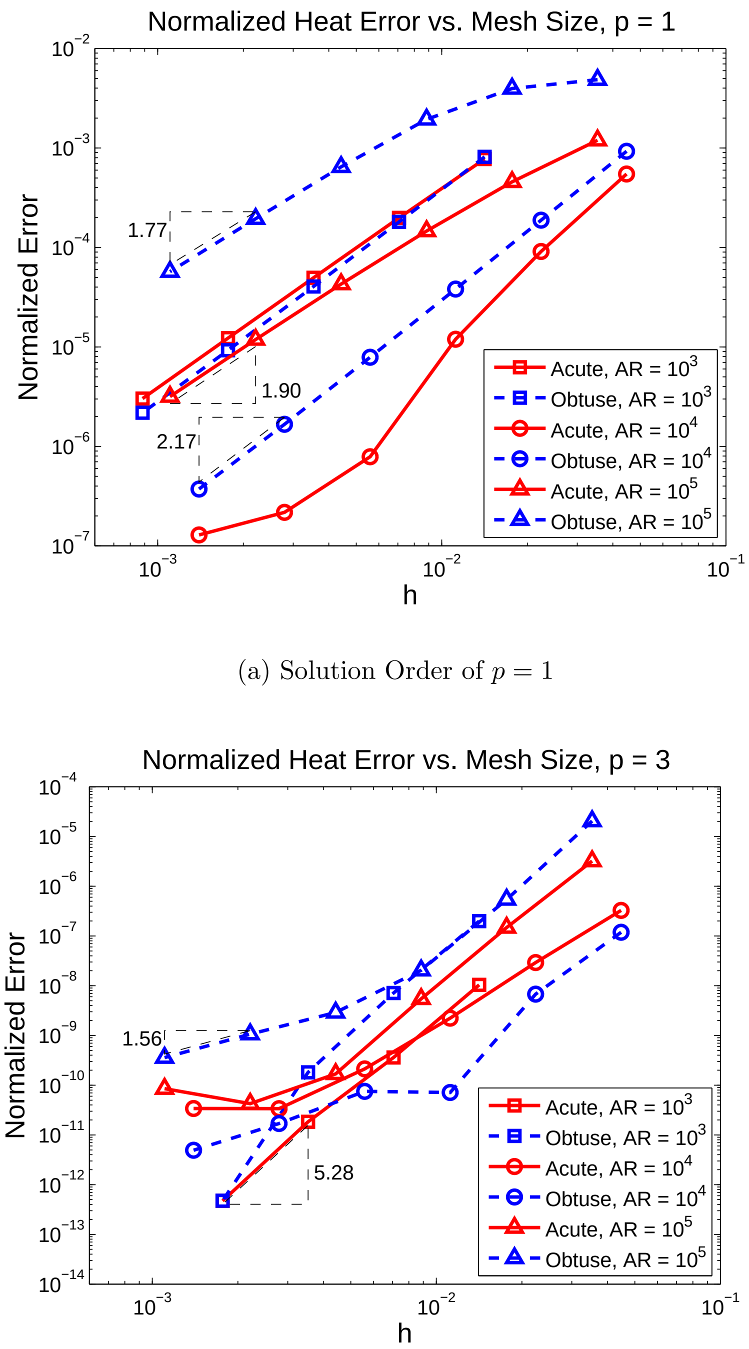


(a) Solution Order of $p = 1$



(b) Solution Order of $p = 3$

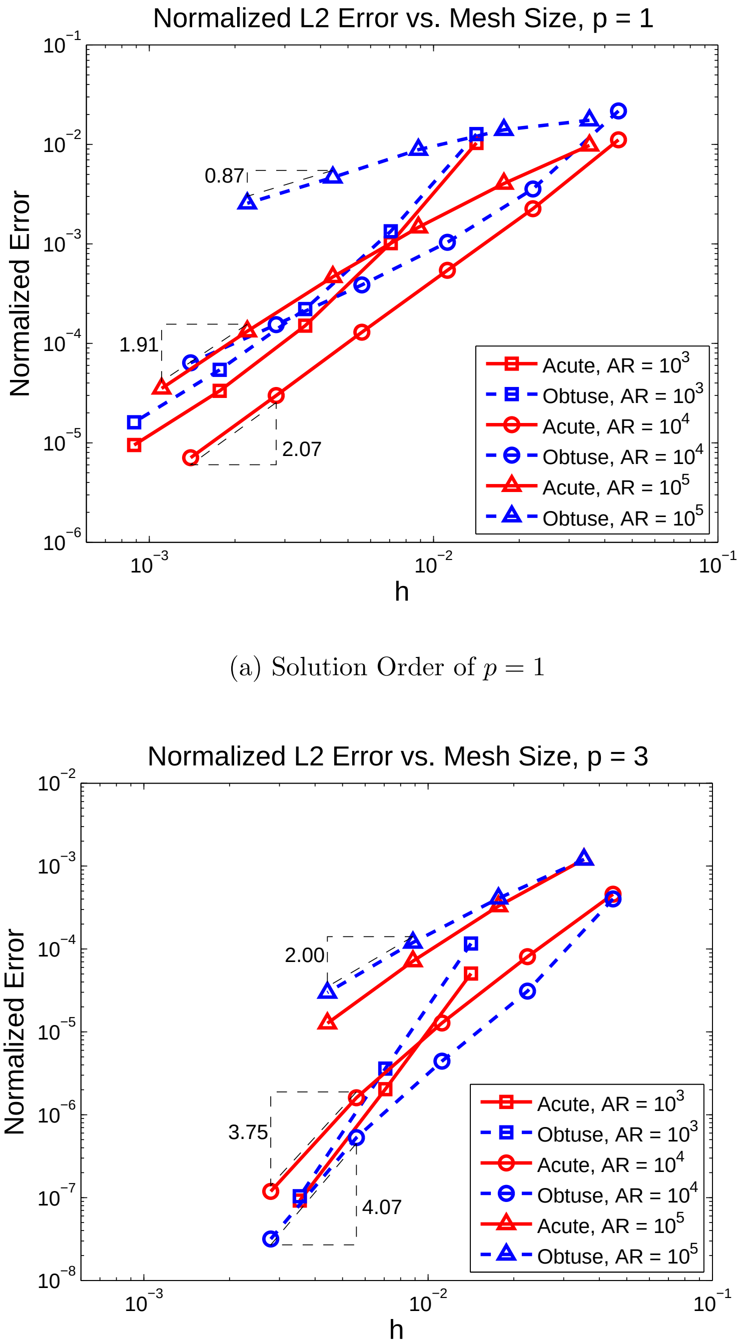
Figure 3-4: Normalized L^2 error for the advection-diffusion problem



(a) Solution Order of $p = 1$

(b) Solution Order of $p = 3$

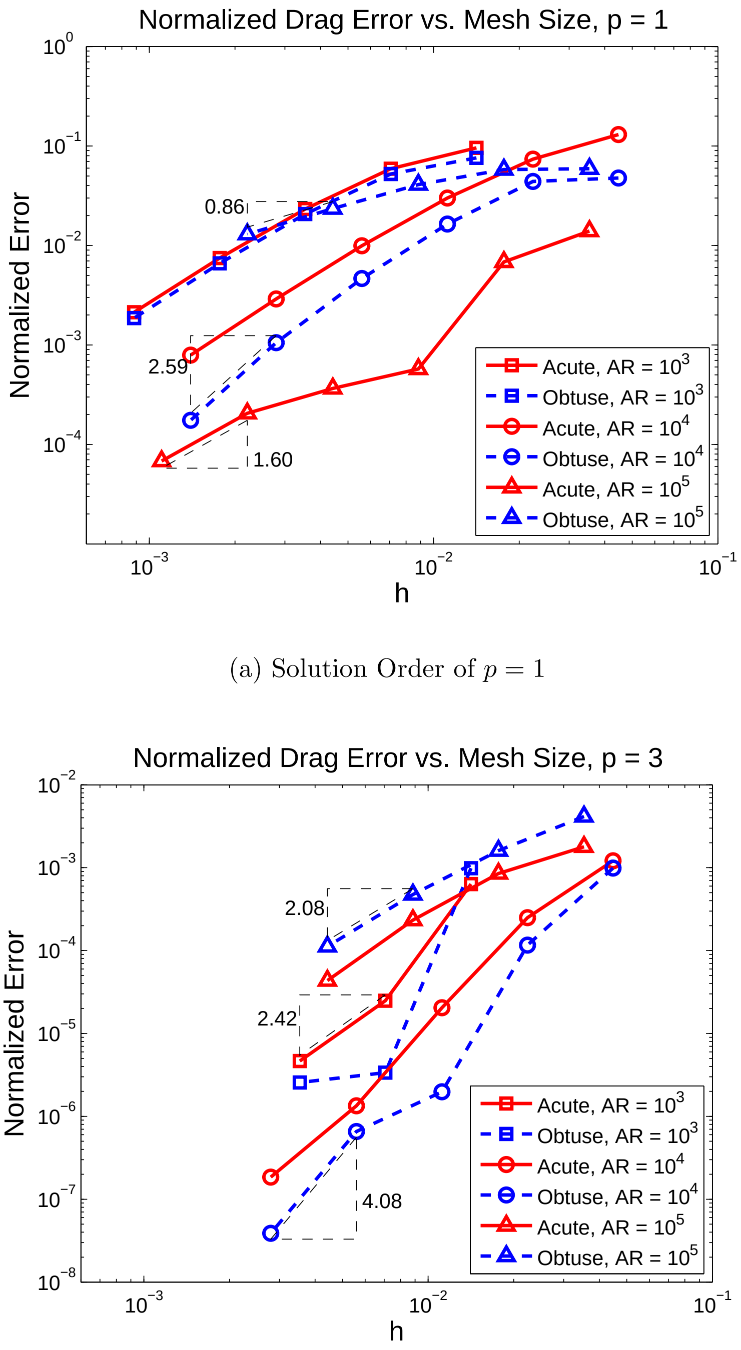
Figure 3-5: Normalized heat error for the advection-diffusion problem



(a) Solution Order of $p = 1$

(b) Solution Order of $p = 3$

Figure 3-6: Normalized L^2 error for the Navier-Stokes equations



(a) Solution Order of $p = 1$

(b) Solution Order of $p = 3$

Figure 3-7: Normalized drag error for the Navier-Stokes equations

Now that the high-order solutions have been shown to be less sensitive to element angles, the use of high-order schemes needs to be justified from the perspective of the degrees of freedom needed for an error level. Figure 3-8 presents the worst and best output errors from the linear solutions and the high-order solutions from Figure 3-7. Clearly, the high-order solutions lead to a lower error for almost all degrees of freedom. To put the error of drag in context, consider for example a typical large, long-range, passenger jet, an error of 1% for drag translates into approximately 4-8 passengers, depending on whether the configuration is limited by fuel volume or weight [25, 73]. Thus, for the two-dimensional boundary layer problem studied, an output error level of $0.01\% - 1\%$ is a realistic range. Looking at this error range, all the meshes considered can reach this level with high-order discretization. However, the best linear solution, which is obtained only on the acute meshes at a very sensitive optimal aspect ratio, can barely compete with the worst $p = 3$ solution. In other words, having an optimal mesh is less important to reach the engineering required error range for high-order discretization. Therefore, one would prefer using high-order schemes, which not only lead to fewer degrees of freedom but more importantly, mitigate the impact of large angles.

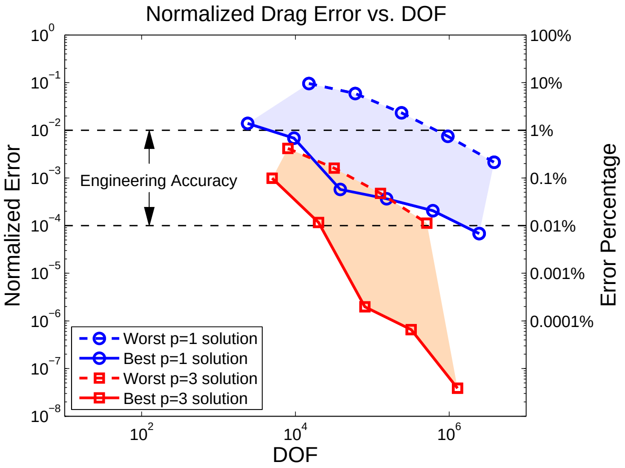


Figure 3-8: Normalized drag error versus DOF for the Navier-Stokes equations

3.4 Conditioning of the Linear System

One important concern of the large-angle elements is their numerical conditioning, which dictates the efficiency of the iterative linear solvers applied. This section compares the conditioning of the linear systems obtained from the acute and obtuse meshes, and studies the efficiency of the incomplete LU preconditioning for these meshes. In this section, the matrices $\mathbf{K}_H$ and $\mathbf{M}_H$ defined in Section 2.4 are denoted by $\mathbf{K}$ and $\mathbf{M}$, respectively, for simplicity.

3.4.1 Eigenvalue Spectrum

All eigenvalue calculations in this section are for the advection-diffusion problem. Table 3.1 lists the ratio of the maximum and the minimum (by moduli) eigenvalues of $\mathbf{M}^{-1}\mathbf{K}$. The spectrum of $\mathbf{M}^{-1}\mathbf{K}$ indicates the ellipticity of the underlying problem and is independent of the type of interpolation basis chosen. For all the aspect ratios considered, the acute and obtuse meshes have similar conditioning.

Mesh	p	$\lambda_{\max}/\lambda_{\min}$	
		Acute	Obtuse
Æ $= 10^3$, 512 elements	1	12.577	8.2904
	3	97.253	75.430
Æ $= 10^4$, 500 elements	1	263.13	386.19
	3	1241.3	1746.0
Æ $= 10^5$, 800 elements	1	10365.1	16590.9
	3	48448.4	60505.5

Table 3.1: Eigenvalues of $\mathbf{M}^{-1}\mathbf{K}$

Figure 3-9 shows the eigenvalue footprint for meshes of different aspect ratios. Clearly, the problem is diffusion dominated as the eigenvalues are mainly real, especially for high aspect ratios. In addition, comparing the eigenvalue footprints for different aspect ratios indicates that the problem becomes more elliptic as the aspect ratio increases, i.e., as the element y-spacing decreases for about the same number of elements. Table 3.1 shows that the conditioning on the acute meshes worsens more slowly than on the obtuse ones as the problem becomes more elliptic.

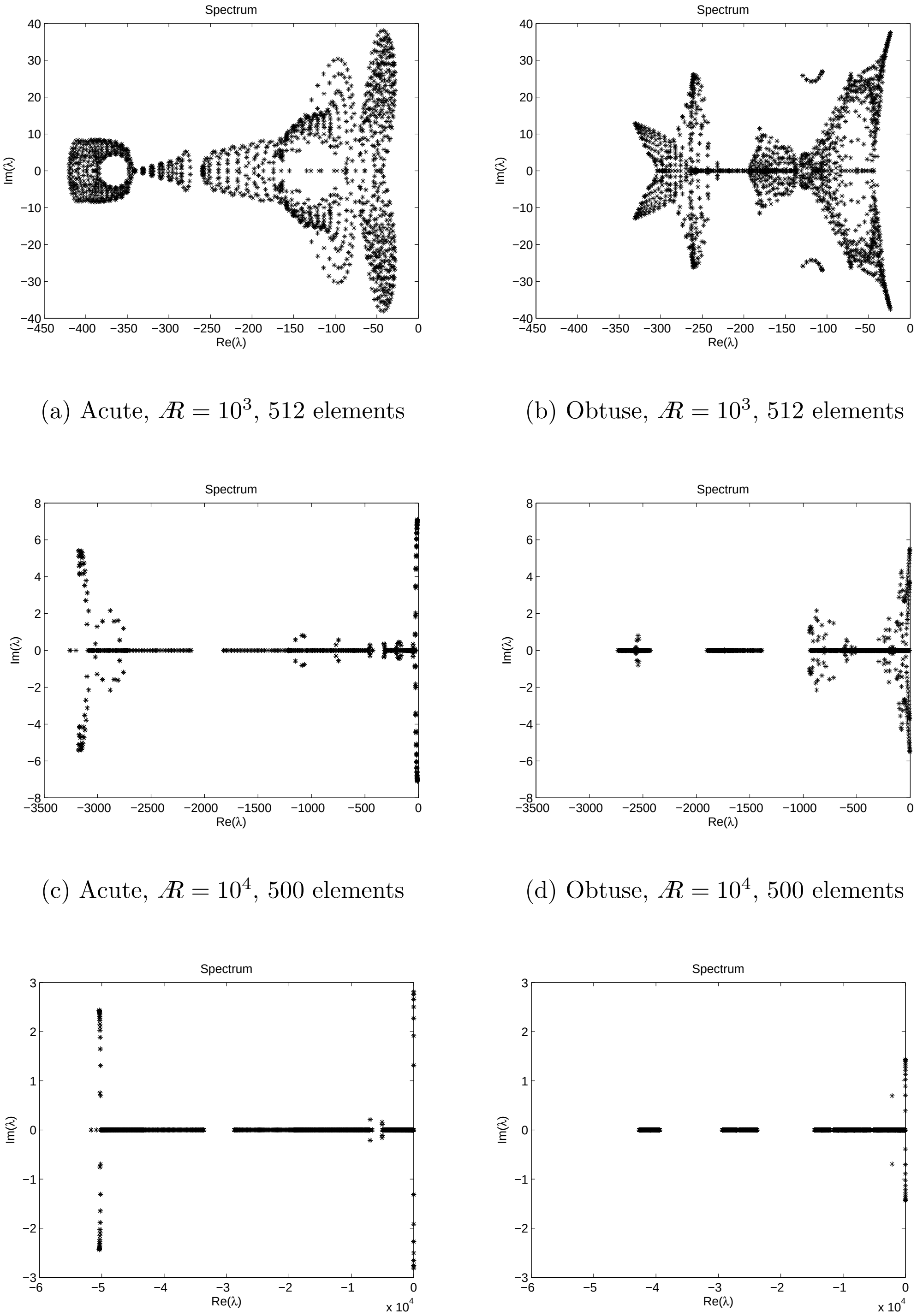
Another way to compare the conditioning is to look at the eigenvalue distributions of the block-Jacobi preconditioned system, $\mathbf{D}^{-1}\mathbf{K}$, where $\mathbf{D}$ is the diagonal blocks of $\mathbf{K}$. The spectrum is expected to fall in the unit circle centered at $(-1, 0)$, and as high-frequency errors can be removed fairly easily through preconditioning, the smallest eigenvalue (by moduli) implies the difficulty in solving the system. Table 3.2 lists the smallest eigenvalues for different meshes, and it confirms that the acute and obtuse elements have similar conditioning. Figure 3-10 shows the eigenvalue footprint of $\mathbf{D}^{-1}\mathbf{K}$ for different meshes.

Mesh	p	$\lambda_{\min}$		$\lambda_{\max}/\lambda_{\min}$	
		Acute	Obtuse	Acute	Obtuse
$AR = 10^3$, 512 elements	1	0.2112	0.24356	8.4589	7.2752
	3	0.0072	0.0096	26.857	19.873
$AR = 10^4$, 500 elements	1	0.0164	0.0103	121.13	193.74
	3	0.0076	0.0059	262.37	338.92
$AR = 10^5$, 800 elements	1	0.00042	0.00032	4714.9	6237.1
	3	0.00021	0.00024	9294.1	8179.0

Table 3.2: Eigenvalues of block-Jacobi preconditioned $\mathbf{K}$

3.4.2 Preconditioning Strategies

Although the conditioning of the linear system arising from the two meshes is similar, comparing preconditioned systems has more practical value as the Krylov subspace methods must be preconditioned to perform well. This section studies the performance of ILU(0) and the more general ILUT(p, τ) on both the acute and the obtuse meshes. Because the previous section has concluded that high-order discretization is preferred, only the preconditioning for the $p = 3$ discretization will be studied. The model problem is mainly elliptic for linear interpolations as shown in Figure 3-9, and it is more elliptic for high-order interpolations, with which the characteristic length of the discretization is shorter. For all the results in this section, the linear solver is iterated until the preconditioned residual reaches 10^{-14}, relative to the initial residual.



(a) Acute, $\mathcal{R} = 10^3$, 512 elements

(b) Obtuse, $\mathcal{R} = 10^3$, 512 elements

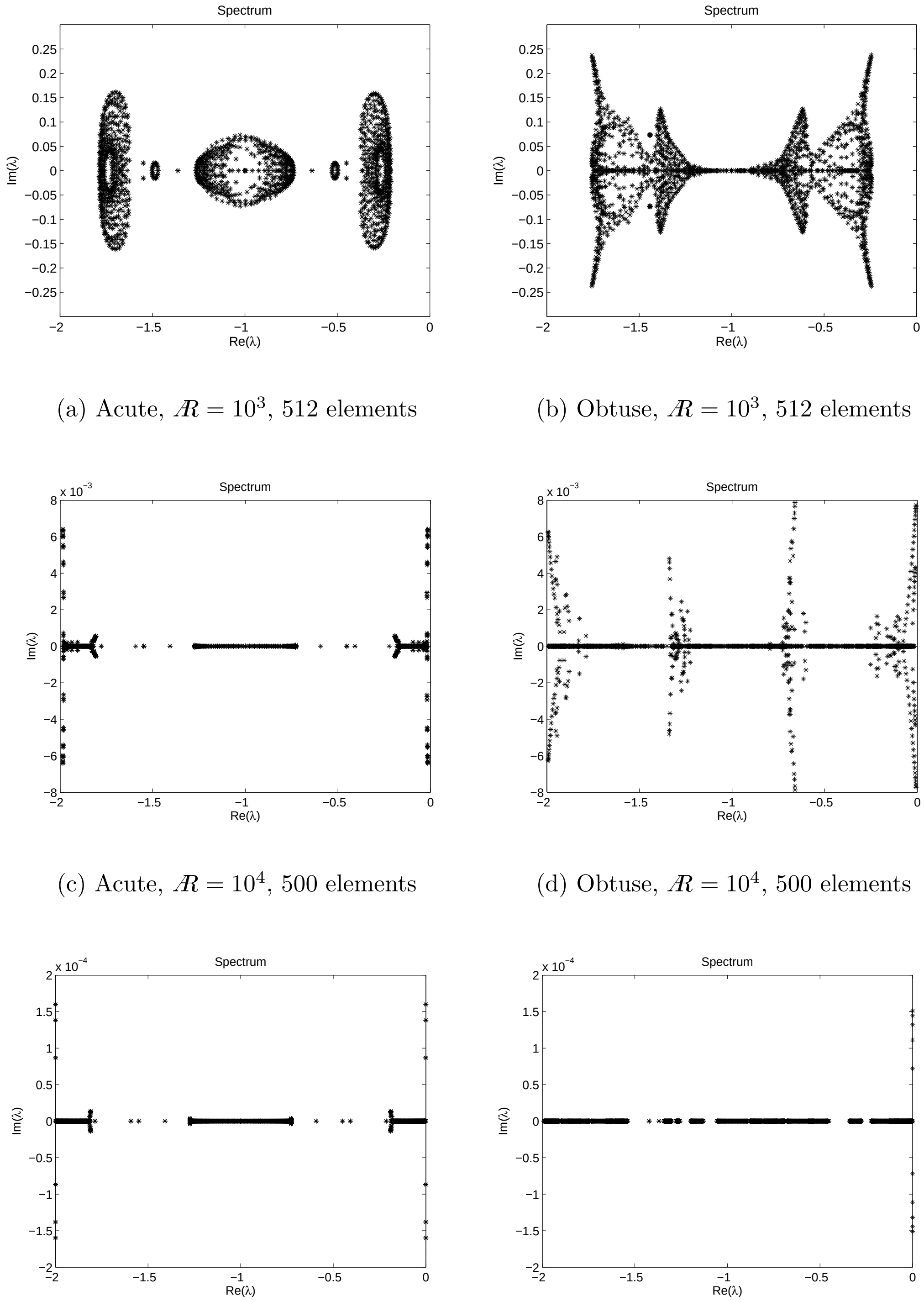
(c) Acute, $\mathcal{R} = 10^4$, 500 elements

(d) Obtuse, $\mathcal{R} = 10^4$, 500 elements

(e) Acute, $\mathcal{R} = 10^5$, 800 elements

(f) Obtuse, $\mathcal{R} = 10^5$, 800 elements

Figure 3-9: Eigenvalue footprint of $\mathbf{M}^{-1}\mathbf{K}$ using linear interpolation



(a) Acute, $AR = 10^3$, 512 elements

(b) Obtuse, $AR = 10^3$, 512 elements

(c) Acute, $AR = 10^4$, 500 elements

(d) Obtuse, $AR = 10^4$, 500 elements

(e) Acute, $AR = 10^5$, 800 elements

(f) Obtuse, $AR = 10^5$, 800 elements

Figure 3-10: Eigenvalue footprint of $\mathbf{D}^{-1}\mathbf{K}$ using linear interpolation

Performance of ILU(0)

A sufficient condition for ILU(0) being the same as an exact factorization is that the linear system being solved is tridiagonal. For the Jacobian matrix arising from the presented DG discretization, this condition is equivalent to the following:

Condition 3.4.1. Each element on the mesh has no more than two neighbors with non-zero coupling.

This condition is equivalent to being able to partition the mesh into independent lines, where within each line, each element has only two neighbors. This condition can be rarely met in practice. However, ILU(0) can work very efficiently with an appropriate reordering if a mesh is close to satisfying Condition 3.4.1, i.e., some elements might have a third neighbor with weak coupling instead of zero coupling.

In general, ILU(0) works poorly for diffusion-dominated problems due to the isotropic nature of diffusion, which usually leads to strong couplings between all neighboring elements. However, for highly anisotropic meshes and/or convection-dominated problems, coupling in one direction can be much weaker than in another. Thus, with an appropriate ordering of unknowns, ILU(0) works efficiently on the acute meshes for the studied model problem. Table 3.3 lists the number of GMRES iterations using ILU(0) for acute and obtuse meshes of different aspect ratios. It also compares the performance of the line reordering scheme [18] and the MDF reordering scheme [63].

Reordering	$A\!\!R = 10^3$, 512 elem.		$A\!\!R = 10^4$, 500 elem.		$A\!\!R = 10^5$, 800 elem.	
	Acute	Obtuse	Acute	Obtuse	Acute	Obtuse
Line	33	29	6	101	13	367
MDF	16	29	6	50	4	183

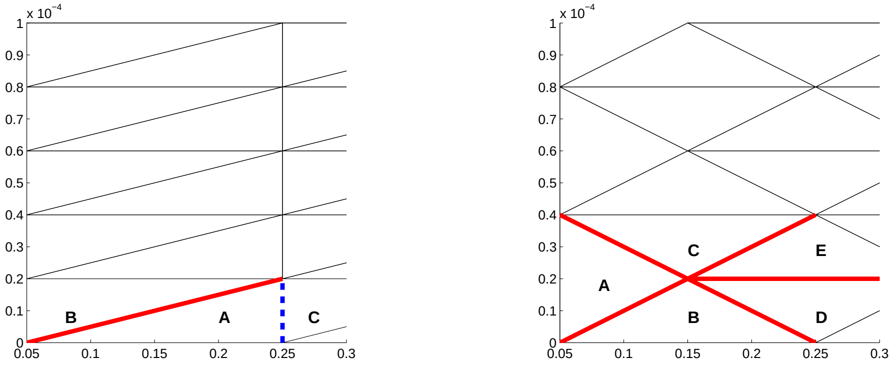
Table 3.3: The GMRES iteration counts using ILU(0) for the model scalar advection-diffusion problem on different meshes

First, note that the MDF reordering outperforms the line reordering for all cases, especially when the problem is more elliptic. Second, looking at only the results using the MDF reordering, the table shows that ILU(0) works poorly for obtuse meshes, and as the problem becomes more elliptic, the efficiency of ILU(0) drops quickly. This can be explained by the strength of coupling between neighboring elements, which governs the performance of ILU(0).

Let the coupling weight for an element κ with its neighbor κ' be the corresponding entry in $\mathbf{K}$ normalized by the diagonal entry for κ, that is, $\eta_{\kappa\kappa'} \equiv \mathbf{K}_{\kappa\kappa'}/\mathbf{K}_{\kappa\kappa}$, where $\mathbf{K}_{ij}$ denotes the ij block in the matrix $\mathbf{K}$ measured in a certain norm, for example, Frobenius norm.

For a diffusion-dominated problem, the coupling weight $\eta_{\kappa\kappa'}$ scales with $(l_f/d_\kappa)^2$, where l_f is the length of the face between κ and κ', and d_κ is a measure of the diameter of the element κ. This is demonstrated in Appendix A.1 through an order analysis for the discretization applied. A highly anisotropic right triangle has one edge much shorter than the other two, and so for the model problem considered, the coupling weight on this short edge is negligible. Thus, the acute meshes are close to satisfying Condition 3.4.1. Furthermore, as the aspect ratio increases, the short edge becomes even shorter, and the coupling across this edge becomes even more insignificant, making ILU(0) perform even better. This is consistent with the results in Table 3.3. On the other hand, an obtuse triangle does not have this property as all its three edges have similar lengths. This is illustrated in Figure 3-11. As the model problem becomes more elliptic, the coupling across each edge on the obtuse meshes becomes stronger. Conversely, if less diffusion were introduced such that the problem became advection dominated, ILU(0) would work equally well on the acute and the obtuse meshes.

In summary, for ILU(0), the key factor for efficiency is not the maximum angle involved, but the connectivity weight with neighbors, or the strength of coupling among elements. If all three edges of one element have equal coupling strength, regardless of the largest angle involved, then Condition 3.4.1 will be far from being satisfied, and ILU(0) will have difficulties working efficiently. In fact, for the same reason, ILU(0) works poorly for a simple diffusion-dominated problem on equilateral triangles, as reported by Persson [63].



(a) Acute Elements

(b) Obtuse Element

Figure 3-11: Coupling weights for diffusion dominated problems. For acute elements, the dashed edge is much shorter than the solid one, so elimination of element A will not introduce a significant fill-in. However, for obtuse elements, all edges have similar length and thus similar connectivity weight, elimination of any element will introduce a large fill-in.

To further demonstrate that angle size by itself is not a determining factor in the efficiency of ILU(0), two diffusion cases with anisotropic diffusivity tensors are implemented. Both cases use the meshes of $\mathcal{AR} = 10^4$ with 500 elements, and only the MDF reordering is applied. An analysis of coupling weights for anisotropic diffusion is shown in Appendix A.2.

Anisotropic Diffusivity Tensor The first case is a pure diffusion problem with a diffusivity tensor given by

$$\bar{\bar{\mu}} = \begin{bmatrix} 1 \times 10^{-8} & 2 \times 10^{-12} \\ 2 \times 10^{-12} & 4 \times 10^{-16} \end{bmatrix}. \tag{3.3}$$

With this diffusivity, the diffusion is zero across one edge of each element on the obtuse meshes, and the coupling is shown in Figure 3-12.

Table 3.4 lists the ratio of maximum and minimum eigenvalues, $\lambda_{\max}/\lambda_{\min}$, for the

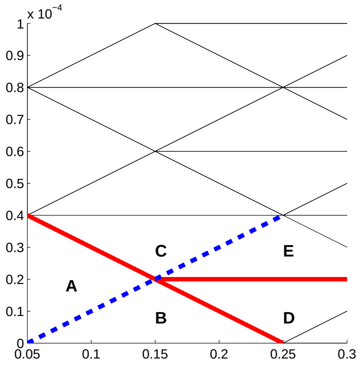


Figure 3-12: Coupling weights for the anisotropic diffusivity in Eqn. (3.3) on the obtuse meshes. The coupling is zero across the dashed edges, so elimination of element A will not introduce any fill-in.

unpreconditioned system, $\mathbf{M}^{-1}\mathbf{K}$, and also the number of GMRES iterations for the ILU(0) preconditioned system with the MDF reordering. As the table shows, although the system for the obtuse mesh is conditioned only a little better than that for the acute one, ILU(0) works very efficiently for the obtuse mesh as argued based on the strength of coupling among elements.

$\lambda_{\max}/\lambda_{\min}$ of $\mathbf{M}^{-1}\mathbf{K}$		GMRES It. for ILU(0)	
Acute	Obtuse	Acute	Obtuse
9.4369×10^4	5.6872×10^4	81	1

Table 3.4: Performance of ILU(0) for anisotropic diffusion in Eqn. (3.3)

The second case has a diffusivity tensor given by

$$\bar{\bar{\mu}} = \begin{bmatrix} 10^{-8} & 0 \\ 0 & \mu \end{bmatrix}, \tag{3.4}$$

where μ controls the diffusivity in the y-direction. As it decreases, the relative importance of the diffusivity in the x-direction increases, and so the coupling strength across the short edges on the acute meshes increases.

Table 3.5 lists $\lambda_{\max}/\lambda_{\min}$ for the unpreconditioned system, and also the number of GMRES iterations for the ILU(0) preconditioned system. As less diffusion is introduced in the y-direction, the unpreconditioned system becomes better conditioned. However, ILU(0) works less efficiently as the three edges of each element become equally weighted.

μ	$\lambda_{\max}/\lambda_{\min}$ of $\mathbf{M}^{-1}\mathbf{K}$	GMRES It. for ILU(0)
10^{-8}	3.122247×10^5	3
10^{-12}	3.114617×10^5	30
10^{-16}	2.137722×10^4	74

Table 3.5: Performance of ILU(0) for anisotropic diffusion in Eqn. (3.4)

Preconditioning on Obtuse Meshes

Knowing that ILU(0) works efficiently for the acute meshes, it is desirable to develop an equally efficient preconditioner for the obtuse meshes. h- and/or p-multigrid preconditioners are known to work well for diffusion-dominated problems, and they have been applied in a DG context [27, 51, 63]. However, in this work, the ILUT(p, τ) is considered for the obtuse meshes, and its efficiency is compared with the ILU(0) results on the acute and obtuse meshes. For the Navier-Stokes equations, the linear system is from the converged solution state.

A fill-in level of two is considered in this work and is found to be more efficient than that of one, but a higher level might be too expensive due to memory concerns. Table 3.6 lists the ILU(0) and ILUT results for the acute and obtuse meshes. As shown in the table, ILU(0) works poorly for the obtuse meshes as explained previously. The ILUT preconditioner results in a significant improvement. For the Navier-Stokes equations, the computational cost on the obtuse meshes using ILUT is comparable to that on the acute meshes using ILU(0). Also, note that as ILU(0) works very efficiently on the acute meshes, especially for the advection-diffusion equation, ILUT does not lead to an obvious improvement on the acute meshes.

Table 3.6: Comparison of preconditioned systems for acute and obtuse meshes [a]

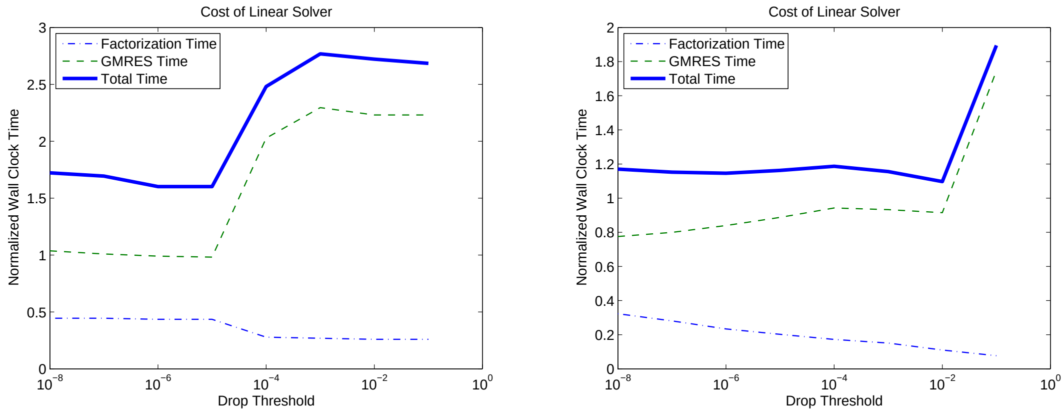
Mesh	Preconditioner	Advection-Diffusion Equation		Navier-Stokes Equations	
		GMRES Iter.	Time[b]	GMRES Iter.	Time[b]
Acute	ILU(0)	6	0.108	86	2.977
Obtuse	ILU(0)	50	0.324	188	7.309
Obtuse	ILUT[c]	11	0.173	69	3.267

[a] $Æ = 10^4$, 500 elements. Solution order $p = 3$.

[b] Total wall clock time of the linear solver including reordering, factorization, and GMRES. The Wall clock time is measured on an Intel Xeon 5130 processor at 2.00GHz. O/S: Red Hat Enterprise Linux 4.

[c] ILUT with a fill-in level of two and a threshold of $\tau = 10^{-6}$.

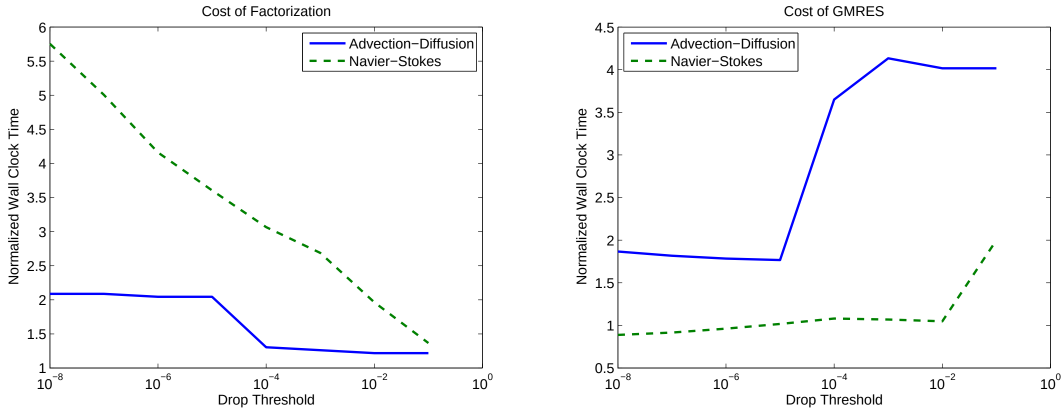
Let the wall clock time of the linear solve using ILU(0) on the acute meshes be $t_a \equiv t_a^{\text{fac}} + t_a^{\text{GMRES}}$, where t_a^{fac} denotes the factorization time and t_a^{GMRES} is the GMRES time. Similarly, let the wall clock time of the linear solve using ILUT on the obtuse meshes be $t_o \equiv t_o^{\text{fac}} + t_o^{\text{GMRES}}$. Figure 3-13 plots the ratio of t_o/t_a versus different drop thresholds for ILUT with a fill-in level of two. With a value of $\tau = 10^{-6}$, t_o is about 60% higher (i.e. slower) than t_a for the advection-diffusion equation. However, it is about 15% higher for the Navier-Stokes equations, for which more complicated characteristics and element-to-element coupling strengths are involved compared to the advection-diffusion equation and so ILU(0) on the acute meshes does not work as efficiently. To further illustrate, Figure 3-14 shows the ratio of $t_o^{\text{fac}}/t_a^{\text{fac}}$ and $t_o^{\text{GRMES}}/t_a^{\text{GMRES}}$ for the two equations. As seen, the factorization time ratio, $t_o^{\text{fac}}/t_a^{\text{fac}}$, is significantly higher for the Navier-Stokes equations as bigger matrix block sizes are involved. On the other hand, the GMRES time ratio, $t_o^{\text{GRMES}}/t_a^{\text{GMRES}}$, is higher for the advection-diffusion equation as ILU(0) is close to an exact factorization for this equation on the acute meshes. Since GMRES time is dominant in the total linear solver time, the difference between t_a and t_o is much smaller for the Navier-Stokes equations than for the advection-diffusion equation. Also, note that for each of the model equations, there is a critical value of τ, above which the GMRES time significantly increases, indicating that important information is dropped in the factorization stage.



(a) Advection-diffusion equation

(b) Navier-Stokes equations

Figure 3-13: Wall clock time for linear solver using ILUT on obtuse meshes normalized by the linear solver time using ILU(0) on acute meshes



(a) Factorization time

(b) GMRES time

Figure 3-14: Factorization time and GMRES time using ILUT on obtuse meshes normalized by the corresponding time using ILU(0) on acute meshes

3.5 Summary

In this chapter, the impact of triangle shapes, including large angles and aspect ratios, on accuracy and stiffness for simulations of highly anisotropic problems has been studied.

When using high-order discretizations, whether the elements are obtuse or acute has little effect on accuracy, as both shapes lead to essentially the same error level. In other words, large angles are not problematic if a high-order solution scheme is used. Such a scheme also leads to a lower error per degree of freedom. However, aspect ratio is a key factor for accuracy, for both linear and high-order solutions and for both acute and obtuse elements.

As for the conditioning of the linear system, large angles are found not to be deleterious. The obtuse meshes lead to a similar conditioning as the acute meshes do. However, the solution scheme for these two meshes differ as they involve different strengths of coupling among elements. In particular, ILU(0) works much less efficiently on the obtuse meshes than on the acute ones. However, with an ILUT preconditioner, solutions on the obtuse meshes can be achieved within a comparable time as on the acute meshes.

Taking into account both the solution accuracy and solver efficiency, it can be concluded that for anisotropic meshing with high-order discretizations, while aspect ratio is critical, large angles can be overcome through small increases in temporal costs in preconditioning.

Chapter 4

Output Error Estimation

This chapter details the error estimation and localization approach applied in this work. The approach is based on the work of Fidkowski [26]. In turn, Fidkowski draws on extensive previous research by Barth and Larson [4], Becker and Rannacher [6, 7], Giles and Süli [35], Giles and Pierce [34], Hartmann and Houston [38], Lu [49], and Venditti and Darmofal [74, 75].

4.1 Error Estimation

Let $\mathcal{R}_H(\cdot,\cdot)$ be a semi-linear form, such as the weighted residual defined in Eqn. (2.6), and let $\mathbf{u}_H \in \mathcal{V}_H$ be such that

$$\mathcal{R}_H(\mathbf{u}_H, \mathbf{v}_H) = 0, \quad \forall \mathbf{v}_H \in \mathcal{V}_H, \tag{4.1}$$

where $\mathcal{V}_H$ is a finite dimensional function space, such as the space, $\mathcal{V}_H^p$, defined in Eqn. (2.4). Let $\mathbf{u} \in \mathcal{V}$ be the exact solution of the governing PDE of interest, where $\mathcal{V}$ is the continuous space. Then for a general nonlinear output, $\mathcal{J}(\cdot)$, the adjoint, or dual, problem reads: find $\boldsymbol{\psi} \in \mathcal{V}$ such that

$$\bar{\mathcal{R}}_H(\mathbf{u}, \mathbf{u}_H; \mathbf{v}, \boldsymbol{\psi}) = \bar{\mathcal{J}}(\mathbf{u}, \mathbf{u}_H; \mathbf{v}), \quad \forall \mathbf{v} \in \mathcal{W}_H \equiv \mathcal{V}_H + \mathcal{V}, \tag{4.2}$$

where $\bar{\mathcal{R}}_H : \mathcal{W}_H \times \mathcal{W}_H \to \mathbb{R}$ and $\bar{\mathcal{J}} : \mathcal{W}_H \to \mathbb{R}$ are mean-value linearizations given by

$$\begin{aligned} \bar{\mathcal{R}}_H(\mathbf{u}, \mathbf{u}_H; \mathbf{v}, \mathbf{w}) &= \int_0^1 \mathcal{R}'_H[\theta \mathbf{u} + (1-\theta)\mathbf{u}_H](\mathbf{v}, \mathbf{w}) d\theta \\ \bar{\mathcal{J}}(\mathbf{u}, \mathbf{u}_H; \mathbf{v}) &= \int_0^1 \mathcal{J}'[\theta \mathbf{u} + (1-\theta)\mathbf{u}_H](\mathbf{v}) d\theta, \end{aligned} \tag{4.3}$$

where the primed notation denotes the Frechét derivative. Then for $\mathbf{v} = \mathbf{u} - \mathbf{u}_H$, Eqn. (4.3) gives

$$\begin{aligned} \bar{\mathcal{R}}_H(\mathbf{u}, \mathbf{u}_H; \mathbf{v}, \mathbf{w}) &= \mathcal{R}_H(\mathbf{u}, \mathbf{w}) - \mathcal{R}_H(\mathbf{u}_H, \mathbf{w}) \\ \bar{\mathcal{J}}(\mathbf{u}, \mathbf{u}_H; \mathbf{v}) &= \mathcal{J}(\mathbf{u}) - \mathcal{J}(\mathbf{u}_H). \end{aligned}$$

Assuming consistency, in that the exact solution $\mathbf{u} \in \mathcal{V}$ satisfies $\mathcal{R}_H(\mathbf{u}, \mathbf{w}) = 0$, $\forall \mathbf{w} \in \mathcal{W}_H$, and using the Galerkin orthogonality as in Eqn. (4.1), the output error can be written as

$$\mathcal{J}(\mathbf{u}) - \mathcal{J}(\mathbf{u}_H) = -\mathcal{R}_H(\mathbf{u}_H, \boldsymbol{\psi} - \boldsymbol{\psi}_H), \tag{4.4}$$

where $\boldsymbol{\psi}_H \in \mathcal{V}_H$ can be arbitrary at this point. By duality, this error can also expressed in terms of the adjoint residuals defined as

$$\bar{\mathcal{R}}^{\psi}_H(\mathbf{u}, \mathbf{u}_H; \mathbf{v}, \mathbf{w}) \equiv \bar{\mathcal{R}}_H(\mathbf{u}, \mathbf{u}_H; \mathbf{v}, \mathbf{w}) - \bar{\mathcal{J}}(\mathbf{u}, \mathbf{u}_H; \mathbf{v}), \quad \forall \mathbf{v}, \mathbf{w} \in \mathcal{W}_H.$$

Then one can show that the output error may be written as

$$\mathcal{J}(\mathbf{u}) - \mathcal{J}(\mathbf{u}_H) = -\bar{\mathcal{R}}^{\psi}_H(\mathbf{u}, \mathbf{u}_H; \mathbf{u} - \mathbf{u}_H, \boldsymbol{\psi}_H), \tag{4.5}$$

again for any $\boldsymbol{\psi}_H \in \mathcal{V}_H$. The detailed derivations can be found in Fidkowski [26].

4.2 Implementation

To make the calculation of Eqn. (4.4) and (4.5) practical, two approximations are employed. First, the mean-value linearizations are replaced by the approximate linearization about the discrete solution, $\mathbf{u}_H$. To reduce the error due to this approximation, $\boldsymbol{\psi}_H \in \mathcal{V}_H$ is set to the discrete adjoint solution [35], which satisfies

$$\mathcal{R}_H^{\psi}(\mathbf{u}_H; \mathbf{v}_H, \boldsymbol{\psi}_H) \equiv \mathcal{R}'_H[\mathbf{u}_H](\mathbf{v}_H, \boldsymbol{\psi}_H) - \mathcal{J}'_H[\mathbf{u}_H](\mathbf{v}_H) = 0, \quad \forall \mathbf{v}_H \in \mathcal{V}_H.$$

The second approximation consists of replacing the exact solution errors $\mathbf{u} - \mathbf{u}_H$ and $\boldsymbol{\psi} - \boldsymbol{\psi}_H$ by approximations $\widetilde{\mathbf{u}}_H - \mathbf{u}_H$ and $\widetilde{\boldsymbol{\psi}}_H - \boldsymbol{\psi}_H$, where $\widetilde{\mathbf{u}}_H$ and $\widetilde{\boldsymbol{\psi}}_H$ are surrogates for $\mathbf{u}$ and $\boldsymbol{\psi}$, that exist in a richer space, $\widetilde{\mathcal{V}}_H$, than $\mathbf{u}_H$ and $\boldsymbol{\psi}_H$. A common approach of constructing $\widetilde{\mathcal{V}}_H$ consists of adding more degrees of freedom to the space $\mathcal{V}_H$ through refining the mesh and/or increasing the interpolation order. In this work, $\widetilde{\mathcal{V}}_H$ is taken as the space of piecewise polynomials of order $p+1$:

$$\widetilde{\mathcal{V}}_H \equiv \{\mathbf{v} \in \left(L^2(\Omega)\right)^m : \mathbf{v}|_\kappa \in (\mathcal{P}^{p+1}(\kappa))^m, \forall \kappa \in \mathcal{T}_H\}.$$

However, obtaining the $p+1$st order finite element solution is expensive. Fidkowski uses an H^1-patch reconstruction of $\mathbf{u}_H$ and $\boldsymbol{\psi}_H$ to obtain $\widetilde{\mathbf{u}}_H$ and $\widetilde{\boldsymbol{\psi}}_H$ [26]. Another approach is to inject $\mathbf{u}_H$ and $\boldsymbol{\psi}_H$ into $\widetilde{\mathcal{V}}_H$ and take a small number of element-block Jacobi iterations on the $p+1$st order discrete problem [56]. Numerical tests suggest that this approach gives better approximations to the true $p+1$st order discrete solutions than using the patch reconstructions [2]. In this work, a small number of element-block Gauss-Seidel iterations are taken on the $p+1$st order discrete problem to obtain $\widetilde{\mathbf{u}}_H$ and $\widetilde{\boldsymbol{\psi}}_H$.

Given the surrogate solutions $\widetilde{\mathbf{u}}_H$ and $\widetilde{\boldsymbol{\psi}}_H$, the output error can be approximated using either the primal residual or the dual residual:

$$|\mathcal{J}(\mathbf{u}) - \mathcal{J}(\mathbf{u}_H)| \approx \epsilon_{prim} \equiv |\widetilde{\mathcal{R}}_H(\mathbf{u}_H, \widetilde{\boldsymbol{\psi}}_H - \boldsymbol{\psi}_H)|, \tag{4.6}$$

$$|\mathcal{J}(\mathbf{u}) - \mathcal{J}(\mathbf{u}_H)| \approx \epsilon_{dual} \equiv |\widetilde{\mathcal{R}}^{\psi}_H(\mathbf{u}_H; \widetilde{\mathbf{u}}_H - \mathbf{u}_H, \boldsymbol{\psi}_H)|, \tag{4.7}$$

where $\widetilde{\mathcal{R}}_H$ and $\widetilde{\mathcal{R}}^{\psi}_H$ denote the primal and dual residuals, respectively, on the space $\widetilde{\mathcal{V}}_H$.

The elemental error indicator is constructed by averaging the primal and dual residual expressions of the error,

$$\epsilon_\kappa = \frac{1}{2}\left(|\widetilde{\mathcal{R}}_H(\mathbf{u}_H, (\widetilde{\boldsymbol{\psi}}_H - \boldsymbol{\psi}_H)|_\kappa)| + |\widetilde{\mathcal{R}}^{\psi}_H(\mathbf{u}_H; (\widetilde{\mathbf{u}}_H - \mathbf{u}_H)|_\kappa, \boldsymbol{\psi}_H)|\right), \tag{4.8}$$

where $|_\kappa$ indicates restriction on the element, κ. The global output error is defined to be the sum of all elemental errors,

$$\epsilon \equiv \sum_\kappa \epsilon_\kappa. \tag{4.9}$$

Chapter 5

Direct Adaptation for Output Error

This chapter presents an adaptive method that directly controls the output error without assuming solution regularity or estimating solution derivatives. The chapter begins with motivating the direct adaptation approach in Section 5.1. Section 5.2 then presents the mesh optimality criteria considered, while Sections 5.3 and 5.4 describe the mesh operators used in this work and the criteria for applying them, respectively. Section 5.5 motivates the use of mesh smoothing, and presents the smoothing mechanism applied. The overall adaptation mechanism is developed in Section 5.6, and the adaptation results are shown in Section 5.7. An isotropic version of the direct adaptation together with its results is presented in Appendix B.

5.1 Motivation

Many of the current adaptation strategies rely on a local error estimate that yields element sizing at each adaptation iteration [26, 74, 77]. These approaches assume an *a priori* local convergence rate for output error on each element, that is,

$$\epsilon_\kappa \sim \mathcal{O}(h_\kappa^{r_\kappa}), \quad \forall \kappa, \tag{5.1}$$

where r_κ is an assumed convergence rate based on interpolation order and solution regularity. However, the solution regularity is often difficult to predict, and its overestimation can easily lead to oscillations in the adaptation process.

An important ingredient in h-adaptation for aerodynamic applications is the ability to prescribe the anisotropy needed based on the flow solution. The anisotropy is commonly defined through a metric tensor [37, 75]. For linear solutions, the dominant method for detecting anisotropy involves estimating the Hessian of a scalar field (e.g. Mach number). This method is introduced in the work of Peraire *et al.* [62] and applied to output-based adaptation by Venditti and Darmofal [75]. Fidkowski extends this idea to high-order solutions for output-based adaptation [26]. Pagnutti and Ollivier-Gooch provide a similar anisotropy detection method based on the estimation of truncation errors [60]. All these methods involve estimating the derivatives of a selected scalar field. For systems of equations, the choice of scalar field is unclear. Although Mach number has produced satisfactory results for the Navier-Stokes equations in the past, its selection is arbitrary. Castro-Díaz *et al.* take into account the metric ellipse of each and every variable of the PDE system of interest [13]. Further, the anisotropy of the dual solution should also be included for output-based adaptation.

Knowing the limitations of these approaches, an adaptation scheme that is insensitive to solution regularity or interpolation order and that does not depend on convergence rate (Eqn. (5.1)) is preferred. In addition, the anisotropy of both the primal and the dual solutions should be taken into account. Such a method has been explored in the work of Park and has shown superiority in terms of output error per degree of freedom [61]. However, his approach applies all possible mesh operations in a greedy manner, which can be computationally prohibitive and can easily lead to a local optimum.

5.2 Mesh Optimality Criterion

For any error estimator being used, the goal of adaptation is to achieve a specified error level, ϵ_0, at a minimum computational cost, or with the least number of elements for a given interpolation order. Different criteria have been proposed including the equidistribution of elemental error [46, 77] and the equidistribution of error density [58]. Díez and Huerta provide a comparison among different criteria [17].

For the criterion of equidistribution of error density, a pointwise target error is defined by

$$\bar{\epsilon} = \frac{\epsilon_0}{A_\Omega}, \tag{5.2}$$

where A_Ω is the area of the computation domain. This target is invariant throughout the adaptation process. For the output error estimator defined in Section 4.2, the error density is defined for each individual element by $\epsilon_\kappa / A_\kappa$.

On the other hand, equidistributing elemental error involves defining a target error for each element given by

$$\bar{\epsilon} = \frac{\epsilon_0}{N_f}, \tag{5.3}$$

where N_f is the number of elements on the "converged" adapted mesh, and is usually predicted based on an *a priori* error analysis [26, 74]. This approach has been proven to lead to the fewest degrees of freedom for a given ϵ_0 [16]. However, the proof assumes not only the local error convergence as in Eqn. (5.1), but also a uniform convergence rate over the entire computational domain. These assumptions can be rarely met in a practical problem.

Oñate and Bugeda compare the two criteria mentioned, and conclude that the equidistribution of error density leads to a mesh with more and smaller elements in large-error region while the equidistribution of elemental error leads to a relatively smooth distribution of element sizes [57, 58]. Although the former criterion seems to be more suitable for adaptation purposes, it might lead to more elements than necessary around singularities [58].

For the adaptation presented in this chapter, only the criterion of equidistribution of error density is applied. Extension to the equidistribution of elemental error is trivial. Section B.2.4 compares the two criteria for a direct isotropic adaptation, and indicate that both lead to similar results.

5.3 Mesh Operators

The goal of equidistribution of either elemental error or error density is achieved with basic mesh operators, which locally refine or coarsen the mesh based on the local error or error density. The mesh operators considered in this work include edge split and edge collapse.

5.3.1 Edge Split

The edge split operator inserts a single new node at the midpoint on an edge L. The neighboring elements of this edge are split to include this new node. The split elements always have positive areas. This operator is illustrated in Figure 5-1.

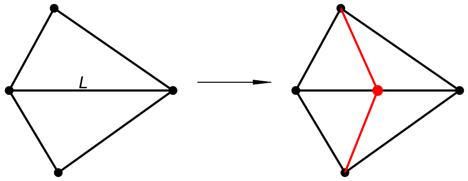


Figure 5-1: Edge split. A new node is inserted at the midpoint of edge L, whose neighboring elements are split.

5.3.2 Edge Collapse

The edge collapse operator removes the neighboring elements of an edge L. One of the two nodes on L is removed, and the remaining elements incident to the removed node are reconnected to the remaining node. This is illustrated in Figure 5-2. Inverted or

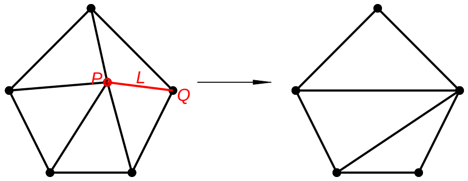


Figure 5-2: Edge collapse. Node P and the elements neighboring edge L are removed. The remaining elements incident to P are reconnected to node Q.

degenerate elements might be produced if the triangles adjacent to the removed node form a concave polygon. Thus, edge collapse is only permitted if the new configuration has positive areas.

5.4 Criteria for Mesh Operations

The edge split and edge collapse described in Section 5.3 are applied to achieve the criterion of equidistribution of error density. Anisotropy is introduced through evaluation of the error change by potential split of an edge.

5.4.1 Edge Split

The decision of splitting an edge consists of two stages. First, all elements are examined for refinement. A refinement parameter is defined for each element by

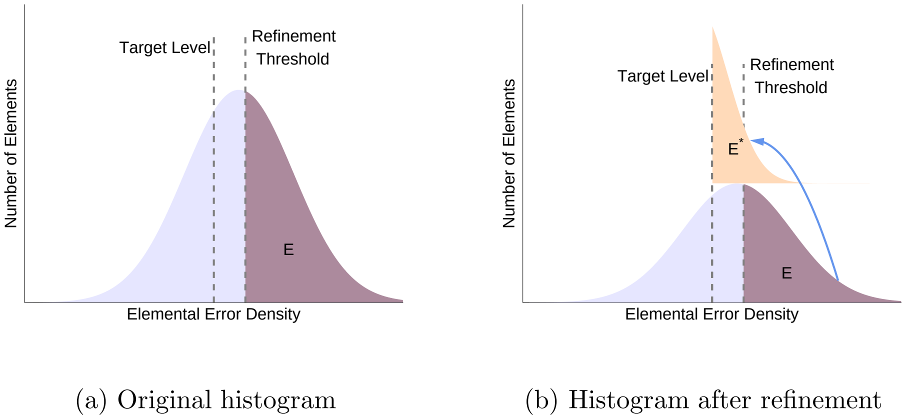
$$\eta_\kappa \equiv \frac{\epsilon_\kappa / A_\kappa}{\bar{\epsilon}},$$

where $\bar{\epsilon}$ is the target level defined in Eqn. (5.2). An element is flagged for refinement if

$$\eta_\kappa \geq \beta, \tag{5.4}$$

where $\beta \geq 1$ is a specified refinement threshold, which controls the aggressiveness of refinement. Setting it to unity means all elements that exceed the target level are refined. In this work, the threshold is fixed at a constant throughout an adaptation process. Nemec and Aftosmis also employ a similar approach in their output-based adaptation and use a decreasing threshold scheme [52]. Although this scheme leads to fewer element counts in early adaptation iterations, the choice of a good threshold decreasing strategy is essentially a matter of tuning, and is not investigated in this work.

The effect of refining the flagged elements can be easily visualized on a histogram of the distribution of elemental error density. This is illustrated in Figure 5-3. Exact equidistribution would mean a single-bin function on the histogram with a height equal to the total number of elements.



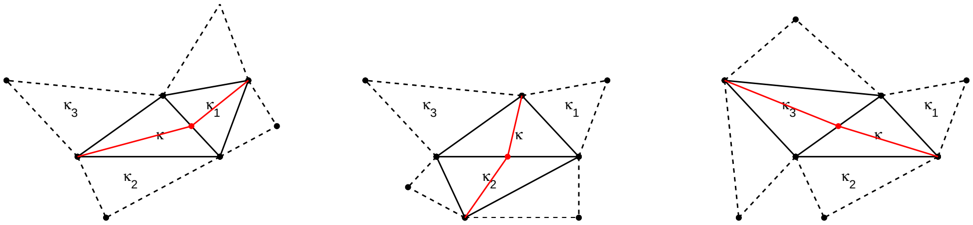
(a) Original histogram

(b) Histogram after refinement

Figure 5-3: Migration of error distribution due to local refinement. Elements in the region E are refined, and their error moves toward the target level.

After elements are flagged for refinement, the flagged elements are sorted with descending η_κ, and examined for edge split in order. For each of these elements, κ, an edge split is performed based on choosing the most competitive subdivision of κ. This is done by evaluating the error change due to trial split of each edge of the element

κ. More specifically, let L_i, $i = 1, 2, 3$, be the i-th edge of κ and κ_i be the neighbor across L_i. A trial split is performed for each L_i, and after the split, let $\mathcal{T}_h^{i'}$ denote the patch of the two sub-elements embedded in κ and the two sub-elements in κ_i, and let $\mathcal{T}_h^i$ be the union of $\mathcal{T}_h^{i'}$ and all the neighbors of the elements in $\mathcal{T}_h^{i'}$. The same DG discretization as in Section 2.2 is then applied on the trial split grid $\mathcal{T}_h^i$ to solve for the primal and dual solutions, $\mathbf{u}_h^i$ and $\boldsymbol{\psi}_h^i$, on $\mathcal{T}_h^i$. During the local solve, the states on the elements of $\mathcal{T}_h^i \backslash \mathcal{T}_h^{i'}$ are frozen at the discretized solutions on the original mesh, $\mathbf{u}_H$ and $\boldsymbol{\psi}_H$. This is illustrated in Figure 5-4, which shows the three trial splits of the element κ.



(a) Split of L_1 (b) Split of L_2 (c) Split of L_3

Figure 5-4: Trial split of the three edges of κ. Elements with dashed edges have states frozen at $\mathbf{u}_H$ and $\boldsymbol{\psi}_H$.

Denote the output error indicator on $\mathcal{T}_h^{i'}$ by $\hat{\epsilon}^i$. Each element on $\mathcal{T}_h^{i'}$ has a contribution to the output error given by

$$\hat{\epsilon}_\kappa^i = \frac{1}{2}\left(|\widetilde{\mathcal{R}}_h(\mathbf{u}_h^i, (\widetilde{\boldsymbol{\psi}}_H - \boldsymbol{\psi}_h^i)|_\kappa)| + |\widetilde{\mathcal{R}}_h^\psi(\mathbf{u}_h^i; (\widetilde{\mathbf{u}}_H - \mathbf{u}_h^i)|_\kappa, \boldsymbol{\psi}_h^i)|\right),$$

where $\widetilde{\mathcal{R}}_h$ and $\widetilde{\mathcal{R}}_h^\psi$ denote the residuals evaluated on the split patch, $\mathcal{T}_h^{i'}$, and $\widetilde{\mathbf{u}}_H$ and $\widetilde{\boldsymbol{\psi}}_H$ are the higher-order truth surrogate solutions on the original mesh, which are obtained from the procedure described in Section 4.2. An edge split indicator for the edge L_i is then defined as

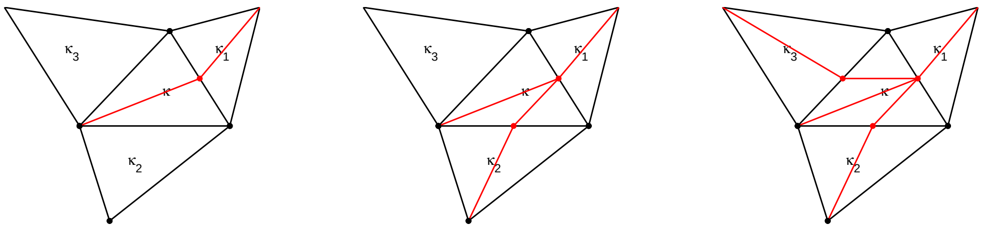
$$\Delta\epsilon^i \equiv \hat{\epsilon}^i - \epsilon^i,$$

where ϵ^i is the original error estimate on the same patch of $\mathcal{T}_h^{i'}$. It would be expected $\Delta\epsilon \leq 0$ if the truth surrogate solutions well represent the $p+1$st order solution.

Without loss of generality, assume $\Delta\epsilon^1 \leq \Delta\epsilon^2 \leq \Delta\epsilon^3 \leq 0$, then the anisotropic edge split algorithm is as following:

1. split the edge L_1;
2. then split the edge L_2 if $\frac{\Delta\epsilon^1}{\Delta\epsilon^2} < \theta_1$, i.e. $\Delta\epsilon^1$ and $\Delta\epsilon^2$ are sufficiently close;
3. if step 2 is performed, then split the edge L_3 if $\frac{\Delta\epsilon^1}{\Delta\epsilon^3} < \theta_2$, i.e. $\Delta\epsilon^1$, $\Delta\epsilon^2$, and $\Delta\epsilon^3$ are sufficiently close.

In the algorithm, θ_1 and θ_2 are specified parameters, and in this work, $\theta_1 = 1.5$ and $\theta_2 = 3$ are chosen. Figure 5-5 shows all the possible splits for one element. Note that uniform refinement of the element κ will be considered in future work if all the three $\Delta\epsilon$'s are sufficiently close.



(a) Split of one edge (b) Split of two edges (c) Split of three edges

Figure 5-5: Possible splits for one element κ

Georgoulis *et al.* employ a similar approach for Cartesian meshes [31, 32]. They also propose to determine the best edge(s) for split by performing local solves on all the possible new configurations and consider the best $\Delta\epsilon$ per DOF added. The two approaches are found to lead to similar results in their work.

5.4.2 Edge Collapse

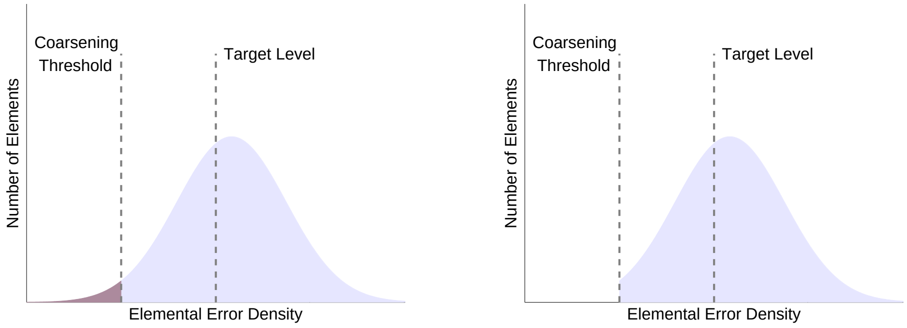
The decision of edge collapse also consists of two stages. First, nodes are examined for removal. The criterion for removing a node is based on node error density, which is defined by

$$\epsilon_P \equiv \frac{\sum_{\kappa \in N(P)} \epsilon_\kappa}{\sum_{\kappa \in N(P)} A_\kappa},$$

where $N(P)$ is the set of elements adjacent to the node P. A node P is flagged for removal if it satisfies the criterion

$$\eta_P \equiv \frac{\epsilon_P}{\bar{\epsilon}} \leq \lambda,$$

where $\lambda \leq 1$ is a specified coarsening threshold. This threshold is held constant through an adaptation process. Figure 5-6 shows the effect of node removal, which alters the lower tail of the error distribution.



(a) Original histogram

(b) Histogram after coarsening

Figure 5-6: Migration of error distribution due to local coarsening. Elements with error lower than the threshold are coarsened.

For each flagged node, edge collapse is applied to an arbitrarily selected edge connected to this node without evaluating the error change. Note that because node removal happens only in regions with very low error, evaluating the error change due to each edge collapse might not be worth the computational effort.

5.5 Mesh Smoothing Procedure

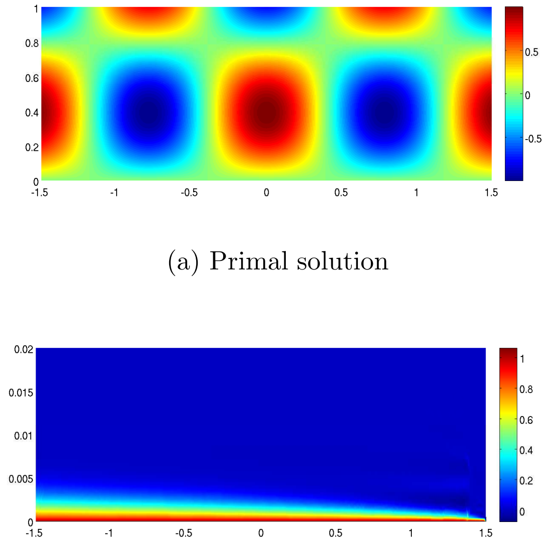
To motivate the need of mesh smoothing in addition to edge split and collapse, an advection-diffusion example is used. The example has a diffusivity of $\mu = 10^{-6}$ and a uniform and horizontal convection velocity with a unit magnitude. The computational domain is a rectangular box of $[-1.5, 1.5] \times [0, 1]$. A source term is added such that the exact solution has the form of

$$u(x, y) = \cos(4x)\sin(4y).$$

Dirichlet boundary conditions are applied on the left, top, and bottom walls, and Neumann boundary condition on the right. The output is the heat transfer across the bottom boundary as in Eqn. (3.2).

The dual problem is also an advection-diffusion equation with the same diffusivity but a convection velocity opposite to the primal problem. The lower boundary has a Dirichlet condition of $\psi = 1$, and the upper and left boundaries have $\psi = 0$. Detailed derivations can be found in [36]. Thus, the dual solution is a thin boundary layer along the bottom wall. Figure 5-7 shows the primal and the dual solutions for this example.

An interpolation order of $p = 3$ is applied, and the initial mesh is shown in Figure 5-8. The adaptation parameters are $\beta = 150$ and $\lambda = 0.01$. Figure 5-9 shows the final adapted mesh. It has the correct anisotropy that captures the boundary layer in the dual solution. However, the mesh in the refined region appears irregular as it is essentially embedded in the initial mesh. Figure 5-10 plots the error estimate and the true error normalized by the true output value versus DOF in the adaptation



(a) Primal solution

(b) Dual solution (y-axis rescaled)

Figure 5-7: Primal and dual solutions of the advection-diffusion example for direct adaptation. Anisotropies of the primal and the dual solutions are completely different.

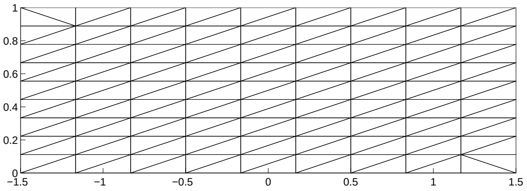
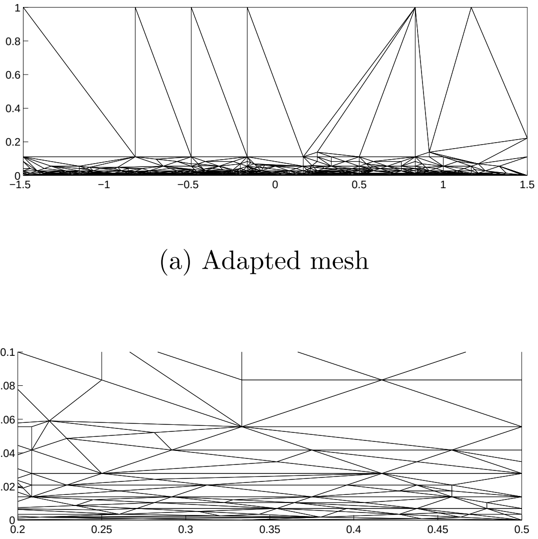


Figure 5-8: Initial mesh for direct adaptation

history. Due to the low quality of the mesh, the true error barely decreases while the error estimate decreases for almost every iteration. To address this issue, a smoothing mechanism is introduced.

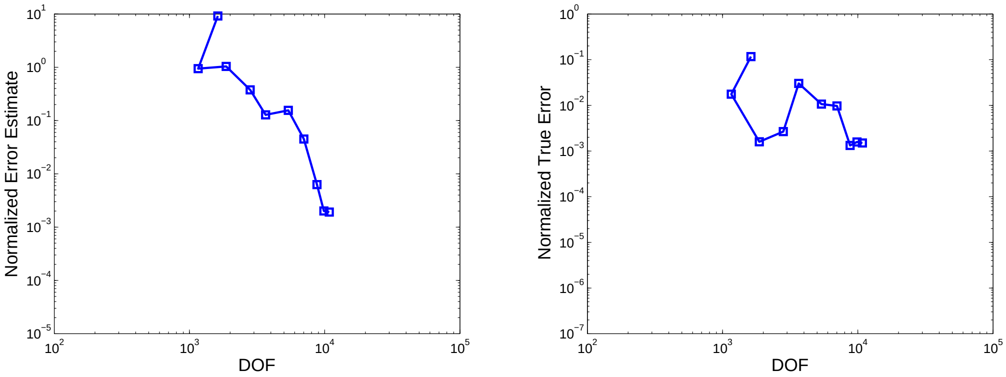
The objective of the smoothing mechanism is to improve the quality of the meshes obtained from the mesh operations without modifying local node density or element anisotropy. In this work, mesh smoothing is performed via the Bi-dimensional Anisotropic Mesh Generator (BAMG), which is a Delaunay-type triangulator taking



(a) Adapted mesh

(b) Adapted mesh (zoom)

Figure 5-9: Adapted mesh from direct adaptation with only mesh operations



(a) Normalized error estimate

(b) Normalized true error

Figure 5-10: Adaptation history of direct adaptation with only mesh operations

as an input metric tensors defined on a background mesh [39]. BAMG is based on constructing an isotropic mesh in the stretched space defined by a desired metric field [9]. For such a method, when the mesh is mapped back to the physical space, large angles can be introduced [41]. However, as shown in Chapter 3, for high-order interpolations, large angles are not problematic for solution accuracy or linear system conditioning.

It should be emphasized that although BAMG is used in this work, other meshing strategies that employ an input metric tensor can also be used, such as an anisotropic centroidal Voronoi tessellations [23], or metric conformity techniques [11, 44].

5.6 Overall Adaptation Mechanism

Denote the mesh obtained from the mesh operations by $\mathcal{T}_H'$. The implied metric of each element on $\mathcal{T}_H'$ is computed and treated as a desired metric. This metric, together with the mesh $\mathcal{T}_H'$, are supplied as an input to BAMG for smoothing. The smoothed mesh should have a similar metric field as $\mathcal{T}_H'$. The detail of one adaptation iteration is described in Algorithm 5.6.1. The adaptation is terminated when the global error is less than the specified target level.

The implied metric is computed using reference-element Jacobian mapping. Let $\mathbf{J}_\kappa$ be the mapping from the reference element to κ, and let $\mathbf{J}_E$ be the mapping from the reference element to the unit equilateral triangle, then the mapping from κ to the unit equilateral triangle is given by $\mathbf{J} = \mathbf{J}_E \mathbf{J}_\kappa{}^{-1}$, and the implied metric is $\mathbf{M} = \mathbf{J}^T\mathbf{J}$. The resulting implied metric is similar to that used in [75].

Algorithm 5.6.1. One iteration of direct adaptation with BAMG smoothing

1. Compute the primal and adjoint solutions, $\mathbf{u}_H$ and $\boldsymbol{\psi}_H$, on the current mesh, $\mathcal{T}_H$, using the solution method presented in Section 2.2.

2. Compute local error indicator from Eqn. (4.8) on $\mathcal{T}_H$.

3. Obtain the mesh $\mathcal{T}_H'$ by applying mesh operations:

- Split edges using the procedure presented in Section 5.4.
- Collapse edges as in Section 5.4.

4. Compute the implied metric $\mathbf{M}$ on $\mathcal{T}'_H$.

5. Obtain the adapted mesh using BAMG with $\mathcal{T}'_H$ and $\mathbf{M}$ as input.

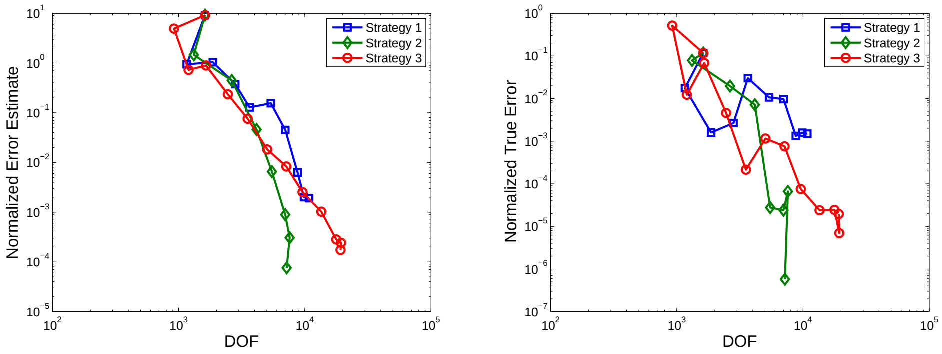
5.7 Numerical Results

In this section, the adaptation algorithm is applied to two advection-diffusion examples to directly control the error of an output. The first example tests the capability of capturing the anisotropy in the dual solution, and the second examines the behavior of the proposed adaptation scheme when the underlying solution is not sufficiently regular. For both cases, only high-order interpolation ($p = 3$) is used because it is not susceptible to large angles that might be introduced from smoothing with the anisotropic Delaunay triangulation. The initial mesh is shown in Figure 5-8, and the adaptation parameters are $\beta = 150$ and $\lambda = 0.01$.

5.7.1 Thin Boundary Layer in Dual Solution

This case is the same as in Section 5.5. Figure 5-11 shows the adaptation results from the direct adaptation with and without BAMG smoothing. It also plots the results using the existing metric-based approach which computes size-request based on an *a priori* assumption and detects anisotropy from primal solution derivatives [26]. The direct adaptation with BAMG smoothing leads to the fewest DOF on the final adapted mesh.

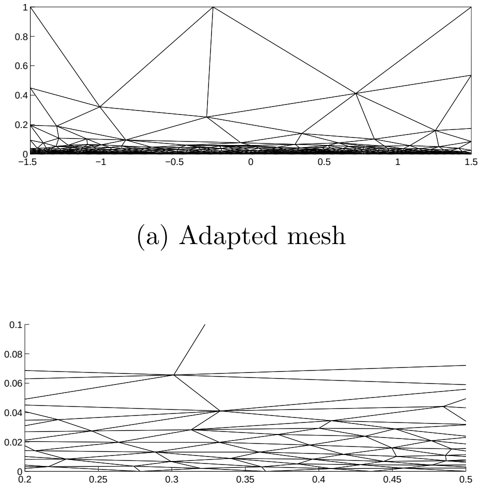
Figures 5-12 and 5-13 show the final adapted meshes from the proposed direct adaptation and the metric-based approach presented in [26], respectively. As shown, the direct adaptation captures the anisotropy in the dual solution, which has much larger stretching in x- than in y-direction. In other words, the proposed direct adaptation scheme implicitly identifies the anisotropy of the primal and dual solution depending on their impact on the output of interest. In contrast, the metric-based approach gives an isotropic mesh since the metric calculation is solely based on the primal solution.



(a) Normalized error estimate

(b) Normalized true error

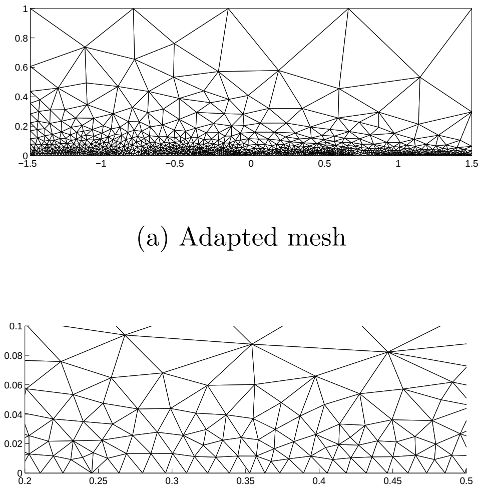
Figure 5-11: Adaptation history of direct adaptation. Strategy 1: direct adaptation, without smoothing. Strategy 2: direct adaptation, with BAMG smoothing. Strategy 3: size-request from *a priori* assumption, anisotropy from primal solution derivatives.



(a) Adapted mesh

(b) Adapted mesh (zoom)

Figure 5-12: Adapted meshes from direct adaptation with BAMG smoothing



(a) Adapted mesh

(b) Adapted mesh (zoom)

Figure 5-13: Adapted meshes from the approach with the *a priori* size-request and anisotropy detection based on primal solution derivatives

5.7.2 Nonsmooth Primal Solution

This case is the same as the previous one except for the source term, which makes an exact solution of the form

$$u(x, y) = \begin{cases} \cos(8x)\sin(4y), & \text{if } x \leq 0 \\ \cos(4x)\sin(4y), & \text{otherwise.} \end{cases}$$

This solution is shown in Figure 5-14, and has a discontinuous second-derivative at $x = 0$. The dual solution is not affected by the source term, so it is the same as for the first case as shown in Figure 5-7.

Figure 5-15 compares the adaptation results between the proposed direct approach and the existing metric-based approach. For the interpolation order used, the metric-based approach oscillates and fails to reach the normalized target error level, 1.826×10^{-2}. This is due to the overestimation of the solution regularity, and thus

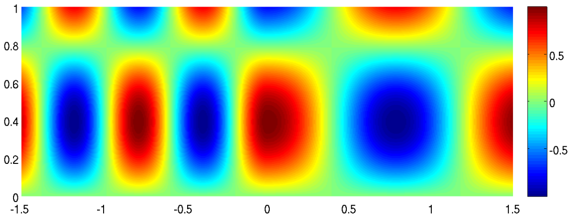
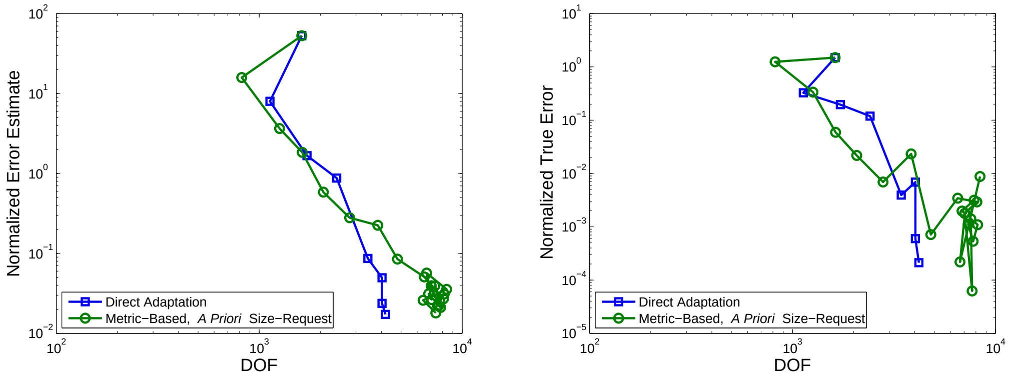


Figure 5-14: Primal solutions of the nonsmooth advection-diffusion example for direct adaptation

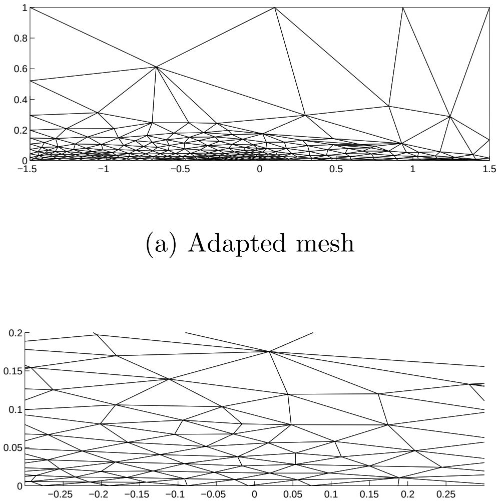
the convergence rate of the output error. In contrast, the proposed direct approach is insensitive to the solution regularity. Figures 5-16 and 5-17 show the adapted meshes from these two approaches. Again, the direct approach captures the anisotropy in the dual solution while the metric-based approach does not. Furthermore, as shown in Figure 5-17, the adapted mesh from the metric-based approach exhibits different anisotropies on the two sides of $x = 0$ as the primal solutions behave differently on those sides.



(a) Normalized error estimate

(b) Normalized true error

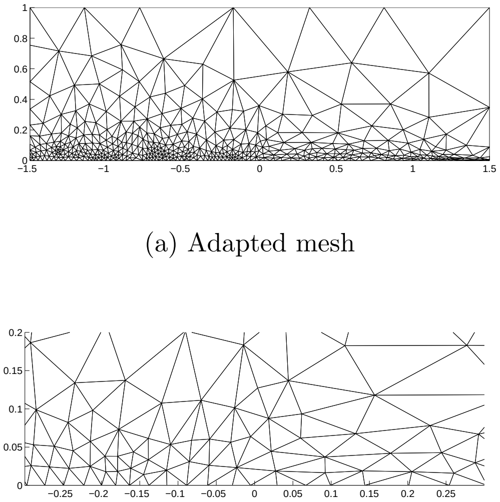
Figure 5-15: Adaptation history of direct adaptation with the BAMG smoothing



(a) Adapted mesh

(b) Adapted mesh (zoom)

Figure 5-16: Adapted meshes from direct adaptation with the BAMG smoothing



(a) Adapted mesh

(b) Adapted mesh (zoom)

Figure 5-17: Adapted meshes from the approach with the *a priori* size-request and anisotropy detection based on primal solution derivatives

Chapter 6

Conclusion

The impact of large angles and element aspect ratios has been studied for boundary layer problems. With a high-order DG discretization, it is seen that the discretization error is not sensitive to the maximum angles involved, but using a correct aspect ratio is critical. In addition, the linear system induced from obtuse meshes has similar conditioning with that from acute meshes. However, the solution scheme for these two meshes differ as they involve different strengths of coupling among elements. For this reason, ILU(0) has been shown less efficient on obtuse meshes. With an ILUT preconditioner, the solution on obtuse meshes can be achieved within a comparable time as on acute meshes. Therefore, for anisotropic meshing with high-order discretizations, while aspect ratio is critical, large angles can be overcome through small increases in preconditioning costs. More preconditioning strategies for obtuse meshes should be considered as ILUT is memory expensive.

Given an output error estimate, a direct adaptation scheme that controls the output error via mesh operations has been developed. Mesh operations include edge split and edge collapse. The decision of these operations is solely based on local output error distribution without any *a priori* assumption on error convergence rate. Anisotropy is introduced by evaluating the error change due to a potential split. An anisotropic Delaunay triangulation is introduced as a smoother via BAMG.

The proposed scheme has been shown to be insensitive to solution regularity and to take into account the anisotropy of the dual solution. The scheme also leads to fewer degrees of freedom for a specified error level than the existing metric-based adaptation.

Appendix A

Coupling Weights in Jacobian Matrix

This appendix derives the scaling of the coupling weights in the Jacobian matrix arising from the DG discretization presented in Chapter 2. A uniform mesh is assumed, and the derivation is for one interior element, κ, with its neighbor, κ', as shown in Figure A-1. The derivation for other neighbors are similar. Denote the face between κ and κ' by σ_f, then its normal is $\hat{\mathbf{n}} = [1, 0]^T$. A pure diffusion problem with a diffusivity tensor $\bar{\bar{\mu}}$ is considered.

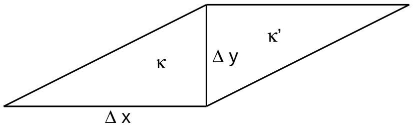


Figure A-1: Example element for deriving the scaling of coupling weights in Jacobian matrix

The residual for the problem considered has only the discretization of the viscous term, which applies the same scheme as in [27]. For the element κ, the residual is given by

$$\mathcal{R}_{H\kappa}(\mathbf{u}_H, \mathbf{v}_H) = \underbrace{\int_{\kappa} \nabla \mathbf{v}_H^T \cdot \bar{\bar{\mu}} \nabla \mathbf{u}_H d\mathbf{x}}_{\text{Term } I} + \underbrace{\int_{\partial\kappa} (\nabla \mathbf{v}_H^T)^+ \left(\widehat{\bar{\bar{\mu}}\mathbf{u}} - \bar{\bar{\mu}}^+ \mathbf{u}_H^+ \right) \cdot \hat{\mathbf{n}} ds}_{\text{Term } II} - \underbrace{\int_{\partial\kappa} \mathbf{v}_H^+ \hat{\mathcal{Q}} \cdot \hat{\mathbf{n}} ds}_{\text{Term } III}, \tag{A.1}$$

where $\widehat{\bar{\bar{\mu}}\mathbf{u}} = \bar{\bar{\mu}}^+ \{\mathbf{u}_H\}$, $\hat{\mathcal{Q}} = \{\bar{\bar{\mu}} \nabla \mathbf{u}_H\} - \eta_f \{\boldsymbol{\delta}_f\}$, and η_f is a specified stabilization parameter. The notation $()^+$ and $()^-$ refers to data on the interior and exterior of an element boundary. The auxiliary variables $\boldsymbol{\delta}_f \in (\mathcal{V}_H^p)^d$ are defined such that

$$\int_{\kappa} \boldsymbol{\tau}_H^T \cdot \boldsymbol{\delta}_{f\kappa} d\mathbf{x} + \int_{\kappa'} \boldsymbol{\tau}_H^T \cdot \boldsymbol{\delta}_{f\kappa'} d\mathbf{x} = \int_{\sigma_f} [\![\mathbf{u}_H]\!]^T \cdot \{\bar{\bar{\mu}} \boldsymbol{\tau}_H\} ds, \quad \forall \boldsymbol{\tau}_H \in (\mathcal{V}_H^p)^d. \tag{A.2}$$

The operators $\{\cdot\}$ and $[\![\cdot]\!]$ denote average and jump operators, respectively.

Following the notations in Section 2.2, the discrete residual vector is given by $\mathbf{R}_H(\mathbf{U}_H)_\kappa = \mathcal{R}_H(\mathbf{u}_H, \boldsymbol{\phi}_\kappa)$ for the element κ, where $\boldsymbol{\phi}_\kappa$ is the set of all basis functions with support in κ, and $\mathbf{u}_H(\mathbf{x}) = \mathbf{U}_H^T \boldsymbol{\phi}_\kappa$. The Jacobian matrix is $\mathbf{K} = \frac{\partial \mathbf{R}_H}{\partial \mathbf{U}_H}$, which involves differentiating each term in Eqn. (A.1) with respect to $\mathbf{U}_H$. To derive the scaling of these terms, assume for each basis function $\boldsymbol{\phi}$ that

$$\begin{aligned} \boldsymbol{\phi} &\sim \mathcal{O}(1), \\ \nabla \boldsymbol{\phi} &\sim \mathcal{O}\left(\left[\frac{1}{\Delta x}, \frac{1}{\Delta y} \right]^T \right). \end{aligned} \tag{A.3}$$

A.1 Isotropic Diffusivity

Consider an isotropic diffusion problem, where the diffusivity tensor $\bar{\bar{\mu}}$ can be simplified to a scalar μ. By assuming Eqn. (A.3), and knowing the normal vector is $\hat{\mathbf{n}} = (1, 0)^T$, the scaling of each term in Eqn. (A.1) is given by

$$\begin{aligned} \text{Term } I &\sim \mathcal{O}\left(\mu\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}\right)A|\mathbf{U}_H|\right) \\ \text{Term } II &\sim \mathcal{O}\left(\mu\frac{1}{\Delta x}\Delta y|\mathbf{U}_H|\right) \\ \text{Term } III &\sim \mathcal{O}\left(\mu\left(\frac{1}{\Delta x}|\mathbf{U}_H|+\boldsymbol{\delta}_f\cdot\hat{\mathbf{n}}\right)\Delta y\right), \end{aligned} \tag{A.4}$$

where $A=\frac{1}{2}\Delta x\Delta y$ is the area of κ, and $|\mathbf{U}_H|$ is a certain measure of the solution vector. From Eqn. (A.2), one can derive the scaling of $\boldsymbol{\delta}_f$ to be $\boldsymbol{\delta}_f\sim\mathcal{O}([\frac{\Delta y}{A}|\mathbf{U}_H|,0]^T)$. Thus, Eqn. (A.4) can be simplified to

$$\begin{aligned} \text{Term } I &\sim \mathcal{O}\left(\mu\left(\frac{\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y}\right)|\mathbf{U}_H|\right) \\ \text{Term } II &\sim \mathcal{O}\left(\mu\frac{\Delta y}{\Delta x}|\mathbf{U}_H|\right) \\ \text{Term } III &\sim \mathcal{O}\left(\mu\frac{\Delta y}{\Delta x}|\mathbf{U}_H|\right). \end{aligned}$$

For the block row corresponding to κ in the matrix $\mathbf{K}$, the diagonal block $K_{\kappa\kappa}$ involves the derivatives of all the three terms, while the off-diagonal block $K_{\kappa\kappa'}$ has the derivatives of the terms II and III. Thus, with the blocks measured in a certain norm, the coupling weight of κ with κ' scales with

$$\frac{K_{\kappa\kappa'}}{K_{\kappa\kappa}}\sim\mathcal{O}\left(\frac{\frac{\Delta y}{\Delta x}}{\frac{\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y}}\right)\sim\mathcal{O}\left(\frac{\Delta y^2}{\Delta x^2+\Delta y^2}\right). \tag{A.5}$$

In other words, the coupling scales with $\mathcal{O}\left(l_f/d_\kappa\right)^2$, where l_f is the length of the face between κ and κ', and d_κ is a measure of the diameter of the element κ.

A.2 Anisotropic Diffusivity

Consider a symmetric positive definite tensor, $\bar{\bar{\mu}}$. Denote its entries by μ_{ij}, $i,j=1,2$, where $\mu_{12}=\mu_{21}$. The derivation in the previous section can be easily extended for such a diffusivity tensor. The scaling of each term in Eqn. (A.1) can be shown to be

$$\begin{aligned} \text{Term } I &\sim \mathcal{O}\left(\left(\mu_{11}\frac{\Delta y}{\Delta x} + 2\mu_{12} + \mu_{22}\frac{\Delta x}{\Delta y}\right)|\mathbf{U}_H|\right) \\ \text{Term } II &\sim \mathcal{O}\left(\left(\mu_{11}\frac{\Delta y}{\Delta x} + \mu_{12}\right)|\mathbf{U}_H|\right) \\ \text{Term } III &\sim \mathcal{O}\left(\left(\mu_{11}\frac{\Delta y}{\Delta x} + \mu_{12}\right)|\mathbf{U}_H|\right). \end{aligned}$$

Thus, the coupling weight of κ with κ' scales with

$$\frac{K_{\kappa\kappa'}}{K_{\kappa\kappa}} \sim \mathcal{O}\left(\frac{\mu_{11}\Delta y^2 + \mu_{12}\Delta x\Delta y}{\mu_{11}\Delta y^2 + \mu_{22}\Delta x^2 + \mu_{12}\Delta x\Delta y}\right). \tag{A.6}$$

This can be reduced to the scaling for isotropic diffusivity as in Eqn. (A.5) when $\mu_{11} = \mu_{22} = \mu$ and $\mu_{12} = 0$. Shewchuk has shown that when the element κ is mapped to a space with an isotropic diffusivity, the stiffness matrix is unchanged for a standard Galerkin discretization [72]. However, for the DG discretization presented, the coupling weight in Eqn. (A.6) is not the same as that when the element κ is mapped to a space with an isotropic diffusivity.

Consider a mapping matrix F such that $F^2 = \bar{\bar{\mu}}^{-1}$. This mapping transforms the physical space $\mathcal{V}$ to a mapped space $\mathcal{F}$ with an isotropic diffusivity tensor. In the space $\mathcal{F}$, the residual in Eqn. (A.1) becomes

$$\mathcal{R}_{H\kappa}(\mathbf{u}_H, \mathbf{v}_H) = \underbrace{\int_{\tilde{\kappa}} \widetilde{\nabla}\mathbf{v}_H^T \cdot \widetilde{\nabla}\mathbf{u}_H d\tilde{\mathbf{x}}}_{\text{Term } I} + \underbrace{\int_{\partial\tilde{\kappa}} (\widetilde{\nabla}\mathbf{v}_H^T)^+ \left(\widehat{\mathbf{u}} - \mathbf{u}_H^+\right) \cdot \tilde{\hat{\mathbf{n}}} d\tilde{s}}_{\text{Term } II} - \underbrace{\int_{\partial\kappa} \mathbf{v}_H^+ \tilde{\hat{\mathcal{Q}}} \cdot \tilde{\hat{\mathbf{n}}} d\tilde{s}}_{\text{Term } III}, \tag{A.7}$$

where $\tilde{(\cdot)}$ represents quantities in $\mathcal{F}$. Also, a vector $\mathbf{v}$ in $\mathcal{V}$ becomes $F\mathbf{v}$ in $\mathcal{F}$, with its length becoming $\sqrt{\mathbf{v}^T F^2 \mathbf{v}} = \sqrt{\mathbf{v}^T \bar{\bar{\mu}}^{-1} \mathbf{v}}$, and its normal vector in $\mathcal{F}$ is given by $\tilde{\hat{\mathbf{n}}} = F^{-1}\hat{\mathbf{n}}$ because $(F\mathbf{v}) \cdot (F^{-1}\hat{\mathbf{n}}) = 0$, where $\hat{\mathbf{n}}$ is the normal vector of $\mathbf{v}$ in $\mathcal{V}$. Also, the area of the element κ in $\mathcal{F}$ becomes $\det(F)A$, and the gradient of a function in $\mathcal{F}$

is given by $\widetilde{\nabla}(\cdot) = F^{-1}\nabla(\cdot)$. Thus, the terms in Eqn. (A.7) scale as

$$\begin{aligned}
\text{Term } I &\sim \mathcal{O}\left(\left[\frac{1}{\Delta x}, \frac{1}{\Delta y}\right] \bar{\bar{\mu}} \left[\frac{1}{\Delta x}, \frac{1}{\Delta y}\right]^T \det(F) A|\mathbf{U}_H|\right) \\
\text{Term } II &\sim \mathcal{O}\left(\left[\frac{1}{\Delta x}, \frac{1}{\Delta y}\right] \bar{\bar{\mu}}\hat{\mathbf{n}} \cdot \sqrt{[0, \Delta y]\, \bar{\bar{\mu}}^{-1}\, [0, \Delta y]^T} |\mathbf{U}_H|\right),
\end{aligned} \tag{A.8}$$

and the term III scales similarly as the term II. Note that $\bar{\bar{\mu}}^{-1}$ is given by

$$\bar{\bar{\mu}}^{-1} = \frac{1}{\det(\bar{\bar{\mu}})} \begin{bmatrix} \mu_{22} & -\mu_{12} \\ -\mu_{12} & \mu_{11} \end{bmatrix} = (\det(F))^2 \begin{bmatrix} \mu_{22} & -\mu_{12} \\ -\mu_{12} & \mu_{11} \end{bmatrix}$$

Thus, with $\hat{\mathbf{n}} = [1, 0]^T$, Eqn. (A.8) can be simplified to

$$\begin{aligned}
\text{Term } I &\sim \mathcal{O}\left(\left(\mu_{11}\frac{\Delta y}{\Delta x} + 2\mu_{12} + \mu_{22}\frac{\Delta x}{\Delta y}\right) \det(F)|\mathbf{U}_H|\right) \\
\text{Term } II &\sim \mathcal{O}\left(\left(\mu_{11}\frac{\Delta y}{\Delta x} + \mu_{12}\right) \sqrt{\mu_{11}} \det(F)|\mathbf{U}_H|\right)
\end{aligned}$$

Thus, the coupling weight of κ with κ' in the mapped space $\mathcal{F}$ scales with

$$\frac{K_{\kappa\kappa'}}{K_{\kappa\kappa}} \sim \mathcal{O}\left(\sqrt{\mu_{11}} \frac{\mu_{11}\Delta y^2 + \mu_{12}\Delta x \Delta y}{\mu_{11}\Delta y^2 + \mu_{22}\Delta x^2 + \mu_{12}\Delta x \Delta y}\right), \tag{A.9}$$

which is different than Eqn. (A.6).

Appendix B

Isotropic Direct Adaptation for Output Error

B.1 Adaptation with Mesh Operations

For isotropic adaptation, mesh operations described in Section 5.3 are simplified to node operations without defining the mesh topology. In particular, edge split is simplified to node insertion at the midpoint of an edge, and edge collapse becomes removal of a node. In this work, the topology after node operations is constructed using the Delaunay triangulation.

B.1.1 Criteria for Mesh Operations

Node Insertion

The criterion for splitting an edge (i.e. inserting a node at its midpoint in the isotropic context) is based on the edge error, whose definition depends on the mesh optimality criterion applied. For the equidistribution of error, the edge error for an edge L is defined as

$$\epsilon_L \equiv \begin{cases} \frac{\epsilon_{\kappa_1}+\epsilon_{\kappa_2}}{A_{\kappa_1}+A_{\kappa_2}}, & \text{for interior } L \\ \frac{\epsilon_{\kappa_1}}{A_{\kappa_1}}, & \text{for boundary } L \end{cases},$$

and for the equidistribution of error, the edge error is

$$\epsilon_L \equiv \begin{cases} \frac{\epsilon_{\kappa_1}+\epsilon_{\kappa_2}}{2}, & \text{for interior } L \\ \epsilon_{\kappa_1}, & \text{for boundary } L \end{cases},$$

where κ_1 and κ_2 are the neighboring elements of the edge L, and ϵ is the local output error indicator given in Eqn. (4.8).

A split parameter, η_L, is then defined for each edge as

$$\eta_L \equiv \frac{\epsilon_L}{\bar{\epsilon}},$$

where $\bar{\epsilon}$ is the target error level defined in Eqn. (5.2) or (5.3) depending on the optimality criterion chosen. An edge is flagged for split if

$$\eta_L \geq \beta, \tag{B.1}$$

where $\beta \geq 1$ is a global refinement threshold, which controls the aggressiveness of refinement. Setting it to unity means all edges that exceed the target level are split. The threshold is constant through an adaptation process.

Node Removal

The criterion for edge collapse (i.e. node removal in the isotropic setting) is based on node error, which is defined as

$$\epsilon_P \equiv \frac{\sum_{\kappa \in N(P)} \epsilon_\kappa}{\sum_{\kappa \in N(P)} A_\kappa}, \text{ for equidistribution of error density,}$$

and

$$\epsilon_P \equiv \frac{\sum_{\kappa \in N(P)} \epsilon_\kappa}{|N(P)|}, \text{ for equidistribution of elemental error,}$$

where $N(P)$ is the set of elements adjacent to the node P.

A node is removed if its error satisfies

$$\eta_P \equiv \frac{\epsilon_P}{\bar{\epsilon}} \leq \lambda, \tag{B.2}$$

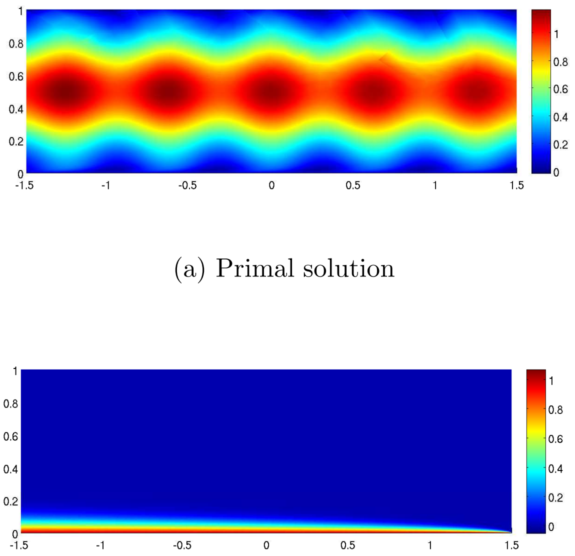
where $\lambda \leq 1$ is a specified coarsening threshold. This threshold is held constant through an adaptation process.

B.1.2 Numerical Results

The described procedure is applied to directly control the output error of a scalar advection-diffusion example on a computational domain of $[-1.5, 1.5] \times [0, 1]$. The test case is the same as Park's for his direct adaptation [61]. The output in interest is the heat along the bottom wall. Figure B-1 shows the primal and the dual solutions on a very fine mesh with an interpolation order of $p = 3$. The reference output value, 0.0080361571338, is computed from this mesh. The primal solution is mainly isotropic but has a larger stretching in x- than in y-direction near the bottom wall while the dual solution is a boundary layer along the bottom.

The optimality criterion applied is the equidistribution of error density, and the adaptation parameters, i.e. the refinement and coarsening thresholds, are set to $\beta = 5$ and $\lambda = 0.01$. Four initial meshes are used as shown in Figure B-2, and a linear interpolation is used.

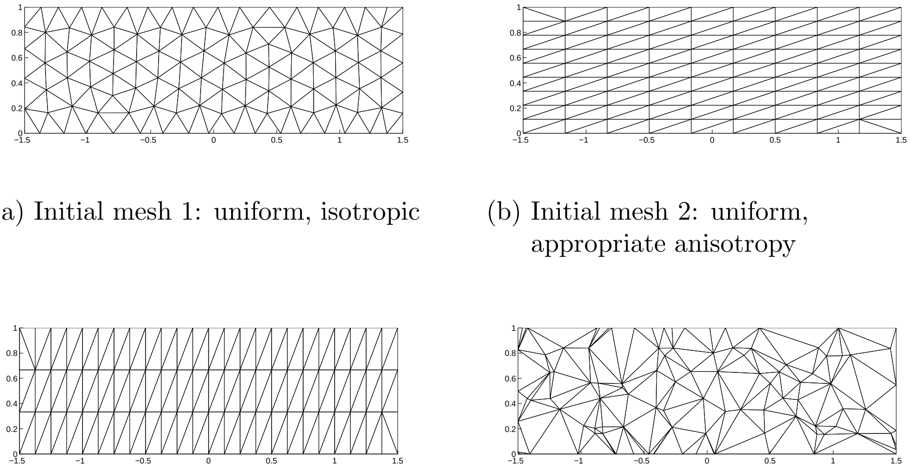
Figure B-3 shows the adaptation history for the initial meshes considered. In particular, the actual output error is plotted versus DOF for each adaptation iteration. Although the output error converges at the same rate for all initial meshes, the actual error value varies drastically on the adapted meshes from initial meshes of different



(a) Primal solution

(b) Dual solution

Figure B-1: Primal and dual solution of the advection-diffusion example for isotropic adaptation



(a) Initial mesh 1: uniform, isotropic

(b) Initial mesh 2: uniform, appropriate anisotropy

(c) Initial mesh 3: uniform, inappropriate anisotropy

(d) Initial mesh 4: random

Figure B-2: Initial meshes for isotropic adaptation anisotropies. To further illustrate, Figure B-4 shows the adapted meshes from the initial meshes 2 and 3, and they have similar relative node densities. However, a closer look at these meshes reveals that the anisotropy on the initial mesh clearly remains on the adapted mesh as shown in Figure B-5. The adapted meshes from the initial meshes 1 and 4 are not shown because they exhibit the same behavior, that is, the initial anisotropy remains through the adaptation process. A mesh smoothing mechanism is presented in the next section to address this issue.

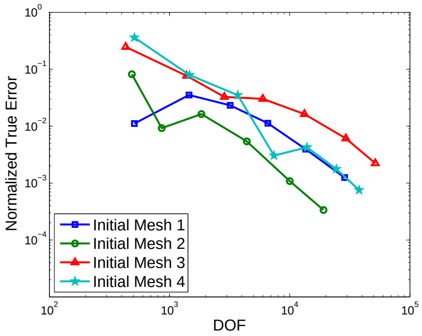
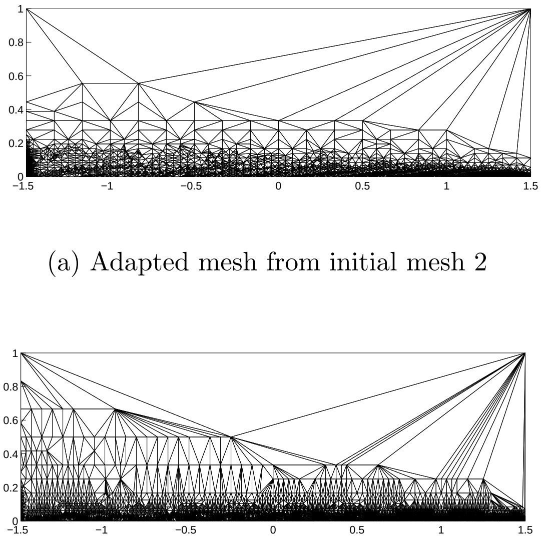


Figure B-3: Actual heat error versus DOF for isotropic direct adaptation

B.2 Centroidal Voronoi Tessellations

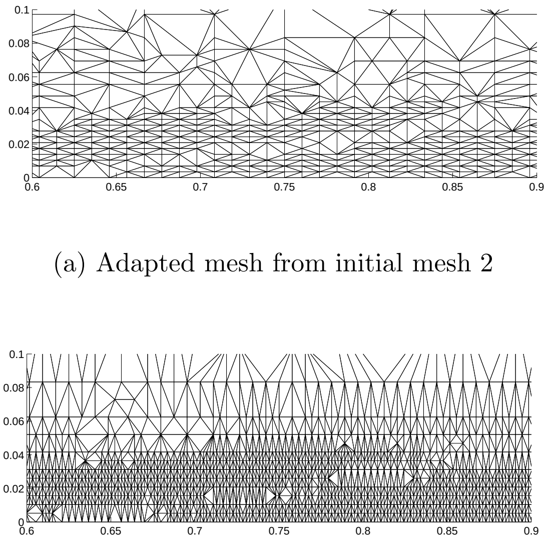
The concept of CVT has been applied in a diverse range of applications [20], but not until recently is it used for quality mesh generation and optimization [21, 22, 76]. It can be characterized as solutions of an optimization problem for an associated energy functional, and its dual centroidal Voronoi-Delaunay triangulation (CVDT) provides high quality triangular meshes [21, 22, 53, 76]. In this work, it is applied for smoothing the meshes obtained from the mesh operations.



(a) Adapted mesh from initial mesh 2

(b) Adapted mesh from initial mesh 3

Figure B-4: Adapted meshes from direct isotropic adaptation



(a) Adapted mesh from initial mesh 2

(b) Adapted mesh from initial mesh 3

Figure B-5: Adapted meshes from direct isotropic adaptation (zoom)

B.2.1 Definition and Properties

Let $\Omega \subseteq \mathbb{R}^N$ be an open set and $\{V_i\}_{i=1}^k$ be a tessellation of Ω. Given a set of points $\{\mathbf{z}_i\}_{i=1}^k$ belonging to $\bar{\Omega}$, the Voronoi region V_i corresponding to $\mathbf{z}_i$ is defined by

$$V_i \equiv \{\mathbf{x} \in \Omega : |\mathbf{x} - \mathbf{z}_i| < |\mathbf{x} - \mathbf{z}_j|, \quad \text{for } j = 1, ..., k, j \neq i\},$$

where $|\cdot|$ denotes the Euclidean distance. The points $\{\mathbf{z}_i\}_{i=1}^k$ are called generators, and the set $\{V_i\}_{i=1}^k$ is called a Voronoi tessellation of Ω. A comprehensive treatment of Voronoi tessellations can be found in [54].

The set $\{V_i\}_{i=1}^k$ is called a centroidal Voronoi tessellations if for each Voronoi region V_i, the generator $\mathbf{z}_i$ coincides with the center of mass of V_i, that is,

$$\mathbf{z}_i = \bar{\mathbf{z}}_i \equiv \frac{\int_{V_i} \mathbf{x}\rho(\mathbf{x})d\mathbf{x}}{\int_{V_i} \rho(\mathbf{x})d\mathbf{x}},$$

where $\rho(\mathbf{x})$ is a density function. A Voronoi tessellation is generally not a CVT. Given a region Ω, an integer $k > 0$, and a density function ρ, the existence of CVT has been proven, but in general, it is not unique [20]. The dual triangulation of a CVT is called the centroidal Voronoi-Delaunay triangulation (CVDT).

CVTs can be characterized as solutions of an optimization problem. Define an energy functional for a tessellation $\{V_i\}_{i=1}^k$ and a set of points $\{\mathbf{z}_i\}_{i=1}^k$ (independent of $\{V_i\}_{i=1}^k$) as

$$\mathcal{F}(\{V_i\}_{i=1}^k, \{\mathbf{z}_i\}_{i=1}^k) \equiv \sum_{i=1}^k \int_{V_i} \rho(\mathbf{x})|\mathbf{x} - \mathbf{z}_i|^2 d\mathbf{x}. \tag{B.3}$$

Then CVTs along with their generators are critical points of the energy functional, and the minimizers are geometrically more stable [20, 21]. A property of CVTs that is useful for the mesh smoothing in this work is the equidistribution of energy when the number of generators is large, that is,

$$\int_{V_i} \rho(\mathbf{x})|\mathbf{x} - \mathbf{z}_i|^2 d\mathbf{x} \approx C, \quad \forall i, \tag{B.4}$$

for some constant C. This property can be established based on Gersho's conjecture, which predicts that on an optimal CVT, as the number of generators becomes large, all Voronoi regions become approximately congruent to the same basic cell that only depends on the dimension [33]. In two-dimension, the basic cell is a regular hexagon, and its dual cell is a regular triangle. This explains why the CVDTs lead to high quality isotropic meshes.

B.2.2 Construction of CVDTs

Denote the mesh obtained from the direct adaptation presented in Section B.1 by $\mathcal{T}_H'$. A density function ρ needs to be defined to drive $\mathcal{T}_H'$ to a CVDT with the nodes on this mesh being the generators. As the number of generators become large, Eqn. (B.4) can be approximated by

$$\rho(\mathbf{x}) A_i^2 \approx C, \quad \forall i, \tag{B.5}$$

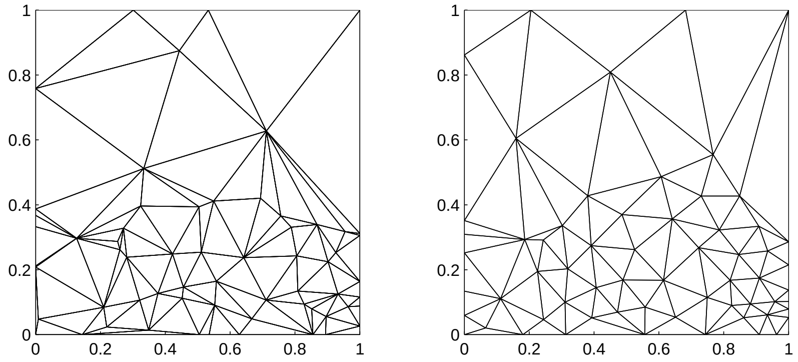
where A_i is the area of the Voronoi region V_i on the final CVT. Define a density function given by

$$\rho = \frac{1}{(A_i^c)^2}, \tag{B.6}$$

where A_i^c is the area of the Voronoi region corresponding to the node i on the mesh $\mathcal{T}_H'$. Eqn. (B.5) and (B.6) lead to $A_i \approx A_i^c$, that is, the Voronoi region on the final CVT has the same size as that on the initial tessellation. In other words, the node density on $\mathcal{T}_H'$ is maintained on the smoothed mesh of CVDT. For simplicity, A_i^c is approximated by one-third of the total area of the neighboring elements on $\mathcal{T}_H'$.

Given the mesh $\mathcal{T}_H'$ and the density $\rho(\mathbf{x})$ defined in Eqn. (B.6), a CVDT that is constrained to the boundary of the computational domain is constructed from the following two stages. First, boundary nodes are treated. Suppose the boundary Γ can be divided into a set of smooth disjoint segments Γ_i. A one-dimensional CVT construction is applied to the nodes on each Γ_i. Note that the boundary corners are always fixed. Second, a two-dimensional CVT construction is applied to all the nodes in the domain Ω, and the boundary nodes are allowed to slide along the boundary segments. A better approach for boundary node treatment involves redefining an energy function for the constrained CVT problem [21, 53]. However, that approach assumes the number of boundary nodes is not known *a priori* while in this work, the number boundary nodes is known from the mesh $\mathcal{T}'_H$. A parallelized probabilistic version of Lloyd's method is applied for the construction of CVT [42, 47]. Its implementation is described in Appendix C. After the CVT is obtained using the nodes on $\mathcal{T}'_H$, its dual Delaunay mesh is constructed and used for the next adaptation iteration.

Figure B-6 shows a random mesh and a CVDT obtained from this mesh using the density defined in Eqn. (B.6). The two meshes have similar node densities while the CVDT has a higher quality.



(a) Initial mesh

(b) CVDT

Figure B-6: A random mesh and a CVDT obtained from this mesh

B.2.3 Overall Adaptation Mechanism

One adaptation iteration consists of mesh operations and smoothing based on the concept of CVT. This is detailed in Algorithm B.2.1. At the end of each iteration, the grid obtained is a CVDT. The adaptation is terminated when the global error is less than the specified target level.

Algorithm B.2.1. One iteration of direct isotropic adaptation with CVT smoothing

1. Compute the primal and adjoint solutions, $\mathbf{u}_H$ and $\boldsymbol{\psi}_H$, on the current mesh, $\mathcal{T}_H$, using the solution method presented in Section 2.2.

2. Compute local error indicator from Eqn. (4.8) on $\mathcal{T}_H$.

3. Apply mesh operations:

    - Split edges by node insertion based on the criterion in Eqn. (B.1).
    - Remove nodes based on Eqn. (B.2).

4. Apply Delaunay triangulation on the new set of nodes.

5. Smooth the mesh using CVT with the density defined in Eqn. (B.6).

B.2.4 Numerical Results

The same test case as in Section B.1.2 is used. The same adaptation parameters are used: $\beta = 5$ and $\lambda = 0.01$.

The sensitivity of initial meshes is studied. The same initial meshes as in Section B.1.2 are used, i.e., those from Figure B-2. The adaptation for all the initial meshes are driven to the same target level. Figure B-7 shows the adaptation histories. All the initial meshes lead to essentially the same error level at the same DOF. To visualize the effect of the CVT smoothing more easily, Figure B-11 shows the adaptation results for the direct adaptation with and without the CVT smoothing. With the smoothing, error is lower at almost any DOF for all initial meshes, especially when the initial mesh has an inappropriate anisotropy. Figures B-9 and B-10 show the adapted meshes from the initial meshes 2 and 3, which have completely different anisotropies. The two adapted meshes appear similar as the initial anisotropy is lost.

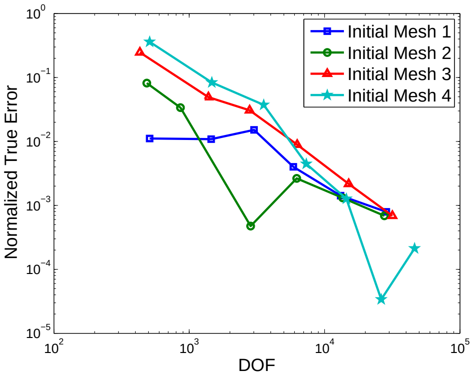
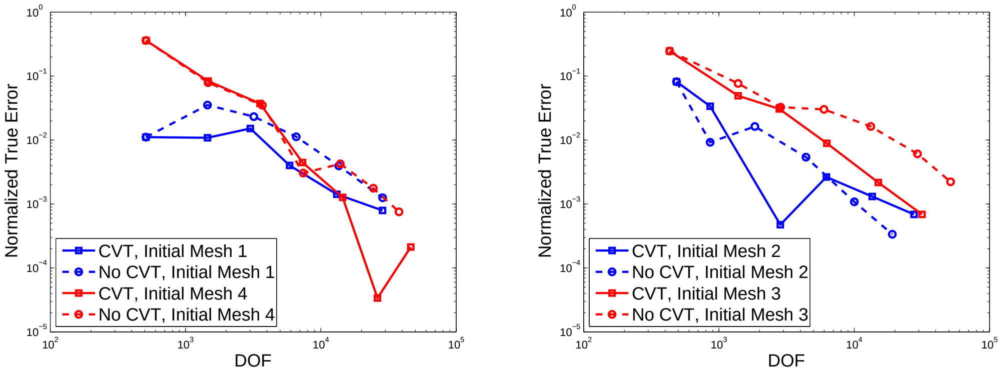


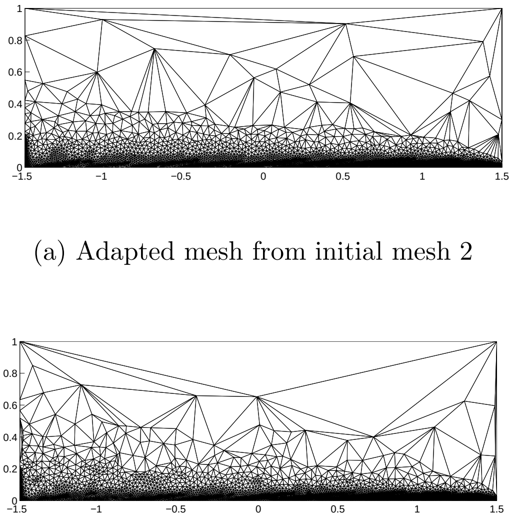
Figure B-7: Adaptation history for direct isotropic adaptation, with CVT smoothing



(a) From initial meshes 1 and 4

(b) From initial meshes 2 and 3

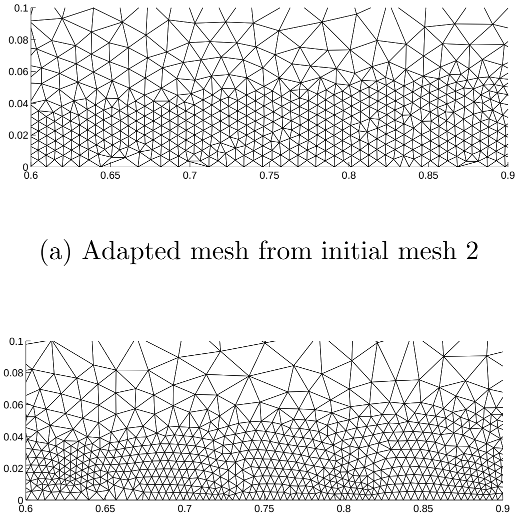
Figure B-8: Comparison between adaptation with and without CVT smoothing



(a) Adapted mesh from initial mesh 2

(b) Adapted mesh from initial mesh 3

Figure B-9: Adapted meshes from direct isotropic adaptation with CVT smoothing



(a) Adapted mesh from initial mesh 2

(b) Adapted mesh from initial mesh 3

Figure B-10: Adapted meshes from direct isotropic adaptation with CVT smoothing (zoom)

As mentioned is Section 5.1, a popular approach for output-based adaptation is based on the *a priori* assumption about the local convergence rate of the output error [26]. Figure B-11 compares the adaptation results for three strategies with interpolation orders of $p = 1$ and $p = 4$:

- Strategy 1: direct isotropic adaptation with the CVT smoothing, equidistribution of error density, $\beta = 2$, $\lambda = 0.001$

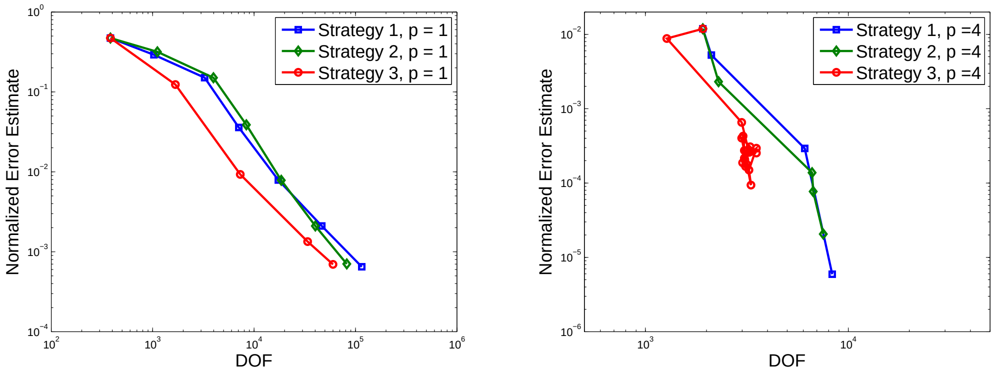
- Strategy 2: direct isotropic adaptation with the CVT smoothing, equidistribution of elemental error, $\beta = 2$, $\lambda = 0.001$

- Strategy 3: application of *a priori* size-request [26], mesh generation using BAMG [39]

As shown, for the linear interpolation, all the three strategies lead to similar DOF at the end when the target error level is met, but using the *a priori* assumption is more efficient for the intermediate iterations. In contrast, for the high-order interpolations, the adaptation using the *a priori* assumption oscillates and fails to reach the target error level. This might be due to the overestimation of the solution regularity and the convergence rate for the output error. Another note is that applying the equidistribution of error density and the equidistribution of elemental error leads to very similar error per DOF.

B.3 Summary

An isotropic direct adaptation scheme has been developed, and the concept of CVT has been applied as a smoothing mechanism. The CVT density is defined such that the node density on the original tessellations remains the same on the final CVT. The results have demonstrated that the adapted mesh is not sensitive to the initial mesh used, and the two optimality criteria - equidistribution of error density and equidistribution of elemental error - result in meshes with similar DOF for a specified error level. Also, while the existing approach with the *a priori* assumption does



(a) $p = 1$

(b) $p = 4$

Figure B-11: Comparison of isotropic adaptation results. Strategy 1: CVT smoothing, equidistribution of error density, $\beta = 2$, $\lambda = 0.001$. Strategy 2: CVT smoothing, equidistribution of elemental error, $\beta = 2$, $\lambda = 0.001$. Strategy 3: application of *a priori* size-request, BAMG.

not always converge due to the overestimation of solution regularity, the proposed scheme is insensitive to the interpolation order. For cases where the existing approach converges, the proposed scheme leads to similar DOF for a specified error level.

Appendix C

Implementation of CVT Construction

A probabilistic version of Lloyd's method for CVT construction is proposed in [42]. It is given in Algorithm C.0.1.

Algorithm C.0.1. Probabilistic Version of Lloyd's Method
Given a region Ω, a density $\rho(\mathbf{x})$, and a set of generators $\{\mathbf{z}\}_{i=1}^{k}$,

1. Generate Q sampling points $\{\mathbf{y}\}_{j=1}^{Q}$ in Ω according to the density $\rho(\mathbf{x})$.

2. For each $i = 1, ..., k$, gather together in the set W_i all sampling points closest to $\mathbf{z}_i$. If W_i is nonempty, replace $\mathbf{z}_i$ by the average of the set W_i, which is an unbiased estimator of the centroid of the Voronoi region V_i corresponding to $\mathbf{z}_i$.

3. If new points meet some convergence criterion, terminate; otherwise, return to Step 1.

In this algorithm, the convergence criterion can be based on a specified decrease in the energy functional defined in Eqn. (B.3). A simpler criterion is by specifying a maximum number of iterations. Ju et al. study the performance of using different number of iterations and different number of sampling points, Q, per iteration [42]. In this work, 200 iterations are applied, and Q is chosen to be 500 times the number of generators. Algorithm C.0.1 is implemented in parallel where each processor takes an independent set of sampling points.

One key step in Algorithm C.0.1 is random sampling according to a probability function. In this work, a continuous piecewise linear density function $\rho(\mathbf{x})$ is defined by interpolating the density function defined on each node by Eqn. (B.6), and a probability function is obtained by normalizing $\rho(\mathbf{x})$. Denote this probability function by $P(\mathbf{x})$. A rejection method is applied for random sampling based on $P(\mathbf{x})$ [67]. However, applying the method directly for $P(\mathbf{x})$ is very expensive. Instead, $P(\mathbf{x})$ can be expressed as

$$P(\mathbf{x}) = P(\boldsymbol{\xi}|\text{element } \kappa) \cdot P(\text{element } \kappa),$$

where $\boldsymbol{\xi}$ is the reference coordinate in κ corresponding to $\mathbf{x}$. For elements of linear geometry, $P(\boldsymbol{\xi})$ can be computed from only the nodal values of probability on the reference element. Thus, the random sampling procedure consists of two steps:

1. Randomly pick a triangle κ based on $P(\kappa)$ using a discrete univariate sampling method.

2. Randomly generate a point inside κ based on $P(\boldsymbol{\xi})$ using a rejection method.

Bibliography

[1] I. Babuška and A. K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13(2):214–226, 1976.

[2] Garrett E. Barter. *Shock Capturing with PDE-Based Artificial Viscosity for an Adaptive, Higher-Order, Discontinuous Galerkin Finite Element Method.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2008.

[3] T. J. Barth. Numerical aspects of computing viscous high reynolds number flows on unstructured meshes. AIAA Paper 1991-0721, 1991.

[4] Timothy J. Barth and Mats G. Larson. A posteriori error estimates for higher order Godunov finite volume methods on unstructured meshes. In R. Herban and D. Kröner, editors, *Finite Volumes for Complex Applications III*, London, 2002. Hermes Penton.

[5] F. Bassi and S. Rebay. GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations. In Karniadakis Cockburn and Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, pages 197–208. Springer, Berlin, 2000.

[6] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math*, 4:237–264, 1996.

[7] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. In A. Iserles, editor, *Acta Numerica.* Cambridge University Press, 2001.

[8] Michele Benzi, Daniel B. Szyld, and Arno van Duin. Orderings for incomplete factorization preconditioning of nonsymmetric problems. *SIAM Journal on Scientific Computing*, 20(5):1652–1670, 1999.

[9] Houman Borouchaki, Paul Louis George, Frédéric Hecht, Patrick Laug, and Eric Saltel. Delaunay mesh generation governed by metric specifications. Part I algorithms. *Finite Elem. Anal. Des.*, 25(1-2):61–83, 1997.

[10] Frank J. Bossen and Paul S. Heckbert. A pliant method for anisotropic mesh generation. In *5th Intl. Meshing Roundtable*, pages 63–74, Oct. 1996.

[11] Carlo L. Bottasso. Anisotropic mesh adaptation by metric-driven optimization. *International Journal for Numerical Methods in Engineering*, 60(3):597–639, 2004.

[12] Weiming Cao. On the error of linear interpolation and the orientation, aspect ratio, and internal angles of a triangle. *SIAM J. Numer. Anal.*, 43(1):19–40, 2005.

[13] M. J. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaptation for flow simulations. *International Journal for Numerical Methods in Fluids*, 25:475–491, 1997.

[14] Long Chen. Mesh smoothing schemes based on optimal delaunay triangulations. In *13th International Meshing Roundtable*, pages 109–120, Sept. 2004.

[15] E. F. D'Azevedo, P. A. Forsyth, and Wei-Pai Tang. Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems. *SIAM J. Matrix Anal. Appl.*, 13(3):944–961, 1992.

[16] Pedro Díez and Giovanni Calderón. Remeshing criteria and proper error representations for goal oriented h-adaptivity. *Comput. Methods Appl. Mech. Engrg*, 196:719–733, 2007.

[17] Pedro Díez and Antonio Huerta. A unified approach to remshing strategies for finite element h-adaptivity. *Comput. Methods Appl. Mech. Engrg*, 176:215–229, 1999.

[18] Laslo T. Diosady and David L. Darmofal. Preconditioning methods for discontinuous Galerkin solutions of the Navier-Stokes equations. *Journal of Computational Physics*, pages 3917–3935, 2009.

[19] C. Dobrzynski and P. Frey. Anisotropic Delaunay mesh adaptation for unsteady simulations. In *17th Intl. Meshing Roundtable*, pages 177–194, Oct. 2008.

[20] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41:637–676, 1999.

[21] Qiang Du and Max Gunzburger. Grid generation and optimization based on centroidal voronoi tessellations. *Applied Mathematics and Computation*, 133:591–607, 2002.

[22] Qiang Du and Desheng Wang. Tetrahedral mesh generation and optimization based on centroidal voronoi tessellations. *International Journal for Numerical Methods in Engineering*, 56:1355–1373, 2003.

[23] Qiang Du and Desheng Wang. Anisotropic centroidal voronoi tessellations and their applications. *SIAM Journal on Scientific Computing*, 26(3):737–761, 2005.

[24] Herbert Edelsbrunner, Tiow Seng Tan, and Roman Waupotitsch. An $o(n^2 \log n)$ time algorithm for the minmax angle triangulation. *SIAM Journal on Scientific and Statistical Computing*, 13(4):994–1008, 1992.

[25] Krzysztof J. Fidkowski. A high-order discontinuous Galerkin multigrid solver for aerodynamic applications. Masters thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2004.

[26] Krzysztof J. Fidkowski. *A Simplex Cut-Cell Adaptive Method for High-Order Discretizations of the Compressible Navier-Stokes Equations.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2007.

[27] Krzysztof J. Fidkowski, Todd A. Oliver, James Lu, and David L. Darmofal. p-multigrid solution of high-order discontiunous Galerkin discretizations of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 207(1):92–113, 2005.

[28] David A. Field. Laplacian smoothing and delaunay triangulations. *Communications in Applied Numerical Methods*, 4:709–712, 1988.

[29] L. Formaggia, S. Micheletti, and S. Perotto. Anisotropic mesh adaptation with applications to CFD problems. In H. A. Mang, F. G. Rammerstorfer, and J. Eberhardsteiner, editors, *Fifth World Congress on Computational Mechanics*, Vienna, Austria, July 7-12 2002.

[30] L. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. Technical report, 1997.

[31] Emmanuil H. Georgoulis, Edward Hall, and Paul Houston. Discontinuous galerkin methods for advection-diffusion-reaction problems on anisotropically refined meshes. *SIAM J. Sci. Comput.*, 30(1):246–271, 2007.

[32] Emmanuil H. Georgoulis, Edward Hall, and Paul Houston. Discontinuous galerkin methods on hp-anisotropic meshes ii: A posteriori error analysis and adaptivity. *Applied Numerical Mathematics*, 59:2179–2194, 2009.

[33] Allen Gersho. Asymptotically optimal block quantization. *IEEE Transaction on Information Theory*, IT-25(5):373–380, 1979.

[34] M. Giles and N. Pierce. Adjoint error correction for integral outputs. In *Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, volume 25. Springer, Berlin, 2002.

[35] M. B. Giles and E. Süli. Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality. In *Acta Numerica*, volume 11, pages 145–236, 2002.

[36] Michael B. Giles and Niles A. Pierce. Adjoint equations in CFD: duality, boundary conditions, and solution behaviour. AIAA Paper 97-1850, 1997.

[37] Wagdi G. Habashi, Julien Dompierre, Yves Bourgault, Djaffar Ait-Ali-Yahia, Michel Fortin, and Marie-Gabrielle Vallet. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. part I: general principles. *Int. J. Numer. Meth. Fluids*, 32:725–744, 2000.

[38] Ralf Hartmann and Paul Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics*, 183(2):508–532, 2002.

[39] F. Hecht. *BAMG: Bidimensional Anisotropic Mesh Generator*. INRIA, Rocquencourt, France, 1998. `http://www-rocq1.inria.fr/gamma/cdrom/www/bamg/eng.htm`.

[40] Paul Houston and Endre Süli. A note on the design of hp-adaptive finite element methods for elliptic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 194:229–243, 2005.

[41] K.E. Jansen and M.S. Shepard. On anisotropic mesh generation and quality control in complex flow problems. In *10th Intl. Meshing Roundtable*, pages 341–349, Oct. 2001.

[42] Lili Ju, Qiang Du, and Max Gunzburger. Probabilistic methods for centroidal voronoi tessellations and their parallel implementations. *Parallel Computing*, 28:1477–1500, 2002.

[43] Yannis Kallinderis and Kengo Nakajima. Finite element method for incompressible viscous flows with adaptive hybrid grids. *AIAA Journal*, 32(8):1617–1625, 1994.

[44] P. Labbé, J. Dompierre, M.-G. Vallet, F. Guilbault, and J.-Y. Trépanier. A universal measure of the conformity of a mesh with respect to an anisotropic metric field. *International Journal for Numerical Methods in Engineering*, 61(15):2675–2695, 2004.

[45] Tobias Leicht and Ralf Hartmann. Anisotropic mesh refinement for discontinuous Galerkin methods in two-dimensional aerodynamic flow simulations. *International Journal for Numerical Methods in Fluids*, 56:2111–2138, 2008.

[46] L.-Y. Li, P. Bettess, and J.W. Bull. Theoretical formulations for adaptive finite element computations. *Communications in Numerical Methods in Engineering*, 11:857–868, 1995.

[47] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transaction on Information Theory*, IT-28(2):129–137, 1982.

[48] Rainald Löhner. Matching semi-structured and unstructured grids for navier-stokes calculations. AIAA Paper 1993-3348, 1993.

[49] James Lu. *An a Posteriori Error Control Framework for Adaptive Precision Optimization Using Discontinuous Galerkin Finite Element Method.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.

[50] D. J. Mavriplis. Results from the 3rd drag prediction workshop using the NSU3D unstructured mesh solver. Aiaa paper, 2007.

[51] Cristian R. Nastase and Dimitri J. Mavriplis. High-order discontinuous Galerkin methods using an *hp*-multigrid approach. *Journal of Computational Physics*, 213(1):330–357, 2006.

[52] Marian Nemec, Michael J. Aftosmis, and Mathias Wintzer. Adjoint-based adaptive mesh refinement for complex geometries. AIAA Paper 2008-725, 2008.

[53] Hoa Nguyen. *Centroidal Voronoi Tessellations for Mesh Generation: From Uniform to Anisotropic Adaptive Triangulations.* PhD dissertation, Florida State University, College of Arts and Sciences, Spring 2008.

[54] A. Okabe, B. Boots, and K. Sugihara. *Sptial Tessellations; Concepts and Applications of Voronoi Diagrams.* Wiley, Chichester, UK, 1992.

[55] Todd A. Oliver. Multigrid solution for high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. Masters thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2004.

[56] Todd A. Oliver. *A Higher-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier-Stokes Equations.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2008.

[57] E. Oñate, Joaquin Arteaga, Julio García, and Roberto Flores. Error estimation and mesh adaptivity in incompressible viscous flows using a residual power appoarch. *Computer Methods in Applied Mechanics and Engineering*, 195:339–362, 2006.

[58] E. Oñate and G. Bugeda. A study of mesh optimality criteria in adaptive finite element analysis. *Engineering Computations*, 10:307–321, 1993.

[59] Doug Pagnutti and Carl Ollivier-Gooch. Delaunay-based anisotropic mesh adaptation. In *17th Intl. Meshing Roundtable*, pages 141–157, Oct. 2008.

[60] Doug Pagnutti and Carl Ollivier-Gooch. A generalized framework for high order anisotropic mesh adaptation. *Computers and Structures*, 87(11-12):670 – 679, 2009.

[61] Michael A. Park. *Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2008.

[62] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 72:449–466, 1987.

[63] Per-Olof Persson and Jaime Peraire. Newton-GMRES preconditioning for Discontinuous Galerkin discretizations of the Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 30(6):2709–2722, 2008.

[64] Per-Olof Persson and Gilbert Strang. A simple mesh generator in MATLAB. *SIAM review*, 46(2):329–345, 2004.

[65] Shmuel Rippa. Long and thin triangles can be good for linear interpolation. *SIAM Journal on Numerical Analysis*, 29(1):257–270, 1992.

[66] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.

[67] S. Ross. *A First Course in Probability.* Prentice Hall, Englewood Cliffs, fifth edition, 1998.

[68] Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[69] Yousef Saad. ILUT: a dual threshold incomplete LU factorization. *Numerical Linear Algebra with Applications*, 1(4):387–402, 1994.

[70] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, 1996.

[71] Jonathan Richard Shewchuk. Lecture notes on Delaunay mesh generation. Technical report, 1999.

[72] Jonathan Richard Shewchuk. What is a good linear finite element? Interpolation, conditioning, anisotropy, and quality measures. Technical report, University of California at Berkeley, 2002.

[73] John C. Vassberg, Mark A. DeHaan, and Tony J. Sclafani. Grid generation requirements for accurate drag predictions based on OVERFLOW calculations. AIAA Paper 2003-4124, 2003.

[74] D. A. Venditti and D. L. Darmofal. Grid adaptation for functional outputs: application to two-dimensional inviscid flows. *Journal of Computational Physics*, 176(1):40–39, 2002.

[75] D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, 187(1):22–46, 2003.

[76] Desheng Wang and Qiang Du. Mesh optimization based on the centroidal voronoi tessellation. *International Journal of Numerical Analysis and Modeling*, 2:100–113, 2005.

[77] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimatior and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering*, 24:337–357, 1987.