

**An Automated Reliable Method for
Two-Dimensional Reynolds-averaged Navier-Stokes
Simulations**

by

James M. Modisette

S.M., Massachusetts Institute of Technology (2008)

B.S., Massachusetts Institute of Technology (2005)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2011

© Massachusetts Institute of Technology 2011. All rights reserved.

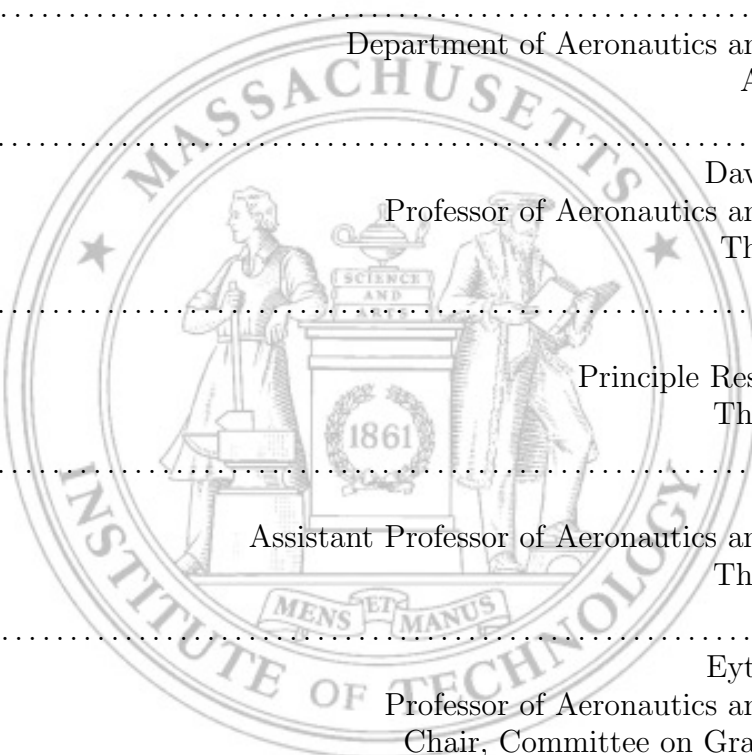
Author
Department of Aeronautics and Astronautics
August 18, 2011

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

Certified by
Robert Haimes
Principle Research Engineer
Thesis Committee

Certified by
Qiqi Wang
Assistant Professor of Aeronautics and Astronautics
Thesis Committee

Accepted by
Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

The seal of the Massachusetts Institute of Technology is a large, faint watermark in the background. It features a circular border with the text "MASSACHUSETTS INSTITUTE OF TECHNOLOGY". Inside the circle, there are two figures: one on the left holding a staff and a book, and one on the right holding a book. Between them is a pedestal with a book and the year "1861". At the bottom of the seal is a banner with the Latin motto "MENS ET MANUS".

An Automated Reliable Method for Two-Dimensional Reynolds-averaged Navier-Stokes Simulations

by
James M. Modisette

Submitted to the Department of Aeronautics and Astronautics
on August 18, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The development of computational fluid dynamics algorithms and increased computational resources have led to the ability to perform complex aerodynamic simulations. Obstacles remain which prevent autonomous and reliable simulations at accuracy levels required for engineering. To consider the solution strategy autonomous and reliable, high quality solutions must be provided without user interaction or detailed previous knowledge about the flow to facilitate either adaptation or solver robustness. One such solution strategy is presented for two-dimensional Reynolds-averaged Navier-Stokes (RANS) flows and is based on: a higher-order discontinuous Galerkin finite element method which enables higher accuracy with fewer degrees of freedom than lower-order methods; an output-based error estimation and adaptation scheme which provides quantifiable measure of solution accuracy and autonomously drive toward an improved discretization; a non-linear solver technique based on pseudo-time continuation and line-search update limiting which improves the robustness for solutions to the RANS equations; and a simplex cut-cell mesh generation which autonomously provides higher-order meshes of complex geometries.

The simplex cut-cell mesh generation method presented here extends methods previously developed to improve robustness with the goal of RANS simulations. In particular, analysis is performed to expose the impact of small volume ratios between arbitrarily cut elements on linear system conditioning and solution quality. Merging of the small cut element into its larger neighbor is identified as a solution to alleviate the consequences of small volume ratios. For arbitrarily cut elements randomness in the algorithm for generating integration rules is identified as a limiting factor for accuracy and recognition of canonical element shapes are introduced to remove the randomness. The cut-cell method is linked with line-search based update limiting for improved non-linear solver robustness and Riemannian metric based anisotropic adaptation to efficiently resolve anisotropic features with arbitrary orientations in RANS flows. A fixed-fraction marking strategy is employed to redistribute element areas and steps toward meshes which equidistribute elemental errors at a fixed degree of freedom.

The benefit of the higher spatial accuracy and the solution efficiency (defined as accuracy per degree of freedom) is exhibited for a wide range of RANS applications including subsonic through supersonic flows. The higher-order discretizations provide more accurate solutions than second-order methods at the same degree of freedom. Furthermore, the cut-cell meshes demonstrate comparable solution efficiency to boundary-conforming meshes while significantly decreasing the burden of mesh generation for a CFD user.

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

Acknowledgments

Contents

1	Introduction	17
1.1	Motivation	17
1.2	Objectives	19
1.3	Solution Strategy Background	20
1.3.1	Higher-Order Method	20
1.3.2	Output-Based Error Estimation and Adaptation	21
1.3.3	Cut-Cell Mesh Generation	24
1.4	Thesis Overview	27
2	Discretization of the RANS-SA Equations	29
2.1	The RANS Equations	29
2.2	The SA Turbulence Model	30
2.3	Spatial Discretization	33
2.4	Shock capturing	37
3	Non-Linear Solution Technique	41
3.1	Pseudo-Time Continuation	41
3.2	Pseudo-Time Solution Update	44
3.3	Line-Search Solution Update Limiting	45
4	Cut-Cell Mesh Generation	49
4.1	Geometry Definition	49
4.2	Intersection Algorithm	51
4.3	Integration for Arbitrary Element Shape	56
4.4	Canonical Shape Recognition	61
4.5	Solution Basis Space	64
5	Small Volume Ratios	71
5.1	Boundary Derivative Outputs with Small Volume Ratios	77
5.2	Analysis of the Conditioning of a One Dimensional Problem with Small Volume Ratios	81
5.2.1	Definitions	81
5.2.2	Bilinear form to linear operator	82
5.2.3	Restriction to finite element space	83
5.2.4	Condition number for operators between Hilbert spaces	84
5.2.5	Linear algebraic representation	86

5.2.6	Relate stiffness matrix to Hilbert space setting	86
5.2.7	Matrix condition number – quasi-uniform mesh	88
5.2.8	Matrix condition number – mesh with a small volume ratio	92
5.2.9	Implications of $\kappa(A) = \frac{C}{\alpha} \mathcal{O}(h^{-2}) \mathcal{O}(VR^{-1})$	99
5.3	Modified Discretization Space	100
5.4	Model Problem Results	101
6	Output-Based Error Estimation and Adaptation	107
6.1	Output-Based Error Estimation	107
6.2	Adaptation Strategy	110
6.2.1	Fixed-Fraction Marking	112
6.2.2	Anisotropy Detection	114
6.2.3	Limit Requested Element Metrics	115
6.2.4	Generation of Continuous Metric Field	117
6.2.5	Metric Request Construction and Explicit Degree of Freedom Control	117
6.2.6	Building Metric Request for Null Cut Elements	118
6.2.7	DOF-“Optimal” Mesh	120
7	Results	125
7.1	Comparison of Boundary-Conforming and Cut-Cell Solution Efficiency	125
7.1.1	NACA0012 Subsonic Euler	125
7.1.2	RAE2822 Subsonic RANS-SA	128
7.1.3	RAE2822 Transonic RANS-SA	130
7.1.4	NACA0012 Supersonic RANS-SA	133
7.1.5	Multi-element Supercritical 8 Transonic RANS-SA	139
7.1.6	Summary	141
7.2	DOF-Controlled Adaptation for Parameter Sweeps	142
7.2.1	Fixed Mesh vs. Adaptive Mesh	142
7.2.2	Comparison of Boundary-Conforming and Cut-Cell	145
7.3	Comparison of Boundary-Conforming and Cut-Cell Solution “Cost”	146
8	Conclusions	155
8.1	Summary and Contributions	155
8.2	Future Work	157
	Bibliography	160

List of Figures

1-1	Computed drag convergence for a wing-alone configuration at $M_\infty = 0.76$, $\alpha = 0.5^\circ$, and $Re = 5 \times 10^6$ with global mesh refinement taken from Mavriplis [83]. Convergence of drag is plotted for the refinement of two mesh families of the same wing geometry.	18
1-2	Convergence history for $p = 0 \rightarrow 3$ of RANS simulations of an RAE2822 airfoil ($M_\infty = 0.734$, $\alpha = 2.79^\circ$, $Re_c = 6.5 \times 10^6$, 8,096 $q = 3$ quadrilateral elements) taken from Bassi et al.[18].	19
1-3	Illustration of the autonomous output-based error estimation and adaptation strategy.	22
1-4	Diagram of the options for converting a linear boundary conforming mesh to a mesh containing higher-order geometry information.	25
3-1	Residual convergence for three boundary-conforming meshes for a subsonic simulation of the RANS-SA equations over the RAE2822 airfoil ($M = 0.3$, $\alpha = 2.31$, $Re = 6.5 \times 10^6$).	48
4-1	Example of spline geometry representation of a NACA0012 airfoil. The spline parameter, s , defines the computational domain to be external to the airfoil. .	50
4-2	Illustration of embedded and farfield domain representation for external flow over an airfoil.	50
4-3	Illustration distinguishing different <i>zerod</i> , <i>oned</i> , and <i>twod</i> objects within a cut grid.	51
4-4	Degenerate intersection cases.	52
4-5	Illustration of <i>oned</i> objects at the leading edge of an airfoil.	53
4-6	Illustration of typical cut elements at an airfoil trailing edge. The left background element straddling the airfoil is treated as two cut elements with each cut element defined by separate loops of <i>oned</i> objects. The arbitrarily cut element at the trailing edge is a single cut element with four neighbors. The direction of the Loop is shown for the element at the trailing edge.	55
4-7	Example of a cut-cell mesh for a NACA0012 airfoil. The spline geometry is shown in red.	56
4-8	An example of the “speckled” 2D integration points for a cut-cell mesh. In order to support $p = 5$ solutions, upwards of 484 points are suggested to adequately cover the interior of the element.	57
4-9	An example domain used with the two-dimensional scalar convection-diffusion model problem. For viewing, the cell aspect ratio is set to 1.	58

4-10	Example of a boundary-curved domain. The boundary-conforming domain is globally linear with a single curved boundary on the geometry surface.	58
4-11	Plot of the heat flux distribution along the inner radial boundary of the computational domain.	59
4-12	Convergence history of the minimum and maximum heat flux distribution error at solution orders 1 through 5, where 100 different sets of “speckled” points are used for integration rules at the four grid refinement levels.	60
4-13	Range of heat flux distribution error over a 100 sets of “speckled” points at $p = 5$ for each grid refinement level.	60
4-14	Triangles and quadrilaterals are recognizable canonical element shapes and improve the quality of the integration rules. The example elements are the canonical version of the cut elements shown in Figure 4-8 with their canonical quadrature points.	61
4-15	Conversion of a three-sided cut element to a higher-order canonical triangle. A $q = 5$ Lagrange basis is used for the illustration.	62
4-16	Maps for element and solution representation.	65
4-17	Two mesh families used to examine the effect of a Cartesian basis compared to a parametric basis on solution accuracy. For viewing the aspect ratio is set to 1.	67
4-18	Comparison of the convergence in the heat flux distribution errors for cases with parametric and Cartesian approximation functions on globally curved higher-order meshes and globally linear meshes with a single curved boundary. The plots indicate, although there is a small deterioration in the error and rates with the Cartesian functions, the Cartesian functions still perform well at higher order, even in boundary-curved meshes.	68
4-19	Illustration of linear shadow element options from typical cut elements at an airfoil’s trailing edge. The \star indicates the preferred option given the element type.	70
5-1	Example of a small volume ratio. Usually, small volume ratios occur when a grid node is just inside the computational domain.	71
5-2	Diagram of mesh when grid has uniform h except for the first element where $h_1 = hVR$	73
5-3	Plot of solution and its derivative for the one-dimensional model problem. The exact solution is plotted along with computed solutions for $N_{\text{element}} = 16$, $p = 3$ and $VR = 1$ and $VR = 10^{-8}$. The inset figures show the solution at the left boundary.	74
5-4	Derivative of the solution for the one-dimensional model problem plotted in the reference space of the leftmost element in the domain with a $VR = 10^{-8}$, $N_{\text{element}} = 16$, $p = 3$	74
5-5	The convergence of the L^2 solution error with varying critical volume ratio. Due to the tiny size of the element with the critical volume ratio, the small volume ratio has no impact on the L^2 error.	75
5-6	The convergence of the broken H^1 solution error with varying critical volume ratio. The critical volume ratio has no impact on the H^1 error.	75

5-7	The convergence of the error in the output $J(u) = \nu \frac{du}{dx} \Big _{x=0}$ for a range of volume ratios for the one-dimensional model problem, Equation (5.1).	76
5-8	Plot showing the variation of the condition number versus element size and volume ratio for the one-dimensional model problem.	77
5-9	The convergence of the error in the output $J(u) = \nu \frac{du}{dx} \Big _{x=0}$ for a range of volume ratios for the one-dimensional model problem, Equation (5.1). The selection of ϱ for evaluating $J_{h,p}(u_{h,p}) = a_{h,p}^{CG}(u_{h,p}, \varrho) - (f, \varrho)$ is critical for limiting the influence of small volume ratios. When $\varrho = \phi_1$, the impact of small volume ratios is large. If ϱ is not a function of VR, like $1 - x$, there is no impact of small volume ratios. These results are from a continuous Galerkin discretization with strong boundary conditions.	80
5-10	Diagram relating the equivalence of the actions of the the interpretation operator, \mathcal{I}_h , and the Hilbert space operator, \mathcal{A}_h , to the action of the matrix, A , and the functional interpretation operator, \mathcal{I}'_h , on Euclidean space, \mathbb{R}^n . (Taken from [71])	88
5-11	The effect of nudging node 1 to eliminate the small volume ratio associated with element A	101
5-12	Illustration of the effect of merging element A into element B . The resulting element, C , maintains the solution basis of element B and the quadrature points are taken from both element A and B	102
5-13	Original and merged domains for the one-dimensional model problem. e_1 and e_2 are merged to form e_m	102
5-14	The convergence of the error in the output $\frac{du}{dx} \Big _{x=0}$, with $J_h(u_h) = a_h^{CG}(u_h, \phi_1) - (f, \phi_1)$ for a range of volume ratios for the one-dimensional model problem, Equation (5.1). Merging removes the impact of the small volume ratio in the domain.	103
5-15	Convergence of the heat flux distribution error for cut-cell meshes on the two-dimensional model problem. The errors in boundary-conforming cut cases are compared to the errors in cut meshes with small volume ratios that have either been merged out or remain.	103
5-16	Boundary distributions of heat flux for the two-dimensional convection-diffusion problem using merged and non-merged cut grids.	105
6-1	Mesh metric-field duality.	111
6-2	Flow chart detailing a single adaptation step.	112
6-3	Fixed fraction adaptation strategy	113
6-4	An example of a limited metric which corresponds to the maximum element coarsening.	117
6-5	Multiply-cut element where the requested metric for element A is passed to nodes 1 and 2 but not node 3.	118
6-6	Cut elements intersecting a viscous wall form a wake-like feature in the background mesh.	120
6-7	Example describing the process of forming requested metrics on null elements.	120
6-8	Example of the initial and DOF-“optimal” meshes for subsonic RAE2822 RANS-SA flow ($M_\infty = 0.3$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$, $p = 3$, $DOF = 40k$).	121

6-9	The the error estimate, the error, the drag, and the degree of freedom adaptation history for a set of initial meshes applied to the subsonic RAE2822 RANS-SA flow ($M_\infty = 0.3$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$, $p = 3$, $\text{DOF} = 40,000$).	122
7-1	Mach number distribution, initial mesh, and the DOF-“optimal” meshes for subsonic NACA0012 Euler flow ($M_\infty = 0.5$, $\alpha = 2.0^\circ$). The Mach contour lines are in 0.05 increments.	126
7-2	Envelopes of drag coefficients and c_d error estimates for subsonic NACA0012 Euler flow ($M_\infty = 0.5$, $\alpha = 2.0^\circ$).	127
7-3	Mach number distribution, initial mesh, and the DOF-“optimal” meshes for for subsonic RAE2822 RANS-SA flow ($M_\infty = 0.3$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$). The Mach contour lines are in 0.05 increments.	128
7-4	Envelopes of drag coefficients and c_d error estimates for subsonic RAE2822 RANS-SA flow ($M_\infty = 0.3$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$).	129
7-5	Mach number distribution, initial mesh, and the DOF-“optimal” meshes for subsonic RAE2822 RANS-SA flow ($M_\infty = 0.729$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$). The Mach contour lines are in 0.025 increments.	130
7-6	Envelopes of drag coefficients and c_d error estimates for transonic RAE2822 RANS-SA flow ($M_\infty = 0.729$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$).	132
7-7	The Mach number distribution for the supersonic NACA0006 RANS-SA flow ($M_\infty = 2.0$, $Re_c = 10^6$, $\alpha = 2.0^\circ$) and the DOF-“optimal” meshes obtained for $p = 1$ and $p = 3$ at $80k$ degrees of freedom adapting to drag. The Mach contour lines are in 0.1 increments.	134
7-8	Envelopes of drag coefficients and c_d error estimates for supersonic NACA0006 RANS-SA flow ($M_\infty = 2.0$, $Re_c = 10^6$, $\alpha = 2.0^\circ$).	135
7-9	Envelopes of pressure signal and error estimates for supersonic NACA0006 RANS-SA flow ($M_\infty = 2.0$, $Re_c = 10^6$, $\alpha = 2.0^\circ$).	137
7-10	The pressure perturbation distribution, $(p(u) - p_\infty)/p_\infty$, for the supersonic NACA0006 RANS-SA flow ($M_\infty = 2.0$, $Re_c = 2.0 \times 10^7$, $\alpha = 2.5^\circ$) and the DOF-“optimal” meshes obtained for $p = 1$ and $p = 3$ at $80k$ degrees of freedom adapting to the pressure signal 50 chords below the airfoil.	138
7-11	Mach number distribution, initial mesh, and the DOF-“optimal” meshes for transonic MSC8 RANS-SA flow ($M_\infty = 0.775$, $Re_c = 2.0 \times 10^7$, $\alpha = -0.7^\circ$).	139
7-12	Envelopes of drag coefficients and c_d error estimates for transonic MSC8 RANS-SA flow ($M_\infty = 0.775$, $Re_c = 2.0 \times 10^7$, $\alpha = -0.7^\circ$).	140
7-13	The lift curve and the c_l error obtained using the fixed mesh and adaptive meshes for the three-element MDA airfoil.	143
7-14	The error indicator distribution, $\log_{10}(\eta_\kappa)$, for the three-element MDA airfoil at $\alpha = 23.28^\circ$ obtained on the 8.10° optimized mesh and the 23.28° optimized mesh.	144
7-15	The Mach number distribution for the three-element MDA airfoil at $\alpha = 23.28^\circ$ obtained on the 8.10° optimized mesh and the 23.28° optimized mesh. The Mach contour lines are in 0.05 increments.	144
7-16	The initial and lift-adapted grids for the three-element MDA airfoil at selected angles of attack.	145

7-17	Envelopes of drag coefficients and c_d error estimates for $\alpha = 8.1^\circ$ MDA RANS-SA flow ($M_\infty = 0.2$, $Re_c = 9 \times 10^6$).	147
7-18	Envelopes of drag coefficients and c_d error estimates for $\alpha = 16.21^\circ$ MDA RANS-SA flow ($M_\infty = 0.2$, $Re_c = 9 \times 10^6$).	148
7-19	Envelopes of drag coefficients and c_d error estimates for $\alpha = 21.34^\circ$ MDA RANS-SA flow ($M_\infty = 0.2$, $Re_c = 9 \times 10^6$).	149
7-20	Envelopes of drag coefficients and c_d error estimates for $\alpha = 23.28^\circ$ MDA RANS-SA flow ($M_\infty = 0.2$, $Re_c = 9 \times 10^6$).	150
7-21	c_l error estimate convergence with adaptation iteration during generation of lift curve for MDA RANS-SA flow ($M_\infty = 0.2$, $Re_c = 9 \times 10^6$) with boundary-conforming and cut-cell meshes.	151

List of Tables

3.1	Summary of physicality check limits on the global CFL number.	44
3.2	Number of basis functions per element, n_{bf} , for a given solution order and reference element.	45
3.3	Summary of line-search limits on the global CFL number.	47
4.1	Table listing the information that is stored to define the different <i>zerod</i> objects.	52
4.2	Table listing the relevant information that is stored to define the different <i>oned</i> objects.	53
4.3	Table comparing heat flux distribution errors calculated using sets of 484 randomly “speckled” points. All results are for $p = 5$	61
4.4	Table comparing heat flux distribution errors calculated using sets of randomly “speckled” points, distributed sampling points, and a canonical-cut grid. The N_{quad} for the “speckled” points is taken from the distributed sampling points to allow for the comparison between the methods. The results are for $p = 5$	63
7.1	Summary of “cost” to generate the lift curve for the MDA airfoil using boundary-conforming and cut-cell meshes.	152

Chapter 1

Introduction

1.1 Motivation

Computational fluid dynamics (CFD) methods have improved greatly over the past few decades, driven by the desire to perform more complex simulations. As Mavriplis et al. [86] describes, “While it is true that capabilities exist that are used successfully in every-day engineering calculations, radical advances in simulation capability are possible through the coupling of increased computational power with more capable algorithms.” Controlling simulation accuracy is a primary issue for the application of CFD to increasingly complex problems.

A critical step in the application of CFD is mesh generation. Meshing is commonly performed by engineers who are required to make decisions about where increased mesh resolution is needed. CFD’s dependence on human interaction is costly in terms of man hours and has the potential to introduce solution errors due to the mesh dependence of CFD solutions. In addition, this dependence on human interaction limits the automation that could be achieved with computational models. In 2007, following the third AIAA Drag Prediction Workshop (DPW-III) [1, 121], Mavriplis [83] used a generic wing-alone geometry at $M_\infty = 0.76$, $\alpha = 0.5^\circ$, and $Re = 5 \times 10^6$ to demonstrate CFD’s dependence on the initial mesh topology. Figure 1-1, taken from Mavriplis [83], shows the convergence of drag with mesh refinement for two families of meshes representing the same wing geometry. Both mesh families consist of four meshes and all the solutions were computed using the NSU3D code, an unstructured mesh Reynolds-averaged Navier-Stokes (RANS) solver [84, 85, 87].

The first set of meshes was generated at NASA Langley using the VGRID grid generation program [109], while the second set of meshes was generated independently at the Cessna Aircraft Company. Typical industry practice for an isolated wing problem is to use one to four million elements. However, as illustrated by Figure 1-1, even with an increase in refinement of an order of magnitude more than typical industry practice, the spread in the computed drag between the two meshes is approximately four drag counts. A Breguet range equation analysis demonstrates that a difference of one drag count for a long-range passenger jet corresponds to approximately four to eight passengers [42, 120]. Thus, the spread of four drag counts between the two mesh families is significant. Generating solutions to engineering-required accuracy of one to one tenth of a drag count is necessary for CFD to be a useful design tool [125].

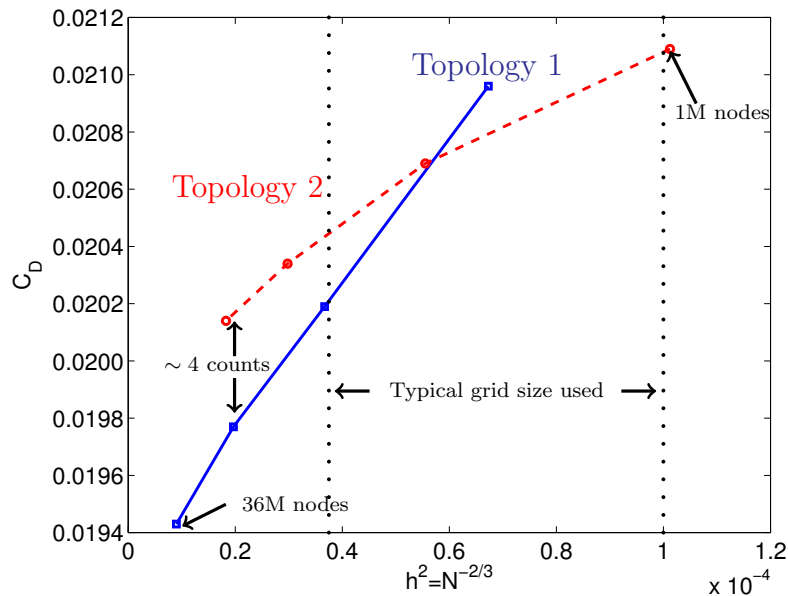


Figure 1-1: Computed drag convergence for a wing-alone configuration at $M_\infty = 0.76$, $\alpha = 0.5^\circ$, and $Re = 5 \times 10^6$ with global mesh refinement taken from Mavriplis [83]. Convergence of drag is plotted for the refinement of two mesh families of the same wing geometry.

In addition to ensuring engineering-required error levels, improving the robustness of RANS solution algorithms is critical. Convergence to a steady state solution can be challenging and tests the limits of a non-linear solver. Generally, while the linear systems are

poorly conditioned, the lack of robustness stems from the non-linearity of the problem. The convergence results of Bassi et al. [18], Figure 1-2, confirm the author’s experience. Typically, in RANS simulations, residual convergence history is dominated by slow overall convergence and a lack of Newton convergence. The poor convergence tends to include spurious residual jumps where, over a single iteration, the residual norm will increase by over an order of magnitude. The residual jump is often followed by a period of residual decrease, but the process appears to arbitrarily repeat itself.

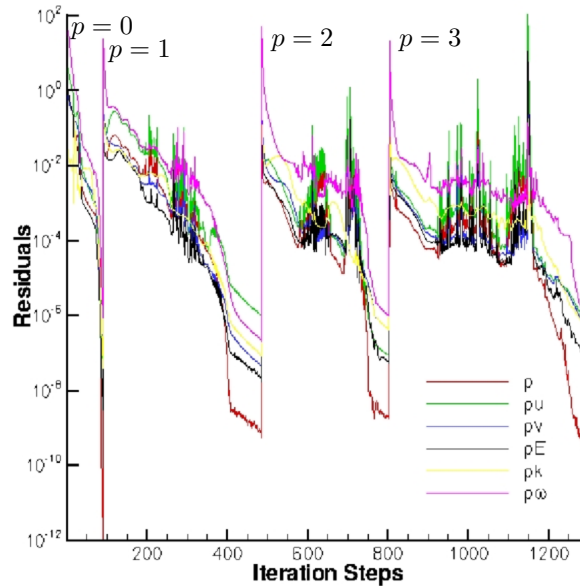


Figure 1-2: Convergence history for $p = 0 \rightarrow 3$ of RANS simulations of an RAE2822 airfoil ($M_\infty = 0.734$, $\alpha = 2.79^\circ$, $Re_c = 6.5 \times 10^6$, 8,096 $q = 3$ quadrilateral elements) taken from Bassi et al.[18].

1.2 Objectives

Algorithm advances are required, in order to meet the demand for more complex CFD simulations. The objective of this work is to develop a reliable solution strategy that provides engineering-required accuracy for the two-dimensional RANS equations. To be reliable the strategy must be fully autonomous without requiring user interaction or detailed previous knowledge about the flow to facilitate either adaptation or solver robustness. To achieve the

desired reliability and engineering-required accuracy, this work presents a solution strategy that incorporates a higher-order discretization, cut-cell meshes, output-based adaptation, and a line search based non-linear solver technique.

1.3 Solution Strategy Background

1.3.1 Higher-Order Method

For the last couple of decades, finite volume discretizations have been the industry standard for CFD in the aerospace industry. Complex simulations using finite volume discretization have been made possible through improvement in computational hardware and solution algorithms. However, traditional industrial finite volume schemes are second-order accurate, where a global uniform mesh refinement results in reduction of solution error by a factor of four, but an increase of eight in the number of degrees of freedom in three dimensions [86]. Higher spatial accuracy may be obtained with fewer degrees of freedom by using a higher-order finite volume scheme, but higher-order finite volume schemes based on reconstruction of the cell or nodal averages extend the numeric stencil and complicate the treatment of boundary conditions [100].

Higher-order finite element discretizations provide an alternative for achieving higher accuracy with fewer degrees of freedom than second-order schemes. This work uses the discontinuous Galerkin (DG) method. The DG method can maintain a compact nearest neighbor stencil (viewed element-wise), as the solution representation is discontinuous across elements and coupling comes only through face fluxes. Higher-order accuracy is obtained in the DG method by increasing the polynomial order used to represent the solution in each element.

The DG method was originally introduced for the neutron transport equation by Reed and Hill [111]. One of the first extensions to the original DG method was by Chavent and Salzano [27] who applied it to non-linear hyperbolic problems using Godunov's flux. Cockburn, Shu, and their co-authors were influential in expanding the use of the DG method. They combined DG spatial discretization with Runge-Kutta explicit time integration for non-linear hyperbolic problems [29–31, 33, 34]. Separately, Allmaras and Giles [4, 5] developed a

second-order DG scheme for the Euler equations. This method is based on taking moments of the Euler equations as suggested by van Leer [119].

DG has also been extended to elliptic problems, beginning with interior penalty (IP) methods [7, 127]. More recently, Bassi and Rebay developed two methods (BR1 and BR2) [15, 16] and applied them to the Navier-Stokes equations. Similarly, Cockburn and Shu developed local discontinuous Galerkin (LDG) for convection-diffusion problems [32]. However, LDG has an extended stencil when it is used for unstructured grid problems in multiple dimensions. The extended stencil led to the development of compact discontinuous Galerkin (CDG) by Peraire and Persson [104]. Rigorous frameworks for analyzing various DG methods have been developed by numerous researchers including Arnold et al. [8] who presented a unified framework to analyze stability and convergence of DG schemes for elliptic problems.

The DG method has additionally been applied to the RANS equations. Specifically, Bassi and Rebay [14, 18] have successfully used the BR2 method for the RANS equations with a k - ω turbulence model [128]. Nguyen et al. [92] used CDG for RANS with the Spalart-Allmaras (SA) turbulence model [115]. Since then, Landmann et al. [77], Burgess, Nastase, and Mavriplis [24], and Hartmann and Houston [63] have also applied variants of the DG discretization to the RANS equations. This work builds off the implementation of Oliver and Darmofal [94, 96, 98], which uses the BR2 method with the SA turbulence model for closure.

1.3.2 Output-Based Error Estimation and Adaptation

Output-based error estimation and adaptation autonomously reduces discretization error by estimating the error in a solution output and generating an improved mesh. Figure 1-3 shows an illustration of the adaptive framework. In this setting, a CFD user specifies a problem, an output of interest, a maximum allowable error, and a maximum run-time. From these inputs the adaptive strategy proceeds by (1) running a simulation on an existing (typically coarse) mesh, (2) computing an error estimate for the output of interest, and (3) determining whether the error tolerance or time constraint was met or if the mesh should be adapted and the process repeated. In the case where the mesh is adapted, the error estimate must be localized to identify regions where the mesh resolution requires improvement. The

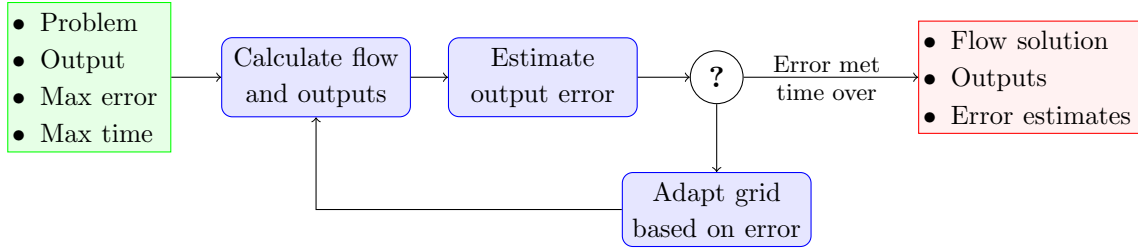


Figure 1-3: Illustration of the autonomous output-based error estimation and adaptation strategy.

adaptation strategy is based on two elements: the output-based error estimate and the mechanics of changing the discretization to improve output error.

Error Estimate

Many methods exist for estimating the error in a solution. For instance, local error estimates can be performed by computing the difference between the current solution and a solution computed on a refined discretization, either from a refined mesh or increased solution order. This estimation strategy focuses on local solution errors and can be viewed similarly to feature-based adaptation where refinement requests are based on large local gradients. The local error estimation can fail in convection problems where small upstream errors can propagate and significantly change output evaluation [116]. For example, small errors can affect the location of boundary layer separation or a shock and lead to a significant change in outputs such as lift or drag.

The error estimation method used in this work is based on the Dual Weighted Residual (DWR) method from Becker and Rannacher [19, 20]. In the DWR method, the error in a solution output, such as lift or drag, is expressed in terms of weighted residuals. The weighted residuals are constructed using the dual problem and Galerkin orthogonality of the finite element discretization. The solution to the dual problem, the adjoint, relates local perturbations to an output of interest. For output-based error estimation the perturbations are the discretization error of the primal problem. The adjoint highlights aspects of the discretization which are most influential to the output of interest, thus it plays a central role in performing output-based error estimation. With the DWR method, asymptotically sharp error estimates can be achieved by multiplying local residuals with the adjoint solution.

Many researchers in the literature have applied the DWR method to the DG discretization with minor differences [45, 59, 61, 62, 65, 78, 82].

Extensions to the DWR method also appear in the literature. Pierce and Giles [52, 54, 108] presented the opportunity for improved output functional evaluation through error correction in the absence of Galerkin orthogonality. Venditti and Darmofal [124] were the first to apply an output-based error estimation and anisotropic adaptive method to the RANS equations. Their work concluded that for a standard finite volume scheme the output-based adaptive approach was superior in terms of reliability, accuracy in computed outputs, and computational efficiency relative to adaptive schemes based on feature detection.

Adaptation

Once an error estimate has been computed, the goal of adaptation is to modify the discretization to decrease the estimated error. There are three general adaptation options: h -adaptation, where the interpolation order remains fixed and the element sizes, h , are adjusted; p -adaptation, where the interpolation order, p , in elements with large error is increased to add resolution while the mesh remains unchanged [9, 82, 114]; or hp -adaptation, where both the interpolation order and the element size are changed [50, 51, 57, 58, 67, 116, 126]. All three of these adaptation strategies have strengths and weaknesses. p -adaptation is dependent on solution regularity. In the presence of solution discontinuities, higher-order interpolations demonstrate Gibbs phenomenon and p -adaptation will be ineffective. However, if sufficient solution smoothness is present, p -adaptation exhibits spectral convergence (in the limit of global increase in solution order). h -adaptation, though limited to polynomial convergence, is particularly useful in shock or boundary layer cases where increased solution resolution is locally needed. The solution regularity of CFD problems in aerospace is limited by singularities and singular perturbations. To achieve engineering required accuracy, resolution of the singular features is needed as opposed to high asymptotic convergence of the error. hp -adaptation would be the most effective adaptation procedure, but the decision between h and p refinement is not trivial.

This work depends on Riemannian metric based anisotropic h -adaptation to efficiently resolve features such as shocks, wakes, and boundary layers with arbitrary orientations. Global re-meshing of the simplex mesh is performed at each adaptation iteration. The element size requests in the adapted mesh are based on a fixed-fraction marking strategy.

With fixed-fraction marking, refinement is requested for a fixed percentage of elements with the largest error while coarsening is requested for a percentage of the elements with the smallest error. The fixed-fraction marking strategy used in this work is distinct from traditional fixed-fraction adaptation based on hierarchical subdivision of elements. The marking strategy is a means to redistribute element sizes within a requested metric field but does not cause a discrete change in the degrees of freedom.

One of the advantages of h -adaptation is it allows for anisotropic mesh refinement. For anisotropic mesh indicators the solution Hessian of Mach number has been used by Venditti and Darmofal [122, 124] for second-order schemes. For a second-order scheme aligning the anisotropic metric with the Hessian equidistributes the interpolation error in the principle metric directions. Fidkowski and Darmofal [45] generalized the Hessian-based analysis to higher-order schemes by basing the principle stretching directions of an element on the maximum $p + 1$ derivative. A more direct approach to mesh adaptation has also been used for anisotropic mesh adaptation by selecting the local mesh refinement of a single element which results in the most competitive subdivision of that element in terms of reduction of the error estimate [26, 51, 66, 101, 117]. An additional method proposed by Leicht and Hartmann [78] uses the inter-element jumps inherent to the DG solution to indicate where anisotropic adaptation is required.

1.3.3 Cut-Cell Mesh Generation

Two details of the solution strategy described above motivate the use of cut-cell mesh generation. The first motivation for cut cells comes from the use of a higher-order discretization, where boundary conditions must contain higher-order information about the geometries they represent. The second motivator for the cut-cell method is adaptation, which requires repeated, reliable, and autonomous mesh generation.

Mesh generation about complex three-dimensional shapes is difficult even for linear (i.e. planar-faced) elements, in particular when high anisotropy is desired near the surface to resolve boundary layers. Mesh generation for boundary layers is sufficiently difficult that many researchers have adopted a hybrid approach. The hybrid approach employs a fixed highly-anisotropic structured boundary layer mesh coupled to an unstructured mesh that fills the computational domain [81, 102, 103]. Even in cases where it is feasible to generate linear

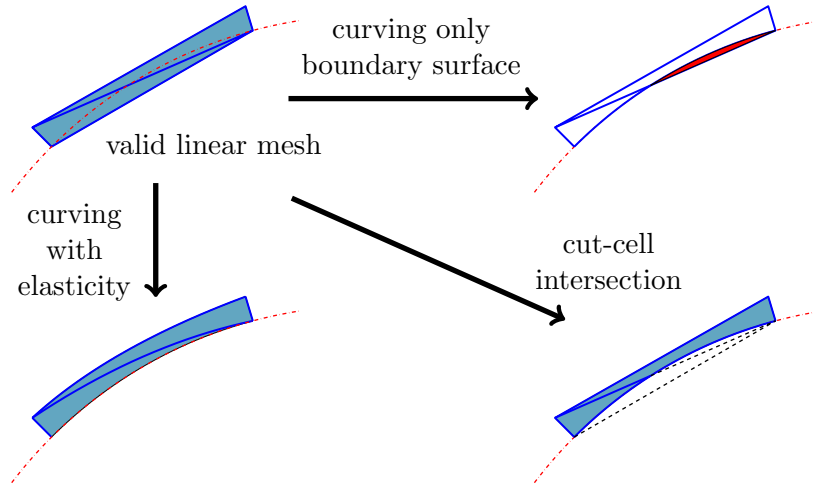


Figure 1-4: Diagram of the options for converting a linear boundary conforming mesh to a mesh containing higher-order geometry information.

boundary-conforming meshes, conversion to a higher-order curved-boundary surface may push through an opposing face as shown in Figure 1-4. One practice to generate a higher-order mesh, also shown in Figure 1-4, is to globally curve a linear boundary conforming mesh with elasticity [93, 98, 107].

A method to tackle the problem of reliably generating meshes of complex geometries with higher-order information is the cut-cell method, shown in Figure 1-4. Purvis and Burkhalter [110] were the first to consider a cut-cell method for a finite volume discretization of the full non-linear potential equations. Purvis and Burkhalter started with a structured Cartesian mesh that did not conform to the geometry and simply “cut” the geometry out. Cut cells allow the grid generation process to become automated, taking a process which was previously human-time intensive and dominated the solution procedure and making it a preprocessing step. While relieving the mesh generation process, the cut-cell method requires an ability to discretize on the arbitrarily shaped cut cells. Purvis and Burkhalter’s method used rectangular/box shaped cells from which the geometry was cut out in a piecewise linear fashion. Although the linear intersections did not provide higher-order geometry, Purvis and Burkhalter laid the foundation for future work with Cartesian cut cells. The full potential equation was also solved using a Cartesian cut-cell method by Young et al.[132] in TRANAIR.

Cart3D, a three-dimensional Cartesian solver for the Euler equations [3], is a current example of the benefits of adding robust cut-cell mesh generation to a flow solver. Cart3D is based on embedding boundaries into Cartesian hexahedral background meshes and has proven capable of handling very complex geometries, like in the space shuttle debris calculations performed by Murman et al.[2]. Work by Nemeč [89–91] has added adjoint-based error estimates and adaptive refinement, which has provided an automated solution procedure for the Euler equations. Along with Cart3D, Cartesian embedded mesh generation has been used extensively in the literature [28, 49, 68].

While providing a robust meshing algorithm, a Cartesian cut-cell mesh limits the achievable directions of anisotropy, making the discretization of arbitrarily-oriented shock waves, boundary layers, or wakes highly inefficient. An application of a Cartesian cut-cell method to the Euler equation for transonic and supersonic flows by Lahur and Nakamura [75, 76] demonstrates the ease in which adaptation can be performed with a Cartesian cut-cell method, yet also the inability for axis aligned anisotropic elements to align with arbitrarily-oriented shock waves. The simplex cut-cell method, introduced by Fidkowski and Darmofal [43–45], offers an autonomous route for generating computational meshes with high arbitrary anisotropy and curved geometry information. Combining the simplex cut-cell method with a higher-order discretization, like the DG method in this work, provides the necessary tools to solve viscous flows over complex geometries. Fidkowski demonstrated the ability of the simplex cut-cell method to solve Euler and Navier-Stokes flows in two dimensions and Euler flows in three dimensions. The method was also used to model a rotor in hover [88].

The cut-cell method is well suited to the DG discretization. DG allows for inter-element jumps of the solution so forming a continuous basis within the computational domain is not necessary. Due to the nature of the cutting procedure the resulting element shapes are arbitrary and the possibility exists for large jumps in element volume across a common face. In order to incorporate cut-cell meshes into a DG discretization, the capability is needed to represent solutions and integrate the residual on arbitrarily shaped elements.

1.4 Thesis Overview

The primary contributions of this work are the following:

- Development of the capability to reliably solve high Reynolds number two-dimensional RANS problems using a higher-order, adaptive, cut-cell method
- Quantification of the impact on solution efficiency (defined as accuracy per degree of freedom) in the transition from boundary-conforming elements to simplex cut cells on a wide range of aerospace problems including subsonic through supersonic conditions and complex geometries
- Analysis of the impact of small volume ratios on linear system conditioning and solution quality, particularly boundary output evaluation, to identify its root cause and develop a method based on the analysis to alleviate the consequences of small volume ratios
- Development of a line-search globalization technique based on the unsteady residual of a pseudo-transient evolution to improve the robustness of non-linear solvers
- Quantification of the impact of randomness on the algorithm for generation integration rules for cut elements and development of integration rules based on canonical shapes where applicable, while otherwise, removal of randomness from the general algorithm for arbitrarily shaped elements
- Development of an adaptation strategy that is less dependent on solution regularity and poor error estimates in under-resolved meshes

There are four primary research groups working on adaptation and higher-order DG discretizations of the RANS equations: the ProjectX team here at MIT, the Hartmann led research group at DLR (the German Aerospace Center), the research group of Fidkowski at University of Michigan, and Bassi's research group at Università di Bergamo. The groups share the common ability to use the DG discretization to perform high-fidelity RANS simulations, but have different methodologies. Currently, the other three research groups rely on structured quadrilateral and hexahedral meshes. The contributions made in this thesis have lead to the unique capability to solve the higher-order DG discretization of the RANS equations on unstructured meshes with simplex cut-cell based adaptation. The unstructured

simplex meshes allow for arbitrarily oriented anisotropy to resolve all flow features uncovered by output-based adaptation, and the cut-cell method provides reliable higher-order geometry representation while decreasing the strain of mesh generation. The simplex meshes reduce simulation cost in terms of degrees of freedom compared to structured meshes for flows with arbitrarily oriented anisotropic features.

The two-dimensional, adaptive, cut-cell solution strategy presented in this thesis has provided a foundation for three-dimensional cut cells for RANS problems. A concern with the simplex cut-cells technique based on linear background meshes was that the resolution of boundary layer features would be inefficient [43]. The results presented in Chapter 7 provide quantifiable evidence that linear based cut-cell meshes can provide equivalent solution efficiency in comparison to boundary-conforming meshes at engineering-required accuracy. The high solution efficiency on the complex two-dimensional problems explored in this thesis provide a motivation for the extension to three-dimensions.

While the contributions have been made for the advancement of a solution strategy for RANS problems, the contributions are intended to be generally applicable to a wide range of problems resulting from the discretization of PDEs. This work relies on the discontinuous Galerkin finite element discretization of the RANS-SA equations presented in Chapter 2. Chapter 3 presents the development of a line-search globalization technique to improve the robustness of non-linear solvers based on pseudo-time continuation. Chapters 4 and 5 describe advancements made to the two-dimensional simplex cut-cell technique. Special attention is paid to the analysis of the impact of small volume ratios which result from the cut-cell method. Chapter 6 reviews the output-based error estimation and adaptation method used in this work. The solution strategy is applied to a wide range of aerospace problems in Chapter 7. Finally, conclusions and ideas for future work are given in Chapter 8.

Chapter 2

Discretization of the RANS-SA Equations

The chapter begins with a brief review of the Reynolds-averaged Navier-Stokes (RANS) equations and the Spalart-Allmaras (SA) turbulence model in Sections 2.1 and 2.2. Section 2.3 shows the spatial discretization and the chapter concludes with Section 2.4, a summary of the shock capturing employed in this work.

2.1 The RANS Equations

The solution to the compressible Navier-Stokes equations for turbulent flows of engineering interest poses a prohibitively expensive problem due to the large range of temporal and spatial scales present in the flows. It is common to solve the Reynolds-averaged Navier-Stokes (RANS) equations which govern the turbulent mean flow. The RANS equations are derived by averaging the Navier-Stokes equations. Favre averaging is used for compressible

flows. The form of the RANS equations in this work are

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i}(\bar{\rho} \tilde{u}_i) = 0, \quad (2.1)$$

$$\frac{\partial}{\partial t}(\bar{\rho} \tilde{u}_i) + \frac{\partial}{\partial x_j}(\bar{\rho} \tilde{u}_j \tilde{u}_i + \bar{p} \delta_{ji}) = \frac{\partial}{\partial x_j} \left[2(\mu + \mu_t) \left(\tilde{s}_{ji} - \frac{1}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ji} \right) \right], \quad i = 1, \dots, d, \quad (2.2)$$

$$\frac{\partial}{\partial t} \left[\bar{\rho} \left(\tilde{e} + \frac{1}{2} \tilde{u}_i \tilde{u}_i \right) \right] + \frac{\partial}{\partial x_j} \left[\bar{\rho} \tilde{u}_j \left(\tilde{h} + \frac{1}{2} \tilde{u}_i \tilde{u}_i \right) \right] = \frac{\partial}{\partial x_j} \left[c_p \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \frac{\partial \tilde{T}}{\partial x_j} \right] + \frac{\partial}{\partial x_j} \left[\tilde{u}_i 2(\mu + \mu_t) \left(\tilde{s}_{ij} - \frac{1}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \right) \right], \quad (2.3)$$

where ρ denotes the density, u_i are the velocity components, p is the pressure, e is internal energy, h is the enthalpy, T is the temperature, $s_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$ is the strain-rate tensor, μ is the dynamic viscosity, μ_t is the dynamic eddy viscosity, Pr is the Prandtl number, Pr_t is the turbulent Prandtl number, d is the spatial dimension, and the summation on repeated indices is implied. The $(\bar{\cdot})$ and $(\tilde{\cdot})$ notation indicates Reynolds-averaging and Favre-averaging.

The RANS equations, Equations (2.1) through (2.3), contain more unknowns than equations requiring closure to solve the system. The remaining unknown, which cannot be computed, is μ_t . μ_t relates the mean flow viscous stresses to the stresses due to turbulent fluctuations. The Spalart-Allmaras turbulence model, described in Section 2.2, closes the RANS system of equations.

To simplify the notation for the remainder of this thesis, the $(\bar{\cdot})$ and $(\tilde{\cdot})$ will be left off. Standard Navier-Stokes flow variables will correspond to their appropriate averaged quantities. For instance, ρ is the Reynolds-averaged density and u_i is the Favre-averaged velocity.

2.2 The SA Turbulence Model

A turbulence model is necessary, in order to close the RANS equations. This work relies on the Spalart-Allmaras (SA) turbulence model [115]. The specific form of the model is based off the work of Oliver [98]. Oliver incorporated modifications to the original SA model to

alleviate issues of negative $\tilde{\nu}$, the working variable for the SA equation. The SA equation is particularly susceptible to negative $\tilde{\nu}$ when employing a higher-order discretizations.

The SA model was selected because of its wide use in the aerospace industry and high regard. The model has accurately simulated attached and mildly separated aerodynamic flows [25, 36, 55, 129].

The model takes the form of a PDE for $\tilde{\nu}$, which is algebraically related to the eddy viscosity, μ_t . The eddy viscosity is given by

$$\mu_t = \begin{cases} \rho\tilde{\nu}f_{v1} & \tilde{\nu} > 0 \\ 0 & \tilde{\nu} \leq 0. \end{cases},$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu},$$

and $\nu = \mu/\rho$ is the kinematic viscosity. Then, $\rho\tilde{\nu}$ is governed by

$$\begin{aligned} \frac{\partial}{\partial t}(\rho\tilde{\nu}) + \frac{\partial}{\partial x_j}(\rho u_j \tilde{\nu}) &= P - D \\ &+ \frac{1}{\sigma} \left[\frac{\partial}{\partial x_j} \left(\eta \frac{\partial \tilde{\nu}}{\partial x_j} \right) + c_{b2} \rho \frac{\partial \tilde{\nu}}{\partial x_j} \frac{\partial \tilde{\nu}}{\partial x_j} \right], \end{aligned} \quad (2.4)$$

where the diffusion constant, η , is

$$\eta = \begin{cases} \mu(1 + \chi), & \chi \geq 0 \\ \mu(1 + \chi + \frac{1}{2}\chi^2), & \chi < 0. \end{cases}, \quad (2.5)$$

the production term, P , is

$$P = \begin{cases} c_{b1} \tilde{S} \rho \tilde{\nu}, & \chi \geq 0 \\ c_{b1} S \rho \tilde{\nu} g_n, & \chi < 0, \end{cases}, \quad (2.6)$$

the destruction term, D , is

$$D = \begin{cases} c_{w1} f_w \frac{\rho \tilde{\nu}^2}{d^2}, & \chi \geq 0 \\ -c_{w1} \frac{\rho \tilde{\nu}^2}{d^2}, & \chi < 0. \end{cases}, \quad (2.7)$$

S is the magnitude of the vorticity, such that

$$\tilde{S} = \begin{cases} S + \bar{S}, & \bar{S} \geq -c_{v2}S \\ S + \frac{S(c_{v2}^2S + c_{v3}\bar{S})}{(c_{v3} - 2c_{v2})S - \bar{S}}, & \bar{S} < -c_{v2}S, \end{cases}, \quad (2.8)$$

and

$$\bar{S} = \frac{\tilde{\nu}f_{v2}}{\kappa^2d^2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}.$$

The remaining closure functions are

$$f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{1/6}, \quad g = r + c_{w2}(r^6 - r), \\ r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2d^2}, \quad g_n = 1 - \frac{f_{g_n}\chi^2}{1 + \chi^2},$$

where d is the distance to the nearest wall, $c_{b1} = 0.1355$, $\sigma = 2/3$, $c_{b2} = 0.622$, $\kappa = 0.41$, $c_{w1} = c_{b1}/\kappa^2 + (1 + c_{b2})/\sigma$, $c_{w2} = 0.3$, $c_{w3} = 2$, $c_{v1} = 7.1$, $c_{v2} = 0.7$, $c_{v3} = 0.9$, and $Pr_t = 0.9$.

In this work, only fully turbulent flows are considered. Hence, the laminar suppression and trip terms from the original SA model are omitted.

The form of the SA model shown in Equation (2.4) is modified from that in [115]. The first modification is the expansion of the original model to compressible flows. The remainder of the modifications handle the case of negative $\tilde{\nu}$. Though the exact solution to Equation (2.4) is for non-negative $\tilde{\nu}$, the discrete solution does not necessarily maintain this property. In fact the solution overshoots which can result from higher-order discretizations on an under-refined mesh, amplifying the occurrence of negative $\tilde{\nu}$ values. Negative $\tilde{\nu}$ values have a strong detrimental impact on the non-linear solution convergence. The complete analysis of the impact of negative $\tilde{\nu}$ and the modifications to correct this behavior can be found in Oliver [98]. The only implemented change made from the model presented by Oliver is the default value of f_{g_n} . The function g_n , and the constant value $f_{g_n} = 10^3$, were originally selected by Oliver to keep $\tilde{\nu}P > 0$ for mildly negative $\tilde{\nu}$ (specifically for $\chi > -\sqrt{1/999}$). Over the course of this work, improved robustness in the non-linear solver is experienced for RANS-SA solutions when f_{g_n} is increased to 10^5 . The increase in f_{g_n} leads to a slightly slower nominal convergence, but superior reliability is experienced.

2.3 Spatial Discretization

The RANS-SA equations can be expressed as a general conservation law given in strong form as

$$\nabla \cdot \mathcal{F}(\mathbf{u}) - \nabla \cdot \mathcal{F}_v(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) \quad \text{in } \Omega, \quad (2.9)$$

where $\mathbf{u} = [\rho, \rho u_i, \rho E, \rho \tilde{v}]^T$ is the conservative state vector, \mathcal{F} is the inviscid flux, \mathcal{F}_v is the viscous flux, \mathbf{S} is the source term, and Ω is the physical domain.

The discontinuous Galerkin finite element method takes the strong form of the conservation laws in Equation (2.9) and derives a weak form. The domain, Ω , is represented by \mathcal{T}_h , a triangulation of the domain into non-overlapping elements κ , where $\bar{\Omega} = \cup \bar{\kappa}$ and $\kappa_i \cap \kappa_j = \emptyset$, $i \neq j$. The set of interior and boundary faces in the triangulation are represented by Γ_i and Γ_b , respectively. The function space of discontinuous, piecewise-polynomials of degree p , \mathcal{V}_h^p , is given by

$$\mathcal{V}_h^p \equiv \{\mathbf{v} \in [L^2(\Omega)]^r \mid \mathbf{v} \circ f_\kappa \in [P^p(\tilde{K}_{\text{refs}})]^r, \forall \kappa \in \mathcal{T}_h\},$$

where r is the dimension of the state vector, P^p denotes the space of polynomials of order p on the reference element \tilde{K}_{refs} , and f_κ denotes the mapping from the reference element to physical space for the element κ . The specific mapping, f_κ used in this work will be detailed in Section 4.5.

To generate the weak form of the governing equations Equation (2.9) is weighted by a test function, $\mathbf{v}_h \in \mathcal{V}_h^p$, and integrated by parts. The weak problem is: find $\mathbf{u}_h(\cdot, t) \in \mathcal{V}_h^p$ such that

$$R_h(\mathbf{u}_h, \mathbf{v}_h) = 0, \quad \forall \mathbf{v}_h \in \mathcal{V}_h^p, \quad (2.10)$$

where

$$R_h(\mathbf{u}_h, \mathbf{v}_h) = R_{h,I}(\mathbf{u}_h, \mathbf{v}_h) + R_{h,V}(\mathbf{u}_h, \mathbf{v}_h) + R_{h,S}(\mathbf{u}_h, \mathbf{v}_h),$$

and $R_{h,I}$, $R_{h,V}$, and $R_{h,S}$ denote the discretizations of the inviscid, viscous, and source terms, respectively.

The discretization of the inviscid terms is given by

$$\begin{aligned} R_{h,I}(\mathbf{w}_h, \mathbf{v}_h) &\equiv - \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \nabla \mathbf{v}_h^T \cdot \mathcal{F}(\mathbf{w}_h) \\ &+ \sum_{F \in \Gamma_i} \int_F (\mathbf{v}_h^+ - \mathbf{v}_h^-)^T \mathbf{H}(\mathbf{w}_h^+, \mathbf{w}_h^-, \vec{n}^+) + \sum_{F \in \Gamma_b} \int_F \mathbf{v}_h^T \mathcal{F}^b \cdot \vec{n}, \end{aligned}$$

where $(\cdot)^+$ and $(\cdot)^-$ denote trace values taken from opposite sides of a face, \vec{n}^+ is the normal vector pointing from $+$ to $-$, \mathbf{H} is a numerical flux function for interior faces, and \mathcal{F}^b is the inviscid boundary flux. The Roe flux [112] is used for the numerical flux, \mathbf{H} . The inviscid boundary flux, \mathcal{F}^b , is evaluated at a boundary state, $\mathbf{u}^b(\mathbf{w}_h, \text{B.C.})$, which can depend on both the interior state and the boundary conditions. The specific implementation of boundary conditions can be found in Oliver [97] and Fidkowski et al. [46].

The viscous terms are discretized using the second method of Bassi and Rebay [16, 17], BR2. Following the method of Bassi and Rebay, the strong form of the conservation laws, Equation (2.9), is written as a system of equations,

$$\nabla \cdot \mathcal{F} - \nabla \cdot \mathcal{Q} = 0 \quad (2.11)$$

$$\mathcal{Q} - \mathcal{F}_v = 0, \quad (2.12)$$

where the viscous flux, \mathcal{F}_v , has a linear dependence on the state gradients such that $\mathcal{F}_v(\mathbf{u}, \nabla \mathbf{u}) = \mathcal{A}(\mathbf{u}) \nabla \mathbf{u}$, and \mathcal{A} is the viscosity matrix. A weak form of the system of equations given in Equations (2.11) and (2.12) is again generated by multiplying the system with test functions, $\mathbf{v}_h \in \mathcal{V}_h^p$ and $\boldsymbol{\tau}_h \in (\mathcal{V}_h^p)^d$, respectively, and integrating by parts to obtain

$$\sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} R_{h,I}(\mathbf{w}_h, \mathbf{v}_h) + \sum_{\kappa \in \mathcal{T}_h} \left[\int_{\kappa} \nabla \mathbf{v}_h^T \cdot \mathcal{Q}_h - \int_{\partial \kappa} \mathbf{v}_h^+ \hat{\mathcal{Q}} \cdot \hat{\mathbf{n}} \right] = 0 \quad (2.13)$$

$$\sum_{\kappa \in \mathcal{T}_h} \left[\int_{\kappa} \boldsymbol{\tau}_h^T \cdot \mathcal{Q}_h + \int_{\kappa} \mathbf{w}_h^T \nabla \cdot (\mathcal{A}^T \boldsymbol{\tau}_h) - \int_{\partial \kappa} (\hat{\mathcal{A}} \mathbf{w})^T \boldsymbol{\tau}_h^+ \cdot \hat{\mathbf{n}} \right] = 0, \quad (2.14)$$

where $(\hat{\cdot})$ denotes numerical flux approximations given discontinuous data across element

faces. By defining $\boldsymbol{\tau}_h \equiv \nabla \mathbf{v}_h$ and integrating by parts the viscous residual can be written as

$$R_{h,V}(\mathbf{w}_h, \mathbf{v}_h) = \sum_{\kappa \in \mathcal{T}_h} \left[\int_{\kappa} \nabla \mathbf{v}_h^T \cdot \mathcal{A} \nabla \mathbf{w}_h + \int_{\partial \kappa} \nabla (\mathbf{v}_h^T)^+ \left(\hat{\mathcal{A}} \mathbf{w} - \mathcal{A}^+ \mathbf{w}_h^+ \right) \cdot \hat{\mathbf{n}} - \int_{\partial \kappa} \mathbf{v}_h^+ \hat{\mathcal{Q}} \cdot \hat{\mathbf{n}} \right]. \quad (2.15)$$

From the BR2 discretization the numerical approximation for the fluxes are $\hat{\mathcal{A}} \mathbf{w} = \mathcal{A}^+ \{\mathbf{w}_h\}$ and $\hat{\mathcal{Q}} = \{\mathcal{A} \nabla \mathbf{u}_h\} - \eta_f \{\tilde{\mathbf{r}}_f(\mathbf{w}_h)\}$. A detailed review of the stability, compactness, and dual consistency of different numerical flux options appears in [8]. With BR2 the viscous discretization becomes

$$\begin{aligned} R_{h,V}(\mathbf{w}_h, \mathbf{v}_h) = & \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \nabla \mathbf{v}_h^T \cdot (\mathcal{A}(\mathbf{w}_h) \nabla \mathbf{w}_h) \\ & - \sum_{f \in \Gamma_i} \int_f \left[\llbracket \mathbf{w}_h \rrbracket^T \cdot \{\mathcal{A}^T(\mathbf{w}_h) \nabla \mathbf{v}_h\} + \llbracket \mathbf{v}_h \rrbracket^T \cdot (\{\mathcal{A}(\mathbf{w}_h) \nabla \mathbf{w}_h\} - \eta_f \{\tilde{\mathbf{r}}_f(\mathbf{w}_h)\}) \right] \\ & - \sum_{f \in \Gamma_b} \int_f \left[(\mathbf{w}_h^+ - \mathbf{u}^b)^T (\bar{\mathbf{n}}^+ \cdot \mathcal{A}^T(\mathbf{u}^b) \nabla \mathbf{v}_h^+) \right. \\ & \quad \left. + \mathbf{v}_h^T \mathcal{F}_v^b \left(\bar{\mathbf{n}} \cdot (\mathcal{A}(\mathbf{u}^b) \nabla \mathbf{w}_h - \eta_f \tilde{\mathbf{r}}_f^b(\mathbf{w}_h)) \right) \right], \end{aligned} \quad (2.16)$$

where \mathcal{F}_v^b is the viscous boundary flux, $\tilde{\mathbf{r}}_f$ and $\tilde{\mathbf{r}}_f^b$ are auxiliary variables, η_f is a stabilization parameter, and the jump, $\llbracket \cdot \rrbracket$, and average, $\{\cdot\}$, operators have to be introduced to simplify the notation. The jump and average operators for scalar variables, θ , and vector variables, ϕ , are given by

$$\begin{aligned} \{\theta\} &= \frac{1}{2}(\theta^+ + \theta^-), & \{\vec{\phi}\} &= \frac{1}{2}(\vec{\phi}^+ + \vec{\phi}^-), \\ \llbracket \theta \rrbracket &= (\theta^+ \bar{\mathbf{n}}^+ + \theta^- \bar{\mathbf{n}}^-), & \llbracket \vec{\phi} \rrbracket &= (\vec{\phi}^+ \cdot \bar{\mathbf{n}}^+ + \vec{\phi}^- \cdot \bar{\mathbf{n}}^-). \end{aligned}$$

The auxiliary variables are defined by the following problems: for each interior face, Γ_i , find $\tilde{\mathbf{r}}_f \in [\mathcal{V}_h^p]^d$ such that

$$\sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \tilde{\mathbf{r}}_h^T \cdot \tilde{\mathbf{r}}_f(\mathbf{w}_h) = \int_{\Gamma_i} \llbracket \mathbf{w}_h \rrbracket^T \cdot \{\mathcal{A}^T(\mathbf{w}_h) \tilde{\mathbf{r}}_h\}, \quad \forall \tilde{\mathbf{r}}_h \in [\mathcal{V}_h^p]^d,$$

and find $\vec{\mathbf{r}}_f^b \in [\mathcal{V}_h^p]^d$

$$\sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \vec{\boldsymbol{\tau}}_h^T \cdot \vec{\mathbf{r}}_f^b(\mathbf{w}_h) = \int_{\Gamma_b} (\mathbf{w}_h^+ - \mathbf{u}^b)^T (\mathcal{A}^T(\mathbf{w}^b) \vec{\boldsymbol{\tau}}_h^+) \cdot \vec{\mathbf{n}}^+, \quad \forall \vec{\boldsymbol{\tau}}_h \in [\mathcal{V}_h^p]^d,$$

for boundary faces.

For all cases in this work the stabilization parameter is set to $\eta_f = 20$. For the BR2 discretization, η_f larger than the number of faces per element implies stability, so an $\eta_f = 20$ is conservative. The conservative value of η_f is selected based on the arbitrary number of faces which can result from the cut-cell mesh generation algorithm presented in Chapter 4.

The source term is discretized using the asymptotically dual consistent or mixed formulation of Oliver [95]. The source discretization is given by

$$R_{h,S}(\mathbf{w}_h, \mathbf{v}_h) = - \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \mathbf{v}_h^T \mathbf{S}(\mathbf{w}_h, \vec{\mathbf{q}}_h), \quad (2.17)$$

where $\vec{\mathbf{q}}_h \in [\mathcal{V}_h^p]^n$ satisfies

$$\begin{aligned} \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \vec{\boldsymbol{\tau}}_h \cdot \vec{\mathbf{q}}_h &= - \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \mathbf{w}_h \nabla \cdot \vec{\boldsymbol{\tau}}_h + \sum_{f \in \Gamma_i} \int_f ([\hat{\mathbf{w}}] \cdot \{\vec{\boldsymbol{\tau}}_h\} + \{\hat{\mathbf{w}}\} [[\vec{\boldsymbol{\tau}}_h]]) \\ &+ \sum_{f \in \Gamma_b} \int_f \mathbf{u}^b \vec{\boldsymbol{\tau}}_h \cdot \vec{\mathbf{n}}, \quad \forall \vec{\boldsymbol{\tau}}_h \in [\mathcal{V}_h^p]^d, \end{aligned} \quad (2.18)$$

and $\hat{\mathbf{w}} = \hat{\mathbf{w}}(\mathbf{w}_h^+, \mathbf{w}_h^-)$ is a numerical flux function. Oliver proved that $\hat{\mathbf{w}}(\mathbf{w}_h^+, \mathbf{w}_h^-) = \{\mathbf{w}_h\}$ provided an asymptotically dual consistent discretization of the source term. The variable $\vec{\mathbf{q}}_h$, in Equation (2.17), can be rewritten in terms of $\nabla \mathbf{w}_h$ and lifting operators. Beginning by integrating Equation(2.18) by parts to give

$$\begin{aligned} \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \vec{\boldsymbol{\tau}}_h \cdot \vec{\mathbf{q}}_h &= \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \vec{\boldsymbol{\tau}}_h \cdot \nabla \mathbf{w}_h + \sum_{f \in \Gamma_i} \left[\int_f [[\hat{\mathbf{w}} - \mathbf{w}_h]] \cdot \{\vec{\boldsymbol{\tau}}_h\} + \int_f \{\hat{\mathbf{w}} - \mathbf{w}_h\} [[\vec{\boldsymbol{\tau}}_h]] \right] \\ &+ \sum_{f \in \Gamma_b} \int_f (\mathbf{w}^b - \mathbf{w}_h) \vec{\boldsymbol{\tau}}_h \cdot \vec{\mathbf{n}}, \quad \forall \vec{\boldsymbol{\tau}}_h \in [\mathcal{V}_h^p]^d. \end{aligned} \quad (2.19)$$

Lifting operators $\vec{\mathbf{r}}_h$ and $\vec{\boldsymbol{\ell}}_h$ can then be defined by: find $\vec{\mathbf{r}}_h(\mathbf{w}_h) \in [\mathcal{V}_h^p]^d$ and $\vec{\boldsymbol{\ell}}_h(\mathbf{w}_h) \in [\mathcal{V}_h^p]^d$

such that

$$\begin{aligned} \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \vec{\tau}_h \cdot \vec{\mathbf{r}}_h(\mathbf{w}_h) &= - \sum_{F \in \Gamma_i} \int_F \llbracket \hat{\mathbf{w}} - \mathbf{w}_h \rrbracket \cdot \{\vec{\tau}_h\} \\ &\quad - \sum_{F \in \Gamma_b} \int_F (\mathbf{w}^b - \mathbf{w}_h) \vec{\tau}_h \cdot \vec{n}, \quad \forall \vec{\tau}_h \in [\mathcal{V}_h^p]^d, \end{aligned} \quad (2.20)$$

$$\sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \vec{\tau}_h \cdot \vec{\ell}_h(\mathbf{w}_h) = - \sum_{F \in \Gamma_i} \int_F \{\hat{\mathbf{w}} - \mathbf{w}_h\} \llbracket \vec{\tau}_h \rrbracket, \quad \forall \vec{\tau}_h \in [\mathcal{V}_h^p]^d. \quad (2.21)$$

Combining Equations (2.19) through (2.21) allows for the state variable $\vec{\mathbf{q}}_h$ to be expressed as

$$\vec{\mathbf{q}}_h = \nabla \mathbf{w}_h - \vec{\mathbf{r}}_h(\mathbf{w}_h) - \vec{\ell}_h(\mathbf{w}_h). \quad (2.22)$$

Finally, Oliver's mixed formulation of the source discretization is obtained when $\vec{\mathbf{q}}_h$ as given in Equation (2.22) is substituted into Equation (2.17),

$$R_{h,S}(\mathbf{w}_h, \mathbf{v}_h) \equiv - \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \mathbf{v}_h^T \mathbf{S}(\mathbf{w}_h, \nabla \mathbf{w}_h - \vec{\mathbf{r}}_h(\mathbf{w}_h) - \vec{\ell}_h(\mathbf{w}_h)). \quad (2.23)$$

2.4 Shock capturing

Shock capturing is performed using the PDE-based artificial viscosity model from Barter [12]. In this model, a shock indicator that measures the local regularity of the solution is used as the forcing term of an elliptic PDE, which in turn generates a smooth artificial viscosity field. The artificial viscosity PDE, which augments the original conservation law, is given by

$$\frac{\partial \epsilon}{\partial t} = \frac{\partial}{\partial x_i} \left(\frac{C_2}{\tau} (\mathcal{M}^{-1})_{ij} \frac{\partial \epsilon}{\partial x_j} \right) + \frac{1}{\tau} \left[\frac{\bar{h}}{p} \lambda_{\max}(\mathbf{u}) S_K(\mathbf{u}) - \epsilon \right] \quad (2.24)$$

where ϵ is the artificial viscosity,

$$\tau = \frac{h_{\min}}{C_1 p \lambda_{\max}(\mathbf{u})}$$

is the time scale based on the maximum wave speed, $\lambda_{\max}(\mathbf{u})$, and the element size, $h_{\min} = (\lambda_{\max}(\mathcal{M}))^{-1/2}$. $\mathcal{M} = \{\mathcal{M}(x)\}_{x \in \Omega}$ is the smooth Riemannian metric tensor field discussed in Section 6.2 which is defined by the tessellation of the mesh, \mathcal{T}_h . The average length scale throughout the domain is given by

$$\bar{h} = (\det(\mathcal{M}))^{-\frac{1}{2d}},$$

and S_K is the shock indicator based on the jump in a scalar quantity across an element face. The jump indicator is cast as,

$$S_k = \frac{1}{|\partial\kappa|} \int_{\partial\kappa} \left| \frac{[c]}{\{c\}} \right| \cdot \hat{\mathbf{n}}$$

where jumps in speed of sound, c , are chosen to locate shocks. The two constants are set to $C_1 = 3.0$ and $C_2 = 5.0$.

Unlike Barter's original equation that used axis aligned bounding boxes to measure the local element sizes, a Riemannian metric tensor measures the local length scale for the PDE [131]. The new formulation provides consistent diffusion of artificial viscosity independent of the coordinate system and enables sharper shock capturing on highly anisotropic elements with arbitrary orientations.

When shock PDE-based capturing is incorporated into RANS-SA system, the shock state, ϵ , is appended to the state vector and an additional source term is included in the system. The strong form of the governing equations, Equation (2.9), becomes

$$\nabla \cdot \mathcal{F}(\mathbf{u}) - \nabla \cdot \mathcal{F}_v(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) + \mathbf{G}(\mathbf{u}) \quad \text{in } \Omega, \quad (2.25)$$

where the state vector is $\mathbf{u} = [\rho, \rho u_i, \rho E, \rho \tilde{v}, \epsilon]^T$, and $\mathbf{G}(\mathbf{u})$ is the source term due to the artificial viscosity equation.

The weak form of the coupled RANS-SA PDE-shock system finds $\mathbf{u}_h(\cdot, t) \in \mathcal{V}_h^p$ such that

$$\begin{aligned} \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} R_{h,I}(\mathbf{u}_h, \mathbf{v}_h) + R_{h,V}(\mathbf{u}_h, \mathbf{v}_h) \\ + R_{h,S}(\mathbf{u}_h, \mathbf{v}_h) + R_{h,G}(\mathbf{u}_h, \mathbf{v}_h) = 0, \quad \forall \mathbf{v}_h \in \mathcal{V}_h^p. \end{aligned} \quad (2.26)$$

The discretization of the shock source term is

$$R_{h,G}(\mathbf{w}_h, \mathbf{v}_h) = - \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \mathbf{v}_h^T \left[\frac{1}{\tau} \left(\frac{\bar{h}}{p} \lambda_{\max} S_k(\mathbf{w}_h) - \epsilon \right) \right].$$

ϵ included in the RANS equations following the physical viscosity model of Persson and Peraire [105], such that the kinematic viscosity is redefined as

$$\nu_{\epsilon} = \nu + \epsilon. \tag{2.27}$$

ν_{ϵ} is used in place of the kinematic viscosity in the RANS equations (Equations (2.1), (2.2), and (2.3)) but not in the SA model, Equation (2.4).

Chapter 3

Non-Linear Solution Technique

One of the primary objectives of this work is to develop a reliable solution strategy for solving the two-dimensional RANS equations. To accomplish that objective, advancements were made to improve the robustness of the non-linear solution technique. This chapter details the standard pseudo-time continuation technique employed to solve non-linear problems in Section 3.1 and presents in Section 3.3 a line-search based globalization technique to increase the sphere of convergence for RANS problems.

3.1 Pseudo-Time Continuation

The non-linear solver employed in this work is based on pseudo-time continuation. The steady-state conservation law given in Equation (2.9) is recast as an unsteady system of equations,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathcal{F}(\mathbf{u}) - \nabla \cdot \mathcal{F}_v(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) \quad \text{in } \Omega,$$

and time integration drives toward the steady-state solution. The spatially discrete problem is cast as the initial value problem of given $U(0)$, find $U(t)$ such that

$$M \frac{dU}{dt} + R_s(U) = 0,$$

where M is the block-diagonal mass matrix,

$$M_{ij} = \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \mathbf{v}_i^T \mathbf{v}_j,$$

and $R_s(\cdot)$ is the spatial residual vector presented in Section 2.3. The i^{th} component of $R_s(\cdot)$ is denoted as

$$[R_s(U)]_i = R_h(\mathbf{u}_h, \mathbf{v}_i).$$

The unsteady terms of the governing equation are included to improve the robustness of the non-linear solver, particularly through initial transients in the solution. For a generic time integrator, using Δt as time steps, the solution procedure can be broken down into three phases [70]:

1. *The initial phase:* U^n is far from the steady state solution and Δt is required to be small. The success of this phase is determined by the stability and accuracy of the temporal integration. Increased accuracy of initial conditions decrease the impact of the initial phase.
2. *The intermediate phase:* The solution is relatively accurate but Δt is still small. The goal of this phase is for Δt to grow without a loss of solution accuracy. In this phase the solution is only relatively accurate and, as Δt grows, it is possible for a single poor solution update to have a large adverse affect on the solution accuracy and the overall convergence of the scheme.
3. *The terminal phase:* Δt is large and the solution is quickly driven to the steady state solution. Only a few iterations are required as Newton convergence is nearly achieved.

Since the goal is to solve for the steady-state solution, temporal accuracy is not a chief concern. Therefore, a first-order backward Euler method is used for time integration. Given a discrete solution, U^n , the solution after one time step, $U^{n+1} = U^n + \Delta U$, is given by solving

$$M^t(\text{CFL})(U^{n+1} - U^n) + R_s(U^{n+1}) = 0, \tag{3.1}$$

where $M^t(\text{CFL})$ is a time-weighted mass matrix such that

$$M_{ij}^t(\text{CFL}) = \sum_{\kappa \in \mathcal{T}_h} \frac{1}{\Delta t_\kappa} \int_\kappa \mathbf{v}_i^T \mathbf{v}_j,$$

and the local time step, Δt_κ , is based on a global CFL number. The local time step based on a global CFL number within each element is

$$\Delta t_\kappa = CFL \frac{h_\kappa}{\lambda_\kappa},$$

where h_κ is a measure of element grid spacing taken as the minimum altitude of the element (For a simplex element the altitude is the straight line through a vertex and perpendicular to the opposite face) and λ_κ is the maximum convective wave speed over the element equal to the magnitude of the velocity plus the speed of sound.

The solution process is marched forward in time until $\|R_s(U^n)\|_2$ is less than a user specified tolerance. In the non-linear solver, the CFL number is updated at each time step based on a physicality check. The physicality check requires that both the density and internal energy, $\rho e = \rho E - \frac{1}{2}\rho(u^2 + v^2)$, are limited to changes of less than 10%. If they change by less than 10% the CFL is increased by a factor of two. If density and internal energy change by 100% or more, then no update is taken, the CFL number is decreased by a factor of ten, and a new solution update, ΔU , is computed by solving Equation (3.1) using the smaller Δt_κ . Otherwise, only a partial update of the solution, limiting density and internal energy changes to 10%, is taken. The non-symmetric updates to the CFL number based on physicality checks are due to the desire to slowly grow the CFL for increased time accuracy in the case of a full update and the need to quickly attain higher time accuracy when no update is taken. The physicality limits on the CFL number are summarized in Table 3.1.

Change in state	Update	CFL change
$\Delta\rho$ and $\Delta\rho e < 10\%$	full: $U^{n+1} = U^n + \Delta U$	$\text{CFL} \leftarrow 2 \cdot \text{CFL}$
$\Delta\rho$ or $\Delta\rho e > 100\%$	none: $U^{n+1} = U^n$	$\text{CFL} \leftarrow \text{CFL}/10$
otherwise	partial: $U^{n+1} = U^n + 10\% \min\left(\frac{\rho^n}{\Delta\rho}, \frac{\rho e^n}{\Delta\rho e}\right) \Delta U$	no change

Table 3.1: Summary of physicality check limits on the global CFL number.

3.2 Pseudo-Time Solution Update

Steady simulations are the objective of this work, thus a single step of Newton’s method can approximately solve Equation (3.1) at each time step [56],

$$\Delta U = U^{n+1} - U^n = - \left(\text{M}^t(\text{CFL}) + \frac{\partial R_s}{\partial U} \Big|_{U^n} \right)^{-1} R_s(U^n). \quad (3.2)$$

The Newton update, ΔU , requires the solution to a large linear system in the form $\mathbf{A}\mathbf{x} = \mathbf{b}$ at every time step, where

$$\mathbf{A} = \text{M}^t(\text{CFL}) + \frac{\partial R_s}{\partial U}, \quad \mathbf{x} = \Delta U, \quad \mathbf{b} = R_s(U^n). \quad (3.3)$$

The matrix \mathbf{A} is referred to as the Jacobian matrix. For the DG discretization the Jacobian matrix has a block-sparse structure with N_e block rows of size n_r , where N_e is the number of elements in the tessellation \mathcal{T}_h and n_r is the number of unknowns per element. In this case $n_r = r \times n_{bf}$, where r is the number of components in the state vector and n_{bf} is the number of basis functions per state. n_{bf} is a function of the solution order, p , and the reference element shape, as shown in Table 3.2. The block rows of the Jacobian matrix contain a non-zero diagonal block, corresponding to the coupling between states within each element, and n_f off-diagonal blocks, corresponding to the coupling between states of neighboring elements. n_f is the number of faces per element (for triangles and quadrilaterals, n_f is 3 and 4 respectively).

p	n_{bf} , Triangle	n_{bf} , Quadrilateral
1	3	4
2	6	10
3	10	16
p	$\frac{(p+1)(p+2)}{2}$	$(p+1)^2$

Table 3.2: Number of basis functions per element, n_{bf} , for a given solution order and reference element.

Due to the size of the Jacobian matrix and its block-sparse structure, an iterative method solves the linear system. As the Jacobian is non-symmetric, a restarted GMRES algorithm is used [38, 113, 118]. In this work, an in-place Block-ILU(0) factorization [39] with block MDF reordering [106] of the Jacobian matrix is the right preconditioner for the GMRES algorithm. When the CFL number in Equation (3.1) is small, the Jacobian matrix is dominated by the block diagonal and the linear system is relatively easy to solve iteratively. Unfortunately, as the time step increases, the coupling between elements becomes increasingly important and the linear system becomes more difficult to solve.

In order to decrease computational expense, an adaptive linear residual criteria is used for each GMRES solve [38, 79]. The idea is based on the fact that when the norm of the spatial residual is large, the accuracy of the linear solve has a limited impact on the performance of the non-linear solver. For example, if the spatial residual norm is $R_s(U^n) = \mathcal{O}(1)$, then driving the linear residual to $\mathbf{r} = \mathcal{O}(10^{-14})$ is uneconomical. Given the linear residual at iteration k of $\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k$, the linear system is solved to a tolerance of:

$$\frac{\|\mathbf{r}^m\|_2}{\|\mathbf{r}^0\|_2} \leq K_A \min \left[1, \left(\frac{\|R_s(U^n)\|_2}{\|R_s(U^{n-1})\|_2} \right)^2 \right],$$

where K_A is a user defined constant set to 10^{-4} for this work and the min is included in case the spatial residual increases at some iteration.

3.3 Line-Search Solution Update Limiting

Though a non-linear solver based on pseudo-time continuation and physicality checks has proven to be successful for a wide range of Euler and Navier-Stokes problems [11, 43, 97],

there are some cases where the non-linear solver fails to converge. Some limitations with the solution procedure are the lack of temporal accuracy and the use of a single Newton iteration at each time step. These approximations provide large time savings to the solution procedure but allow for the possibility of poor solution updates, particularly in cases which exhibit strong non-linearities. In fact, without physicality checks the updates can result in a solution with either non-physical quantities (negative density or pressure for example) or a solution which simply jumps between solution paths in a non-converging nature. The SA turbulence model equation, described in Section 2.2, provides a highly non-linear test to the existing solution process, which it often fails. Without limits on the updates to the SA state in the existing solution procedure, updates from Equation (3.2) for the SA model can be as large as three orders of magnitude relative to the SA working variable.

In an unsteady simulation, the non-linear equation corresponding to backward Euler time integration, Equation (3.1), would be solved iteratively for U^{n+1} which ensures that during each time step the unsteady residual,

$$R_t(U^{n+1}) \equiv \frac{1}{\Delta t} M(U^{n+1} - U^n) + R_s(U^{n+1}),$$

decreases or vanishes completely. Yet, since only a single Newton sub-iteration is used to compute the update, nothing can be said about the unsteady residual at U^{n+1} for the standard pseudo-time continuation procedure.

Line searches are introduced to increase the reliability of the non-linear solver by expanding the global sphere of convergence of Newton’s method. The idea is that given a decent direction, the line search ensures an “acceptable” update, where an “acceptable” update is problem dependent. For a minimization problem the “acceptable” update might be one that decreases the function being minimized [37]. Line searches have been used previously in [6, 41, 56, 69, 70, 73, 74].

In this work, the purpose of using line-search limiting for the update, $\Delta U = U^{n+1} - U^n$, is to ensure a decrease in the unsteady residual at each time step which would exist naturally if Equation (3.1) was solved iteratively. The line-search limiting performs the following operations:

Line-search η	Update	CFL change
$\eta = 1$	full update: $U^{n+1} = U^n + \Delta U$	$\text{CFL} \leftarrow 2 \cdot \text{CFL}$
$\eta < \eta_{\min}$	no update: $U^{n+1} = U^n$	$\text{CFL} \leftarrow \text{CFL}/10$
otherwise	partial update: $U^{n+1} = U^n + \eta \Delta U$	no change

Table 3.3: Summary of line-search limits on the global CFL number.

1. Take ΔU from physicality check
2. Set $\eta = 1$ and $\tilde{U} = U^n + \eta \cdot \Delta U$
3. Compute $R_t(\tilde{U}) = \frac{1}{\Delta t} M \eta \cdot \Delta U + R_s(\tilde{U})$
4. Do while $\left(\|R_t(\tilde{U})\| > \|R_s(U^n)\| \right)$
 - Set $\eta \leftarrow \frac{\eta}{2}$
 - Update $\tilde{U} = U^n + \eta \cdot \Delta U$
 - Compute $R_t(\tilde{U}) = \frac{1}{\Delta t} M \eta \cdot \Delta U + R_s(\tilde{U})$

The line-search algorithm determines η such that the unsteady residual decreases. However, just as in large solutions updates with the physicality check presented in Section 3.1, if the line-search η becomes less than 10%, no update is taken, the CFL is decreased by a factor of two, and a new solution update is computed. Table 3.3 summarizes the CFL number limiting by the line-search η .

A choice of what norm is appropriate to measure the unsteady residual remains for the line search. If the system of equations being solved were non-dimensionalized, such that the residual corresponding to each equation was of similar magnitude, then a single 2-norm of the residual would suffice. However, in the case of the RANS equations, the SA equation residual is difficult to balance relative to the Navier-Stokes equations (which often can be suitably non-dimensionalized). In the case of a single residual norm performing a line search when the equation residuals are not balanced, the line search will only track the equation with the largest residual and have no control over the other equations. For that reason the line search is performed over each equation of the residual individually, except for the momentum equations which are combined due to their similarity. Step 4. of the line-search limit is replaced by

4. Do while $\left(\|R_t^k(\tilde{U})\| > \|R_s^k(U^n)\| \right)$, for $k = \text{mass, mom, energy, SA}$.

In the case of shock problems no line search is performed on the artificial viscosity PDE, Equation (2.24), due to the small size of the time step required to guarantee a decrease in its unsteady residual. Figure 3-1 shows the residual convergence history for three boundary-conforming meshes for a subsonic simulation of the RANS equations over the RAE2822 airfoil ($M = 0.3$, $\alpha = 2.31^\circ$, $Re = 6.5 \times 10^6$). For a coarse grid, Figure 3-1(a), the line search requires five more non-linear iterations to converge the problem. On the other hand, for both of the finer meshes, Figure 3-1(b) and (c), the line search robustly enables the solution to converge to the steady-state answer.

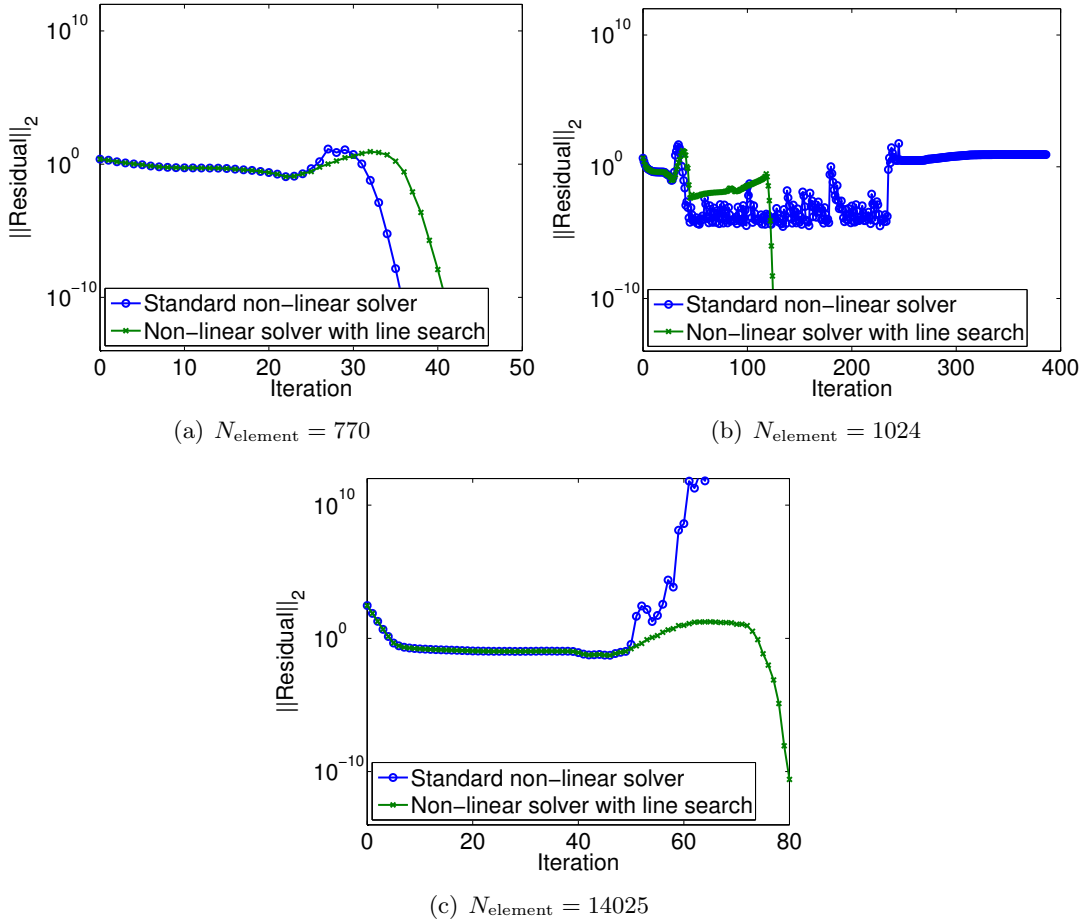


Figure 3-1: Residual convergence for three boundary-conforming meshes for a subsonic simulation of the RANS-SA equations over the RAE2822 airfoil ($M = 0.3$, $\alpha = 2.31$, $Re = 6.5 \times 10^6$).

Chapter 4

Cut-Cell Mesh Generation

In order to incorporate cut-cell mesh generation into the DG discretization presented in section 2.3, the cutting algorithm requires the capability to reliably intersect linear simplex elements with a curved geometry and accurately integrate on the resulting cut elements. The cutting algorithm is performed as a preprocessing step to the solution calculation. The following sections describe advances to the implementation first presented by Fidkowski [43] that allow for reliable simplex cut-cell simulations in two dimensions.

4.1 Geometry Definition

Cubic splines represent the higher-order embedded geometry. Cubic splines are an interpolated fit through a set of spline knots that provide continuity in the slope and the second derivative at each spline knot. The splines yield an efficient representation of higher-order geometry while allowing for simple evaluation of the geometry between spline knots and providing analytic intersection with linear segments. The geometry is a cubic function of a single spline arc-length parameter, s .

The orientation of the splines defines the computational domain. As one travels along the spline in increasing spline arc-length parameter the computational domain is always on the left. Figure 4-1 shows an example of a spline and its knots for a NACA0012 airfoil. The computational domain will be external to the airfoil based on the direction of increasing s .

The simplex background mesh, from which the higher-order geometry is cut, is utilized to define the size of the computational domain and the boundary conditions. The boundaries

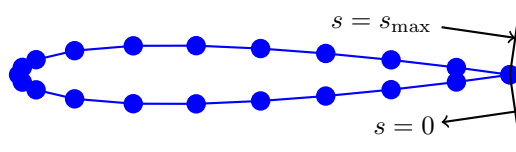


Figure 4-1: Example of spline geometry representation of a NACA0012 airfoil. The spline parameter, s , defines the computational domain to be external to the airfoil.

of the background mesh are used to impose farfield or symmetry-plane boundary conditions. For external flow problems, the background mesh is typically square or rectangular with the airfoil centered in the domain as seen in Figure 4-2 (a). Figure 4-2 (b) shows that cuts through symmetry planes support symmetric flows.

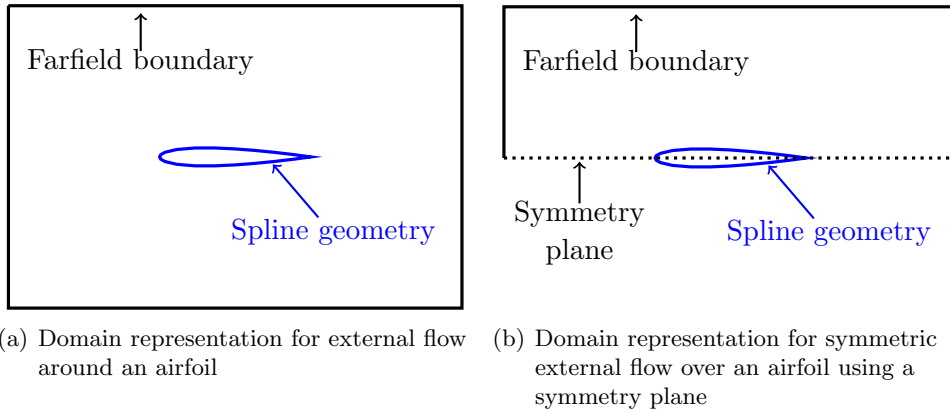


Figure 4-2: Illustration of embedded and farfield domain representation for external flow over an airfoil.

Initial cut-cell meshes can be generated with geometric adaptation. Refinement is based on isotropically refining elements whose bounding box intersects the embedded spline geometry until the refinement reaches a user prescribed number of elements. An element bounding box is defined as the rectangle entirely containing the element that is aligned with the longest segment connection two points in the element and has the smallest area measure. Geometric adaptation provides a crude starting mesh for the output-based error estimation and adaptation procedure described in Chapter 6.

4.2 Intersection Algorithm

Starting with a cubic spline representation of the geometry and a simplex background mesh, the intersection algorithm determines which elements are cut and the exact topology of those cut elements. Point intersections between splines and element faces are computed by solving a cubic intersection problem described in Appendix D of [43]. The point intersections will be referred to as *zerod* objects. *zerod* objects include background grid nodes, spline knots, and spline-face intersections. The illustration in Figure 4-3 provides examples of *zerod* objects and highlights the higher-dimension geometry objects that will be constructed. Table 4.1 provides the relevant information that is stored for each *zerod* type to completely define the different types. A small number of degenerate *zerod* objects shown in Figure 4-4 are also handled. Logic is built into the intersection algorithm to take care of the degenerate cases.

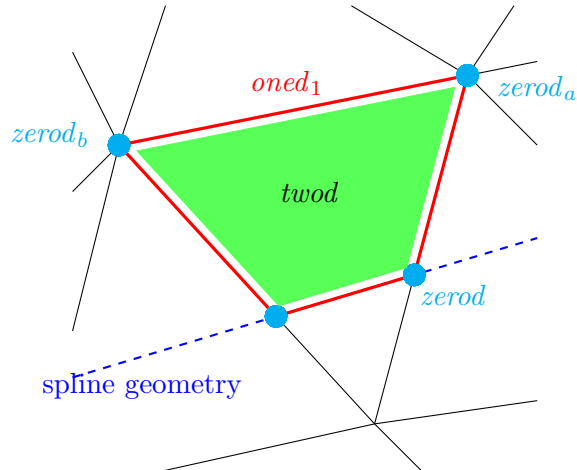


Figure 4-3: Illustration distinguishing different *zerod*, *oned*, and *twod* objects within a cut grid.

From the set of *zerod* objects, a set of *oned* objects is built. *oned* objects uniquely link a pair of *zerod* points. For instance, in Figure 4-3, *oned*₁ is the one dimensional link between the *zerod*_a and *zerod*_b objects. Based on topology, different *oned* objects are referred to by applying different names. An “embedded face” refers to the portion of a spline within a background element. The “embedded face” runs along the spline geometry between spline knots and/or spline-face intersections. The faces of background domain that intersect the

<i>zerod</i> object	Stored information
Grid node	<ul style="list-style-type: none"> • Parent background grid node
Spline knot	<ul style="list-style-type: none"> • Parent spline knot
Spline-face intersection	<ul style="list-style-type: none"> • Face index of background face • Spline index of background spline segment • s_{int}, spline parameter of the intersection • x_{face}, coordinate of the intersection on the face

Table 4.1: Table listing the information that is stored to define the different *zerod* objects.

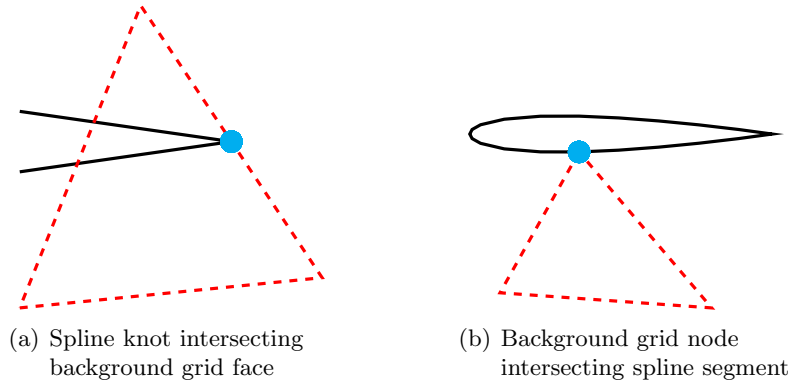


Figure 4-4: Degenerate intersection cases.

spline are referred to as “cut faces.” Figure 4-5 illustrates the different *oned* objects and Table 4.2 lists the information that is stored to define the *oned* objects.

As discussed in Section 4.1, the direction of increasing spline arc-length parameter determines the valid computational domain. This information is used at each spline-face intersection to determine whether the “cut faces” connected to the intersection are in the computational domain. If a “cut face” is not in the computational domain, it is treated as null-*oned*. The null-*oned* information is propagated throughout the null regions, to the right of increasing spline arc-length parameter, by traversing along non-cut background mesh faces or “whole faces.”

After all null-*oned* objects are removed from consideration, the intersection algorithm moves through each background element turning them into *twod* objects. If the background element is uncut and has three non-null *oned*s it is marked as a whole *twod* object. Con-

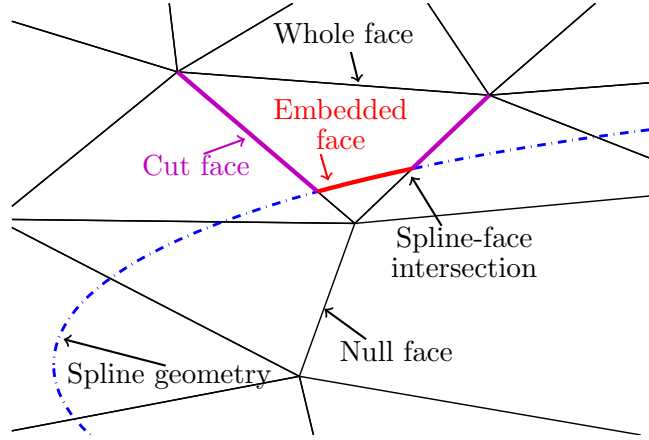


Figure 4-5: Illustration of *oned* objects at the leading edge of an airfoil.

<i>oned</i> object	Stored information
Embedded face	<ul style="list-style-type: none"> • Parent spline segment index • Index of <i>zerod</i> endpoints, iz_0 and iz_1 ($s(iz_0) < s(iz_1)$) • Background element index
Cut face	<ul style="list-style-type: none"> • Parent background face index • Index of <i>zerod</i> endpoints, iz_0 and iz_1
Whole face	<ul style="list-style-type: none"> • Parent background face index • Index of <i>zerod</i> endpoints, iz_0 and iz_1
Null face	Nothing, this face is not relevant for the computational domain

Table 4.2: Table listing the relevant information that is stored to define the different *oned* objects.

versely, if the background element is comprised entirely of null *oned* objects it is skipped completely. For all of the remaining background elements which have “cut faces,” the first step is to collect all *oned*s associated with each background element.

Once all relevant *oned* objects have been associated with a given background element, the next step is determining the precise topology of the element in the computational domain. The generation of loops connecting *oned* objects isolates the valid portion of each background element. Each loop encloses separate regions of the original background element and typically contains no loops within it. The only possibility for a loop to be entirely contained within another is if the entire geometry is contained inside a single background element. In this case,

a reality check is performed to ensure valid cutting. Each loop that is formed is considered a *twod* object, a valid region of background elements inside the computational domain. The loop generation algorithm follows:

Loop Generation

1. From OneD, the list of valid *oned* objects associated with a background element, select an “embedded face,” ie_0
2. Begin a loop with $iz_{start} = \text{OneD}[ie_0].iz_0$, the initial *zerod* on the “embedded face,” ie_0 , such that traversing around the loop keeps the interior on the left
3. Set $ie_{cur} = ie_0$ and $iz_{cur} = \text{OneD}[ie_0].iz_1$
4. Do while ($iz_{cur} \neq iz_{start}$)
 - (a) Add ie to the Loop and mark it as used in OneD
 - (b) Find the unused faces in OneD with

$(\text{OneD}[ie_t].iz_0 == iz_{cur})$	or	$(\text{OneD}[ie_t].iz_1 == iz_{cur})$
• Set $ie = ie_t$		• Set $ie = ie_t$
• Set iz_{cur} to $\text{OneD}[ie_t].iz_1$		• Set iz_{cur} to $\text{OneD}[ie_t].iz_0$
5. If all *oned* objects in OneD are marked as used, loop generation is complete. Otherwise, a new unused “embedded face,” ie_0 , is selected and the process returns to step 2.

The loop generation need not be unique as it is dependent on the selection of ie_0 . However, the manner in which the loops are generated ensures precise knowledge of the cut element topology. A single background element can be cut multiple times leading to two or more *twod* objects. Figure 4-6 provides an example of a background element being cut into two *twod* objects, each defined by separate loops of *oned* objects. The separate cut regions of a multiply-cut element are treated as individual cut cells with their own basis to represent the flow solution and their own integration rules for residual evaluation. Within a cut-cell mesh there is no limit on the number of faces a *twod* object can have. For instance, the background element enclosing the trailing edge in Figure 4-6 has six faces; two “embedded faces,” two “cut faces,” and two “whole faces.” One additional property provided by the loop generation algorithm is the cut-element interior is always to the left as one traverses

along the Loop. The direction of the Loop is shown in Figure 4-6 for the element intersecting the trailing edge of an airfoil.

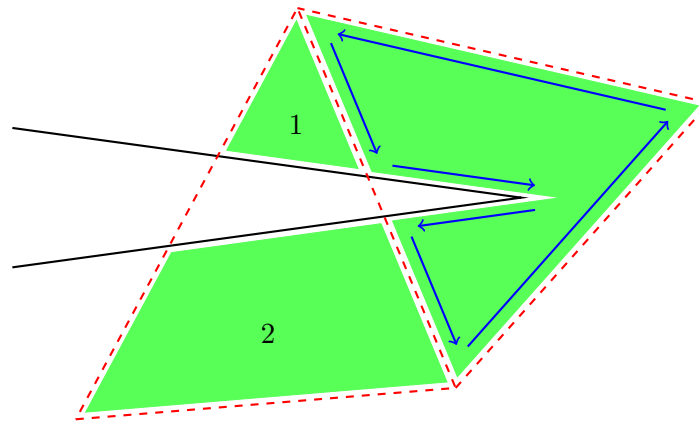


Figure 4-6: Illustration of typical cut elements at an airfoil trailing edge. The left background element straddling the airfoil is treated as two cut elements with each cut element defined by separate loops of *oned* objects. The arbitrarily cut element at the trailing edge is a single cut element with four neighbors. The direction of the Loop is shown for the element at the trailing edge.

The entire cutting and topology building algorithm can be summarized as follows:

1. Build a set of *zerod* objects, including background grid nodes, spline knots, and spline-face intersections
2. Build a set of *oned* objects, where each *oned* object links a pair of *zerod* points (iz_0 and iz_1)
3. Based on the direction of validity for each spline-face intersection, identify null-*oned* objects and propagate that information throughout the non-computational domain
4. For each background element
 - (a) Collect all non-null-*oned* objects associated with the background element
 - (b) Form loops from the list of *oned* objects
 - (c) Each loop becomes a *twod* object or element in the computational domain

Figure 4-7 displays an example of a cut-cell mesh around a NACA0012 where the background elements completely contained by the geometry have been removed. For background

elements that are cut by the spline geometry, the entire element is shown though the solution is only valid within the region of the element inside the computational domain.

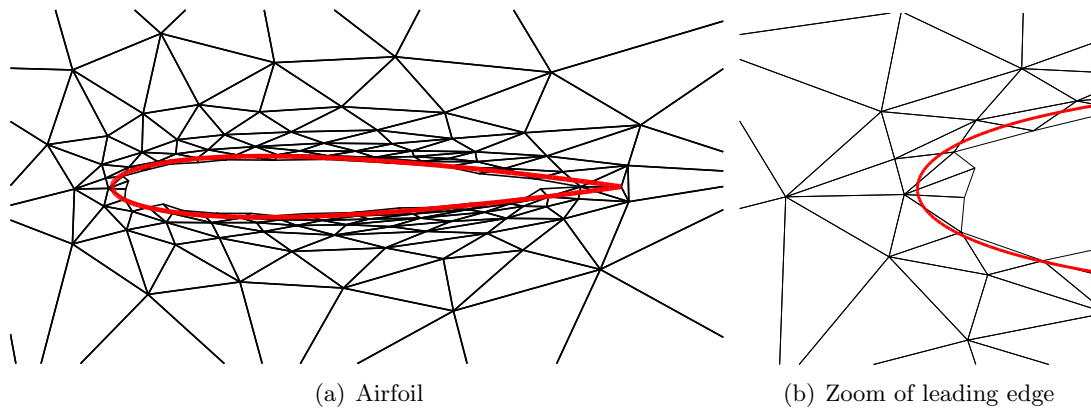


Figure 4-7: Example of a cut-cell mesh for a NACA0012 airfoil. The spline geometry is shown in red.

4.3 Integration for Arbitrary Element Shape

A technique to integrate over arbitrary shapes is required to include cut cells in a DG discretization. One approach for generating integration rules is to subdivide each cut element into a set of possibly-curved sub-triangles. Though this approach would provide an optimal set of quadrature rules for integration in each sub-triangle, it returns to the original problem of meshing with curved boundaries. For that reason a more general approach was developed by Fidkowski [43] for cut elements.

The method presented by Fidkowski was based on using “speckled” sampling points in the cut element and then applying the divergence theorem to compute quadrature-like integration weights such that $\int_{\kappa} f(\mathbf{x}) d\kappa \approx \sum_{q=1}^{N_{quad}} w_q f(\mathbf{x}_q)$. Further details can be found in [43–45, 88].

The general method developed for generating quadrature-like integration rules proves to be sufficient for two-dimensional cut-cell cases, but a very large number of quadrature points are necessary for higher-order approximations. For example, for $p = 5$ cases more than 484 points are needed for element area integration, assuming a required quadrature order of $2p + 1$ [43]. An example set of points is shown in Figure 4-8. Since the quadrature

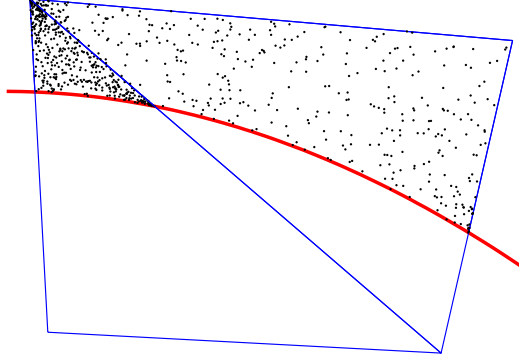


Figure 4-8: An example of the “speckled” 2D integration points for a cut-cell mesh. In order to support $p = 5$ solutions, upwards of 484 points are suggested to adequately cover the interior of the element.

rules (points and weights) are stored for each cut element and the solution must be sampled at each point during residual evaluation, there is a significant added computational cost and memory requirement.

Originally, the “speckling” was performed by, first, randomly selecting a point along the surface of the element boundary, then inwardly projecting rays (randomly at $\pm 15^\circ$ off the inward normal) from that point and randomly selecting a point along the ray between where it enters and exits the element. A large number of sampling points guaranteed coverage of the entire region of an arbitrarily cut element.

A two-dimensional model problem is explored in order to quantify the solution dependence on the randomness in the quadrature rules. The two-dimensional scalar convection-diffusion problem is

$$\nabla \cdot (\vec{V}u) - \frac{1}{Pe} \nabla^2 u = S, \quad (4.1)$$

with a vortex flow velocity field, $V_\theta = \frac{1}{r}$ and $V_r = 0$, and the exact solution is

$$u_e = 1 - \exp\left(\frac{-(r - R)}{\sqrt{\frac{\theta r}{Pe}}}\right).$$

A relatively low Peclet number, $Pe = 100$, is used to limit the necessary anisotropy of the meshes. For all cases, an aspect ratio of 10 is used. Figure 4-9 shows a model of the domain.

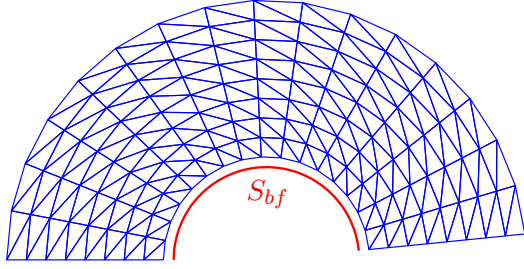


Figure 4-9: An example domain used with the two-dimensional scalar convection-diffusion model problem. For viewing, the cell aspect ratio is set to 1.

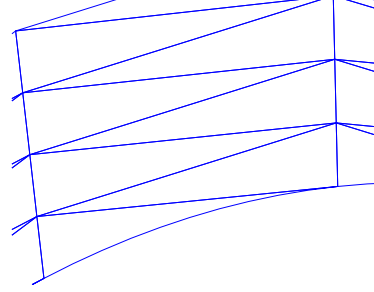


Figure 4-10: Example of a boundary-curved domain. The boundary-conforming domain is globally linear with a single curved boundary on the geometry surface.

In order to measure solution accuracy, the heat flux distribution error, defined as

$$\text{HFDE} = \sqrt{\int_{S_{bf}} \left(\nu \frac{\partial u_h}{\partial n} - \nu \frac{\partial u_e}{\partial n} \right)^2 dS}, \quad (4.2)$$

is used, where S_{bf} is the inner radial surface of Ω labeled in Figure 4-9. The heat flux distribution error is sensitive to oscillations in the local heat flux, $\nu \frac{\partial u}{\partial n}$, along S_{bf} which is important for the assessment of solution quality on the boundary where the impact of cut cells is largest. Global error measures, like L^2 or broken H^1 error, were found to be insensitive to solution fluctuations on S_{bf} . The heat flux distribution error is not a norm, but it provides more information about the solution quality than looking at the error in the heat flux, $\int_{S_{bf}} \nu \frac{\partial u}{\partial n} dS$. Figure 4-11 shows the true heat flux distribution for the convection diffusion model problem.

A cut-boundary-curved mesh shown in Figure 4-10 explores the solution dependence on the randomness in the quadrature rules. The cut-boundary-curved mesh is the result of mapping a linear mesh from (r, θ) space to (x, y) space and then intersecting a curved embedded surface. The embedded surface passes exactly through the boundary nodes of the linear mesh so the result is a mesh with curved faces on the embedded boundary and linear faces elsewhere. The selection of this grid topology allows the integration procedure to be isolated from the influence of small volume ratios which will be discussed in Chapter 5.

Figure 4-12 shows the convergence history of the heat flux distribution error. At each

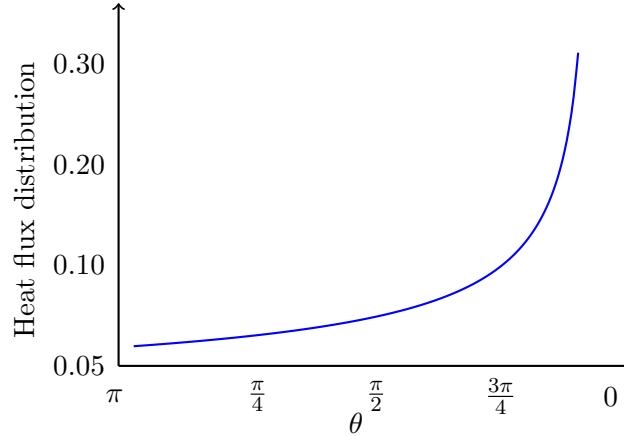


Figure 4-11: Plot of the heat flux distribution along the inner radial boundary of the computational domain.

solution order and grid refinement level, 100 different sets of randomly “speckled” points are utilized for integration rules. For each polynomial order, the convergence of the minimum and maximum heat flux distribution errors are plotted against grid refinement. The minimum and maximum heat flux distribution errors for $p = 1$ through $p = 4$ lay on top of each other showing no noticeable variation in the convergence of the heat flux distribution error as the integration rules change with different sets of 484 random “speckled” points in each cut element. However, when using $p = 5$ polynomials to represent the solution, the heat flux distribution error depends on the set of random points.

Figure 4-13 and Table 4.3 show the variation of the heat flux distribution error at $p = 5$ for each of the four grid refinement levels. Figure 4-13 shows the impact of randomness is strongest for the coarse grid. On the coarsest grid, with $p = 5$, the heat flux distribution error ranges from 6.7×10^{-2} to 5.2×10^{-5} , a factor of over 800. The impact of randomness is less apparent for the refined meshes. For the finer meshes, the difference between the minimum and maximum heat flux distribution error is only a factor of 10. The differing error levels imply that an unacceptable level of randomness exists within the random “speckling” process.

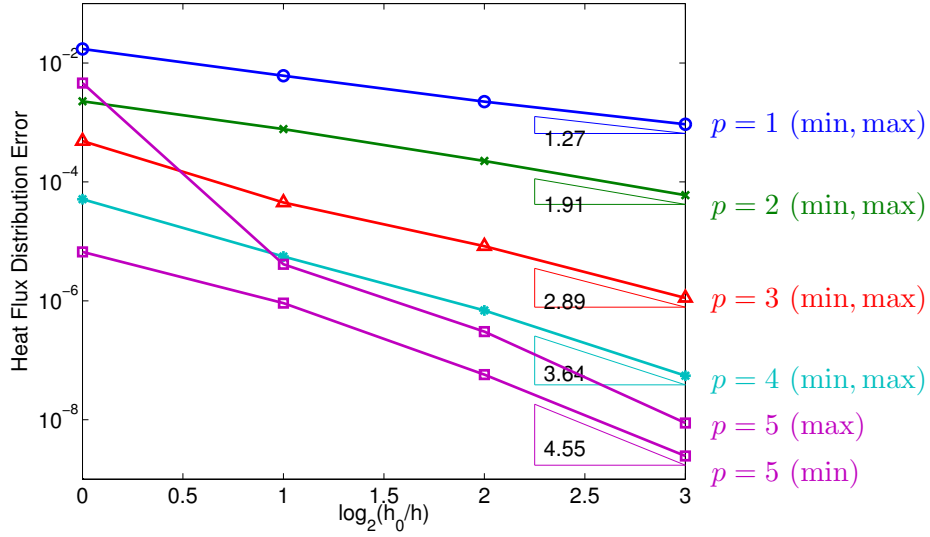


Figure 4-12: Convergence history of the minimum and maximum heat flux distribution error at solution orders 1 through 5, where 100 different sets of “speckled” points are used for integration rules at the four grid refinement levels.

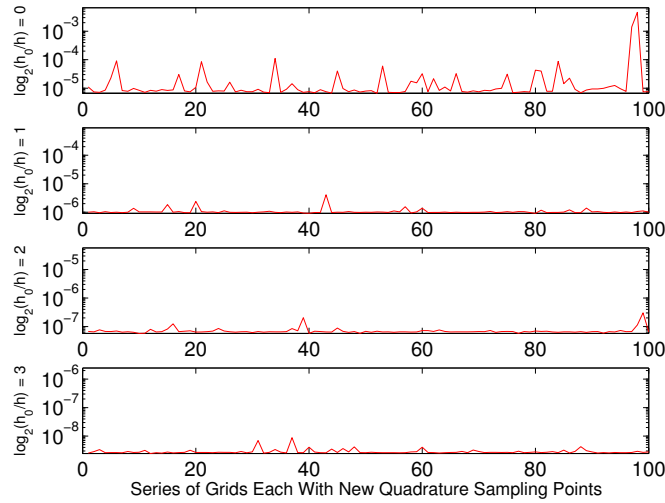


Figure 4-13: Range of heat flux distribution error over a 100 sets of “speckled” points at $p = 5$ for each grid refinement level.

	$\frac{h_0}{h} = 1$	$\frac{h_0}{h} = 2$	$\frac{h_0}{h} = 4$	$\frac{h_0}{h} = 8$
Max over random sets	7.447×10^{-3}	6.529×10^{-5}	3.387×10^{-6}	1.309×10^{-7}
Min over random sets	7.282×10^{-6}	9.063×10^{-7}	5.740×10^{-8}	2.444×10^{-9}

Table 4.3: Table comparing heat flux distribution errors calculated using sets of 484 randomly “speckled” points. All results are for $p = 5$.

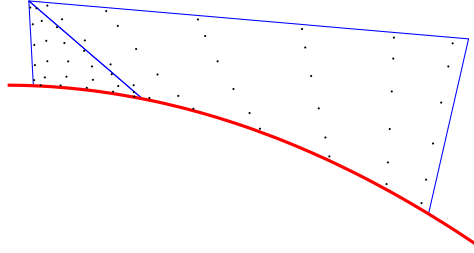


Figure 4-14: Triangles and quadrilaterals are recognizable canonical element shapes and improve the quality of the integration rules. The example elements are the canonical version of the cut elements shown in Figure 4-8 with their canonical quadrature points.

4.4 Canonical Shape Recognition

Recognizing canonical element shapes (i.e. triangles and quadrilaterals) increases the quality of the integration rules. Figure 4-14 provides an example of typical cut elements that are recognized as canonical shapes. In two dimensions, most cut elements have three or four sides. The primary exception is the element at a trailing edge with two embedded surface faces cutting into it, as shown in Figure 4-6. Three- and four-sided cut shapes can be mapped to triangles and quadrilaterals respectively, to utilize standard integration rules in the reference space of the canonical element with provable accuracy in the reference space.

Cut elements become canonical elements via the higher-order geometry information from the faces of arbitrarily cut triangles or quadrilaterals. A Lagrange basis from fifth order polynomials ($q = 5$) represents the curved-canonical elements. Since the Lagrange basis can span multiple spline segments there is a loss of precision in the geometry definition, but as in higher-order boundary-conforming meshes, the error in the geometry is assumed to be less than the error induced by discretizing the flow equations.

To determine the location of the interior Lagrange nodes, a single-element linear elasticity

problem is solved using a Poisson ratio of 1. In the elasticity problem the location of the boundary nodes specifies Dirichlet boundary conditions. The conversion of a three-sided cut element to a curved triangle is shown in Figure 4-15. The canonical-triangle recognition algorithm is:

1. Identify a cut element with 3 faces, Figure 4-15 (a)
2. Create a linear canonical element by connecting the three *zerod* objects, Figure 4-15 (b)
3. Add equally spaced higher-order Lagrange nodes along the spline arc and cut faces, Figure 4-15 (c)
4. Use a single element linear elasticity problem with Dirichlet boundary conditions to solve for interior higher-order node locations, Figure 4-15 (d)
5. Check over element limit points to ensure positive Jacobians throughout

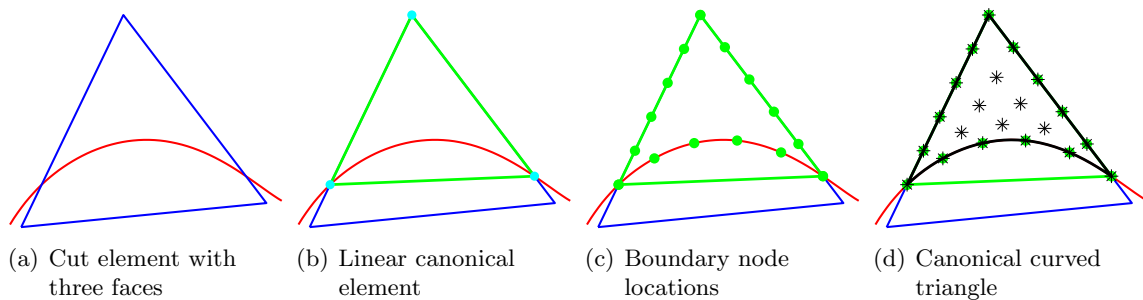


Figure 4-15: Conversion of a three-sided cut element to a higher-order canonical triangle. A $q = 5$ Lagrange basis is used for the illustration.

Step 5 in the conversion to canonical shapes algorithm is to check for positive Jacobians in the mapping from reference space to physical space for the canonical triangle. To check the mapping, the Jacobian is evaluated on the quadrature points used for area and face integration. If the Jacobian at any limit point is negative, the canonical conversion process is abandoned and the element is treated as arbitrarily cut. The recognition of canonical quadrilaterals follows the same procedure.

	$\frac{h_0}{h} = 1$	$\frac{h_0}{h} = 2$	$\frac{h_0}{h} = 4$	$\frac{h_0}{h} = 8$
N_{quad} points	1521	519	545	555
Max over random sets	7.447×10^{-3}	6.529×10^{-5}	3.387×10^{-6}	1.309×10^{-7}
Min over random sets	7.282×10^{-6}	9.063×10^{-7}	5.740×10^{-8}	2.444×10^{-9}
Distributed points	7.391×10^{-6}	1.003×10^{-6}	6.596×10^{-8}	2.653×10^{-9}
Canonical-cut grid	8.025×10^{-6}	9.205×10^{-7}	5.755×10^{-8}	2.810×10^{-9}

Table 4.4: Table comparing heat flux distribution errors calculated using sets of randomly “speckled” points, distributed sampling points, and a canonical-cut grid. The N_{quad} for the “speckled” points is taken from the distributed sampling points to allow for the comparison between the methods. The results are for $p = 5$.

Comparing Figures 4-8 and 4-14 visually shows the difference between the cut elements and the canonical elements. The figures exhibit the required number of quadrature points for integration of eleventh order polynomials between standard cut elements ($N_{quad} = 484$) and cut elements which have been converted to canonical elements ($N_{quad} = 33$ to 49). Table 4.4 compares the integration rules generated from canonical elements to the rules generated from randomly “speckled” sampling points. The heat flux distribution error for the canonical integration rules is close to the minimum heat flux distribution error over the 100 different sets of randomly “speckled” points.

Converting most cut elements to canonical elements leads to the possibility of using more “speckling” points, or a more costly algorithm, for generating integration rules for those elements that cannot be converted. Therefore, with fewer arbitrarily cut elements more time and memory can be spent on each one. In this work the decision was made to increase the number of sampling points to improve the integration rules and reduce the variability. However, an issue arises when more sampling points are used. More sampling points increases the probability of points being located close enough that the conditioning of the QR factorization used to solve for the sampling weights becomes worse. In order to alleviate the potentially poor conditioning of the QR factorization, distributed sampling points are used within the arbitrarily cut elements. The distributed points are generated from a tensor product of Clenshaw-Curtis hierarchal points in the bounding box of the cut elements. The process begins with a set of at least 484 distributed points. A check of each point determines

if it is valid (inside the cut element) or invalid. After the inside/outside check if there are less than 484 points, the Clenshaw-Curtis points are refined and inside/outside checks are performed on the new points. The number of distributed points is $484 \leq N_{\text{quad}} < 1936$.

The last row in Table 4.4 shows the heat flux distribution errors computed using distributed points. The heat flux distribution error using distributed points compares well to the minimum error from the sets of randomly “speckled” points (based on the same N_{quad}) and the integration rules based on canonical shape recognition. However, generating a set of distributed points is more costly than the “speckling” approach of Fidkowski because the distributed points need additional inside-outside checks and still requires storage of the integration rules. In practice, the added expense is worthwhile as the reliability of the solution procedure increases.

It must be noted that the recognition of canonical shapes will be more difficult in three dimensions. In three dimensions, the canonical options will not simply be tetrahedra and quadrilaterals, requiring the addition of transitional elements such as pyramids and prisms. In principle, a similar canonical conversion process will work. First, cut-tetrahedra faces can be identified as triangles or quadrilaterals. The resulting interior volumes can be recognized as a canonical object formed by a set of base volume elements. The proposed three-dimensional canonical conversion process requires meshing standard objects in linear reference space. For instance, a tetrahedron with a node cut off could be recognized as a triangular prism.

4.5 Solution Basis Space

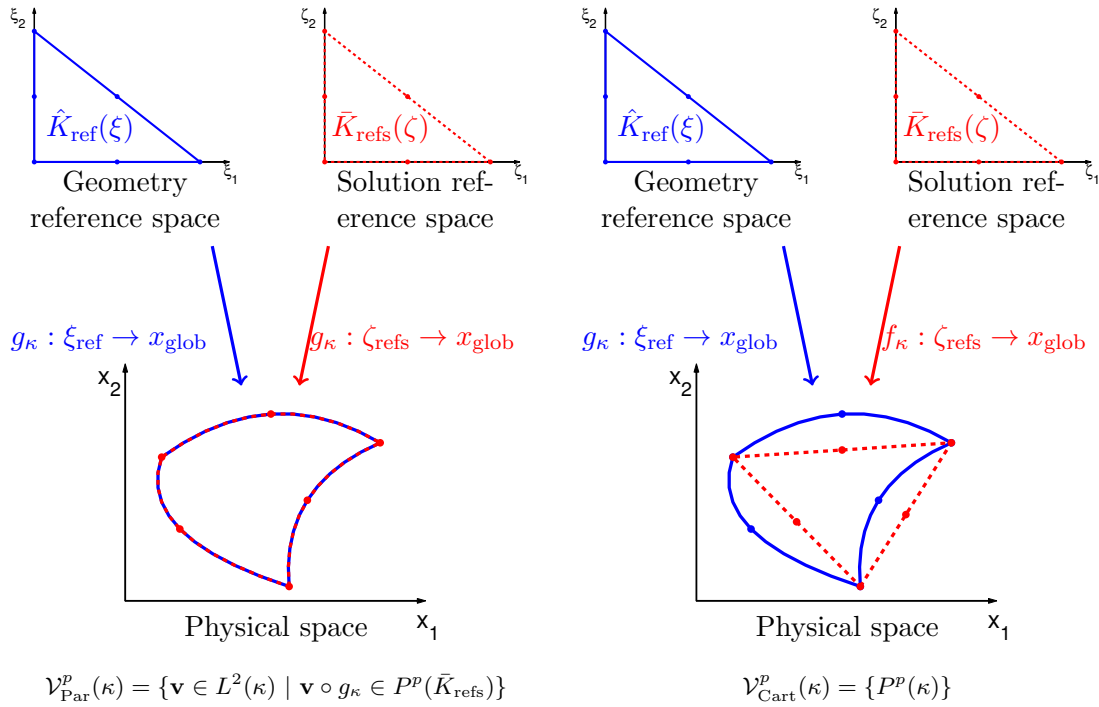
A typical higher-order boundary-conforming mesh is given by \mathcal{T}_h , which is a subdivision of the domain, Ω , into a set of elements each represented as κ , i.e. $\bar{\Omega} = \cup \bar{\kappa}$. The shape of each element, κ , is defined by a coordinate transformation from the reference element, \hat{K}_{ref} , to the physical space, through $g_\kappa \in [P^q(\hat{K}_{\text{ref}})]^d$, where P^q denotes the space of polynomials of order q and d is the physical dimension.

In finite element methods the solution within each element is defined using basis functions. The basis functions are obtained by mapping some function defined on the solution reference element, \bar{K}_{refs} , to the physical element. In this work, the solution is represented

with polynomials of order p in the solution reference element, $P^p(\bar{K}_{\text{refs}})$. For parametric bases, the mapping of the approximation function from the reference space to the physical space is the same as the coordinate mapping, g_κ , as shown by Figure 4-16(a). The g_κ mapping of the solution reference space to the physical space results in a function space of $\mathbf{v}_h \in \mathcal{V}_{\text{Par}}^p(\kappa)$ where

$$\mathcal{V}_{\text{Par}}^p(\kappa) = \{\mathbf{v} \in L^2(\kappa) \mid \mathbf{v} \circ g_\kappa \in P^p(\bar{K}_{\text{refs}})\}.$$

The basis functions are not polynomial in physical space, but allow for easy implementation of strongly enforced Dirichlet conditions on the boundary of the computational domain.



(a) Parametric element mapping where the solution mapping is the same as the coordinate mapping g_κ (b) Cartesian element mapping where the solution mapping, f_κ , is different from the coordinate mapping, g_κ

Figure 4-16: Maps for element and solution representation.

When using a DG finite element discretization and weakly enforced boundary conditions, another possibility for the solution space exists. The solution space can be formed on linear shadow elements creating a Cartesian basis. The shadow elements are obtained by truncating

the $q > 1$ portion of g_κ resulting in the mapping f_κ , an affine mapping from the solution reference space to linear shadow element. The Cartesian basis functions support polynomials in physical space such that $\mathbf{v}_h \in \mathcal{V}_{\text{Cart}}^p(\kappa)$ where

$$\mathcal{V}_{\text{Cart}}^p(\kappa) = \{P^p(\kappa)\},$$

and can also be expressed as polynomials in the reference element, \bar{K}_{refs} :

$$\mathcal{V}_{\text{Cart}}^p(\kappa) = \{\mathbf{v} \in L^2(\kappa) \mid \mathbf{v} \circ f_\kappa \in P^p(\text{Ext}_\kappa(\bar{K}_{\text{refs}}))\},$$

where $\text{Ext}_\kappa(\bar{K}_{\text{refs}})$ is the extension of \bar{K}_{refs} . The extension on \bar{K}_{refs} is necessary because the physical element can extend outside the linear shadow element or truncate it. When points are located in κ , but outside the shadow element, and are mapped to the reference element, they lie outside of \bar{K}_{refs} but within $\text{Ext}_\kappa(\bar{K}_{\text{refs}})$. By definition, the extension of \bar{K}_{refs} is

$$\text{Ext}_\kappa(\bar{K}_{\text{refs}}) = \{\zeta \in \mathbb{R}^d \mid \zeta = f_\kappa^{-1}(g_\kappa(\xi)), \xi \in \hat{K}_{\text{ref}}\}.$$

Figure 4-16(b) shows the g_κ mapping of the coordinates and the f_κ mapping of the linear shadow element from the reference elements, \hat{K}_{ref} and \bar{K}_{refs} .

In the context of the cut-cell method, Cartesian functions provide an avenue to represent solutions on arbitrarily cut elements. On the arbitrarily cut elements a polynomial mapping from the reference space (simplex or quadrilateral) to the physical space is not guaranteed to exist and a parametric basis cannot be used. In place of the parametric mapping each cut element is associated with a linear shadow element and a Cartesian basis represents the solution. The two-dimensional scalar convection-diffusion problem, Equation (4.1), is used to examine the effect on solution accuracy of a Cartesian basis compared to a parametric basis.

Two different mesh families measure the difference between the Cartesian and parametric bases. The first mesh family, shown in Figure 4-17(a), is formed by globally mapping a $q = 5$ mesh in (r, θ) to (x, y) , resulting in a mesh where all elements are curved. The second mesh family, shown in Figure 4-17(b), is generated to simulate cut elements. To do this a linear mesh is mapped from (r, θ) to (x, y) , then only the inner radial boundary face is curved.

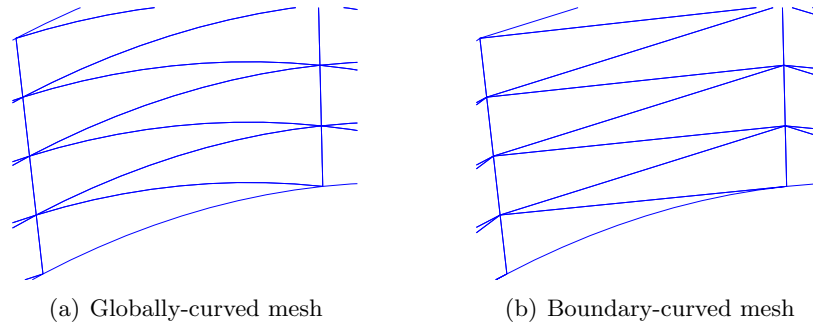


Figure 4-17: Two mesh families used to examine the effect of a Cartesian basis compared to a parametric basis on solution accuracy. For viewing the aspect ratio is set to 1.

Thus, the second mesh family can be thought of as a linear background mesh with the curved surface “cut” out of it. The second mesh family is the boundary-conforming equivalent to the mesh used to evaluate integration rules in Sections 4.3 and 4.4.

Figure 4-18 shows the heat flux distribution error convergence comparison between parametric and Cartesian bases on both globally-curved meshes and boundary-curved meshes. Considering first the globally-curved mesh, the Cartesian basis results in slightly higher heat flux distribution errors and worse convergence rates in the asymptotic range, but the heat flux distribution errors on the initial grid for the Cartesian basis are actually lower. Both bases result in the same answer and are interchangeable for the globally-curved mesh without an effect on accuracy. Figure 4-18(c,d) shows the convergence of the heat flux distribution error on the boundary-curved mesh. For the boundary-curved mesh, the Cartesian basis, Figure 4-18(d), performs significantly better than the parametric functions, Figure 4-18(c) in terms of convergence rate and absolute error level. The parametric functions with the boundary-curved mesh, Figure 4-18(c), result in both poor accuracy and low convergence rates. This implies, for cut elements recognized as canonical shapes, that Cartesian bases should be used for solution representation.

In Figure 4-18, an important comparison between a boundary-conforming method and a cut-cell method can be made between the globally-curved case with parametric functions, Figure 4-18(a), and the boundary-curved case with Cartesian functions, Figure 4-18(d). The boundary-curved Cartesian functions result in marginally inferior convergence rates compared with the globally-curved parametric basis. However, the difference in convergence

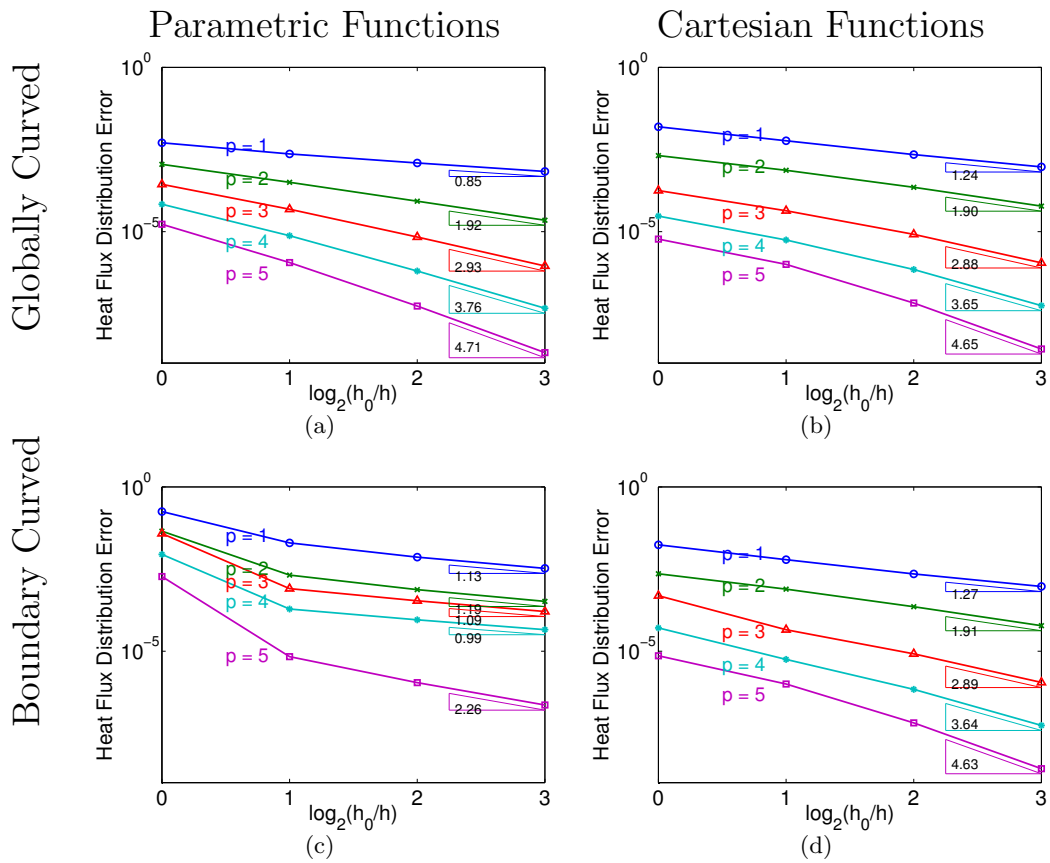


Figure 4-18: Comparison of the convergence in the heat flux distribution errors for cases with parametric and Cartesian approximation functions on globally curved higher-order meshes and globally linear meshes with a single curved boundary. The plots indicate, although there is a small deterioration in the error and rates with the Cartesian functions, the Cartesian functions still perform well at higher order, even in boundary-curved meshes.

rates does not lead to a significant discrepancy in accuracy between the globally-curved case with parametric functions and the boundary-curved case with Cartesian functions. The similarity in the accuracy of the two approximation functions signifies quantifiable evidence that high quality solutions can be generated with Cartesian approximation functions for the cut-cell method.

For application to arbitrarily cut elements, three options can determine appropriate linear shadow elements for use with the Cartesian basis functions. The three options in descending order of preference are:

1. The $q = 1$ portion of g_κ from canonical shape recognition – used for all canonical elements
2. The parent linear background element – preferred when the arbitrarily cut element area accounts for more than 50% of the background element area
3. The half of a bounding box (oriented for tightest fit) that contains more of the cut element – used when the first two options are unavailable

Figure 4-19 gives examples of each of the shadow element options when cutting elements around the trailing edge. Three cut elements are shown along with the shadow element options which can represent the solution in each cut element. Little to no variation in the heat flux distribution error or L^2 error norm is observed when comparing the different shadow element options. The hierarchy of shadow element options is set up in an effort to provide the best overlap between the shadow element and the cut element.

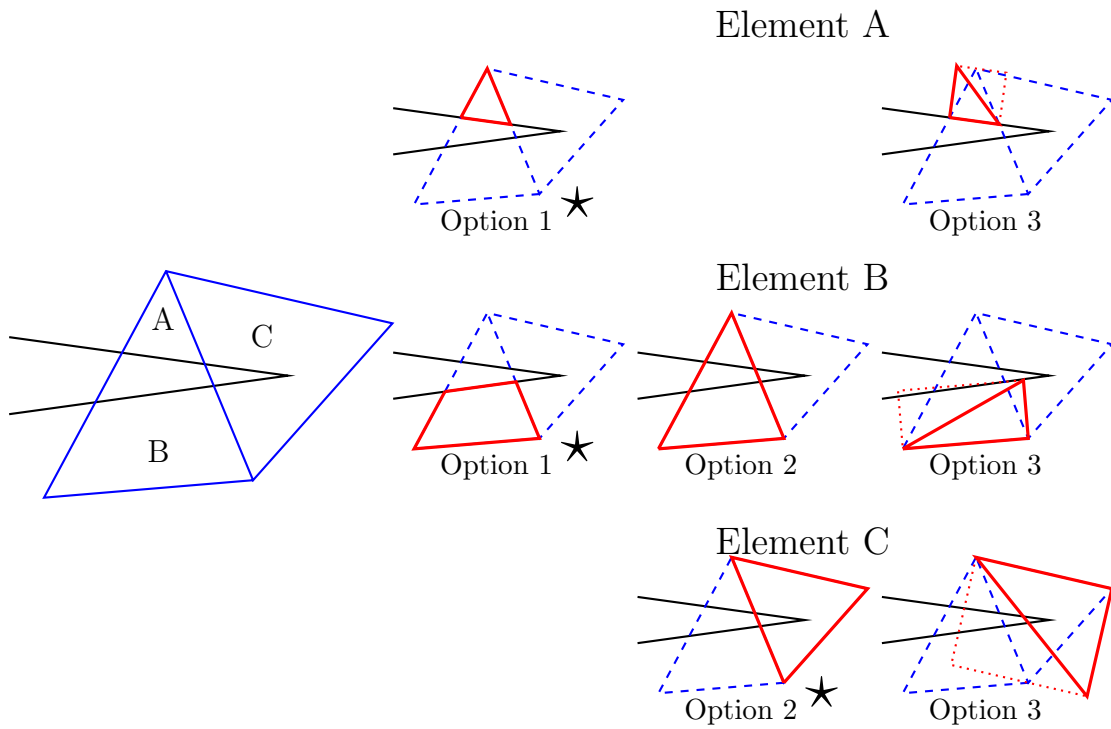


Figure 4-19: Illustration of linear shadow element options from typical cut elements at an airfoil's trailing edge. The ★ indicates the preferred option given the element type.

Chapter 5

Small Volume Ratios

In the cut-cell method, small volume ratios occur when an arbitrarily small cut element is next to a much larger neighbor as shown in Figure 5-1. Small volume ratios have two detrimental effects. First, small volume ratios result in poor output evaluation of derivative quantities along a cut boundary and the quality of estimation does not improve with mesh refinement. Second, small volume ratios cause the linear system to be ill conditioned.

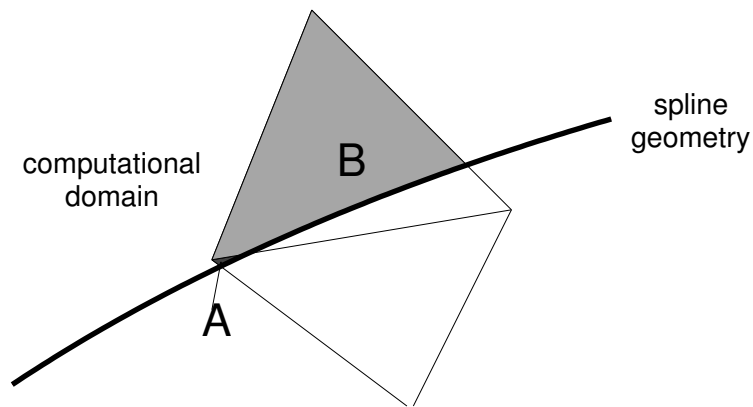


Figure 5-1: Example of a small volume ratio. Usually, small volume ratios occur when a grid node is just inside the computational domain.

A one-dimensional reaction-diffusion problem demonstrates the impact of small volume ratios. The problem is governed by

$$\begin{aligned} u - \nu \frac{\partial^2 u}{\partial x^2} &= f(x) \quad \text{in } \Omega = (0, 1), \\ u &= g \quad \text{on } \partial\Omega, \end{aligned} \tag{5.1}$$

where $\nu > 0$, $f(x)$ is specified such that the exact solution is $u(x) = \sin(2\pi x) - x + 1$, shown in Figure 5-3(a), and g provides the boundary conditions. A DG finite element method is used, where the viscous flux is computed using the second method of Bassi and Rebay [16]. The domain, Ω , represented discretely by \mathcal{T}_h , is split into a set of elements, κ , where $\bar{\Omega} = \cup \bar{\kappa}$ and the faces are represented by Γ . A function space of piece-wise, discontinuous polynomials of degree p is defined as

$$V_{h,p} \equiv \{v \in L^2(\Omega) \mid v|_{\kappa} \in P^p(\kappa), \forall \kappa \in \mathcal{T}_h\},$$

where $P^p(\kappa)$ denotes the space of degree p polynomials on element κ . The variational problem is to find $u_{h,p} \in V_{h,p}$ such that

$$a_{h,p}(u_{h,p}, v_{h,p}) = \langle f, v_{h,p} \rangle, \quad \forall v_{h,p} \in V_{h,p}, \quad (5.2)$$

where $f \in V'_{h,p}$ and $\langle \cdot, \cdot \rangle$ denotes the duality pairing. The bilinear form is given by

$$\begin{aligned} a_{h,p}(w_{h,p}, v_{h,p}) &= \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \nu \nabla v_{h,p} \cdot \nabla w_{h,p} + w_{h,p} v_{h,p} \\ &\quad - \sum_{\Gamma \in \mathcal{T}_h} \int_{\Gamma} \{\nabla v_{h,p}\} \cdot \llbracket w_{h,p} \rrbracket + \llbracket v_{h,p} \rrbracket \cdot \{\nabla w_{h,p}\} \\ &\quad - \sum_{\Gamma \in \mathcal{T}_h} \int_{\Gamma} \llbracket v_{h,p} \rrbracket \cdot r_{h,p}^{\Gamma}(\llbracket w_{h,p} \rrbracket), \end{aligned} \quad (5.3)$$

where the lifting operator, $r_{h,p}^{\Gamma} \in [V_{h,p}]^d$, is defined as

$$\sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \tau \cdot r_{h,p}^{\Gamma}(\llbracket w_{h,p} \rrbracket) = - \int_{\Gamma} \{\tau\} \cdot \llbracket w_{h,p} \rrbracket, \quad \forall \tau \in [V_{h,p}]^d.$$

In this model problem, the forcing term is given by

$$\langle f, v_{h,p} \rangle = \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} f v_{h,p}.$$

The volume ratio is defined as

$$\text{VR}_i = \frac{h_i}{\max_{j \in \text{neighbor}_i} (h_j)},$$

h_i is the length of element i , and $j \in \text{neighbor}_i$ is defined as all elements j sharing a common face with element i . The model domain has uniformly sized elements with one small element on the left boundary as shown in Figure 5-2. Thus, the critical volume ratio for the model problem is

$$\text{VR} \equiv \text{VR}_1 = \frac{h_1}{h_2}.$$

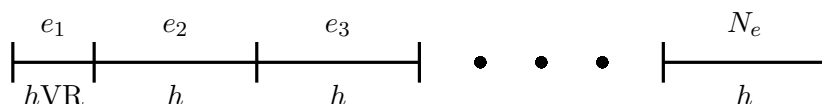


Figure 5-2: Diagram of mesh when grid has uniform h except for the first element where $h_1 = h\text{VR}$.

Figure 5-3(a) shows the exact solution and the finite element solution with $\text{VR} = 1$ and $\text{VR} = 10^{-8}$. Small volume ratios have a negligible impact on the solution. However, in looking at the derivative of the solution in Figure 5-3(b), the effect of the small volume ratio on the solution is noticeable at the left boundary, where an oscillation is present for a solution with $\text{VR} = 10^{-8}$. Since the impact of the element with the small volume ratio is difficult to observe in physical space, the derivative of the solution is plotted for a $\text{VR} = 10^{-8}$ in the reference space of the leftmost element in Figure 5-4. In the element reference space the oscillation in the derivative is clear.

The L^2 error of the solution, $\|u_{h,p} - u\|_{L^2(\Omega)}$, is unaffected by small volume ratios. Figure 5-5 shows the L^2 error convergence at different critical volume ratios and polynomial solution orders with only small changes evident due to volume ratio. The solution error can also be measured in a broken H^1 norm defined by

$$\|v\|_{H^1(\Omega, \mathcal{T}_h)}^2 \equiv \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} (v^2 + v_x^2).$$

In this norm, as shown in Figure 5-6, the error is also unaffected by small volume ratios.

Though the global error measures appear independent of small volume ratios, derivative outputs at the boundary are dependent upon them. Using a dual consistent discretization

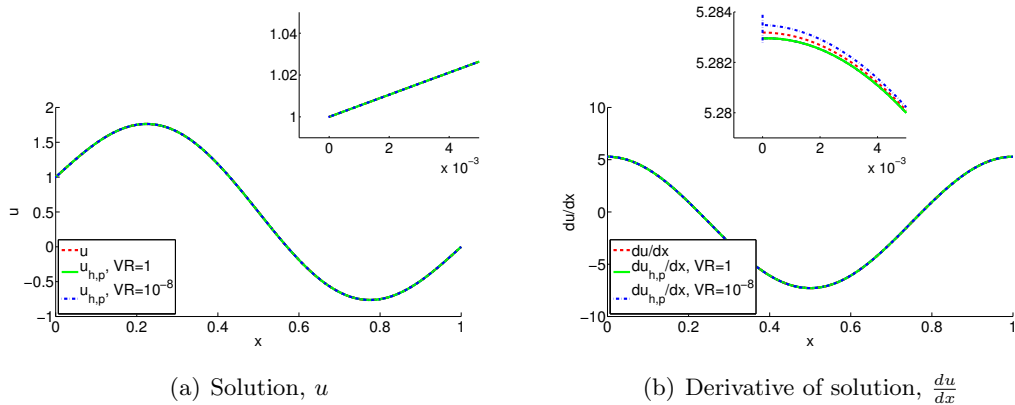


Figure 5-3: Plot of solution and its derivative for the one-dimensional model problem. The exact solution is plotted along with computed solutions for $N_{\text{element}} = 16$, $p = 3$ and $VR = 1$ and $VR = 10^{-8}$. The inset figures show the solution at the left boundary.

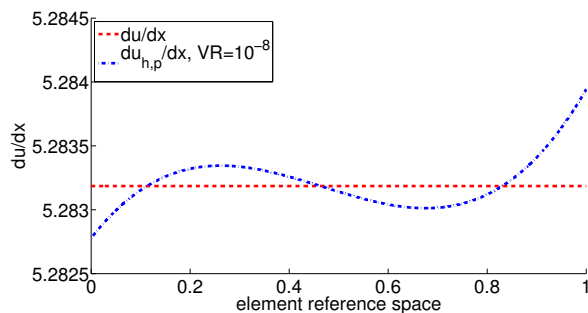


Figure 5-4: Derivative of the solution for the one-dimensional model problem plotted in the reference space of the leftmost element in the domain with a $VR = 10^{-8}$, $N_{\text{element}} = 16$, $p = 3$.

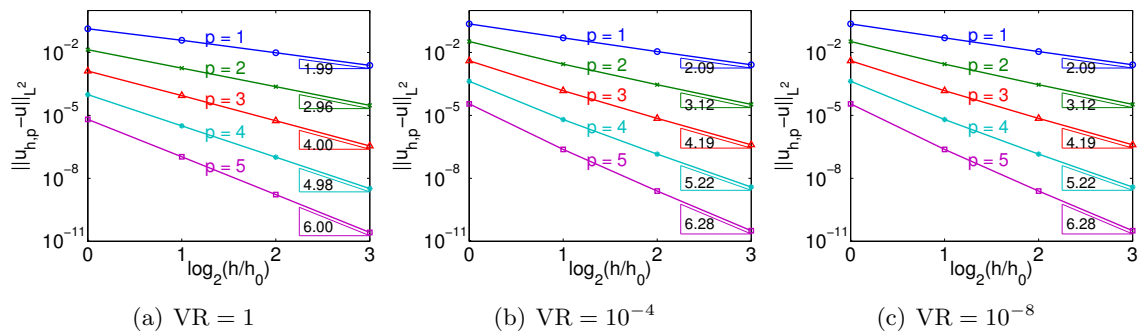


Figure 5-5: The convergence of the L^2 solution error with varying critical volume ratio. Due to the tiny size of the element with the critical volume ratio, the small volume ratio has no impact on the L^2 error.

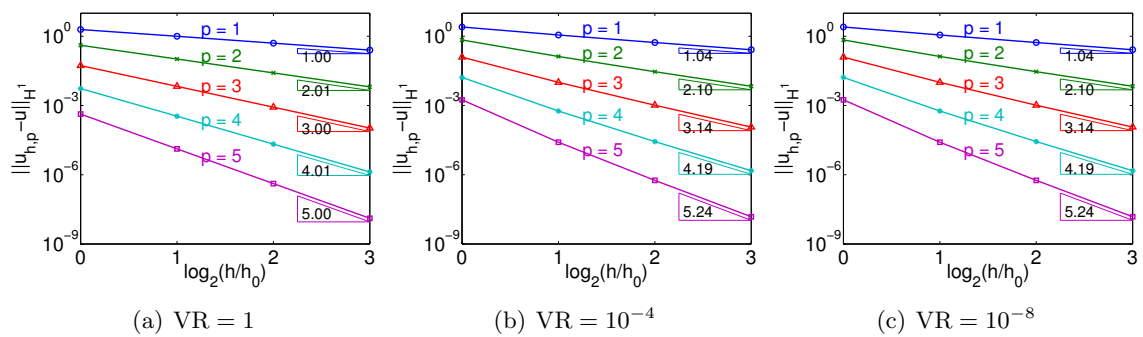


Figure 5-6: The convergence of the broken H^1 solution error with varying critical volume ratio. The critical volume ratio has no impact on the H^1 error.

of Equation (5.1), superconvergence at a rate of h^{2p} is expected for the output, $J(u) = \nu \frac{du}{dx} \Big|_{x=0}$ [53, 60, 99]. The expected rate of convergence comes from

$$\begin{aligned} J(u) - J_{h,p}(u_{h,p}) &\leq C \|u - u_{h,p}\|_E \| \psi - \psi_{h,p} \|_E \\ &\leq \mathcal{O}(h^p) \mathcal{O}(h^p) \end{aligned}$$

where the energy norm is given by $\|v\|_E^2 = a_{h,p}(v, v)$, and ψ is the solution to the dual problem. Figure 5-7 shows that even at volume ratios of $\mathcal{O}(10^{-4})$, the error in the boundary output, $J(u) = \nu \frac{du}{dx} \Big|_{x=0}$, is significant and cannot be neglected. The small volume ratios eliminate all the benefits of the higher-order DG discretization. In this case $\nu \frac{du}{dx} \Big|_{x=0}$ is computed dual consistently using the lifted numerical flux as

$$J_{h,p}(u_{h,p}) = \nu (\nabla u_{h,p} + r_{h,p}^\Gamma(\llbracket u_{h,p} \rrbracket)).$$

Figure 5-6 shows that $\|u - u_{h,p}\|_E$ converges at the expected rate of h^p . The impact of small volume ratios is therefore on $\|\psi - \psi_{h,p}\|_E$ which leads to a loss of superconvergence in $J(u)$.

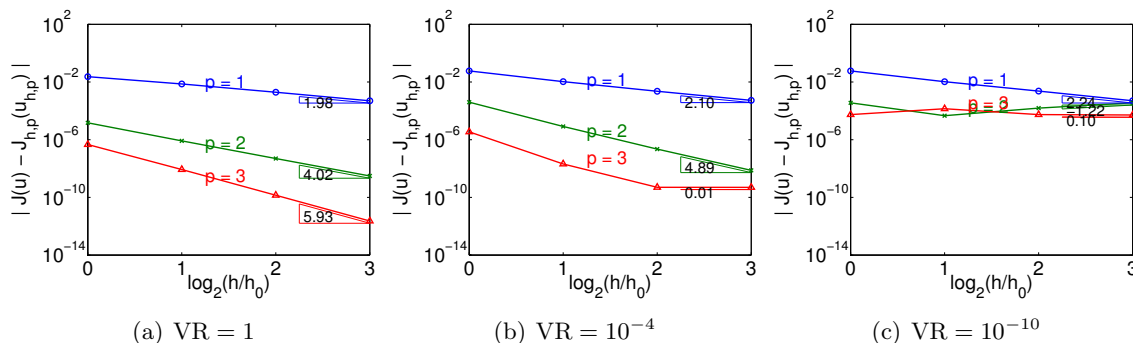


Figure 5-7: The convergence of the error in the output $J(u) = \nu \frac{du}{dx} \Big|_{x=0}$ for a range of volume ratios for the one-dimensional model problem, Equation (5.1).

A second impact of small volume ratios is their effect on the conditioning of the linear system. The consequence of small volume ratios is seen in the condition number of the stiffness matrix for the one-dimensional model problem, Equation (5.2). Figure 5-8 shows the variation in the condition number of the stiffness matrix versus h (the element size of all but the left most element in the one-dimensional model domain). As expected for a

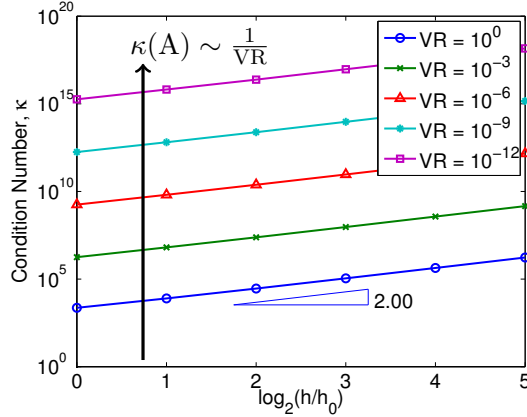


Figure 5-8: Plot showing the variation of the condition number versus element size and volume ratio for the one-dimensional model problem.

diffusion problem, the condition number scales as $\mathcal{O}(h^{-2})$ for a fixed volume ratio. The volume ratio also has a significant impact on the condition number of the stiffness matrix, scaling it by $\mathcal{O}(\text{VR}^{-1})$. The poor conditioning of the linear system can affect the entire solution procedure for a non-linear solver and potentially lead to unreliable convergence.

Though the results presented above are for a DG discretization, it is important to note that the impact of small volume ratios is not due to the discretization choice. The next section demonstrates that the same consequences of small volume ratios for the one-dimensional reaction-diffusion problem can arise using a continuous Galerkin discretization. Therefore, choosing a DG discretization is not the cause of the issues presented above.

5.1 Boundary Derivative Outputs with Small Volume Ratios

With careful attention to output evaluation, it is possible to relieve the impact of small volume ratios on the output even with derivative based outputs. Switching to a continuous Galerkin discretization with strong boundary conditions, the standard weak formulation of the one-dimensional reaction-diffusion problem, Equation (5.1), is: find $u \in U_q^{CG}$ such that

$$a^{CG}(u, v) = \langle f, v \rangle, \quad \forall v \in V_0^{CG}, \quad (5.4)$$

where

$$a^{CG}(w, v) = \int_{\Omega} \nu \nabla w \cdot \nabla v + wv.$$

The function spaces are defined as

$$V_0^{CG} = H_0^1(\Omega), \text{ and } U_q^{CG} = q + H_0^1(\Omega) = \{v \in H^1(\Omega) \mid \gamma_{0,\partial\Omega}(v) = g\},$$

where $\gamma_{0,\partial\Omega} : H^1(\Omega) \rightarrow H^{\frac{1}{2}}(\partial\Omega)$ is the classical trace operator and $q \in H^1(\Omega)$ is selected such that $\gamma_{0,\partial\Omega}(q) = g$.

The finite-dimensional solution is represented using a set of Lagrange basis functions, $\{\phi_i\}$ and basis coefficients, u_i , such that $u_{h,p} = \sum_{i=1}^N \phi_i u_i \in U_{h,p,q}^{CG}$ where

$$\begin{aligned} V_{h,p,0}^{CG} &= \{v \in V_0^{CG} \mid v|_{\kappa} \in P^p(\kappa), \forall \kappa \in \mathcal{T}_h\}, \\ U_{h,p,q}^{CG} &= q + V_{h,p,0}^{CG}. \end{aligned}$$

The discrete problem is: find $u_{h,p} \in U_{h,p,q}^{CG}$ such that

$$a_h^{CG}(u_{h,p}, v_{h,p}) = \langle f, v_{h,p} \rangle, \quad \forall v_{h,p} \in V_{h,p,0}^{CG}. \quad (5.5)$$

To achieve superconvergence, $\mathcal{E} \sim h^{2p}$, of the error for the output $J(u) = \nu \frac{du}{dx} \Big|_{x=0}$ using the continuous Galerkin method, one cannot directly evaluate the derivative:

$$J_{h,p}(u_{h,p}) = \nu \frac{du_{h,p}}{dx} \Big|_{x=0} = \nu \sum_{i=1}^N \frac{d\phi_i}{dx} \Big|_{x=0} u_i. \quad (5.6)$$

Two arguments exist for why Equation (5.6) does not provide superconvergence of the output error. First, the output functional is unbounded so there is a loss of regularity in the dual solution. Additionally, the continuous dual problem for $J(u) = \nu \frac{du}{dx} \Big|_{x=0}$ is

$$\begin{aligned} \psi - \nu \frac{\partial^2 \psi}{\partial x^2} &= 0 \quad \text{in } \Omega = (0, 1), \\ \psi(0) &= 1/\nu, \quad \psi(1) = 0. \end{aligned} \quad (5.7)$$

Following the definition of Lu [82], a finite element formulation, Equation (5.5), together

with the discrete functional, Equation (5.6), is dual-consistent if ψ , the solution to the dual PDE, Equation (5.7), satisfies the discrete adjoint residual

$$a_{h,p}^{CG}(v_{h,p}, \psi) = J_{h,p}(v_{h,p}), \quad \forall v_{h,p} \in U_{h,p,q}^{CG}.$$

When Equation (5.6) is used to evaluate the output, the discretization is dual-inconsistent. The continuous dual solution does not satisfy the dual of the discrete problem, since $\psi_{h,p} \in V_{h,p,0}^{CG}$ and $\psi(0)$ is non-zero, $\psi \notin V_0^{CG}$. Due to the lack of dual consistency, the expected order of convergence for $J_{h,p}(u_{h,p}) - J(u)$ is only p . Following the work of Giles and Süli [53], the output $J(u) = \nu \frac{du}{dx}|_{x=0}$ can be computed dual consistently as $\int_{\partial\Omega} \boldsymbol{\sigma}(u) \cdot \hat{\mathbf{n}} \varrho ds$, where $\boldsymbol{\sigma}(u) = \nu \nabla u$ and $\varrho \in H^1(\Omega)$ is a weighting function with $\varrho(0) = 1$ and $\varrho(1) = 0$. This is computed as

$$\begin{aligned} J(u) &= \int_{\partial\Omega} \boldsymbol{\sigma} \cdot \hat{\mathbf{n}} \varrho ds = \int_{\Omega} \nabla \cdot [\boldsymbol{\sigma} \varrho] \\ &= - \int_{\Omega} \varrho f + \int_{\Omega} \nabla \cdot [\boldsymbol{\sigma} \varrho] + \int_{\Omega} \varrho \underbrace{f}_{u - \nabla \cdot \boldsymbol{\sigma}} \\ &= - \int_{\Omega} \varrho f + \int_{\Omega} \nabla \cdot [\boldsymbol{\sigma} \varrho] - \int_{\Omega} \varrho \nabla \cdot \boldsymbol{\sigma} + \int_{\Omega} \varrho u \\ &= - \int_{\Omega} \varrho f + \int_{\Omega} \boldsymbol{\sigma} \cdot \nabla \varrho + \int_{\Omega} \varrho u \\ &= -(f, \varrho) + a^{CG}(u, \varrho), \end{aligned} \tag{5.8}$$

where (\cdot, \cdot) is the L^2 inner product such that $(\cdot, \cdot) : L^2 \times L^2 \rightarrow \mathbb{R}$.

Naively, in the presence of small volume ratios, the weighting function in Equation (5.8) can be set to the Lagrange basis corresponding to the left boundary, ϕ_1 . Figures 5-9 (a)-(c) show the convergence in output error when $J_{h,p}(u_{h,p})$ is computed as $a_{h,p}^{CG}(u_{h,p}, \phi_1) - (f, \phi_1)$. Figure 5-9 (a) demonstrate the ability for superconvergence when $J_{h,p}(u_{h,p}) = a_{h,p}^{CG}(u_{h,p}, \varrho) - (f, \varrho)$, but Figures 5-9 (b)-(c) show the impact of small volume ratios is seen in the derivative based output.

In order to improve the convergence of the adjoint, the weighting function should not be a function of VR. Figures 5-9 (d)-(f) show the result of evaluating $J_{h,p}(u_{h,p})$ using $\varrho = 1 - x$. The careful selection of ϱ to be independent of VR allows the entire impact of a small volume ratio on the derivate boundary output to be removed. The selection of the

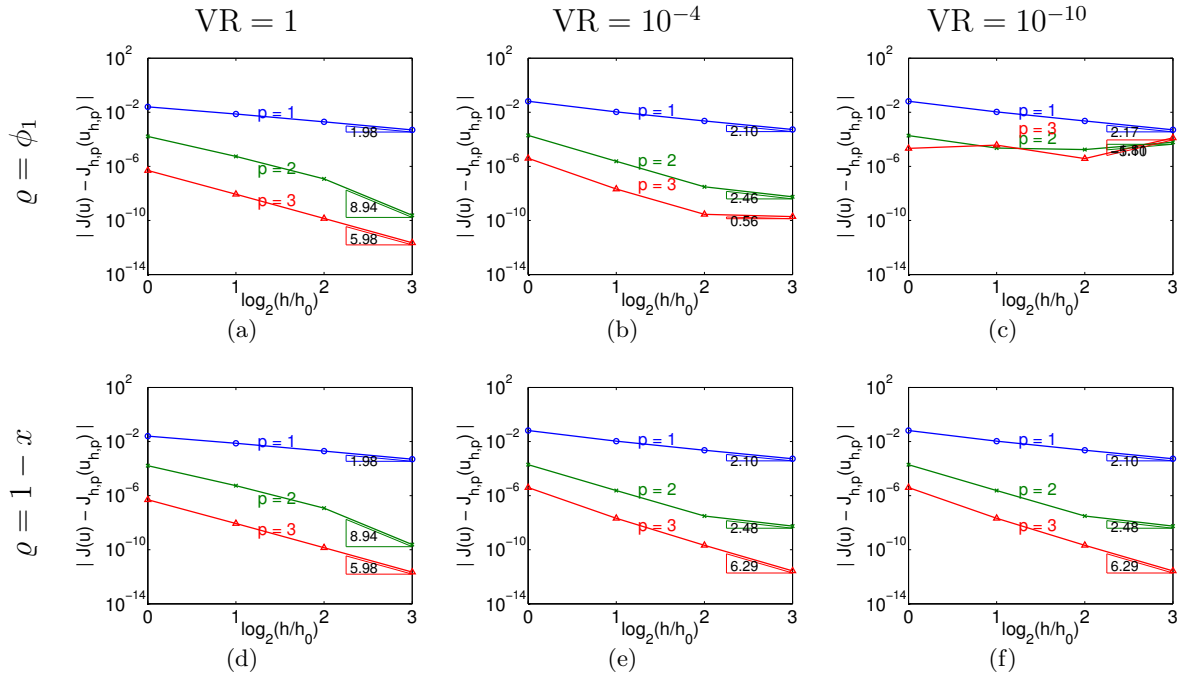


Figure 5-9: The convergence of the error in the output $J(u) = \nu \frac{du}{dx} \Big|_{x=0}$ for a range of volume ratios for the one-dimensional model problem, Equation (5.1). The selection of ρ for evaluating $J_{h,p}(u_{h,p}) = a_{h,p}^{CG}(u_{h,p}, \rho) - (f, \rho)$ is critical for limiting the influence of small volume ratios. When $\rho = \phi_1$, the impact of small volume ratios is large. If ρ is not a function of VR, like $1 - x$, there is no impact of small volume ratios. These results are from a continuous Galerkin discretization with strong boundary conditions.

weight function allows for evaluation of boundary derivative outputs that are important for aerodynamic flows, like heat flux and skin friction. In higher-spatial dimensions and complex geometries, ϱ could be specified using the distance function which must be computed for the SA equation. Though the removal of the impact of small volume ratios is demonstrated using a continuous Galerkin discretization, the implementation of Equation (5.8) with a weight function independent of volume ratio removes the impact of small volume ratios with a discontinuous Galerkin discretization as well. However, it is still necessary to understand the connection between small volume ratios and the conditioning of the stiffness matrix.

5.2 Analysis of the Conditioning of a One Dimensional Problem with Small Volume Ratios

The analysis presented in this section is an extension of the work of Kirby [71], who explored the idea of connecting functional analysis techniques, common in finite element analysis, with numerical linear algebra. The goal is to understand the impact of small volume ratios on the conditioning of the finite element stiffness matrix of elliptic problems using the bridge between functional analysis and numerical linear algebra.

5.2.1 Definitions

Before beginning, some definitions are required. A real Hilbert space, V , equipped with an inner product $(\cdot, \cdot)_V$ and an associated norm $\|\cdot\|_V$, is the starting point. The topological dual to V is V' with $\langle \cdot, \cdot \rangle$ denoting the duality pairing. Given any $f \in V'$, the dual norm is $\|f\|_{V'} \equiv \sup_{v \in V} \frac{|\langle f, v \rangle|}{\|v\|_V}$.

Given two Hilbert spaces, V_1 and V_2 , the Banach or complete space of bounded linear maps exists between V_1 and V_2 , denoted as $\mathcal{L}(V_1, V_2)$. For the linear map $T : V_1 \rightarrow V_2$, the norm associated with the Banach space is $\|T\|_{\mathcal{L}(V_1, V_2)} \equiv \sup_{v \in V_1} \frac{\|Tv\|_{V_2}}{\|v\|_{V_1}}$.

The one-dimensional reaction-diffusion problem, Equation (5.4), can be restated as the following: find $u \in V$ such that

$$a(u, v) = \langle f, v \rangle, \quad \forall v \in V, \tag{5.9}$$

where $f \in V'$. To simplify the notation, V is used to denote the Hilbert space, $H^1(\Omega)$.

By definition, a bilinear form $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ is *continuous* if there exists $0 \leq C < \infty$ with

$$C = \sup_{w, v \in V} \frac{|a(w, v)|}{\|w\|_V \|v\|_V} \quad \text{or} \quad |a(w, v)| \leq C \|w\|_V \|v\|_V, \quad \forall w, v \in V. \quad (5.10)$$

$a(\cdot, \cdot)$ is said to be *coercive* if there exists $0 < \alpha < \infty$ with

$$\alpha = \inf_{w \in V} \frac{a(w, w)}{\|w\|_V^2} \quad \text{or} \quad a(w, w) \geq \alpha \|w\|_V^2, \quad \forall w \in V. \quad (5.11)$$

5.2.2 Bilinear form to linear operator

Given a continuous linear operator $\mathcal{A} : V \rightarrow V'$, two constants associated with the operator are defined as

$$C_{\mathcal{A}} \equiv \|\mathcal{A}\|_{\mathcal{L}(V, V')} = \sup_{w \in V} \frac{\|\mathcal{A}w\|_{V'}}{\|w\|_V} \quad \text{and} \quad \alpha_{\mathcal{A}} \equiv \inf_{w \in V} \frac{\langle \mathcal{A}w, w \rangle}{\|w\|_V^2}.$$

An equivalence exists between continuous bilinear forms on $V \times V$ and continuous linear operators from $V \rightarrow V'$. If $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ is a continuous bilinear form, there exists $\mathcal{A} : V \rightarrow V'$ such that

$$\langle \mathcal{A}w, v \rangle = a(w, v) \quad \forall w, v \in V.$$

The equivalence implies that \mathcal{A} must be continuous if $a(\cdot, \cdot)$ is continuous and that $C_{\mathcal{A}}$ equals C from

$$\begin{aligned} C_{\mathcal{A}} &\equiv \sup_{w \in V} \frac{\|\mathcal{A}w\|_{V'}}{\|w\|_V} \\ &= \sup_{w \in V} \left[\frac{1}{\|w\|_V} \sup_{v \in V} \frac{|\langle \mathcal{A}w, v \rangle|}{\|v\|_V} \right] = \sup_{w \in V} \sup_{v \in V} \frac{|a(w, v)|}{\|w\|_V \|v\|_V} \\ &= C. \end{aligned} \quad (5.12)$$

As well, the operator constant $\alpha_{\mathcal{A}}$ is equal to the coercivity constant of the bilinear form,

$a(\cdot, \cdot)$, from

$$\alpha_{\mathcal{A}} \equiv \inf_{w \in V} \frac{\langle \mathcal{A}w, w \rangle}{\|w\|_V^2} = \inf_{w \in V} \frac{a(w, w)}{\|w\|_V^2} = \alpha. \quad (5.13)$$

5.2.3 Restriction to finite element space

The linear operator for finite element approximations is obtained by restricting the operator $\mathcal{A} : V \rightarrow V'$ to some finite-dimensional subspace V_h . The finite element operator, $\mathcal{A}_h : V_h \rightarrow V_h'$, satisfies the property

$$\langle \mathcal{A}w_h, v_h \rangle = \langle \mathcal{A}_h w_h, v_h \rangle, \quad \forall w_h, v_h \in V_h.$$

The restriction is denoted as $\mathcal{A}_h = \mathcal{A}|_{V_h}$ for convenience from here on. For this analysis, piecewise polynomial functions are considered for V_h corresponding to a conforming Galerkin discretization. The basis for V_h is given by $\{\phi_i\}_{i=1}^n$, where $n = \dim(V_h)$. $w_h \in V_h$ can be expressed as $w_h = \sum_{i=1}^n w_i \phi_i$, and the 2-norm is given as $\|w\|_2 = \sqrt{\sum_{i=1}^n w_i^2}$. In an attempt to maintain consistent notation throughout this work, Roman type distinguishes the vector of coefficients $w \in \mathbb{R}^n$ from the italicised $w_h \in V_h$. Matrices and vectors are also in Roman type, while members of Hilbert spaces and operators on Hilbert spaces are italicised.

The discrete problem is to find $u_h \in V_h$ such that

$$\langle \mathcal{A}u_h, v_h \rangle = a(u_h, v_h) = \langle f, v_h \rangle, \quad \forall v_h \in V_h. \quad (5.14)$$

$u \in \mathbb{R}^n$ satisfies the linear algebra equation

$$A_{ij}u_j = f_i, \quad 1 \leq i \leq n, \quad (5.15)$$

where the stiffness matrix, A_{ij} , equals $a(\phi_j, \phi_i)$ and forcing term, f_i , equals $\langle f, \phi_i \rangle$.

The discrete operator corresponding to a conforming finite element method has now been defined as a restriction of the continuous operator, \mathcal{A} . The restriction of \mathcal{A} to $V_h \subset V$ leads to the discrete operator \mathcal{A}_h . \mathcal{A}_h inherits the properties of the underlying PDE in infinite-dimensional Hilbert space. In other words, when the continuous coercive operator

$\mathcal{A} : V \rightarrow V'$ is restricted to a subspace $V_h \subset V$, a continuous coercive operator from $V_h \rightarrow V'_h$ is obtained such that

$$\begin{aligned} C_h &\equiv \sup_{w_h, v_h \in V_h} \frac{\langle \mathcal{A}_h w_h, v_h \rangle}{\|w_h\|_{V_h} \|v_h\|_{V_h}} \\ &= \sup_{w_h, v_h \in V_h} \frac{\langle \mathcal{A} w_h, v_h \rangle}{\|w_h\|_{V_h} \|v_h\|_{V_h}} \leq \sup_{w, v \in V} \frac{\langle \mathcal{A} w, v \rangle}{\|w\|_V \|v\|_V} = C \end{aligned} \quad (5.16)$$

$$\begin{aligned} \alpha_h &\equiv \inf_{w_h \in V_h} \frac{\langle \mathcal{A}_h w_h, w_h \rangle}{\|w_h\|_{V_h}^2} \\ &= \inf_{w_h \in V_h} \frac{\langle \mathcal{A} w_h, w_h \rangle}{\|w_h\|_{V_h}^2} \geq \inf_{w \in V} \frac{\langle \mathcal{A} w, w \rangle}{\|w\|_V^2} = \alpha. \end{aligned} \quad (5.17)$$

From Equations (5.16) and (5.17), continuity and coercivity constants for the finite-dimensional operators are bound by properties of the original continuous infinite-dimensional operator, \mathcal{A} . It is noted that the continuity and coercivity constants for the discrete operator, \mathcal{A}_h , are bound independent of the particular subspace, V_h , selected to represent the solution approximation, u_h . Thus, the continuity and coercivity constants are bound independent of polynomial order and mesh spacing.

For clarity, this analysis is limited to conforming methods so that bounds on the continuity and coercivity constants remain mesh independent. Not all finite element methods result from the restriction of the weak operator to the discrete space. While the definitions of C_h and α_h remain unchanged for non-conforming methods where $V_h \not\subset V$ or problems with mesh dependent operators, $\mathcal{A}_h \neq \mathcal{A}|_{V_h}$, the finite-dimensional continuity and coercivity constants are no longer simply bound by their infinite-dimensional counterparts, C and α .

5.2.4 Condition number for operators between Hilbert spaces

Given that an operator between Hilbert spaces, $\mathcal{A} \in \mathcal{L}(V, V')$, has a bounded inverse, the condition number of \mathcal{A} is defined as

$$\kappa(\mathcal{A}) \equiv \|\mathcal{A}\|_{\mathcal{L}(V, V')} \|\mathcal{A}^{-1}\|_{\mathcal{L}(V', V)}. \quad (5.18)$$

$\|\mathcal{A}\|_{\mathcal{L}(V,V')}$ is equal to C from Equation (5.12). $\|\mathcal{A}^{-1}\|_{\mathcal{L}(V',V)}$ can be expressed as

$$\begin{aligned}\|\mathcal{A}^{-1}\|_{\mathcal{L}(V',V)} &\equiv \sup_{f \in V'} \frac{\|\mathcal{A}^{-1}f\|_V}{\|f\|_{V'}} \\ &= \sup_{f \in V'} \sup_{v \in V} \frac{(\mathcal{A}^{-1}f, v)_V}{\|f\|_{V'} \|v\|_V} \\ &= \sup_{f \in V'} \sup_{v \in V} \frac{(u(f), v)_V}{\|f\|_{V'} \|v\|_V},\end{aligned}\tag{5.19}$$

where $\mathcal{A}^{-1}f$ is equivalent to u parameterized by f , such that $\mathcal{A}^{-1}f = u(f)$. By the Schwarz inequality this is bounded as,

$$\|\mathcal{A}^{-1}\|_{\mathcal{L}(V',V)} \leq \sup_{f \in V'} \sup_{v \in V} \frac{\|u(f)\|_V \|v\|_V}{\|f\|_{V'} \|v\|_V} = \sup_{f \in V'} \frac{\|u(f)\|_V}{\|f\|_{V'}}.\tag{5.20}$$

From the Lax-Milgram continuous dependence result $\|u(f)\|_V \leq \frac{1}{\alpha} \|f\|_{V'} \forall f \in V'$ (see for example [23] Section 2.7.7), so the norm of the inverse operator becomes

$$\|\mathcal{A}^{-1}\|_{\mathcal{L}(V',V)} \leq \sup_{f \in V'} \frac{1}{\alpha} \frac{\|f\|_{V'}}{\|f\|_{V'}} = \frac{1}{\alpha}.\tag{5.21}$$

The condition number of the continuous operator is given as

$$\kappa(\mathcal{A}) \equiv \|\mathcal{A}\|_{\mathcal{L}(V,V')} \|\mathcal{A}^{-1}\|_{\mathcal{L}(V',V)} \leq \frac{C}{\alpha}.$$

Restricting the Hilbert space to $V_h \subset V$, and once again letting \mathcal{A}_h be \mathcal{A} restricted to V_h , the condition number of $\mathcal{A}_h \in \mathcal{L}(V_h, V_h')$ is bounded by

$$\kappa(\mathcal{A}_h) \equiv \|\mathcal{A}_h\|_{\mathcal{L}(V_h, V_h')} \|\mathcal{A}_h^{-1}\|_{\mathcal{L}(V_h', V_h)} \leq \frac{C_h}{\alpha_h} \leq \frac{C}{\alpha}\tag{5.22}$$

from Equation (5.16) and (5.17). Note that the result of Equation (5.22) only strictly applies in the case of conforming finite element methods where $V_h \subset V$ and methods without mesh dependent operators where $\mathcal{A}_h = \mathcal{A}|_{V_h}$.

5.2.5 Linear algebraic representation

A connection must be made when moving from the theory of finite element spaces to the practicality of linear algebra. The connection will tie members of finite element spaces to corresponding linear algebraic objects. The first link is the mapping $\mathcal{I}_h : \mathbb{R}^n \rightarrow V_h$ defined by

$$\mathcal{I}_h \mathbf{w} = \sum_{i=1}^n w_i \phi_i = w_h. \quad (5.23)$$

\mathcal{I}_h acts as an interpretation operator, taking a vector of basis coefficients and returning a function used to represent a finite element solution. \mathcal{I}_h is invertible and \mathcal{I}_h^{-1} reformulates a function in V_h as a vector of coefficients.

Additionally, vectors can be interpreted as linear functionals by a second mapping $\mathcal{I}'_h : \mathbb{R}^n \rightarrow V'_h$, where

$$\langle \mathcal{I}'_h \mathbf{f}, v_h \rangle \equiv \sum_{i=1}^n f_i (\mathcal{I}_h^{-1} v_h)_i = \mathbf{f}^T (\mathcal{I}_h^{-1} v_h). \quad (5.24)$$

Through \mathcal{I}'_h , the vector \mathbf{f} is treated as a linear functional on V_h by computing its dot product with the vector of coefficients of the input function, v_h .

The mapping, \mathcal{I}'_h , can be associated with an adjoint mapping as

$$\langle g_h, v_h \rangle = \langle \mathcal{I}'_h g, v_h \rangle \equiv \langle g, (\mathcal{I}'_h)^* v_h \rangle, \quad (5.25)$$

where $(\mathcal{I}'_h)^* : V_h \rightarrow \mathbb{R}^n$ is the adjoint operator of \mathcal{I}'_h . By inspection, $(\mathcal{I}'_h)^*$ is equivalent to $\mathcal{I}_h^{-1} : V_h \rightarrow \mathbb{R}^n$. So, the adjoint of \mathcal{I}'_h is \mathcal{I}_h^{-1} (or $(\mathcal{I}'_h)^* = \mathcal{I}_h^{-1}$). Similarly,

$$((\mathcal{I}'_h)^{-1} g_h)^T \mathbf{v} = \langle g_h, ((\mathcal{I}'_h)^{-1})^* \mathbf{v} \rangle = \langle g_h, \mathcal{I}_h \mathbf{v} \rangle, \quad (5.26)$$

so the adjoint of $(\mathcal{I}'_h)^{-1}$ is \mathcal{I}_h (or $((\mathcal{I}'_h)^{-1})^* = \mathcal{I}_h$).

5.2.6 Relate stiffness matrix to Hilbert space setting

With the use of the interpretation operators, \mathcal{I}_h and \mathcal{I}'_h , a strong connection exists between a variational problem in Hilbert space and the stiffness matrix resulting from discretization

of the bilinear statement. Given any $w_h \in V_h$ with $w_h = \sum_{i=1}^n w_i \phi_i$ where $w = (\mathcal{I}_h)^{-1} w_h$, then $\mathcal{A}_h w_h \in V'_h$ satisfies

$$\mathcal{A}_h w_h = \mathcal{I}'_h(Aw).$$

This is demonstrated by letting $v_h = \sum_{i=1}^n v_i \phi_i \in V_h$, then

$$\begin{aligned} \langle \mathcal{A}_h w_h(w), v_h(v) \rangle &= a(w_h(w), v_h(v)) = a\left(\sum_{i=1}^n w_i \phi_i, \sum_{j=1}^n v_j \phi_j\right) \\ &= \sum_{i,j=1}^n w_i v_j a(\phi_i, \phi_j) = \sum_{i,j=1}^n w_i v_j A_{ji} \\ &= (Aw)^T v \quad \forall w, v \in \mathbb{R}^n. \end{aligned} \tag{5.27}$$

Replacing the functional f with Aw in Equation (5.24) leads to

$$\langle \mathcal{I}'_h(Aw), v_h \rangle = (Aw)^T (\mathcal{I}_h^{-1} v_h) = (Aw)^T v. \tag{5.28}$$

Combining Equations (5.27) and (5.28) gives

$$\mathcal{A}_h w_h(w) = \mathcal{I}'_h(Aw), \quad \forall w \in \mathbb{R}^n. \tag{5.29}$$

The relationship between the stiffness matrix, A , and the continuous operator, \mathcal{A}_h on V_h , can be written as

$$\begin{aligned} \mathcal{A}_h(\mathcal{I}_h w) &= \mathcal{I}'_h(Aw) \\ (\mathcal{I}'_h)^{-1} \mathcal{A}_h \mathcal{I}_h w &= Aw, \quad \forall w \in \mathbb{R}^n. \end{aligned} \tag{5.30}$$

It follows that

$$(\mathcal{I}'_h)^{-1} \mathcal{A}_h \mathcal{I}_h = A. \tag{5.31}$$

Equation (5.31) is identical to $\mathcal{I}'_h A = \mathcal{A}_h \mathcal{I}_h$, which conveys the equivalence of the action of the interpretation operator, \mathcal{I}_h , and the Hilbert space operator, \mathcal{A}_h , to the action of the

linear system matrix, A , and the functional interpretation operator, \mathcal{I}'_h . The equivalence of these actions is graphically shown in Figure 5-10.

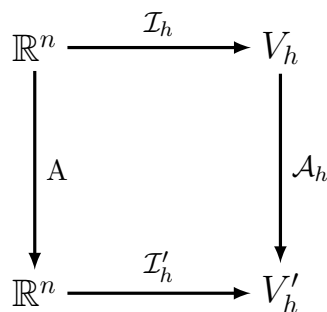


Figure 5-10: Diagram relating the equivalence of the actions of the the interpretation operator, \mathcal{I}_h , and the Hilbert space operator, \mathcal{A}_h , to the action of the matrix, A , and the functional interpretation operator, \mathcal{I}'_h , on Euclidean space, \mathbb{R}^n . (Taken from [71])

5.2.7 Matrix condition number – quasi-uniform mesh

The condition number of the stiffness matrix is dependent on the properties of the mesh used to define the finite-dimensional solution. Before analyzing the condition number of the linear system corresponding to the one-dimensional domain with a small volume ratio shown in Figure 5-2, the condition number for quasi-uniform meshes is first presented. Following the definition of Schwab [114], a family of meshes, $\{\mathcal{T}^j\}$, is quasi-uniform, if there exists positive constants a_1, a_2 independent of j such that

$$0 < a_1 \leq \frac{\max_{1 \leq i \leq N_e(\mathcal{T}^j)} h_i^{\mathcal{T}^j}}{\min_{1 \leq i \leq N_e(\mathcal{T}^j)} h_i^{\mathcal{T}^j}} \leq a_2 < \infty.$$

From a linear algebra perspective, the condition number of A , relative to the norm $\|\cdot\|$, is given by $\kappa(A) \equiv \|A\| \|A^{-1}\|$. If the norm is replaced with the standard 2-norm, ($\|\cdot\| = \|\cdot\|_2$), then $\kappa(A) = \frac{\sigma_1}{\sigma_n}$, where σ_1 and σ_n are the maximum and minimum singular values, respectively.

Relying on the interpretation operators and the continuous operator the definition of the stiffness matrix given by Equation (5.31) gives an alternative view of the matrix condition number. As condition numbers satisfy the multiplicative inequality, the stiffness matrix

condition number is expressed as

$$\begin{aligned}\kappa(\mathbf{A}) &= \kappa((\mathcal{I}'_h)^{-1} \mathcal{A}_h \mathcal{I}_h) \\ &\leq \kappa((\mathcal{I}'_h)^{-1}) \kappa(\mathcal{A}_h) \kappa(\mathcal{I}_h)\end{aligned}\tag{5.32}$$

$$\leq \kappa((\mathcal{I}'_h)^{-1}) \left(\frac{C}{\alpha}\right) \kappa(\mathcal{I}_h).\tag{5.33}$$

In order to compute the matrix condition number the condition number of the interpretation operators, \mathcal{I}_h and $(\mathcal{I}'_h)^{-1}$, must be bound. The interpretation operators condition numbers are given by

$$\kappa(\mathcal{I}_h) \equiv \|\mathcal{I}_h\|_{\mathcal{L}(\mathbb{R}^n, V_h)} \|\mathcal{I}_h^{-1}\|_{\mathcal{L}(V_h, \mathbb{R}^n)}\tag{5.34}$$

$$\kappa((\mathcal{I}'_h)^{-1}) \equiv \|(\mathcal{I}'_h)^{-1}\|_{\mathcal{L}(V'_h, \mathbb{R}^n)} \|\mathcal{I}'_h\|_{\mathcal{L}(\mathbb{R}^n, V'_h)}.\tag{5.35}$$

By relying on two facts, the condition number of the inverse of the functional interpretation operator, $\kappa((\mathcal{I}'_h)^{-1})$, is equal to the condition number of the function interpretation operator, $\kappa(\mathcal{I}_h)$. First, the condition number of the inverse of an operator is equal to the condition number of the operator itself. This follows from the definition of the condition number, i.e.

$$\kappa(\mathcal{I}_h) = \|\mathcal{I}_h\|_{\mathcal{L}(\mathbb{R}^n, V_h)} \|\mathcal{I}_h^{-1}\|_{\mathcal{L}(V_h, \mathbb{R}^n)} = \kappa(\mathcal{I}_h^{-1}).\tag{5.36}$$

Second, the operator norm of \mathcal{I}'_h is equal to the operator norm of its adjoint, $(\mathcal{I}'_h)^*$ [10], such that

$$\|\mathcal{I}'_h\|_{\mathcal{L}(\mathbb{R}^n, V'_h)} = \|(\mathcal{I}'_h)^*\|_{\mathcal{L}(V_h, \mathbb{R}^n)}.\tag{5.37}$$

Thus, the conditioning of the functional interpretation operator, \mathcal{I}'_h , is equivalent to the conditioning of the adjoint operator of the functional interpretation operator, $(\mathcal{I}'_h)^*$, introduced in Equation (5.25), or

$$\kappa(\mathcal{I}'_h) = \kappa((\mathcal{I}'_h)^*).\tag{5.38}$$

Since $(\mathcal{I}_h^l)^* = \mathcal{I}_h^{-1}$, and $\kappa(\mathcal{I}_h^{-1}) = \kappa(\mathcal{I}_h)$, then

$$\kappa(\mathcal{I}_h^l) = \kappa(\mathcal{I}_h). \quad (5.39)$$

To compute the matrix condition number all that is left is to find the condition number of the interpretation operators, \mathcal{I}_h . Since $\mathcal{I}_h \mathbf{w}$ represents the polynomial functions in V_h that are stored as the vector of coefficients \mathbf{w} for bases with compact support, an equivalence between $\|\mathcal{I}_h \mathbf{w}\|_{L^2}$ and $\|\mathbf{w}\|_2$ exists [35]. The scaling relationship is expressed as

$$c_1 h^{\frac{1}{2}} \|\mathbf{w}\|_2 \leq \|\mathcal{I}_h \mathbf{w}\|_{L^2} \leq c_2 h^{\frac{1}{2}} \|\mathbf{w}\|_2, \quad \forall \mathbf{w} \in \mathbb{R}^n \quad (5.40)$$

where the constants c_1 and c_2 are independent of element size and \mathbf{w} .

Up to this point the definitions and inequalities in Section 5.2 have been for generic PDEs involving the Hilbert space V , but going forward the bound on $\kappa(\mathcal{I}_h)$ will be for the reaction-diffusion problem presented in Equation (5.9) where V is H^1 . The operator norm of \mathcal{I}_h is

$$\|\mathcal{I}_h\|_{\mathcal{L}(\mathbb{R}^n, V_h)} = \sup_{\mathbf{w} \in \mathbb{R}^n} \frac{\|\mathcal{I}_h \mathbf{w}\|_{V_h}}{\|\mathbf{w}\|_2} = \sup_{\mathbf{w} \in \mathbb{R}^n} \frac{\|\mathcal{I}_h \mathbf{w}\|_{H^1}}{\|\mathbf{w}\|_2}. \quad (5.41)$$

where $V_h \subset H^1$.

Under the quasi-uniform mesh assumption, the inverse inequality, (see for example [23] section 4.5) can be used to bound the operator norm by

$$\begin{aligned} \|\mathcal{I}_h\|_{\mathcal{L}(\mathbb{R}^n, V_h)} &\leq \max_{\mathbf{w} \in \mathbb{R}^n} \frac{Bh^{-1} \|\mathcal{I}_h \mathbf{w}\|_{L^2}}{\|\mathbf{w}\|_2} && \text{(by inverse inequality)} \\ &\leq \max_{\mathbf{w} \in \mathbb{R}^n} \frac{Bh^{-1} (c_2 h^{\frac{1}{2}} \|\mathbf{w}\|_2)}{\|\mathbf{w}\|_2} && \text{(by Equation (5.40))} \\ &= Bc_2 h^{-\frac{1}{2}}, && \end{aligned} \quad (5.42)$$

where B and c_2 are constants independent of element size.

The operator norm of \mathcal{I}_h^{-1} is given

$$\begin{aligned}
\|\mathcal{I}_h^{-1}\|_{\mathcal{L}(V_h, \mathbb{R}^n)} &= \sup_{w_h \in V_h} \frac{\|\mathcal{I}_h^{-1}w_h\|_2}{\|w_h\|_{V_h}} \\
&= \sup_{w_h \in V_h} \frac{\|\mathcal{I}_h^{-1}w_h\|_2}{\|w_h\|_{H^1}} \\
&\leq \max_{w_h \in V_h} \frac{\|\mathcal{I}_h^{-1}w_h\|_2}{\|w_h\|_{L^2}} && \text{(by } \|u_h\|_{H^1} \geq \|u_h\|_{L^2}\text{)} \\
&\leq \max_{w_h \in V_h} \frac{\|\mathcal{I}_h^{-1}w_h\|_2}{c_1 h^{\frac{1}{2}} \|\mathcal{I}_h^{-1}w_h\|_2} && \text{(by Equation (5.40))} \\
&= \frac{1}{c_1 h^{\frac{1}{2}}}. \tag{5.43}
\end{aligned}$$

Combining Equations (5.42) and (5.43), the interpretation operator condition number is

$$\begin{aligned}
\kappa(\mathcal{I}_h) &\equiv \|\mathcal{I}_h\|_{\mathcal{L}(\mathbb{R}^n, V_h)} \|\mathcal{I}_h^{-1}\|_{\mathcal{L}(V_h, \mathbb{R}^n)} \\
&\leq \left(B c_2 h^{-\frac{1}{2}} \right) \left(\frac{1}{c_1 h^{\frac{1}{2}}} \right) \\
&= B' h^{-1}, \tag{5.44}
\end{aligned}$$

where the constant B' is independent of element size.

Returning to the original problem of the matrix condition number from Equation (5.33), the matrix condition number can be bound using the interpretation conditioning as

$$\begin{aligned}
\kappa(A) &\leq B' h^{-1} \frac{C}{\alpha} B' h^{-1} \\
&\leq \frac{C}{\alpha} \mathcal{O}(h^{-2}), \tag{5.45}
\end{aligned}$$

where C and α are the continuity and coercivity constants of the continuous bilinear form, respectively, and B' coming from the bound of the interpretation operators' condition numbers is independent of element size. It is important to recall that the bound given in Equation (5.45) is dependent on the use of the multiplicative inequality in Equation (5.32) and the assumption of a quasi-uniform mesh made in Equation (5.42).

5.2.8 Matrix condition number – mesh with a small volume ratio

In the case of non-quasi-uniform meshes the use of the inverse inequality (in Equation (5.42)) to bound the condition number of the stiffness matrix is too loose. For example, when an element with an arbitrarily small volume ratio is present the bound on the condition number based on the multiplicative inequality would be $\kappa(\mathbf{A}) = \mathcal{O}(\frac{1}{\sqrt{\mathbb{R}^2 h^2}})$. Returning to the multiplicative inequality, Equation (5.32), the stiffness matrix condition number is expressed as

$$\begin{aligned}
 \kappa(\mathbf{A}) &= \kappa((\mathcal{I}'_h)^{-1} \mathcal{A}_h \mathcal{I}_h) \\
 &\leq \kappa((\mathcal{I}'_h)^{-1}) \kappa(\mathcal{A}_h) \kappa(\mathcal{I}_h) \\
 &\leq \kappa((\mathcal{I}'_h)^{-1}) \left(\frac{C}{\alpha}\right) \kappa(\mathcal{I}_h) \\
 &= \|(\mathcal{I}'_h)^{-1}\|_{\mathcal{L}(V'_h, \mathbb{R}^n)} \|\mathcal{I}'_h\|_{\mathcal{L}(\mathbb{R}^n, V'_h)} \left(\frac{C}{\alpha}\right) \|\mathcal{I}_h\|_{\mathcal{L}(\mathbb{R}^n, V_h)} \|\mathcal{I}_h^{-1}\|_{\mathcal{L}(V_h, \mathbb{R}^n)} \quad (5.46)
 \end{aligned}$$

The four terms of Equation (5.46) remaining to be analyzed depend on the ability to bound $\|\mathcal{I}_h w\|_{H^1}$ above and below by $\|w\|_2$ on the one-dimensional domain shown in Figure 5-2. The bound will be presented for $w_h \in V_h \subset H^1$, which is the space of continuous piecewise polynomials of order p such that

$$V_h = \{w_h \in H^1(\Omega) \mid w_h|_{e_i} \in P^p(e_i), \forall e_i \in \mathcal{T}_h\},$$

where the domain, Ω , has been split into a set, \mathcal{T}_h , of N_e elements, e_i . w_h is represented using a set basis function and basis coefficients, w_i . The basis functions considered here are Lagrange, $\{\phi_i\}_{i=1}^n$, but the bound will not be limited to a particular choice of basis functions. The function w_h can be expressed as $w_h = \sum_{i=1}^n \phi_i w_i = \mathcal{I}_h w$, where $w \in \mathbb{R}^n$, with $n = N_e p + 1$.

For each element in \mathcal{T}_h the H^1 semi-norm can be expressed as

$$\begin{aligned}
 |w_h|_{H^1(e_i)}^2 &= \int_{e_i} \left(\frac{d\phi_j}{dx} w_j\right)^2 dx = \int_{e_i} \frac{d\phi_j}{dx} \frac{d\phi_k}{dx} w_j w_k dx = \frac{1}{h_i} \int_{K_{\text{ref}}} \frac{d\phi_j}{d\xi} \frac{d\phi_k}{d\xi} w_j w_k d\xi \\
 &= \frac{1}{h_i} w_{e_i}^T \hat{K} w_{e_i},
 \end{aligned}$$

where w_{e_i} is the set of w_j s corresponding to element e_i and the summation on the repeated

indices is implied. w_{e_i} can be expressed with a restriction operator, $R : \mathbb{R}^n \rightarrow \mathbb{R}^{p+1}$, such that $w_{e_i} = R_i w$. The elemental H^1 semi-norm can be compactly written as

$$|w_h|_{H^1(e_i)}^2 = \frac{1}{h_i} w^T R_i^T \hat{K} R_i w.$$

\hat{K} is the reference element stiffness matrix corresponding to the Laplace operator so it is positive semi-definite, i.e. $w_{e_i}^T \hat{K} w_{e_i} \geq 0$, $\forall w_{e_i} \in \mathbb{R}^{p+1}$.

The L^2 norm of w_h is given by

$$\|w_h\|_{L^2(e_i)}^2 = \int_{e_i} (\phi_j w_j)^2 dx = h_i w^T R_i^T \hat{M} R_i w,$$

where \hat{M} is the reference element mass matrix which is positive definite, i.e. $w_{e_i}^T \hat{M} w_{e_i} > 0$, $\forall w_{e_i} \in \mathbb{R}^{p+1}$.

Combining the H^1 semi-norm and the L^2 norm of w_h gives the H^1 norm as

$$\begin{aligned} \|w_h\|_{H^1(\Omega)}^2 &= \sum_{i=1}^n \frac{1}{h_i} w^T R_i^T \hat{K} R_i w + h_i w^T R_i^T \hat{M} R_i w \\ &= w^T \left[\sum_{i=1}^n \frac{1}{h_i} R_i^T \hat{K} R_i + h_i R_i^T \hat{M} R_i \right] w \\ &= w^T \left[\sum_{i=1}^n R_i^T \left(\frac{1}{h_i} \hat{K} + h_i \hat{M} \right) R_i \right] w \\ &= w^T B w, \end{aligned}$$

where B is an assembled stiffness matrix. B is the result of discretizing the one-dimensional problem

$$\frac{d^2 u}{dx^2} + u = f, \quad \text{in } \Omega. \quad (5.47)$$

Equation (5.47) can be written in weak form as: find $u \in H^1$ such that

$$a^{\text{RD}}(u, v) = (f, v), \quad \forall v \in H^1.$$

The bound on $\|w_h\|_{H^1(\Omega)}$ in terms of $\|w\|_2$ can be expressed in terms of the eigenvalues

of \mathbf{B} or equivalently the singular values of \mathbf{B} since it is symmetric positive definite as

$$\lambda_{\min}(\mathbf{B}) \|w\|_2^2 \leq \|w_h\|_{H^1(\Omega)}^2 \leq \lambda_{\max}(\mathbf{B}) \|w\|_2^2.$$

In order to compute a bound on the minimum singular value of \mathbf{B} the work of Fried [48] is followed. The minimum singular value of \mathbf{B} is given by

$$\sigma_{\min}(\mathbf{B}) = \frac{1}{\sigma_{\max}(\mathbf{B}^{-1})} = \frac{1}{\|\mathbf{B}^{-1}\|_2} \geq \frac{1}{\|\mathbf{B}^{-1}\|_{\infty}}.$$

So, if it is possible to bound $\|\mathbf{B}^{-1}\|_{\infty}$, a bound is provided for $\sigma_{\min}(\mathbf{B})$. Since \mathbf{B} is a positive definite and symmetric matrix of rank n , the ∞ -norm of \mathbf{B}^{-1} can be bounded as

$$\|\mathbf{B}^{-1}\|_{\infty} < n \max_i (\mathbf{B}^{-1})_{ii}, \quad i = 1, 2, \dots, n. \quad (5.48)$$

The proof of Equation (5.48) follows from the facts that since \mathbf{B}^{-1} is positive definite and symmetric $(\mathbf{B}^{-1})_{ii} > 0$, and that for any i and j , $(\mathbf{B}^{-1})_{ii} + (\mathbf{B}^{-1})_{jj} > |(\mathbf{B}^{-1})_{ij}|$.

The bound on $\max_i (\mathbf{B}^{-1})_{ii}$ will rely on the continuous properties of the exact solution to Equation(5.47) which is bounded by its Green's function, $G(x, x)$. The bound on $\max_i (\mathbf{B}^{-1})_{ii}$ is given as

$$\max_i (\mathbf{B}^{-1})_{ii} \leq \Gamma \quad i = 1, 2, \dots, n, \quad (5.49)$$

where $\Gamma = \max[G(x, x)]$. The proof of Equation (5.49) comes via the coercivity of the bilinear form $a^{\text{RD}}(u - u_h, u - u_h) \geq 0$ and Galerkin orthogonality, $a^{\text{RD}}(u - u_h, u_h) = 0$, where remaining consistent with notation, the finite element solution is denoted as u_h and the exact solution is given as u . Using coercivity and orthogonality gives

$$\begin{aligned} a^{\text{RD}}(u - u_h, u - u_h) &\geq 0 \\ a^{\text{RD}}(u - u_h, u) &\geq 0 \\ (f, u - u_h) &\geq 0 \\ (f, u) &\geq (f, u_h). \end{aligned} \quad (5.50)$$

Now, if f is chosen to be a point forcing term, then Equation (5.50) gives that $u \geq u_h$. In other words, at the point of application of the force, u is never less than the finite element solution u_h . The response at the nodal points of the finite element solution due to a point force at node i is just the i^{th} column in B^{-1} corresponding to the point force. In particular, if the point force occurs at node i the finite element response at i is $(B^{-1})_{ii}$. Since the exact solution at a point x is given by $G(x, x)$, and $u \geq u_h$, then $G(x, x) \geq (B^{-1})_{ii}$ and Equation (5.49) is proved.

Combining Equations (5.48) and (5.49), the minimum singular value of B can be bound as

$$\sigma_{\min}(B) \geq \frac{1}{\|B^{-1}\|_{\infty}} \geq \frac{1}{n\Gamma}.$$

n is related to the number of elements in the domain as $n = N_e p + 1$ and for the grid shown in Figure 5-2 $N_e = \frac{1}{h} + 1 - \text{VR}$, so $n = p \left(\frac{1}{h} + 1 - \text{VR} \right) + 1$. In the limit of small volume ratios, n goes to $\frac{p}{h} + p + 1$. The bound on the stiffness matrix singular minimum value in the limit of small volume ratio becomes

$$\lambda_{\min}(B) = \sigma_{\min}(B) \geq \frac{1}{\Gamma \left(\frac{p}{h} + p - 1 \right)} \geq D_L h,$$

where D_L is a constant independent of mesh size.

Next, the Global Eigenvalue Theorem [47] is employed to bound the maximum eigenvalue of B . If w_{e_i} corresponds to the portion of w associated with the i^{th} element, then

$$w^T w \leq \sum_i^{N_e} w_{e_i}^T w_{e_i} \leq n e_{\max} w^T w, \quad (5.51)$$

where $n e_{\max}$ corresponds to the maximum number of elements meeting at a node. The proof of Equation (5.51) comes from considering w of length n and the number of elements that meet at a node as $n e_i$, then the same w_i^2 will appear $n e_i$ times as

$$\sum_i^{N_e} w_{e_i}^T w_{e_i} = \sum_{i=1}^n w_i^2 n e_i,$$

which gives the limit

$$\sum_i^{N_e} \mathbf{w}_{e_i}^T \mathbf{w}_{e_i} \leq \mathbf{w}^T \mathbf{w} \, n e_{\max}.$$

For the one-dimensional problem considered here, $n e_{\max} = 2$.

Now, letting the maximum eigenvalue of \mathbf{B} be denoted as $\lambda_{\max}^{\mathbf{B}}$ and the maximum eigenvalue of the element stiffness matrix be denoted as $\lambda_{\max}^{\mathbf{B}_{e_i}}$, then

$$\lambda_{\max}^{\mathbf{B}} \leq n e_{\max} \max_i \left(\lambda_{\max}^{\mathbf{B}_{e_i}} \right), \quad (5.52)$$

for i ranging from 1 to N_e . Equation (5.52) can be shown by considering that for each element

$$\mathbf{w}_{e_i}^T \mathbf{B}^{e_i} \mathbf{w}_{e_i} \leq \lambda_{\max}^{\mathbf{B}_{e_i}} \mathbf{w}_{e_i}^T \mathbf{w}_{e_i}, \quad i = 1, 2, \dots, N_e \quad (5.53)$$

Then, if \mathbf{w} is assumed to be the normalized eigenvector corresponding to $\lambda_{\max}^{\mathbf{B}}$ such that

$$\lambda_{\max}^{\mathbf{B}} = \mathbf{w}^T \mathbf{B} \mathbf{w} = \sum_{i=1}^{N_e} \mathbf{w}_{e_i}^T \mathbf{B}^{e_i} \mathbf{w}_{e_i}, \quad (5.54)$$

combining Equations (5.51), (5.53), and (5.54) gives

$$\begin{aligned} \lambda_{\max}^{\mathbf{B}} &= \sum_{i=1}^{N_e} \mathbf{w}_{e_i}^T \mathbf{B}^{e_i} \mathbf{w}_{e_i} \leq \sum_{i=1}^{N_e} \lambda_{\max}^{\mathbf{B}_{e_i}} \mathbf{w}_{e_i}^T \mathbf{w}_{e_i} \leq \max_{i=j, \dots, N_e} \lambda_{\max}^{\mathbf{B}_{e_j}} \sum_{i=1}^{N_e} \mathbf{w}_{e_i}^T \mathbf{w}_{e_i} \\ &\leq n e_{\max} \max_{j=1, \dots, N_e} \lambda_{\max}^{\mathbf{B}_{e_j}}. \end{aligned}$$

For a small volume ratio $\max_{i=1, \dots, N_e} \lambda_{\max}^{\mathbf{B}_{e_i}}$ will be dominated by the $\frac{1}{hVR}$ term from $\mathbf{B}_{e_1} = hVR\hat{M} + \frac{1}{hVR}\hat{K}$ such that

$$\lambda_{\max}^{\mathbf{B}} \leq \frac{D_H}{hVR},$$

where D_H is a constant independent of mesh size.

So, a bound of $\|w_h\|_{H^1(\Omega)}$ from above and below can now be expressed as

$$\begin{aligned}\lambda_{\min}(\mathbf{B})\|\mathbf{w}\|_2^2 &\leq \|w_h\|_{H^1(\Omega)}^2 \leq \lambda_{\max}(\mathbf{B})\|\mathbf{w}\|_2^2 \\ D_L h \|\mathbf{w}\|_2^2 &\leq \|w_h\|_{H^1(\Omega)}^2 \leq \frac{D_H}{h\mathbf{VR}}\|\mathbf{w}\|_2^2,\end{aligned}\tag{5.55}$$

where D_L and D_H are independent of element size but dependent on the polynomial order and basis functions selected to represent w_h .

Returning to the four components of Equation (5.46) remaining to be bound, by definition, $\|\mathcal{I}_h\|_{\mathcal{L}(\mathbb{R}^n, V_h)}$ is given by

$$\|\mathcal{I}_h\|_{\mathcal{L}(\mathbb{R}^n, V_h)} \equiv \sup_{\mathbf{w} \in \mathbb{R}^n} \frac{\|\mathcal{I}_h \mathbf{w}\|_{V_h}}{\|\mathbf{w}\|_2} = \sup_{\mathbf{w} \in \mathbb{R}^n} \frac{\|\mathcal{I}_h \mathbf{w}\|_{H^1}}{\|\mathbf{w}\|_2}.\tag{5.56}$$

Combining Equations (5.55) and (5.56) provides a bound on $\|\mathcal{I}_h\|_{\mathcal{L}(\mathbb{R}^n, V_h)}$:

$$\begin{aligned}\|\mathcal{I}_h\|_{\mathcal{L}(\mathbb{R}^n, V_h)} &\leq \max_{\mathbf{w} \in \mathbb{R}^n} \frac{\sqrt{\frac{D_H}{h\mathbf{VR}}}\|\mathbf{w}\|_2}{\|\mathbf{w}\|_2} \\ &= \sqrt{\frac{D_H}{h\mathbf{VR}}}.\end{aligned}\tag{5.57}$$

Furthermore, since $\|(\mathcal{I}'_h)^{-1}\|_{\mathcal{L}(V'_h, \mathbb{R}^n)} = \|((\mathcal{I}'_h)^{-1})^*\|_{\mathcal{L}(\mathbb{R}^n, V_h)}$ and $((\mathcal{I}'_h)^{-1})^* = \mathcal{I}_h$, then the norm of the inverse of the functional interpretation operator is bounded as

$$\|(\mathcal{I}'_h)^{-1}\|_{\mathcal{L}(V'_h, \mathbb{R}^n)} \leq \sqrt{\frac{D_H}{h\mathbf{VR}}}.\tag{5.58}$$

$\|\mathcal{I}_h^{-1}\|_{\mathcal{L}(V_h, \mathbb{R}^n)}$ is defined as

$$\|\mathcal{I}_h^{-1}\|_{\mathcal{L}(V_h, \mathbb{R}^n)} = \sup_{w_h \in V_h} \frac{\|\mathcal{I}_h^{-1} w_h\|_2}{\|w_h\|_{V_h}}\tag{5.59}$$

Substituting $w_h = \mathcal{I}_h v$ into Equation (5.59) gives

$$\begin{aligned}
\|\mathcal{I}_h^{-1}\|_{\mathcal{L}(V_h, \mathbb{R}^n)} &= \sup_{v \in \mathbb{R}^n} \frac{\|v\|_2}{\|\mathcal{I}_h v\|_{V_h}} \\
&= \left(\inf_{v \in \mathbb{R}^n} \frac{\|\mathcal{I}_h v\|_{V_h}}{\|v\|_2} \right)^{-1} \\
&= \left(\inf_{v \in \mathbb{R}^n} \frac{\|\mathcal{I}_h v\|_{H^1}}{\|v\|_2} \right)^{-1}
\end{aligned} \tag{5.60}$$

Once again, Equation (5.55) can be used to bound $\|v_h\|_{H^1}$ from below with $\sqrt{D_L h} \|v\|_2$. The bound on $\|v_h\|_{H^1}$ allows for $\|\mathcal{I}_h^{-1}\|_{\mathcal{L}(V_h, \mathbb{R}^n)}$ to be expressed as

$$\begin{aligned}
\|\mathcal{I}_h^{-1}\|_{\mathcal{L}(V_h, \mathbb{R}^n)} &\leq \left(\min_{w \in \mathbb{R}^n} \frac{\sqrt{D_L h} \|v\|_2}{\|v\|_2} \right)^{-1} \\
&= \sqrt{\frac{1}{D_L h}}.
\end{aligned} \tag{5.61}$$

Likewise, $\|\mathcal{I}'_h\|_{\mathcal{L}(\mathbb{R}^n, V'_h)}$ is defined as

$$\|\mathcal{I}'_h\|_{\mathcal{L}(\mathbb{R}^n, V'_h)} = \sup_{g \in \mathbb{R}^n} \frac{\|\mathcal{I}'_h g\|_{V'_h}}{\|g\|_2}. \tag{5.62}$$

Again, substituting $g = (\mathcal{I}'_h)^{-1} g$ into Equation (5.62) gives

$$\begin{aligned}
\|\mathcal{I}'_h\|_{\mathcal{L}(\mathbb{R}^n, V'_h)} &= \sup_{g \in V'_h} \frac{\|g\|_{V'_h}}{\|(\mathcal{I}'_h)^{-1} g\|_2} \\
&= \left(\inf_{g \in V'_h} \frac{\|(\mathcal{I}'_h)^{-1} g\|_2}{\|g\|_{V'_h}} \right)^{-1},
\end{aligned} \tag{5.63}$$

where $\|g\|_{V'_h} \equiv \sup_{v_h \in V_h} \frac{\langle g, v_h \rangle}{\|v_h\|_{V_h}} = \left(\inf_{v_h \in V_h} \frac{\|v_h\|_{V_h}}{\langle g, v_h \rangle} \right)^{-1}$. Plugging in the definition of $\|g\|_{V'_h}$ into Equation (5.63) gives

$$\|\mathcal{I}'_h\|_{\mathcal{L}(\mathbb{R}^n, V'_h)} = \left(\inf_{g \in V'_h} \left[\|(\mathcal{I}'_h)^{-1} g\|_2 \inf_{v_h \in V_h} \frac{\|v_h\|_{V_h}}{\langle g, v_h \rangle} \right] \right)^{-1}. \tag{5.64}$$

Using the interpretation operators allows v_h to be expressed as $\mathcal{I}_h \mathbf{v}$, which leads to

$$\|\mathcal{I}'_h\|_{\mathcal{L}(\mathbb{R}^n, V'_h)} = \left(\inf_{g \in V'_h} \left[\|\mathcal{I}'_h\|^{-1} g\|_2 \inf_{\mathbf{v} \in \mathbb{R}^n} \frac{\|\mathcal{I}_h \mathbf{v}\|_{V_h}}{\langle g, \mathcal{I}_h \mathbf{v} \rangle} \right] \right)^{-1}. \quad (5.65)$$

The adjoint of the functional interpretation operator, $\langle g, \mathcal{I}_h \mathbf{v} \rangle = (\mathcal{I}'_h g)^T \mathbf{v} = ((\mathcal{I}'_h)^{-1} g)^T \mathbf{v}$ can be used to simplify the expression for $\|\mathcal{I}'_h\|_{\mathcal{L}(\mathbb{R}^n, V'_h)}$ as

$$\begin{aligned} \|\mathcal{I}'_h\|_{\mathcal{L}(\mathbb{R}^n, V'_h)} &= \left(\inf_{g \in V'_h} \left[\|\mathcal{I}'_h\|^{-1} g\|_2 \inf_{\mathbf{v} \in \mathbb{R}^n} \frac{\|\mathcal{I}_h \mathbf{v}\|_{V_h}}{((\mathcal{I}'_h)^{-1} g)^T \mathbf{v}} \right] \right)^{-1} \\ &\leq \left(\inf_{g \in V'_h} \left[\|\mathcal{I}'_h\|^{-1} g\|_2 \inf_{\mathbf{v} \in \mathbb{R}^n} \frac{\|\mathcal{I}_h \mathbf{v}\|_{V_h}}{\|\mathcal{I}'_h\|^{-1} g\|_2 \|\mathbf{v}\|_2} \right] \right)^{-1} \\ &= \left(\inf_{\mathbf{v} \in \mathbb{R}^n} \frac{\|\mathcal{I}_h \mathbf{v}\|_{V_h}}{\|\mathbf{v}\|_2} \right)^{-1} = \|\mathcal{I}_h^{-1}\|_{\mathcal{L}(V_h, \mathbb{R}^n)} \\ &\leq \sqrt{\frac{1}{D_L h}}. \end{aligned} \quad (5.66)$$

Using the individual bounds on the four unknowns in Equation (5.46) (given in (5.57), (5.58), (5.61), (5.66)), the condition number of the stiffness matrix can be bound as

$$\kappa(\mathbf{A}) \leq \frac{C}{\alpha} \frac{1}{h^2 \mathbf{VR}} \frac{D_H}{D_L}. \quad (5.67)$$

This condition number bound only differs to the one presented for the quasi-uniform case, Equation (5.45), by the additional $\mathcal{O}(\mathbf{VR}^{-1})$ term. The bound presented in Equation (5.67) matches the numerical results motivating this work in Figure 5-8.

5.2.9 Implications of $\kappa(\mathbf{A}) = \frac{C}{\alpha} \mathcal{O}(h^{-2}) \mathcal{O}(\mathbf{VR}^{-1})$

Important conclusions can be drawn about the stiffness matrix condition number's dependence on small volume ratios. The first is that the volume ratio impact on the condition number is not due to any impact of the small volume ratio on the variational statement. Assuming a conforming finite element approximation, the conditioning of the continuous linear operator, \mathcal{A}_h , is bound independently of the interpolation space used to represent functions in the Hilbert spaces. It is the interpretation operators used to connect linear algebraic objects to members of the finite element spaces that are impacted by small volume ratios. To

remove the impact of small volume ratios on the condition number of A , the discretization space must be modified so that the elements with small volume ratios are explicitly removed. Removal of the elements with small volume ratios removes the impact of small volume ratios on the interpretation operators, \mathcal{I}_h and \mathcal{I}'_h .

It may be possible for isolated test problems to correctly modify the bilinear form such that it has the inverse dependence on small volume ratios, as seen by the interpretation operators. For example, in a one-dimensional reaction-diffusion problem using a continuous Galerkin discretization, it is possible to introduce a jump penalty weighted by VR that correctly cancels out the VR impact on the interpretation operator. However, in more complex problems (in particular in higher dimensions) if the jump penalty did not lead to a clean cancellation it could make the conditioning problem worse. In higher dimension the use of a jump penalty is significantly more complicated.

5.3 Modified Discretization Space

A few options are considered to modify the discretization space and remove the impact of small volume ratios on the interpretation operators. One possibility is to nudge all the problem nodes (grid nodes located in the computational domain, but close to the embedded surface) outside of the computational domain or onto the embedded surface. By nudging the problem nodes the typical grid typology, which results in small volume ratios, is eliminated. For example, Figure 5-11 illustrates how nudging node 1 outside of the computational domain removes element A and the small volume ratio associated with the element.

The method of grid-node nudging has been used successfully in two dimensions [43]. However, as the geometry becomes more complex and the cut-cell method is extended to three dimensions, deciding on the nudging direction becomes a difficult problem. In order to maintain an algorithm which is extendable to three dimensions, small volume ratios are eliminated by merging, following the work of [28, 49].

Essentially, because the elements with small volume ratios are so small in comparison to their neighbors, solution quality is not affected if the small elements are conglomerated into their neighbor sharing the largest common face. The common face between the large and small neighbor is eliminated and the integration rules for the small element and its

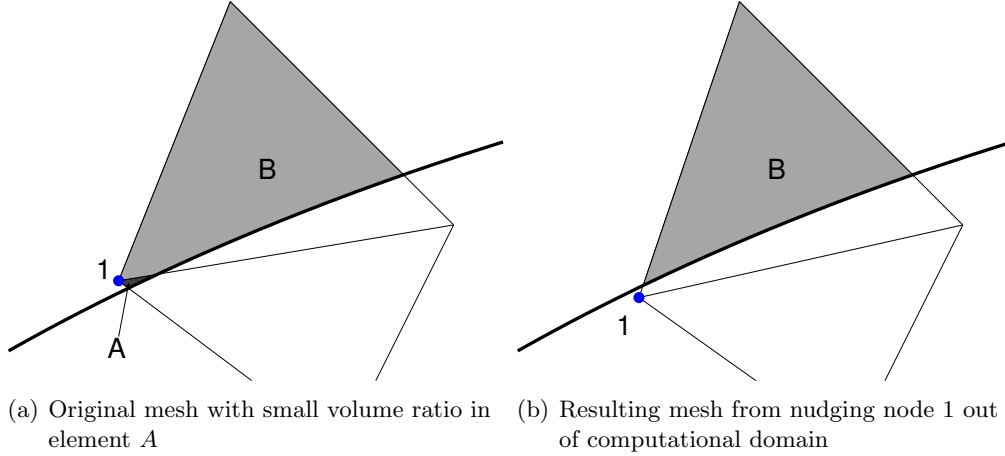


Figure 5-11: The effect of nudging node 1 to eliminate the small volume ratio associated with element A .

other non-common faces are added to the larger neighbor. The basis used to represent the polynomial solution remains the basis originating from the larger element. An illustration of merging is shown in Figure 5-12, where A is merged into element B resulting in a slightly large element, C . For the results presented in Chapter 7 all elements with volume ratios less than $VR_{\text{crit}} = 10^{-5}$ are merged. When the merging technique is implemented, element merging continues one internal face at a time until the global minimum volume ratio is above a set tolerance.

5.4 Model Problem Results

Returning to the one-dimensional reaction-diffusion problem of Equation (5.1), the elements with small volume ratios are removed using the merging technique illustrated in Figure 5-13. Once the mesh is merged the derivative boundary output, $\frac{du}{dx}|_{x=0}$, recovers its optimal convergence properties even when $J_h(u_h) = a_h^{CG}(u_h, \phi_1) - (f, \phi_1)$. Figure 5-14 shows that, regardless of the volume ratio (1 or 10^{-10}), merging allows for optimal convergence of the boundary output.

Merging is also tested on the two-dimensional convection-diffusion problem presented in Section 4.5 Equation (4.1). To set up cut-cell meshes with small volume ratios, the outer boundary location of the structured background mesh (Figure 4-9(a)), remains fixed, while

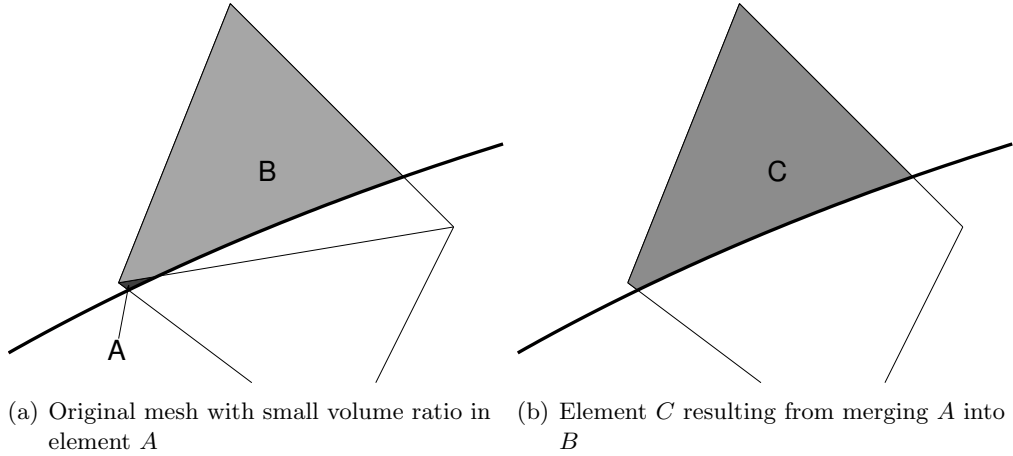


Figure 5-12: Illustration of the effect of merging element A into element B . The resulting element, C , maintains the solution basis of element B and the quadrature points are taken from both element A and B .

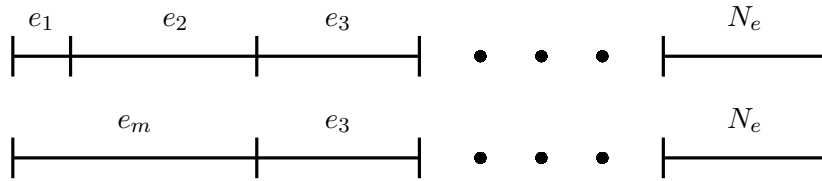


Figure 5-13: Original and merged domains for the one-dimensional model problem. e_1 and e_2 are merged to form e_m .

the remaining nodes shift radially inwards. The nodes on the inner boundary shift such that volume ratios $\mathcal{O}(10^{-6})$ are present after cutting. The volume ratio tolerance used in this test is 10^{-2} .

Figure 5-15 shows the convergence history of the heat flux distribution error for three different cut-cell grids. The first is referred to as a boundary-conforming cut grid as it is a globally linear background mesh with a higher-order embedded surface that exactly intersects the background grid nodes on the boundary. The resulting cut elements have two linear faces and one curved face. The other two grids are the merged and non-merged version of the $\text{VR} = \mathcal{O}(10^{-6})$ cut grids. For the derivative based output of interest, a convergence rate of p is expected based on the H_1 error convergence. The boundary-conforming cut grid almost achieves the expected rate of convergence, while the merged and non-merged results

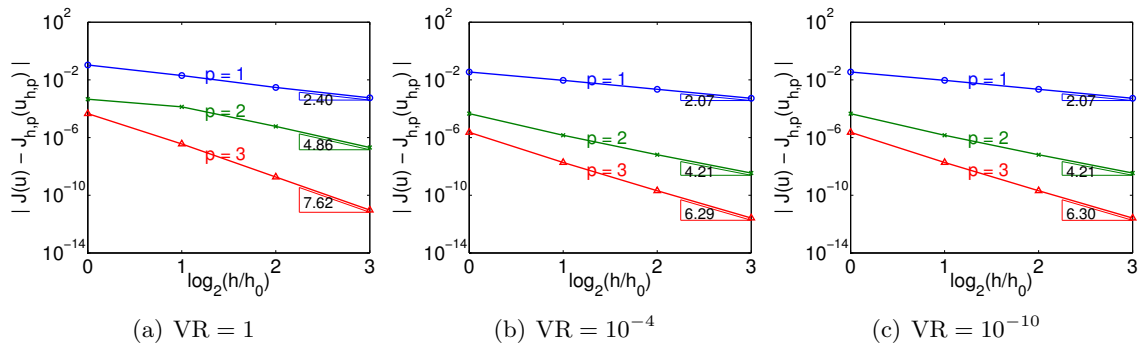


Figure 5-14: The convergence of the error in the output $\left. \frac{du}{dx} \right|_{x=0}$, with $J_h(u_h) = a_h^{CG}(u_h, \phi_1) - (f, \phi_1)$ for a range of volume ratios for the one-dimensional model problem, Equation (5.1). Merging removes the impact of the small volume ratio in the domain.

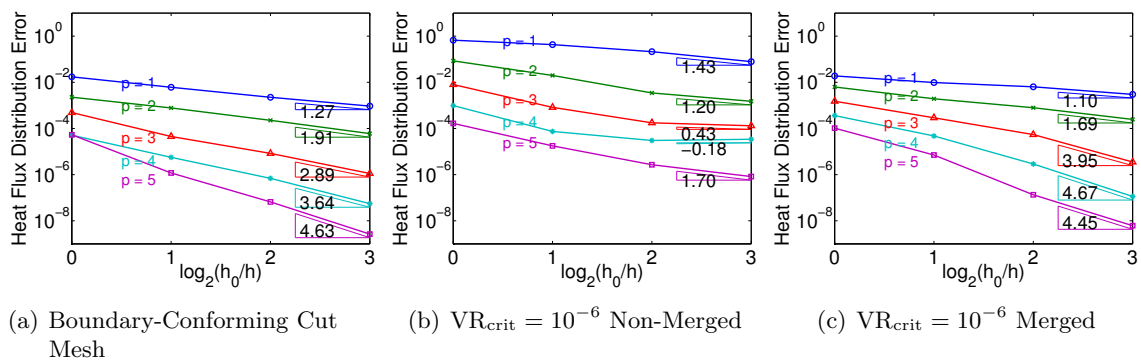


Figure 5-15: Convergence of the heat flux distribution error for cut-cell meshes on the two-dimensional model problem. The errors in boundary-conforming cut cases are compared to the errors in cut meshes with small volume ratios that have either been merged out or remain.

have different convergence trends. The heat flux distribution error for the grids with small volume ratios (i.e. without merging) converges at suboptimal rates and a large error penalty has been added. With merging, the heat flux distribution error is approximately the same as the boundary-conforming cut mesh. The improved heat flux distribution error illustrates the importance of merging in the presence of small volume ratios.

The importance of merging becomes more evident in Figure 5-16, where the heat flux distribution is plotted over the range of grid refinements and solution orders. As the mesh is refined and the solution order is increased, regardless of merging, the heat flux distribution improves. In fact, highly resolved meshes with higher-order solutions (bottom right in Figure 5-16) do not show oscillatory behavior. In the under-resolved meshes without merging there is a large penalty in the heat flux distribution. In the context of an adaptive method, a larger loss of accuracy in under-resolved meshes is troublesome as it decreases the effectiveness of the error estimate.

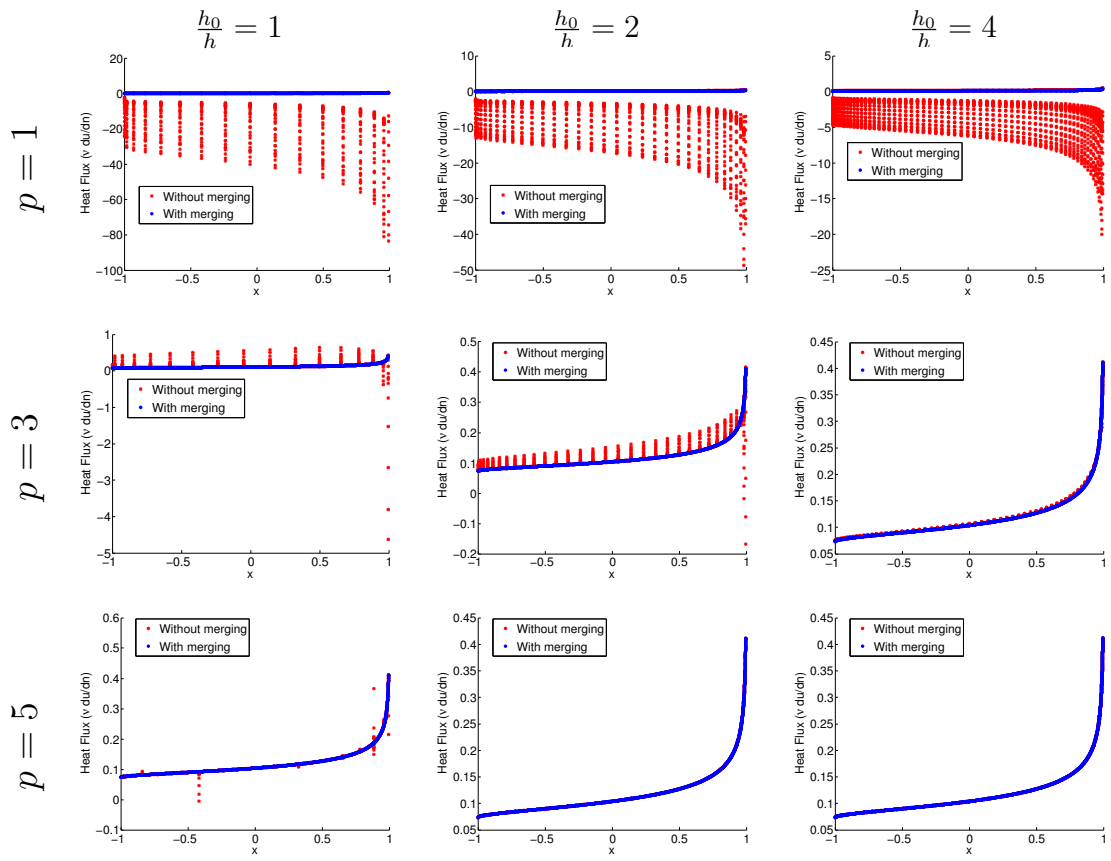


Figure 5-16: Boundary distributions of heat flux for the two-dimensional convection-diffusion problem using merged and non-merged cut grids.

Chapter 6

Output-Based Error Estimation and Adaptation

This chapter details the dual-weighted residual (DWR) method of Becker and Rannacher [20] and describes how output-based adaptation is applied to the cut-cell mesh generation technique.

6.1 Output-Based Error Estimation

The error estimation strategy employed in this work is based on the dual-weighted residual method (DWR) of Becker and Rannacher [20]. Extensive previous research has been done by Barth and Larson [13], Giles and Süli [53], Hartmann and Houston [59], Lu [82], Venditti [123], and Fidkowski [43]. The error estimation analysis presented here is closely related to the work of Yano [130].

Considering a semi-linear form with a solution, $\mathbf{u}_{h,p} \in V_{h,p}$, such that

$$R_{h,p}(\mathbf{u}_{h,p}, \mathbf{v}_{h,p}) = 0, \quad \forall \mathbf{v}_{h,p} \in V_{h,p}, \quad (6.1)$$

where $V_{h,p}$ is an appropriate finite dimensional functional space such as \mathcal{V}_h^p defined in Section 2.3. If consistency is assumed, the exact solution, $\mathbf{u} \in V$, satisfies the discrete approximation, $R_{h,p}(\mathbf{u}, \mathbf{v}_{h,p})$, $\forall \mathbf{v}_{h,p} \in V_{h,p}$. For a general output of interest, $J(\cdot)$, the adjoint or

dual problem is: find $\boldsymbol{\psi} \in V$ such that

$$R_{h,p}[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{v}, \boldsymbol{\psi}) = J'[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{v}), \quad \forall \mathbf{v} \in V_{h,p} + V.$$

The $R'_{h,p}[\cdot]$ and $J'[\cdot]$ are mean-value linearizations given by

$$\begin{aligned} R'_{h,p}[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{v}, \mathbf{w}) &= \int_0^1 R'_{h,p}[\theta \mathbf{u} + (1 - \theta) \mathbf{u}_{h,p}](\mathbf{v}, \mathbf{w}) d\theta, \\ J'[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{v}) &= \int_0^1 J'[\theta \mathbf{u} + (1 - \theta) \mathbf{u}_{h,p}](\mathbf{v}) d\theta, \end{aligned}$$

where the primed-bracket notation denotes the Frechét derivative and $\mathbf{v}, \mathbf{w} \in V_{h,p} + V$. Then, taking $\mathbf{v} = \mathbf{u}_{h,p} - \mathbf{u}$,

$$\begin{aligned} R'_{h,p}[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{u}_{h,p} - \mathbf{u}, \mathbf{w}) &= R_{h,p}(\mathbf{u}_{h,p}, \mathbf{w}) - \cancel{R_{h,p}(\mathbf{u}, \mathbf{w})} \xrightarrow{0} \\ J'[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{u}_{h,p} - \mathbf{u}) &= J(\mathbf{u}_{h,p}) - J(\mathbf{u}). \end{aligned}$$

The output error can be expressed as

$$\begin{aligned} \mathcal{E} = J(\mathbf{u}_{h,p}) - J(\mathbf{u}) &= J'[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{u}_{h,p} - \mathbf{u}) \\ &= R'_{h,p}[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{u}_{h,p} - \mathbf{u}, \boldsymbol{\psi}) \\ &= R_{h,p}(\mathbf{u}_{h,p}, \boldsymbol{\psi}). \end{aligned} \tag{6.2}$$

If the exact adjoint solution is known, the error can be computed exactly by evaluating the primal residual. By defining the adjoint residual,

$$R_{h,p}^\psi[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{v}, \mathbf{w}) \equiv R'_{h,p}[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{v}, \mathbf{w}) - J'[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{v}), \quad \forall \mathbf{v}, \mathbf{w} \in V_{h,p} + V,$$

and the discrete adjoint, $\boldsymbol{\psi}_{h,p} \in V_{h,p}$ such that

$$R_{h,p}^\psi[\overline{\mathbf{u}\mathbf{u}_{h,p}}](\mathbf{v}_{h,p}, \boldsymbol{\psi}_{h,p}) = 0, \quad \forall \mathbf{v}_{h,p} \in V_{h,p}.$$

The output error can also be expressed in terms of the dual residual weighted by the exact

primal solution:

$$\begin{aligned}
\mathcal{E} = J(\mathbf{u}_{h,p}) - J(\mathbf{u}) &= J'[\overline{\mathbf{u}\mathbf{u}}_{h,p}](\mathbf{u}_{h,p} - \mathbf{u}) \\
&= R'_{h,p}[\overline{\mathbf{u}\mathbf{u}}_{h,p}](\mathbf{u}_{h,p} - \mathbf{u}, \boldsymbol{\psi}_{h,p}) - R_{h,p}^\psi[\overline{\mathbf{u}\mathbf{u}}_{h,p}](\mathbf{u}_{h,p} - \mathbf{u}, \boldsymbol{\psi}_{h,p}) \\
&= \cancel{R_{h,p}(\mathbf{u}_{h,p}, \boldsymbol{\psi}_{h,p})} \xrightarrow{0} - \cancel{R_{h,p}(\mathbf{u}, \boldsymbol{\psi}_{h,p})} \xrightarrow{0} - R_{h,p}^\psi[\overline{\mathbf{u}\mathbf{u}}_{h,p}](\mathbf{u}_{h,p} - \mathbf{u}, \boldsymbol{\psi}_{h,p}) \\
&= -R_{h,p}^\psi[\overline{\mathbf{u}\mathbf{u}}_{h,p}](\mathbf{u}_{h,p}, \boldsymbol{\psi}_{h,p}) + R_{h,p}^\psi[\overline{\mathbf{u}\mathbf{u}}_{h,p}](\mathbf{u}, \boldsymbol{\psi}_{h,p}) \\
&= R_{h,p}^\psi[\overline{\mathbf{u}\mathbf{u}}_{h,p}](\mathbf{u}, \boldsymbol{\psi}_{h,p}). \tag{6.3}
\end{aligned}$$

Since \mathbf{u} and $\boldsymbol{\psi}$ are not known, two approximations are used to enable the use of output-based error estimates. The first approximation replaces the exact mean value linearizations with linearizations about $\mathbf{u}_{h,p}$. $\boldsymbol{\psi}_{h,p} \in V_{h,p}$ is set to the solution of

$$R_{h,p}^\psi[\mathbf{u}_{h,p}](\mathbf{v}_{h,p}, \boldsymbol{\psi}_{h,p}) = 0, \quad \forall \mathbf{v}_{h,p} \in V_{h,p}, \tag{6.4}$$

where $R_{h,p}^\psi[\mathbf{u}_{h,p}](\mathbf{v}_{h,p}, \mathbf{w}_{h,p})$ is the adjoint residual based only on linearizations about \mathbf{u}_h :

$$R_{h,p}^\psi[\mathbf{u}_{h,p}](\mathbf{v}_{h,p}, \mathbf{w}_{h,p}) = R'_{h,p}[\mathbf{u}_{h,p}](\mathbf{v}_{h,p}, \mathbf{w}_{h,p}) - J'[\mathbf{u}_{h,p}](\mathbf{v}_{h,p}), \quad \mathbf{v}_{h,p}, \mathbf{w}_{h,p} \in V_{h,p}.$$

The second approximation replaces the exact errors, $\mathbf{u}_h - \mathbf{u}$ and $\boldsymbol{\psi}_h - \boldsymbol{\psi}$, with $\mathbf{u}_{h,p} - \tilde{\mathbf{u}}_{h,p'}$ and $\boldsymbol{\psi}_{h,p} - \tilde{\boldsymbol{\psi}}_{h,p'}$, where $\tilde{\mathbf{u}}_{h,p'}$ and $\tilde{\boldsymbol{\psi}}_{h,p'}$ are considered truth surrogate solutions. The truth surrogate solutions are sought from the $p + 1$ order piecewise-polynomial space. $\tilde{\mathbf{u}}_{h,p'}$ is obtained by solving Equation (6.1) approximately on $V_{h,p+1}$, using 10 time steps of pseudo time continuation. Each time step, described in Section 3.1, requires a single Newton step to approximate the solution update. A simpler block smoothing scheme [98] is found to result in an unreliable error estimation for problems with shocks and separation. As the objective is to enable robust and automated adaptation, the additional cost of Newton steps is deemed justifiable. The dual surrogate solution, $\tilde{\boldsymbol{\psi}}_{h,p'} \in V_{h,p'}$, is obtained by solving

$$R_{h,p'}^\psi[\tilde{\mathbf{u}}_{h,p'}](\mathbf{v}_{h,p'}, \boldsymbol{\psi}_{h,p'}) = 0, \quad \forall \mathbf{v}_{h,p'} \in V_{h,p'},$$

exactly.

The error expressions in Equations (6.2) and (6.3) can be approximated as:

$$\mathcal{E} = J(\mathbf{u}) - J(\mathbf{u}_{h,p}) \approx R_{h,p}(\mathbf{u}_{h,p}, \tilde{\boldsymbol{\psi}}_{h,p'}) \quad (6.5)$$

$$\approx R_{h,p}^{\psi}[\tilde{\mathbf{u}}_{h,p'}](\tilde{\mathbf{u}}_{h,p'}, \boldsymbol{\psi}_{h,p}). \quad (6.6)$$

Using the two error estimates, a local error indicator can be constructed on each element by averaging Equation (6.5) and (6.6). For each element κ , the error is

$$\eta_{\kappa} \equiv \frac{1}{2} \left[\left| R_{h,p}(\mathbf{u}_{h,p}, \tilde{\boldsymbol{\psi}}_{h,p'}|_{\kappa}) \right| + \left| R_{h,p}^{\psi}[\tilde{\mathbf{u}}_{h,p'}](\tilde{\mathbf{u}}_{h,p'}|_{\kappa}, \boldsymbol{\psi}_{h,p}) \right| \right], \quad (6.7)$$

where the notation, $|_{\kappa}$, indicates restriction to the element κ , and the absolute values are included to provide a conservative error estimate. The global error is approximated as

$$\mathcal{E} \approx \sum_{\kappa} \eta_{\kappa}.$$

The error estimate is not a bound, but an indicator of global output error.

6.2 Adaptation Strategy

Mesh adaptation is used to autonomously modify the discretization space to generate solutions that have a decreased output error. The objective of the mesh adaptation strategy is to generate a mesh that realizes the lowest output error for a given cost. The output error is estimated using the DWR framework discussed in Section 6.1 and degrees of freedom are used as a cost metric. Adaptation can be expressed as a discrete optimization problem:

$$\mathcal{T}_{h,\text{Opt.}} = \arg \min_{\mathcal{T}_h} \mathcal{E}^D(\mathcal{T}_h) \quad \text{s.t.} \quad \mathcal{C}^D(\mathcal{T}_h) = \text{DOF}_{\text{target}}.$$

$\mathcal{E}^D(\mathcal{T}_h)$ is the discrete output error function and $\mathcal{C}^D(\mathcal{T}_h)$ is the discrete cost function. The nodal connectivity within each instance of \mathcal{T}_h makes the discrete optimization problem intractable. In order to relax the discrete optimization problem the approximability of \mathcal{T}_h can be encoded in a Riemannian metric field, $\boldsymbol{\mathcal{M}} = \{\mathcal{M}(x)\}_{x \in \Omega}$ [22]. A mesh metric-field

duality exists such that

$$\begin{aligned}\mathcal{M} &= \text{ImpliedMetric}(\mathcal{T}_h) \\ \mathcal{T}_h &= \text{MeshGeneration}(\mathcal{M}).\end{aligned}$$

Figure 6-1 provides an example of the mesh metric-field duality. For each element in \mathcal{T}_h an implied metric can be defined as

$$\mathcal{M}_{\kappa,\text{imp}} = \text{ImpliedMetric}(\kappa), \quad \forall \kappa \in \mathcal{T}_h.$$

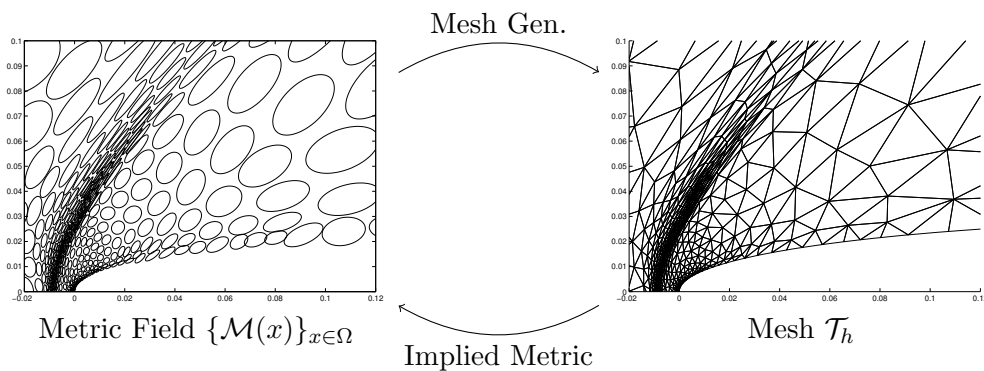


Figure 6-1: Mesh metric-field duality.

Multiple discrete meshes conform to a given metric field, but the approximability of each mesh is assumed to result in the same output error. The introduction of the metric field allows the discrete optimization problem to be recast:

$$\mathcal{M}_{\text{Opt.}} = \arg \min_{\mathcal{M}} \mathcal{E}(\mathcal{M}) \quad \text{s.t.} \quad \mathcal{C}(\mathcal{M}) = \text{DOF}_{\text{target}}.$$

The objective of each adaptation step is to move toward $\mathcal{M}_{\text{Opt.}}$, where $\mathcal{M}_{\text{Opt.}}$ is the metric field that equidistributes the elemental error indicator, η_{κ} , at a fixed degree of freedom. Each adaptation step follows the flow chart shown in Figure 6-2 and evolves remeshing of the simplex mesh using BAMG [64], which generates linear anisotropic metric conforming meshes.

Stepping toward $\mathcal{M}_{\text{Opt.}}$ at a fixed degree of freedom allows the adaptation strategy to

generate what will be termed a DOF-“optimal” mesh, which in this work is defined to have equidistributed elemental errors. DOF-“optimal” meshes are demonstrated in Section 6.2.7. Optimal mesh configurations for a higher-order discretization require significant grading of the mesh toward singularities[131]. The adaptation strategy, with the degree of freedom control, produces this grading through a series of adaptation steps.

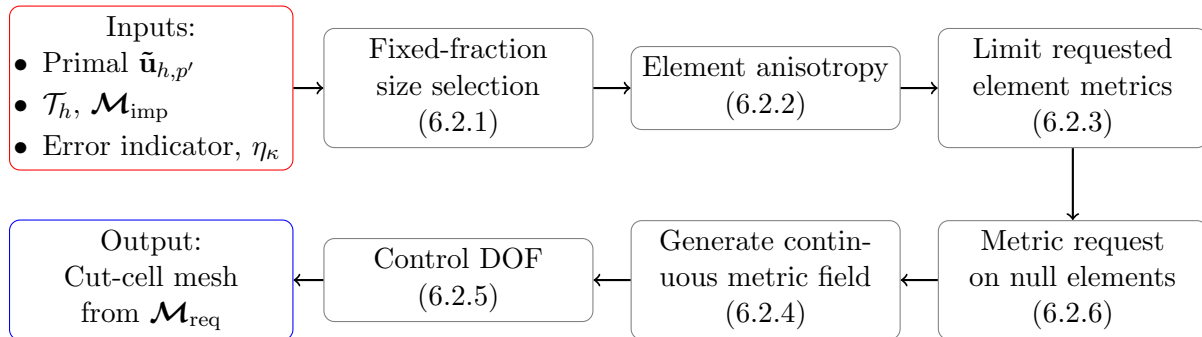


Figure 6-2: Flow chart detailing a single adaptation step.

For cut-cell meshes the output-based adaptation is performed on the background mesh described in Section 4.1. Though the elementwise error indicator and solution is only defined inside the computational domain, the optimal metric field must be based on the background mesh. Otherwise, after the first cut mesh, the adaptation would be restricted back to boundary conforming mesh generation and the flexibility of cut-cell meshing would be lost.

6.2.1 Fixed-Fraction Marking

A fixed-fraction marking strategy is used to control the size of each element in the requested metric field, \mathcal{M}_{req} . In the fixed-fraction marking strategy, the top f_r fraction of the elements with the largest error are marked for refinement and the bottom f_c fraction of the elements with the smallest error are marked for coarsening. Figure 6-3 illustrates the goal of the fixed-fraction adaptation to redistribute element areas and equidistribute the error. The initial mesh in Figure 6-3 (a) has a wide distribution of elemental errors. As the elements with high error are refined and the elements with low error are coarsened the error distribution tightens up in Figures 6-3 (b) and (c) and there is a decrease in total error.

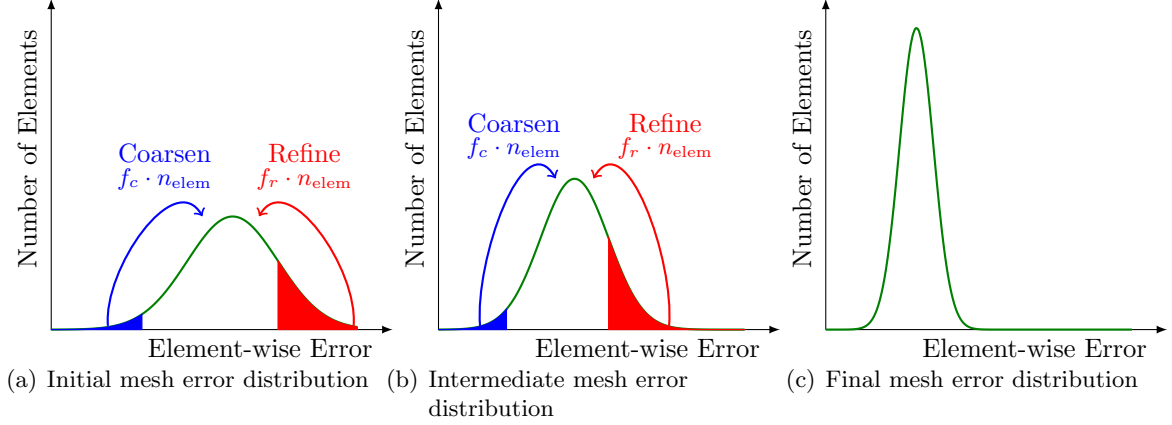


Figure 6-3: Fixed fraction adaptation strategy

The requested element size, A_{req} , is specified by

$$A_{\text{req}} = \alpha_{\text{ff}} A_{\text{imp}},$$

where A_{imp} is the current element size based on the implied metric and α_{ff} is the refinement rate which is set based on whether the element is marked for refinement, coarsening, or no change. The fixed-fraction marking strategy is only used to determine the area of the element in the requested metric field. Requested element shape is determined by the anisotropy detection strategy described in Section 6.2.2. For this work, the parameters are set to $f_r = f_c = 0.2$ and $\alpha_{\text{ff}} = 1/4$ and 2 for refinement and coarsening respectively.

Some clarifying remarks are in order to distinguish the fixed-fraction marking strategy from fixed-fraction adaptation based on hierarchical subdivision of elements. With hierarchical element subdivisions, fixed-fraction adaptation directly controls the change in degrees of freedom. The marking strategy employed here is only used to set the relative elemental areas in the requested metric field. The final requested metric field is scaled so the resulting mesh has the desired degrees of freedom as will be described in Section 6.2.5. The fixed-fraction strategy is a means to redistribute element areas in an attempt to equidistribute local errors. Metric-based adaptation allows for the equidistribution of element errors because it allows for continuous variation of element areas. Hierarchical subdivision, on the

other hand, only permits discrete mesh changes and cannot move away from the original mesh topology.

When the fixed-fraction marking strategy is applied to cut-cell meshes element area redistribution occurs on the background mesh. The elements with the largest error are still marked for refinement and the elements with the smallest error are still marked for coarsening, but the area change request acts on the background mesh. A_{imp} is the area of the background cut element which is updated based on α_{ff} .

6.2.2 Anisotropy Detection

In order to efficiently resolve shocks, boundary layers, and wakes encountered in the aerodynamics applications, the element's orientation and stretching must be aligned to the flow features. The anisotropy detection used here is based on the work by Venditti and Darmofal [124], which was extended to higher-order methods by Fidkowski and Darmofal [45]. The framework attempts to minimize the interpolation error of the solution within an element. The interpolation errors are controlled by the principal directions of the Riemannian metric tensor. The dominant principal direction is aligned with the direction of the maximum $p + 1$ derivative of the Mach number, $M_{\text{max}}^{(p+1)}$. The second principal direction is then selected to equidistribute the interpolation error in the two principal directions. Assuming the interpolation error in the Mach number converges at the rate of $r = p + 1$, the principal lengths, h_1 and h_2 , of the element anisotropy request metric, \mathcal{M}_{ani} , satisfy

$$\left(\frac{h_2(\mathcal{M}_{\text{ani}})}{h_1(\mathcal{M}_{\text{ani}})} \right)^r = \frac{M_{\text{max}}^{(p+1)}}{M_{\perp}^{(p+1)}}, \quad (6.8)$$

where $M_{\perp}^{(p+1)}$ is the derivative of the Mach number in the direction perpendicular to $M_{\text{max}}^{(p+1)}$. Since requested element size is selected by the fixed-fraction marking strategy, the requested anisotropic metric, \mathcal{M}_{ani} , is specified to have a determinant of unity. Following [43, 124], the Mach number is selected as a single scalar quantity to represent the solution behavior. Typically, the convergence rate, r , is assumed to be $p + 1$. However, the rate is reduced to $r = 1$ when the shock indicator (described in Section 2.4) is on, and the derivative quantities used in Equation (6.8) are replaced by the first derivatives, $M_{\text{max}}^{(1)}$ and $M_{\perp}^{(1)}$. The $p + 1$ derivative of the Mach number is obtained from the primal truth surrogate solution, $\tilde{\mathbf{u}}_{h,p'}$.

The dependence of the anisotropy request on the truth surrogate solution is another reason for performing the 10 pseudo-time steps to obtain a robust $p + 1$ approximate solution. Due to the possibility of an arbitrarily small M_{\perp} , the requested element aspect ratio is limited to 500 to reduce the demands on the mesh generator for improved robustness.

Using the area request A_r and the anisotropy metric \mathcal{M}_{ani} , the anisotropic element metric request, \mathcal{M}_{req} , is constructed as

$$\mathcal{M}_{\text{req}} = A_{\text{imp}}^{-2/d} \mathcal{M}_{\text{ani}}.$$

Every cut element in the computational domain possess its own requested anisotropy metric. A_{imp} is taken from the background element implied metric and \mathcal{M}_{ani} comes from $\tilde{\mathbf{u}}_{h,p'}$ in the cut element.

6.2.3 Limit Requested Element Metrics

Limits are placed on the allowable change from the implied metric to the requested metric on each element. The limits on element metric changes are in place to ensure mesh realizability and to maintain approximability. The assumption is made that if the changes to the implied metric field are small the mesh generator will be able to realize a mesh conforming to the requested metric field. A region also exists within which the local error estimates and anisotropy requests are trusted. The metric change limits maintain the approximability within that trust region. Simultaneous matrix reduction [21] interpolates between the implied metric and the requested metric, \mathcal{M}_{imp} and \mathcal{M}_{req} respectively, and determines the metric which conforms to the limits.

Let M_{imp} be the symmetric positive definite matrix given by the implied metric, \mathcal{M}_{imp} , and let M_{req} be the symmetric positive definite matrix given by the requested metric, \mathcal{M}_{req} . An interpolation from M_{imp} to M_{req} is given by

$$M(t) = E^{-T} \begin{bmatrix} 1/h_1^2(t) & 0 \\ 0 & 1/h_2^2(t) \end{bmatrix} E^{-1}, 0 \leq t \leq 1,$$

where $E = [e_1, e_2]$ is the matrix of eigenvectors of $M_{\text{imp}}^{-1} M_{\text{req}}$. In the interpolation $M(0) = M_{\text{imp}}$ and $M(1) = M_{\text{req}}$. Arithmetical progression is selected for element area, $A(t) =$

$h_1(t)h_2(t)$, and the principle stretching direction, $h_1(t)$, i.e.,

$$\begin{aligned} h_1(t) &= h_1^{\text{imp}} + t(h_1^{\text{req}} - h_1^{\text{imp}}), \\ A(t) &= A^{\text{imp}} + t(A^{\text{req}} - A^{\text{imp}}). \end{aligned}$$

$h_2(t)$ is then given by $A(t)/h_1(t)$.

The requested metric after limiting will be given by $M(\ell)$ where ℓ is the maximum value of t such that the principle directions of $M(\ell)$ stay within the max refinement and coarsening factors. The principle direction constraints are equivalent to

$$\frac{1}{R_{\text{lim}}} \leq \frac{h_i(t)}{h_i^{\text{imp}}} \leq C_{\text{lim}}, \text{ for } i = 1, 2.$$

In this work the maximum refinement is $R_{\text{lim}} = 4.0$ and the maximum coarsening is $C_{\text{lim}} = 2.0$. For the first principle direction, h_1 , the constraint on ℓ_1 is given by

$$\begin{aligned} \text{If } \frac{1}{R_{\text{lim}}} &\leq \frac{h_1^{\text{req}}}{h_1^{\text{imp}}} \leq C_{\text{lim}}, & \ell_1 &= 1, \\ \text{Else if } h_1^{\text{req}} &> C_{\text{lim}}h_1^{\text{imp}}, & \ell_1 &\text{ s.t. } h_1(\ell_1) = C_{\text{lim}}h_1^{\text{imp}}, \\ \text{Else ,} & & \ell_1 &\text{ s.t. } h_1(\ell_1) = \frac{1}{R_{\text{lim}}}h_1^{\text{imp}}, \end{aligned}$$

and for h_2 ,

$$\begin{aligned} \text{If } \frac{1}{R_{\text{lim}}} &\leq \frac{A^{\text{req}}h_1(0)}{A(0)h_1^{\text{req}}} \leq C_{\text{lim}}, & \ell_2 &= 1, \\ \text{Else if } \frac{A^{\text{req}}}{h_1^{\text{req}}} &> C_{\text{lim}}\frac{A^{\text{imp}}}{h_1^{\text{imp}}}, & \ell_2 &\text{ s.t. } \frac{A(\ell_2)}{h_1(\ell_2)} = C_{\text{lim}}\frac{A^{\text{imp}}}{h_1^{\text{imp}}}, \\ \text{Else ,} & & \ell_2 &\text{ s.t. } \frac{A(\ell_2)}{h_1(\ell_2)} = \frac{1}{R_{\text{lim}}}\frac{A^{\text{imp}}}{h_1^{\text{imp}}}. \end{aligned}$$

The limiting ℓ is given by $\min(\ell_1, \ell_2)$. For each element the limited requested metric is the tensor corresponding to $M(\ell)$. Figure 6-4 provides an example of the limited requested element metric.

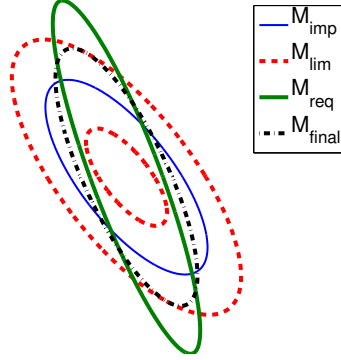


Figure 6-4: An example of a limited metric which corresponds to the maximum element coarsening.

6.2.4 Generation of Continuous Metric Field

A C^0 metric field is constructed from the piecewise constant requested element metrics on the background mesh. At each background grid node the requested element metrics from the elements surrounding it are averaged. Barycentric averaging in the length space is used so that the metric at each node is:

$$\mathcal{M}_{\text{node}} = \left(\frac{1}{\dim(\mathcal{N}_{\text{node}})} \sum_{\kappa \in \mathcal{N}_{\text{node}}} \mathcal{M}_{\kappa}^{-\frac{1}{2}} \right)^{-2}.$$

$\mathcal{N}_{\text{node}}$ is the set of all elements surrounding the node. In building $\mathcal{N}_{\text{node}}$, care is taken for cut elements. The requested metric for a cut element is not passed to all the background nodes of the parent background element. As shown in Figure 6-5, the requested metric for cut element A is passed to nodes 1 and 2 but not node 3.

6.2.5 Metric Request Construction and Explicit Degree of Freedom Control

The requested Riemannian metric field, $\mathcal{M}_{\text{req}} = \{\mathcal{M}_{\text{node}}(x)\}_{x \in \Omega}$, is scaled in order to control the adapted mesh's degrees of freedom. Following the work on the discrete-continuous mesh duality proposed by Loseille [80], the degrees of freedom of a mesh conforming to a

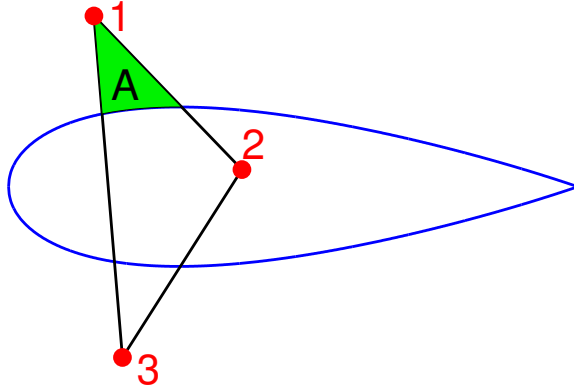


Figure 6-5: Multiply-cut element where the requested metric for element A is passed to nodes 1 and 2 but not node 3.

Riemannian metric field is approximated by

$$\text{DOF}(\mathcal{M}_{\text{req}}) = \int_{\Omega} C_{p,\kappa} \sqrt{\det(\mathcal{M}_{\text{req}})} dx,$$

where $C_{p,\kappa}$ is a constant depending on the solution order and the element shape. For example, for a p -th order polynomial simplex element in two dimension, $C_{p,\kappa} = (2/\sqrt{3})(p+1)(p+2)$.

The final degree of freedom controlled metric field is

$$\mathcal{M}_{\text{req,final}} = \left(\frac{\text{DOF}_{\text{target}}}{\text{DOF}(\mathcal{M}_{\text{req}})} \right)^{2/d} \mathcal{M}_{\text{req}}. \quad (6.9)$$

In the case of cut-cell mesh adaptation the degree of freedom control is performed on the background mesh. The only difference in degree of freedom control between cut-cell and boundary-conforming meshes is that $C_{p,\kappa}$, in the cut cell case, is set to zero for null background elements.

6.2.6 Building Metric Request for Null Cut Elements

The elements within the computational domain all fall under the control of the fixed-fraction size selection and the anisotropy shape selection described previously. In a cut-cell mesh the background grid also contains a set of null elements, those completely outside the computa-

tional domain. In order to perform metric based adaptation, a metric request must be made for each null element.

The most straightforward option to generate null element metric requests is to use each element's implied metric. However, there are two motivations to consider when generating improved null element metric requests. First, the cut elements intersecting a viscous solid wall boundary form a wake-like feature in the background mesh. The wake-like feature is a result of high mesh grading for boundary layer resolution on the side of the computational domain. Figure 6-6 provides an example of the wake-like viscous wall feature. The C^0 constructed metric field on the background mesh needs to contain the desired anisotropy request to resolve the wake-like feature of the viscous wall. Since Barycentric averaging is used to pass the element request to background nodes, the correct anisotropy will exist in the metric field if the null elements connected to cut elements have a similar anisotropy request. A second motivation to consider for the null elements metric request stems from the degree of freedom control. When using the explicit degree of freedom control the possibility exists that at each adaptation iteration the metric field is scaled to refine every element, i.e. when $\text{DOF}(\mathcal{M}) < \text{DOF}_{\text{target}}$, $\mathcal{M}_{\text{req,final}}$ will result in more elements than \mathcal{M}_{req} . If $\text{DOF}(\mathcal{M})$ is less than $\text{DOF}_{\text{target}}$ over the course of multiple adaptation iterations, and the implied metric is used as the requested metric, the entire null region of the background will be filled with an increasingly large number of elements, slowing down the mesh generation and the cutting processes.

Background null element layers help define the requested null element metrics. Figure 6-7 shows an example of the null element layers. The layers are generated through face connectivity such that each element's layer index is one plus the minimum of the layer indices of its face neighbors. In other words, the layer index is defined as the minimum number of elements that must be visited by passing through faces before reaching a cut element.

For each null element, the requested metric is taken as a scaled Barycentric average of the requested metrics of its faces neighbors one layer up. In order to help control the total number of null background elements, an element area growth rate of $\text{GR} = 1.1$ is applied to each layer of null background elements. For example, in Figure 6-7, null element C would have a metric request equal to the metric request for the background element of cut element

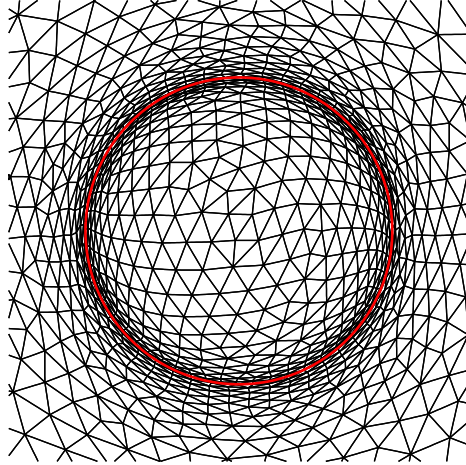


Figure 6-6: Cut elements intersecting a viscous wall form a wake-like feature in the background mesh.

A. Similarly, E 's metric request would be equal to B 's. For element D the metric request would be $\mathcal{M}_{req,D} = \text{GR}^{\text{index}} \left[\frac{1}{2} \left(\mathcal{M}_{req,C}^{-\frac{1}{2}} + \mathcal{M}_{req,E}^{-\frac{1}{2}} \right) \right]^{-2}$, where the index of element D is 1.

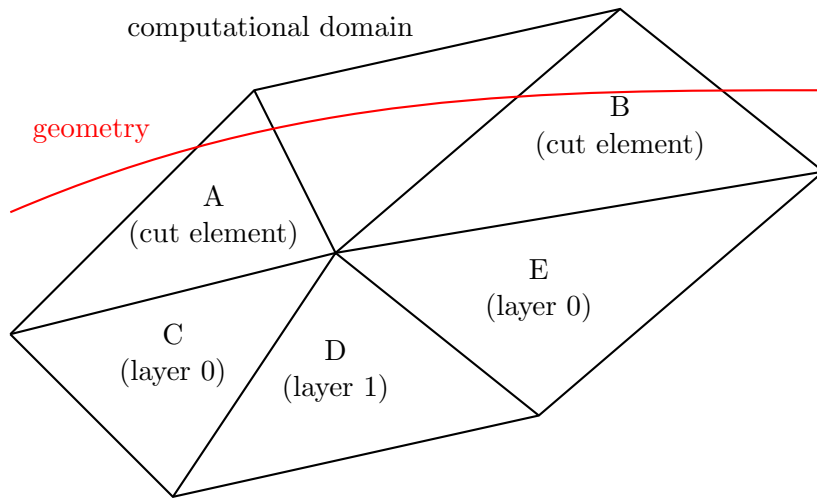


Figure 6-7: Example describing the process of forming requested metrics on null elements.

6.2.7 DOF-“Optimal” Mesh

The adaptation strategy presented in this section enables the generation of meshes at a fixed degree of freedom that have equidistributed local element error estimates. These meshes

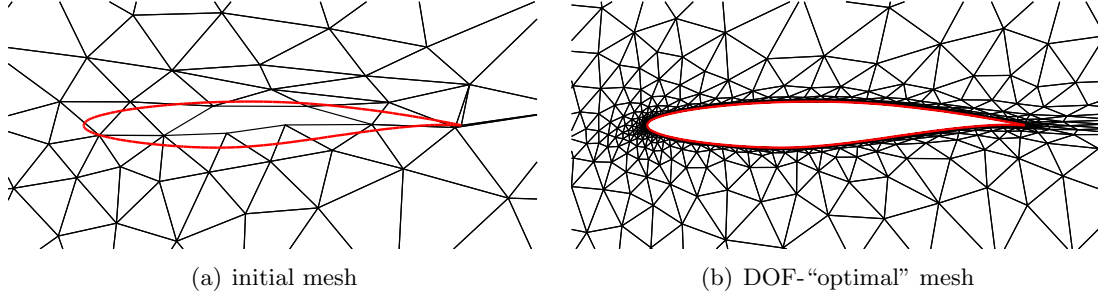


Figure 6-8: Example of the initial and DOF-“optimal” meshes for subsonic RAE2822 RANS-SA flow ($M_\infty = 0.3$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$, $p = 3$, $\text{DOF} = 40k$).

are considered DOF-“optimal.” The degree of freedom control mechanism presented in Section 6.2.5 is the key feature that allows for the generation of the DOF-“optimal” meshes.

Subsonic RANS-SA flow over a RAE2822 airfoil at $M_\infty = 0.3$, $Re_c = 6.5 \times 10^6$, and $\alpha = 2.31^\circ$ is used to demonstrate the generation of DOF-“optimal” meshes. A set of meshes based on subsonic Euler flow, with the coarsest initial mesh shown in Figure 6-8(a), are a set of starting points for the $p = 3$ turbulent cases. Adaptation is performed at a fixed degree of freedom of $40k$. Figure 6-9 shows the error estimate, the error, the drag, and the degree of freedom history for the set of meshes considered. After 15 iterations, each cases stabilizes at $40k$ degrees of freedom and an error of 0.1 drag counts. The meshes seen after adaptation iteration 15 are considered DOF-“optimal” with an example shown in Figure 6-8(b). The error is computed relative to the drag coefficient obtained on an adapted $p = 3$, $\text{DOF} = 160,000$ mesh.

The adaptation leads to an error reduction of three orders of magnitude for each initial mesh with the initial degrees of freedom considered ranging from $3k$ to $30k$. The error reduction is achieved through aggressive redistribution of element areas and the use of highly anisotropic elements in the boundary layers and the wake. After 15 adaptation iterations the meshes are optimized for the subsonic RANS-SA flow. However, the c_d error estimates continue to fluctuate around one drag count. The constant degree of freedom adaptation produces a family of DOF-“optimal” meshes. For this example, all the meshes after adaptation iteration 15 are considered DOF-“optimal” and the meshes have similar metric fields but slightly different triangulations. To account for the fluctuation in the error estimate and the output quantities, all the results in this thesis present DOF-“optimal” errors in terms

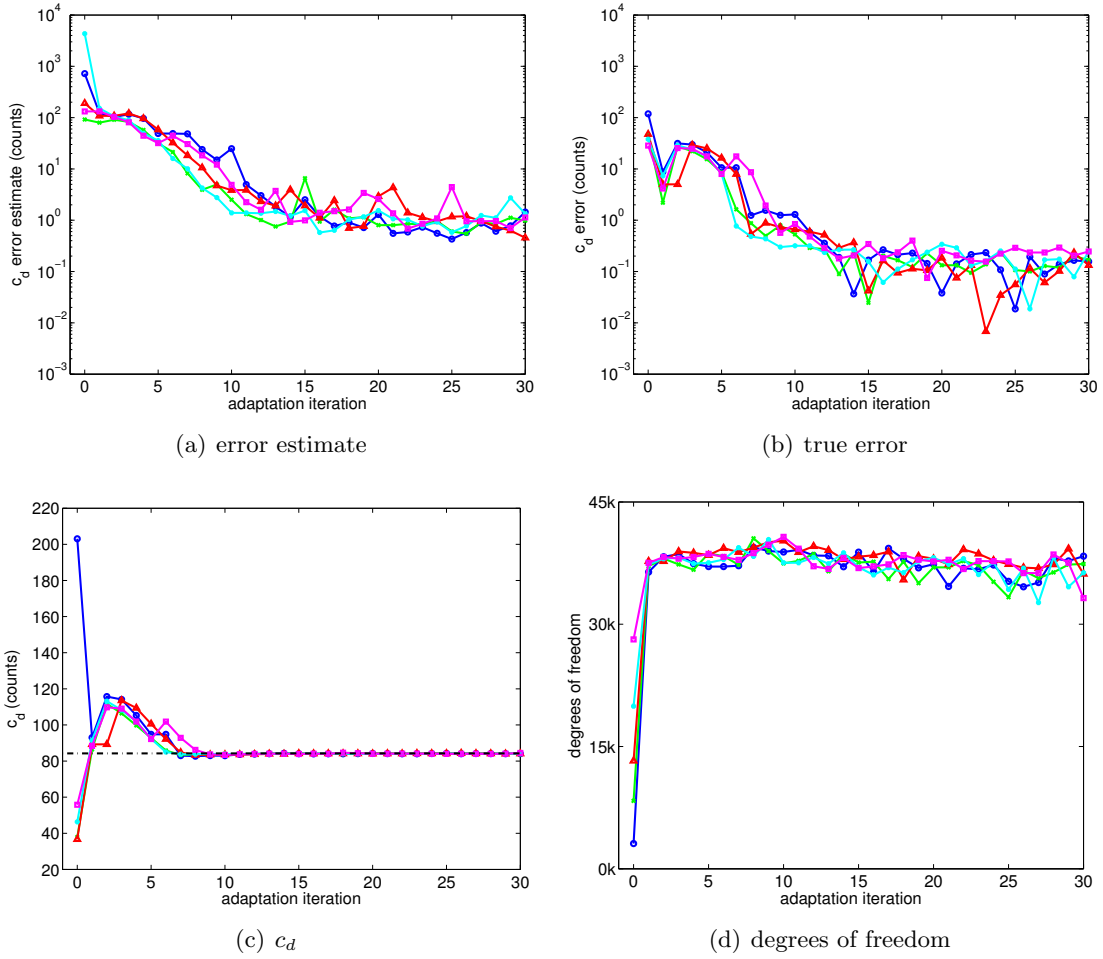


Figure 6-9: The the error estimate, the error, the drag, and the degree of freedom adaptation history for a set of initial meshes applied to the subsonic RAE2822 RANS-SA flow ($M_\infty = 0.3$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$, $p = 3$, $DOF = 40,000$).

of a family of meshes. All cases are run at least 10 adaptation iterations beyond where the error estimate stabilizes. The output quantities are presented as the envelope of outputs from the last five DOF-“optimal” meshes for each degree of freedom. The error estimate values are obtained by averaging the five error estimates.

Chapter 7

Results

This chapter quantifies the impact on solution efficiency (defined as accuracy per degree of freedom) in the transition from boundary-conforming elements to simplex cut cells. In Section 7.1 a wide range of aerospace problems are solved, including subsonic through supersonic regimes and complex geometries. In Section 7.2, a parametric study is performed over angle of attack to construct a lift curve for a high-lift multi-element airfoil and Section 7.3 compares the computational cost of boundary-conforming and cut-cell solutions for this parameter study.

7.1 Comparison of Boundary-Conforming and Cut-Cell Solution Efficiency

To demonstrate the difference in solution efficiency between cut-cell and elastically-curved meshes five flows are considered: NACA0012 subsonic Euler flow, RAE2822 subsonic RANS-SA flow, RAE2822 transonic RANS-SA flow, NACA0006 supersonic RANS-SA flow, and MSC8 transonic RANS-SA flow.

7.1.1 NACA0012 Subsonic Euler

The first case that quantifies the difference in solution efficiency between cut-cell and boundary-conforming meshes is subsonic Euler flow over a NACA0012 airfoil at $M_\infty = 0.5$ and $\alpha = 2.0^\circ$. The flow solution and initial cut-cell mesh is shown in Figure 7-1. Figure 7-2 shows the convergence in the envelopes of drag coefficients and the error estimate for the family of

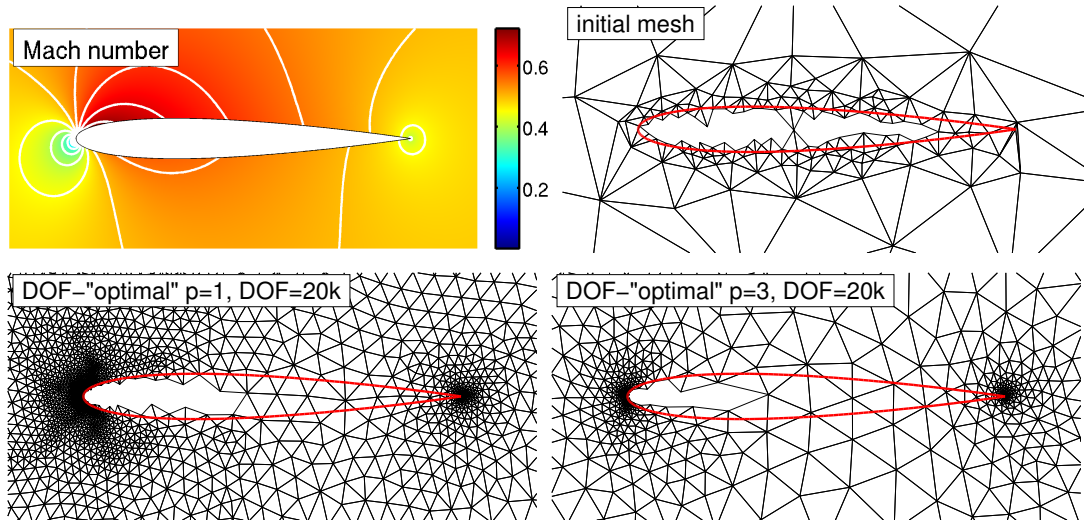


Figure 7-1: Mach number distribution, initial mesh, and the DOF-“optimal” meshes for subsonic NACA0012 Euler flow ($M_\infty = 0.5$, $\alpha = 2.0^\circ$). The Mach contour lines are in 0.05 increments.

DOF-“optimal” cut-cell and boundary-conforming meshes. The largest solution efficiency gap exists for the $p = 3$, $\text{DOF} = 2.5k$ mesh, where the efficiency gap is defined as the difference between the cut-cell and boundary conforming error estimate at a given degree of freedom. There are only 250 elements in these meshes with high grading toward the leading and trailing edge, so the accuracy gap is understandable. The cut-cell meshes achieve the same rate of convergence, and the gap in solution efficiency is almost non-existent other than for the $p = 3$, $\text{DOF} = 2.5k$ case.

In terms of the solution accuracy, the $p = 2$ and $p = 3$ discretizations are superior to the $p = 1$ discretization for high-fidelity simulation requiring the drag error estimate of less than 1 count. Figures 7-2 (a) and (b) show the envelopes of the drag coefficient for the families of DOF-“optimal” meshes. $p = 2$ and 3 exhibit quicker convergences to the reference solution computed using $p = 3$, $\text{DOF} = 40k$ for both the cut-cell and boundary-conforming cases. The subsonic NACA0012 Euler flow indicates that if the trailing edge singularity is handled correctly, the benefit of high spacial accuracy can be realized for both cut-cell and boundary conforming meshes. The DOF-“optimal” $p = 1$ and $p = 3$ meshes shown in Figure 7-1 illustrate the mesh grading which is required to manage the trailing edge singularity.

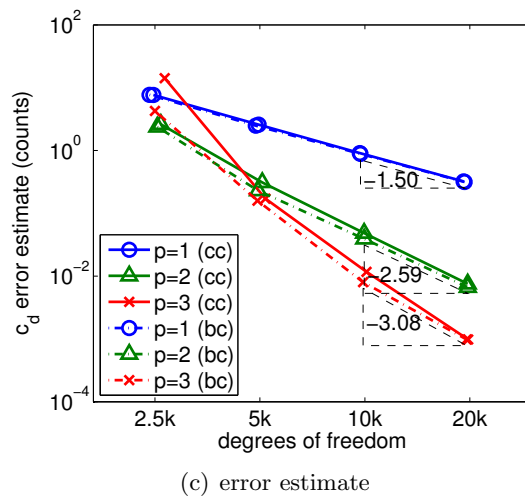
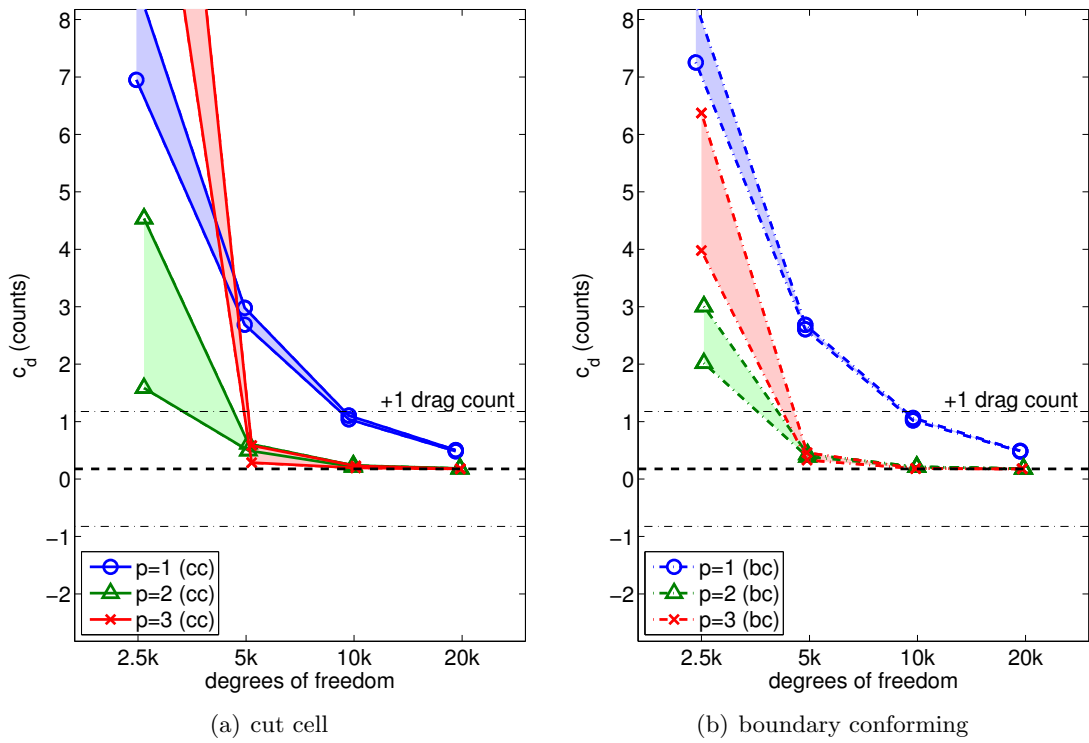


Figure 7-2: Envelopes of drag coefficients and c_d error estimates for subsonic NACA0012 Euler flow ($M_\infty = 0.5$, $\alpha = 2.0^\circ$).

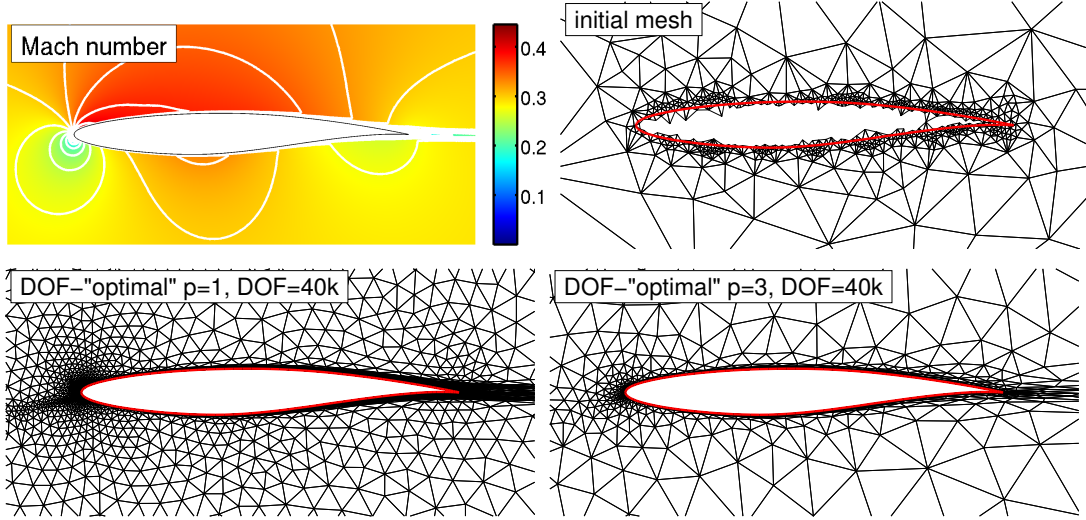


Figure 7-3: Mach number distribution, initial mesh, and the DOF-“optimal” meshes for for subsonic RAE2822 RANS-SA flow ($M_\infty = 0.3$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$). The Mach contour lines are in 0.05 increments.

7.1.2 RAE2822 Subsonic RANS-SA

The same RANS-SA flow over a RAE2822 airfoil as in Section 6.2.7 is considered to demonstrate the difference in solution efficiency between cut-cell and boundary-conforming meshes ($M_\infty = 0.3$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$). Figure 7-4(c) shows the convergence in the drag coefficient error estimate for DOF-“optimal” cut-cell and boundary-conforming meshes. For this subsonic RANS-SA flow a noticeable loss in solution efficiency exists between boundary-conforming and cut-cell meshes. The resolution demands for this case are simple, requiring only the boundary layer (including the boundary layer edge), wake, and stagnation streamline to be fully resolved. The inability of the cut-cell method to resolve curved geometries limits the solution efficiency. However, the cut-cell meshes do achieve the same rate of convergence as their boundary-conforming companions, and the gap in solution efficiency decreases as the total number of degrees of freedom increases.

For this case, the benefit of higher-order discretizations is not quite as clear as for the subsonic Euler flow over a NACA0012 airfoil. The drag coefficient error estimate in Figure 7-4(c) does show that, for all degrees of freedom considered, the $p = 2$ and $p = 3$ solutions provide a lower error estimate than $p = 1$, but the largest error estimate shown is 10 drag counts. The drag coefficient envelopes from the families of DOF-“optimal” meshes, shown

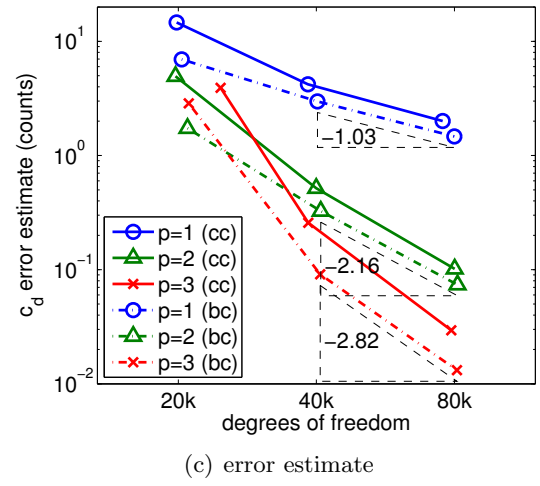
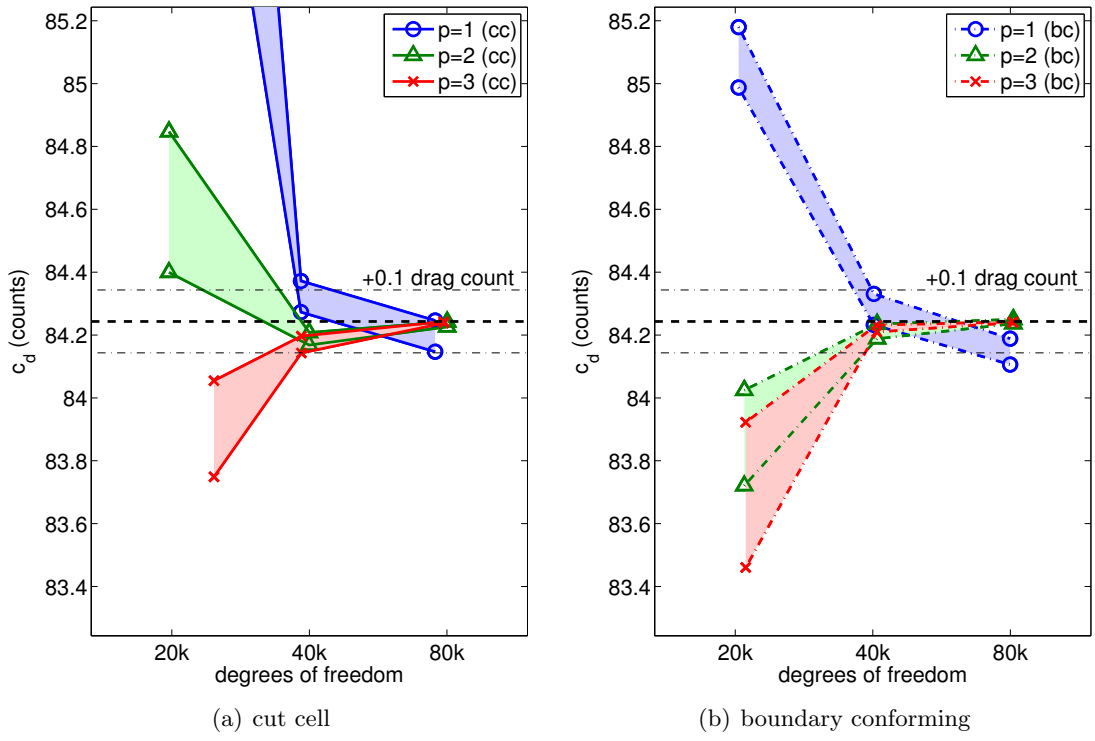


Figure 7-4: Envelopes of drag coefficients and c_d error estimates for subsonic RAE2822 RANS-SA flow ($M_\infty = 0.3$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$).

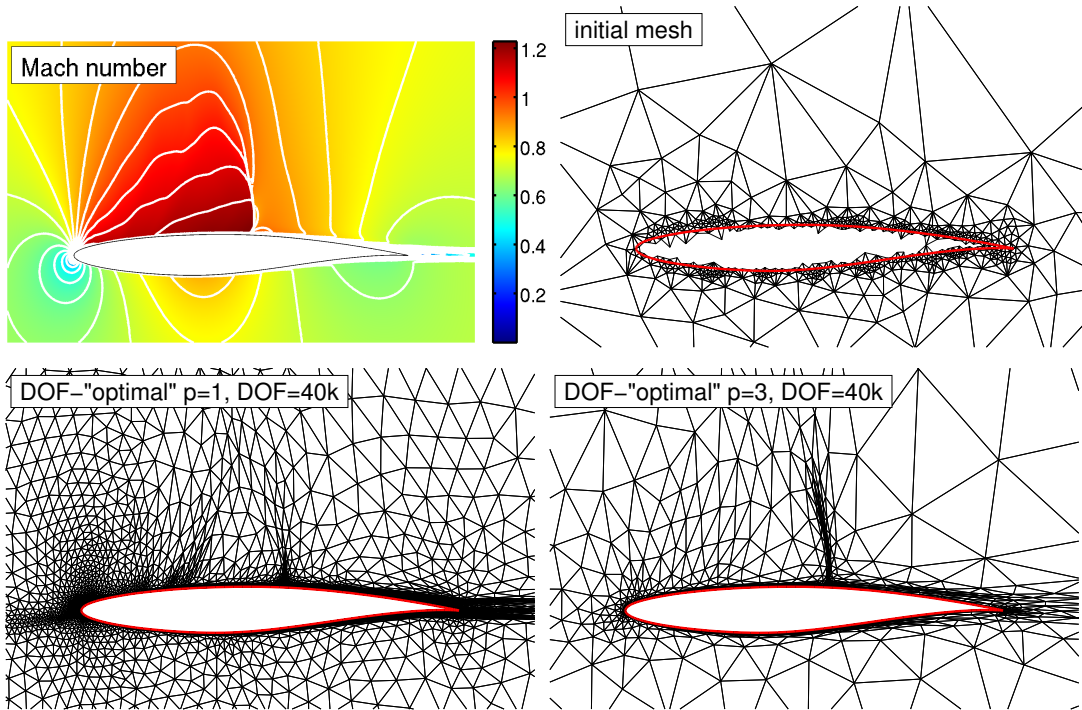


Figure 7-5: Mach number distribution, initial mesh, and the DOF-“optimal” meshes for subsonic RAE2822 RANS-SA flow ($M_\infty = 0.729$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$). The Mach contour lines are in 0.025 increments.

in Figures 7-4(a) and (b) point out that, once 40k degrees of freedom are employed, all the solution orders provide a c_d value within 0.1 drag counts of the reference solution taken from a $p = 3$, $DOF = 160k$ case. However, the $p = 1$ results have a band of about 0.1 counts, at 40k and 80k degrees of freedom, while the $p = 2$ and $p = 3$ c_d bands are much tighter and decrease with degrees of freedom. For this simple RANS-SA flow the true benefit of the $p = 3$ discretization is seen after the c_d errors are already less than engineering-required accuracy. However, the higher-order discretizations perform no worse than $p = 1$ discretization for this case.

7.1.3 RAE2822 Transonic RANS-SA

Having shown the effectiveness of the cut-cell method for subsonic cases, the solution efficiency is now quantified in the presence of shocks. The case considered is drag-based

adaptation for transonic flow over an RAE2822 airfoil at $M_\infty = 0.729$, $Re_c = 6.5 \times 10^6$, and $\alpha = 2.31^\circ$.

The Mach number distribution and the drag-adapted meshes obtained using a $p = 3$ discretization and $40k$ degrees of freedom are shown in Figure 7-5. The mesh is graded aggressively toward the airfoil surface, the boundary layer edges, and the shock. The Mach contour indicates that the combination of anisotropic grid refinement and the shock capturing algorithm enables sharp resolution of the shock.

Figure 7-6 shows the drag output envelopes and the associated error estimates for the families of DOF-“optimal” meshes. The error estimates, shown in Figure 7-6(c), show that for this transonic RANS-SA case the cut-cell method achieves the same level of solution efficiency as the boundary-conforming method, except for the $p = 3$ case at low degrees of freedom which has a small gap. This case requires aggressive mesh grading toward the shock as well as the boundary layer and wake. The flow has become more complicated with more features to resolve so the impact of the loss of cut-cell solution efficiency from resolving the curved geometry is decreased and cut-cell meshes provide nearly the same level of accuracy as boundary conforming meshes.

With the use of an adaptive strategy that provides arbitrary anisotropy and aggressive mesh grading toward the shock feature, the $p = 2$ discretization is more efficient than the $p = 1$ discretization at estimated error levels of 10 drag counts. For a higher-fidelity solution, the $p > 1$ discretizations are superior. The convergence of the drag envelopes, shown in Figures 7-6(a) and (b), confirm the enhanced drag convergence of the higher-order discretizations. Since, the higher-order discretizations provide a more efficient representation of the smooth boundary layer feature, they can use more degrees of freedom to resolve the shock. Thus, even though the higher-order method does not improve the efficiency of resolving the shock, at a fixed degree of freedom count they can be more accurate than $p = 1$ solutions for transonic problems.

Figure 7-6(c) also exemplifies the limit of applying higher-order discretizations to complex flow problems with an insufficient number of degrees of freedom. The $p = 3$ discretization at $20k$ degrees of freedom is outperformed by the $p = 1$ and $p = 2$ discretizations. 2,000 elements are inadequate to resolve both the boundary layer and the shock, even with $p = 3$

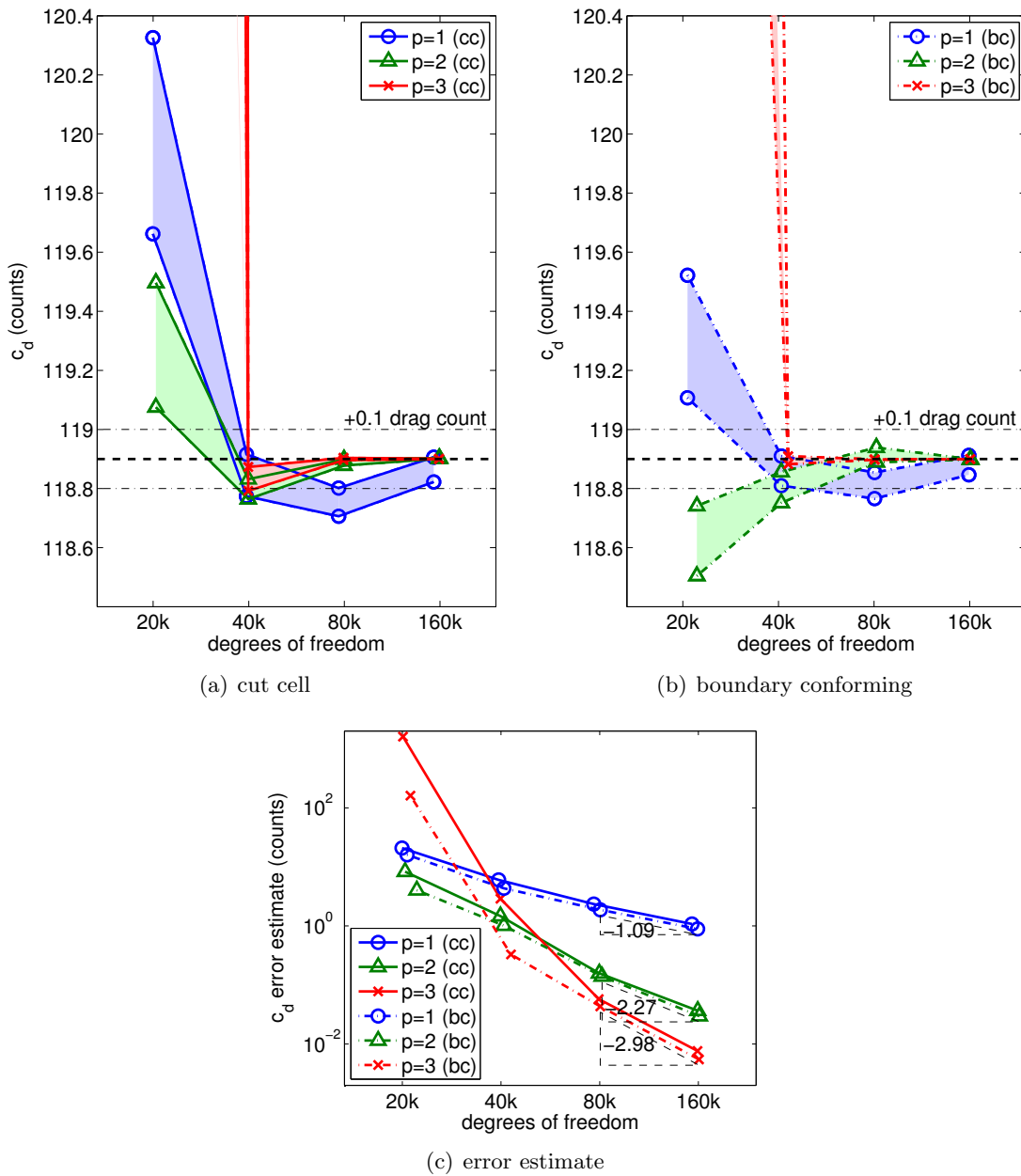


Figure 7-6: Envelopes of drag coefficients and c_d error estimates for transonic RAE2822 RANS-SA flow ($M_\infty = 0.729$, $Re_c = 6.5 \times 10^6$, $\alpha = 2.31^\circ$).

polynomials. For low- and moderate-fidelity simulations, using a minimum number of degrees of freedom, the lower-order discretizations are more accurate.

7.1.4 NACA0012 Supersonic RANS-SA

The second shock problem considered is a supersonic flow over a NACA0006 airfoil with the flow condition $M_\infty = 2.0$, $Re_c = 10^6$, and $\alpha = 2.0^\circ$. Two outputs of interest will be considered: the drag and the pressure signal 50 chords below the airfoil given by

$$J(u) = \int_{\Gamma_{\text{line}}} (p(u) - p_\infty)^2 ds,$$

where p_∞ is the free stream pressure and Γ_{line} runs from $(70c, -50c)$ to $(125c, -50c)$.

The Mach number distribution and the drag based DOF-“optimal” meshes for the $p = 1$ and $p = 3$ discretization having $80k$ degrees of freedom are shown in Figure 7-7. The meshes show aggressive refinement toward shock and boundary layer singularities. The adjoint Mach cone emerging from the training edge is particularly visible in the $p = 1$ mesh. Downstream of the adjoint Mach cone the mesh resolution decreases, as expected, as discretizations errors outside of the cone are inconsequential for drag calculations. However, the shock and wake still appear to be resolved with anisotropic elements which is an artifact of the Mach number based anisotropy leading to large, but stretched elements.

Again, like the transonic RAE2822 presented in Section 7.1.3, the cut-cell method’s drag error estimate convergence shown in Figure 7-8(c) achieves the same solution efficiency as the boundary-conforming meshes. For the sonic boom problem, the resolution of the bow shock is of utmost importance and there is no difference in shock resolution between the two mesh generation methods.

The convergence of the drag error estimate for the problem shows that, due to the low regularity of the dominant flow feature, the benefit of the $p = 2$ and $p = 3$ discretizations is difficult to realize even with the adaptive algorithm. The higher-order discretizations reduce the number of elements required to resolve the boundary layer sufficiently so that the method is more efficient than the $p = 1$ discretization overall. This shock problem also uncovers the limit of the effectiveness of higher-order discretizations at low degrees of freedoms. At $40k$ degrees of freedom the $p = 3$ discretization is not able to compete with the $p = 1$ and $p = 2$

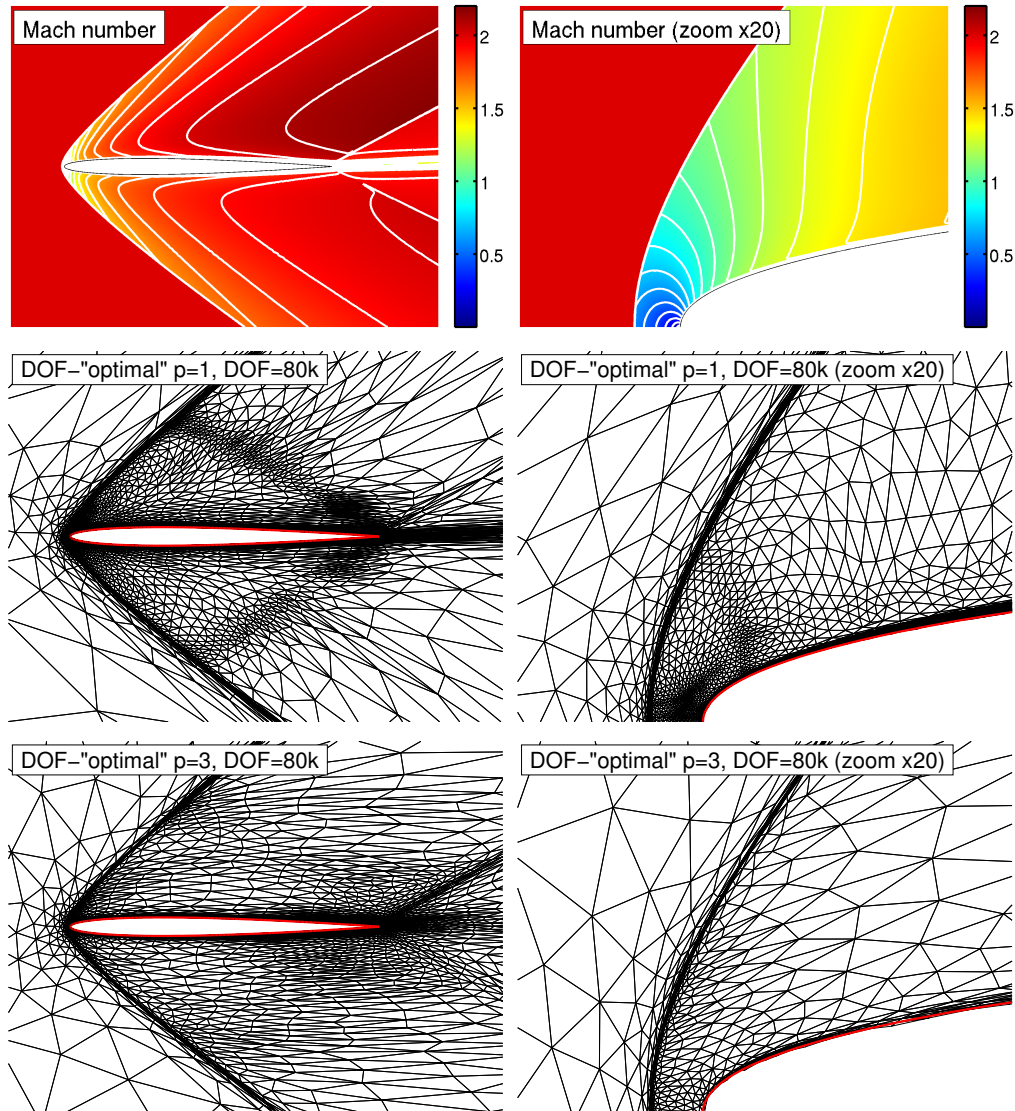


Figure 7-7: The Mach number distribution for the supersonic NACA0006 RANS-SA flow ($M_\infty = 2.0$, $Re_c = 10^6$, $\alpha = 2.0^\circ$) and the DOF-“optimal” meshes obtained for $p = 1$ and $p = 3$ at $80k$ degrees of freedom adapting to drag. The Mach contour lines are in 0.1 increments.

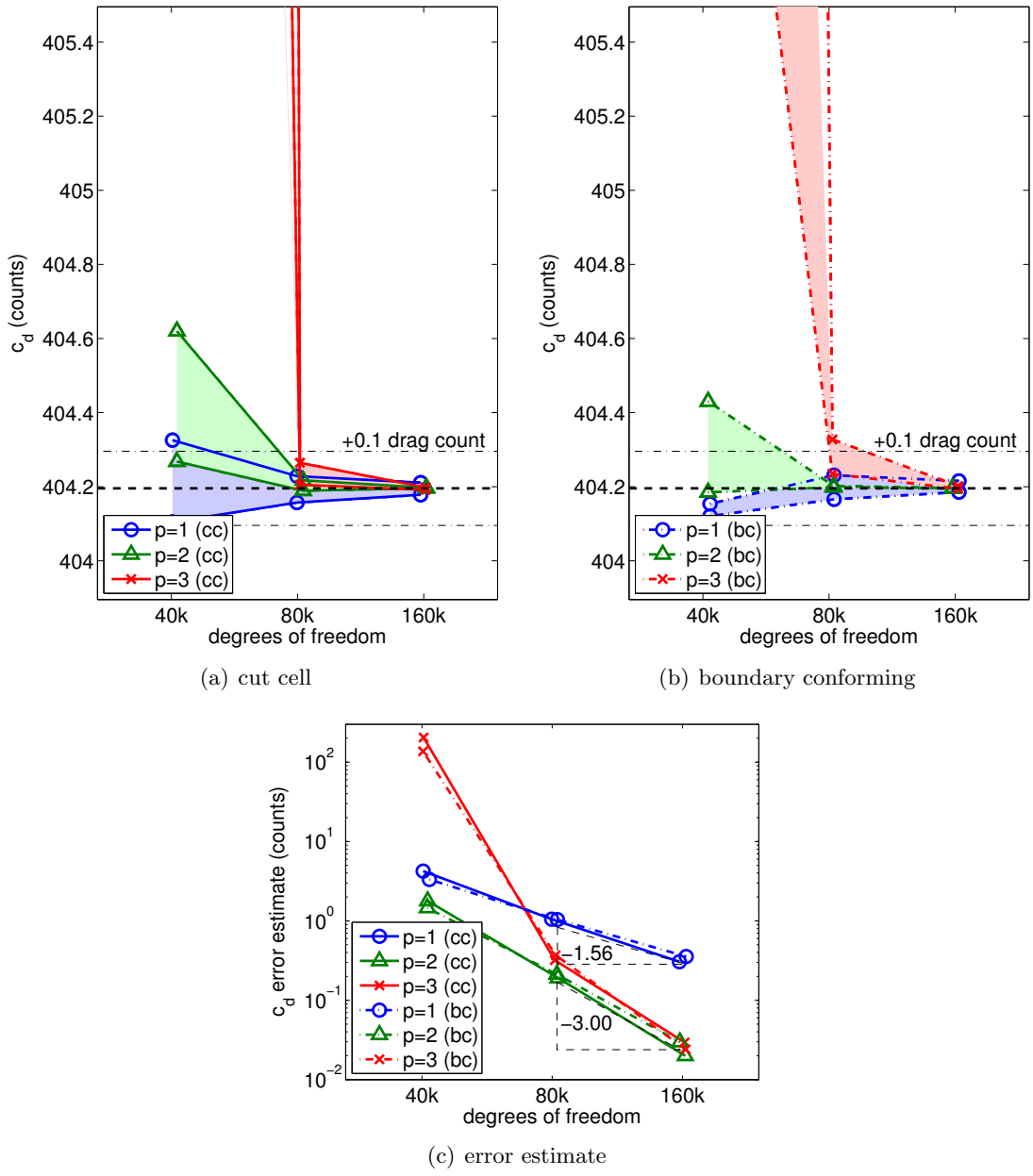


Figure 7-8: Envelopes of drag coefficients and c_d error estimates for supersonic NACA0006 RANS-SA flow ($M_\infty = 2.0$, $Re_c = 10^6$, $\alpha = 2.0^\circ$).

discretizations due to the inability to resolve the shock with a sufficient number of elements. In fact, the increase in element count for the $p = 2$ discretization at $80k$ and $160k$ degrees of freedom is more effective than the higher-spatial accuracy of the $p = 3$ discretization at the same degree of freedom count. The c_d values in Figures 7-8(a) and (b) confirm the superior performance of the adaptive $p = 2$ discretization, showing faster convergence toward the stationary value, but also illustrate the struggles of the $p = 3$ discretization for this flow.

The pressure signal based adaptation for the supersonic NACA0012 flow leads to the same conclusions on solution efficiency as the drag based adaptation. Figure 7-9 shows the envelopes of the pressure signal and the error estimate. Again, with the more complicated flow features away from the boundary, the cut-cell method achieves the same level of solution efficiency as the boundary-conforming meshes. The higher-order discretizations also outperform the $p = 1$ discretization in providing high fidelity output evaluation.

Figure 7-10 shows the pressure perturbation, $(p(u) - p_\infty) / p_\infty$, the mass adjoint, and the $p = 1$ and $p = 3$ DOF-“optimal” meshes at $80k$ degrees of freedom. Anisotropic elements are used to resolve the boundary layer and propagate the shock signature. The meshes also demonstrate the behavior of output-based adaptation to focus only on discretization errors which influence the output of interest. The shock above the airfoil is not resolved nearly as sharply as the lower shock because, as the adjoint shows, the upper shock has no influence on the pressure signal. Unfortunately, the meshes also show that anisotropic elements are generated in the flow direction downstream of the trailing shock. The added anisotropy downstream of the trailing shock is not appropriate for this flow and these meshes are not optimal for providing high accuracy pressure signals. The unnecessary anisotropy is due to artificial variations in flow quantities along the shock [12] which are then convected downstream and lead to gradients in the Mach number normal to the convective direction. The anisotropy detection algorithm, based on the higher-order derivatives of the Mach number, attempts to capture the artificial streaks. With each adaptation iteration the streaks actually become worse as the stretched elements do a better job of convecting the artificial variation downstream. This case highlights the shortcomings of an anisotropy detection algorithm based only on the derivatives of the Mach number and is a motivation for future work on anisotropy detection.

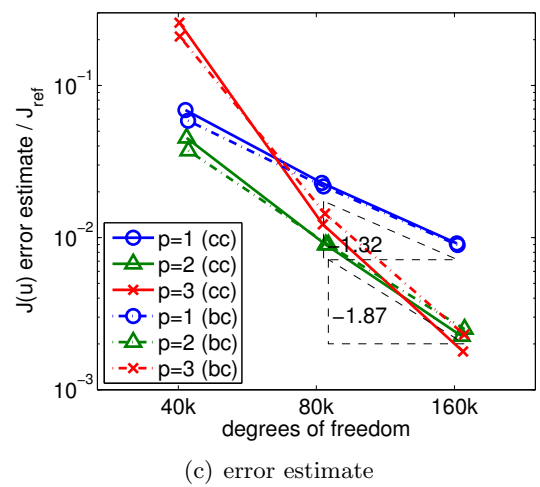
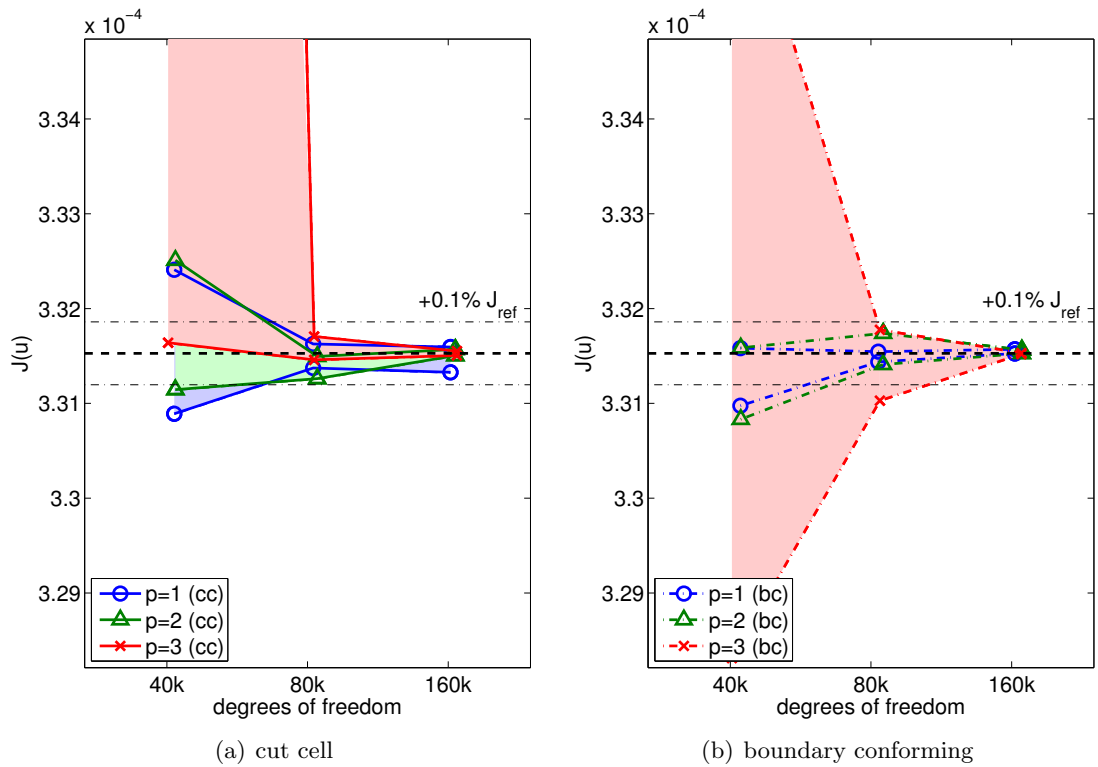


Figure 7-9: Envelopes of pressure signal and error estimates for supersonic NACA0006 RANS-SA flow ($M_\infty = 2.0$, $Re_c = 10^6$, $\alpha = 2.0^\circ$).

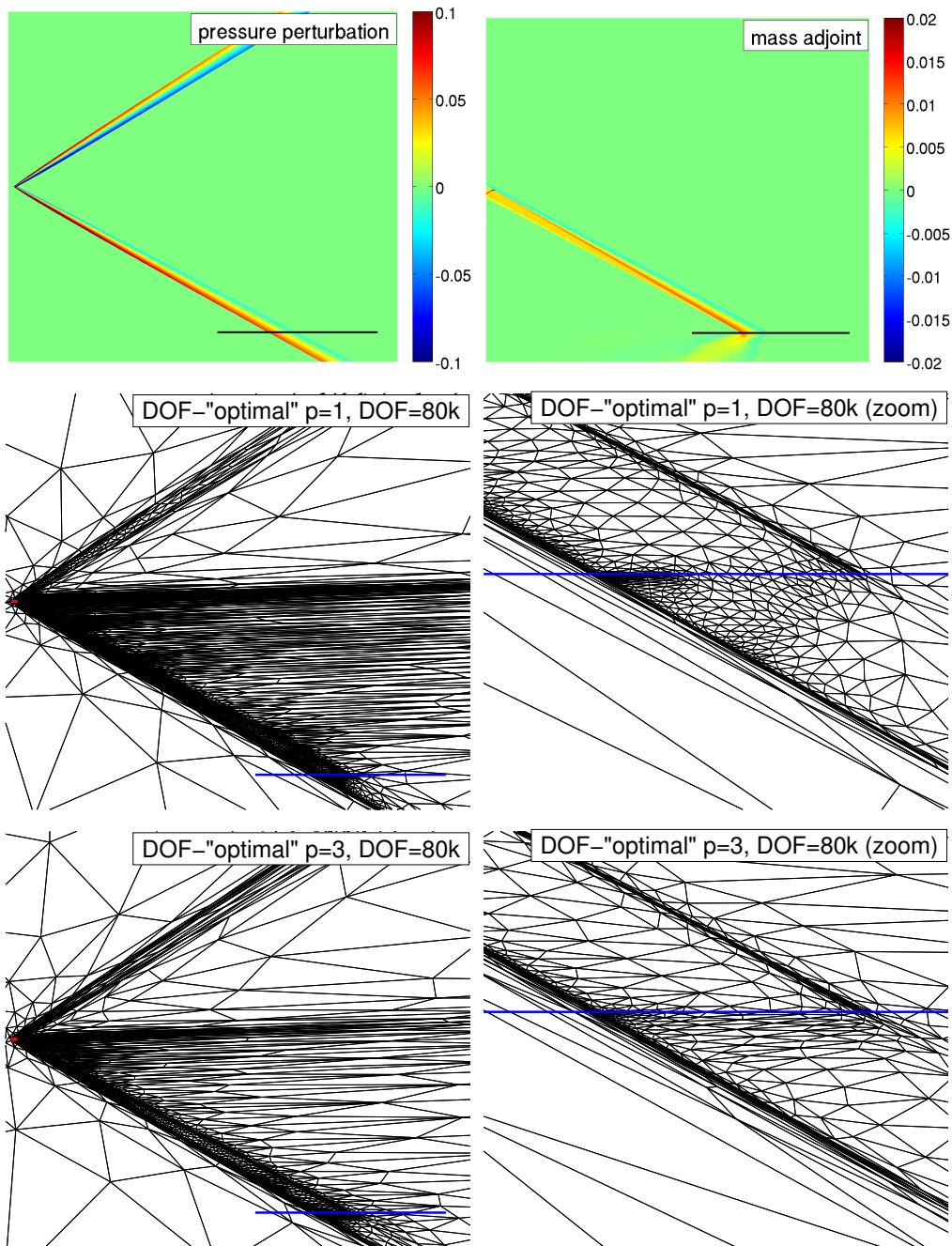


Figure 7-10: The pressure perturbation distribution, $(p(u) - p_\infty) / p_\infty$, for the supersonic NACA0006 RANS-SA flow ($M_\infty = 2.0$, $Re_c = 2.0 \times 10^7$, $\alpha = 2.5^\circ$) and the DOF-“optimal” meshes obtained for $p = 1$ and $p = 3$ at 80k degrees of freedom adapting to the pressure signal 50 chords below the airfoil.

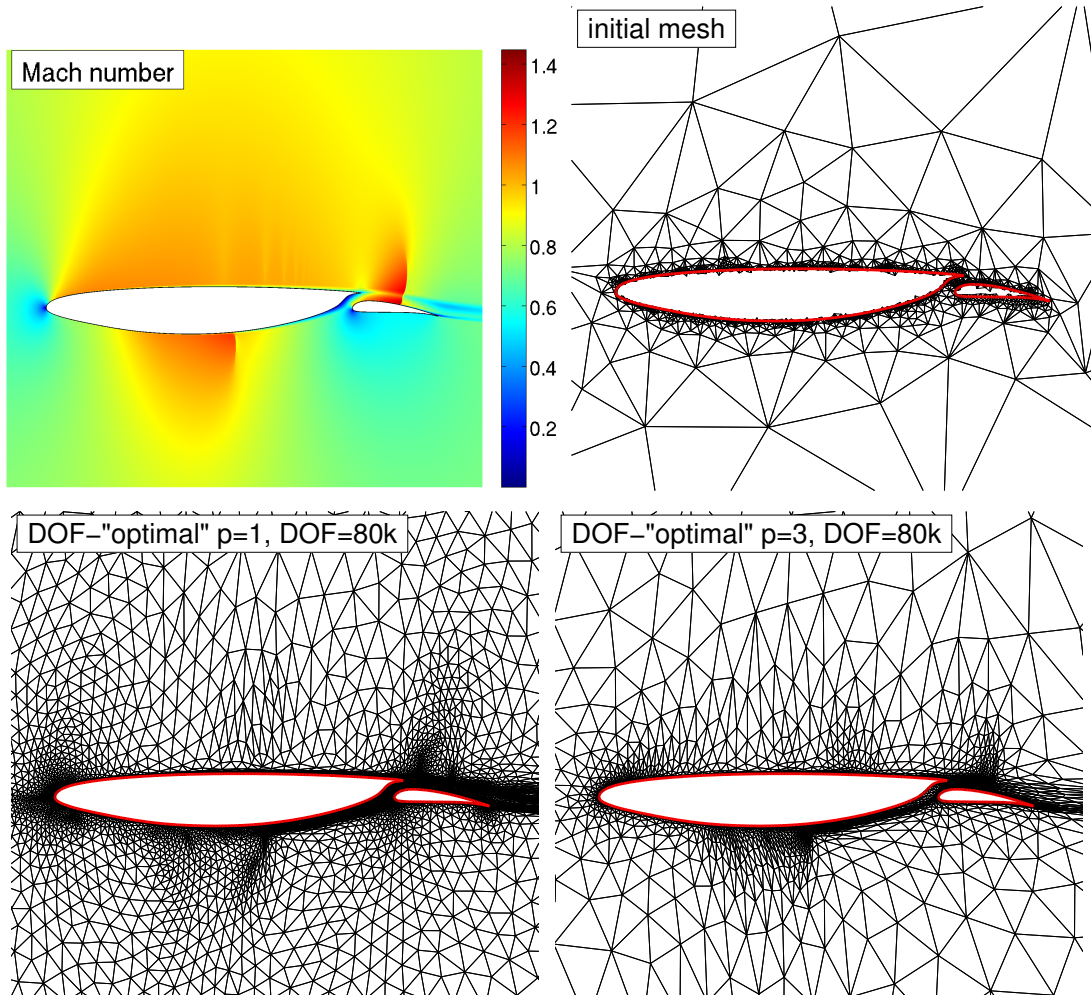


Figure 7-11: Mach number distribution, initial mesh, and the DOF-“optimal” meshes for transonic MSC8 RANS-SA flow ($M_\infty = 0.775$, $Re_c = 2.0 \times 10^7$, $\alpha = -0.7^\circ$).

7.1.5 Multi-element Supercritical 8 Transonic RANS-SA

The final case which will be used to quantify the difference in solution efficiency between boundary-conforming and cut-cell meshes is transonic flow over a multi-element supercritical 8 (MSC8) airfoil from Drela[40] with blunted trailing edges. The flow conditions are $M_\infty = 0.775$, $Re_c = 2.0 \times 10^7$, and $\alpha = -0.7^\circ$. The flow field for the MSC8, shown in Figure 7-11, is complex with the main body wake passing through the shock rising from the flap. The case presents a good challenge for the non-linear solution technique presented in Chapter 3.

Figure 7-12 shows the drag output envelopes and the associated error estimates for the

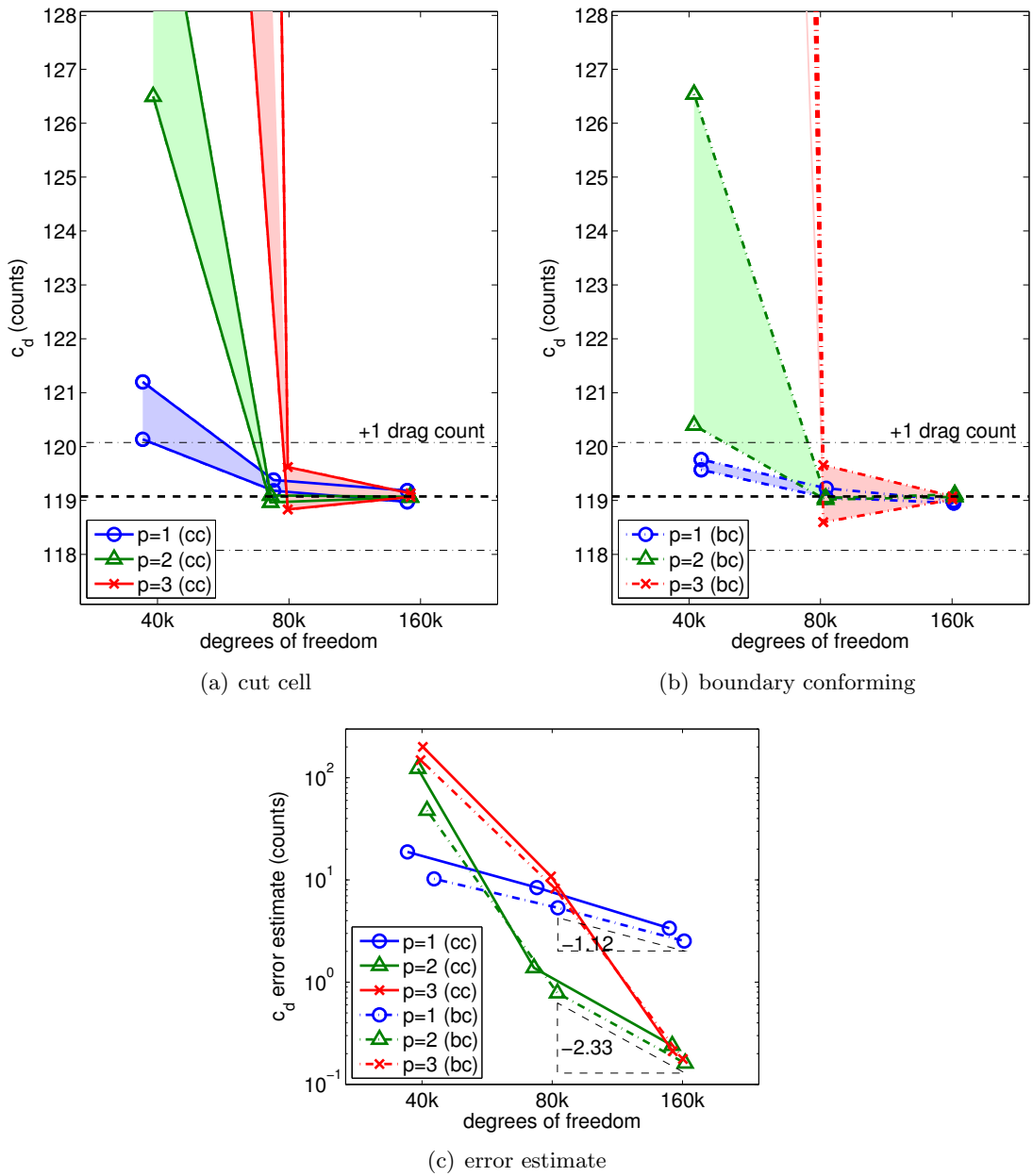


Figure 7-12: Envelopes of drag coefficients and c_d error estimates for transonic MSC8 RANS-SA flow ($M_\infty = 0.775$, $Re_c = 2.0 \times 10^7$, $\alpha = -0.7^\circ$).

families of DOF-“optimal” meshes. The error estimates, shown in Figure 7-12(c), show that for this transonic RANS-SA case the cut-cell method achieves the same level of solution efficiency as the boundary-conforming method. The MSC8 airfoil again demonstrates the limit of the $p = 3$ discretization for low degrees of freedom. The $p = 3$ discretization is outperformed in terms of c_d error estimate by the $p = 1$ discretization through $80k$ degrees of freedom. In terms of the envelopes of drag coefficient shown in Figure 7-12(a) and (b), it appears that at $80k$ degrees of freedom the $p = 1$ discretization provides a more accurate solution than the $p = 3$ discretization.

7.1.6 Summary

The results presented in this section have quantified the loss in solution efficiency in the conversion to cut-cell meshes from boundary-conforming meshes. Over a range of airfoil geometries and flow conditions, the cut-cell method compares well to and is competitive with its boundary-conforming counterpart. For all the cases considered, the cut-cell meshes achieve the same asymptotic rate of convergence in the output error estimates. For cases which are dominated by singular perturbations on the geometry, like the subsonic RANS-SA RAE2822 presented in Section 7.1.2, a loss of solution efficiency does exist for the cut-cell method. The geometry representation using cut linear background elements limits the cut-cell method as it is unable to effectively resolve higher-order geometry with a small number of elements. As the total number of degrees of freedom are increased, the solution efficiency gap decreases since the local curvature of the airfoils decrease relative to the element spacing along the tangential direction. The implication of the decrease in the solution efficiency gap is that for high-fidelity simulations cut-cell meshes are competitive with boundary-conforming meshes.

As the complexity of the cases considered increase, whether in terms of geometry or flow features like shocks, there is also a decrease in the difference between cut-cell and boundary-conforming solution efficiencies. The cut-cell method is equally capable of resolving wakes and shocks as the boundary-conforming meshes. So, as the relative importance of near geometry resolution, the cut-cell and boundary-conforming capabilities become more similar.

Though the results presented here have demonstrated that the cut-cell meshes are competitive with boundary-conforming meshes, a few words of caution are needed. These results

are based on the adaptation strategy presented in Section 6.2 which has a goal of generating DOF-“optimal” meshes where the local elemental errors are equidistributed. For each case, equidistributing the error for the lowest degree of freedom family of meshes is most difficult and is limited by the adaptation and meshing mechanics. For every case, 10 extra adaptation iterations were used beyond where the error estimate stabilized (discussed in Section 6.2.7) in an attempt to remove any limitations from the mechanics. However, the possibility remains that, particularly for the low degree of freedom cases, the boundary-conforming strategy are more limited than the cut-cell strategy by the meshing mechanics due to the necessity of conforming to the geometry.

7.2 DOF-Controlled Adaptation for Parameter Sweeps

7.2.1 Fixed Mesh vs. Adaptive Mesh

The explicit degree of freedom control based adaptation is beneficial for performing parameter sweeps. The explicit control allows the adaptation to begin from a DOF-“optimized” mesh, go on to vary a parameter, and then return a new DOF-“optimal” mesh at the same degree of freedom using a small number of adaptive iterations. The effectiveness of adaptation for parameter sweeps is demonstrated by constructing the lift curve for the McDonnell Douglas Aerospace (MDA) three-element airfoil (30P-30N) [72]. For the lift curve, the freestream Mach number and Reynolds number are set to $M_\infty = 2.0$ and $Re_c = 9 \times 10^6$, and the angle of attack is varied from 0.0° to 24.5° . $p = 2$ polynomials at $90k$ degrees of freedom are used for each mesh.

In order to show the benefit of adaptation, the lift curve for DOF-“optimal” meshes is compared to the lift curve based on a fixed mesh. For the comparison, the fixed mesh is taken as the DOF-“optimal” mesh for $\alpha = 8.1^\circ$. Figure 7-13(a) shows the lift curves obtained using the fixed and adaptive meshes for both the boundary-conforming and the cut-cell method. The lift curves show that the fixed mesh closely matches the adaptive result for $0^\circ < \alpha < 20^\circ$. However, for $\alpha > 20^\circ$, the lift calculation on the fixed mesh becomes unreliable and the c_l is significantly underestimated due to premature separation. For the parameter sweep performed here, the selection of cut-cell or boundary conforming meshes has little impact on the lift curves.

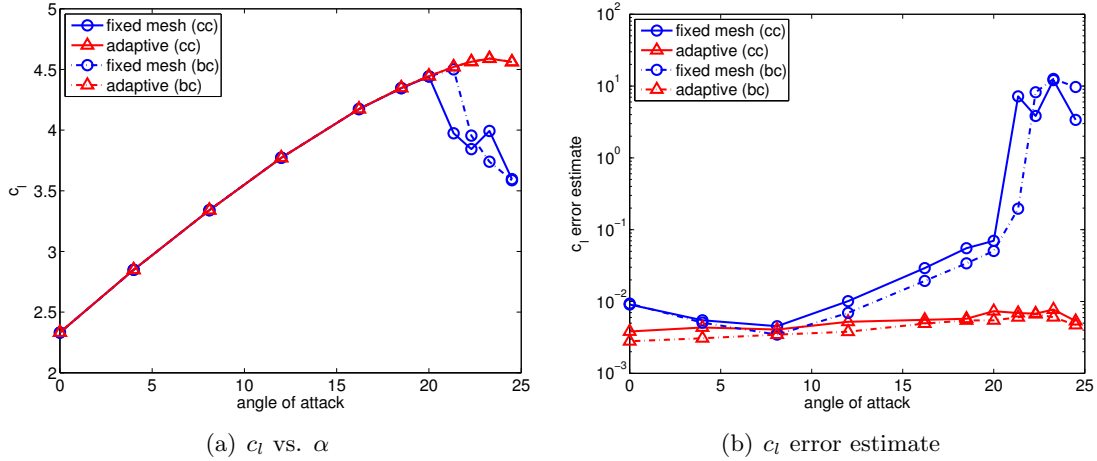
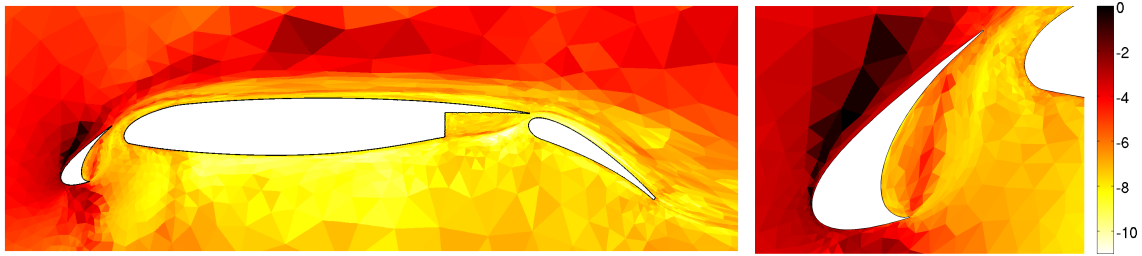


Figure 7-13: The lift curve and the c_l error obtained using the fixed mesh and adaptive meshes for the three-element MDA airfoil.

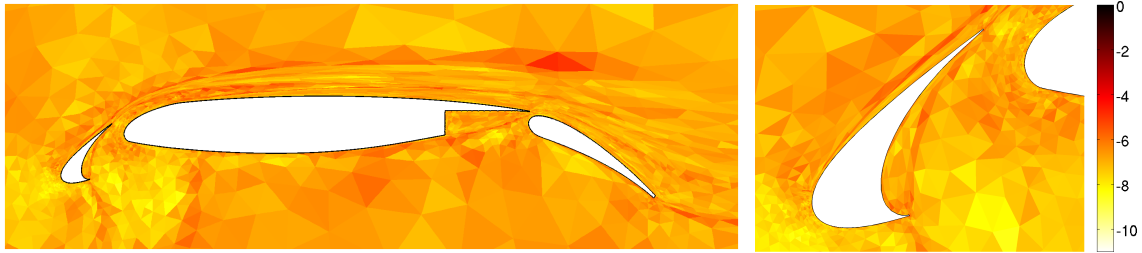
The separation for $\alpha > 20^\circ$ on the fixed mesh presents a good example of the benefit of using output-based error estimates. Figure 7-13(b) shows that the error indicator correctly identifies the lack of confidence in the solution for the high angle of attack cases on the fixed mesh. The c_l error estimate for those cases is on the order of 10. The local error estimate shown in Figure 7-14 indicates that the elements on the upper surface of the slat dominate the fixed mesh c_l error for an angles of attack of 23.28° . Figure 7-14 also shows that with adaptive refinement, the c_l error estimate is more equidistributed. With adaptive refinement, the c_l error estimates in Figure 7-13(b) remain less than one hundredth of a percent of c_l for the entire range of angles of attack considered. Utilizing the same number of degrees of freedom, the adaptation strategy is able to efficiently produce higher quality solutions compared to employing a single fixed mesh.

Figure 7-15 shows the Mach number distribution obtained for $\alpha = 23.28^\circ$ flow on the fixed and adapted meshes. As the error indicator distribution in Figure 7-14 shows, the fixed mesh lacks resolution on the front side of the leading edge slat. The lack of resolution causes extra numerical dissipation to induce separation on the upper surface of the slat. For the fixed mesh the sonic pocket is absent from the slat.

The initial mesh and the adapted meshes are shown in Figure 7-16 for a range of angles of attack. The resolution distribution within each mesh exhibits the flow features which are important for the different angles of attack. At low angles of attack, the wake coming off

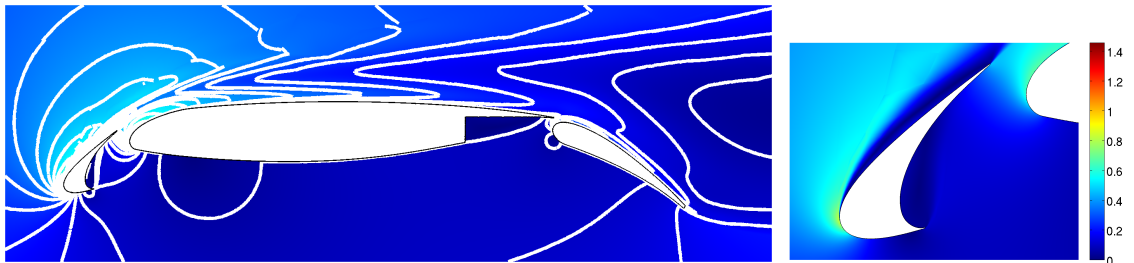


(a) fixed mesh

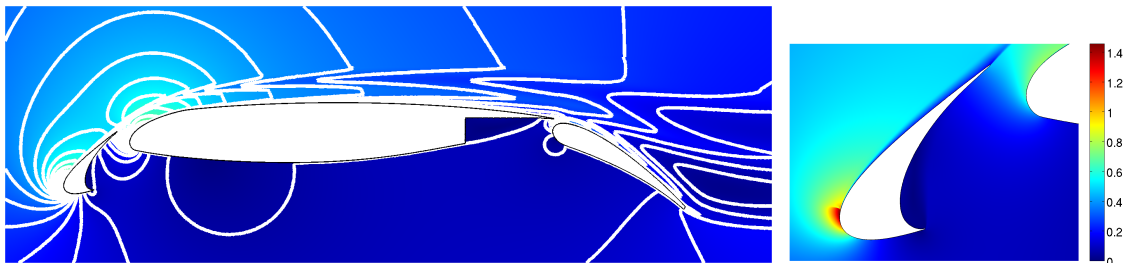


(b) adapted mesh

Figure 7-14: The error indicator distribution, $\log_{10}(\eta_\kappa)$, for the three-element MDA airfoil at $\alpha = 23.28^\circ$ obtained on the 8.10° optimized mesh and the 23.28° optimized mesh.



(a) fixed mesh



(b) adapted mesh

Figure 7-15: The Mach number distribution for the three-element MDA airfoil at $\alpha = 23.28^\circ$ obtained on the 8.10° optimized mesh and the 23.28° optimized mesh. The Mach contour lines are in 0.05 increments.

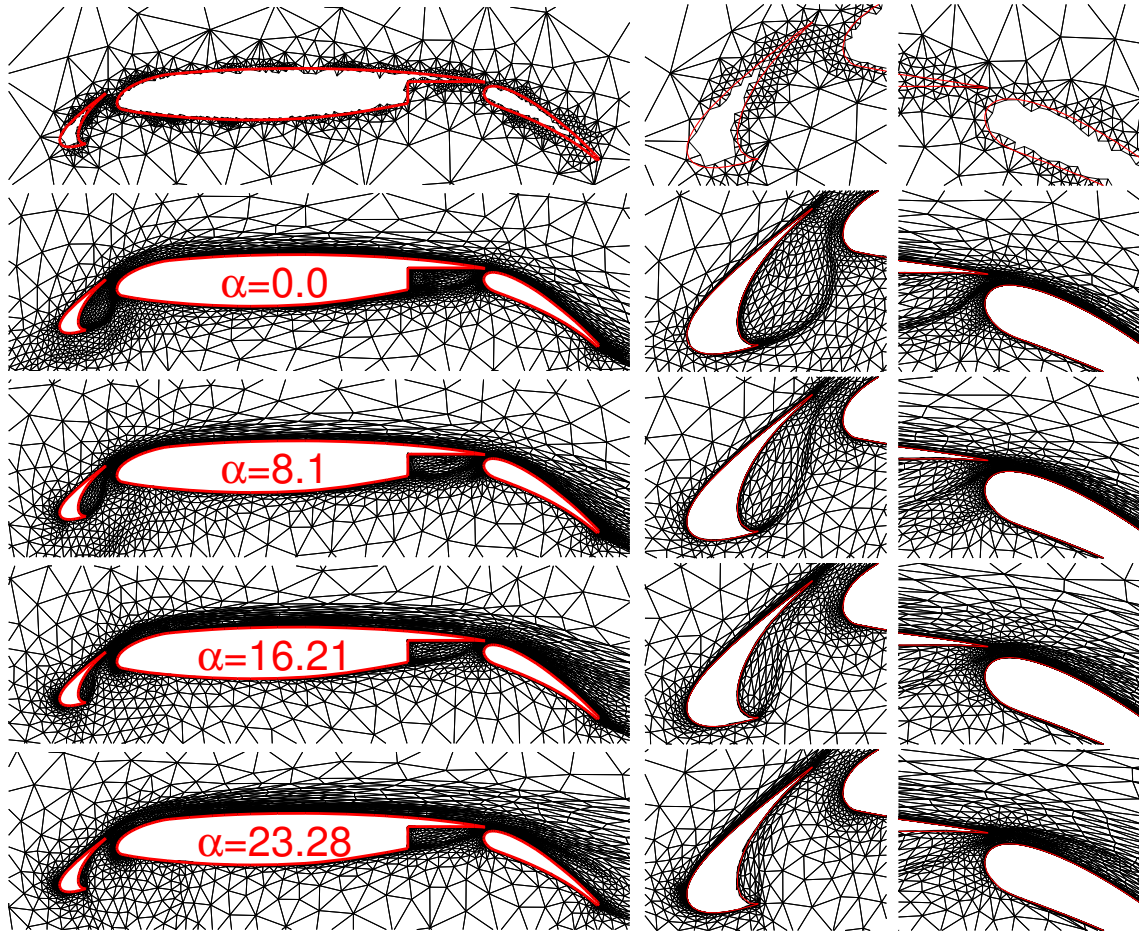


Figure 7-16: The initial and lift-adapted grids for the three-element MDA airfoil at selected angles of attack.

the bottom edge of the slat must be captured to account for its influence along the main body and flap. For higher angles of attack, this wake feature no longer exists, but capturing the acceleration and shock on the upper side of the slat becomes important for accurate lift evaluation.

7.2.2 Comparison of Boundary-Conforming and Cut-Cell

The MDA three-element airfoil provides a complex geometry which can be used to compare the solution efficiency of boundary-conforming and cut-cell meshes. For four angles of attack: $\alpha = 8.1^\circ$, 16.21° , 21.34° , and 23.28° , families of DOF-“optimal” meshes are generated for a square domain with farfield boundary conditions imposed 200 chords from the airfoil.

Figures 7-17 through 7-20 show the envelopes of drag coefficients and the error estimates for drag-based adaptation.

For the complex multi-element geometry there is a large benefit of higher-order discretizations to achieve high-fidelity output evaluations. The ability of the higher-order discretizations to convect information about the flow from the slat downstream to the flap is advantageous for accurate output evaluations. For a desired error estimate of one drag count the $p = 2$ and $p = 3$ discretizations are clear winners. Over the range of angles of attack for the MDA airfoil analyzed here, $p = 2$ is the best choice as it performs well in the low degree of freedom range while providing improved asymptotic error estimate behavior.

Once again, the cut-cell method is competitive with the boundary-conforming meshes. Due to the complex geometry, there are many flow features away from the geometry which are captured equally well by the cut-cell method. These comparison results are included for completeness, but lead to no new conclusions about the difference in solution efficiency between cut-cell and boundary-conforming meshes.

7.3 Comparison of Boundary-Conforming and Cut-Cell Solution “Cost”

Generation of the lift curve for the MDA airfoil, discussed in Section 7.2, is used to compare the solution “cost” differences between boundary-conforming and cut-cell meshes. This lift curve is computed for $M_\infty = 2.0$, $Re_c = 9 \times 10^6$ flow and the angle of attack is varied from 0.0° to 24.5° . $p = 2$ polynomials at $90k$ degrees of freedom are used for each mesh. For the comparison all computations are performed in serial on a Linux machine with an Intel i7-2600 3.40GHz processor and 12 Gbyte of RAM.

The lift curve is generated by taking the DOF-“optimal” at $\alpha = 8.1^\circ$ and $40k$ degrees of freedom and adapting at $90k$ degrees of freedom. The angle of attack is then adjusted slightly and new DOF-“optimal” meshes are generated iteratively for each angle of attack. Figure 7-21 shows the c_l error estimate convergence with adaptation iterations for each angle of attack for both cut-cell and boundary-conforming meshes. For the $\alpha = 8.1^\circ$ four adaptation iterations for the boundary-conforming meshes and five adaptation iterations for cut-cell meshes are necessary to reach the stationary point. For all the other angle of attack

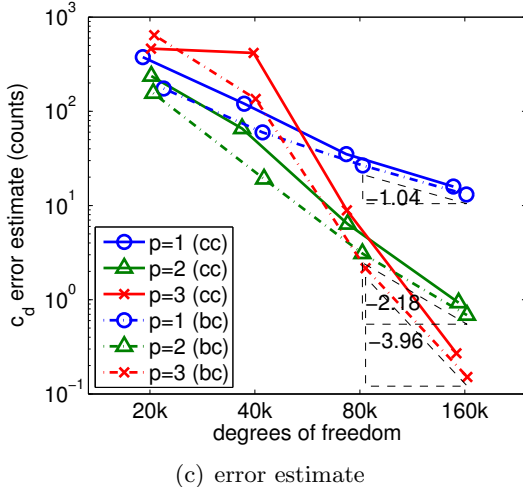
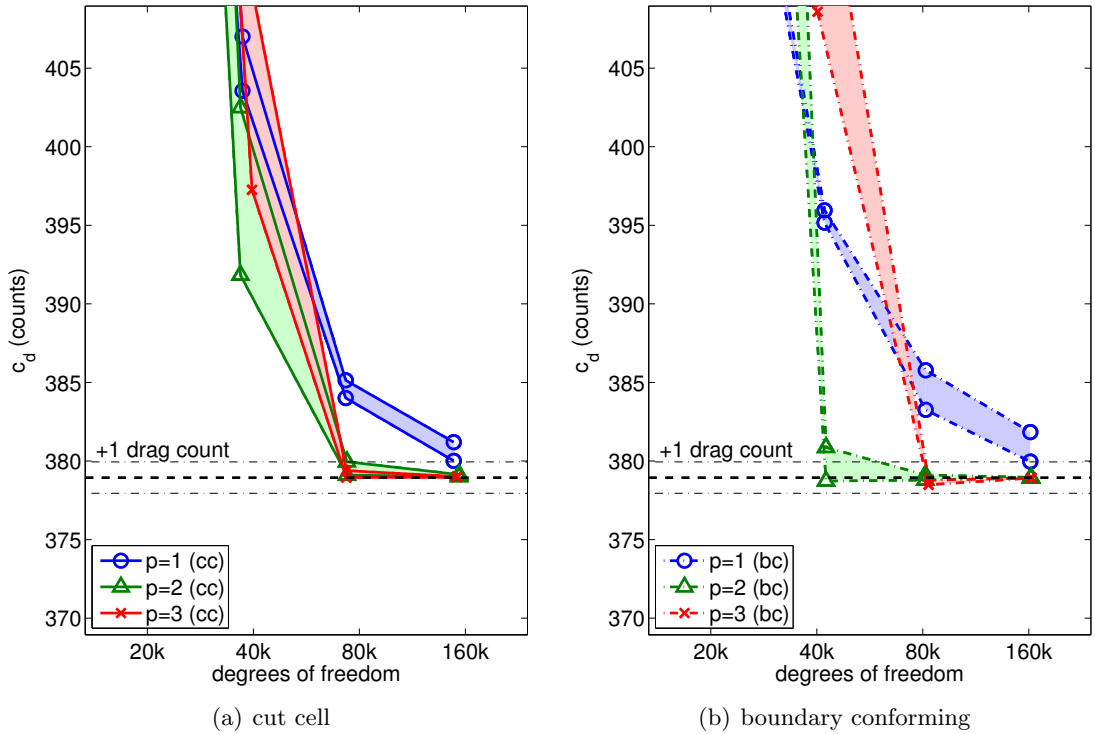


Figure 7-17: Envelopes of drag coefficients and c_d error estimates for $\alpha = 8.1^\circ$ MDA RANS-SA flow ($M_\infty = 0.2$, $Re_c = 9 \times 10^6$).

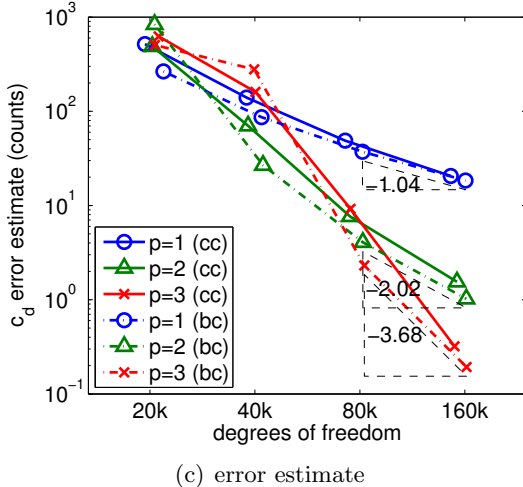
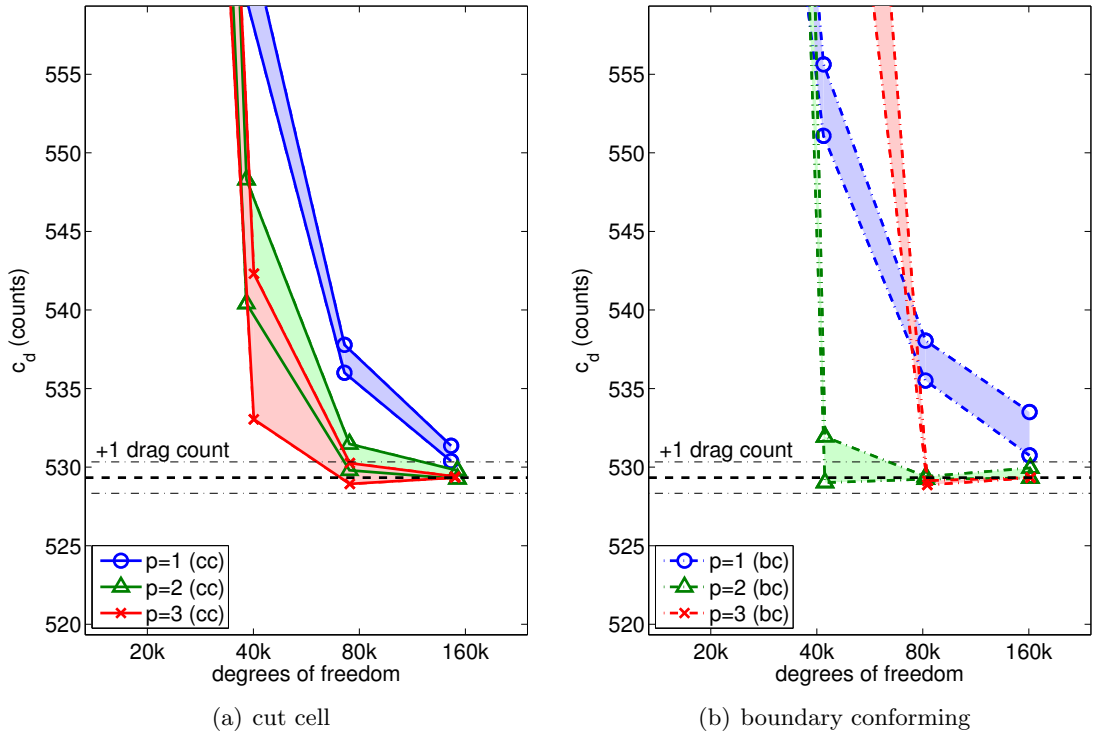


Figure 7-18: Envelopes of drag coefficients and c_d error estimates for $\alpha = 16.21^\circ$ MDA RANS-SA flow ($M_\infty = 0.2$, $Re_c = 9 \times 10^6$).

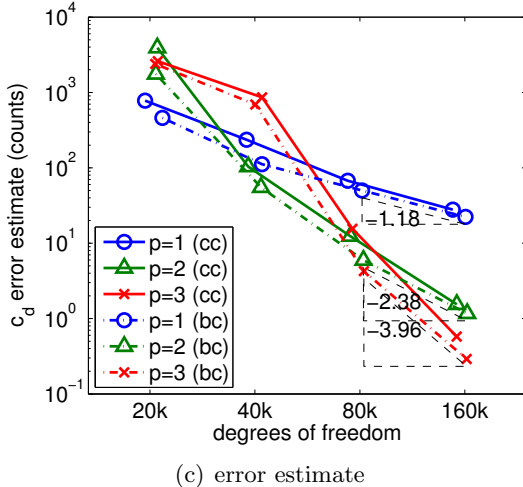
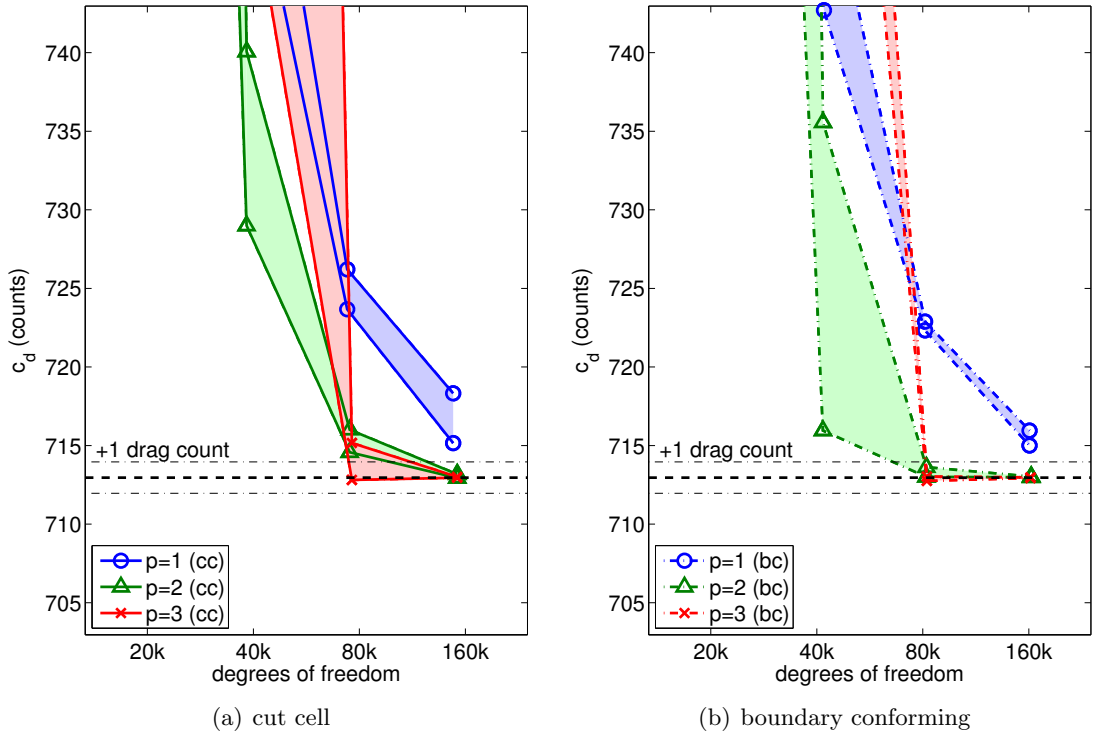


Figure 7-19: Envelopes of drag coefficients and c_d error estimates for $\alpha = 21.34^\circ$ MDA RANS-SA flow ($M_\infty = 0.2$, $Re_c = 9 \times 10^6$).

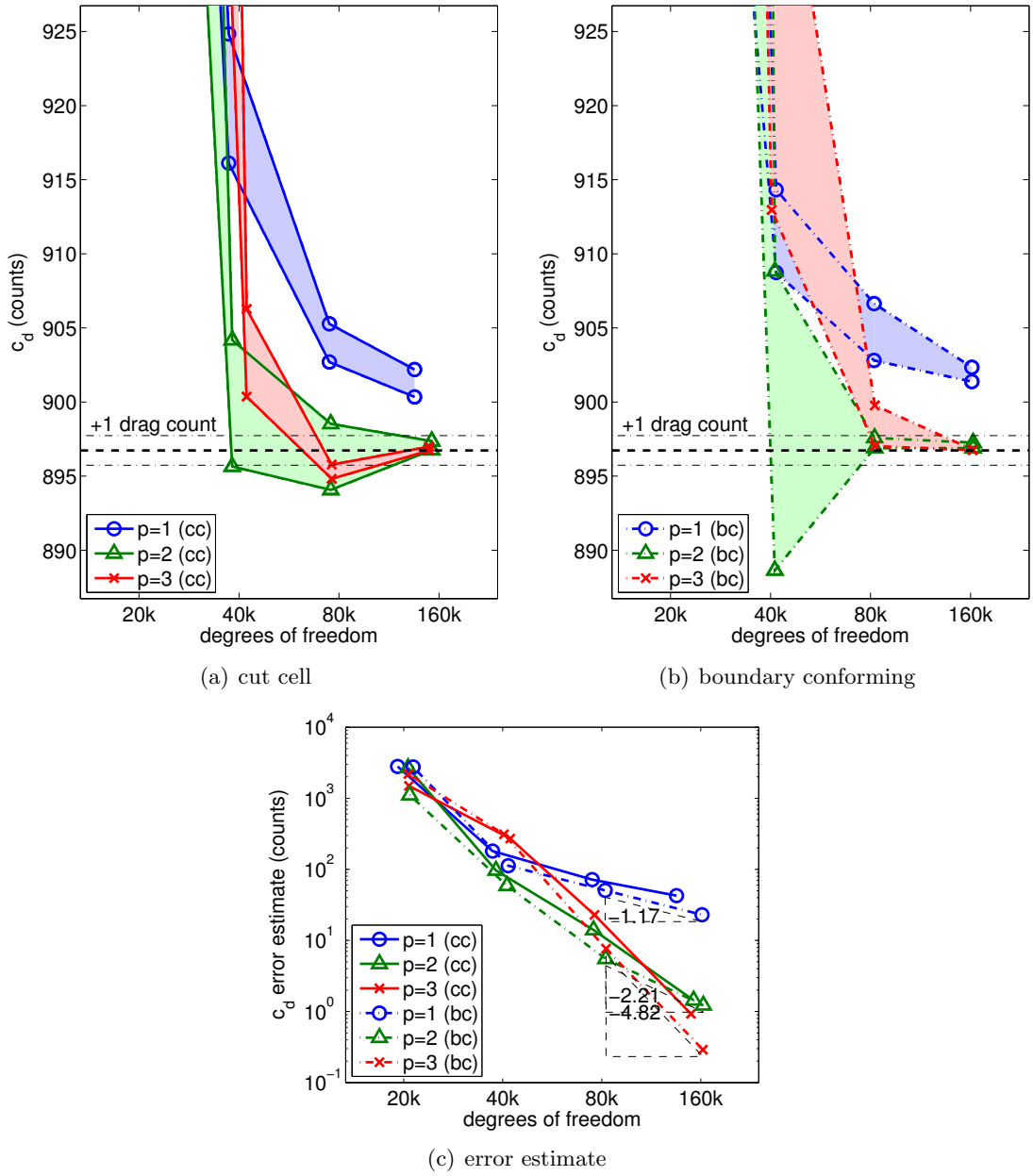


Figure 7-20: Envelopes of drag coefficients and c_d error estimates for $\alpha = 23.28^\circ$ MDA RANS-SA flow ($M_\infty = 0.2$, $Re_c = 9 \times 10^6$).

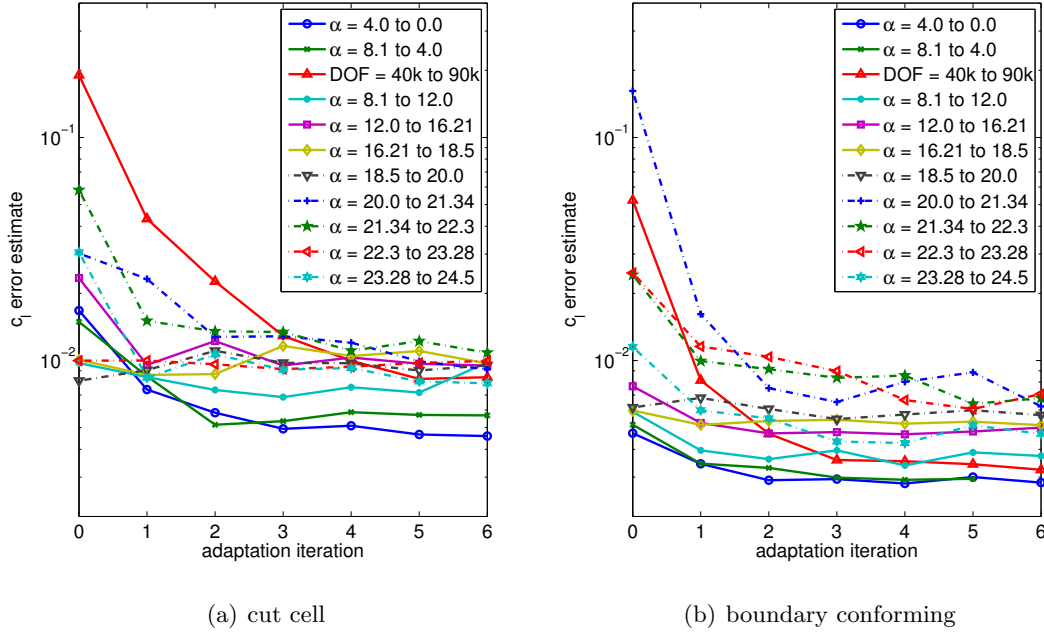


Figure 7-21: c_l error estimate convergence with adaptation iteration during generation of lift curve for MDA RANS-SA flow ($M_\infty = 0.2$, $Re_c = 9 \times 10^6$) with boundary-conforming and cut-cell meshes.

variations only two adaptation iterations (or three non-linear solves) are required to reach the stationary point.

Table 7.1 includes a summary of the solution “cost” measured by the number of adaptation iterations, the wall-clock time, the number of non-linear iterations, and the number of GMRES iterations. In general the cut-cell method is 10% slower than the boundary-conforming method. For the lift curve, the cut-cell meshes actually require fewer non-linear iterations than the boundary-conforming meshes. There is no reason to believe that the non-linear solver performs better on the cut-cell meshes. The small variation in the non-linear iteration count is likely due to variations in the meshes and the quality of the solution transfer between the adapted meshes. Though the cut-cell meshes take fewer non-linear iterations, they require more GMRES iterations to solve the linear systems. The difference in the GMRES iteration count between the cut-cell and boundary-conforming meshes is expected. Merging is used to remove all small volume ratios less than 10^{-5} . However, the cut-cell meshes still contain smaller volume ratios than the boundary-conforming meshes

(a) Cut cell

α	Adapt iter.	Time (hr)	Non-linear iter.	GMRES iter.
$\alpha = 0.0$	2	8.85	540	20639
$\alpha = 4.0$	2	7.29	478	15379
$\alpha = 8.1$	5	6.55	392	20579
$\alpha = 12.0$	2	8.75	587	14890
$\alpha = 16.21$	2	8.77	569	15333
$\alpha = 18.5$	2	6.13	401	11434
$\alpha = 20.0$	2	5.38	343	10459
$\alpha = 21.34$	2	6.17	403	14843
$\alpha = 22.3$	2	6.33	432	13767
$\alpha = 23.28$	2	5.67	409	13687
$\alpha = 24.5$	2	8.14	584	14441
Total	25	78.05	5138	165451

(b) Boundary conforming

α	Adapt iter.	Time (hr)	Non-linear iter.	GMRES iter.
$\alpha = 0.0$	2	7.48	550	18065
$\alpha = 4.0$	2	6.80	484	16155
$\alpha = 8.1$	4	4.80	290	15393
$\alpha = 12.0$	2	7.48	581	14688
$\alpha = 16.21$	2	6.87	540	13473
$\alpha = 18.5$	2	5.78	433	11882
$\alpha = 20.0$	2	5.58	416	11913
$\alpha = 21.34$	2	6.58	469	17372
$\alpha = 22.3$	2	5.58	416	12423
$\alpha = 23.28$	2	6.52	492	14784
$\alpha = 24.5$	2	8.18	691	15602
Total	24	71.65	5362	161750

Table 7.1: Summary of “cost” to generate the lift curve for the MDA airfoil using boundary-conforming and cut-cell meshes.

which impacts the condition of the linear system and leads to the increase in the number of GMRES iterations.

Wall clock time is included as a measure of solution “cost” since it is a very important property of a solution procedure, however, the cut-cell solution strategy used here has not been developed with computational efficiency as a primary objective. In the cut-cell implementation flexibility was always selected over computational performance. On the other hand, the boundary-conforming implementation has undergone some optimization to improve its performance [79].

In terms of system memory usage the existing implementation of the cut-cell method uses essentially the same amount as the boundary-conforming method. Before the recognition of canonical elements, the quadrature rules for each cut element was stored. For large cases the storage of the quadrature rules was a large memory demand, particularly in the extension of cut cells to three dimensions. With the recognition of canonical shape the cut-cell method only makes a small perturbation to the memory usage of boundary-conforming meshes due to the low number of arbitrarily cut element.

Chapter 8

Conclusions

8.1 Summary and Contributions

This thesis presents a higher-order, adaptive, simplex cut-cell solution strategy for the two-dimensional RANS-SA equations. The main contributions of this work are an analysis of the impact of small volume ratios from arbitrarily cut elements on linear system conditioning and solution quality, a development of a line-search globalization technique based on the unsteady residual of a pseudo-transient evolution to improve the robustness of non-linear solvers, a development of an adaptation strategy that is insensitive to solution regularity and poor error estimates on under-resolved meshes, and a quantification of the impact on solution efficiency in the transition from boundary-conforming to cut-cell meshes.

The small volume ratio analysis concludes that for elliptic problems the condition number of the linear system resulting from a continuous Galerkin discretization scales as $\mathcal{O}(\frac{1}{h^2\text{VR}})$. The additional VR dependence of the condition number is solely caused by the discretization space and is independent of the variational statement. The impact of small volume ratios is illustrated for scalar model problems. The numerical experiments for continuous and discontinuous Galerkin finite element methods agree with the analysis. The merging method is introduced in order to remove the impact of small volume ratios by modifying the discretization space. For the model problem merging eliminates the impact of small volume ratios on the linear system condition number and the solution quality.

The addition of line-search update limiting to the pseudo-time continuation based non-linear solver has been enabling for this research. In previous work, Oliver developed an

unsteady adaptation procedure where the need to converge the steady state solution prior to adaptation was eliminated in order to counter difficulties in solving the RANS-SA equations on under-resolved meshes [94]. The unsteady adaptation strategy was necessary to improve the robustness of the solution procedure for a subsonic high-lift three-element airfoil [98]. The results of this work demonstrate that the improved reliability from line-search update limiting of the non-linear solver allows for the computation of steady-state solutions to the RANS-SA equations on under-resolved meshes. For the comparison between boundary-conforming and cut-cell meshes, over two thousand three-element airfoil cases were successfully converged to steady state.

An adaptation strategy is presented that iterates toward DOF-“optimal” simplex meshes where the local elemental error estimates are equidistributed. The adaptation strategy employs an output-based error estimate and explicit control of degrees of freedom. The strategy contains no assumed rates of convergence for element sizing and is, therefore, independent of solution regularity and low quality error estimates from under-resolved meshes during the early cycles of adaptation. With the adaptation strategy, DOF-“optimal” meshes are generated that can realize the benefit of higher-order discretizations at low degrees of freedom. The key features of these meshes are strong grading toward singularities and arbitrarily oriented anisotropic resolutions for boundary layers, wakes, and shocks.

The solution efficiency of cut-cell meshes is compared to boundary-conforming meshes over a range of airfoil geometries and RANS-SA flows ranging from subsonic to supersonic conditions. Cut-cell meshes achieve the same asymptotic rate of convergence in the output error estimates as boundary-conforming meshes. For a subsonic RAE2822 RANS-SA flow, which is dominated by singular perturbations on the geometry, a loss of solution efficiency is present for the cut-cell method. The geometry representation using cut linear background elements limits the cut-cell method as it is unable to effectively resolve higher-order geometry with a small number of elements. As the total number of degrees of freedom are increased, the solution efficiency gap decreases such that for high-fidelity simulations cut-cell meshes are competitive with boundary-conforming meshes even for simple cases with singular perturbations on the geometry. For more complex cases, whether in terms of geometry or flow features like shocks, the cut-cell and boundary-conforming solution efficiencies are nearly equivalent. The cut-cell method is equally capable of resolving wakes and shocks as

the boundary-conforming meshes. Based on the two-dimensional results, three-dimensional cut cells are a realistic mesh generation strategy for high-fidelity simulations since cut-cell meshes give comparable solution efficiency to boundary conforming meshes for RANS-SA flows and greatly reduce the burden of mesh generation.

The results illustrate that, with suitable mesh generation, higher-order methods are superior to lower-order methods for RANS-SA simulations of subsonic, transonic, and supersonic flows. Additionally, the advantage of a higher-order discretization can be achieved at error estimate levels as high as 10 drag counts. The combination of higher-order methods, a robust non-linear solver, an output-based adaptation algorithm, and simplex cut-cell mesh generation provides a solution strategy that is one step closer to making fully automated CFD a reality.

8.2 Future Work

Extension of simplex cut-cells to RANS simulations in three dimensions

The two-dimensional RANS-SA cut-cell results in this work motivate the extension of the cut-cell method to three-dimensional viscous problems. Fidkowski previously applied simplex cut cells to three-dimensional Euler flows but had concerns that a cut-cell technique based on linear background meshes would be too inefficient for boundary layer resolution [43]. The two-dimensional evidence provided here shows that linear based cut cells are competitive with boundary-conforming meshes. The cut-cell method in three dimensions simplifies mesh generation and allows for fully automated simulations of complex geometries.

Further application of two-dimensional cut cells

In this work, the simplex cut-cell method is applied to external aerodynamic flows. The cut-cell method for the RANS equations can also be applied to internal flows. One application of particular interest is a fully-coupled fluid-structures heat transfer problems where the cut geometry acts as the interface between flow equations on one side and the heat equation for the structures. Adaptation is performed on the cut-cell background mesh which doesn't respect geometry boundaries to capture important features of both systems of governing equations.

Automation in adaptation strategy

Currently, adaptation strategy iterates toward DOF-“optimal” meshes which equidistribute elemental error estimates at a fixed degree of freedom. The strategy is geared toward the specific purpose of generating the best mesh while maintaining a specific degree of freedom. The adaptation strategy is well suited for academic exercises such as the comparison performed between cut-cell and boundary-conforming meshes. However, in its current form, the adaptation strategy is limited for practical engineering problems to parameters sweeps at a user-fixed degree of freedom. To make CFD a fully automated tool for engineering, the necessity exists for the degrees of freedom to be controlled by the adaptation algorithm instead of by the user. The key feature to increase the autonomy of the adaptation algorithm is the ability to access the optimality status of the current mesh. Given the optimality status and the requested error tolerance the decision can be made to either increase, decrease, or maintain the degrees of freedom. If the error is above the requested tolerance and the mesh is optimal then the degrees of freedom should be increased. On the other hand, if the local elemental error estimates are not equidistributed, the degrees of freedom should remain constant or decrease to allow for the redistribution of element sizes at a lower cost.

Coupling primal and dual solutions to anisotropy detection

The DOF-“optimal” mesh generated in this work did not necessarily minimize the output error for a given degree of freedom. Equidistribution of the local errors is a necessary condition, but not a sufficient condition for true output error minimization. In the existing adaptation strategy, element size distribution is optimized, but the element shapes may not be. Element shapes from the anisotropy detection algorithm are based on $p + 1$ derivatives of the Mach number. In cases where the Mach number does not represent all the anisotropic features in the flow, the element shapes are not optimal. Output errors are bounded by the error in the primal and dual solutions. So, to generate optimal meshes at a given degree of freedom, element anisotropy must capture features in both the primal and dual solutions. For example, with flow over an airfoil and drag as the desired output, the adjoint solution varies rapidly across the stagnation streamline. Mach number based anisotropy does not capture this flow feature efficiently. One possible option for coupling anisotropy detection to the primal and dual solution is to cast the adaptation algorithm in an optimization framework

and take adaptation steps in the direction of decreasing the output error as described by Yano and Darmofal [131]

Bibliography

- [1] “3rd AIAA CFD Drag Prediction Workshop.” San Francisco, CA. <http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop3>.
- [2] Aftosmis, M. J. “Solution adaptive Cartesian grid methods for aerodynamic flows with complex geometries.” In *von Karman Institute for Fluid Dynamics, Lecture Series 1997-02*. Rhode-Saint-Genése, Belgium, Mar. 3-7, 1997.
- [3] Aftosmis, M. J., Berger, M. J., and Melton, J. M. “Adaptive Cartesian mesh generation.” In Thompson, J. F., Soni, B. K., and Weatherill, N. P., editors, *Handbook of Grid Generation*. CRC Press, 1998.
- [4] Allmaras, S. R. *A coupled Euler/Navier-Stokes algorithm for 2-D unsteady transonic shock/boundary-layer interaction*. PhD thesis, Massachusetts Institute of Technology, 1989.
- [5] Allmaras, S. R. and Giles, M. B. “A second-order flux split scheme for the unsteady 2-D Euler equations on arbitrary meshes.” AIAA 1987-1119-CP, 1987.
- [6] Armijo, L. “Minimization of functions having Lipschitz continuous first partial derivatives.” *Pacific Journal of Mathematics*, 16(1):1–2, 1966.
- [7] Arnold, D. N. “An interior penalty finite element method with discontinuous elements.” *SIAM Journal on Numerical Analysis*, 19:742–760, 1982.
- [8] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D. “Unified analysis of discontinuous Galerkin methods for elliptical problems.” *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002.
- [9] Babuska, I., Szabo, B. A., and Katz, I. N. “The p -version of the finite element method.” *SIAM Journal on Numerical Analysis*, 18(3):515–545, 1981.
- [10] Backman, G. and Narici, L. *Functional Analysis*. Academic Press, New York and London, 1966.
- [11] Barter, G. E. *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.

- [12] Barter, G. E. and Darmofal, D. L. “Shock capturing with PDE-based artificial viscosity for DGFEM: Part I, Formulation.” *Journal of Computational Physics*, 229(5):1810–1827, 2010.
- [13] Barth, T. J. and Larson, M. G. “A posteriori error estimates for higher order Godunov finite volume methods on unstructured meshes.” In Herban, R. and Kröner, D., editors, *Finite Volumes for Complex Applications III*, London, 2002. Hermes Penton.
- [14] Bassi, F., Crivellini, A., Rebay, S., and Savini, M. “Discontinuous Galerkin solution of the Reynolds averaged Navier-Stokes and $k\text{-}\omega$ turbulence model equations.” *Computers & Fluids*, 34:507–540, May-June 2005.
- [15] Bassi, F. and Rebay, S. “A High-order discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations.” *Journal of Computational Physics*, 131:267–279, 1997.
- [16] Bassi, F. and Rebay, S. “GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations.” In Cockburn, K. and Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, pages 197–208. Springer, Berlin, 2000.
- [17] Bassi, F. and Rebay, S. “Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations.” *International Journal for Numerical Methods in Fluids*, 40:197–207, 2002.
- [18] Bassi, F., Crivellini, A., Ghidoni, A., and Rebay, S. “High-order discontinuous Galerkin discretization of transonic turbulent flows.” AIAA 2009-180, 2009.
- [19] Becker, R. and Rannacher, R. “A feed-back approach to error control in finite element methods: Basic analysis and examples.” *East-West Journal of Numerical Mathematics*, 4:237–264, 1996.
- [20] Becker, R. and Rannacher, R. “An optimal control approach to a posteriori error estimation in finite element methods.” In Iserles, A., editor, *Acta Numerica*. Cambridge University Press, 2001.
- [21] Borouchaki, H., George, P. L., Hecht, F., Laug, P., and Saltel, E. “Delaunay mesh generation governed by metric specifications. Part I algorithms.” *Finite Elements in Analysis and Design*, 25(1-2):61–83, 1997.
- [22] Bossen, F. J. and Heckbert, P. S. “A Pliant Method for Anisotropic Mesh Generation.” In *5th Intl. Meshing Roundtable*, pages 63–74, Oct. 1996.
- [23] Brenner, S. C. and Scott, L. R. *The Mathematical Theory of Finite Element Methods, Third Edition*. Springer, New York, 2008.
- [24] Burgess, N., Nastase, C., and Mavriplis, D. “Efficient solution techniques for discontinuous Galerkin discretizations of the Navier-Stokes equations on hybrid anisotropic meshes.” AIAA 2010-1448, 2010.
- [25] Celic, A. and Hirschel, E. “Comparison of eddy-viscosity turbulence models in flows with adverse pressure gradient.” *AIAA Journal*, 44(10), 2006.

- [26] Ceze, M. and Fidkowski, K. J. “Output-driven anisotropic mesh adaptation for viscous flows using discrete choice optimization.” *AIAA* 2010–170, 2010.
- [27] Chavent, G. and Salzano, G. “A finite element method for the 1D water flooding problem with gravity.” *Journal of Computational Physics*, 42:307–344, 1982.
- [28] Clarke, D. K., Salas, M. D., and Hassan, H. A. “Euler calculations for multielement airfoils using Cartesian grids.” *AIAA Journal*, 24(3):353–358, 1986.
- [29] Cockburn, B., Karniadakis, G., and Shu, C. “The development of discontinuous Galerkin methods.” In *Lecture Notes in Computational Science and Engineering*, volume 11. Springer, 2000.
- [30] Cockburn, B., Lin, S. Y., and Shu, C. W. “TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems.” *Journal of Computational Physics*, 84:90–113, 1989.
- [31] Cockburn, B. and Shu, C. W. “TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework.” *Mathematics of Computation*, 52:411–435, 1989.
- [32] Cockburn, B. and Shu, C. W. “The local discontinuous Galerkin method for time-dependent convection-diffusion systems.” *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, December 1998.
- [33] Cockburn, B. and Shu, C. W. “The Runge-Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems.” *Journal of Computational Physics*, 141:199–224, 1998.
- [34] Cockburn, B. and Shu, C.-W. “Runge-Kutta discontinuous Galerkin methods for convection-dominated problems.” *Journal of Scientific Computing*, pages 173–261, 2001.
- [35] Dahmen, W., Faermann, B., Graham, I. G., Hackbusch, W., and Sauter, S. A. “Inverse Inequalities on Non-Quasiuniform Meshes and Application to the Mortar Element Method.” *Mathematics of Computation*, 73(247):1107–1138, 2003.
- [36] De Rango, S. and Zingg, D. W. “High-order spatial discretization for turbulent aerodynamic computations.” *AIAA Journal*, 39(7), 2001.
- [37] Dennis, J. E. and Schnabel, R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Inc., New Jersey, 1983.
- [38] Diosady, L. T. “A linear multigrid preconditioner for the solution of the Navier-Stokes equations using a discontinuous Galerkin discretization.” Masters thesis, Mass. Inst. of Tech., CDO, May 2007.
- [39] Diosady, L. T. and Darmofal, D. L. “Preconditioning methods for discontinuous Galerkin solutions of the Navier-Stokes equations.” *Journal of Computational Physics*, pages 3917–3935, 2009.
- [40] Drela, M. Personal Communication via email, November 2010.

- [41] Farthing, M. W., Kees, C. E., Coffey, T. S., Kelley, C. T., and Miller, C. T. “Efficient steady-state solution techniques for variably saturated groundwater flow.” *Advances in Water Resources*, 26(8):833 – 849, 2003.
- [42] Fidkowski, K. J. “A High-Order Discontinuous Galerkin Multigrid Solver for Aerodynamic Applications.” Masters thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2004.
- [43] Fidkowski, K. J. *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.
- [44] Fidkowski, K. J. and Darmofal, D. L. “An adaptive simplex cut-cell method for discontinuous Galerkin discretizations of the Navier-Stokes equations.” AIAA 2007-3941, 2007.
- [45] Fidkowski, K. J. and Darmofal, D. L. “A triangular cut-cell adaptive method for higher-order discretizations of the compressible Navier-Stokes equations.” *Journal of Computational Physics*, 225:1653–1672, 2007.
- [46] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L. “ p -Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations.” *Journal of Computational Physics*, 207(1):92–113, 2005.
- [47] Fried, I. “Bounds on the extremal eigenvalues of the finite element stiffness and mass matrices and their spectral condition number.” *Journal of Sound and Vibration*, 22(4):407–418, 1972.
- [48] Fried, I. “The l_2 and l_∞ condition numbers of the finite element stiffness and mass matrices, and the pointwise convergence of the method.” *The mathematics of finite elements and applications; Proceedings of the Brunel University conference of the Institute of Mathematics and Its Applications*, pages 163–174, 1973.
- [49] Gaffney, R. L., Hassan, H. A., and Salas, M. D. “Euler calculations for wings using Cartesian grids.” AIAA 87-0356, jan 1987.
- [50] Georgoulis, E. H., Hall, E., and Houston, P. “Discontinuous Galerkin methods on hp -anisotropic meshes I: A priori error analysis.” *International Journal of Computing Science and Mathematics*, 1(2–4), 2007.
- [51] Georgoulis, E. H., Hall, E., and Houston, P. “Discontinuous Galerkin methods on hp -anisotropic meshes II: A posteriori error analysis and adaptivity.” *Applied Numerical Mathematics*, 59:2179–2194, 2009.
- [52] Giles, M. and Pierce, N. “Adjoint error correction for integral outputs.” In Barth, T., Griebel, M., Keyes, D. E., Nieminen, R. M., Roose, D., and Schlick, T., editors, *Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, volume 25. Springer, Berlin, 2002.

- [53] Giles, M. B. and Süli, E. “Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality.” In *Acta Numerica*, volume 11, pages 145–236, 2002. doi: 10.1017/S096249290200003X.
- [54] Giles, M., Pierce, N., and Suli, E. “Progress in adjoint error correction for integral functionals.” *Computing and Visualisation in Science*, 6(2-3):113–121, 2004.
- [55] Godin, P., Zingg, D., and Nelson, T. “High-lift aerodynamic computations with one- and two-equation turbulence models.” *AIAA Journal*, 35(2), 1997.
- [56] Gropp, W., Keyes, D., McInnes, L. C., and Tidriri, M. D. “Globalized Newton-Krylov-Schwarz algorithms and software for parallel implicit CFD.” *International Journal of High Performance Computing Applications*, 14(2):102–136, 2000.
- [57] Guo, B. and Babuka, I. “The $h - p$ version of the finite element method, Part 1: The basic approximation results.” *Computational Mechanics*, 1:21–41, 1986.
- [58] Guo, B. and Babuka, I. “The $h - p$ version of the finite element method, Part 2: General results and applications.” *Computational Mechanics*, 1:203–220, 1986.
- [59] Hartmann, R. “Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations.” *International Journal for Numerical Methods in Fluids*, 51:1131–1156, 2006.
- [60] Hartmann, R. “Adjoint Consistency Analysis of Discontinuous Galerkin Discretizations.” *SIAM Journal on Numerical Analysis*, 45(6):2671–2696, 2007.
- [61] Hartmann, R. and Houston, P. “Goal-oriented a posteriori error estimation for multiple target functionals.” In Hou, T. and Tadmor, E., editors, *Hyperbolic Problems: Theory, Numerics, Applications*, pages 579–588. Springer-Verlag, 2003.
- [62] Hartmann, R. and Houston, P. “Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations.” *Journal of Computational Physics*, 183(2): 508–532, 2002.
- [63] Hartmann, R. and Houston, P. “Error estimation and adaptive mesh refinement for aerodynamic flows.” In Deconinck, H., editor, *VKI LS 2010-01: 36th CFD/ADIGMA course on hp-adaptive and hp-multigrid methods, Oct. 26-30, 2009*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2009.
- [64] Hecht, F. “BAMG: Bidimensional Anisotropic Mesh Generator.” 1998. <http://www-rocq1.inria.fr/gamma/cdrom/www/bamg/eng.htm>.
- [65] Houston, P. and Süli, E. “ hp -adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems.” *SIAM Journal on Scientific Computing*, 23(4): 1226–1252, 2001.
- [66] Houston, P., Georgoulis, E. H., and Hall, E. “Adaptivity and a posteriori error estimation for DG methods on anisotropic meshes.” *International Conference on Boundary and Interior Layers*, 2006.

- [67] Houston, P. and Süli, E. “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.
- [68] Karman, S. L. “SPLITFLOW: A 3D unstructured Cartesian/prismatic grid CFD code for complex geometries.” AIAA 1995-0343, 1995.
- [69] Kelley, C. T. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, 1995.
- [70] Kelley, C. T. and Keyes, D. E. “Convergence analysis of pseudo-transient continuation.” *SIAM Journal on Numerical Analysis*, 35(2):508–523, 1998.
- [71] Kirby, R. C. “From functional analysis to iterative methods.” *SIAM Review*, 42(2): 269–293, 2010.
- [72] Klausmeyer, S. M. and Lin, J. C. “Comparative results from a CFD challenge over a 2D three-element high-lift airfoil.” NASA Technical Memorandum 112858, 1997.
- [73] Knoll, D. A. and McHugh, P. R. “Enhanced nonlinear iterative techniques applied to a nonequilibrium plasma flow.” *SIAM Journal on Scientific Computing*, 19(1):291–301, 1998.
- [74] Knoll, D. A., Prinja, A. K., and Campbell, R. B. “A direct Newton solver for the two-dimensional Tokamak edge plasma fluid equations.” *Journal of Computational Physics*, 104(2):418–426, 1993.
- [75] Lahur, P. R. and Nakamura, Y. “A new method for thin body problem in Cartesian grid generation.” AIAA 99-0919, 1999.
- [76] Lahur, P. R. and Nakamura, Y. “Anisotropic Cartesian grid adaptation.” AIAA 2000-2243, 2000.
- [77] Landmann, B., Kessler, M., Wagner, S., and Krämer, E. “A parallel, high-order discontinuous Galerkin code for laminar and turbulent flows.” *Computers & Fluids*, 37 (4):427 – 438, 2008.
- [78] Leicht, T. and Hartmann, R. “Anisotropic mesh refinement for discontinuous Galerkin methods in two-dimensional aerodynamic flow simulations.” *International Journal for Numerical Methods in Fluids*, 56:2111–2138, 2008.
- [79] Liu, E. H.-L. “Optimization and validation of discontinuous Galerkin code for the 3D Navier-Stokes equations.” Masters thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, January 2011.
- [80] Loseille, A. and Alauzet, F. “Continuous mesh model and well-posed continuous interpolation error estimation.” INRIA RR-6846, 2009.
- [81] Loseille, A. and Löhner, R. “On 3D Anisotropic Local Remeshing for Surface, Volume and Boundary Layers.” In *Proceedings of the 18th International Meshing Roundtable*, pages 611–630. Springer Berlin Heidelberg, 2009.

- [82] Lu, J. *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.
- [83] Mavriplis, D. J. “Results from the 3rd Drag Prediction Workshop using the NSU3D unstructured mesh solver.” AIAA 2007-256, 2007.
- [84] Mavriplis, D. J. and Venkatakrishnan, V. “A 3D agglomeration multigrid solver for the Reynolds-averaged Navier-Stokes equations on unstructured meshes.” *International Journal for Numerical Methods in Fluids*, 23(6):527–544, 1996.
- [85] Mavriplis, D. J. and Venkatakrishnan, V. “A unified multigrid solver for the Navier-Stokes equations on mixed element meshes.” *International Journal of Computational Fluid Dynamics*, 8(4):247–263, 1997.
- [86] Mavriplis, D. J., Darmofal, D., Keyes, D., and Turner, M. “Petaflops opportunities for the NASA fundamental aeronautics program.” AIAA 2007-4084, 2007.
- [87] Mavriplis, D. J. and Pirzadeh, S. “Large-scale parallel unstructured mesh computations for 3D high-lift analysis.” *AIAA J. Aircraft*, 36:987–998, 1999.
- [88] Modisette, J. M. and Darmofal, D. L. “An output-based adaptive and higher-order method for a rotor in hover.” AIAA 2008-7343, 2008.
- [89] Nemec, M., Aftosmis, M., Murman, S., and Pulliam, T. “Adjoint formulation for an embedded-boundary Cartesian method.” AIAA 2005-0877, 2005.
- [90] Nemec, M. and Aftosmis, M. J. “Adjoint error estimation and adaptive refinement for embedded-boundary Cartesian meshes.” AIAA 2007-4187, 2007.
- [91] Nemec, M., Aftosmis, M. J., and Wintzer, M. “Adjoint-based adaptive mesh refinement for complex geometries.” AIAA 2008-725, 2008.
- [92] Nguyen, N., Persson, P.-O., and Peraire, J. “RANS solutions using high order discontinuous Galerkin methods.” AIAA 2007-0914, 2007.
- [93] Nielsen, E. J. and Anderson, W. K. “Recent improvements in aerodynamic design optimization on unstructured meshes.” *AIAA Journal*, 40(6):1155–1163, 2002. doi: 10.2514/2.1765.
- [94] Oliver, T. and Darmofal, D. “An unsteady adaptation algorithm for discontinuous Galerkin discretizations of the RANS equations.” AIAA 2007-3940, 2007.
- [95] Oliver, T. and Darmofal, D. “Analysis of dual consistency for discontinuous Galerkin discretizations of source terms.” *SIAM Journal on Numerical Analysis*, 47(5):3507–3525, 2009.
- [96] Oliver, T. and Darmofal, D. “Impact of turbulence model irregularity on high-order discretizations.” AIAA 2009-953, 2009.
- [97] Oliver, T. A. “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.

- [98] Oliver, T. A. *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.
- [99] Oliver, T. A. and Darmofal, D. L. “An analysis of dual consistency for discontinuous Galerkin discretization of source terms.” ACDL report, Massachusetts Institute of Technology, 2007.
- [100] Ollivier-Gooch, C. and Altena, M. V. “A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation.” *Journal of Computational Physics*, 181(2):729–752, 2002.
- [101] Park, M. A. *Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2008.
- [102] Park, M. A. “Low boom configuration analysis with FUN3D adjoint simulation framework.” AIAA 2011-3337, 2011.
- [103] Park, M. A. and Carlson, J.-R. “Turbulent output-based anisotropic adaptation.” AIAA 2010-168, 2010.
- [104] Peraire, J. and Persson, P.-O. “The compact discontinuous Galerkin (CDG) method for elliptic problems.” *SIAM Journal on Scientific Computing*, 30(4):1806–1824, 2008.
- [105] Persson, P.-O. and Peraire, J. “Sub-cell shock capturing for discontinuous Galerkin methods.” AIAA 2006-0112, 2006.
- [106] Persson, P.-O. and Peraire, J. “Newton-GMRES preconditioning for Discontinuous Galerkin discretizations of the Navier-Stokes equations.” *SIAM Journal on Scientific Computing*, 30(6):2709–2722, 2008.
- [107] Persson, P.-O. and Peraire, J. “Curved mesh generation and mesh refinement using Lagrangian solid mechanics.” AIAA 2009-0949, 2009.
- [108] Pierce, N. A. and Giles, M. B. “Adjoint recovery of superconvergent functionals from PDE approximations.” *SIAM Review*, 42(2):247–264, 2000.
- [109] Pirzadeh, S. “Three-dimensional unstructured viscous grids by the advancing-layers method.” *AIAA Journal*, 34(1):43–49, 1996.
- [110] Purvis, J. W. and Burkhalter, J. E. “Prediction of critical Mach number for store configurations.” *AIAA Journal*, 17(11):1170–1177, 1979.
- [111] Reed, W. H. and Hill, T. R. “Triangular mesh methods for the neutron transport equation.” Report Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
- [112] Roe, P. L. “Approximate Riemann solvers, parameter vectors, and difference schemes.” *Journal of Computational Physics*, 43(2):357–372, 1981.

- [113] Saad, Y. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 1996.
- [114] Schwab, C. *p- and hp- Finite Element Methods*. Oxford Science Publications, Great Clarendon Street, Oxford, UK, 1998.
- [115] Spalart, P. R. and Allmaras, S. R. “A one-equation turbulence model for aerodynamics flows.” AIAA 1992-0439, January 1992.
- [116] Süli, E. and Houston, P. “Adaptive finite element approximation of hyperbolic problems.” In Barth, T., Griebel, M., Keyes, D. E., Nieminen, R. M., Roose, D., and Schlick, T., editors, *Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, volume 25. Springer, Berlin, 2002.
- [117] Sun, H. “Impact of triangle shapes using high-order discretizations and direct mesh adaptation for output error.” Masters thesis, Massachusetts Institute of Technology, Computation for Design and Optimization Program, 2009.
- [118] Trefethen, L. N. and Bau, D. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.
- [119] van Leer, B. “Towards the ultimate conservative difference scheme. IV - A new approach to numerical convection.” *Journal of Computational Physics*, 23:276–299, 1977.
- [120] Vassberg, J. C., DeHaan, M. A., and Sclafani, T. J. “Grid generation requirements for accurate drag predictions based on OVERFLOW calculations.” AIAA 2003-4124, 2003.
- [121] Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Einfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., k.R. Laffin, and Mavriplis, D. J. “Summary of the Third AIAA CFD Drag Prediction Workshop.” AIAA 2007-260, 2007.
- [122] Venditti, D. A. *Grid Adaptation for Functional Outputs of Compressible Flow Simulations*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2002.
- [123] Venditti, D. A. and Darmofal, D. L. “Grid adaptation for functional outputs: application to two-dimensional inviscid flows.” *Journal of Computational Physics*, 176(1): 40–69, 2002. doi: 10.1006/jcph.2001.6967.
- [124] Venditti, D. A. and Darmofal, D. L. “Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows.” *Journal of Computational Physics*, 187(1):22–46, 2003.
- [125] Venkatakrishnan, V., Mani, M., and Yong, D. Personal Communication via email, June 2011.
- [126] Wang, L. and Mavriplis, D. J. “Adjoining-based *h-p* adaptive discontinuous Galerkin methods for the compressible Euler equations.” AIAA 2009-952, 2009.

- [127] Wheeler, M. “An elliptic collocation-finite element method with interior penalties.” *SIAM Journal on Numerical Analysis*, 15:152–161, 1978.
- [128] Wilcox, D. C. “Reassessment of the scale-determining equation for advanced turbulence models.” *AIAA Journal*, 26(11):1299–1310, 1988.
- [129] Wilcox, D. C. *Turbulence Modeling for CFD*. DCW Industries, Inc., La Cañada, California, 2006.
- [130] Yano, M. and Darmofal, D. “On dual-weighted residual error estimates for p -dependent discretizations.” ACDL Report TR-11-1, Massachusetts Institute of Technology, 2011.
- [131] Yano, M., Modisette, J. M., and Darmofal, D. “The importance of mesh adaptation for higher-order discretizations of aerodynamic flows.” AIAA 2011–0000, June 2011.
- [132] Young, D. P., Melvin, R. G., Bieterman, M. B., Johnson, F. T., Samant, S. S., and Bussoletti, J. E. “A higher-order boundary treatment for Cartesian-grid methods.” *Journal of Computational Physics*, 92:1–66, 1991.