On the Potential Impact of Curved Meshing for Higher-order Adaptive Mesh Simulations

by

Christopher Womack

B.S., Massachusetts Institute of Technology (2021)

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

Master of Science in Aerospace Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2024

Authored by: Christopher Womack
Department of Aeronautics and Astronautics
January 11, 2024

Certified by: Dr. David L. Darmofal
Jerome C. Hunsaker Professor
Thesis Supervisor

Accepted by: Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

On the Potential Impact of Curved Meshing for Higher-order Adaptive Mesh Simulations

by

Christopher Womack

Submitted to the Department of Aeronautics and Astronautics
on January 11, 2024, in partial fulfillment of the
requirements for the degree of
Master of Science in Aerospace Engineering

Abstract

Higher order, adaptive finite element methods have demonstrated the ability to significantly reduce the human and computational cost of accurately approximating the solution to partial differential equations (PDEs). In this thesis, we consider the potential advantages of incorporating higher-order element shapes, i.e. curved meshes, into an adaptive process through the use of a mesh-based, geometric mapping. While previous work has considered the generation of curved meshes to account for geometry curvature, less research has attempted to curve meshes to control error in an adaptive process. This work considers adaptive finite element methods for the advection-diffusion PDE in both Cartesian and polar coordinate systems, with the polar coordinate transformation serving to demonstrate the potential benefits of incorporating curvature into an adaptive meshing process. Results are presented for both uniform and adaptive refinement, considering first a volume output problem, followed by a boundary output problem; analytic solutions to these canonical problems are derived and presented as well. The results of this investigation demonstrate that, for each polynomial order, discretization, and output functional tested, solving the advection-diffusion equation in a polar coordinate system achieves significantly higher levels of accuracy in computing output quantities of interest. These results also showcase the potential improvements which are possible with the use of an adaptive process which incorporates element curving to control error. Additionally, adjoint analysis performed in this work shows how the form the primal output functional affects the adjoint PDE and boundary conditions.

Thesis Supervisor: Dr. David L. Darmofal
Title: Jerome C. Hunsaker Professor

Acknowledgments

First and foremost I would like to thank my incredible trio of advisors, Dave, Marshall, and Steve, for the countless hours you spent mentoring me, helping me learn variational calculus and dyadic notation, and of course, debugging SANS. Because of you I have been able to grow so much as not only a researcher, but as a person, and without you this work truly would not have been possible. Thanks to your infectious enthusiasm, working with you and learning from has been such a pleasure. Thank you for a great two years of research in the ACDL/ACSEL.

I would like to thank Dr. Cory Vincent Frontin for his endless help throughout my time here at MIT. It was a strange experience going from being your UROP in 2019 to working one desk over in 2021, but you have been a great friend and a patient colleague - I can't count the number of times you had to explain to me how different pieces of SANS worked (including just the command line). Congratulations on becoming *Dr.* Frontin and congratulations to both you and Judy for your biggest undertaking yet. I wish you, Judy, and Cassius nothing but the best.

I would like to thank my family for being my number one supporters in all of my endeavors. I know I can always count on you all to be there for me regardless of what it is I'm doing, and I can't thank you enough for believing in me, even if some of you don't believe that math without numbers is actually math.

I would like to thank Darya for being my rock through this strenuous journey, and for always bringing warmth and joy into my life.

I would like to thank all of my friends both in and around the lab who I have been fortunate enough to go on this journey with. To Josh, Sarah, Mike, Saba, Emily, Loek, Danny, Joanna, Madelyn, Maggie, Tanya, and so many others, thank you for making my time here not just memorable, but downright legendary.

I would like to thank the amazing communities I have been fortunate enough to be a part of during my time here at MIT both as an undergraduate and graduate student: Phi Delta Theta, Camp Kesem, MISTI, GA^3, GSU, SPI, TPP, Burton Conner, and the Ballroom Dance Team. Thank you for all the opportunities you have given me to grow, and a special thanks to all my Toucans who have made this past year even more fun than the first.

Finally, I would like to thank Oak Ridge National Laboratory for supporting this work. Because of you, I have been able to pursue fascinating research over the past two years, and I hope that my results will contribute to the future of higher-order finite element methods in a meaningful way.

Contents

List of Figures

Chapter 1

Introduction

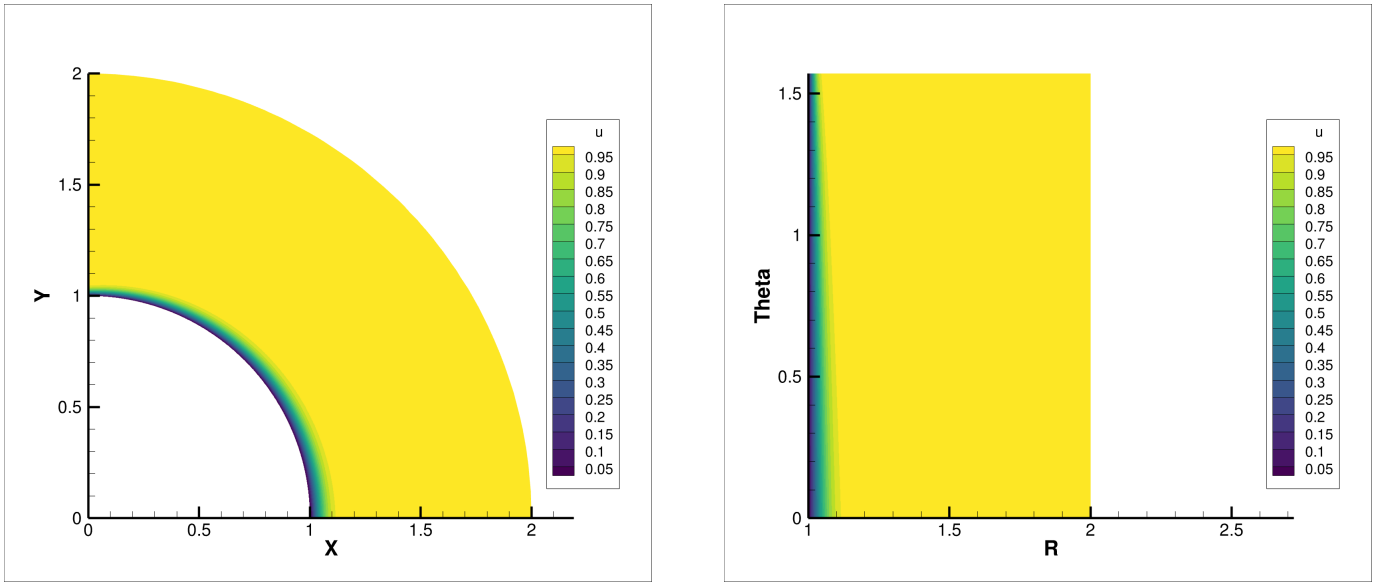
1.1 Motivation

Higher order, adaptive finite element methods have demonstrated the ability to significantly reduce the human and computational cost of accurately approximating the solution to partial differential equations (PDEs). The significant human cost of generating meshes is essentially eliminated through the use of an error-based adaptive method. Further, the computational costs can often be reduced compared to achieving the same error using second-order accurate finite volume and similar methods on best-practice *a priori* meshes. This is true even accounting for the additional cost of performing error estimation and adapting the mesh. Even without adaptation and higher order accuracy, linear finite element methods (which are also nominally second-order accurate) have been shown to achieve significantly less error on the same a priori mesh than second-order accurate finite volume methods. A sampling of past research that illustrates the potential impact of higher-order adaptive finite element method includes the following work [1]–[15].

In this thesis, we consider the potential advantages of incorporating higher-order element shapes, i.e. curved meshes, into an adaptive process. We will generally refer to higher-order element shapes as curved meshing to decrease the possible confusion with higher-order element solution spaces. Further, we will denote the polynomial order of the solution space as p, while the polynomial order of the element shape will be q. To date, while a variety of researchers have considered the generation of curved meshes to account for geometry curvature [16]–[20], less research has attempted to curve meshes to control error in an adaptive process. Sanjaya et al. showed the potential for improved accuracy in prediction of output errors when utilizing the higher-order, metric-conforming mesh generation algorithm (HOMES) [21], and

Zhang et al. provided a methodology to control interpolation error in the creation of a higher-order mesh using an *a priori* estimate [22]. This work implements an estimate for interpolation error over element edges as opposed to entire elements, providing a potential framework towards more general and widely applicable results. Rochery also investigated the use of *a priori* error estimates for higher-order metric-based anisotropic mesh adaptation, deriving a general version of the metric-based interpolation error for a linear mesh given by [23], [24], finding significantly higher rates of convergence on optimized, curved meshes than those of the corresponding straight meshes [25]. On the other hand, work from Botti and Di Pietro showed that, unless an optimal polynomial face degree is chosen, curved meshes can actually reduce the order of convergence relative to a straight mesh; this work instead proposes the use of physical frame polynomials over mesh elements [26]. Moxey et al. showed similar results by providing interpolation error bounds for curvilinear elements, demonstrating $(\lfloor p/q \rfloor + 1)$-th order convergence in the worst case for p-th order finite elements on degree q meshes [27]. As a result, they advocate for adaptation based on the control of $(\lfloor p/q \rfloor + 1)$-th order derivatives in the context of reference-frame/classic Galerkin schemes. However, results from Sanjaya et al. as well as Rochery showed $\mathcal{O}(h^{1/2})$ superconvergence in 2D of the L_2 norm of interpolation error, with Docampo-Sánchez et al. additionally observing $2p$ superconvergence of surface-related error through disparity measure minimization and a curve re-parameterization [28].

The approach taken in this thesis to quantify the potential benefits of using mesh curving to help control errors in an adaptive process is to consider a problem in which the PDE solution has a boundary layer behavior along a curved surface. Then, we transform the problem into a coordinate system in which the geometry and dominant solution behavior is aligned with the coordinate directions. For simplicity, we choose to solve the advection-diffusion equation in an annular geometry with the exact solution as shown in Figure 1-1a. Then, we transform that problem into the polar-cylindrical coordinate system where the equivalent geometry and the solution are shown in Figure 1-1b. As a result, the problem in polar-cylindrical coordinates has straight geometry and the solution anisotropy is well aligned with the coordinate directions. Thus, the polar-cylindrical representation effectively embeds the curvature of the domain and solution such that adaptive solutions of the transformed problem can be used as a surrogate for the performance of an adaptive meshing algorithm in Cartesian coordinates that incorporates mesh curving (as well as mesh size and anisotropy) to control errors. As we demonstrate through our examples, significant benefits exist for using mesh curvature, in conjunction with mesh size and anisotropy, to adaptively control errors.



(a) *Cartesian coordinates* (x, y). **(b)** *Polar-cylindrical coordinates* (r, θ).

Figure 1-1: *Thin boundary layer representation in both Cartesian and polar-cylindrical coordinate systems.*

1.2 Background

1.2.1 Mesh Adaptation

Finite element solution quality depends largely on the quality of the mesh, which, when generated by hand, is dependent on the user's *a priori* knowledge. This poses a particularly significant issue when novel geometries or complex flow fields are considered: even with an experienced user, how can they be certain they are generating a mesh that adequately solves the problem of interest? Adaptive meshing techniques seek to eliminate this risk by removing the potential of operator error from the meshing process, automatically generating a mesh that controls the discretization error for a given problem without the need for *a priori* knowledge.

In this work, we will consider output-based adaptation, a technique that aims to minimize the error in an output quantity of interest, such as lift or drag, relying on *a posteriori* error estimation [2], [29]–[32]. Error estimates are obtained by the Dual-Weighted Residual (DWR) framework for finite element methods developed by Becker and Rannacher [29], [33], which uses the adjoint PDE to quantify sensitivities in the output functional to errors in the numerical solution. As a result, the areas targeted by the algorithm are those that impact error in the output; the mesh is refined in areas with high output error and coarsened in areas with low output error.

1.2.2 Higher-Order Discretizations

While the finite volume discretization is currently the primary method used in industry CFD codes, finite element methods have been increasing in popularity due to the ease with which higher-order accurate solutions can be obtained. Improved accuracy on a given mesh can be achieved through mesh refinement, known as h refinement, where h is some measure of grid size. Finite volume methods are typically second-order accurate; the solution error decreases as $\mathcal{O}(h^2)$ as the grid is refined, leading to a need for extremely fine meshes in order to achieve a high degree of accuracy. Another way to improve accuracy is through increasing the polynomial order of the basis function in a given element, referred to as p refinement. p refinement is possible for the finite volume scheme, but results in high computational costs due to the use of extended stencils. Babuska et al. [34] developed the p-type method for higher-order finite element discretizations, which allows for the additional accuracy associated with p refinement without extending the stencil when viewed by elements.

Finite element methods traditionally use the Continuous Galerkin (CG) discretization. Unfortunately this method is unstable for pure advection, and, more generally, requires additional stabilization even for advection-diffusion problems unless the mesh is adequately refined. The Streamline Upwind Petrov-Galerkin (SUPG) [35], [36], Galerkin Least-Square (GLS) [37], and Variational Multiscale Method (VMS) [38], are the most commonly used stabilization schemes for CG, though they each have issues when considered alongside adaptive meshing. For advection-diffusion, SUPG and GLS are stable, but are not adjoint consistent for $p > 1$, while VMS is adjoint consistent but not reliably stable for $p > 1$. These issues of stability and adjoint consistency are further amplified for nonlinear systems of PDEs.

Another approach is to use a Discontinuous Galerkin (DG) discretization, which does have reliable stability and adjoint consistency for advection-dominated problems even with $p > 1$ [39]. These desirable properties come at the cost of duplicate DOFs along inter-element boundaries.

The final finite element method to consider is the Variational Multiscale Method with Discontinuous Subscales (VMSD) [40], which is both stable and adjoint consistent, making it desirable for output-based adaptation. This method works by solving a global, continuous solution field and superimposing it with a local, discontinuous solution field, achieving increased accuracy along with the benefits of both the CG and DG methods. With static condensation of the elemental DG solution states, the VMSD global solve is the same size as a CG method, i.e. for a given grid, there are fewer DOFs required to achieve the same level of error for VMSD relative to DG. While we consider VMSD in this work, we expect similar benefits from the closely related Embedded DG method [41].

1.3 Outline

The outline of this thesis is as follows: in Chapter 2 the strong and weak forms of the advection-diffusion PDE are introduced in both Cartesian and polar-cylindrical coordinate systems. This is supplemented by a brief discussion of the Solution Adaptive Numerical Solver (SANS), which is used to discretize and solve the governing equations, including an explanation of the mesh adaptation algorithm used throughout this work. Chapter 3 then provides a comprehensive derivation of the adjoint PDE for the advection-diffusion equation, providing boundary conditions (BCs) on the adjoint PDE as well as constraints on the primal output functional. Then in Chapter 4, numerical results are presented for two-dimensional test cases in Cartesian, and polar-cylindrical coordinate systems, demonstrating the potentially significant benefits of incorporating mesh curvature to adaptively control discretization error. Finally, in Chapter 5 some conclusions are drawn based on the numerical results found.

Chapter 2

Methodology

The strong and weak forms of the governing equation in both Cartesian and polar coordinates are presented in Section 2.1. Section 2.2 then discusses some nuances of implementing a polar-cylindrical finite element method. Section 2.3 outlines the process used to estimate output errors. Finally, Section 2.4 outlines the adaptation algorithm, MOESS.

Throughout this section, the following conventions are used for vector notation:

$$\begin{aligned} \text{Cartesian: } \vec{a} &= (a_x, a_y, a_z) \\ \text{Polar-Cylindrical: } \vec{a} &= (a_r, a_\theta, a_z) \end{aligned} \tag{2.1}$$

All non-Cartesian derivations in this chapter will consider the three-dimensional, polar-cylindrical coordinate system consisting of (r, θ, z). References to the axisymmetric problem (invariance in the $\hat{\theta}$ direction, the two-dimensional coordinate system (r, z)) and the polar problem (invariance in the $\hat{z}$ direction, the two-dimensional coordinate system (r, θ)) will be made throughout this section, and their derivations are omitted as they are subsets of the polar-cylindrical coordinate system.

2.1 Governing Equations

Strong Form and Assumptions

This work considers the steady advection-diffusion equation to model fluid flow, which is written as,

$$\nabla \cdot (\vec{a}u - \mathcal{K} \cdot \nabla u) - f(\vec{x}) = 0 \tag{2.2}$$

Where $\vec{a}$ is the advective velocity, $\mathcal{K}$ is the diffusion tensor, and $f(\vec{x})$ denotes the forcing function. Note that, in this notation, the forcing function is a function only of the spatial location, $\vec{x}$, and not of the state, u. This work will consider the simplified case of a divergence-free advective velocity field, $\nabla \cdot \vec{a} = 0$.
We now analyze the differences between the Cartesian and polar-cylindrical formulations of the strong form PDE.

Cartesian Coordinates

In Cartesian coordinates, the gradient and divergence operators are defined as,

$$\begin{aligned} \nabla u &= \frac{\partial u}{\partial x}\hat{x} + \frac{\partial u}{\partial y}\hat{y} + \frac{\partial u}{\partial z}\hat{z} \\ \nabla \cdot \vec{f} &= \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z} \end{aligned} \tag{2.3}$$

Using these definitions, the advective term can be expanded as,

$$\begin{aligned} \nabla \cdot (\vec{a}u) &= u\nabla \cdot \vec{a} + (\nabla u) \cdot \vec{a} \\ &= a_x \frac{\partial u}{\partial x} + a_y \frac{\partial u}{\partial y} + a_z \frac{\partial u}{\partial z} \end{aligned} \tag{2.4}$$

with the assumption of a divergence-free velocity field.
We first consider the case of the full viscous tensor, $\mathcal{K}$, which is better written in dyadic notation as:

$$\begin{aligned} \mathcal{K} = &\left(k_{xx}\hat{x} + k_{xy}\hat{y} + k_{xz}\hat{z}\right)\hat{x} \\ &+ \left(k_{yx}\hat{x} + k_{yy}\hat{y} + k_{yz}\hat{z}\right)\hat{y} \\ &+ \left(k_{zx}\hat{x} + k_{zy}\hat{y} + k_{zz}\hat{z}\right)\hat{z} \end{aligned} \tag{2.5}$$

We begin by first calculating the inner term,

$$\begin{aligned} \mathcal{K} \cdot \nabla u = &\left(k_{xx}\frac{\partial u}{\partial x} + k_{xy}\frac{\partial u}{\partial y} + k_{xz}\frac{\partial u}{\partial z}\right)\hat{x} \\ &+ \left(k_{yx}\frac{\partial u}{\partial x} + k_{yy}\frac{\partial u}{\partial y} + k_{yz}\frac{\partial u}{\partial z}\right)\hat{y} \\ &+ \left(k_{zx}\frac{\partial u}{\partial x} + k_{zy}\frac{\partial u}{\partial y} + k_{zz}\frac{\partial u}{\partial z}\right)\hat{z} \end{aligned} \tag{2.6}$$

We now take the divergence of this term, giving our result,

$$\begin{aligned}
\nabla \cdot (\mathcal{K} \cdot \nabla u) = \nabla \cdot \Bigg[& \left(k_{xx}\frac{\partial u}{\partial x} + k_{xy}\frac{\partial u}{\partial y} + k_{xz}\frac{\partial u}{\partial z} \right) \hat{x} \\
& + \left(k_{yx}\frac{\partial u}{\partial x} + k_{yy}\frac{\partial u}{\partial y} + k_{yz}\frac{\partial u}{\partial z} \right) \hat{y} \\
& + \left(k_{zx}\frac{\partial u}{\partial x} + k_{zy}\frac{\partial u}{\partial y} + k_{zz}\frac{\partial u}{\partial z} \right) \hat{z} \Bigg] \\
= \quad & \left(\frac{\partial k_{xx}}{\partial x} + \frac{\partial k_{yx}}{\partial y} + \frac{\partial k_{zx}}{\partial z} \right) \frac{\partial u}{\partial x} \\
& + \left(\frac{\partial k_{xy}}{\partial x} + \frac{\partial k_{yy}}{\partial y} + \frac{\partial k_{zy}}{\partial z} \right) \frac{\partial u}{\partial y} \\
& + \left(\frac{\partial k_{xz}}{\partial x} + \frac{\partial k_{yz}}{\partial y} + \frac{\partial k_{zz}}{\partial z} \right) \frac{\partial u}{\partial z} \\
& + k_{xx}\frac{\partial^2 u}{\partial x^2} + k_{yy}\frac{\partial^2 u}{\partial y^2} + k_{zz}\frac{\partial^2 u}{\partial z^2} \\
& + (k_{xy} + k_{yx})\frac{\partial^2 u}{\partial x \partial y} + (k_{xz} + k_{zx})\frac{\partial^2 u}{\partial x \partial z} + (k_{yz} + k_{zy})\frac{\partial^2 u}{\partial y \partial z}
\end{aligned} \tag{2.7}$$

With the assumption of a constant, scalar viscosity, μ, which corresponds to a constant diagonal viscous tensor (i.e. $k_{ij} = 0$, $\forall i \neq j$, $k_{xx} = k_{yy} = k_{zz} = \mu$), this expression becomes,

$$\nabla \cdot (\mu \nabla u) = \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \tag{2.8}$$

This formulation will be considered in the first section of Chapter 4.

Polar-Cylindrical Coordinates

The polar-cylindrical gradient and divergence operators are defined as,

$$\begin{aligned}
\nabla u &= \frac{\partial u}{\partial r}\hat{r} + \frac{1}{r}\frac{\partial u}{\partial \theta}\hat{\theta} + \frac{\partial u}{\partial z}\hat{z} \\
\nabla \cdot \vec{f} &= \frac{1}{r}\frac{\partial}{\partial r}(r f_r) + \frac{1}{r}\frac{\partial f_\theta}{\partial \theta} + \frac{\partial f_z}{\partial z}
\end{aligned} \tag{2.9}$$

The advective term is expanded as,

$$\begin{aligned}
\nabla \cdot (\vec{a} u) &= u \nabla \cdot \vec{a} + (\nabla u) \cdot \vec{a} \\
&= a_r \frac{\partial u}{\partial r} + \frac{a_\theta}{r}\frac{\partial u}{\partial \theta} + a_z \frac{\partial u}{\partial z}
\end{aligned} \tag{2.10}$$

with the assumption of a divergence-free velocity field.

As in the Cartesian case, we begin the full viscous tensor. In polar-cylindrical coordinates the viscous tensor is defined as,

$$\begin{aligned}\mathcal{K} = &\left(k_{rr}\hat{r} + k_{r\theta}\hat{\theta} + k_{rz}\hat{z}\right)\hat{r} \\ &+ \left(k_{\theta r}\hat{r} + k_{\theta\theta}\hat{\theta} + k_{\theta z}\hat{z}\right)\hat{\theta} \\ &+ \left(k_{zr}\hat{r} + k_{z\theta}\hat{\theta} + k_{zz}\hat{z}\right)\hat{z}\end{aligned} \tag{2.11}$$

We begin by first calculating the inner term,

$$\begin{aligned}\mathcal{K}\cdot\nabla u = &\left(k_{rr}\frac{\partial u}{\partial r} + \frac{k_{r\theta}}{r}\frac{\partial u}{\partial\theta} + k_{rz}\frac{\partial u}{\partial z}\right)\hat{r} \\ &+ \left(k_{\theta r}\frac{\partial u}{\partial r} + \frac{k_{\theta\theta}}{r}\frac{\partial u}{\partial\theta} + k_{\theta z}\frac{\partial u}{\partial z}\right)\hat{\theta} \\ &+ \left(k_{zr}\frac{\partial u}{\partial r} + \frac{k_{z\theta}}{r}\frac{\partial u}{\partial\theta} + k_{zz}\frac{\partial u}{\partial z}\right)\hat{z}\end{aligned} \tag{2.12}$$

We now take the divergence of this term,

$$\begin{aligned}\nabla\cdot(\mathcal{K}\cdot\nabla u) = &\nabla\cdot\left[\left(k_{rr}\frac{\partial u}{\partial r} + \frac{k_{r\theta}}{r}\frac{\partial u}{\partial\theta} + k_{rz}\frac{\partial u}{\partial z}\right)\hat{r}\right. \\ &+ \left(k_{\theta r}\frac{\partial u}{\partial r} + \frac{k_{\theta\theta}}{r}\frac{\partial u}{\partial\theta} + k_{\theta z}\frac{\partial u}{\partial z}\right)\hat{\theta} \\ &\left.+ \left(k_{zr}\frac{\partial u}{\partial r} + \frac{k_{z\theta}}{r}\frac{\partial u}{\partial\theta} + k_{zz}\frac{\partial u}{\partial z}\right)\hat{z}\right] \\ = &\frac{1}{r}\frac{\partial}{\partial r}\left[r\left(k_{rr}\frac{\partial u}{\partial r} + \frac{k_{r\theta}}{r}\frac{\partial u}{\partial\theta} + k_{rz}\frac{\partial u}{\partial z}\right)\right] \\ &+ \frac{1}{r}\frac{\partial}{\partial\theta}\left(k_{\theta r}\frac{\partial u}{\partial r} + \frac{k_{\theta\theta}}{r}\frac{\partial u}{\partial\theta} + k_{\theta z}\frac{\partial u}{\partial z}\right) \\ &+ \frac{\partial}{\partial z}\left(k_{zr}\frac{\partial u}{\partial r} + \frac{k_{z\theta}}{r}\frac{\partial u}{\partial\theta} + k_{zz}\frac{\partial u}{\partial z}\right)\end{aligned} \tag{2.13}$$

Expanding this gives our result,

$$\begin{aligned}\nabla \cdot (\mathcal{K} \cdot \nabla u) = & \frac{\partial u}{\partial r}\left[\frac{\partial k_{rr}}{\partial r} + \frac{1}{r}\left(k_{rr} + \frac{\partial k_{r\theta}}{\partial \theta}\right) + \frac{\partial k_{rz}}{\partial z}\right] \\ & + \frac{1}{r}\frac{\partial u}{\partial \theta}\left[\frac{\partial k_{\theta r}}{\partial r} + \frac{1}{r}\frac{\partial k_{\theta\theta}}{\partial \theta} + \frac{\partial k_{\theta z}}{\partial z}\right] \\ & + \frac{\partial u}{\partial z}\left[\frac{\partial k_{zr}}{\partial r} + \frac{1}{r}\left(k_{zr} + \frac{\partial k_{z\theta}}{\partial \theta}\right) + \frac{\partial k_{zz}}{\partial z}\right] \\ & + k_{rr}\frac{\partial^2 u}{\partial r^2} + k_{\theta\theta}\frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} + k_{zz}\frac{\partial^2 u}{\partial z^2} \\ & + \left(k_{r\theta} + k_{\theta r}\right)\frac{1}{r}\frac{\partial^2 u}{\partial r \partial \theta} + \left(k_{rz} + k_{zr}\right)\frac{\partial^2 u}{\partial r \partial z} + \left(k_{\theta z} + k_{z\theta}\right)\frac{1}{r}\frac{\partial^2 u}{\partial \theta \partial z}\end{aligned} \tag{2.14}$$

As in the Cartesian case, we assume constant, scalar μ,

$$\nabla \cdot (\mu \nabla u) = \mu\left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} + \frac{\partial^2 u}{\partial z^2}\right) \tag{2.15}$$

Weak Form and Finite Element Statement

In order to see how the differences in the Cartesian and polar-cylindrical governing equations manifest in the implementation, the weak form and corresponding finite element statement are derived. Beginning from Equation 2.2 and considering a domain Ω, with boundary $\partial\Omega$, the weak form is obtained by weighting the equation with a function v and then integrating by parts:

$$\begin{aligned}\mathcal{R}(u, v) = & \int_\Omega \Big(- \nabla v \cdot (\vec{a} u - \mathcal{K} \cdot \nabla u) - v f \Big)\, dV \\ & + \int_{\partial\Omega} v\, (\vec{a} u - \mathcal{K} \cdot \nabla u) \cdot \hat{n}\, dS\end{aligned} \tag{2.16}$$

$\mathcal{R}(u, v)$ here is the residual. It should be noted that the weak formulation shown above does not include treatment of the BCs. From this we are able to obtain a finite element statement for the advection-diffusion problem by introducing a tessellation $\mathcal{T}_h$ of Ω into non-overlapping elements with polynomial solutions on each element. The resulting discrete problem is:
Find $u_{h,p} \in \mathcal{V}_{h,p}$ such that,

$$\mathcal{R}_h(u_{h,p}, v_{h,p}) = 0, \quad \forall v_{h,p} \in \mathcal{V}_{h,p} \tag{2.17}$$

The choice of space as well as $\mathcal{R}_h$ depend on the specific method (e.g. CG, DG, VMSD, etc.).

2.2 Implementation of polar-cylindrical FEM

Although both the Cartesian and polar-cylindrical formulations are identical when written in divergence form, the expansion of terms is necessary to highlight the nuance present in the implementation of the polar-cylindrical finite element method relative to the Cartesian problem. Differences arise between the polar-cylindrical and the Cartesian governing equations from the additional factors of r and $\frac{1}{r}$ present in the polar-cylindrical gradient and divergence definitions. Similar differences also occur for the volume and surface terms including the normal vectors:

Volume and Surface Differentials

Considering now the 2D case, in the Cartesian coordinate system, volume and surface differentials are defined as,

$$dS = \sqrt{dx^2 + dy^2} \tag{2.18}$$

$$dV = dx\, dy \tag{2.19}$$

while in polar coordinates (no variation in the $\hat{z}$ direction), these quantities are defined as,

$$dS = \sqrt{dr^2 + r^2\, d\theta^2} \tag{2.20}$$

$$dV = r\, dr\, d\theta \tag{2.21}$$

Note the introduction of the differential multiplier r in the definition of dV, which is also present in the definition of dS (though this differential only appears on a boundary which varies in the $\hat{\theta}$ direction). This appears anywhere integration over an element/the domain occurs, including the weak forms of both the primal and adjoint PDE as well as stabilization schemes (SUPG, GLS, and VMS for CG and the BR2 lifting operators for DG and VMSD).

Normal Vectors

In the Cartesian coordinate system, the unit normal vector is defined as,

$$\hat{n} = (dy\, \hat{x} - dx\, \hat{y})/dS \tag{2.22}$$

which is in contrast to the polar coordinate system, where the differential multiplier r appears once again. In the polar case considered in Chapter 4, our normal vector is defined as,

$$\hat{n} = (r\, d\theta\, \hat{r} - dr\, \hat{\theta})/dS \tag{2.23}$$

2.3 Output Error Estimation

Adaptivity in this context of this work is driven by the ability to estimate the error in a given output based on the DWR method. We first separate the residual into a bilinear and linear form:

$$\mathcal{R}_h(u_{h,p}, v_{h,p}) \equiv a_h(u_{h,p}, v_{h,p}) - l_h(v_{h,p}) \tag{2.24}$$

Given a linear output functional of interest, $\mathcal{J}(u)$, the exact adjoint solution $w \in \mathcal{V} + \mathcal{V}_{h,p}$ is related to the output functional by (see Chapter 3),

$$\mathcal{R}_h^*(v, w) \equiv a_h(v, w) - \mathcal{J}(v) = 0 \quad \forall v \in \mathcal{V} + \mathcal{V}_{h,p}, \tag{2.25}$$

where $\mathcal{V}$ is the solution space and $\mathcal{V}_{h,p}$ is the discrete solution space. The exact output error, $\mathcal{E}$, is then calculated through the DWR framework using the adjoint,

$$\mathcal{E} \equiv \mathcal{J}(u) - \mathcal{J}(u_{h,p}) = -\mathcal{R}_h(u_{h,p}, w - v_{h,p}) \quad \forall v_{h,p} \in \mathcal{V}_{h,p} \tag{2.26}$$

Calculating the error in this manner depends on the exact adjoint, which is unknown. Instead, the adjoint is calculated in an enriched $\hat{p} = p + 1$ space, and this is used as an approximation for the exact adjoint, $w \approx w_{h,\hat{p}}$.

2.4 Mesh Optimization

Mesh optimization seeks to solve the following optimization problem for an optimal mesh, $\mathcal{T}^*$,

$$\mathcal{T}^* = \underset{\mathcal{T} \in \mathbb{T}(\Omega)}{\arg\inf}\, \mathcal{E}(\mathcal{T}) \tag{2.27a}$$

$$\text{s.t. } \mathbb{C}(\mathcal{T}) \leq \mathbb{C}, \tag{2.27b}$$

where $\mathbb{T}(\Omega)$ is the space of meshes that conform to the physical domain Ω, $\mathcal{E}$ is the error, such as defined in Section 2.3, and $\mathbb{C}(\cdot)$ is a cost functional. Performing this optimization on a discrete mesh is intractable, and we instead perform the minimization over a continuous metric field, a smooth field composed of symmetric positive definite matrices $\mathcal{M}(x)$. The duality between a discrete mesh, $\mathcal{T}_h$, and the representation of a continuous mesh using a Riemannian metric field, $\mathcal{M} = \mathcal{M}(x)|_{x\in\Omega}$ was demonstrated by Loiselle [42]–[44]. To exploit mesh-metric duality, a mesh generator is used that is metric-conforming, i.e. able to generate meshes with edge lengths e which are approximately unit under the metric: $e^T\mathcal{M}(x)e \approx 1$. Using mesh-metric duality, Equation 2.27 becomes,

$$\mathcal{M}^* = \underset{\mathcal{M}\in\mathbb{M}(\Omega)}{\arg\inf}\ \mathcal{E}(\mathcal{M}), \quad \text{s.t. } \mathbb{C}(\mathcal{M}) \leq N \tag{2.28}$$

where the bound on the cost function is defined as the number of degrees of freedom in the mesh and is bounded by a user-defined parameter N.

This work considers the Mesh Optimization via Error Sampling and Synthesis (MOESS) algorithm to model the sensitivity of an error functional to perturbations in the mesh (i.e. changes in element shape and size) [45]–[47]. In the context of this work, MOESS aims to optimize the mesh to minimize the error between the computed output and the true output, which is given by either a volume or boundary output functional as discussed in Chapter 4. The global error model is given by the summation of local errors, denoted η, which are assumed to only be a function of the local metric. This model is given by:

$$\mathcal{E}(\mathcal{M}) = \int_\Omega \eta\left(\mathcal{M}(x), x\right) dx \tag{2.29}$$

MOESS constructs surrogate models for the local error via local sampling, which measures sensitivity to local mesh perturbations. MOESS thus does not directly solve Equation 2.27, instead using mesh-metric duality to solve Equation 2.28.

For a given problem, this optimization is iteratively repeated. To complete one adaptation iteration, the ensuing steps are followed:

1. Compute the primal $u_{h,p}$ and $\hat{p} = p + 1$ adjoint $w_{h,\hat{p}}$

2. Compute local error estimates η_0 for the current local metric $\mathcal{M}_0$

3. Split elements and performing sampling to form a surrogate error model, using the methods described in Refs. [3], [4], [45]

4. Solve the non-linear optimization problem, Equation 2.27, via the Method of Moving Asymptotes [48] to find the metric field that minimizes the error at a given number of degrees of freedom

5. Generate mesh using a metric-conforming mesher

Throughout Chapter 4, the metric-conforming mesh generators avro [49] and EPIC [50] are utilized.

Chapter 3

Adjoint Analysis

This section aims to show how the form of the primal output functional, which is assumed to be a general, combined volume-boundary output, affects the adjoint PDE and BCs. The results of this analysis are then applied in Chapter 4 to construct analytic adjoint solutions with desirable properties (e.g. boundary layer like behavior). While this analysis can be completed when $\nabla \cdot \vec{a} \neq 0$, the assumption of a divergence-free velocity field is carried into this analysis.

For clarity in this analysis we introduce the subscripts $\mathcal{R}_w(u, w)$ and $\mathcal{R}_s(u, w)$ to denote the weak and strong form residuals respectively; these subscripts are used to differentiate the forms used in this chapter (i.e. residuals with the imposition of BCs) from the forms presented in Chapter 2.1. In Section 3.1.1 the general adjoint along with its restrictions is derived for the pure advection case, and the same analysis is then conducted in Section 3.1.2 for the advection diffusion PDE.

3.1 General Enforcement of Boundary Conditions

3.1.1 Advection

Consider first the pure advection case (i.e. $\mu = 0$) in a domain Ω with boundary $\partial\Omega$. We consider the following Dirichlet BC,

$$u = u_D \quad \forall \vec{x} \in \partial\Omega_D \tag{3.1}$$

where $\partial\Omega_D$ is the boundary on which Dirichlet BCs are applied. We begin by deriving the Galerkin weak form weighted residual with weak imposition of the BCs. This follows from a Lagrange multiplier implementation of the Dirichlet BC [51]–[53]. The Lagrange multiplier can be identified with the weak-

form boundary flux as follows. Beginning from the strong form residual, which comes from integrating Equation 2.2, our Lagrange multiplier approach begins with the inclusion of one additional term,

$$\begin{aligned}\mathcal{R}_s(u, w, \lambda_D) = &\int_\Omega w \left[\nabla \cdot (\vec{a}u) - f\right] dV \\ &+ \int_{\partial\Omega_D} \lambda_D (u - u_D)\, dS\end{aligned} \tag{3.2}$$

We assume the output will be of the form,

$$\mathcal{J}(u) = \int_\Omega gu\, dV + \int_{\partial\Omega} h_a u(\vec{a} \cdot \hat{n})\, dS \tag{3.3}$$

where the parameter h_a is a constant or function of position. Integrating our strong form residual by parts gives,

$$\begin{aligned}\mathcal{R}_s(u, w, \lambda_D) = &\int_\Omega u(-\nabla w \cdot \vec{a})\, dV - \int_\Omega wf\, dV \\ &+ \int_{\partial\Omega} wu(\vec{a} \cdot \hat{n})\, dS + \int_{\partial\Omega_D} \lambda_D (u - u_D)\, dS\end{aligned} \tag{3.4}$$

We then invoke duality, giving,

$$\begin{aligned}\mathcal{J}(u) - R_s(u, w, \lambda_D) &= \int_\Omega u(\nabla w \cdot \vec{a} + g)\, dV + \int_\Omega wf\, dV \\ &\quad - \int_{\partial\Omega} wu(\vec{a} \cdot \hat{n})\, dS + \int_{\partial\Omega} h_a u(\vec{a} \cdot \hat{n})\, dS \\ &\quad - \int_{\partial\Omega_D} \lambda_D (u - u_D)\, dS \\ &= \mathcal{J}^*(w) - R_s^*(w, u, \lambda_D)\end{aligned} \tag{3.5}$$

From this, we readily identify the adjoint output functional and strong form residual:

$$\mathcal{J}^*(w) = \int_\Omega wf\, dV + \int_{\partial\Omega_D} \lambda_D u_D\, dS \tag{3.6}$$

$$\begin{aligned}R_s^*(w, u, \lambda_D) = &- \int_\Omega u(\nabla w \cdot \vec{a} + g)\, dV \\ &+ \int_{\partial\Omega} wu(\vec{a} \cdot \hat{n})\, dS - \int_{\partial\Omega} h_a u(\vec{a} \cdot \hat{n})\, dS \\ &+ \int_{\partial\Omega_D} \lambda_D u\, dS\end{aligned} \tag{3.7}$$

The goal of the adjoint analysis is that this expression is to be independent of u. Since u is arbitrary, we must have $-\nabla w \cdot \vec{a} - g = 0$ on Ω. This defines the strong form of the adjoint. Next, we have the following terms on $\partial\Omega_D$ from our adjoint strong form residual,

$$u\left[(h_a - w)\vec{a} \cdot \hat{n} - \lambda_D\right] \tag{3.8}$$

where the bracketed terms must equal zero for our expression to be independent of u.

If we want $\mathcal{J}^*(w)$ to be linear in w then λ_D cannot depend on h_a (including h_a would result in $\mathcal{J}^*(w)$ as an affine functional of w). As a result, there are two options for λ_D and h_a:

1. $\lambda_D = -w(\vec{a} \cdot \hat{n})$ and $h_a = 0$. This would result in:

$$\mathcal{J}_{D,v}(u) = 0, \quad \mathcal{J}^*_{D,v}(w) = -\int_{\Gamma_1} w u_D (\vec{a} \cdot \hat{n})\, dS. \tag{3.9}$$

   where $\mathcal{J}_{D,v}(u)$ and $\mathcal{J}^*_{D,v}(w)$ are the boundary contributions to the output on the Dirichlet boundary where the v denotes the primal output comes from the viscous flux, which in this case is zero.

2. $\lambda_D = 0$ and $h_a = w$. This would result in:

$$\mathcal{J}_{D,a}(u) = \int_{\Gamma_2} h_a u (\vec{a} \cdot \hat{n})\, dS, \quad \mathcal{J}^*_{D,a}(w) = 0. \tag{3.10}$$

   where a denotes that these are the boundary contributions when $h_a \neq 0$.

This gives the complete output functional,

$$\mathcal{J}(u) = \int_\Omega g u\, dV + \int_{\Gamma_2} h u (\vec{a} \cdot \hat{n})\, dS \tag{3.11}$$

Note that $\partial\Omega_D$ has been broken into Γ_1 and Γ_2, which represent the choices one can make for λ_D and h_a in 1. and 2. respectively, Now, we begin from the primal strong form with λ_D as determined above to identify our weak form residual,

$$\begin{aligned} \mathcal{R}_s(u, w) = &\int_\Omega w\left[\nabla \cdot (\vec{a}u) - f\right] dV \\ &- \int_{\Gamma_1} w(u - u_D)(\vec{a} \cdot \hat{n})\, dS \end{aligned} \tag{3.12}$$

Integrating this expression by parts gives our weak form,

$$\begin{aligned}\mathcal{R}_w(u,w) = &\int_\Omega \left[-\nabla w \cdot (\vec{a}u) - fw\right] dV \\ &+ \int_{\Gamma_1} wu_D(\vec{a}\cdot\hat{n})\, dS \\ &+ \int_{\Gamma_2} wu(\vec{a}\cdot\hat{n})\, dS\end{aligned} \tag{3.13}$$

In practice, we frequently "upwind" the boundary advection contribution. Thus:

- For $\partial\Omega_D$ with $\vec{a}\cdot\hat{n} < 0$, the boundary advective term is $\int_{\Gamma_1} wu_D(\vec{a}\cdot\hat{n})\, dS$, which is consistent with the viscous output, i.e. the residual component on Γ_1.
- For $\partial\Omega_D$ with $\vec{a}\cdot\hat{n} > 0$, we denote the boundary advective term is $\int_{\Gamma_2} wu(\vec{a}\cdot\hat{n})\, dS$, which is consistent with the advective output, i.e. the residual component on Γ_2.

Note that we now denote Γ_1 as the inflow boundary ($\vec{a}\cdot\hat{n} < 0$, formerly $\partial\Omega_D$) and Γ_2 as the outflow boundary ($\vec{a}\cdot\hat{n} > 0$); this distinction is made to ensure well-posedness of the resulting solution, as well-posedness requires that we cannot impose a Dirichlet BC on an outflow boundary.

3.1.2 Advection Diffusion

Consider now the advection-diffusion PDE with the following Dirichlet and Neumann BCs,

$$\begin{aligned} u &= u_D \quad \forall \vec{x} \in \partial\Omega_D \\ \nabla u \cdot \hat{n} &= b_N \quad \forall \vec{x} \in \partial\Omega_N \end{aligned} \tag{3.14}$$

where $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$. As before, we begin by deriving the Galerkin weak form weighted residual with weak imposition of the BCs through a Lagrange multiplier approach. We start from the strong form residual,

$$\begin{aligned}\mathcal{R}_s(u,w,\lambda_D,\lambda_N) = &\int_\Omega w\left[\nabla\cdot(\vec{a}u - \mu\nabla u) - f\right] dV \\ &+ \int_{\partial\Omega_D} \lambda_D(u-u_D)\, dS + \int_{\partial\Omega_N} \mu\lambda_N(\nabla u\cdot\hat{n} - b_N)\, dS\end{aligned} \tag{3.15}$$

We assume the output will be of the form,

$$\mathcal{J}(u) = \int_\Omega gu\, dV + \int_{\partial\Omega} (h_a\vec{a}u - h_v\mu\nabla u)\cdot\hat{n}\, dS \tag{3.16}$$

where parameters h_a and h_v are constants or functions of position. Integrating our strong form residual by parts gives,

$$\begin{aligned}\mathcal{R}_s(u,w,\lambda_D,\lambda_N) = &\int_\Omega u\left[-\nabla w\cdot\vec{a}-\nabla\cdot(\mu\nabla w)\right]\,dV - \int_\Omega wf\,dV\\ &+\int_{\partial\Omega} w\left[\vec{a}u-\mu\nabla u\right]\cdot\hat{n}\,dS + \int_{\partial\Omega} u\mu\nabla w\cdot\hat{n}\,dS\\ &+\int_{\partial\Omega_D}\lambda_D(u-u_D)\,dS + \int_{\partial\Omega_N}\mu\lambda_N(\nabla u\cdot\hat{n}-b_N)\,dS\end{aligned} \tag{3.17}$$

We then invoke duality, giving,

$$\begin{aligned}\mathcal{J}(u) - R_s(u,w,\lambda_D,\lambda_N) = &\int_\Omega u\left[\nabla w\cdot\vec{a}+\nabla\cdot(\mu\nabla w)+g\right]\,dV + \int_\Omega wf\,dV\\ &-\int_{\partial\Omega} w\left[\vec{a}u-\mu\nabla u\right]\cdot\hat{n}\,dS - \int_{\partial\Omega} u\mu\nabla w\cdot\hat{n}\,dS\\ &+\int_{\partial\Omega}\left[h_a u(\vec{a}\cdot\hat{n}) - h_v\mu(\nabla u\cdot\hat{n})\right]\,dS\\ &-\int_{\partial\Omega_D}\lambda_D(u-u_D)\,dS - \int_{\partial\Omega_N}\mu\lambda_N(\nabla u\cdot\hat{n}-b_N)\,dS\\ &=\mathcal{J}^*(w) - R_s^*(w,u,\lambda_D,\lambda_N)\end{aligned} \tag{3.18}$$

From this, we readily identify the adjoint output functional and strong form residual:

$$\mathcal{J}^*(w) = \int_\Omega wf\,dV + \int_{\partial\Omega_D}\lambda_D u_D\,dS + \int_{\partial\Omega_N}\mu\lambda_N b_N\,dS \tag{3.19}$$

$$\begin{aligned}R_s^*(w,u,\lambda_D,\lambda_N) = &-\int_\Omega u\left[\nabla w\cdot\vec{a}+\nabla\cdot(\mu\nabla w)+g\right]\,dV\\ &+\int_{\partial\Omega} w\left[\vec{a}u-\mu\nabla u\right]\cdot\hat{n}\,dS + \int_{\partial\Omega} u\mu\nabla w\cdot\hat{n}\,dS\\ &-\int_{\partial\Omega}\left[h_a u(\vec{a}\cdot\hat{n}) - h_v\mu(\nabla u\cdot\hat{n})\right]\,dS\\ &+\int_{\partial\Omega_D}\lambda_D u\,dS + \int_{\partial\Omega_N}\lambda_N\mu\nabla u\cdot\hat{n}\,dS\end{aligned} \tag{3.20}$$

The goal of the adjoint analysis is that this expression is to be independent of u and $\nabla u\cdot\hat{n}$. Since u is arbitrary, we must have $-\nabla w\cdot\vec{a}-\nabla\cdot(\mu\nabla w)-g=0$ on Ω. This defines the strong form of the adjoint. Next, we have the following terms on $\partial\Omega_D$ from our adjoint strong form residual,

$$u\left[(h_a-w)\vec{a}\cdot\hat{n}-\mu\nabla w\cdot\hat{n}-\lambda_D\right]-\mu\nabla u\cdot\hat{n}\left[h_v-w\right] \tag{3.21}$$

As in the previous case, the bracketed terms must be zero. The second term in this expression can only be zero if $w = h_v$. Now, we set the u multiplier to zero.

If we want $\mathcal{J}^*(w)$ to be linear in w then λ_D cannot depend on h_a, h_v (including h_a and h_v would result in $\mathcal{J}^*(w)$ as an affine functional of w). As a result, there are two options for λ_D and h_a:

1. $\lambda_D = -\hat{a} \cdot \hat{n} w - \mu \nabla w \cdot \hat{n}$ and $h_a = 0$. This would result in:

$$\mathcal{J}_{D,v}(u) = -\int_{\Gamma_1} h_v \mu \nabla u \cdot \hat{n}\, dS, \quad \mathcal{J}^*_{D,v}(w) = -\int_{\Gamma_1} u_D(\vec{a} \cdot \hat{n} w + \mu \nabla w \cdot \hat{n})\, dS \tag{3.22}$$

   where $\mathcal{J}_{D,v}(u)$ and $\mathcal{J}^*_{D,v}(w)$ are the boundary contributions on the Dirichlet boundary where the v denotes the primal output comes from the viscous flux.

2. $\lambda_D = -\mu \nabla w \cdot \hat{n}$ and $h_a = w = h_v = h_f$ where f denotes a combined "flux". This would result in:

$$\mathcal{J}_{D,f}(u) = \int_{\Gamma_2} h_f(\vec{a} \cdot \hat{n} u - \mu \nabla u \cdot \hat{n})\, dS, \quad \mathcal{J}^*_{D,f}(w) = -\int_{\Gamma_2} \mu \nabla w \cdot \hat{n} u_D\, dS \tag{3.23}$$

The Dirichlet boundaries are incorporated into the output functional as,

$$\mathcal{J}(u) = \int_\Omega g u\, dV - \int_{\Gamma_1} h_v \mu \nabla u \cdot \hat{n}\, dS + \int_{\Gamma_2} h_f(\vec{a} \cdot \hat{n} u - \mu \nabla u \cdot \hat{n})\, dS \tag{3.24}$$

Note that $\partial\Omega_D$ has been broken into Γ_1 and Γ_2, which represent the choices one can make for λ_D and h_a in 1. and 2. respectively, Now, we begin from the primal strong form with λ_D as determined above to identify our weak form residual. As a note, we only consider the Dirichlet boundary for now,

$$\begin{aligned} \mathcal{R}_s(u, w) = &\int_\Omega w \left[\nabla \cdot (\vec{a} u - \mu \nabla u) - f\right] dV \\ &- \int_{\Gamma_1} \left[(\vec{a} w + \mu \nabla w) \cdot \hat{n}\right](u - u_D)\, dS \\ &- \int_{\Gamma_2} (\mu \nabla w \cdot \hat{n})(u - u_D)\, dS \end{aligned} \tag{3.25}$$

Integrating this expression by parts gives our weak form,

$$\begin{aligned}\mathcal{R}_w(u,w) = &\int_\Omega \left[-\nabla w \cdot (\vec{a}u - \mu\nabla u) - fw\right] dV \\ &+ \int_{\Gamma_1} \left[w(\vec{a}u_D - \mu\nabla u) - \mu\nabla w(u-u_D)\right]\cdot\hat{n}\, dS \\ &+ \int_{\Gamma_2} \left[w(\vec{a}u - \mu\nabla u) - \mu\nabla w(u-u_D)\right]\cdot\hat{n}\, dS\end{aligned} \tag{3.26}$$

In practice, we frequently "upwind" the boundary advection contribution. Thus:

- For $\partial\Omega_D$ with $\vec{a}\cdot\hat{n} < 0$, the boundary advective term is $\int_{\Gamma_1} wu_D(\vec{a}\cdot\hat{n})\, dS$, which is consistent with the viscous output, i.e. the residual component on Γ_1.

- For $\partial\Omega_D$ with $\vec{a}\cdot\hat{n} > 0$, we denote the boundary advective term is $\int_{\Gamma_2} wu(\vec{a}\cdot\hat{n})\, dS$, which is consistent with the advective output, i.e. the residual component on Γ_2.

Γ_1 is the inflow boundary ($\vec{a}\cdot\hat{n} < 0$) and Γ_2 is the outflow boundary ($\vec{a}\cdot\hat{n} > 0$). As we can see from these results, Dirichlet BCs are applied on both inflow and outflow boundaries. $\partial\Omega_N$ is the inflow boundary on which Neumann BCs are applied. The distinction between which BC is applied on which boundary is made to ensure well-posedness in the advective limit of the problem (i.e. $\mu \to 0$). In the advective limit, as discussed in the previous section, we cannot set a Dirichlet BC on an outflow boundary, and as such this BC will disappear in the advective limit. We are also unable to set any Neumann or Robin BCs in the advective limit, choosing to set them on an inflow boundary in this case to mimic the choice of no Dirichlet BCs on an outflow boundary.

We now consider the treatment of $\partial\Omega_N$, which has the following terms from our adjoint strong form residual,

$$u\left[(h_a - w)\vec{a}\cdot\hat{n} - \mu\nabla w\cdot\hat{n}\right] - \mu\nabla u\cdot\hat{n}\left[h_v - w + \lambda_N\right] \tag{3.27}$$

In this case, we are left with the constraints:

$$w\vec{a}\cdot\hat{n} + \mu\nabla w\cdot\hat{n} = h_a\vec{a}\cdot\hat{n}, \quad \lambda_N = w, \quad h_v = 0. \tag{3.28}$$

which leads to the corresponding output contributions,

$$\mathcal{J}_N(u) = \int_{\partial\Omega_N} h_a\vec{a}\cdot\hat{n}u\, dS, \quad \mathcal{J}_N^*(w) = \int_{\partial\Omega_N} \mu w b_N\, dS. \tag{3.29}$$

Combining this result with the Dirichlet boundary gives the complete strong form weighted residual,

$$\begin{aligned}\mathcal{R}_s(u,w) = &\int_\Omega w\left[\nabla\cdot(\vec{a}u - \mu\nabla u) - f\right]\,dV \\ &- \int_{\Gamma_1}\left[(\vec{a}w + \mu\nabla w)\cdot\hat{n}\right](u-u_D)\,dS \\ &- \int_{\Gamma_2}(\mu\nabla w\cdot\hat{n})(u-u_D)\,dS \\ &+ \int_{\partial\Omega_N}\mu w(\nabla u\cdot\hat{n} - b_N)\,dS\end{aligned} \tag{3.30}$$

As well as the output functional,

$$\begin{aligned}\mathcal{J}(u) = \int_\Omega gu\,dV &- \int_{\Gamma_1} h\mu\nabla u\cdot\hat{n}\,dS \\ &+ \int_{\Gamma_2} h(\vec{a}u - \mu\nabla u)\cdot\hat{n}\,dS \\ &+ \int_{\partial\Omega_N} h\vec{a}u\cdot\hat{n}\,dS\end{aligned} \tag{3.31}$$

Integrating our strong form by parts gives the complete weak residual,

$$\begin{aligned}\mathcal{R}_w(u,w) = &\int_\Omega\left[-\nabla w\cdot(\vec{a}u - \mu\nabla u) - fw\right]\,dV \\ &+ \int_{\Gamma_1}\left[w(\vec{a}u_D - \mu\nabla u) - \mu\nabla w(u-u_D)\right]\cdot\hat{n}\,dS \\ &+ \int_{\Gamma_2}\left[w(\vec{a}u - \mu\nabla u) - \mu\nabla w(u-u_D)\right]\cdot\hat{n}\,dS \\ &+ \int_{\partial\Omega_N} w(\vec{a}u\cdot\hat{n} - \mu b_N)\,dS\end{aligned} \tag{3.32}$$

Chapter 4

Numerical Results

This chapter considers the results of simulating the advection-diffusion problem in both Cartesian and polar coordinates systems, using both uniform and adaptive refinement in tandem with several finite element schemes.

4.1 Primal Problem

For this section, we consider the advection-diffusion problem within an annulus, the region between two concentric circles. We define this region as $\Omega \equiv [r_0, r_1] \times [\theta_0, \theta_1]$, with $r_0 = 1$, $r_1 = 2$, $\theta_0 = 0, \theta_1 = \pi/2$, as shown in Figure 4-1.

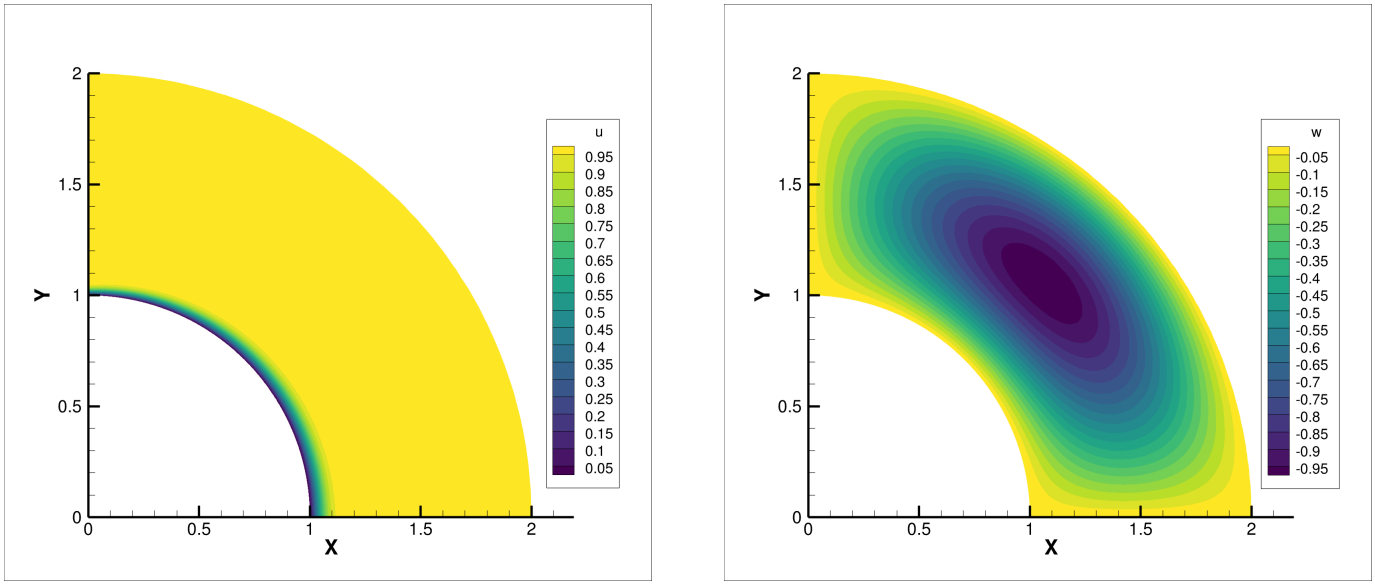
We first aim to find a primal solution which produces a Blasius-type boundary layer. This boundary layer will be located along the edge of the inner circle, increasing in thickness with decreasing θ. We assume a functional form of,

$$u(n) = u_0 \operatorname{erf}\left(\frac{n}{2}\right) = u_0 \operatorname{erf}\left(\frac{r - R}{2\sqrt{(\theta - \theta_{LE})\mu r/a_0}}\right) \tag{4.1}$$

where $n = \frac{r-R}{\sqrt{(\theta-\theta_{LE})\mu r/a_0}}$ is chosen to take into account the curvature of the annulus. With this choice of $u(n)$, we have a non-zero forcing term, f in the primal PDE, which we find using the method of manufactured solutions (MMS). The analytic function, $u(n)$, is used to impose Dirichlet BCs on all four domain boundaries for the volume output problem.

As θ is defined to be increasing in the counter-clockwise direction, while the boundary layer we have shown here increases in thickness as it traverses the annulus in the clockwise direction; we must therefore

define our advective velocity to be strictly negative in the $\hat{\theta}$ direction ($a_0 < 0$) and the leading edge of our boundary layer, θ_{LE}, must fall outside our domain and be larger than θ (i.e. $0 \leq \theta \leq \pi/2 < \theta_{LE}$).



(a) *Primal solution.* **(b)** *Adjoint solution.*

Figure 4-1: *Primal and adjoint solutions to volume output problem in the annulus domain using Cartesian coordinates solved adaptively with VMSD using* $p = 3$, $q = 3$ *and* $\approx 37,000$ *requested DOFs.*

4.2 Volume Output Problem

4.2.1 Volume Output and Adjoint

We utilize a volume output functional of the form,

$$\mathcal{J}(u) = \int_{\Omega} gu \, dV \tag{4.2}$$

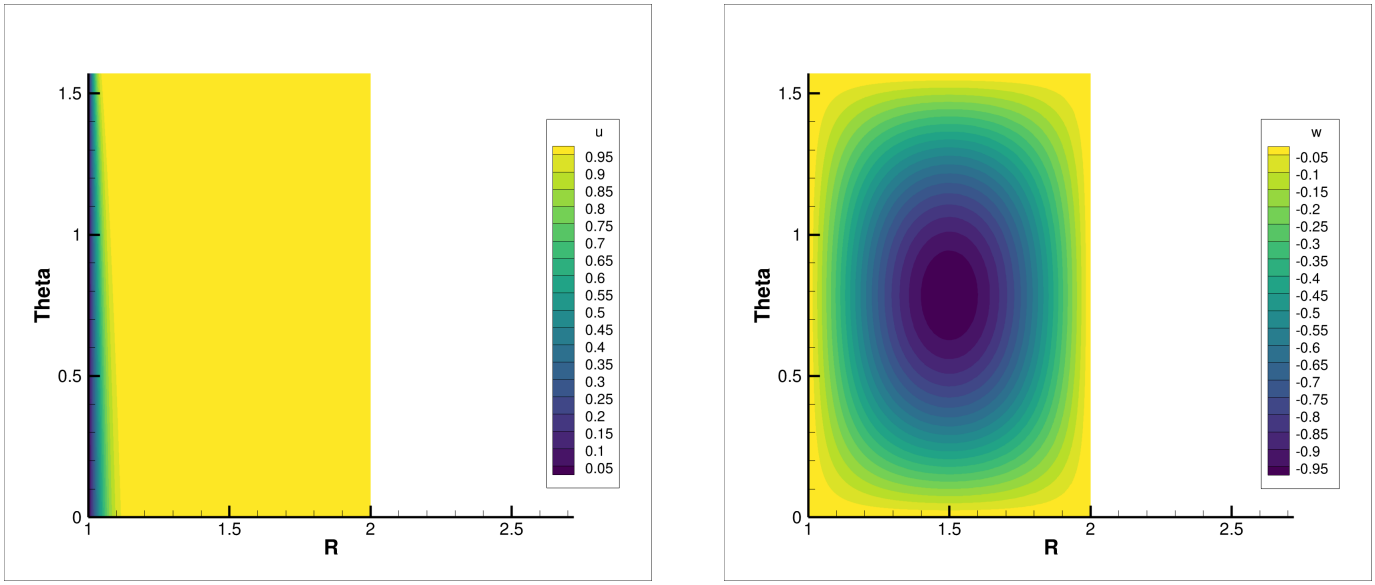
where g is calculated using MMS from the adjoint PDE.

For this problem, we utilize a product of sinusoids for the adjoint, satisfying the requirement of a homogenous solution along all four boundaries in the case of a volume output; this requirement comes from our adjoint analysis, Equation 3.18. The adjoint is given by,

$$w(r, \theta) = \sin(\pi r) \sin(2\theta) \tag{4.3}$$

This analysis is consistent between both polar and Cartesian coordinate systems, allowing us to simply

choose the form of (4.1) and (4.3) which corresponds to our coordinate system of choice. The adjoint can be seen in Figure 4-1b. While in Cartesian coordinates the computational domain is equivalent to the physical domain (e.g. the domain shown in Figure 4-1), in polar coordinates the domain is represented by a simple rectangle, as shown in Figure 4-2.



(a) *Primal solution.* **(b)** *Adjoint solution.*

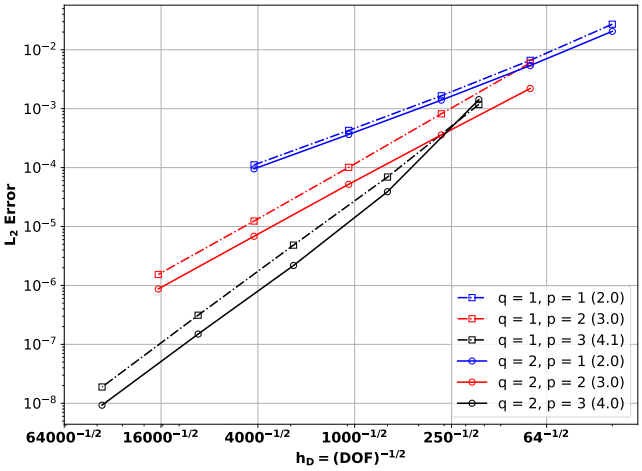
Figure 4-2: *Primal and adjoint solutions volume output problem in the annulus domain using polar coordinates solved adaptively with VMSD using* $p = 3, q = 1$ *and* $\approx 37,000$ *requested DOFs.*

With the choice of parameters $\mu = 1/2000$, $a_0 = -3/5$, $R = 1$, $\theta_{LE} = 3\pi/5$, we are able to create a boundary layer roughly 1/10 the thickness of the domain, resulting in an analytic output $\mathcal{J}(u) = -0.0038915838044611$.

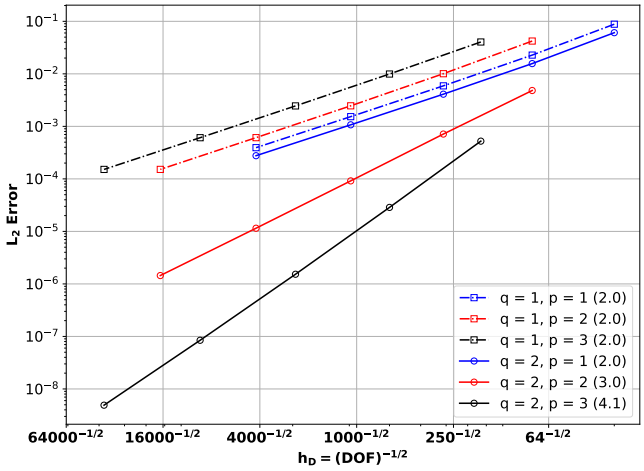
4.2.2 Impact of Grid Order

One of the major considerations of this work is the impact of the order, q, of the geometry itself when solving in Cartesian coordinates. To better understand this relationship, we consider solving the advection-diffusion equation using uniform refinement with the volume output introduced previously. Uniform refinement consists of first generating a structured quadrilateral mesh with elements of uniform size and shape, and then subdividing each element in half; the results of this study can be seen in Figure 4-3. While in this case, convergence of the primal L_2 is not affected by the grid order, we see that convergence of both the adjoint L_2 error and output error are severely hindered by the order of the geometry, with a failure to achieve superconvergence in the output. This issue has been observed in the context of adaptive finite element methods by Ursachi, and it was found that, for problems with curved geometries, the grid order

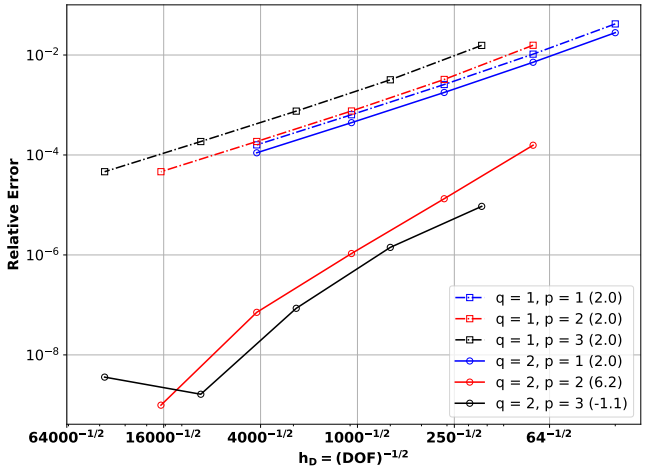
must be at least equivalent to the polynomial order, i.e. $q \geq p$ [54]. This is reflected by this study, as with $q = 2$, we are able to recover superconvergence of the output functional. Because of this, all Cartesian problems in the annulus domain are solved isoparametrically.



(a) *Primal L_2 error.*



(b) *Adjoint L_2 error.*



(c) *Relative output error.*

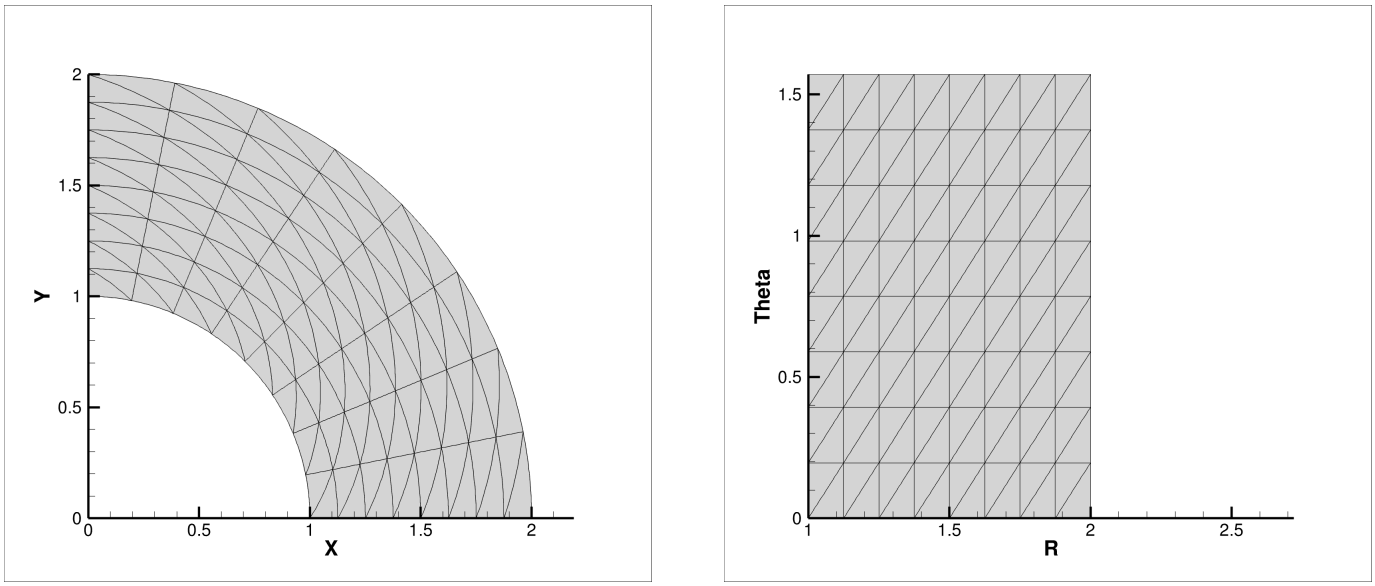
Figure 4-3: *Uniform refinement results for unstabilized CG discretization for grid orders $q = 1, 2$.*

4.2.3 Cartesian and Polar Comparison

To compare the use Cartesian and polar coordinates, we utilize VMSD to assess their performance relative to each other; VMSD consistently produced the lowest error per degree of freedom of any scheme tested, and thus, we use this as our "best-in-class" discretization.

We begin with a comparison of uniform refinement in both the Cartesian and polar coordinate systems. Representative uniform meshes in both Cartesian and polar coordinates can be seen in Figure 4-4. The

results can be seen in Figure 4-5. Unless otherwise stated, convergence data is shown with the results colored corresponding to polynomial order, with the approximate rates of convergence (calculated from the last two data points in each best-fit line) included as parenthetical data. Figures 4-5a and 4-5b show essentially identical performance in terms of L_2 error in resolving the both the primal and adjoint solutions for both coordinate systems. We would thus expect to see the same type of behavior in resolving the output between the two cases, which is reflected in Figure 4-5c. Relative output error is defined as $\frac{|\mathcal{J}_{\text{calc}} - \mathcal{J}_{\text{true}}|}{\mathcal{J}_{\text{true}}}$, where $\mathcal{J}_{\text{calc}}$ and $\mathcal{J}_{\text{true}}$ are the calculated and exact output values respectively.

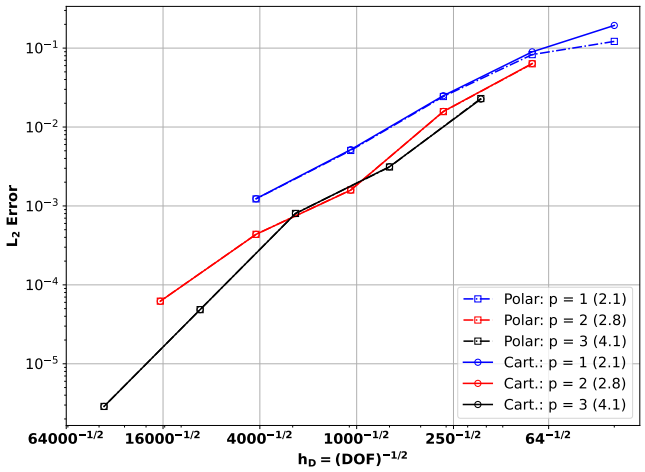


(a) *Cartesian mesh.* **(b)** *Polar mesh.*

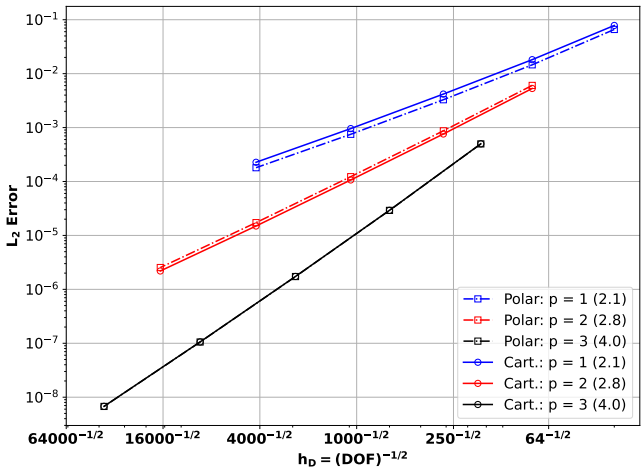
Figure 4-4: *Uniform meshes in both Cartesian and polar coordinates, using $q = 3$ and $q = 1$, respectively.*

We now move on to the primary goal of this work, a comparison of adaptive refinement using the MOESS algorithm between the two coordinate systems, the results of which can be seen in Figure 4-6. The MOESS algorithm works to minimize the error in the output functional, and thus, does not necessarily aim to achieve the same level of overall fidelity in either the primal or adjoint solution as the uniform refinement case, instead only refining in areas which have a significant impact on the output. The adjoint solution is fairly benign in this case, and refinement of the adjoint does not appear to have a significant impact on the output value, leading to higher L_2 error in the adjoint relative to the uniform refinement case, while the primal is resolved to a much higher fidelity. We see that, while adaptation in polar coordinates underperforms against its Cartesian counterpart when resolving the adjoint, it has higher performance not only in resolving the primal, but more importantly in resolving the output value.

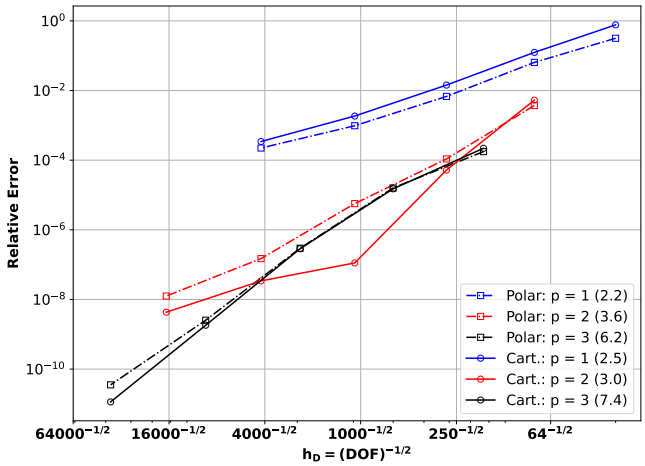
The reasoning behind the better performance of polar adaptivity relative to Cartesian likely stems from how curvature is embedded natively into the polar computational domain. Cartesian adaptation is only



(a) *Primal L_2 error.*



(b) *Adjoint L_2 error.*



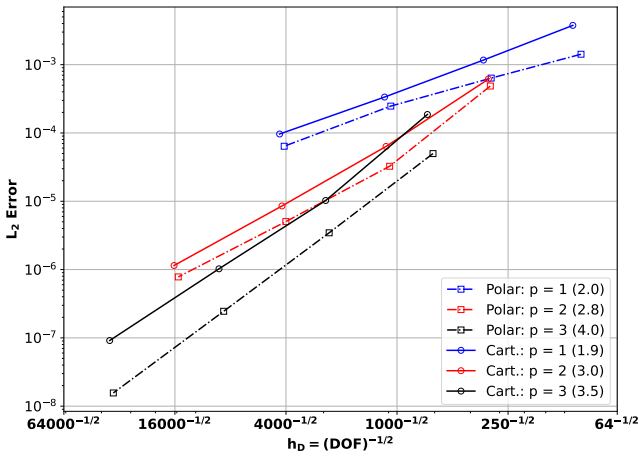
(c) *Relative output error.*

Figure 4-5: *Uniform refinement results for volume output problem using VMSD discretization in both Cartesian and polar coordinates.*

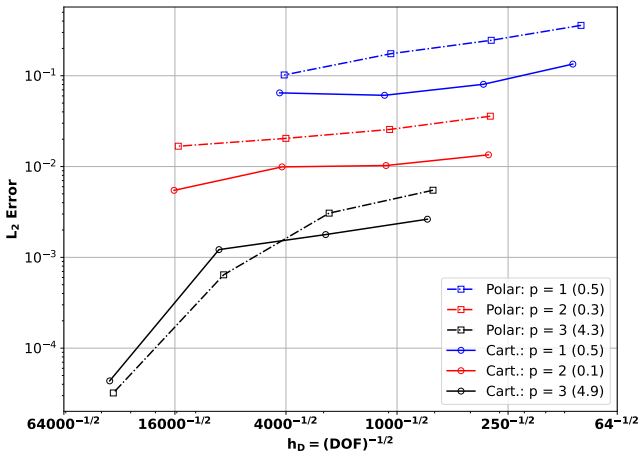
able to produce curved elements at orders $p = q > 1$, and the metric-conforming mesh generator EPIC only produces curvature to align with the boundaries of the domain; this curvature is then propagated inwards from the boundaries, which may or may not result in curved elements on the interior. Examples of $q = 1$ and $q = 3$ adaptively generated meshes can be seen in Figure 4-7. On the other hand, a $q = 1$ mesh in polar coordinates has a curved representation when transformed into Cartesian coordinates, as is illustrated by Figure 4-8. One major way this lack of interior curvature manifests is in the production of isotropic elements in the boundary layer.

When adaptively refining a mesh to capture a thin boundary layer, it is expected that there will be significant anisotropy (i.e. elements with a high aspect ratio) to capture the behavior of the boundary layer, with the potential for more isotropic elements away from the boundary layer. We see some anisotropy

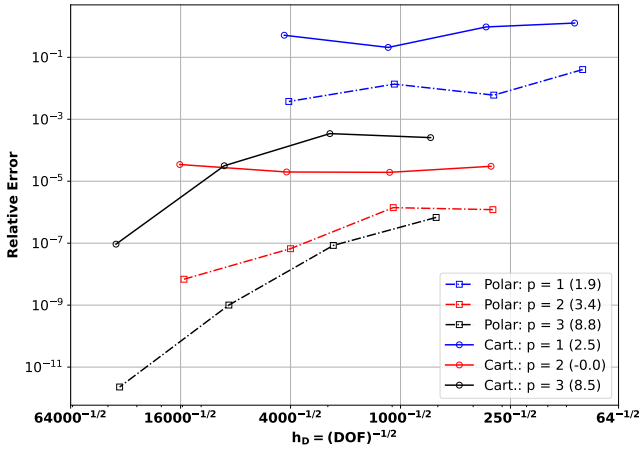
close to the wall in both the Cartesian and polar coordinate systems, as seen in Figure 4-9, along with isotropic elements in the interior of the domain. As the MOESS algorithm adaptively refines the mesh to resolve errors in both the primal and adjoint solutions, we see the relatively benign behavior from the adjoint resulting in lower anisotropy than we would typically expect. Despite this, we see a higher level of anisotropy in the polar coordinate system relative to the Cartesian as we move further from the wall.



(a) *Primal L_2 error.*



(b) *Adjoint L_2 error.*

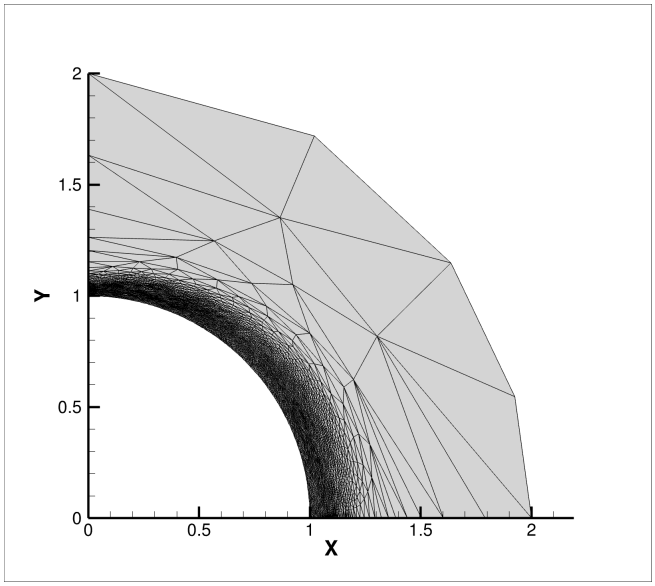


(c) *Relative output error.*

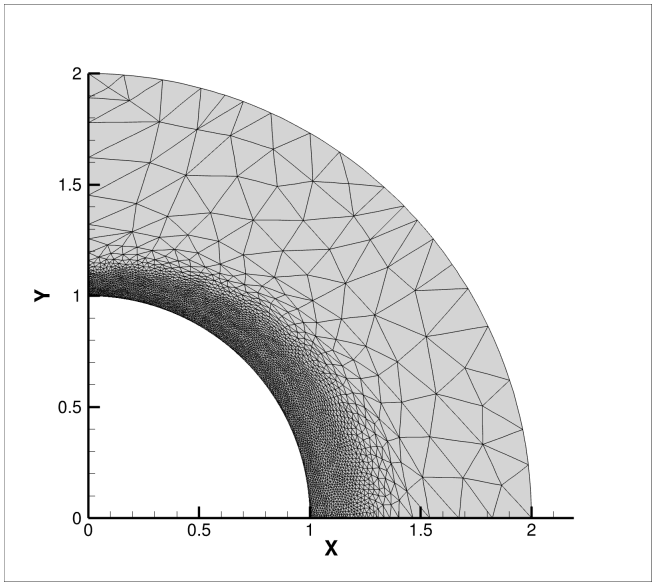
Figure 4-6: *Adaptive refinement results for volume output problem using VMSD discretization in both Cartesian and polar coordinates.*

4.2.4 Additional Discretizations

In addition to VMSD, both unstabilized CG and DG methods were tested, the results of which can be found in Figures 4-10 and 4-11 respectively. Both discretizations show extremely similar behavior to VMSD, with slightly higher levels of error on average. For CG, this is because there are no guarantees on



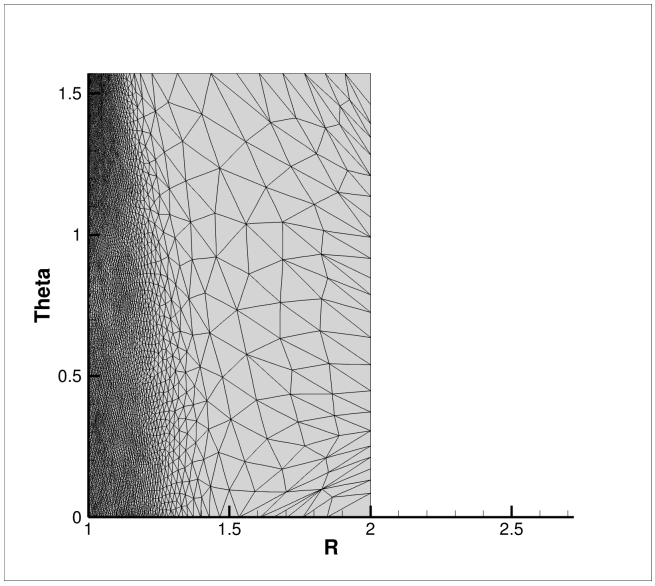
(a) $p = 1$, $q = 1$ *and* $\approx 4,000$ *requested DOFs.*



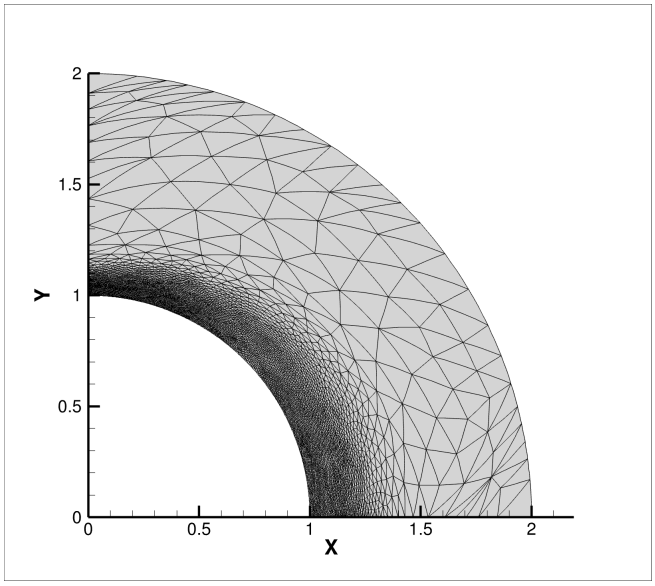
(b) $p = 3$, $q = 3$ *and* $\approx 37,000$ *requested DOFs.*

Figure 4-7: *Adaptive meshes generated for volume output problem in the annulus domain using Cartesian coordinates solved adaptively with VMSD.*

stability until the Peclet number is approximately equal to 1; in this case, since μ is small relative to the advective velocity, a Peclet number of 1 is not achieved in the range of degrees of freedom tested for CG. For DG, the higher error per degree of freedom can be attributed to the duplication of degrees of freedom across elements, meaning that a greater number of degrees of freedom is required to achieve the same level of accuracy as VMSD. We see that, for both polar CG and DG, despite small differences in the primal and adjoint L_2 errors, the additional curvature of the mesh leads to a significantly more accurate calculation of the output than the Cartesian case.

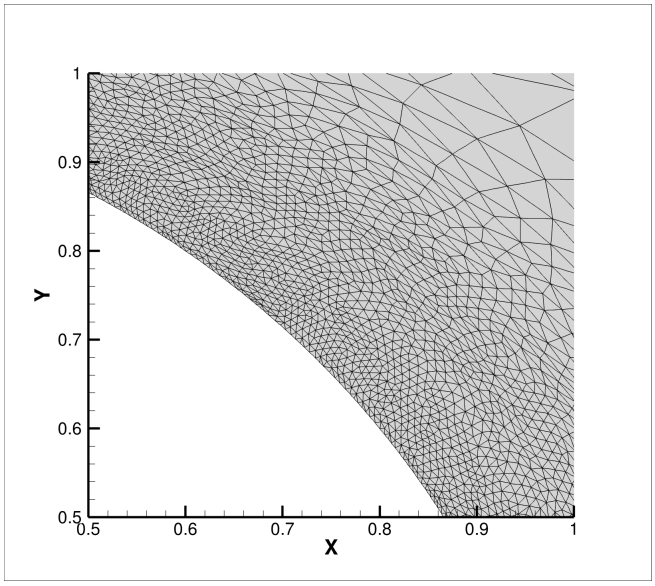


(a) *Polar computational domain* (r, θ).

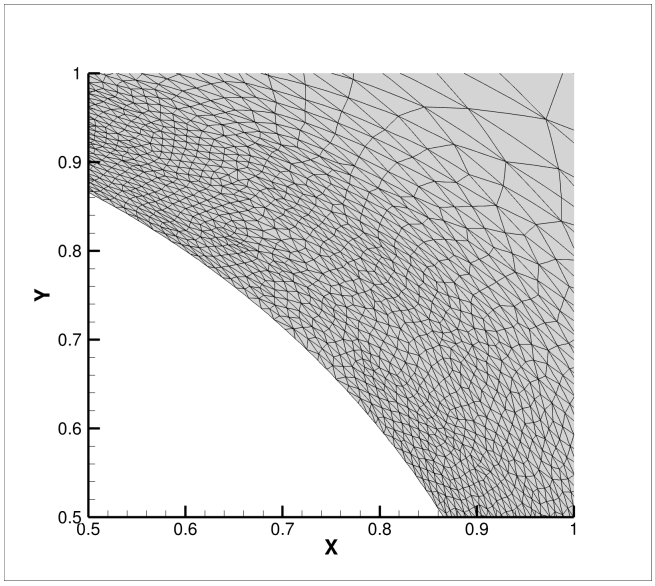


(b) *Physical domain* (x, y).

Figure 4-8: *Adaptive meshes generated for volume output problem in the annulus domain using polar coordinates solved adaptively with VMSD using* $p = 3$ *and* $\approx 37,000$ *requested DOFs.*

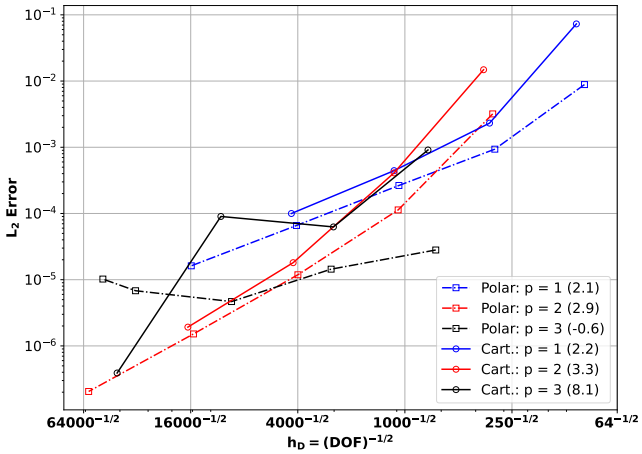


(a) *Cartesian mesh.*

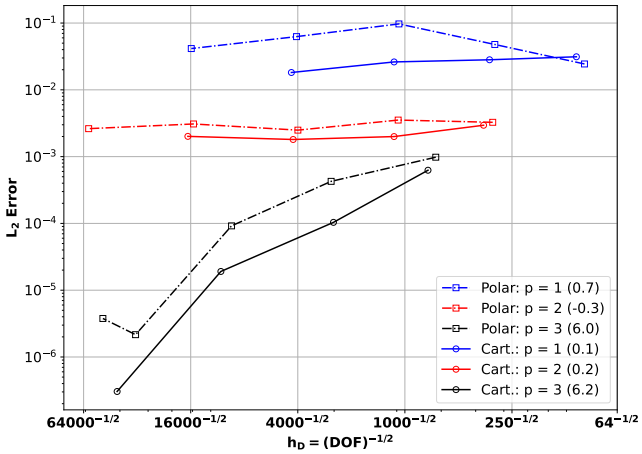


(b) *Polar mesh.*

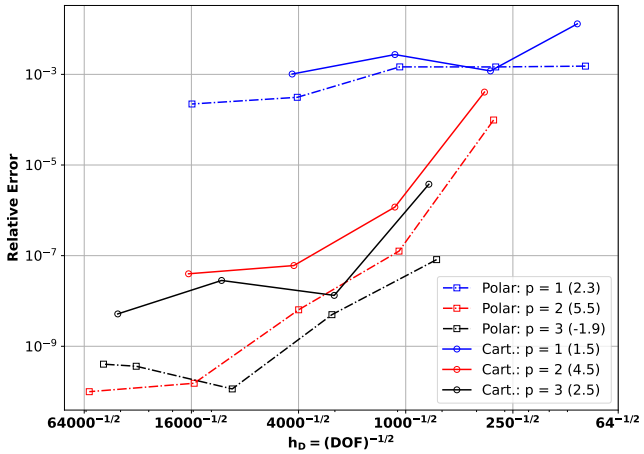
Figure 4-9: *Zoomed views of adaptive meshes generated for both Cartesian and polar volume output problems in the annulus domain solved adaptively with VMSD using* $p = 3$ *and* $\approx 37,000$ *requested DOFs.*



(a) *Primal L_2 error.*

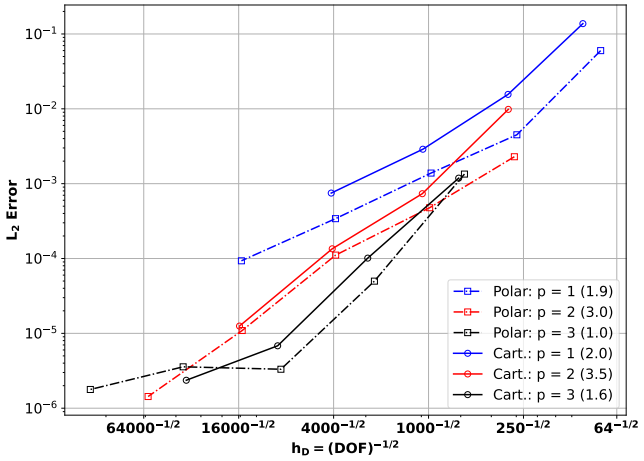


(b) *Adjoint L_2 error.*

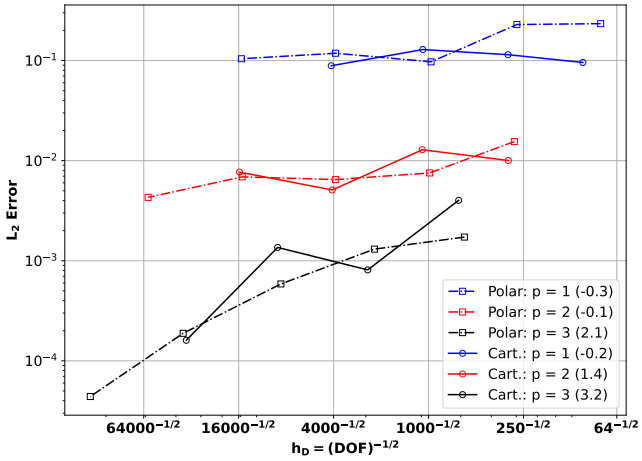


(c) *Relative output error.*

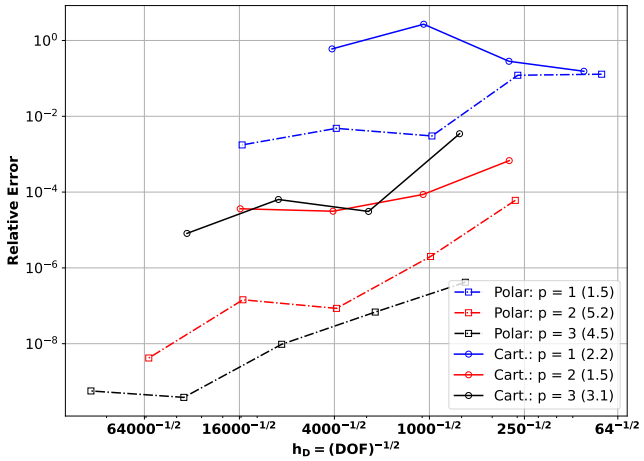
Figure 4-10: *Adaptive refinement results for volume output problem using unstabilized CG discretization in both Cartesian and polar coordinates.*



(a) *Primal L_2 error.*



(b) *Adjoint L_2 error.*



(c) *Relative output error.*

Figure 4-11: *Adaptive refinement results for volume output problem using DG discretization in both Cartesian and polar coordinates.*

4.3 Boundary Output Problem

To emulate a more realistic quantity of interest such as skin friction, we now aim to create a boundary output in the annulus domain. The exact solution to our primal PDE will again be (4.1), but we now require an adjoint which allows for a boundary output calculation to be performed.

4.3.1 Boundary Output and Adjoint

We now assume zero diffusion in the θ direction in both the primal and adjoint PDEs, which leads to a different primal forcing function, f, than in the volume output problem. We again use MMS to derive f. Without this assumption, obtaining an analytic adjoint using separation of variables would not be possible for this problem; the additional factors of r present in the gradient and divergence operators in polar coordinates render it impossible to break the adjoint solution two independent functions. Having an analytic adjoint solution for this problem is critical, as without it, we would have a non-zero adjoint forcing term, g. From Equation 3.31, we know that a non-zero g corresponds to a primal volume output functional, and thus, g must be zero to ensure a boundary output problem. We begin by deriving an analytic solution to our modified version of the advection-diffusion PDE, in which there is zero θ diffusion,

$$-\frac{a_\theta}{r}\frac{\partial w}{\partial \theta} - \frac{\mu}{r}\frac{\partial}{\partial r}\left(r\frac{\partial w}{\partial r}\right) = 0 \tag{4.4}$$

We now break the adjoint into $w(r,z) = R(r)\Theta(\theta)$,

$$-\frac{a_\theta R(r)\Theta'(\theta) + \mu\Theta(\theta)(R'(r) + rR''(r))}{r} = -\mu\frac{R'(r)}{R(r)} - \mu r\frac{R''(r)}{R(r)} - a_\theta\frac{\Theta'(\theta)}{\Theta(\theta)} \tag{4.5}$$

We will treat this now as two independent ODEs,

$$-\mu\frac{R'(r)}{R(r)} - \mu r\frac{R''(r)}{R(r)} = -c^2 \tag{4.6}$$

$$-a_\theta\frac{\Theta'(\theta)}{\Theta(\theta)} = c^2 \tag{4.7}$$

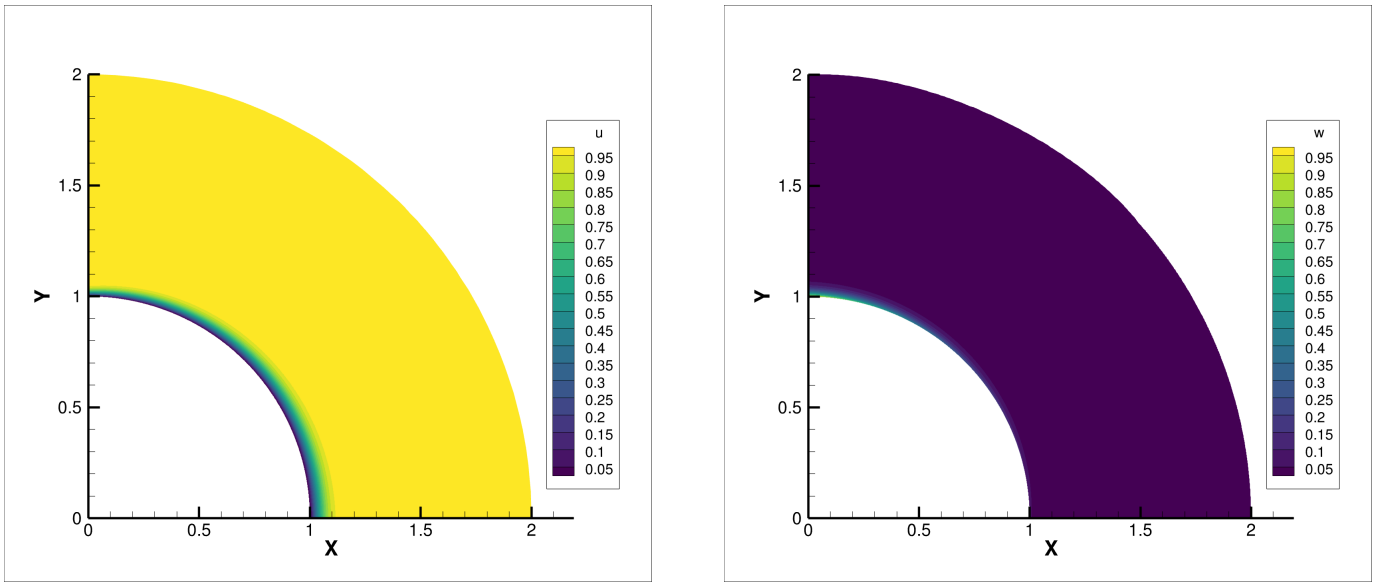
Since one is a function of r alone and the other a function of θ alone, the only possible solution is if both are equal to a constant. We first solve for $R(r)$,

$$-\mu\frac{R'(r)}{R(r)} - \mu r\frac{R''(r)}{R(r)} = -c^2, \quad R(r_0) = R_0, \quad R(r_1) = 0$$
$$R(r) = R_0 \left(I_0\left(\frac{2c\sqrt{r_1}}{\sqrt{\mu}}\right) K_0\left(\frac{2c\sqrt{r}}{\sqrt{\mu}}\right) - I_0\left(\frac{2c\sqrt{r}}{\sqrt{\mu}}\right) K_0\left(\frac{2c\sqrt{r_1}}{\sqrt{\mu}}\right)\right) \Bigg/ \left(I_0\left(\frac{2c\sqrt{r_1}}{\sqrt{\mu}}\right) K_0\left(\frac{2c\sqrt{r_0}}{\sqrt{\mu}}\right) - I_0\left(\frac{2c\sqrt{r_0}}{\sqrt{\mu}}\right) K_0\left(\frac{2c\sqrt{r_1}}{\sqrt{\mu}}\right)\right) \tag{4.8}$$

Where $I_0(r)$ and $K_0(r)$ correspond to modified Bessel functions of the 1st and 2nd kind, respectively. We now solve for $\Theta(\theta)$, noting that we are only able to impose one BC on our solution because this is now a first-order ODE,

$$-a_\theta\frac{\Theta'(\theta)}{\Theta(\theta)} = c^2, \quad \Theta\left(\frac{\pi}{2}\right) = 1$$
$$\Theta(\theta) = e^{\frac{c^2\pi}{2a_\theta} - \frac{c^2\theta}{a_\theta}} \tag{4.9}$$

Our analytic adjoint is given by the product $w(r,z) = R(r)\Theta(\theta)$, seen in Figure 4-12.



(a) *Primal solution.* **(b)** *Adjoint solution.*

Figure 4-12: *Primal and adjoint solutions to boundary output problem in the annulus domain using Cartesian coordinates solved adaptively with VMSD using* $p = 3$, $q = 3$ *and* $\approx 37,000$ *requested DOFs.*

For the Cartesian implementation of the PDE, the advective terms are calculated through the transformations outlined at the beginning of Section 4.2. However, the diffusive terms are more nuanced, as zero diffusion in the $\hat{\theta}$ direction does not correspond to zero diffusion in a single Cartesian coordinate direction.

We can no longer treat this as a scalar coefficient of diffusion, μ, instead utilizing a viscous tensor,

$$\mathcal{K}(x,y) = \begin{pmatrix} k_{xx}(x,y) & k_{xy}(x,y) \\ k_{yx}(x,y) & k_{yy}(x,y) \end{pmatrix} \propto \begin{pmatrix} k_{rr} & k_{r\theta} \\ k_{\theta r} & k_{\theta\theta} \end{pmatrix} \tag{4.10}$$

The polar and Cartesian unit vectors are related by,

$$\hat{r} = \hat{i}\cos(\theta) + \hat{j}\sin(\theta) \quad \hat{\theta} = -\hat{i}\sin(\theta) + \hat{j}\cos(\theta) \tag{4.11}$$

$$\hat{i} = \hat{r}\cos(\theta) - \hat{\theta}\sin(\theta) \quad \hat{j} = \hat{r}\sin(\theta) + \hat{\theta}\cos(\theta) \tag{4.12}$$

Let $c = \cos(\theta)$, $s = \sin(\theta)$. Using dyadics, we can transform our polar diffusivity tensor into Cartesian coordinates,

$$\begin{aligned}
\mathcal{K} &= \hat{r}\hat{r}\,k_{rr} + \hat{r}\hat{\theta}\,k_{r\theta} + \hat{\theta}\hat{r}\,k_{\theta r} + \hat{\theta}\hat{\theta}\,k_{\theta\theta} \\
&= (\hat{i}c + \hat{j}s)(\hat{i}c + \hat{j}s)k_{rr} + (\hat{i}c + \hat{j}s)(-\hat{i}s + \hat{j}c)k_{r\theta} \\
&\quad + (-\hat{i}s + \hat{j}c)(\hat{i}c + \hat{j}s)k_{\theta r} + (-\hat{i}s + \hat{j}c)(-\hat{i}s + \hat{j}c)k_{\theta\theta} \\
&= (\hat{i}\hat{i}c^2 + \hat{i}\hat{j}cs + \hat{j}\hat{i}cs + \hat{j}\hat{j}s^2)k_{rr} + (-\hat{i}\hat{i}cs + \hat{i}\hat{j}c^2 - \hat{j}\hat{i}s^2 + \hat{j}\hat{j}cs)k_{r\theta} \\
&\quad + (-\hat{i}\hat{i}cs - \hat{i}\hat{j}s^2 + \hat{j}\hat{i}c^2 + \hat{j}\hat{j}cs)k_{\theta r} + (\hat{i}\hat{i}s^2 - \hat{i}\hat{j}cs - \hat{j}\hat{i}cs + \hat{j}\hat{j}c^2)k_{\theta\theta} \\
&= \hat{i}\hat{i}(c^2k_{rr} - cs(k_{r\theta} + k_{\theta r}) + s^2k_{\theta\theta}) + \hat{i}\hat{j}(cs(k_{rr} - k_{\theta\theta}) + c^2k_{r\theta} - s^2k_{\theta r}) \\
&\quad + \hat{j}\hat{i}(cs(k_{rr} - k_{\theta\theta}) - s^2k_{r\theta} + c^2k_{\theta r}) + \hat{j}\hat{j}(s^2k_{rr} + cs(k_{r\theta} + k_{\theta r}) + c^2k_{\theta\theta}) \\
&\equiv \hat{i}\hat{i}k_{xx} + \hat{i}\hat{j}k_{xy} + \hat{j}\hat{i}k_{yx} + \hat{j}\hat{j}k_{yy}
\end{aligned} \tag{4.13}$$

Finally, our boundary functional is given by,

$$\begin{aligned}
\mathcal{J}(u) = \oint_\Gamma h(\vec{a}u - \mathcal{K}\cdot\nabla u)\cdot\hat{n}\,dS &= \oint_{\theta_0}^{\theta_1} (-rw\mathcal{K}\cdot\nabla u)|_{r_0}\cdot(-\hat{r})\,d\theta \\
&\quad + \oint_{r_0}^{r_1} (w(\vec{a}u - \mathcal{K}\cdot\nabla u))|_{\theta_0}\cdot(-\hat{\theta})\,dr \\
&\quad + \oint_{r_0}^{r_1} (w(\vec{a}u - \mathcal{K}\cdot\nabla u))|_{\theta_1}\cdot(\hat{\theta})\,dr
\end{aligned} \tag{4.14}$$

With the choice of parameters $\mu = 1/2000$, $a_0 = -3/5$, $c = 1$, $r_0 = 1\,, r_1 = 2$, $R_0 = 1$, our analytic output is given by $\mathcal{J}(u) = -0.0000392729413449$.

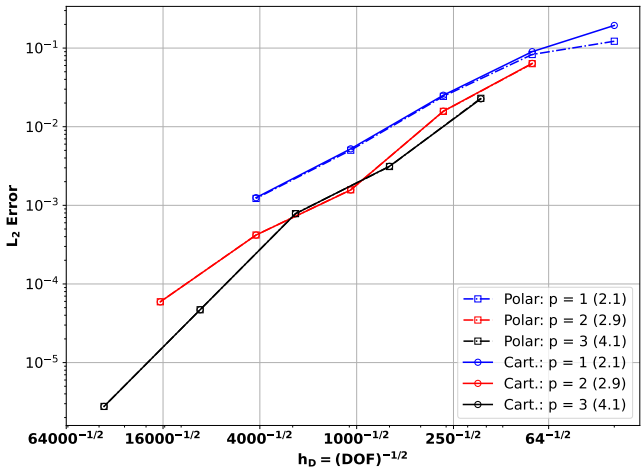
4.3.2 Cartesian and Polar Comparison

As in the case of the volume output, we will primarily discuss VMSD, as it consistently outperformed all other discretizations which were tested. We begin our analysis of the boundary output through a comparison of uniform refinement between the two coordinate systems, shown in Figure 4-13. The results show once more that when utilizing uniform refinement, the convergence of the L_2 errors as well as the output functional are nearly identical. While small variations occur, likely due to the reasons discussed in Section 4.2.3, the differences in curvature is limited in effect, as the meshes are almost equivalent.

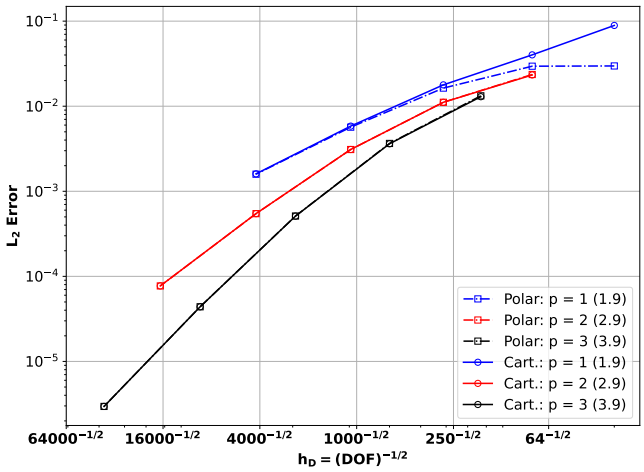
The adaptive results tell a different story, however. For this boundary output case, both the primal and adjoint solutions have nearly the same boundary layer behavior along the inner radius of the annulus and have nearly constant flow fields away from the inner wall. Thus, resolving either the primal or the adjoint solution will result in the other solution being resolved as well, which was not the case when for the volume output. We see that polar adaptation continues to outperform Cartesian adaptation in terms of primal L_2 and output error, and now outperforms the Cartesian case when resolving the adjoint too. The relative performance in resolving the output error is perhaps the most striking, as the polar case outperforms the Cartesian case by several orders of magnitude of accuracy for $p = 1$ and $p = 2$. For $p = 3$, the Cartesian case struggles to converge. This is likely due to a bug in the way the adaptive algorithm handles curved elements, and a discussion of this can be seen in Appendix A. Even accounting for the subpar convergence of the $p = 3$ adaptive Cartesian results, we still see several orders of magnitude improvement in terms of output error when considering the polar case.

Because the adjoint is also zero, or nearly zero, away from the inner wall, accurate calculation of the boundary output will essentially only demand high resolution along the inner wall, and this behavior is observed for both the Cartesian and polar coordinates systems. However, the Cartesian mesh is not able to achieve near the same level of anisotropy as the polar mesh. This is likely part of the reason behind why the Cartesian case struggled to resolve the output to the same precision as the polar case.

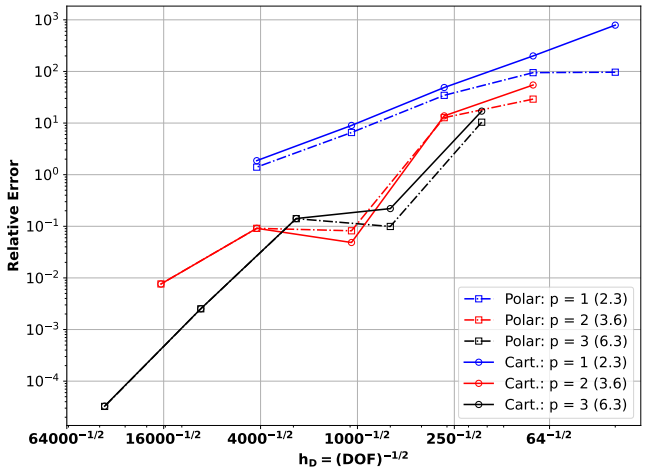
These results offer two main conclusions. First, these results indicate the potential advantages of modeling the impact of curvature on error estimation, which is discussed in Section 5.2. Second, they demonstrate the need for a metric-conforming mesh generator capable of generating curved elements to be able to fully leverage the advantages of higher-order finite element methods, namely, superconvergence of the output functional. Without the ability to produce highly anisotropic, curved elements, as were shown to be extremely effective from the polar, mesh-based mapping, accuracy will be limited when considering problems in curved domains.



(a) *Primal L_2 error.*



(b) *Adjoint L_2 error.*

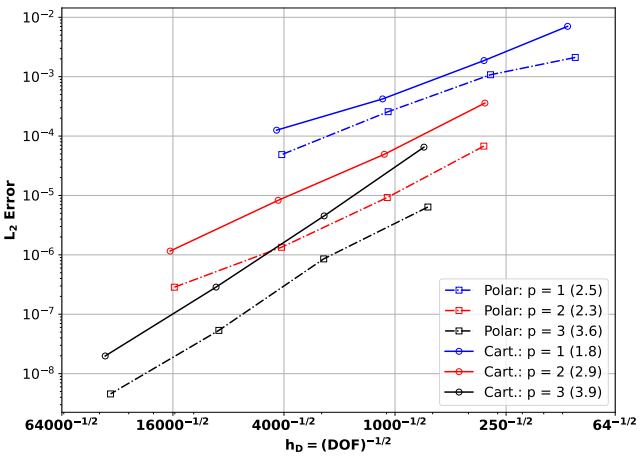


(c) *Relative output error.*

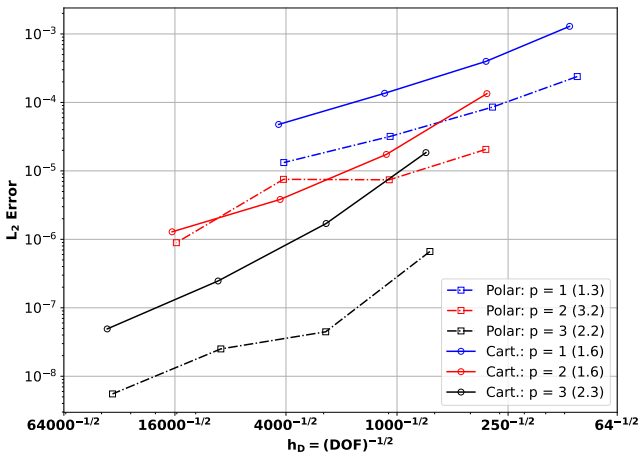
Figure 4-13: *Uniform refinement results for boundary output problem using VMSD discretization in both Cartesian and polar coordinates.*

4.3.3 Additional Discretizations

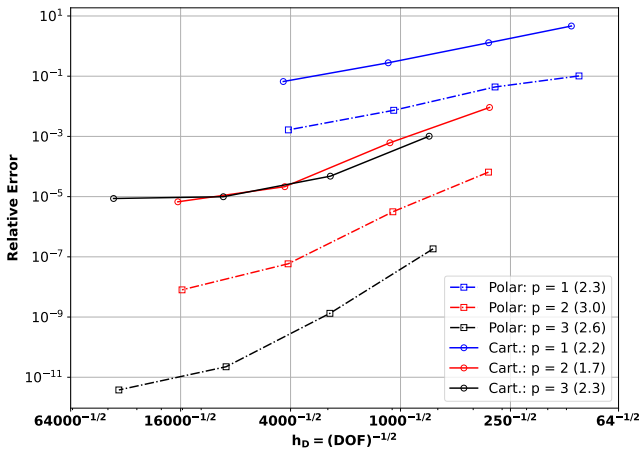
In addition to VMSD, both unstabilized CG and DG methods were tested, the results of which can be found in Figures 4-17 and 4-18 respectively. As was the case with VMSD, we see that both CG and DG are able to resolve both the primal and adjoint solutions more accurately in the polar case than in the Cartesian. While the Cartesian case sees convergence similar behavior to VMSD, the DG polar discretization is only limited by the number of degrees of freedom; the CG discretization on the other hand has no guarantees on stability, and it appears that the polar discretization encounters either a stability issue or a zero crossing, causing the error to spike for the final iteration.



(a) *Primal L_2 error.*

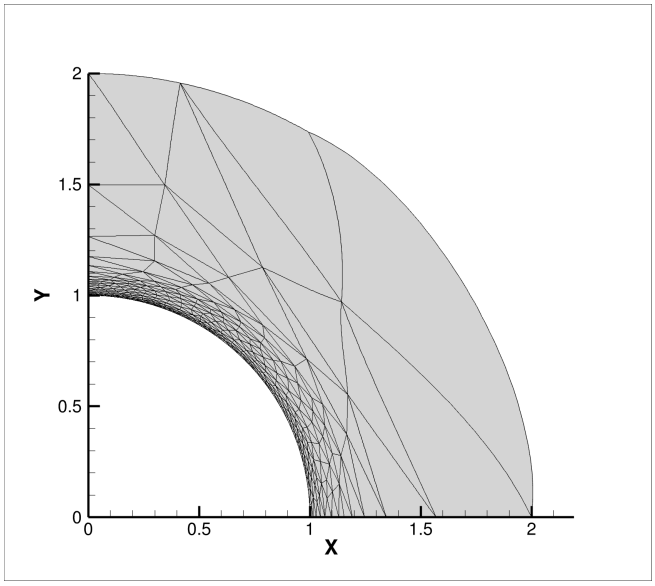


(b) *Adjoint L_2 error.*

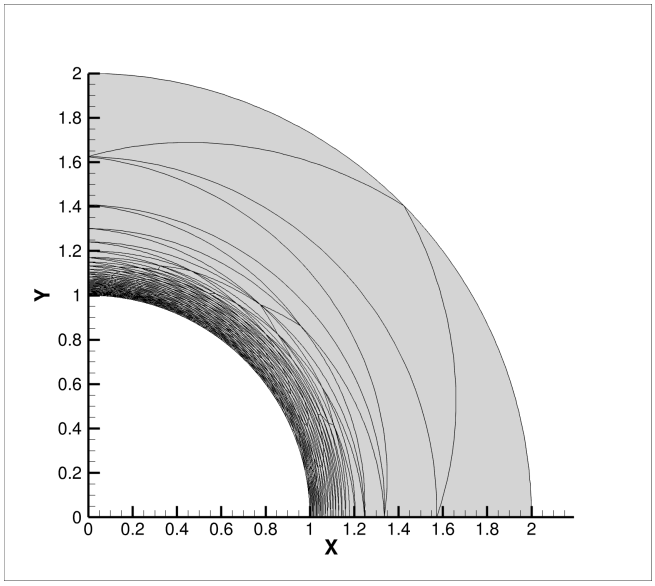


(c) *Relative output error.*

Figure 4-14: *Adaptive refinement results for boundary output problem using VMSD discretization in both Cartesian and polar coordinates.*

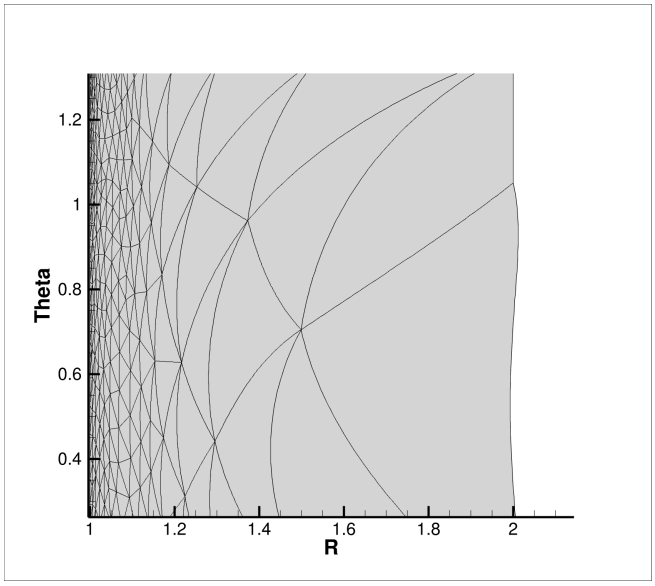


(a) *Cartesian mesh.*

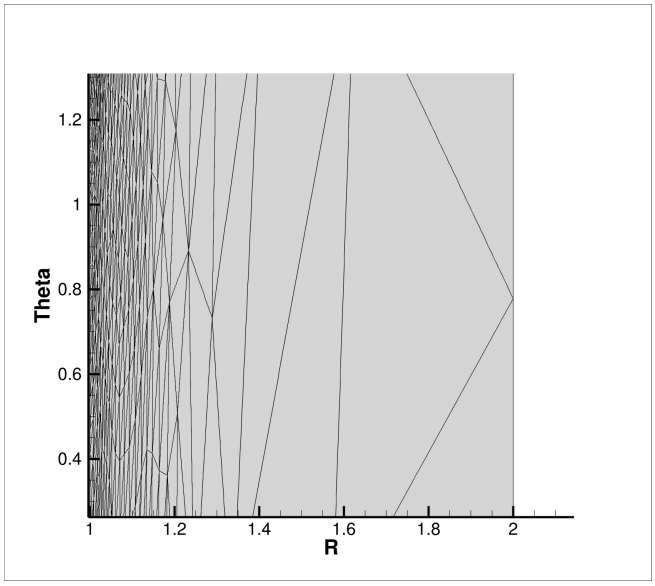


(b) *Polar mesh.*

Figure 4-15: *Adaptive meshes generated for both Cartesian and polar boundary output problems in the annulus domain solved adaptively with VMSD using* $p = 3$ *and* $\approx 2,400$ *requested DOFs.*

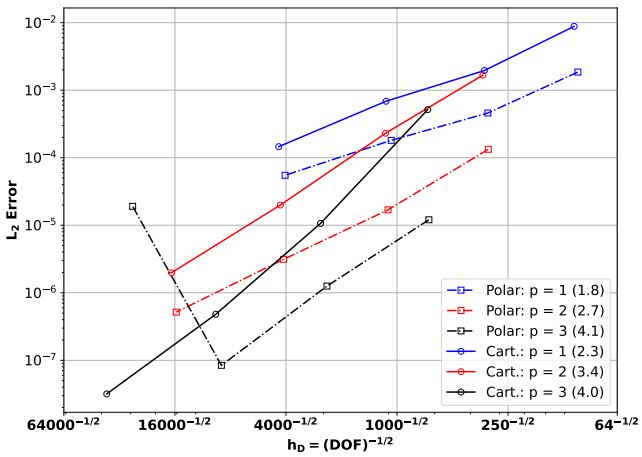


(a) *Cartesian mesh.*

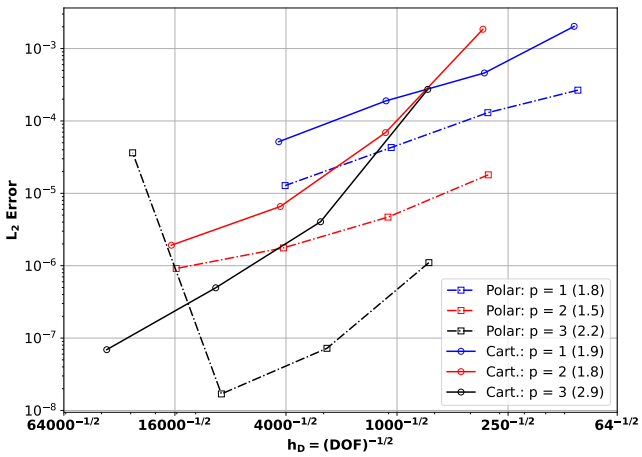


(b) *Polar mesh.*

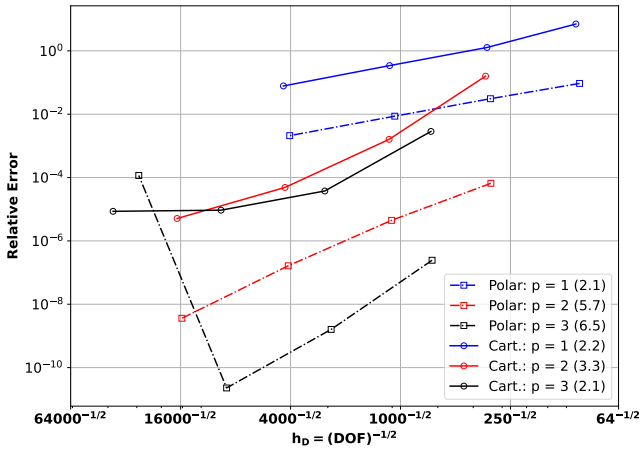
Figure 4-16: *Cartesian and polar meshes from Figure 4-15 converted into* (r, θ) *coordinates. Meshes have been zoomed in to highlight anisotropy near the wall.*



(a) *Primal L_2 error.*

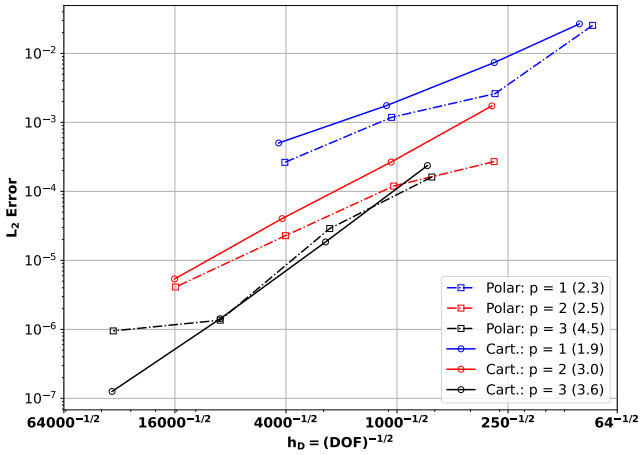


(b) *Adjoint L_2 error.*

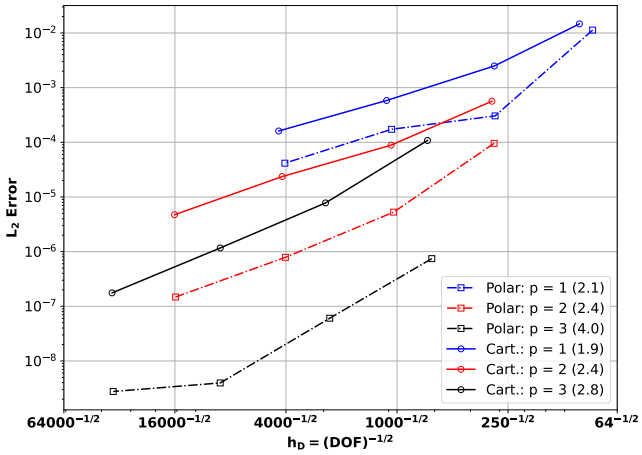


(c) *Relative output error.*

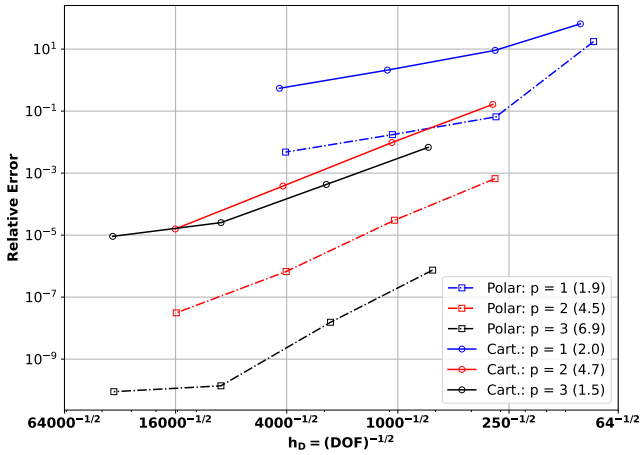
Figure 4-17: *Adaptive refinement results for boundary output problem using unstabilized CG discretization in both Cartesian and polar coordinates.*



(a) *Primal L_2 error.*



(b) *Adjoint L_2 error.*



(c) *Relative output error.*

Figure 4-18: *Adaptive refinement results for boundary output problem using DG discretization in both Cartesian and polar coordinates.*

Chapter 5

Conclusion

5.1 Summary

In this thesis, the effects of mesh curvature on higher-order methods and mesh adaptation have been investigated for the advection-diffusion equation in a curved domain. In particular, the results of this investigation demonstrate the potential for mesh curvature to control error. For each polynomial order, discretization, and output functional tested, solving the advection-diffusion equation in a polar coordinate system to introduce appropriate element curving achieved significantly higher levels of accuracy in computing output quantities of interest. These results also showcase the potential improvements over existing higher-order, adaptive finite element methods which are possible with the use of a metric-conforming mesh generator capable of generating curved elements.

In addition to this, adjoint analysis performed in this work demonstrated how the form of the primal output functional, which is assumed to be a general, combined volume-boundary output, affects the adjoint PDE and BCs; these results allow us to construct analytic adjoint solutions with desirable properties (e.g. boundary layer like behavior). Analytic primal and adjoint solutions for both volume and boundary output functionals in Cartesian and polar coordinate systems were derived, providing a framework through which to test the performance of future adaptive methods for curved mesh generation.

5.2 Future Work

In addition to resolving this issues discussed in Appendix A, the most clear extension of this work is the development of an adaptive process which utilizes mesh curvature to control error. One could envision the

inclusion of mesh curvature into the MOESS edge-splitting process. i.e. instead of merely splitting each edge when performing the local error sampling process, the effects of different amounts of curvature could be considered as well. This would allow for the algorithm to produce curved elements that provide better approximation spaces than is possible with straight element shapes. The test cases and analytic solutions presented in this work can be used to aid in the development of such an adaptive process, as it provides a consistent methodology and results with which future methods can be compared.

Another extension of this work is to investigate the potential of incorporating the misrepresentation of domain geometry into the error estimation process. Section 4.2.2 showcases the impact of the geometry order on the convergence of the output quantity of interest, illustrating the limitations that exist when the geometry is not accurately represented by the mesh. If there was instead a quantification for this type of error, we may see better performance in the case of adaptive meshing. For example, in the case of utilizing $q = 1$ elements in the annulus domain, there is the potential to improve the performance of the adaptive process if it were to account for errors in the representation of the curved boundaries with linear elements. This could also be incorporated into the previous suggestion for accounting for curvature as part of the mesh adaptation process, allowing for higher order elements to conform to the geometry and for that curvature to exist throughout the domain, as opposed to the process utilized in this work (forcing the mesh to match the boundaries of the geometry and propagating that curvature into the interior of the domain).

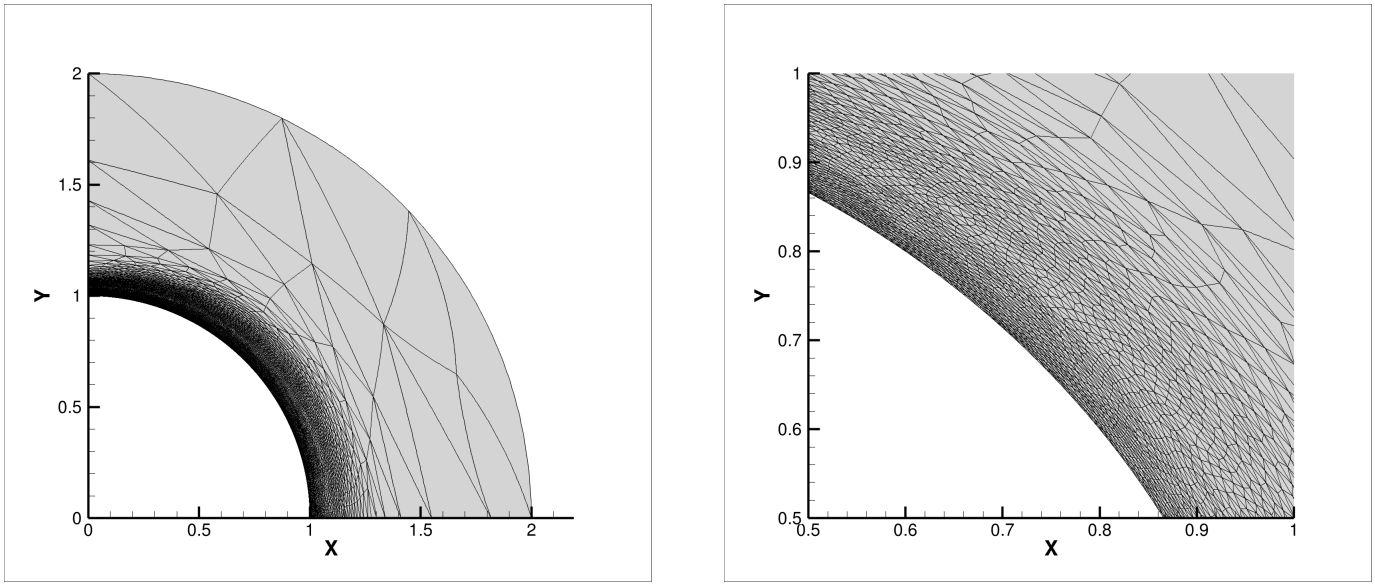
Appendix A

Adaptivity Issues

This appendix will briefly discuss the issues faced in performing adaptation on the $q = 3$ annulus mesh in the case of the boundary output problem, offering potential reasons for these issues as well as a discussion of future work towards resolving this issue.

A.1 Low Viscosity Boundary Output

The results for the low viscosity boundary output problem solved with VMSD on the finest mesh after 40 adaptive iterations can be seen in Figure A-1. Recalling the results shown in Section 4.3.2, we know the error in the computed output for the Cartesian case to be significantly higher than expected, with the error appearing to plateau at a magnitude of 10^{-5}, while its polar counterpart approaches machine precision. It is difficult to diagnose the issue from this mesh alone, but we are able to zoom in to better see the behavior of elements on the boundary, as seen in Figure A-1b. It is still difficult to see exactly what is occurring, but we speculate that there is too much refinement occurring on this inner boundary; i.e. there are too many elements being placed along this boundary. However, the boundary layer in this problem is quite thin, which may be contributing to this concentration of elements.



(a) *Final low viscosity Cartesian mesh.* **(b)** *Zoomed view.*

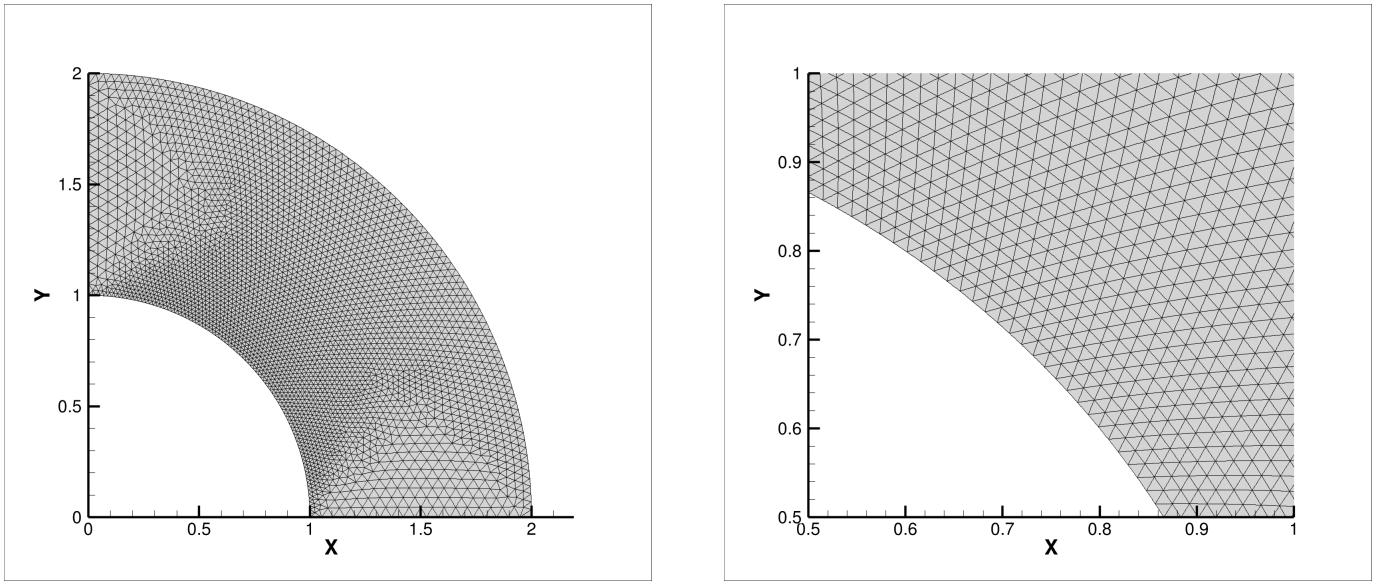
Figure A-1: *Adaptive mesh generated for the Cartesian boundary output problem with low viscosity in the annulus domain solved adaptively with VMSD using* $p = 3$ *and* $\approx 37,000$ *requested DOFs.*

A.2 High Viscosity Boundary Output

To investigate this issue further, we increase the viscosity, thereby increasing the thickness of the boundary layer significantly; with this additional thickness, we would expect to see far fewer elements concentrated along that inner boundary. We begin our investigation by analyzing the behavior of the mesh at each adaptive iteration, noting any major deviations from the expected behavior. Figure A-2 shows the initial Cartesian mesh for context, and Figures A-3, A-4, and A-5 show the mesh after 2, 3, and 40 (the maximum number) of adaptive iterations respectively. We see that, after only 2 adaptive iterations, the concentration of the elements along the inner boundary is far greater than expected, with an even greater concentration elements being seen after the 3rd iteration. By iteration 40, the elements are extremely tightly concentrated along the inner wall, and the elements themselves are small, without a high degree of anisotropy. While not as apparent in the earlier iterations, we also see a higher than expected level of refinement along the outer boundary where the boundary layer is smoothly varying; we would expect a similar level of refinement along this boundary as is present in the center of the domain because of this. These two behaviors point towards the curvature on these boundaries as a potential source of the errors occurring in the adaptation process.

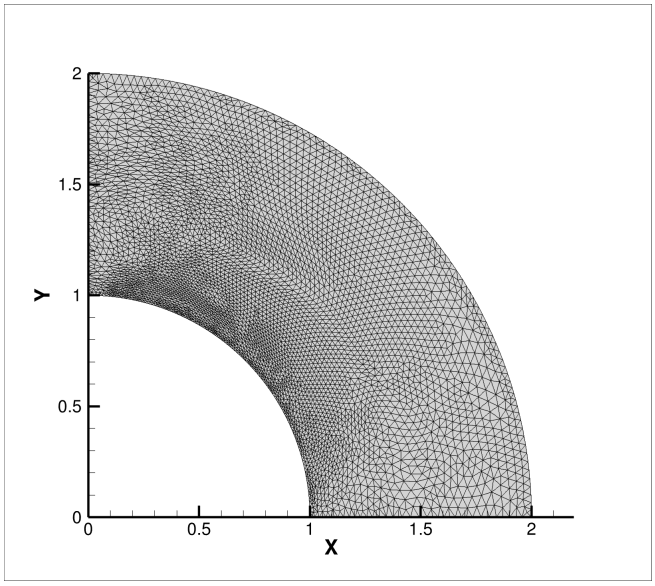
Another approach taken to investigate this issue was to compare the metric request to the metric implied by the mesh produced from the request. These metrics appeared consistent in our analysis. Because of

this evidence, we believe that the issue most likely lies somewhere in the adaptive process itself, perhaps within the error modeling or local sampling process. One place this could be occurring is in the local edge-splitting process. The issue is only seen on higher-order, specifically $q = 3$, meshes, and may stem from how nodes on the interior of a curved element are positioned on the interior of the curved sub-elements post-edge-splitting. However, this commentary is speculative, and further investigation is required to bring finality to this discussion.

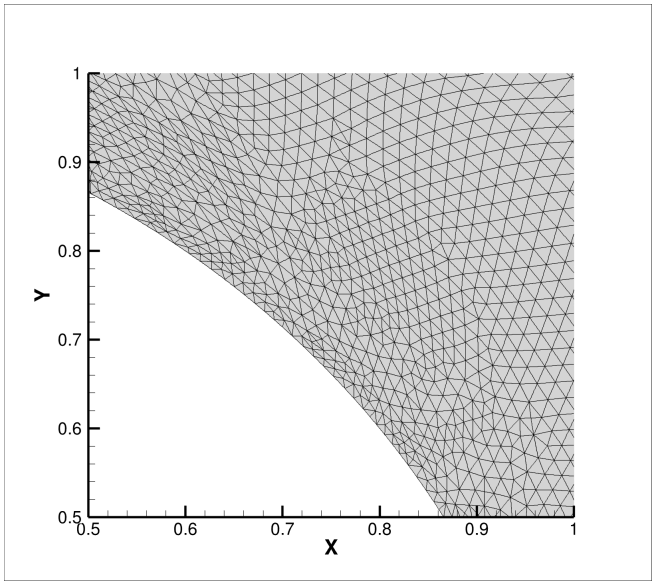


(a) *Initial Cartesian mesh.* **(b)** *Zoomed view.*

Figure A-2: *Initial Cartesian mesh for VMSD case with* $p = 3$ *and* $\approx 37,000$ *requested DOFs.*

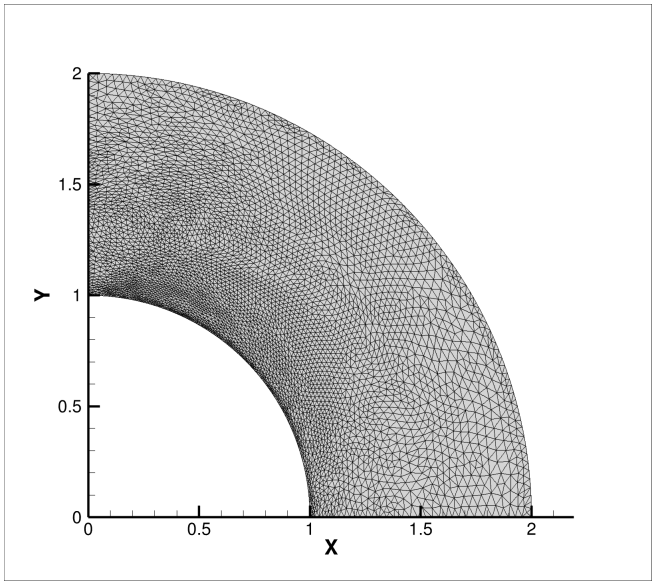


(a) *Cartesian mesh after 2 adaptive iterations.*

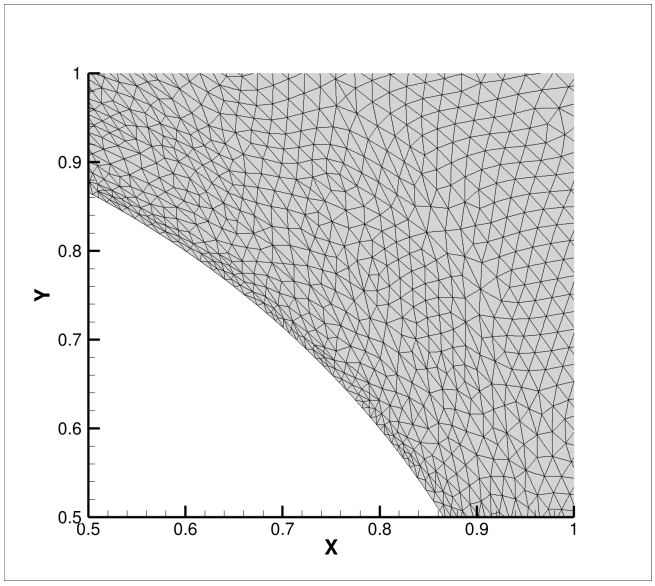


(b) *Zoomed view.*

Figure A-3: *Cartesian mesh for VMSD case after 2 adaptive iterations with* $p = 3$ *and* $\approx 37,000$ *requested DOFs.*

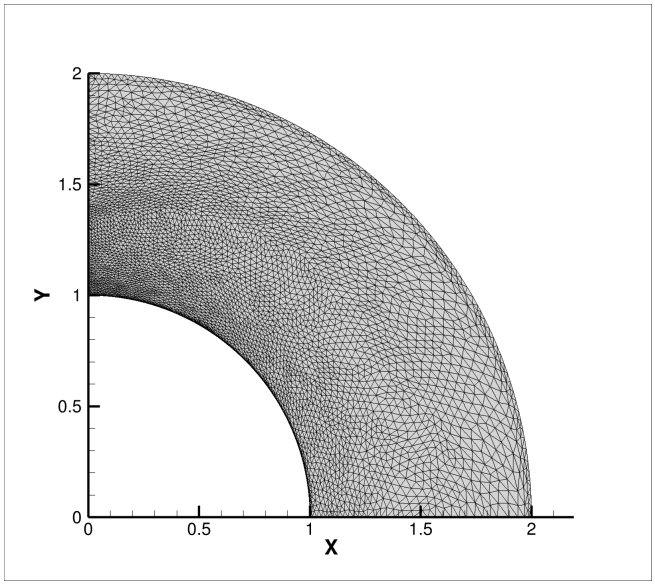


(a) *Cartesian mesh after 3 adaptive iterations.*

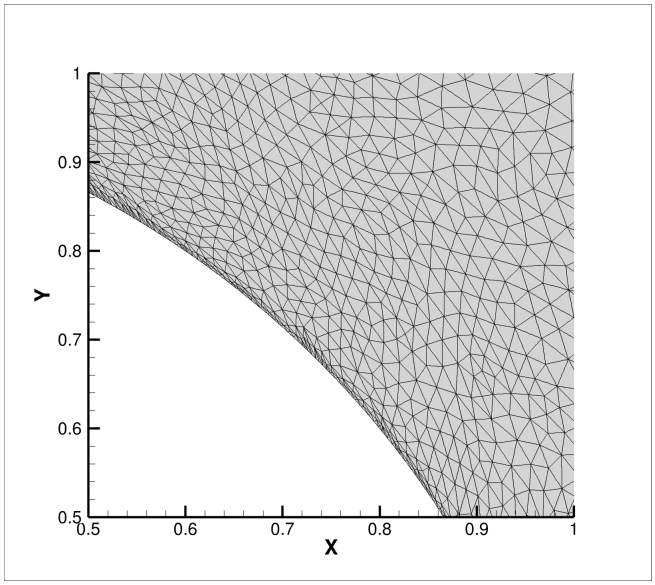


(b) *Zoomed view.*

Figure A-4: *Cartesian mesh for VMSD case after 3 adaptive iterations with* $p = 3$ *and* $\approx 37,000$ *requested DOFs.*



(a) *Final Cartesian mesh.*



(b) *Zoomed view.*

Figure A-5: *Cartesian mesh for VMSD case after 40 adaptive iterations with* $p = 3$ *and* $\approx 37,000$ *requested DOFs.*

Appendix B

Axisymmetric Coordinate System

In this section, we outline work similar to that seen in Chapter 4, now comparing the performance of the adaptive algorithm between an axisymmetric coordinate system to the Cartesian coordinate system. As this work does pertain to coordinate transformations and not the potential for improvements based on higher-order meshing, it is kept in this appendix for clarity in the main body of this work.

B.1 Primal Output and Residual

The gradient and divergence operators are defined in axisymmetric coordinates as,

$$\begin{aligned} \nabla u &= \frac{\partial u}{\partial r}\hat{r} + \frac{\partial u}{\partial z}\hat{z} \\ \nabla \cdot \vec{f} &= \frac{1}{r}\frac{\partial}{\partial r}(r f_r) + \frac{\partial f_z}{\partial z} \end{aligned} \tag{B.1}$$

Assuming scalar μ and substituting these definitions into Equations 3.31 and 3.30 gives,

$$\begin{aligned} \mathcal{J}(u) = \int_\Omega g u \, dV &- \oint_{\Gamma_1} h\mu \left(\frac{\partial u}{\partial r}\hat{r} + \frac{\partial u}{\partial z}\hat{z} \right) \cdot \hat{n} \, dS \\ &+ \oint_{\Gamma_2} h \left[\vec{a} u - \mu \left(\frac{\partial u}{\partial r}\hat{r} + \frac{\partial u}{\partial z}\hat{z} \right) \right] \cdot \hat{n} \, dS \end{aligned} \tag{B.2}$$

$$\begin{aligned} \mathcal{R}_s(u, w) = \int_\Omega w \left[a_r \frac{\partial u}{\partial r} + a_z \frac{\partial u}{\partial z} - \mu \left(\frac{1}{r}\frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial r^2} + \frac{\partial^2 u}{\partial z^2} \right) - f \right] dV \\ - \oint_{\Gamma_1} \left[\left[\vec{a} w + \mu \left(\frac{\partial w}{\partial r}\hat{r} + \frac{\partial w}{\partial z}\hat{z} \right) \right] \cdot \hat{n} \right] (u - u_D) \, dS \\ - \oint_{\Gamma_2} \mu \left[\left(\frac{\partial w}{\partial r}\hat{r} + \frac{\partial w}{\partial z}\hat{z} \right) \cdot \hat{n} \right] (u - u_D) \, dS \end{aligned} \tag{B.3}$$

where Γ_1 and Γ_2 are defined in Section 3.1.2.

In the axisymmetric case, our normal vector and differentials are defined as

$$dS = \sqrt{dr^2 + dz^2} \tag{B.4}$$

$$dV = r\, dr\, dz \tag{B.5}$$

$$\hat{n} = (dz\,\hat{r} - dr\hat{z})/dS \tag{B.6}$$

B.2 Volume Output

We first consider the advection-diffusion problem within a rectangular domain. This is defined to be the unit square in Cartesian coordinates, $\Omega \equiv [x_0, x_1] \times [y_0, y_1]$, with $x_0 = 0$, $x_1 = 1$, $y_0 = 0$, $y_1 = 1$ (Figure B-1), and a rectangle in axisymmetric coordinates, $\Omega \equiv [r_0, r_1] \times [z_0, z_1]$, with $r_0 = 0.5$, $r_1 = 1$, $z_0 = 0$, $z_1 = 1$ (Figure B-2). This is done in order to avoid the singularity at $r = 0$, while solving a problem analogous to the Cartesian case. Unless otherwise stated, all problems in this section were solved using the unstabilized CG, DGBR2, and VMSDBR2 discretizations.

For our primal solution we propose to use the double-boundary layer problem as it appears in Section 2.4.1 of Arthur Huang's PhD thesis [40].

$$\nabla \cdot (\vec{a}u - \mu \nabla u) = 0, \tag{B.7}$$

where $\vec{a} \in \mathbb{R}^2$ is the advective velocity, $\mu \in \mathbb{R}^+$ is the viscosity, and the divergence and gradient operators are defined as,

$$\nabla \cdot \vec{f} = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} \tag{B.8}$$

$$\nabla u = \frac{\partial u}{\partial x}\hat{x} + \frac{\partial u}{\partial y}\hat{y} \tag{B.9}$$

We apply Dirichlet boundary conditions of the form,

$$u = u_D \quad \forall \vec{x} \in \partial\Omega \tag{B.10}$$

where u_D is taken from an exact solution to (B.7), given by,

$$u(\vec{x}) = 1 - \prod_{i=1}^{2} \frac{1 - e^{-\frac{a_i(1-x_i)}{\mu}}}{1 - e^{-\frac{a_i}{\mu}}}. \tag{B.11}$$

We select the same specific parameters as were used in Huang's PhD,

$$a_1 = 3/5, \quad a_2 = 4/5, \quad \mu = 1/50 \tag{B.12}$$

this results in a boundary layer thickness of roughly 1/10 the thickness of the domain, behavior we aim to emulate with all boundary layer variations utilized in this work. Our output functional takes the form,

$$\mathcal{J}(u) = \int_{\Omega} gu \, dV \tag{B.13}$$

where g comes from the strong form of the adjoint PDE, $g = -\vec{a} \cdot \nabla w - \nabla \cdot (\mu \nabla w)$. From Section 3.1, we know that an output of this form requires an adjoint solution with homogenous, Dirichlet boundary conditions, $w = 0, \quad \forall \vec{x} \in \partial\Omega$. $g(x, y)$ is chosen using the method of manufactured solutions such that the exact adjoint is a smooth function with continuous derivatives and homogenous boundary conditions,

$$w(r, z) = \sin(\pi x) \sin(\pi y) \tag{B.14}$$

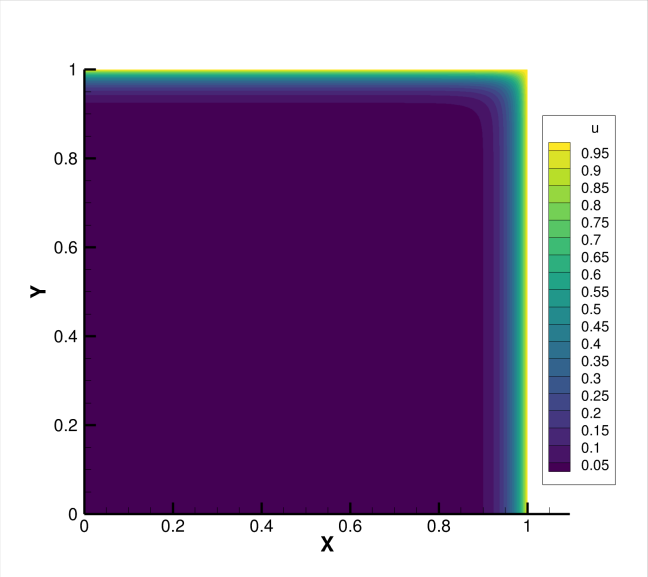
which requires,

$$\begin{aligned} g(x, y) &= -\vec{a} \cdot \nabla w - \nabla \cdot (\mu \nabla w) \\ &= -\pi a_1 \cos \pi x_1 \sin \pi x_2 - \pi a_2 \sin \pi x_1 \cos \pi x_2 + 2\pi^2 \mu \sin \pi x_1 \sin \pi x_2 \end{aligned} \tag{B.15}$$

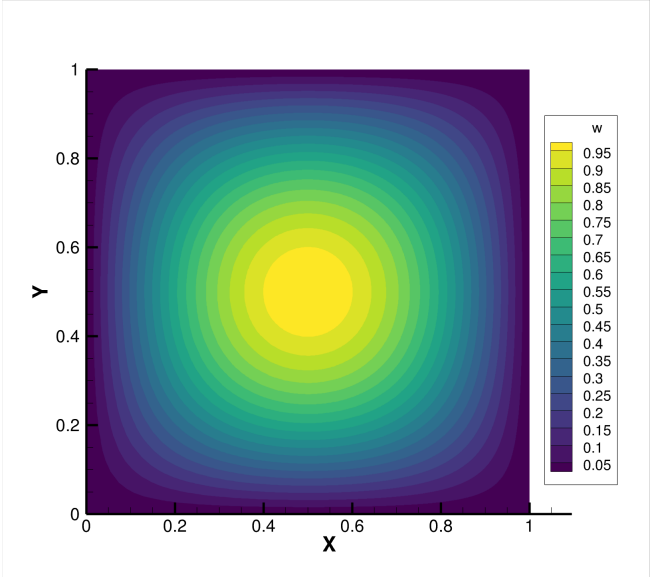
This choice of $g(x, y)$ results in an output $\mathcal{J}(u) = 0.080339559180953$.

We now propose a corresponding axisymmetric formulation to the double boundary layer problem, beginning from Equation B.7, choosing a constant μ and $\vec{a} = (a_r, a_z) = (U_1 \frac{r_1}{r}, W)$, such that $\vec{a}$ is divergence-free,

$$a_r \frac{\partial u}{\partial r} + a_z \frac{\partial u}{\partial z} - \frac{\mu}{r} \frac{\partial}{\partial r} \left(r \frac{\partial u}{\partial r} \right) - \mu \frac{\partial^2 u}{\partial z^2} = 0 \tag{B.16}$$



(a) *Primal solution.*



(b) *Adjoint solution.*

Figure B-1: *Primal and adjoint solutions to volume output problem in the square domain using Cartesian coordinates solved adaptively with VMSD using* $p = 3$ *and* $\approx 37,000$ *requested DOFs.*

As in the Cartesian case, we apply Dirichlet boundary conditions of the form,

$$u = u_D \quad \forall \vec{x} \in \partial\Omega \tag{B.17}$$

where u_D is taken from an exact solution to (B.16), given by,

$$u(\vec{x}) = 1 - \left[\frac{1 - (r/r_1)^{\frac{1}{\alpha_r}}}{1 - (r_0/r_1)^{\frac{1}{\alpha_r}}}\right]\left[\frac{1 - e^{-\frac{1}{\alpha_z}\left(1-\frac{z}{z_1}\right)}}{1 - e^{-\frac{1}{\alpha_z}\left(1-\frac{z_0}{z_1}\right)}}\right]. \tag{B.18}$$

We propose using the following parameters to produce an analytic solution with similar behavior and relative boundary layer thickness to the Cartesian case,

$$\begin{gathered} U_1 = 3/5, \quad W = 4/5, \quad \mu = 1/50, \\ r_0 = 0.5, \quad r_1 = 1, \quad z_0 = 0, \quad z_1 = 1 \\ \alpha_r = \frac{\mu}{U_1 r_1}, \quad \alpha_z = \frac{\mu}{W z_1} \end{gathered} \tag{B.19}$$

The output functional is, again, given by (4.2), with the modification that $dV = r\,dr\,dz$ in the axisymmetric coordinate system. The same constraints apply to the adjoint as in the Cartesian case.

While $g = -\vec{a} \cdot \nabla w - \nabla \cdot (\mu \nabla w)$ as in the Cartesian case, the different definitions of the gradient and

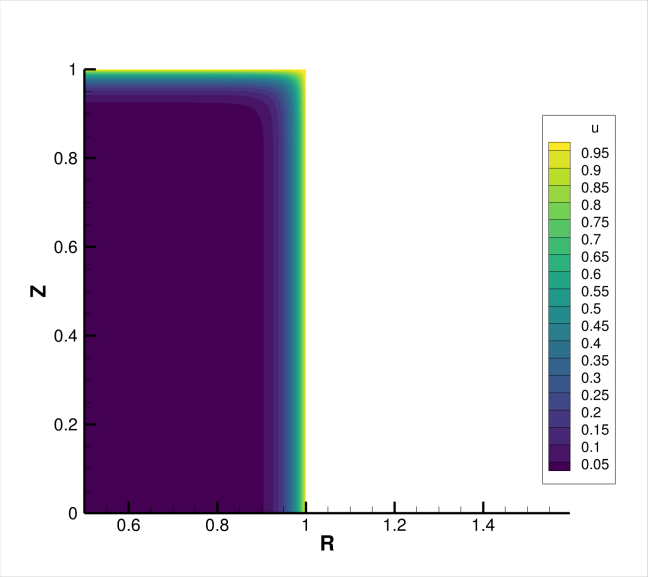
divergence operators must be in taken into account when utilized the method of manufactured solutions, namely, the additional factors of r present in the viscous terms. $g(r,z)$ is chosen using the method of manufactured solutions such that the exact adjoint is a smooth function with continuous derivatives with homogenous boundary conditions,

$$w(r,z) = \sin(2\pi r)\sin(\pi z) \tag{B.20}$$

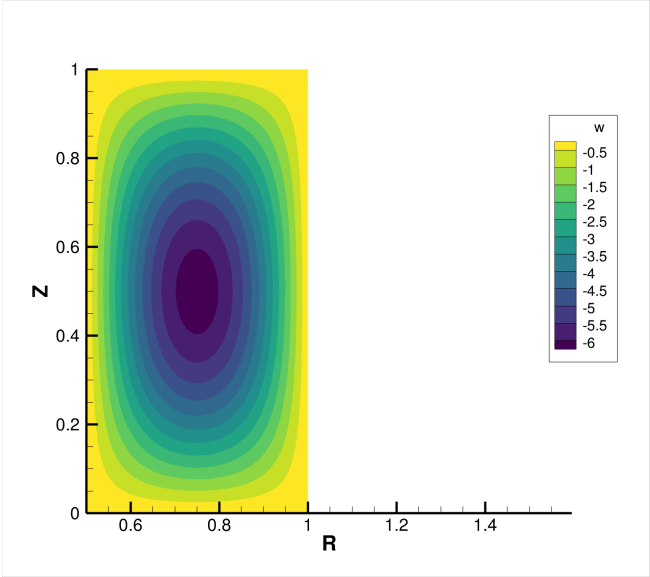
This adjoint solution can be seen in Figure B-2. Solving for $g(r,z)$ gives,

$$\begin{aligned} g(r,z) &= -\vec{a}\cdot\nabla w - \nabla\cdot(\mu\nabla w) \\ &= -2a_r\pi\cos(2\pi r)\sin(\pi z) - a_z\pi\sin(2\pi r)\cos(\pi z) \\ &\quad - 2\pi\frac{\mu}{r}\cos(2\pi r)\sin(\pi z) + 5\mu\pi^2\sin(2\pi r)\sin(\pi z) \end{aligned} \tag{B.21}$$

This choice of $g(r,z)$ results in an output $\mathcal{J}(u) = -0.5999462915963727$. Results for solving this system with adaptive refinement with the VMSD, unstabilized CG, and DG discretizations can be seen in Figures B-3-B-5. Unlike the case of the polar coordinates system, we observe no significant differences between the performance of the axisymmetric and Cartesian coordinate systems. This is likely due to the fact that the domains in both cases are near identical, and there is no geometry being more accurately represented by either coordinate system. As in the previous sections, the adaptive algorithm outperforms uniform refinement, the results of which were omitted for brevity.

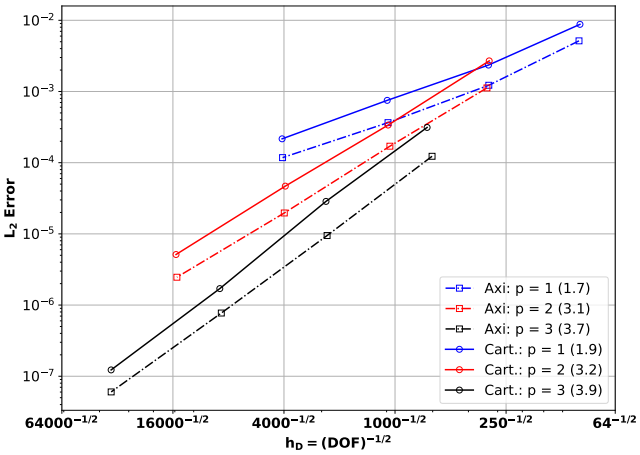


(a) *Primal solution.*

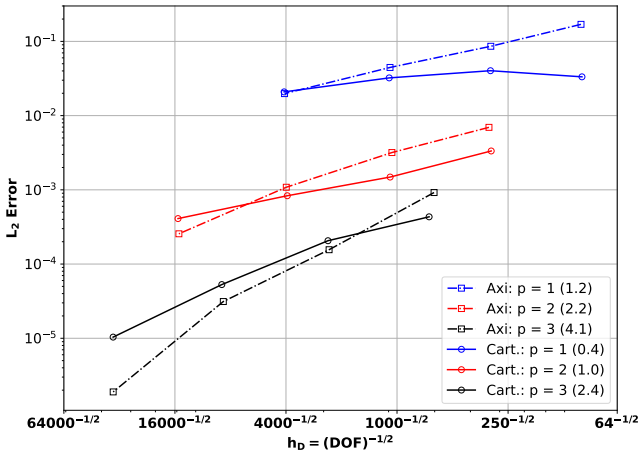


(b) *Adjoint solution.*

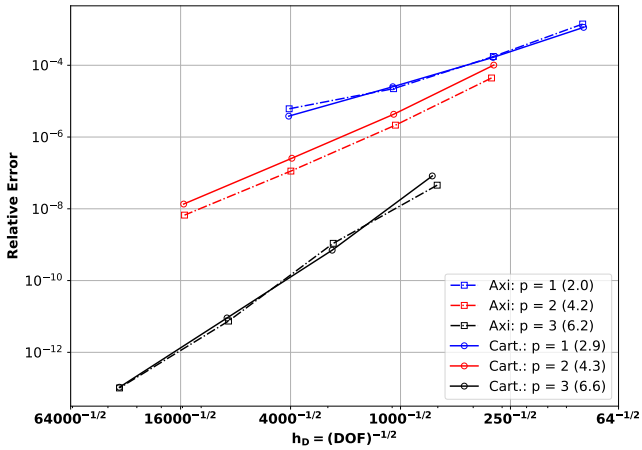
Figure B-2: *Primal and adjoint solutions to volume output problem in the square domain using axisymmetric coordinates solved adaptively with VMSD using* $p = 3$ *and* $\approx 37,000$ *requested DOFs.*



(a) *Primal L_2 error.*

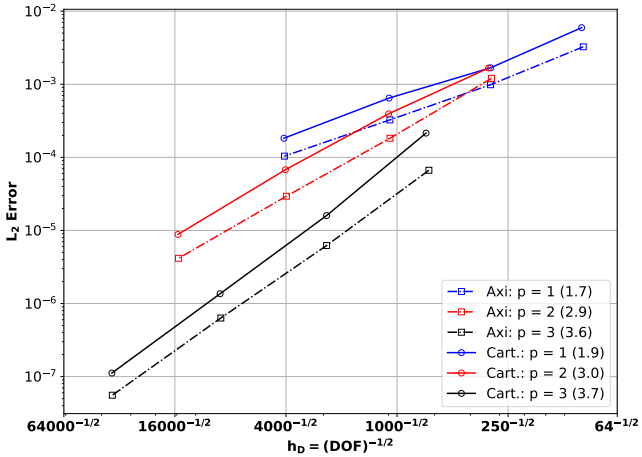


(b) *Adjoint L_2 error.*

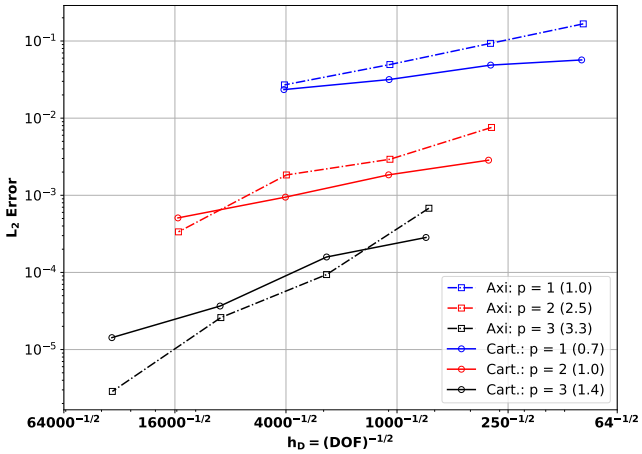


(c) *Relative output error.*

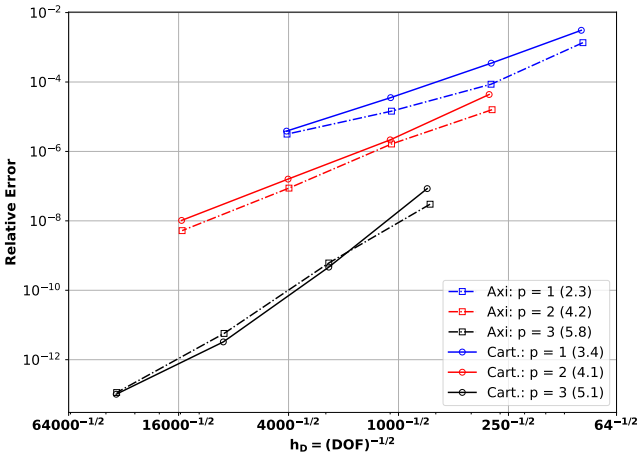
Figure B-3: *Adaptive refinement results for VMSD discretization in both Cartesian and axisymmetric coordinates.*



(a) *Primal L_2 error.*

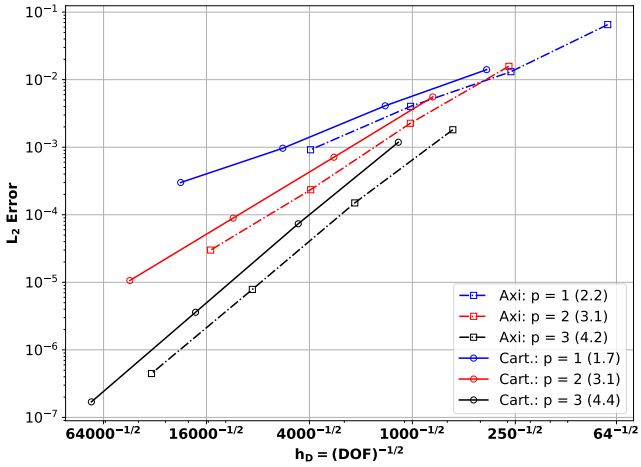


(b) *Adjoint L_2 error.*

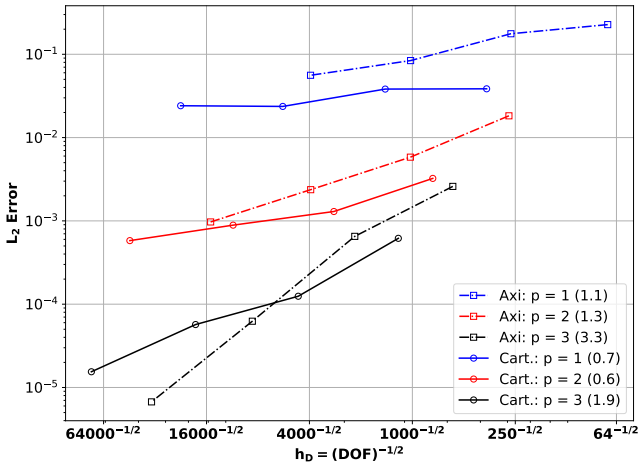


(c) *Relative output error.*

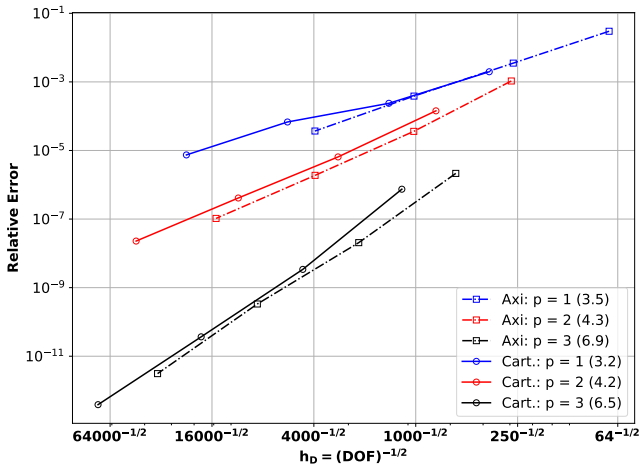
Figure B-4: *Adaptive refinement results for unstabilized CG discretization in both Cartesian and axisymmetric coordinates.*



(a) *Primal L_2 error.*



(b) *Adjoint L_2 error.*



(c) *Relative output error.*

Figure B-5: *Adaptive refinement results for DG discretization in both Cartesian and axisymmetric coordinates.*

B.3 Boundary Output

We now seek to find a solution which produces boundary layer like behavior which is more akin to that of a real-world problem. For this, we utilize a boundary output instead of a volume output, as this more closely emulates the quantities of interest in a real-world problem (e.g. skin friction over an airfoil). For this we aim to create a solution to the advection-diffusion equation which creates a primal boundary layer along the bottom wall which increases in thickness with x, and an adjoint boundary layer along the same wall which decreases in thickness with x. We start from,

$$a\frac{\partial u}{\partial x} - \mu\frac{\partial^2 u}{\partial y^2} = 0 \tag{B.22}$$

where we choose,

$$u(x,y) = u_0 f(n), \quad n = \frac{y}{\sqrt{x\mu/a}} \tag{B.23}$$

which is known to produce a boundary layer with behavior similar to that of the classical Blasius boundary layer. Solving for $y(x,n)$ and inserting this into Equation B.22 gives,

$$-\frac{au_0(nf'(n) + 2f''(n))}{2x} = 0 \tag{B.24}$$

Solving for f,

$$f(n) = C_1\sqrt{\pi}\text{erf}\left(\frac{n}{2}\right) + C_2 \tag{B.25}$$

We now use the conditions $n = 0 \rightarrow \text{erf}(n) = 0$, $f(n) = 0$ and $\lim_{n\rightarrow\infty}\text{erf}(n) \rightarrow 1$, $f(n) \rightarrow 1$ to solve for the unknown coefficients,

$$f(n=0) = C_2 = 0, \tag{B.26}$$

$$\lim_{n\rightarrow\infty} f(n) = C_1\sqrt{\pi} = 1 \rightarrow C_1 = \frac{1}{\sqrt{\pi}} \tag{B.27}$$

This gives the final expression for $f(n)$ and thus, $u(x,y)$,

$$f(n) = \text{erf}\left(\frac{n}{2}\right) = \text{erf}\left(\frac{y}{2\sqrt{x\mu/a}}\right) \tag{B.28}$$

We now consider the full advection-diffusion equation with a source term, g, and a zero advective velocity in the y direction. Since $au_x - \mu u_{yy} = 0$ by construction, we will create g using the method of manufactured solutions to force our second order x derivative to go to zero,

$$au_x - \mu(u_{xx} + u_{yy}) - g = 0 \rightarrow \mu u_{xx} = -g \tag{B.29}$$

$$g(x,y) = \frac{u_0\, y\, e^{\left(-\frac{ay^2}{4\mu x}\right)}\ (-6\mu x + ay^2)}{8\sqrt{\pi}x^3\sqrt{\frac{x\mu}{a}}} \tag{B.30}$$

To create an analytic adjoint solution, we begin from the adjoint PDE with a zero advective velocity in the y direction for consistency,

$$-a\frac{\partial w}{\partial x} - \mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right) = 0 \tag{B.31}$$

which is self-adjoint in the diffusive term and has a flipped advective velocity. We now implement a separation of variables approach, breaking the adjoint into $w(x,y) = X(x)Y(y)$,

$$-Y(y)(aX'(x) + \mu X''(x)) - \mu X(x)Y''(y) = -a\frac{X'(x)}{X(x)} - \mu\frac{X''(x)}{X(x)} - \mu\frac{Y''(y)}{Y(y)} \tag{B.32}$$

We will treat this now as two independent ODEs,

$$-a\frac{X'(x)}{X(x)} - \mu\frac{X''(x)}{X(x)} = c^2 \tag{B.33}$$

$$-\mu\frac{Y''(y)}{Y(y)} = -c^2 \tag{B.34}$$

Since one is a function of x alone and the other a function of y alone, the only possible solution is if both are equal to a constant.

Before solving this system, we note that, if we choose homogenous boundary conditions on $X(x)$, we end up with a trivial solution (i.e. $X(x) = 0 \quad \forall x$). Therefore, a non-zero solution can only be found in the case of non-homogenous boundary conditions. Based on the analysis in Section 3.1, we know that our adjoint solution is constrained by our output functional of choice, and must be non-zero over any boundaries which

we integrate on. With non-homogenous boundary conditions required on three boundaries to fully define the adjoint, our boundary output functional must, therefore, be defined over the same three boundaries; in this case, those boundaries are found at $x = 0$, $x = 1$ and $y = 0$. Our boundary functional is then given by,

$$\begin{aligned}\mathcal{J}(u) = \oint_\Gamma h(\vec{a}u - \mu\nabla u)\cdot\hat{n}\,dS = &\oint_{x_0}^{x_1}(-\mu w\nabla u)|_{y_0}\cdot(-\hat{y})\,dx \\ &+ \oint_{y_0}^{y_1}(w(\vec{a}u-\mu\nabla u)\cdot)|_{x_0}(-\hat{x})\,dy \\ &+ \oint_{y_0}^{y_1}(w(\vec{a}u-\mu\nabla u)\cdot)|_{x_1}(\hat{x})\,dy\end{aligned} \tag{B.35}$$

Noting that there is no component of the advective velocity in the $\hat{y}$ direction and our weight function, h, is equal to the adjoint, w, along the boundaries. We now solve for $X(x)$,

$$\begin{aligned}-a\frac{X'(x)}{X(x)} - \mu\frac{X''(x)}{X(x)} = c^2, \quad X(x_0) = X_0, \quad X(x_1) = X_1 \\ X(x) = \frac{X_0\left(e^{(Ax+Bx_1)} - e^{(Ax_1+Bx)}\right) + X_1\left(e^{(Ax_0+Bx)} - e^{(Ax+Bx_0)}\right)}{e^{(Ax_0+Bx_1)} - e^{(Ax_1+Bx_0)}} \\ A = \frac{1}{2}\left(-\frac{a}{\mu} + \frac{\sqrt{a^2-4c^2\mu}}{\mu}\right), \quad B = \frac{1}{2}\left(-\frac{a}{\mu} - \frac{\sqrt{a^2-4c^2\mu}}{\mu}\right)\end{aligned} \tag{B.36}$$

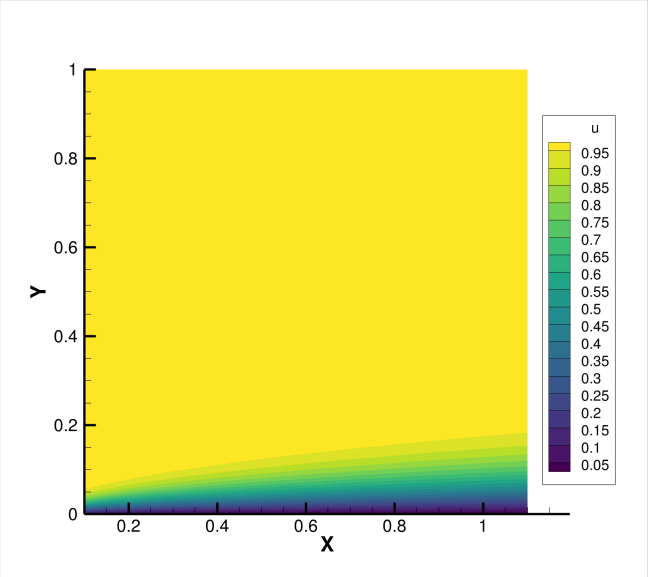
We now solve for $Y(y)$,

$$\begin{aligned}-\mu\frac{Y''(y)}{Y(y)} = -c^2, \quad Y(0) = 1, \quad Y(1) = 0 \\ Y(y) = -\frac{e^{\left(-\frac{cy}{\sqrt{\mu}}\right)}\left(e^{\left(\frac{2cy}{\sqrt{\mu}}\right)} - e^{\left(\frac{2c}{\sqrt{\mu}}\right)}\right)}{e^{\left(\frac{2c}{\sqrt{\mu}}\right)} - 1}\end{aligned} \tag{B.37}$$

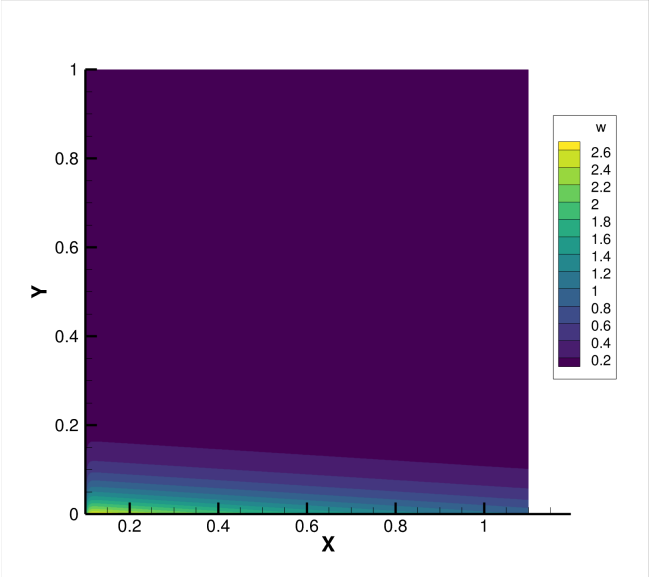
Our analytic adjoint is thus given by $w(x, y) = X(x)Y(y)$. With the choice of parameters $\mu = 1/50$, $a = 3/5$, $c = 1$, $x_0 = 0\,, x_1 = 1$, $X_0 = 1$, $X_1 = 1$, $y_0 = 0$, $y_1 = 1$, our analytic output is given by $\mathcal{J}(u) = 0.1678554248307046$.

For the corresponding axisymmetric case, we aim to create a solution to the advection-diffusion equation which creates a primal boundary layer along the inner wall which increases in thickness with r, and an adjoint boundary layer along the same wall which decreases in thickness with r. We assume a zero z advective velocity. We start from,

$$a_r\frac{\partial u}{\partial r} - \mu\frac{\partial^2 u}{\partial z^2} = 0 \tag{B.38}$$



(a) *Primal solution.*



(b) *Adjoint solution.*

Figure B-6: *Primal and adjoint solutions to boundary output problem in the square domain using Cartesian coordinates solved adaptively with VMSD using* $p = 3$ *and* $\approx 37,000$ *requested DOFs.*

Note that a_r must again be divergence free and is defined as before, $a_r = a_0 \frac{r_1}{r}$. We now choose,

$$u(r, z) = u_0 f(n), \quad n = \frac{z}{\sqrt{r\mu/a_r}} \tag{B.39}$$

which is known to produce a boundary layer with behavior similar to that of the classical Blasius boundary layer. Solving for $z(r, n)$ and inserting this into Equation B.38 gives,

$$-\frac{a_0 u_0 r_1 (n f'(n) + f''(n))}{r^2} = 0 \tag{B.40}$$

Solving for f,

$$f(n) = C_1 \sqrt{\frac{\pi}{2}} \text{erf}\left(\frac{n}{\sqrt{2}}\right) + C_2 \tag{B.41}$$

We now use the conditions $n = 0 \rightarrow \text{erf}(n) = 0$, $f(n) = 0$ and $\lim_{n\to\infty} \text{erf}(n) \rightarrow 1$, $f(n) \rightarrow 1$ to solve for the unknown coefficients,

$$f(n = 0) = C_2 = 0, \tag{B.42}$$

$$\lim_{n\to\infty} f(n) = C_1 \sqrt{\frac{\pi}{2}} = 1 \rightarrow C_1 = \sqrt{\frac{2}{\pi}} \tag{B.43}$$

This gives the final expression for $f(n)$ and thus, $u(r,z)$,

$$f(n) = \text{erf}\left(\frac{n}{\sqrt{2}}\right) = \text{erf}\left(\frac{z}{\sqrt{2r\mu/a_r}}\right) \tag{B.44}$$

We use the method of manufactured solutions again to derive a source term to drive the second order r derivative to zero,

$$a_r\frac{\partial u}{\partial r} - \mu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{\partial^2 u}{\partial z^2}\right) - g = 0 \rightarrow \frac{\mu}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) = -g \tag{B.45}$$

$$g(r,z) = \frac{u_0\, z\, e^{\left(-\frac{a_0 r_1 z^2}{2\mu r^2}\right)}\ (-\mu r^2 + a_0 r_1 z^2)}{r^4\sqrt{\frac{\mu r^2}{a_0 r_1}}}\sqrt{\frac{2}{\pi}} \tag{B.46}$$

The same constraints on the adjoint and output functional are present in the axisymmetric case as were present in the Cartesian case, with the caveat that our output functional now has r as a differential multiplier within the integrand.

To create an analytic adjoint solution, we begin from the adjoint PDE with a zero advective velocity in the z direction for consistency,

$$-a_r\frac{\partial w}{\partial r} - \frac{\mu}{r}\frac{\partial}{\partial r}\left(r\frac{\partial w}{\partial r}\right) - \mu\frac{\partial^2 w}{\partial z^2} = 0 \tag{B.47}$$

which is self-adjoint in the diffusive term and has a flipped advective velocity. We now implement a separation of variables approach, breaking the adjoint into $w(r,z) = R(r)Z(z)$,

$$\frac{Z(z)((\mu + a_0 r_1 R'(r)) + \mu r R''(r))}{r} - \mu R(r)Z''(z) = -\frac{\mu}{r}\frac{R'(r)}{R(r)} - \frac{a_0 r_1}{r}\frac{R'(r)}{R(r)} - \mu\frac{R''(r)}{R(r)} - \mu\frac{Z''(z)}{Z(z)} \tag{B.48}$$

We will treat this now as two independent ODEs,

$$-\frac{\mu}{r}\frac{R'(r)}{R(r)} - \frac{a_0 r_1}{r}\frac{R'(r)}{R(r)} - \mu\frac{R''(r)}{R(r)} = c^2 \tag{B.49}$$

$$-\mu\frac{Z''(z)}{Z(z)} = -c^2 \tag{B.50}$$

Since one is a function of r alone and the other a function of z alone, the only possible solution is if both

are equal to a constant. We first solve for $R(r)$,

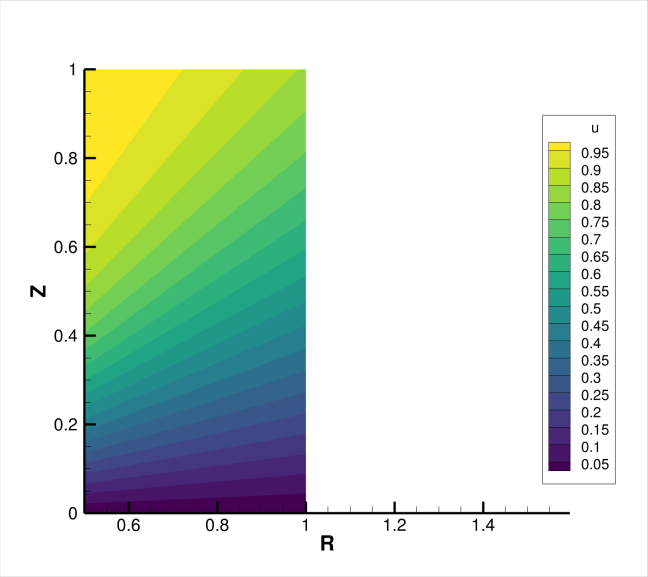
$$
\begin{aligned}
-\frac{\mu}{r}\frac{R'(r)}{R(r)} - \frac{a_0 r_1}{r}\frac{R'(r)}{R(r)} - \mu\frac{R''(r)}{R(r)} = c^2, \quad R(r_0) = R_0, \quad R(r_1) = R_1 \\
R(r) = r^{-A}\Bigg(Y_A\left(\frac{cr}{\sqrt{\mu}}\right)\left(r_0^A R_0 J_A\left(\frac{cr_1}{\sqrt{\mu}}\right) - r_1^A R_1 J_A\left(\frac{cr_0}{\sqrt{\mu}}\right)\right) \\
+J_A\left(\frac{cr}{\sqrt{\mu}}\right)\left(r_1^A R_1 Y_A\left(\frac{cr_0}{\sqrt{\mu}}\right) - r_0^A R_0 Y_A\left(\frac{cr_1}{\sqrt{\mu}}\right)\right)\Bigg)\Bigg/ \\
\left(J_A\left(\frac{cr_1}{\sqrt{\mu}}\right)Y_A\left(\frac{cr_0}{\sqrt{\mu}}\right) - J_A\left(\frac{cr_0}{\sqrt{\mu}}\right)Y_A\left(\frac{cr_1}{\sqrt{\mu}}\right)\right)
\end{aligned}
\tag{B.51}
$$

Where $A = \frac{a_0 r_1}{2\mu}$ and $J_A(r)$ and $Y_A(r)$ correspond to Bessel functions of the 1st and 2nd kind, respectively. Note that the use of Bessel functions imposes the additional constraint that $A \in \mathbb{Z}_{\geq 0}$. To meet this constraint, we impose that $r_1 = 1$ and $\frac{a_0}{2\mu} = 1$ Now we solve for $Z(z)$,

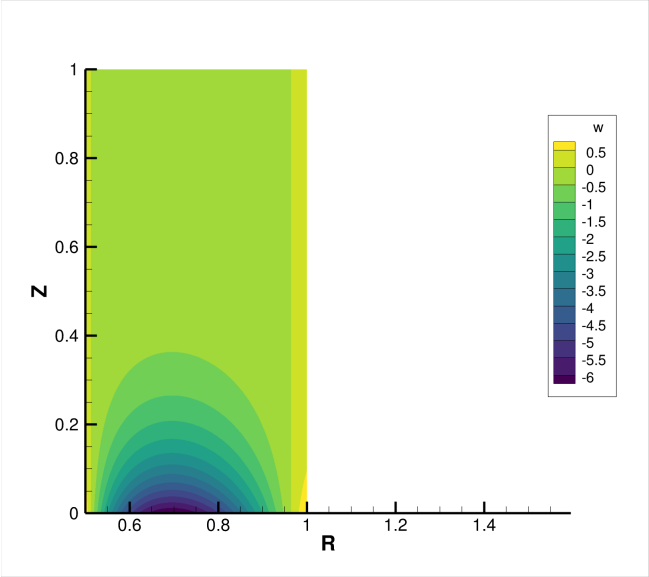
$$
\begin{aligned}
-\mu\frac{Z''(z)}{Z(z)} = -c^2, \quad Z(0) = 1, \quad Z(1) = 0 \\
Y(y) = -\frac{e^{\left(-\frac{cz}{\sqrt{\mu}}\right)}\left(e^{\left(\frac{2cz}{\sqrt{\mu}}\right)} - e^{\left(\frac{2c}{\sqrt{\mu}}\right)}\right)}{e^{\left(\frac{2c}{\sqrt{\mu}}\right)} - 1}
\end{aligned}
\tag{B.52}
$$

Our analytic adjoint is given by $w(r, z) = R(r)Z(z)$. Due to the restriction from using Bessel functions, we are unable to produce a boundary layer of roughly 1/10 the thickness of the domain in this case. With the choice of parameters $\mu = 1/50$, $a = 1/25$, $c = 1$, $r_0 = 0.5\,, r_1 = 1$, $R_0 = 1$, $R_1 = 1$, $z_0 = 0$, $z_1 = 1$, our analytic output is given by $\mathcal{J}(u) = -0.04234874846431988$.

Results for solving this system with adaptive refinement with the VMSD, unstabilized CG, and DG discretizations can be seen in Figures B-8-B-10. Unlike the previous case, we do now observe a significant difference between the performance of the axisymmetric and Cartesian coordinate systems. However, this is an artifact of the resolution with which the exact solution for the Cartesian case was resolved at; the axisymmetric and Cartesian domains are similar, but not exactly the same, resulting in different exact outputs for the two cases. Attempts were made to resolve the output at an arbitrarily high resolution, but the memory requirements were too high given the time constraints, limiting the accuracy. Because of this, the primal, adjoint, and, most notably, the output struggle to converge in the Cartesian case. In the event the exact solution were available, we would expect to see near identical performance between the two coordinate systems, as in the case of the volume output. As in the previous sections, the adaptive algorithm outperforms uniform refinement, the results of which were omitted for brevity.

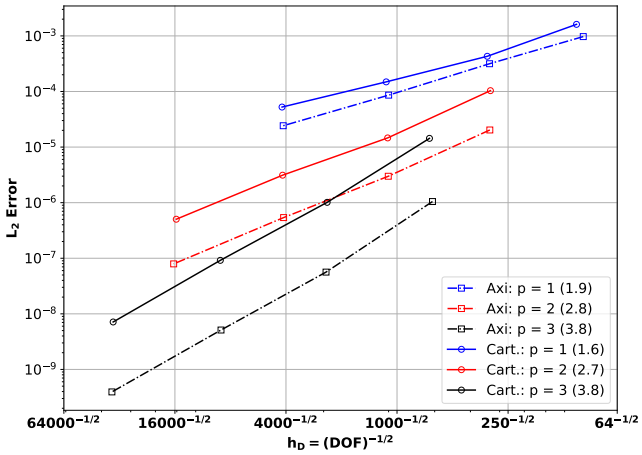


(a) *Primal solution.*

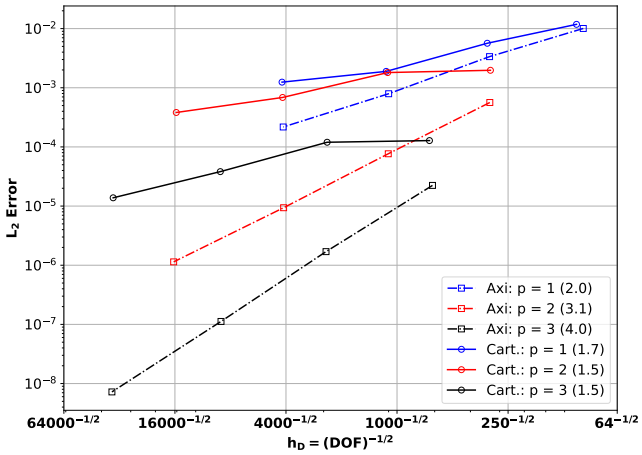


(b) *Adjoint solution.*

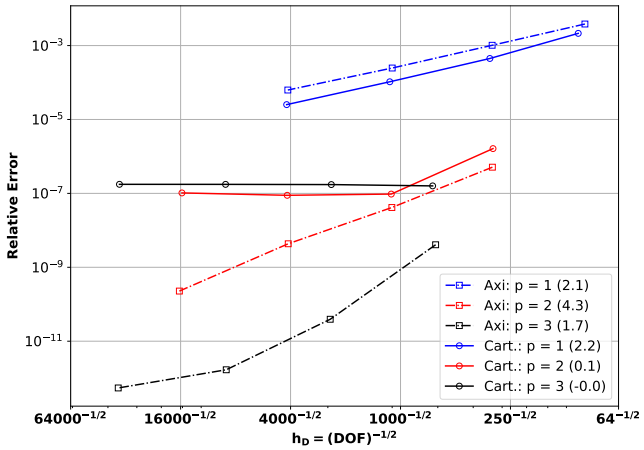
Figure B-7: *Primal and adjoint solutions to boundary output problem in the square domain using axisymmetric coordinates solved adaptively with VMSD using* $p = 3$ *and* $\approx 37,000$ *requested DOFs.*



(a) *Primal L_2 error.*

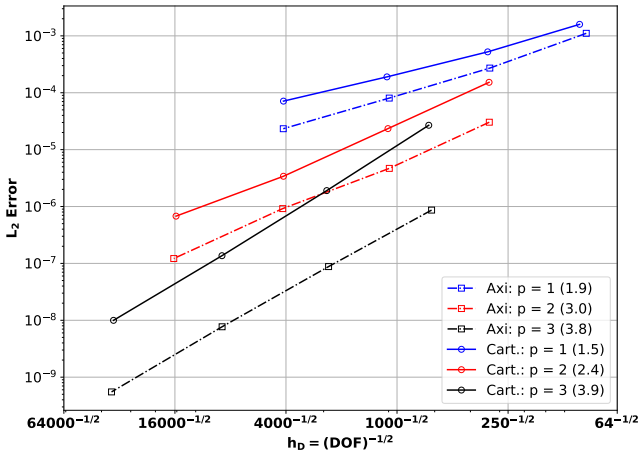


(b) *Adjoint L_2 error.*

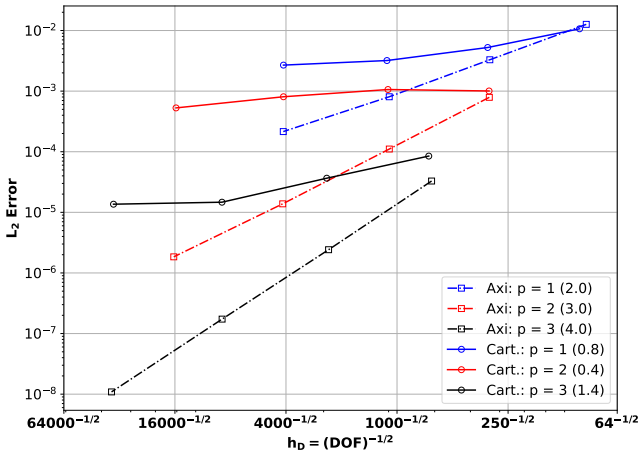


(c) *Relative output error.*

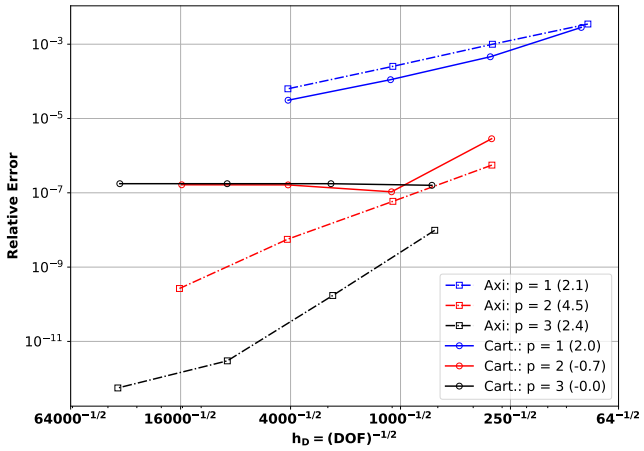
Figure B-8: *Adaptive refinement results for VMSD discretization in both Cartesian and axisymmetric coordinates.*



(a) *Primal L_2 error.*

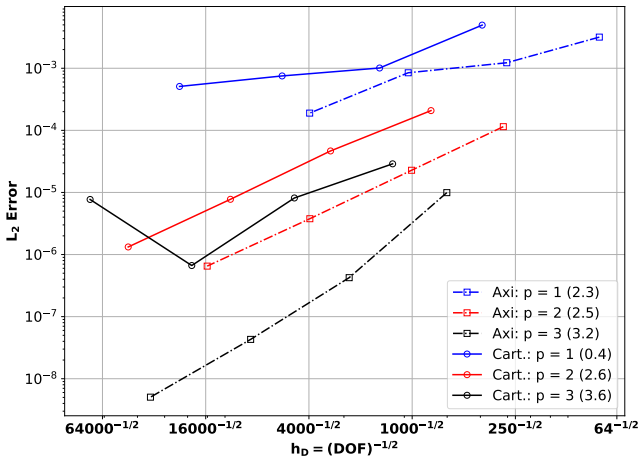


(b) *Adjoint L_2 error.*

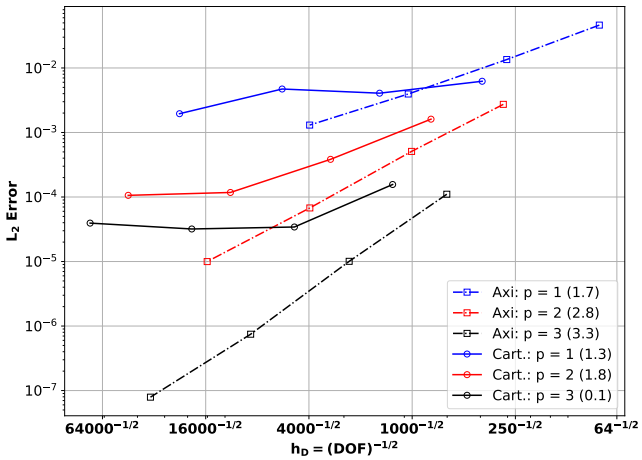


(c) *Relative output error.*

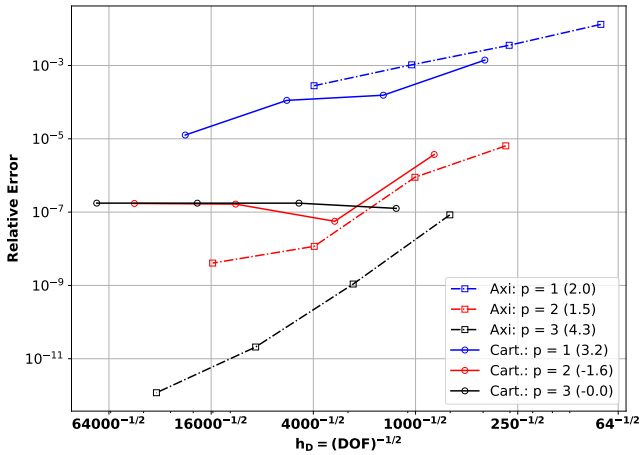
Figure B-9: *Adaptive refinement results for unstabilized CG discretization in both Cartesian and axisymmetric coordinates.*



(a) *Primal L_2 error.*



(b) *Adjoint L_2 error.*



(c) *Relative output error.*

Figure B-10: *Adaptive refinement results for DG discretization in both Cartesian and axisymmetric coordinates.*

Appendix C

Additional Derivations and Definitions

C.1 Nonlinear Adjoint Derivation

Chapter 3 illustrates how the adjoint PDE can be derived in the case of the linear advection-diffusion PDE. This section will repeat this analysis, considering now the case of a non-linear, scalar PDE. The primal PDE and BCs are given by,

$$\nabla \cdot [F(u)) - G(u, \nabla u)] + S(u, \nabla u) = 0, \quad \text{on } \Omega \tag{C.1}$$

$$(F(u) - G(u, \nabla u)) \cdot \hat{n} = b, \quad \text{on } \partial\Omega \tag{C.2}$$

where $G(u, \nabla u)$ is defined as,

$$G(u, \nabla u) = \mathcal{K}(u) \cdot \nabla u \tag{C.3}$$

$$\mathcal{K}(u) = \begin{pmatrix} \mathcal{K}_{xx}(u) & \mathcal{K}_{xy}(u) \\ \mathcal{K}_{xy}^t(u) & \mathcal{K}_{yy}(u) \end{pmatrix} \tag{C.4}$$

Flux BCs are chosen here to simplify the analysis, but a similar analysis is able to be completed for any combination of BCs, as shown in Chapter 3.

Our analysis begins with the following Galerkin primal statement: Find $u \in \mathcal{V}$ such that $R(u, w) = 0$ for all $w \in \mathcal{V}$, where the weighted strong-form primal residual is,

$$\begin{aligned} R(u, w) = &\int_\Omega w^t \left[\nabla \cdot (F(u) - G(u, \nabla u)) + S(u, \nabla u)\right] \\ &- \int_{\partial\Omega} w^t \left[(F(u) - G(u, \nabla u)) \cdot \hat{n} - b\right] \end{aligned} \tag{C.5}$$

The corresponding weak-form primal residual is obtained by integration by parts of the volume term, giving,

$$R(u,w) = \int_\Omega \left[-\nabla w^t \cdot (F(u) - G(u,\nabla u)) + w^t S(u,\nabla u) \right] + \int_{\partial\Omega} w^t b \tag{C.6}$$

We now linearize this form with respect to $u = \epsilon u + u'$ using the Frechet derivative; we linearize about a mean state $\overline{u}$ with perturbation $\epsilon u'$, where non-dimensional $\epsilon << 1$,

$$\begin{aligned} R'[\overline{u}](u',w) &= \lim_{\epsilon\to 0} \frac{R(\overline{u}+\epsilon u', w) - R(\overline{u}, w)}{\epsilon} \\ &= \int_\Omega -\nabla w^t \cdot \left[\left(\frac{\partial F}{\partial u} - \frac{\partial G}{\partial u} \right) u' - \mathcal{K} \cdot \nabla u' \right] + w^t \left(\frac{\partial S}{\partial u} u' + \frac{\partial S}{\partial \nabla u} \cdot \nabla u' \right) \end{aligned} \tag{C.7}$$

We assume our output will be of the form,

$$\mathcal{J}(u) = \int_\Omega g_v(u) + \int_{\partial\Omega} g_b(u) \tag{C.8}$$

Note that, while the output functional could include the diffusive term $\int_{\partial\Omega} g_{b_2}((\mathcal{K}\cdot\nabla u)\cdot\hat{n})$ where $\hat{n}$ is the unit normal, we choose to omit this term to simplify the analysis. Linearizing this output functional gives,

$$\mathcal{J}'[\overline{u}](u') = \int_\Omega \frac{\partial g_v}{\partial u} u' + \int_{\partial\Omega} \frac{\partial g_b}{\partial u} u' \tag{C.9}$$

From here, we invoke duality,

$$J'[\overline{u}](u') - R'[\overline{u}](u',w) = J^*[\overline{u}](w) - R^*[\overline{u}](w,u') \tag{C.10}$$

where all Jacobians and the diffusion matrix $\mathcal{K}$ are evaluated at state $\overline{u}$.

To identify J^* and R^*, we now integrate the diffusion and source terms in the linearized primal residual by parts to isolate u' as a weight,

$$\begin{aligned} R'[\overline{u}](u',w) = \int_\Omega \Bigg[&- \nabla w^t \cdot \left(\frac{\partial F}{\partial u} - \frac{\partial G}{\partial u} \right) - \nabla \cdot \left(\nabla w^t \mathcal{K} \right) + w^t \frac{\partial S}{\partial u} - \nabla \cdot \left(w^t \frac{\partial S}{\partial \nabla u} \right) \Bigg] u' \\ &+ \int_{\partial\Omega} \left(\nabla w^t \mathcal{K} + w^t \frac{\partial S}{\partial \nabla u} \right) u' \cdot \hat{n} \end{aligned} \tag{C.11}$$

From this we can identify that $J^* = 0$ and,

$$\begin{aligned} R^*[\overline{u}](w,u') = \int_\Omega (u')^t \Bigg[&- \left(\frac{\partial F}{\partial u} - \frac{\partial G}{\partial u}\right)^t \cdot \nabla w - \nabla \cdot \left(\mathcal{K}^t \cdot \nabla w\right) + \left(\frac{\partial S}{\partial u}\right)^t w \\ &- \left(\frac{\partial S}{\partial \nabla u}\right)^t \nabla w - \nabla \cdot \left(\frac{\partial S}{\partial \nabla u}\right)^t w - \left(\frac{\partial g_v}{\partial u}\right)^t \Bigg] \\ + \int_{\partial\Omega} (u')^t &\left[\left(\mathcal{K}^t \nabla w + \left(\frac{\partial S}{\partial \nabla u}\right)^t w \right) \cdot \hat{n} - \left(\frac{\partial g_b}{\partial u}\right)^t \right] \end{aligned} \tag{C.12}$$

Dual consistency requires that $R^*[\overline{u}](w,u') = 0$ for $w \in \mathcal{V}$, $u' \in \mathcal{V}$ in order to satisfy the analytic adjoint PDE and BCs, which results in our adjoint PDE,

$$\begin{aligned} &- \left(\frac{\partial F}{\partial u} - \frac{\partial G}{\partial u}\right)^t \cdot \nabla w - \nabla \cdot \left(\mathcal{K}^t \cdot \nabla w\right) + \left(\frac{\partial S}{\partial u}\right)^t w \\ &\quad - \left(\frac{\partial S}{\partial \nabla u}\right)^t \nabla w - \nabla \cdot \left(\frac{\partial S}{\partial \nabla u}\right)^t w = \left(\frac{\partial g_v}{\partial u}\right)^t \quad \text{on } \Omega \end{aligned} \tag{C.13}$$

with BCs given by,

$$\left(\mathcal{K}^t \cdot \nabla w + \left(\frac{\partial S}{\partial \nabla u}\right)^t w \right) \cdot \hat{n} = \left(\frac{\partial g_b}{\partial u}\right)^t \quad \text{on } \partial\Omega \tag{C.14}$$

C.2 CG Stabilization Definitions

This section outlines several CG stabilization schemes and, unless otherwise stated, each discussion outlined pertains only to the scheme's implementation in SANS. We consider the same nonlinear PDE and BCs given by C.1 and C.2. In this case however, the weighted residual is augmented by a stabilization term, e.g. $R_{\text{stab}}(u,w) = R(u,w) + \text{stabilization}$.

Consider a domain Ω, with boundary $\partial\Omega$ and tessellation $\mathcal{T}_h$ of K non-overlapping elements. Let κ refer to a given element volume, and $\partial\kappa$ to the boundary of that element. In SANS, all the different stabilization schemes currently use the same type for $\boldsymbol{\tau}^{-1}$, based on [55] and identified in the code as option "Glasby". This is defined as,

$$[\boldsymbol{\tau}]^{-1} = \sum_k \left(\left| \nabla\phi_k \cdot \frac{\partial F(u_h)}{\partial u_h} \right| + \nabla\phi_k \cdot \frac{\partial G(u_h, \nabla u_h)}{\partial(\nabla u_h)} \cdot \nabla\phi_k \right) + \frac{\partial S(u_h, \nabla u_h)}{\partial u_h} \tag{C.15}$$

where $\frac{\partial F(u_h)}{\partial u_h}$ is the advective flux jacobian, $\frac{\partial G(u_h, \nabla u_h)}{\partial(\nabla u_h)}$ is the viscous flux jacobian, and $\frac{\partial S(u_h, \nabla u_h)}{\partial u_h}$ is the source jacobian.

The strong form of the primal PDE, $r(u_h)$, is then multiplied by $\boldsymbol{\tau}$ for each scheme. Each scheme has a different 'pre-multiplier' associated with it, as is discussed below.

C.2.1 SUPG

For SUPG, the pre-multiplier is equal to the advective flux jacobian (same as above), and the complete integrand takes the form,

$$\sum_{e\in\mathcal{T}}\int_{\Omega}\nabla\phi_k\cdot\left(\frac{\partial F(u_h)}{\partial u_h}(\boldsymbol{\tau}r(u_h))\right)\,dV \tag{C.16}$$

C.2.2 GLS

For GLS, the pre-multiplier has components from the advective flux, diffusive flux, and source (and source gradient if the solution requires it), and the complete integrand takes the form,

$$\begin{aligned}\sum_{e\in\mathcal{T}}\int_{\Omega}\Bigg[\Bigg(&\nabla\phi_k\cdot\frac{\partial F(u_h)}{\partial u_h}-H_{\phi_k}:\frac{\partial G(u_h,\nabla u_h)}{\partial(\nabla u_h)}\\&+\phi_k\frac{\partial S(u_h,\nabla u_h)}{\partial u_h}+\nabla\phi_k\cdot\frac{\partial\vec{S}(u_h,\nabla u_h)}{\partial(\nabla u_h)}\Bigg)(\boldsymbol{\tau}r(u_h))\Bigg]\,dV\end{aligned} \tag{C.17}$$

where the "double dot" is defined using dyadic notation as,

$$\mathbf{A}:\mathbf{B}=(\hat{i}\hat{i}A+\hat{i}\hat{j}B+\hat{j}\hat{i}C+\hat{j}\hat{j}D):(\hat{i}\hat{i}E+\hat{i}\hat{j}F+\hat{j}\hat{i}G+\hat{j}\hat{j}H)=AE+BF+CG+DH \tag{C.18}$$

$$\mathbf{A}:\mathbf{B}=\mathbf{B}:\mathbf{A}=\mathbf{A}^t:\mathbf{B}^t \tag{C.19}$$

C.2.3 VMS

For VMS, the pre-multiplier has components from the advective flux, diffusive flux, diffusive flux gradient, source (and source gradient if the solution requires it), and the complete integrand takes the form,

$$\begin{aligned}\sum_{e\in\mathcal{T}}\int_{\Omega}\Bigg[\Bigg(&\nabla\phi_k\cdot\frac{\partial F(u_h)}{\partial u_h}+H_{\phi_k}:\frac{\partial G(u_h,\nabla u_h)}{\partial(\nabla u_h)}+\nabla\phi_k\cdot\nabla\left(\frac{\partial G(u_h,\nabla u_h)}{\partial(\nabla u_h)}\right)\\&-\phi_k\frac{\partial S(u_h,\nabla u_h)}{\partial u_h}+\nabla\phi_k\cdot\frac{\partial\vec{S}(u_h,\nabla u_h)}{\partial(\nabla u_h)}+\phi_k\nabla\left(\frac{\partial\vec{S}(u_h,\nabla u_h)}{\partial(\nabla u_h)}\right)\Bigg)(\boldsymbol{\tau}r(u_h))\Bigg]\,dV\end{aligned} \tag{C.20}$$

C.2.4 Output Correction

When using these stabilization schemes in SANS, a correction term to the output must be added. Assuming an output of the form,

$$\mathcal{J}(u_h) = \int_\Omega g_v(u_h)) + \int_{\partial\Omega} g_b(u_h) \tag{C.21}$$

We define our correction as,

$$\tilde{u}_h = u_h - \boldsymbol{\tau} r(u_h) \tag{C.22}$$

Leading to the corrected output term,

$$\mathcal{J}(u) = \int_\Omega g_v(\tilde{u}_h) + \int_{\partial\Omega} g_b(\tilde{u}_h) \tag{C.23}$$

Bibliography

[1] M. Yano, J. M. Modisette, and D. L. Darmofal, "The importance of mesh adaptation for higher-order discretizations of aerodynamic flows," AIAA 2011-3852, 2011.

[2] K. Fidkowski and D. Darmofal, "Review of output-based error estimation and mesh adaptation in computational fluid dynamics," *AIAA Journal*, vol. 49, no. 4, pp. 673–694, 2011.

[3] M. Yano and D. L. Darmofal, "An optimization-based framework for anisotropic simplex mesh adaptation," *Journal of Computational Physics*, vol. 231, no. 22, pp. 7626–7649, 2012.

[4] M. Yano and D. Darmofal, "An optimization framework for anisotropic simplex mesh adaptation: Application to aerodynamic flows," AIAA 2012–0079, 2012.

[5] Z. Wang, K. Fidkowski, R. Abgrall, *et al.*, "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, vol. 72, no. 8, pp. 811–845, 2013, ISSN: 1097-0363.

[6] D. L. Darmofal, S. R. Allmaras, M. Yano, and J. Kudo, "An adaptive, higher-order discontinuous galerkin finite element method for aerodynamics," AIAA 2013-2871, 2013.

[7] M. A. Ceze and K. J. Fidkowski, "Drag prediction using adaptive discontinuous finite elements," *AIAA Journal of Aircraft*, vol. 51, no. 4, pp. 1284–1294, 2014.

[8] M. A. Ceze and K. J. Fidkowski, "High-order output-based adaptive simulations of turbulent flow in two dimensions," *AIAA Journal*, vol. 54, no. 2, pp. 2611–2625, 2016.

[9] J. T. Erwin and R. S. Glasby, "Application of HPCMP CREATETM-AV COFFE for three-dimensional turbulent flow cases," in *54th AIAA Aerospace Sciences Meeting*, 2016.

[10] R. S. Glasby and J. T. Erwin, "Comparison between HPCMP CREATETM-AV COFFE and Kestrel for two and three-dimensional turbulent flow cases," in *54th AIAA Aerospace Sciences Meeting*, 2016.

[11] T. Michal, D. Babcock, D. Kamenetskiy, *et al.*, "Comparison of fixed and adaptive unstructured grid results for Drag Prediction Workshop 6," *Journal of Aircraft*, vol. 55, no. 4, pp. 1420–1432, 2017.

[12] D. Prosser and R. S. Glasby, "Evaluation and improvement of robustness, speed, and accuracy of the COFFE CFD solver," in *AIAA SciTech 2019 Forum*, 2019.

[13] J. S. Masters, C. E. Lynch, and R. S. Galsby, "Kestrel results on a 2d cut of the high lift common research model wing," AIAA 2021–0942, 2020.

[14] C.-I. Ursachi, M. C. Galbraith, S. R. Allmaras, and D. L. Darmofal, "Output-based adaptive reynolds-averaged navier–stokes higher-order finite element solutions on a multielement airfoil," *AIAA Journal*, vol. 59, no. 7, pp. 2532–2545, 2021.

[15] M. C. Galbraith, C.-I. Ursachi, D. Chandel, *et al.*, "Comparing multi-element airfoil flow solutions using multiple solvers with output-based adapted meshes," *AIAA Journal*, vol. 60, no. 4, pp. 2629–2643, 2022.

[16] P.-O. Persson and J. Peraire, "Curved mesh generation and mesh refinement using Lagrangian solid mechanics," AIAA 2009-0949, 2009.

[17] M. Fortunato and P.-O. Persson, "High-order unstructured curved mesh generation using the Winslow equations," *Journal of Computational Physics*, vol. 307, pp. 1–14, 2016.

[18] E. Ruiz-Gironés, X. Roca, and J. Sarrate, "High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3d CAD representation," *Computer-Aided Design*, vol. 72, pp. 52–64, 2015.

[19] E. Ruiz-Gironés, J. Sarrate, and X. Roca, "Generation of curved high-order meshes with optimal quality and geometric accuracy," *Procedia Engineering*, vol. 163, pp. 315–327, 2016, 25th International Meshing Roundtable, ISSN: 1877-7058.

[20] A. Kashi and H. Luo, "Curved mesh generation using radial basis functions," AIAA 2016-3179, 2016.

[21] D. P. Sanjaya, K. Fidkowski, and S. M. Murman, "Comparison of algorithms for high-order, metric-based mesh optimization," in *AIAA SciTech 2020 Forum*, 2020, p. 1141.

[22] R. Zhang, A. Johnen, J.-F. Remacle, F. Henrotte, and A. Bawin, "The generation of unit p2 meshes: Error estimation and mesh adaptation," *International Meshing Roundtable (virtual)*, pp. 1–13, 2021.

[23] A. Loseille and F. Alauzet, "Continuous mesh framework part i: Well-posed continuous interpolation error," *SIAM Journal on Numerical Analysis*, vol. 49, no. 1, pp. 38–60, 2011.

[24] A. Loseille and F. Alauzet, "Continuous mesh framework part ii: Validations and applications," *SIAM Journal on Numerical Analysis*, vol. 49, no. 1, pp. 61–86, 2011.

[25] L. Rochery, "High-order metric-based anisotropic mesh adaptation," Ph.D. dissertation, Université Paris-Saclay, 2023.

[26] L. Botti and D. A. Di Pietro, "Assessment of hybrid high-order methods on curved meshes and comparison with discontinuous Galerkin methods," *Journal of Computational Physics*, vol. 370, pp. 58–84, 2018.

[27] D. Moxey, S. P. Sastry, and R. M. Kirby, "Interpolation error bounds for curvilinear finite elements and their implications on adaptive mesh refinement," *Journal of Scientific Computing*, vol. 78, no. 2, pp. 1045–1062, 2019.

[28] J. Docampo-Sánchez, E. Ruiz-Gironés, and X. Roca, "An Efficient Solver to Approximate CAD Curves with Super-Convergent Rates," in *2022 SIAM International Meshing Roundtable (IMR), Virtual Conference*, 2022.

[29] R. Becker and R. Rannacher, "An optimal control approach to a posteriori error estimation in finite element methods," in *Acta Numerica*, A. Iserles, Ed., vol. 10, Cambridge University Press, 2001, pp. 1–102.

[30] M. B. Giles and E. Süli, "Adjoint methods for PDEs: A posteriori error analysis and postprocessing by duality," in *Acta Numerica*, vol. 11, 2002, pp. 145–236.

[31] D. A. Venditti and D. L. Darmofal, "Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows," *Journal of Computational Physics*, vol. 187, no. 1, pp. 22–46, 2003.

[32] R. Hartmann and P. Houston, "Goal-oriented a posteriori error estimation for multiple target functionals," in *Hyperbolic Problems: Theory, Numerics, Applications*, T. Hou and E. Tadmor, Eds., Springer-Verlag, 2003, pp. 579–588.

[33] R. Becker and R. Rannacher, "A feed-back approach to error control in finite element methods: Basic analysis and examples," *East-West Journal of Numerical Mathematics*, vol. 4, pp. 237–264, 1996.

[34] I. Babuska, B. A. Szabo, and I. N. Katz, "The p-version of the finite element method," *SIAM Journal on Numerical Analysis*, vol. 18, no. 3, pp. 515–545, 1981.

[35] T. J. R. Hughes, "A simple scheme for developing upwind finite elements," *International Journal for Numerical Methods in Engineering*, vol. 12, pp. 1359–1365, 9 1978.

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

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

[38] T. J. R. Hughes and G. Sangalli, "Variational multiscale analysis: The fine-scale green's function, projection, optimization, localization, and stabilized methods," *SIAM Journal on Numerical Analysis*, vol. 45, no. 2, pp. 539–557, 2007.

[39] F. Bassi and S. Rebay, "GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations," in *Discontinuous Galerkin Methods: Theory, Computation and Applications*, K. Cockburn and Shu, Eds., Berlin: Springer, 2000, pp. 197–208.

[40] A. C. Huang, "An adaptive variational multiscale method with discontinuous subscales for aerodynamic flows," PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2020.

[41] J. Peraire, N. Nguyen, and B. Cockburn, "An embedded discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations," AIAA 2011-3228, 2011.

[42] A. Loseille and F. Alauzet, "Optimal 3D highly anisotropic mesh adaptation based on the continuous mesh framework," in *Proceedings of the 18th International Meshing Roundtable*, Springer Berlin Heidelberg, 2009, pp. 575–594.

[43] A. Loseille and F. Alauzet, "Continuous mesh model and well-posed continuous interpolation error estimation," INRIA RR-6846, 2009, p. 54.

[44] A. Loseille, A. Dervieux, and F. Alauzet, "On 3-d goal-oriented anisotropic mesh adaptation applied to inviscid flows in aeronautics," AIAA 2010-1067, 2010.

[45] H. A. Carson, A. C. Huang, M. C. Galbraith, S. R. Allmaras, and D. L. Darmofal, "Mesh optimization via error sampling and synthesis: An update," AIAA 2020–0087, 2020.

[46] M. Yano, "An optimization framework for adaptive higher-order discretizations of partial differential equations on anisotropic simplex meshes," PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2012.

[47] H. A. Carson, "Provably convergent anisotropic output-based adaptation for continuous finite element discretizations," PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2019.

[48] K. Svanberg, "A class of globally convergent optimization methods based on conservative convex separable approximations," *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 555–573, 2002.

[49] P. C. Caplan, "Four-dimensional anisotropic mesh adaptation for spacetime numerical simulations," PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2019.

[50] T. Michal and J. Krakos, "Anisotropic mesh adaptation through edge primitive operations," AIAA 2012–159, 2012.

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

[52] I. Babuka, "The finite element method with Lagrangian multipliers," *Numerische Mathematik*, vol. 20, no. 3, pp. 179–192, 1973.

[53] S. Kollmannsberger, A. Özcan, J. Baiges, M. Ruess, E. Rank, and A. Reali, "Parameter-free, weak imposition of Dirichlet boundary conditions and coupling of trimmed and non-conforming patches," *International Journal for Numerical Methods in Engineering*, vol. 101, no. 9, pp. 670–699, 2015.

[54] C.-I. Ursachi, "Output-based adaptive rans solutions using higher-order fem on a multi-element airfoil," Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2020.

[55] R. S. Glasby and J. T. Erwin, "Introduction to COFFE: The next-generation HPCMP CREATE^{TM}-AV CFD solver," AIAA 2016-0567, 2016.