# An Optimization-based Framework for Anisotropic Simplex Mesh Adaptation

Masayuki Yano*,[a], David L. Darmofal[a]

[a] *Aerospace Computational Design Laboratory, Massachusetts Institute of Technology*
*77 Massachusetts Ave. 37-442, Cambridge, MA 02139, USA*

**Abstract**

We present a general framework for anisotropic $h$-adaptation of simplex meshes. Given a discretization and any element-wise, localizable error estimate, our adaptive method iterates toward a mesh that minimizes error for a given degrees of freedom. Utilizing mesh-metric duality, we consider a continuous optimization problem of the Riemannian metric tensor field that provides an anisotropic description of element sizes. First, our method performs a series of local solves to survey the behavior of the local error function. This information is then synthesized using an affine-invariant tensor manipulation framework to reconstruct an approximate gradient of the error function with respect to the metric tensor field. Finally, we perform gradient descent in the metric space to drive the mesh toward optimality. The method is first demonstrated to produce optimal anisotropic meshes minimizing the $L^2$ projection error for a pair of canonical problems containing a singularity and a singular perturbation. The effectiveness of the framework is then demonstrated in the context of output-based adaptation for the advection-diffusion equation using a high-order discontinuous Galerkin discretization and the dual-weighted residual (DWR) error estimate. The method presented provides a unified framework for optimizing both the element size and anisotropy distribution using an *a posteriori* error estimate and enables efficient adaptation of anisotropic simplex meshes for high-order discretizations.

*Key words:* anisotropic adaptation, high-order methods, functional optimization, Riemannian metric field, local solves

## 1. Introduction

As the combination of powerful computers and numerical methods enables increasingly complex simulations, developing an effective means of quantifying and controlling the numerical errors has become increasingly important. Numerous localizable error estimates have been developed in the finite element community, as summarized in work by, for example, Ainsworth and Oden for energy-based estimates [1] and Becker and Rannacher for output-based estimates [2]. While these error estimates provide an assessment of the solution quality, most estimates only assign a scalar value to each element and are insufficient for making the anisotropic or $hp$ adaptation decisions—the decisions necessary for extracting the full potential of flexible finite element discretizations. In this work, we propose a general anisotropic mesh adaptation framework that can be used with any discretization equipped with an element-wise localizable error estimate. The framework naturally inherits the versatility of the underlying discretization and error estimate.

A popular approach to anisotropic adaptivity for simplex meshes is to incorporate the solution Hessian information, which dictates the linear interpolation error. The early application of Hessian anisotropy detection to feature-based adaptation include the work of Peraire et al. [3] and Castro-Diaz et al. [4].

*Corresponding author
*Email address:* myano@mit.edu (Masayuki Yano)

More recently, several variants of Hessian-based anisotropic adaptation have been proposed for output error control, including: Formaggia et al.'s approach based on the Hessian of the dual solution [5, 6]; Venditti and Darmofal's approach based on the Hessian of a scalar component of the primal solution [7]; and Fidkowski and Darmofal's [8] and Leicht and Hartmann's [9] generalization of Venditti's approach to higher-order discretizations by using the $p+1$ derivative. While the Hessian-based methods produce simplex meshes with arbitrarily-oriented anisotropy, there are several limitations. In the context of output error control, the anisotropy decision only accounts for either the primal or dual solution, which may exhibit different directional features from the other solution. For a system of equations, the choice of a scalar field to guide the anisotropic decision is arbitrary, and the decision requires *a priori* knowledge of the equation behavior. For higher-order discretizations, the use of $p+1$ derivative may lead to a lack of robustness for underresolved features.

This work develops a unified framework for anisotropic $h$-adaptation for problems with localizable error estimates. We cast the mesh optimization problem as a continuous constrained optimization problem of a parameter field describing the mesh. In the case of isotropic $h$-adaptation, a natural choice is to parametrize the mesh by a scalar-valued field $\boldsymbol{h} = \{h(x)\}_{x\in\Omega}$ that describes the element diameter distribution. In the case of anisotropic $h$-adaptation, a convenient choice is a tensor-valued field $\boldsymbol{\mathcal{M}} = \{\mathcal{M}(x)\}_{x\in\Omega}$ of Riemannian metric tensors. In particular, utilizing the mesh-metric duality proposed by Loseille and Alauzet [10], we pose the tensor field optimization problem as

$$\boldsymbol{\mathcal{M}}^* = \arg\inf_{\boldsymbol{\mathcal{M}}} \mathcal{E}(\boldsymbol{\mathcal{M}}) \quad \text{s.t.} \quad \mathcal{C}(\boldsymbol{\mathcal{M}}) = N,$$

where $\mathcal{E}(\cdot)$ is the error functional, $\mathcal{C}(\cdot)$ is the cost functional, and $N$ is the maximum permissible cost. To evaluate the variation of the error functional with respect to the tensor field, we first survey the behavior of the error function using a series of local solves. Then, we synthesize the information using a novel error interpolation model that builds on the affine-invariant framework for tensor manipulation introduced by Pennec et al. [11], creating a surrogate model for the error functional $\mathcal{E}(\cdot)$. Finally, we propose an algorithm that iterates toward optimality by manipulating the tensor-valued field.

The idea of using local solves to guide the anisotropy decision for the DWR-based adaptation has been previously proposed. Both Georgoulis et al. [12] and Ceze and Fidkowski [13] used local solves to guide their anisotropy decisions on quadrilateral[1] meshes. However, the perspective set forth in those works is that of steepest descent in the discrete space, in which the local solves are used to guide the sequence of anisotropic subdivision of quadrilateral elements, which permit orthogonal directional decomposition. An anisotropic error estimation and adaptation algorithm of Richter [14] and Leicht and Hartmann [15] also rely on tensor-product structure of quadrilateral elements and perform discrete optimization. A similar discrete optimization based approach for simplices were pursued by Park [16] and Sun [17] using a sequence of edge splits. In this work, we consider a continuous optimization problem and use local solves to estimate the gradient of the error functional. In other words, this work employs local solves as a means to solve the infinite dimensional optimization problem on the metric tensor field; the previous approaches consider the finite dimensional discrete optimization problem where the dimensionality is governed by the complexity of the current mesh. In fact, the anisotropic mesh optimization formulation presented in this work is an extension of the idea for isotropic $h$-adaptation presented by Brandt [18], Rannacher [19], and Section 5 of Becker and Rannacher [2].

As the proposed method is entirely driven by the behavior of the *a posteriori* error estimate, it inherits the versatility of the underlying discretization and error estimate. Thus, the method is particularly powerful when combined with a versatile discretization and error estimate, such as a discontinuous Galerkin (DG) method—which provides an arbitrarily high-order discretization of a wide class of PDEs on complex geometries—and the dual-weighted residual (DWR) error estimate—a general output-based error estimation framework. In the context of DG and DWR, the framework naturally incorporates all components of primal and adjoint solutions and generates an anisotropic simplex mesh optimized for the output at the given discretization order and degrees of freedom.

[1] hexahedrons in three dimensions

This paper is organized as follows. Section 2 provides examples of error estimation and localization techniques that can be used in conjunction with our general adaptation framework. Section 3 reviews mesh-metric duality concepts that are essential to develop our algorithm. Section 4 presents our anisotropic simplex mesh adaptation framework. Section 5 numerically verifies the ability of our algorithm to generate optimal meshes in the context of $L^2$ error control with known optimal mesh distributions. Section 6 demonstrates the ability of the framework to effectively control the output error for a series of advection-diffusion problems.

## 2. Error Estimation and Localization

The anisotropic adaptation framework that we present in Section 4 assumes the presence of a discretization and an element-wise localizable error estimate. We will use two combinations of a discretization and an error estimate in conjunction with our adaptation algorithm in this work. The first is the $L^2$ projector with the $L^2$ error control. The second is the discontinuous Galerkin method with the dual-weighted residual output-error control.

### *2.1. $L^2$ Projector and $L^2$ Error Control*

We consider a simple case of error control problems using the $L^2$ projector as the "solver." Our objective is to control the $L^2$ error of the solution. This solver-error pair eliminates the issues associated with the stability of discretization and allows us to focus on the ability of the space to approximate, or represent, a given function. Moreover, using polynomial interpolation theory and calculus of variations, we can compute the optimal mesh distribution for representing the function, as we will demonstrate in Section 5. Thus, the $L^2$ error control is well-suited for initial verification of our adaptation algorithm.

The finite-dimensional approximation space we consider is denoted by $V_{h,p}$. The space is defined on a triangulation, $\mathcal{T}_h$, of domain $\Omega \subset \mathbb{R}^d$ into elements, $\kappa$, and is given by

$$V_{h,p} = \{v_{h,p} \in L^2(\Omega) : v_{h,p}|_\kappa \in \mathcal{P}_p(\kappa), \forall \kappa \in \mathcal{T}_h\},$$

where $\mathcal{P}_p(\kappa)$ is the space of degree $p$ polynomials on element $\kappa$. We will use variable $d$ to denote the physical dimension of the domain throughout this work. Note that we do not enforce the continuity of the function across the element interfaces.

The $L^2$ projection "solver" finds the solution $u_{h,p} \in V_{h,p}$ that minimizes the square of the $L^2$ projection error, i.e.

$$u_{h,p} = \underset{v_{h,p} \in V_{h,p}}{\arg\inf} \ \mathcal{E}(v_{h,p}),$$

where

$$\mathcal{E}(v_{h,p}) \equiv \|u - v_{h,p}\|^2_{L^2(\Omega)} = \int_\Omega (u - v_{h,p})^2 dx.$$

Because the approximation space $V_{h,p}$ is discontinuous across element interfaces, the $L^2$ projection problem can be solved element-by-element. A straightforward localization of the error functional to elements yields

$$\eta_\kappa \equiv \|u - u_{h,p}\|^2_{L^2(\kappa)} = \int_\kappa (u - u_{h,p})^2 dx, \quad \kappa \in \mathcal{T}_h,$$

and the local errors satisfy $\mathcal{E} = \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa$.

Because the solution $u_{h,p}$ for an approximation space $V_{h,p}$ is unique for a given $u$, we can directly associate the error with the approximation space, i.e.

$$\mathcal{E}(V_{h,p}) = \mathcal{E}(u_{h,p}(V_{h,p})),$$

where $u_{h,p}(V_{h,p})$ signifies that $u_{h,p}$ is a unique function of $V_{h,p}$. The localized errors, $\{\eta_\kappa\}_{\kappa \in \mathcal{T}_h}$, are also uniquely determined by the approximation space $V_{h,p}$. Thus, the $L^2$ solver-error pair yields the global error, $\mathcal{E}$, and the local errors, $\{\eta_\kappa\}_{\kappa \in \mathcal{T}_h}$, as a function of the approximation space $V_{h,p}$. The unique association of the global and local errors with the mesh configuration will be a key in developing our adaptation algorithm.

*2.2. Discontinuous Galerkin Methods and Output Error Control*

The goal of output error control is to measure and control the error in a certain output quantity of a process governed by a PDE. In this work, we consider a general steady state conservation law. The governing equations are of the form

$$\nabla \cdot \mathcal{F}^{\text{conv}}(u, x) - \nabla \cdot \mathcal{F}^{\text{diff}}(u, \nabla u, x) = \mathcal{S}(u, \nabla u, x), \quad \forall x \in \Omega$$

where $u(x) \in \mathbb{R}^m$ is the state vector, $\mathcal{F}^{\text{conv}}$ is the convective flux function, $\mathcal{F}^{\text{diff}}$ is the diffusive flux function, $\mathcal{S}$ is the source function, and $m$ denotes the number of components of the state. The output, $J$, is given by

$$J = \mathcal{J}(u)$$

where $\mathcal{J}$ is the output functional of interest. For many engineering applications, $\mathcal{J}$ is an integral quantity on select surfaces, e.g. drag or heat transfer rate, or volume, e.g. strain energy release rate.

An approximation to the desired output is obtained by discretizing the conservation law and evaluating the discrete output functional. In particular, this work employs a high-order DG finite element method, resulting in the weak form: Find $u_{h,p} \in V_{h,p}$ such that

$$\mathcal{R}_{h,p}(u_{h,p}, v_{h,p}) = 0, \quad \forall v_{h,p} \in V_{h,p}, \tag{1}$$

where $\mathcal{R}_{h,p}(\cdot,\cdot) : V_{h,p} \times V_{h,p} \to \mathbb{R}$ is the semilinear form corresponding to the conservation law. This work uses an upwinding flux function for the convective flux and Bassi and Rebay's second discretization (BR2) [20] for the diffusive flux. Once $u_{h,p} \in V_{h,p}$ is obtained, the desired output is estimated by

$$J_{h,p} = \mathcal{J}_{h,p}(u_{h,p}),$$

where $\mathcal{J}_{h,p} : V_{h,p} \to \mathbb{R}$ is the discrete functional that maintains dual consistency [21, 22, 23].

The objective of functional error estimation is to approximate the true error,

$$\mathcal{E}_{\text{true}} \equiv J - J_{h,p} = \mathcal{J}(u) - \mathcal{J}_{h,p}(u_{h,p}).$$

This work relies on the DWR method [24, 2] to estimate the output error and to localize the error. For brevity, we omit the derivation of the method and state the main results. The output error can be expressed as

$$\mathcal{E}_{\text{true}} = -\mathcal{R}_{h,p}(u_{h,p}, \psi), \tag{2}$$

where $\psi \in V$ is the adjoint solution satisfying

$$\bar{\mathcal{R}}'_{h,p}[u, u_{h,p}](v, \psi) = \bar{\mathcal{J}}'_{h,p}[u, u_{h,p}](v), \quad \forall v \in V, \tag{3}$$

where $\bar{\mathcal{R}}'_{h,p}[u, u_{h,p}](\cdot,\cdot)$ and $\bar{\mathcal{J}}'_{h,p}[u, u_{h,p}](\cdot)$ denote the mean value linearization of $\mathcal{R}_{h,p}(\cdot,\cdot)$ and $\mathcal{J}_{h,p}(\cdot)$ with respect to the first argument evaluated about the interval $[u, u_{h,p}]$ of the true and finite element solutions. We note that by Galerkin orthogonality we can rewrite Eq. (2) to arrive at another error representation

$$\mathcal{E}_{\text{true}} = \bar{\mathcal{R}}'_{h,p}[u, u_{h,p}](u - u_{h,p}, \psi - \psi_{h,p}), \tag{4}$$

which states that the output error is a weighted product of the primal error, $u - u_{h,p}$, and the adjoint error, $\psi - \psi_{h,p}$. In practice, the true adjoint $\psi$ is approximated by $\psi_{h,\hat{p}} \in V_{h,\hat{p}}$ obtained by solving Eq. (3) linearized about $u_{h,p}$ on an enriched space $V_{h,\hat{p}}$ with $\hat{p} = p + 1$. Thus, the DWR error estimate is provided by

$$\mathcal{E}_{\text{true}} \approx -\mathcal{R}_{h,p}(u_{h,p}, \psi_{h,\hat{p}}).$$

For the purpose of mesh adaptation, we define a more conservative error estimate that results from summing locally positive quantities, i.e.

$$\mathcal{E} \equiv \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa,$$

where the element-wise localized error estimate $\kappa$ is defined by

$$\eta_\kappa \equiv |\mathcal{R}_{h,p}(u_{h,p}, \psi_{h,\hat{p}}|_\kappa)|.$$

The combination of the DG solver and the DWR error estimate yields global and local errors, which will be used by our adaptation algorithm.

Assuming the solution, $u_{h,p} \in V_{h,p}$, to the discrete problem in Eq. (1) is unique for a given $V_{h,p}$, the approximate output, $J_{h,p} = \mathcal{J}_{h,p}(u_{h,p})$, is uniquely determined for a given $V_{h,p}$. Thus, similar to the $L^2$ error control case, we can associate the global error with the approximation space in the sense that

$$\mathcal{E}(V_{h,p}) = \sum_{\kappa \in \mathcal{T}_h} |\mathcal{R}_{h,p}(u_{h,p}(V_{h,p}), \psi_{h,\hat{p}}(V_{h,\hat{p}})|_\kappa)|.$$

The localized errors also are a function of the approximation space in the sense that

$$\eta_\kappa(V_{h,p}) = |\mathcal{R}_{h,p}(u_{h,p}(V_{h,p}), \psi_{h,\hat{p}}(V_{h,\hat{p}})|_\kappa)|.$$

Thus, we have a means of measuring the global and local errors as functions of the approximation space $V_{h,p}$. Note that, because the output error is related to the primal and adjoint errors by Eq. (4), an effective control of the output error requires $V_{h,p}$ that accounts for the errors in both the primal and adjoint solutions.

## 3. Riemannian Metric and Metric Manipulation

In this section, we develop technical tools that will facilitate the presentation of our adaptation algorithm in Section 4.

### *3.1. Description*

Let us briefly review the concept of Riemannian metric field used to encode an anisotropic description of element sizes. Notation used in here—and throughout the rest of the work—follows closely that of Loseille et al. [10]. A Riemannian metric field $\boldsymbol{\mathcal{M}} = \{\mathcal{M}(x)\}_{x \in \Omega}$ is a smoothly varying field of symmetric positive definite (SPD) matrices. The length of a vector $\overrightarrow{ab}$ under the metric is given by

$$\ell_{\boldsymbol{\mathcal{M}}}(\overrightarrow{ab}) = \int_0^1 \sqrt{\overrightarrow{ab}^T \mathcal{M}(a + \overrightarrow{ab}s)\overrightarrow{ab}}\, ds.$$

Then, a metric-conforming triangulation is defined as a triangulation such that all edges are unit length under the metric. In practice, we relax the constraint to

$$\frac{1}{\sqrt{2}} \leq \ell_{\boldsymbol{\mathcal{M}}}(e) \leq \sqrt{2}, \quad \forall e \in \text{Edges}(\mathcal{T}_h).$$

The metric-conforming triangulation is not unique; however, a family of metric-conforming triangulations have similar approximation properties because the edges must meet the length constraints imposed by the metric field. Thus, a metric field is a convenient means of encoding the approximability of a triangulation; we will refer to this duality between the metric field and triangulation as metric-mesh duality. Loseille et al. has carried out a rigorous study of the duality for linear interpolation error [10].

*3.2. Metric Manipulation Framework*

The most intuitive method of manipulating a tensor, perhaps, is to simply treat the tensor as an array of numbers and to directly modify the entries of the matrix in the standard Euclidean sense, i.e.

$$\mathcal{M} = \mathcal{M}_0 + \delta\mathcal{M},$$

where $\mathcal{M}_0$ is the original matrix, $\delta\mathcal{M}$ is the modification to the matrix, and $\mathcal{M}$ is the new matrix. However, this method is unsuited for our purpose, as the update, $\delta\mathcal{M}$, must be chosen carefully to maintain the positive definiteness of the tensor. Furthermore, the entries of the update $\delta\mathcal{M}$ are not strongly related to the change in the approximation property of the space. The approximability of the space equipped with a Riemannian metric can be described using the directional length, $h(e)$, defined by

$$h(e;\mathcal{M}) \equiv (e^T\mathcal{M}e)^{-1/2}$$

where $e$ is a unit vector specifying the direction of interest. The change in the approximability in a given direction, or the ratio of the directional lengths between the configurations induced by $\mathcal{M}$ and $\mathcal{M}_0$, is

$$\frac{h(e;\mathcal{M})}{h(e;\mathcal{M}_0)} = \left(\frac{e^T\mathcal{M}_0^{1/2}e}{e^T\mathcal{M}^{1/2}e}\right)^{1/2}.$$

With the entry-wise direct manipulation of $\mathcal{M}_0$, the change in this ratio of directional lengths is not strongly related to the magnitude of the entries of $\delta\mathcal{M}$.

Instead, we consider the tensor manipulation framework that results from endowing the tensor space with an affine-invariant Riemannian metric introduced by Pennec et al. [11]. The affine-invariant metric produces a manifold structure where matrices with zero and infinite eigenvalues are infinite distance from any SPD matrix and a geodesic joining any two tensors is unique. On the Riemannian manifold induced by the affine-invariant metric, the exponential map of a tangent vector $S \in Sym_d$ in the tangent space about $\mathcal{M}_0$ to the manifold is given by

$$\mathcal{M}(S) \equiv \mathcal{M}_0^{1/2}\exp(S)\mathcal{M}_0^{1/2},$$

where $\exp(\cdot)$ is the matrix exponential. Conversely, the logarithmic map of a tensor $\mathcal{M}$ to the tangent space about $\mathcal{M}_0$ is given by

$$S \equiv \log(\mathcal{M}_0^{-1/2}\mathcal{M}\mathcal{M}_0^{-1/2}),$$

where $\log(\cdot)$ is the matrix logarithm. The distance between two tensors $\mathcal{M}$ and $\mathcal{M}_0$ is equal to $\| \log(\mathcal{M}_0^{-1/2}\mathcal{M}\mathcal{M}_0^{-1/2})\|_F$, where $\|\cdot\|_F$ denotes the Frobenius norm of the matrix [11]. As the tangent vector $S$ specifies the change in the metric field, $S$ is referred to as the *step matrix* from hereon. With this choice, the ratio of the directional length is bounded by

$$\exp\left(-\frac{1}{2}\|S\|_F\right) \le \exp\left(-\frac{1}{2}\lambda_{\max}(S)\right) \le \frac{h(e;\mathcal{M}(S))}{h(e;\mathcal{M}_0)} \le \exp\left(-\frac{1}{2}\lambda_{\min}(S)\right) \le \exp\left(\frac{1}{2}\|S\|_F\right). \quad (5)$$

In other words, we can control the change in the directional approximability by controlling the magnitude of $S$. The proof of the relationship is provided in A.1.

By decomposing the step matrix, $S$, into the isotropic and the tracefree parts, we can gain a better insight into the manipulation of the tensors in the tangent space. Let us denote the decomposition by

$$S = sI + \tilde{S},$$

where $s = \mathrm{tr}(S)/d$ such that $\mathrm{tr}(\tilde{S}) = 0$. The exponential map of the decomposed step tensor yields

$$\mathcal{M}(sI+\tilde{S}) = \mathcal{M}_0^{1/2}\exp(sI+\tilde{S})\mathcal{M}_0^{1/2} = \exp(s)\mathcal{M}_0^{1/2}\exp(\tilde{S})\mathcal{M}_0^{1/2}.$$

The expression shows that the isotropic part, $sI$, simply scales the resulting tensor while preserving the shape. In contrary, the change induced by the tracefree part, $\tilde{S}$, modifies the shape while preserving the volume, or the determinant, i.e.

$$\begin{aligned}\det(\mathcal{M}(\tilde{S})) &= \det(\mathcal{M}_0^{1/2}\exp(\tilde{S})\mathcal{M}_0^{1/2}) = \det(\mathcal{M}_0)\det(\exp(\tilde{S}))\\ &= \det(\mathcal{M}_0)\exp(\mathrm{tr}(\tilde{S})) = \det(\mathcal{M}_0).\end{aligned}$$

The decomposition yields a convenient means of manipulating the size and the shape separately, which we exploit in designing the optimization algorithm.

### *3.3. Practical Aspects of Metric-Mesh Duality*

For completeness, let us describe a few practical aspects of mesh-metric duality. The Riemannian metric associated with an element $\kappa$ is a metric such that all edges of the element have unit length with respect to the metric. That is, we solve a $(d+1)(d+2)/2$-dimensional linear system for the coefficients of $\mathcal{M}_\kappa$,

$$e^T\mathcal{M}_\kappa e = 1, \quad \forall e \in \mathrm{Edges}(\kappa).$$

This metric associated with the element $\kappa$ is representative of the metric field over the region covered by $\kappa$.

Many anisotropic mesh generators require metric specified at vertices of the mesh, which is used to construct a smoothly varying metric field $\boldsymbol{\mathcal{M}}$ within the meshing algorithm. Let us denote these vertex metrics by $\{\mathcal{M}_\nu\}_{\nu\in\mathcal{V}}$, where $\mathcal{V}$ is the set of vertices of the triangulation. To reconstruct vertex-based metrics $\{\mathcal{M}_\nu\}_{\nu\in\mathcal{V}}$ from elemental metrics $\{\mathcal{M}_\kappa\}_{\kappa\in\mathcal{T}_h}$, we take the mean of the elemental metrics of the elements surrounding the vertex, i.e.

$$\mathcal{M}_\nu = \mathrm{mean}^{\mathrm{affinv}}(\{\mathcal{M}_\kappa\}_{\kappa\in\omega(\nu)}).$$

Here $\omega(\nu)$ is the set of elements surrounding $\nu$. The mean of the set of metrics is defined as the minimizer of the sum of the squared distance in the affine invariance sense, as defined by Pennec et al. [11], i.e.

$$\mathrm{mean}^{\mathrm{affinv}}(\{\mathcal{M}_\kappa\}_{\kappa\in\omega(\nu)}) = \arg\min_{\mathcal{M}} \sum_{\kappa\in\omega(\nu)} \| \log\left(\mathcal{M}_\kappa^{-1/2}\mathcal{M}\mathcal{M}_\kappa^{-1/2}\right)\|_F^2.$$

The mean value is computed iteratively using the intrinsic gradient descent algorithm described in [11]. Using these steps, we can generate either elemental (discontinuous) or vertex-based (continuous) representation of the metric field associated with a triangulation.

To perform the inverse mapping, i.e. from a metric field to a mesh, we can use any metric-conforming mesh generator. We have used Bidimensional Anisotropic Mesh Generator (BAMG) [25] for all the results shown in this work.

## 4. Anisotropic Adaptation Algorithm

### *4.1. Output Error Minimization Problem*

In Section 2, we introduced the means of expressing discretization errors as a function of the approximation space, $V_{h,p}$. The objective of our adaptation is to find the space $V_{h,p}^*$ that minimizes the error for a given dimension of $V_{h,p}$, i.e.

$$V_{h,p}^* = \arg\inf_{V_{h,p}} \mathcal{E}(V_{h,p}) \quad \text{s.t.} \quad \dim(V_{h,p}) \le N,$$

where $N$ is the maximum permissible dimension of $V_{h,p}$ and is often set by the available computational resource. In particular, if $V_{h,p}$ consists of elements with a constant polynomial order $p$, then $V_{h,p}$ is described

by the triangulation $\mathcal{T}_h$ and the scalar $p$, i.e. $V_{h,p} = V_{h,p}(\mathcal{T}_h, p)$. Thus, for a fixed $p \in \mathbb{R}^+$, the optimization problem simplifies to that of finding the optimal triangulation $\mathcal{T}_h^*$ such that

$$\mathcal{T}_h^* = \arg\inf_{\mathcal{T}_h} \mathcal{E}(V_{h,p}(\mathcal{T}_h, p)) \quad \text{s.t.} \quad \dim(V_{h,p}(\mathcal{T}_h, p)) \leq N. \tag{6}$$

This is a discrete-continuous optimization problem, as the triangulation $\mathcal{T}_h$ is defined by the node locations and the connectivity of the nodes. In general, the problem is intractable.

In order to find an approximate solution to the problem, we consider a continuous relaxation of the discrete problem, following the approach pursued by Loseille et al. [10]. In particular, we appeal to the fact that the Riemannian metric field $\boldsymbol{\mathcal{M}} = \{\mathcal{M}(x)\}_{x\in\Omega}$ captures the approximability of the triangulation $\mathcal{T}_h$. Thus, we can cast a continuous relaxation of the discrete problem, Eq. (6), as

$$\boldsymbol{\mathcal{M}}^* = \arg\inf_{\boldsymbol{\mathcal{M}}} \mathcal{E}(V_{h,p}(\mathcal{T}_h(\boldsymbol{\mathcal{M}}), p)) \quad \text{s.t.} \quad \dim(V_{h,p}(\mathcal{T}_h(\boldsymbol{\mathcal{M}}), p)) \leq N.$$

For brevity, we write the optimization problem as

$$\boldsymbol{\mathcal{M}}^* = \arg\inf_{\boldsymbol{\mathcal{M}}} \mathcal{E}(\boldsymbol{\mathcal{M}}) \quad \text{s.t.} \quad \mathcal{C}(\boldsymbol{\mathcal{M}}) \leq N, \tag{7}$$

where $\mathcal{E}$ and $\mathcal{C}$ are the error and cost functionals that map the metric tensor field to the error and cost, respectively. The expression assumes that the polynomial order, $p$, is constant and fixed. The extension of the continuous optimization framework to *hp*-adaptation would require introduction of the solution order field $\{p(x)\}_{x\in\Omega}$.

### *4.2. Error Locality Assumption*

In order to solve the optimization problem Eq. (7), we need a means of approximating the behavior of the error and cost functionals. If we use the degrees of freedom as the measure of cost, then the cost functional takes the form

$$\mathcal{C}(\mathcal{M}) = \int_\Omega c(\mathcal{M}(x), x) dx,$$

where $c(\cdot,\cdot) : Sym_d^+ \times \mathbb{R}^d \to \mathbb{R}^+$ is the local cost function. In the view of metric-mesh duality [10], the local cost function for the discontinuous piecewise polynomial space is given by

$$c(\mathcal{M}(x), x) = c_p \sqrt{\det(\mathcal{M}(x))}, \tag{8}$$

where $c_p$ is the degrees of freedom associated with a reference element normalized by the size of the reference element, $\hat{\kappa}$. For example, the coefficient associated with a triangular element, whose reference element is the unit equilateral triangle, is $c_p = (2/\sqrt{3})(p+1)(p+2)$. This choice allows us to recover the correct elemental cost, $\rho_\kappa$, when the local cost function is integrated over an element, i.e.

$$\rho_\kappa = \int_\kappa c(\mathcal{M}(x), x) dx \approx \int_\kappa c_p \sqrt{\det(\mathcal{M}_\kappa)} dx = c_p |\hat{\kappa}| = \text{dof}(\hat{\kappa}).$$

To estimate the behavior of the error functional, we make an assumption that the functional results from a sum of the local contributions, i.e.

$$\mathcal{E}(\mathcal{M}) = \int_\Omega e(\mathcal{M}(x), x) dx,$$

where $e(\cdot,\cdot) : Sym_d^+ \times \mathbb{R}^d \to \mathbb{R}^+$ is the local error function that maps the local approximability described by $\mathcal{M}(x)$ to the local contribution to the output error. This locality assumption is formally only applicable to errors that only depend on local properties, e.g. $L^2$ projection errors. However, we have found that the

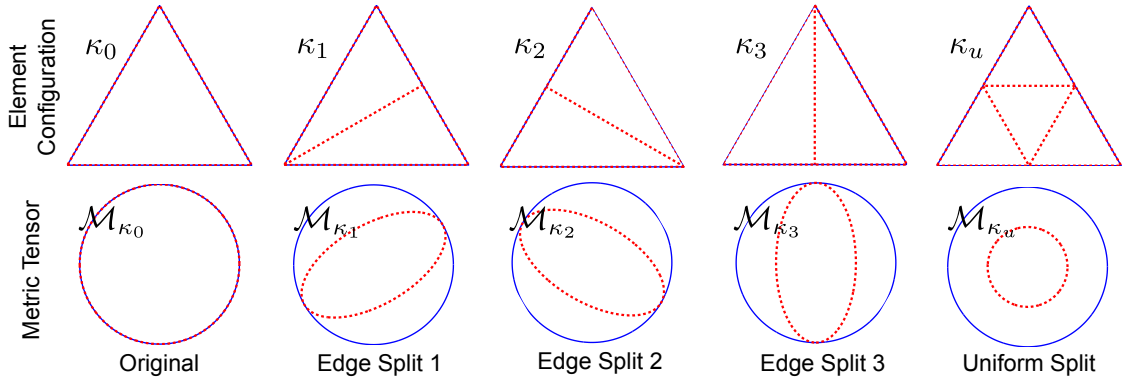



Figure 1: The original, edge split, and uniformly split configurations used to sample the local error function behavior. The metrics implied by the sampled configurations are shown in dashed lines.

algorithm developed based on the assumption works well in practice for output-based error estimates for DG discretizations. Under the locality assumption, we can write the elemental error contribution as

$$\eta_\kappa = \int_\kappa e(\mathcal{M}(x), x)dx \approx \int_\kappa e(\mathcal{M}_\kappa, x)dx \quad \Rightarrow \quad \eta_\kappa = \eta_\kappa(\mathcal{M}_\kappa).$$

That is, the elemental error—or the error associated with the region covered by $\kappa$—is a function of the metric $\mathcal{M}_\kappa$ that encodes the approximation properties of the region covered by $\kappa$. In general, the dependency of $\eta_\kappa$ on $\mathcal{M}_\kappa$ is not known *a priori*. Our goal is to estimate the behavior of $\eta_\kappa(\mathcal{M}_\kappa)$ using local sampling and a novel metric-error interpolation scheme.

### *4.3. Local Error Sampling*

The goal of local error sampling step is to probe the behavior of the local elemental error $\eta_\kappa$ as a function of the local metric $\mathcal{M}_\kappa$. Here, we probe the functional dependency by directly monitoring the behavior of the elemental error or *a posteriori* error estimate for several different local configurations. Let us first describe the procedure in context of the $L^2$ error control.

We consider $n_{\text{config}}$ configurations obtained by locally splitting the edges, as shown in Figure 1. We will denote the configuration obtained by the $i$-th local modification by $\kappa_i$. By convention, $\kappa_0$ is the original configuration. For configuration $\kappa_i$, we resolve the $L^2$ projection problem to obtain the associated solution $u_{h,p}^{\kappa_i}$, i.e.

$$u_{h,p}^{\kappa_i} = \underset{v_{h,p}^{\kappa_i} \in V_{h,p}(\kappa_i)}{\arg\inf} \|u - v_{h,p}^{\kappa_i}\|^2_{L^2(\kappa)},$$

where $V_{h,p}(\kappa_i)$ is the piecewise polynomial space associated with $\kappa_i$. Once we obtain the solution, we can compute the error associated with the configuration, $\eta_{\kappa_i}$, i.e.

$$\eta_{\kappa_i} = \|u - u_{h,p}^{\kappa_i}\|^2_{L^2(\kappa)}.$$

We expect the error $\eta_{\kappa_i}$, $i > 0$, to be lower than that of the original configuration, $\eta_{\kappa_0}$, because $V_{h,p}(\kappa_i) \supset V_{h,p}(\kappa_0)$, $i > 0$. Different configurations yield different reduction in the error, depending on how the approximability of the space is modified by the edge split operation with respect to the function $u$. In particular, we encode the approximability of configuration $\kappa_i$ into the associated metric $\mathcal{M}_{\kappa_i}$, the affine invariant mean of the elemental metric tensors of the split configuration, i.e.

$$\mathcal{M}_{\kappa_i} = \text{mean}^{\text{affinv}}(\{\mathcal{M}_{\kappa_i}^j\}_{j=1}^{n_{\text{elem}}^{\text{split}}}),$$

where $n_{\text{elem}}^{\text{split}} = 2$ for edge split and $n_{\text{elem}}^{\text{split}} = 4$ for uniform split. Repeating the procedure for all $n_{\text{config}}$ configurations, we construct metric-error pairs

$$\{\mathcal{M}_{\kappa_i}, \eta_{\kappa_i}\}_{i=1}^{n_{\text{config}}}.$$

These pairs capture the anisotropic behavior of the local error because anisotropic edge split configurations are included in $n_{\text{config}}$ configurations.

The construction of metric-error pairs for the DG discretization with the DWR error estimate follows a similar procedure. First, we solve an element-wise local problem associated with $\kappa_i$. The local solution, $u_{h,p}^{\kappa_i} \in V_{h,p}(\kappa_i)$, is a function defined on the subdivided mesh, $\kappa_i$, that satisfies

$$R_{h,p}^{\kappa_i}(u_{h,p}^{\kappa_i}, v_{h,p}^{\kappa_i}) = 0, \quad \forall v_{h,p}^{\kappa_i} \in V_{h,p}(\kappa_i),$$

where the local semilinear form, $R_{h,p}^{\kappa_i}(\cdot,\cdot)$, sets the boundary fluxes on $\kappa_i$ assuming the solution on the neighbor elements does not change. Then, we recompute the localized DWR error estimate corresponding to the subdivided mesh as

$$\eta_{\kappa_i} \equiv |\mathcal{R}_{h,p}(u_{h,p}^{\kappa_i}, \psi_{h,p+1}|_{\kappa_0})|.$$

Due to the local Galerkin orthogonality of the DG scheme, we can rewrite the local error as

$$\eta_{\kappa_i} = |\mathcal{R}_{h,p}(u_{h,p}^{\kappa_i}, (\psi_{h,p+1} - \psi_{h,p}^{\kappa_i})|_{\kappa_0})|.$$

The equality signifies that the local sampling procedure automatically accounts for the improvement in the adjoint approximability resulting from the local refinement even though the local adjoint problem is not explicitly solved.[2] Thus, the local sampling technique based on the *a posteriori* error estimate automatically captures the behaviors of both primal and dual solutions. Finally, we compute the local metric associated with $\kappa_i$, $\mathcal{M}_{\kappa_i}$, to construct metric-error pairs $\{\mathcal{M}_{\kappa_i}, \eta_{\kappa_i}\}_{i=1}^{n_{\text{config}}}$.

The construction of the local metric-to-error map in principle extends to any localizable error estimate. The only requirement is that the local error estimates based on the local solves are representative of the errors that may be observed by solving the global problems with a similar local approximability. While the requirement is not formally satisfied for solver-error pairs that do not meet the error locality assumption, such as the DG discretization with the DWR error estimate, the procedure works well in practice.

### *4.4. Local Error Model Synthesis*

The goal of the model synthesis step is to construct a continuous metric-error function $\eta_\kappa(\cdot) : Sym_d^+ \to \mathbb{R}^+$ from the pairs $\{\mathcal{M}_{\kappa_i}, \eta_{\kappa_i}\}_{i=1}^{n_{\text{config}}}$ collected in the sampling stage. We aim to improve the robustness and the efficiency of the adaptation process by introducing a continuous model and optimizing the surrogate model, as oppose to, for example, simply combining a fixed-fraction size selection strategy with a competitive anisotropic selection based on the minimum error-to-dof configuration as often done for quadrilateral meshes [12, 13]. The noisiness of an individual anisotropic error sample collected on a simplex element is documented in [16]; We improve the robustness of the adaptation decision process by synthesizing the potentially noisy metric-error samples using a regression-based framework. Further, the continuous error model permits continuous adjustment of all elements in the mesh every adaptation cycle, reducing the number of adaptation iterations required to reach optimality compared to a fixed-fraction type strategy.

Our interpolation framework builds on Pennec's affine invariant framework for tensor manipulation [11] briefly reviewed in Section 3.2. First, we recall that the logarithmic map of a metric about the original configuration $\mathcal{M}_{\kappa_0}$ provides a convenient means of characterizing the change in the approximability of the region, as discussed in Section 3.2. Thus, we will measure the changes in the configuration as

$$S_{\kappa_i} = \log\left(\mathcal{M}_{\kappa_0}^{-1/2}\mathcal{M}_{\kappa_i}\mathcal{M}_{\kappa_0}^{-1/2}\right), \quad i = 0, \ldots, n_{\text{config}}.$$

[2] We have also experimented with solving the $p+1$ local dual problems as done in [12] for quadrilateral elements, but numerically observed no quantifiable difference in the quality of the error estimate and hence the adaptation efficiency.

Note that, by construction, the original configuration, $\mathcal{M}_{\kappa_0}$, maps to the origin, i.e. $S_{\kappa_0} = 0$. Similarly, we measure the associated changes in the errors as

$$f_{\kappa_i} = \log\left(\eta_{\kappa_i}/\eta_{\kappa_0}\right), \quad i = 0, \ldots, n_{\text{config}}.$$

Again, the original error, $\eta_{\kappa_0}$, maps to zero by construction.

Once we have the pairs $\{S_{\kappa_i}, f_{\kappa_i}\}_{i=1}^{n_{\text{config}}}$ that characterizes the change in the error as a function of the change in the configuration, our objective is to construct a continuous function $f_\kappa(\cdot) : Sym_d \to \mathbb{R}$. We choose to construct a linear function in the entries of $S_\kappa$,

$$f_\kappa(S_\kappa) = \text{tr}(R_\kappa S_\kappa). \tag{9}$$

To find an appropriate $d \times d$ symmetric matrix $R_\kappa$ that governs the behavior of the linear function, we perform the least-squares regression of the known data, i.e.

$$R_\kappa = \underset{Q \in Sym_d}{\arg\min} \sum_{i=1}^{n_{\text{config}}} \left(f_{\kappa_i} - \text{tr}(QS_{\kappa_i})\right)^2.$$

Note that, if $n_{\text{config}}$ is equal to the degrees of freedom of the symmetric matrix $R_\kappa$ (e.g. three in two dimensions), the regression becomes an interpolation, and the resulting linear function matches exactly at the data points. In our case, we use four configurations (i.e. three anisotropic edge splits and one uniform refinement), so the linear function is not an interpolant.

The local error model is given as

$$\eta_\kappa(S_\kappa) = \eta_{\kappa_0} \exp\left(\text{tr}\left(R_\kappa S_\kappa\right)\right).$$

One of the important properties of the proposed error reconstruction scheme is that the quality of the reconstruction is not affected by the current configuration, $\mathcal{M}_{\kappa_0}$. In other words, the quality of the model—and subsequent adaptation decisions—is preserved even on high aspect ratio elements encountered in anisotropic adaptation. This property is proved in A.2.

We can gain a better insight into the error function behavior by decomposing the rate tensor $R_\kappa$ into the isotropic and the tracefree parts, i.e.

$$R_\kappa = r_\kappa I + \tilde{R}_\kappa,$$

where $\text{tr}(\tilde{R}_\kappa) = 0$. Combined with the decomposition of the step tensor $S_\kappa$ into $S_\kappa = s_\kappa I + \tilde{S}_\kappa$, the local error model simplifies to

$$\begin{aligned}
\eta_\kappa(s_\kappa I + \tilde{S}_\kappa) &= \eta_{\kappa_0} \exp\left(\text{tr}\left((r_\kappa I + \tilde{R}_\kappa)(s_\kappa I + \tilde{S}_\kappa)\right)\right) \\
&= \eta_{\kappa_0} \exp\left(r_\kappa s_\kappa d + \text{tr}\left(\tilde{R}_\kappa \tilde{S}_\kappa\right)\right) \\
&= \eta_{\kappa_0} \exp(r_\kappa s_\kappa d) \exp\left(\text{tr}\left(\tilde{R}_\kappa \tilde{S}_\kappa\right)\right),
\end{aligned}$$

where the cross terms vanish because $\text{tr}(\tilde{R}_\kappa I) = 0$ and $\text{tr}(\tilde{S}_\kappa I) = 0$. The decomposition shows that the trace of $R_\kappa$ controls the change in the error under isotropic scaling, and the tracefree part of $R_\kappa$ controls change in the error under shape modification. Thus, the rate matrix $R_\kappa$ can be thought of as a generalization of the convergence rate for isotropic scaling to anisotropic manipulation. A precise relationship between our anisotropic error model and the standard isotropic error model,

$$\eta_\kappa^{\text{iso}}(h) = \eta_{\kappa_0} \left(\frac{h}{h_0}\right)^{r_\kappa^{\text{iso}}},$$

where $r_\kappa^{\text{iso}}$ is the isotropic convergence rate, is derived in A.3.

*4.5. Local Cost Model*

The element-wise cost function model, $\rho_\kappa$, is obtained by directly integrating the continuous local cost function over an element, i.e.

$$\begin{aligned}\rho_\kappa(S_\kappa) &= \int_\kappa c(\mathcal{M}(x),x)dx = \int_\kappa c_p\sqrt{\det\mathcal{M}(x)}dx = \int_\kappa c_p\sqrt{\det(\mathcal{M}_{\kappa_0}^{1/2}\exp(S_\kappa)\mathcal{M}_{\kappa_0}^{1/2})}dx \\ &= \int_\kappa c_p\sqrt{\det(\mathcal{M}_{\kappa_0}^{1/2}\exp(s_\kappa I+\tilde{S}_\kappa)\mathcal{M}_{\kappa_0}^{1/2})}dx = \rho_{\kappa_0}\exp\left(\frac{1}{2}\mathrm{tr}(S_\kappa)\right) = \rho_{\kappa_0}\exp\left(\frac{d}{2}s_\kappa\right).\end{aligned}$$

Note that the cost is only a function of $s_\kappa$, which controls the scaling of the tensor, and not $\tilde{S}_\kappa$, which controls the shape.

*4.6. Optimization of the Surrogate Model*

The final step of the adaptation algorithm is to optimize the metric field $\boldsymbol{\mathcal{M}}$, described by vertex values $\{\mathcal{M}_\nu\}_{\nu\in\mathcal{V}}$. The vertex-based metric can then be used to generate a metric-conforming mesh using an anisotropic mesh generator. To manipulate the metric tensors at vertices, we describe the changes in the tangent space about the original configuration, $\mathcal{M}_{\nu_0}$, and use the exponential map, i.e.

$$\mathcal{M}_\nu(S_\nu) = \mathcal{M}_{\nu_0}^{1/2}\exp\left(S_\nu\right)\mathcal{M}_{\nu_0}^{1/2}. \tag{10}$$

Here, $S_\nu \in Sym_d$ describes the change in the metric at vertex $\nu$. Thus, given $\{\mathcal{M}_{\nu_0}\}_{\nu\in\mathcal{V}}$, our objective is to choose the step matrices $\{S_\nu\}_{\nu\in\mathcal{V}}$ to reduce the error.

To solve the optimization problem, we first need to write the objective function $\mathcal{E}$ and the cost constraint $\mathcal{C}$ in terms of the optimization variables $\{S_\nu\}_{\nu\in\mathcal{V}}$. Substitution of the local error model into the error functional yields

$$\mathcal{E}(\boldsymbol{\mathcal{M}}) = \int_\Omega e(x,\mathcal{M}(x))dx \approx \sum_{\kappa\in\mathcal{T}_h}\eta_\kappa(\mathcal{M}_\kappa(S_\kappa)). \tag{11}$$

In other words, we have approximated the behavior of the error functional in terms of the changes in the configuration in each region covered by $\kappa$, $S_\kappa$. We assign the change in the configuration over an element $S_\kappa$ as the simple arithmetic mean of the changes at its vertices. That is, denoting the vertices of $\kappa$ by $\mathcal{V}(\kappa)$, we have

$$S_\kappa = \overline{\{S_\nu\}_{\nu\in\mathcal{V}(\kappa)}} \equiv \frac{1}{|\mathcal{V}(\kappa)|}\sum_{\nu\in\mathcal{V}(\kappa)}S_\nu.$$

Substitution of the expression into the error model Eq. (11) yields our objective function,

$$\mathcal{E}\left(\{S_\nu\}_{\nu\in\mathcal{V}}\right) = \sum_{\kappa\in\mathcal{T}_h}\eta_\kappa\left(\overline{\{S_\nu\}_{\nu\in\mathcal{V}(\kappa)}}\right).$$

Similarly, we can write our cost constraint in terms of our $\{S_\nu\}_{\nu\in\mathcal{V}}$ as

$$\mathcal{C}\left(\{S_\nu\}_{\nu\in\mathcal{V}}\right) = \sum_{\kappa\in\mathcal{T}_h}\rho_\kappa\left(\overline{\{S_\nu\}_{\nu\in\mathcal{V}(\kappa)}}\right).$$

We note that the error model is a good approximation of the error behavior only in vicinity of the original configuration, because the model is built from the local samples of the configuration. Thus, we need to limit the change in the metric field in each step. This is accomplished by limiting the entries of $S_\nu$, $\nu\in\mathcal{V}$, i.e.

$$|(S_\nu)_{ij}| \le \alpha, \quad i,j=1,\ldots,d, \quad \forall\nu\in\mathcal{V},$$

where the constant $\alpha$ specifies the region over which the metric-error map is considered reliable. For this work, we use $\alpha = 2\log(2)$, which limits the change in the approximability to 2 in any direction—the range over which the sampling is performed and the error model is reliable.

By introducing the surrogate error and cost functions, we have turned our infinite dimensional optimization problem of the metric tensor field (with an unknown error function) into a finite dimensional optimization of vertex step matrices. The surrogate optimization problem for the optimal $\{S_\nu\}_{\nu\in\mathcal{V}}$ is

$$\{S_\nu^*\}_{\nu\in\mathcal{V}} = \underset{\{S_\nu\}_{\nu\in\mathcal{V}}}{\arg\inf}\, \mathcal{E}\left(\{S_\nu\}_{\nu\in\mathcal{V}}\right) \tag{12}$$

$$\text{s.t.} \quad \mathcal{C}\left(\{S_\nu\}_{\nu\in\mathcal{V}}\right) = N \tag{13}$$

$$|(S_\nu)_{ij}| \leq \alpha, \quad i,j = 1,\ldots,d, \quad \forall \nu \in \mathcal{V}. \tag{14}$$

We emphasize that we do not intend to solve the problem exactly, because our error model, based on local sampling and surrogate model, is only an approximation to the true problem. Thus, investing a large computational effort into solving the surrogate optimization problem would be counterproductive.

Assuming that the current configuration is sufficiently close to the optimal configuration such that the constraints Eq. (14) are inactive, the first order optimality condition for the optimization problem Eq. (12)-(13) is given by

$$\frac{\partial \mathcal{E}}{\partial s_\nu} + \lambda \frac{\partial \mathcal{C}}{\partial s_\nu} = 0, \tag{15}$$

$$\frac{\partial \mathcal{E}}{\partial \tilde{S}_\nu} = 0, \quad \forall \nu \in \mathcal{V}, \tag{16}$$

for some Lagrange multiplier $\lambda$, where $s_\nu = \mathrm{tr}(S_\nu)/d$ and $\tilde{S}_\nu$ is the trace-free part of $S_\nu$. The first condition, Eq. (15), is a global condition for the size distribution. In particular, if we define the "local" Lagrange multiplier as

$$\lambda_\nu \equiv \frac{\partial \mathcal{E}}{\partial s_\nu} \Big/ \frac{\partial \mathcal{C}}{\partial s_\nu},$$

then we must have $\lambda_\nu = \lambda$, $\forall \nu \in \mathcal{V}$. The global coupling is provided by the Lagrange multiplier, $\lambda$. The local Lagrange multiplier, $\lambda_\nu$, is interpreted as the marginal improvement in the local error for a given investment in the local cost, which is the degrees of freedom in the context of mesh adaptation. The global condition states that, at optimality, the investment to any element results in the same marginal improvement in the error.

The second condition, Eq. (16), is a local condition that states that the error is stationary with respect to the shape change. Note that this second optimality condition is satisfied if

$$\tilde{R}_\kappa = 0, \quad \forall \kappa \in \mathcal{T}_h. \tag{17}$$

The shape change, induced by $\tilde{S}_\nu$, does not affect the cost. Thus, if $\tilde{R}_\kappa \neq 0$, then we can reduce the error by choosing a $\tilde{S}_\nu$ such that $\mathrm{tr}\left(\tilde{R}_\kappa \overline{\{S_\nu\}_{\nu\in\mathcal{V}(\kappa)}}\right) < 0$ without affecting the cost. Thus, the stationarity with respect to the shape change is required at optimality.

If the current configuration is far from the optimal configuration, then some of the constraints Eq. (14) become active and the equalities in the two optimality conditions Eq. (15) and (16) are replaced by inequalities on those constrained variables.

Let us now propose a gradient-based algorithm to solve the surrogate optimization problem Eq. (12)-(14). We again emphasize that our objective is to only approximately solve the problem. Our algorithm for solving the optimization problem is:

-1. Evaluate/reconstruct $\rho_{\kappa_0}$, $\eta_{\kappa_0}$, and $R_\kappa$ that define local cost and error models

0. Set $\delta s = \alpha/n_{\text{step}}$, which controls the incremental change in the metric such that the maximum change over $n_{\text{step}}$ steps is limited to $\alpha$. This enforces Eq. (14) and prevents large changes that would render our error model inaccurate. $S_\nu^0 = 0$, $\forall \nu \in \mathcal{V}$. Set $n = 0$.
1. Compute vertex derivatives, $\partial\mathcal{E}/\partial s_\nu$, $\partial\mathcal{E}/\partial\tilde{S}_\nu$, and $\partial\mathcal{C}/\partial s_\nu$ and the local Lagrange multiplier $\lambda_\nu \equiv (\partial\mathcal{E}/\partial s_\nu)/(\partial\mathcal{C}/\partial s_\nu)$ about $\{S_\nu^n\}_{\nu\in\mathcal{V}}$.
2. Work toward equidistributing the local Lagrange multiplier and satisfying the global optimality condition, Eq. (15), by updating the isotropic part of $S_\nu$ according to:
   - Refine top 30% of the vertices $\nu$ with the largest $\lambda_\nu$ by setting $S_\nu^{n+1/3} = S_\nu^n + \delta s I$
   - Coarsen top 30% of the vertices $\nu$ with the smallest $\lambda_\nu$ by setting $S_\nu^{n+1/3} = S_\nu^n - \delta s I$

   This fixed-fraction type refinement results in a more robust mesh adaptation than a simple steepest descent, which can behave poorly when the error is dominated by few elements.
3. Work toward satisfying the local shape optimality condition, Eq. (16), by updating the anisotropic part of $S_\nu$ according to $S_\nu^{n+2/3} = S_\nu^{n+1/3} - \delta s(\partial\mathcal{E}/\partial\tilde{S}_\nu)/(\partial\mathcal{E}/\partial s_\nu)$.
4. Rescale $S_\nu^{n+2/3}$ to obtain a metric field with desired degrees of freedom. That is, $S_\nu^n = S_\nu^{n+2/3} + \beta I$, where $\beta$ is selected to satisfy Eq. (13).
5. Set $n = n + 1$. If $n < n_{\text{step}}$ go back to 1.

After obtaining the desired field of vertex step matrices $\{S_\nu\}_{\nu\in\mathcal{V}}$, we modify the vertex metrics using the exponential map, Eq. (10). Finally, the resulting the metric field, described by the vertex values, is fed to a metric-conforming mesh generator to generate a new mesh.

The proposed adaptation algorithm is independent of the particular coordinate representation of the tensors. This property implies that the same physical problem represented in two different coordinate systems produces an identical sequences of tensor fields with respect to the physical problem. The property is proved in A.4.

### *4.7. Practical Considerations for Output-Based Adaptation*

Let us now summarize properties of the optimization algorithm that are particularly important for practical output-based mesh adaptation.

- The method handles any discretization order and mixed orders
- The method uses the simplex remeshing strategy, which allow for arbitrarily-oriented anisotropic elements.
- The method does not make any *a priori* assumption about the convergence behavior of the error. Because no *a priori* assumptions are utilized on the convergence rate, the method is more robust when features are under-resolved in the presence of a singularity or singular perturbation.
- Both the sizing and the anisotropy decisions are driven directly by the *a posteriori* error estimates, which automatically captures the behaviors of both the primal and dual solutions as well as all components of the states. Improved efficiency for problems in which primal and adjoint solutions exhibit different directional features is expected.
- The local error model synthesis
- The method inherits the versatility of the adjoint-based error estimate, which exclusively governs adaptation decisions. For example, the framework straightforwardly extends to different governing equations (e.g. Navier-Stokes, structural elasticity, Maxwell's).
- Of the three steps of the adaptation algorithm (local error sampling, error model synthesis, and surrogate model optimization), the local error sampling constitutes majority (over 90%) of the computational cost. The error sampling for each element equates to residual evaluation and error estimation

on approximately 10 elements (three 2-element edge split samples and one 4-element uniform refinement sample). Thus, in theory, the proposed sampling strategy could increase the computational cost by a factor of 10 for a small linear problem where the global linear solve cost is a small fraction of the residual evaluation cost. However, in practice, the perfectly-scalable local solves become a smaller fraction of the global solve, which scales superlinearly, as the problem size grows. The trend is more pronounced for a nonlinear problem, in which local problems require much fewer Newton iterations than the global problem. (For a typical two-dimensional Reynolds-averaged Navier-Stokes flow over an isolated airfoil, the adaptation cost constitutes less than 10% of the flow solve [26].)

## 5. $L^2$ Error Control

We verify the ability of the proposed mesh optimization algorithm to produce optimal meshes in a series of $L^2$ error control problems. As previously mentioned, the $L^2$ error control problems are suited for verification purposes as the errors are truly localized and the analytical solutions are attainable from the polynomial interpolation theory and calculus of variations. We will consider two problems; the first problem is isotropic, and the second problem is anisotropic.

### *5.1. $r^\alpha$-Type Corner Singularity*

We consider a solution with a $r^\alpha$-type corner singularity, where $r$ is the distance from the singular corner and $\alpha > 0$ is a constant determining the strength of the singularity. In particular, we consider the solution to the classical $L$-shaped domain Poisson problem, whose exact solution is given by

$$u = r^{2/3} \sin\left[\frac{2}{3}\left(\theta + \frac{\pi}{2}\right)\right],$$

where $r^2 = x_1^2 + x_2^2$ and $\tan(\theta) = x_2/x_1$.

#### *5.1.1. Analytical Solution*

We analytically obtain the optimal mesh size distribution using polynomial interpolation theory and calculus of variations, following a similar approach pursued by Schwab [27] and Loseille and Alauzet [10]. To simplify the analysis, let us restrict ourselves to a domain with circular outer boundary of the form

$$\Omega = \{x \in \mathbb{R}^2 : \|x\| \leq R,\ \theta_1 \leq \arctan(x_2/x_1) \leq \theta_2\}.$$

Then, the error functional can be expressed as

$$\mathcal{E}(h) = \int_\Omega C_p h^{2(p+1)} \left|\frac{\partial^{p+1} u}{\partial r^{p+1}}\right|^2 dx = \int_0^R C_{\alpha,p,\theta} h^{2(p+1)} r^{2\alpha-2p-1} dr$$

Similarly, the cost functional on $\Omega$ is

$$\mathcal{C}(h) = \int_\Omega c_p h^{-2} dx = \int_0^R c_{p,\theta} h^{-2} r dr$$

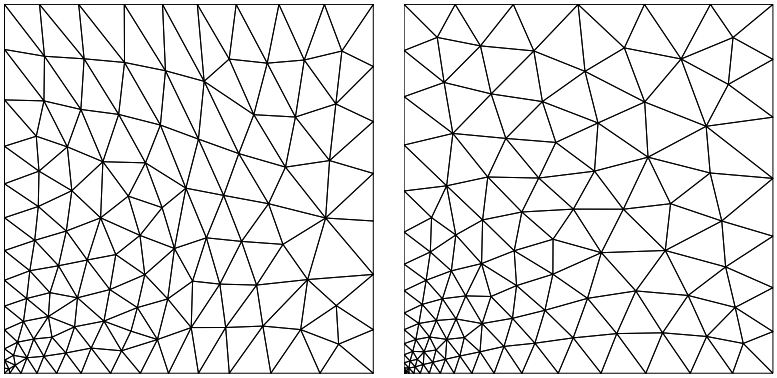
Forming the Lagrangian and solving for the stationary point, we obtain

$$h = C_{\alpha,p,\theta_1,\theta_2} r^{k^{\text{analytic}}}$$

where the optimal grading coefficient is given by

$$k^{\text{analytic}} = 1 - \frac{\alpha + 1}{p + 2}.$$

The expression shows that the grading becomes stronger as $\alpha$ decreases or $p$ increases for the corner singularity.



(a) $p = 1$ (b) $p = 3$

Figure 2: The optimized meshes for the corner singularity problem containing approximately 200 elements.

*5.1.2. Numerical Solution*

We apply the proposed mesh optimization algorithm to a corner singularity problem with $\alpha = 2/3$. Examples of optimized meshes obtained for the problem using $p = 1$ and 3 approximation spaces are shown in Figure 2. Each mesh contains approximately 200 elements. The stronger grading toward the singularity located at the bottom left corner for the $p = 3$ mesh is evident from the figure.

A more quantitative assessment of the optimization strategy is obtained by studying the distribution of the element size, $h$, against the distance from the singularity, $r$, and comparing the distribution with the analytical result. Figure 3 shows the distribution of $h$ against $r$ for the optimized meshes. The element size $h$ is computed based on the volume, i.e. $h = \det\left(\mathcal{M}_\kappa\right)^{-1/4}$ where $\mathcal{M}_\kappa$ is the elemental implied metric. The distance $r$ is measured from the singularity to the centroid of elements. The optimization is performed for $p = 1$ and $p = 3$ at the degrees of freedom count of 1000 and 4000. The optimal grading coefficient calculated analytically for $p = 1$ and 3 are $k^{\text{analytic}} = 0.44$ and 0.67, respectively. Knowing the optimal values of $h$ and $r$ varies linear in log-log space, we also plot the least-squares fit to $\log(h)$ vs. $\log(r)$. The proposed optimization algorithm produces meshes with the grading factor of $k = 0.47$ and 0.67 for $p = 1$ and $p = 3$, respectively, at 4000 degrees of freedom. Thus, the optimal grading is automatically obtained for each $p$ without any *a priori* knowledge of the solution behavior for the two $p$'s.

*5.2. Boundary Layer Problem*

We consider a boundary layer solution resulting from a singular perturbation. The solution is essentially one-dimensional, but we regularize the solution by adding a constant $p+1$ derivative in the parallel direction, i.e.

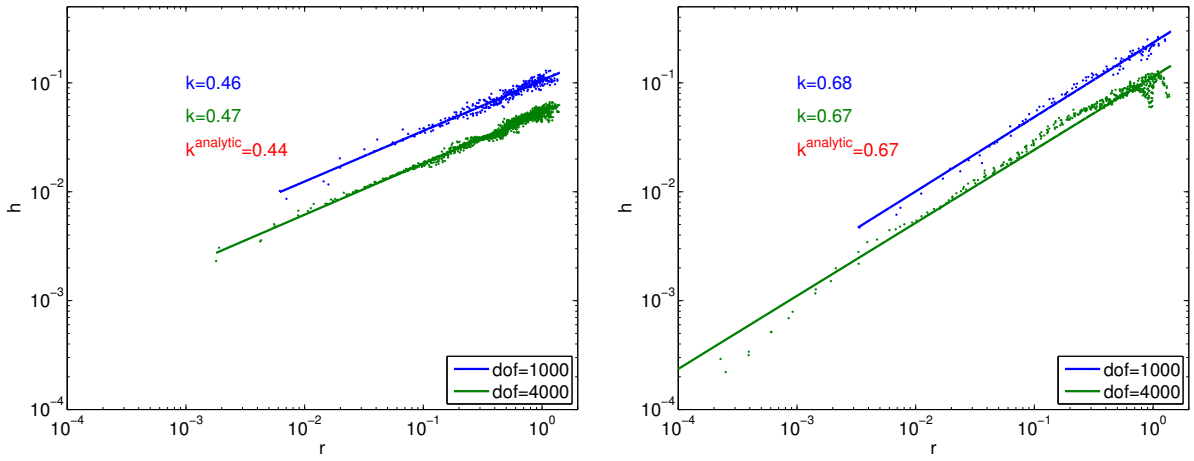
$$u(x_1, x_2) = \exp\left(-\frac{x_1}{\epsilon}\right) + \frac{\beta}{(p+1)!} x_2^{p+1},$$

where $\epsilon$ is the characteristic length of the singular perturbation, $\beta$ is the regularization constant, and $x_1$ and $x_2$ are the coordinates perpendicular and parallel, respectively, to the boundary.

*5.2.1. Analytical Solution*

We employ anisotropic interpolation theory and calculus of variations to find the optimal anisotropic element size distribution. Because all cross derivatives vanish for the function $u$ considered, the error functional takes on a particularly simple form

$$\begin{aligned}\mathcal{E}(h_1, h_2) &= \int_\Omega h_1^{2(p+1)} |u_{x_1}^{(p+1)}|^2 + h_2^{2(p+1)} |u_{x_2}^{(p+1)}|^2 dx_1 dx_2 \\ &= \int_\Omega h_1^{2(p+1)} \left(-\frac{1}{\epsilon}\right)^{2(p+1)} \exp\left(-\frac{2x_1}{\epsilon}\right) + h_2^{2(p+1)} \beta^2 dx_1 dx_2.\end{aligned}$$




(a) $p = 1$ (b) $p = 3$

Figure 3: The element size ($h$) vs. the distance of the element centroid from the corner ($r$) for the optimized meshes for the corner singularity problem with $\alpha = 2/3$. The lines and equations shown are the least-squares fit in $\log(h)$ vs. $\log(r)$.

The cost functional is

$$\mathcal{C}(h_1, h_2) = \int_\Omega C_p h_1^{-1} h_2^{-1} dx.$$

For convenience, let us express the solution in terms of $h_1$ and the aspect ratio, $\mathcal{AR} = h_2/h_1$, instead of in terms of $h_1$ and $h_2$.

Finding the stationary point of the Lagrangian formed by the error and cost functionals, the optimality condition for $h_1$ is

$$h_1 = C_{p,\epsilon} \exp(k_1^{\text{analytic}} x_1)$$

with the optimal characteristic thickness given by

$$\delta = \frac{1}{k_1^{\text{analytic}}} = \epsilon \left(p + \frac{3}{2}\right) \left(1 - \frac{1}{4p^2 + 12p + 9}\right).$$

We note that this optimal characteristic thickness is close to that of the one-dimensional boundary layer problem, $\delta_{1\text{d}} = \epsilon(p + 3/2)$. Unlike the corner singularity case, the optimal mesh grading decreases as $p$ increases.

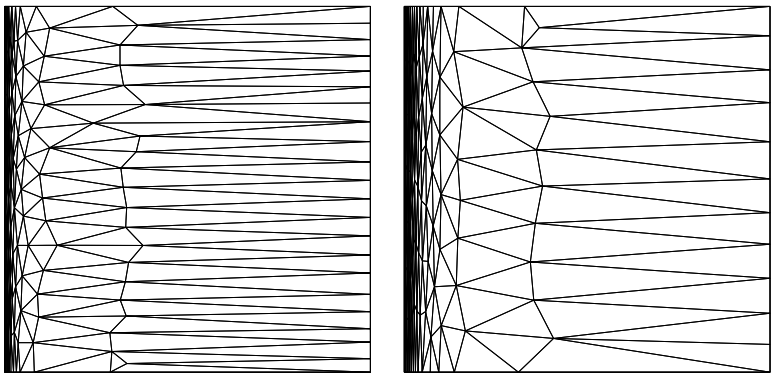
Similarly, finding the stationary point of the Lagrangian, we find the optimal aspect ratio distribution for this problem as

$$\mathcal{AR}^{\text{analytic}} = \mathcal{AR}_0^{\text{analytic}} \exp(k_{\mathcal{AR}}^{\text{analytic}} x_1)$$

with the aspect ratio at the root, $\mathcal{AR}_0$, and the grading factor, $k_{\mathcal{AR}}$, given by

$$\mathcal{AR}_0^{\text{analytic}} = \frac{1}{\beta^{\frac{1}{p+1}} \epsilon} \quad \text{and} \quad k_{\mathcal{AR}}^{\text{analytic}} = -\frac{1}{\epsilon(p+1)}.$$

Note that the maximum aspect ratio is achieved on the boundary, and it decreases exponentially away from the boundary.



(a) $p = 1$ (b) $p = 3$

Figure 4: Examples of optimized boundary layer meshes for $p = 1$ and $p = 3$ containing approximately 200 elements.

| | $k_1$ | $\mathcal{AR}_0$ | $k_{\mathcal{AR}}$ |
|---|---|---|---|
| $p = 1$ numerical | 41.9 | 43.9 | -50.3 |
| $p = 1$ analytical | 41.7 | 50.0 | -50.0 |
| $p = 3$ numerical | 22.7 | 46.8 | -23.3 |
| $p = 3$ analytical | 22.5 | 50.0 | -25.0 |

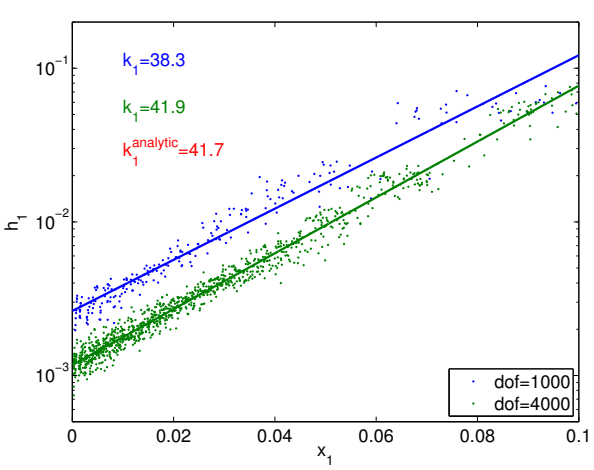
Table 1: Summary of the optimized mesh parameters.

*5.2.2. Numerical Solution*

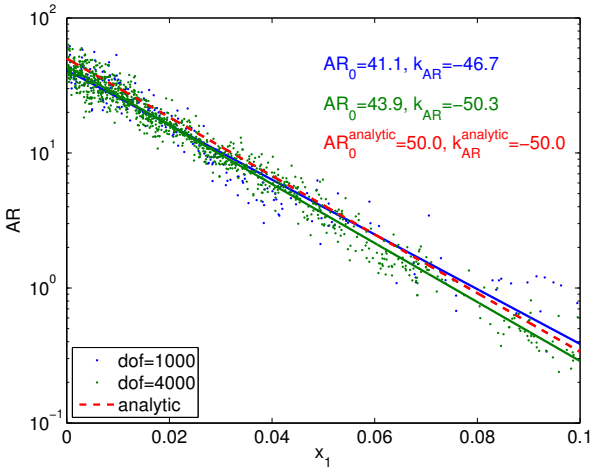
We apply the proposed optimization algorithm to the boundary layer problem with $\epsilon = 1/100$ and $\beta = 2^{p+1}$, which results in $\mathcal{AR}_0 = 50$. Figure 4 shows examples of $p = 1$ and $p = 3$ optimized meshes. Each mesh contains approximately 200 elements. Highly anisotropic elements are employed to resolve the boundary layer on the left boundary. Visually, the $p = 3$ optimized mesh exhibits a weaker $h_1$ grading toward the boundary layer, as predicted by the analytical result.

Again, a more quantitative assessment of the optimized meshes is provided by Figure 5. The $h_1$ and $h_2$ value for each element is computed by first calculating the elemental implied metric $\mathcal{M}_\kappa$, and then taking $h_1 = (\mathcal{M}_\kappa)_{11}^{-1/2}$ and $h_2 = (\mathcal{M}_\kappa)_{22}^{-1/2}$. Figure 5(a) shows the distribution of $h_1$ against the distance from the boundary $x_1$ in log-linear scale for $p = 1$ discretization with 1000 and 4000 degrees of freedom. The distribution is essentially linear in the $\log(h_1)$-$x_1$ space, and the least-squares fit in the space shows that the grading factor in the direction perpendicular to the boundary is $k_1 = 41.9$ (for dof = 4000), which agrees with the analytical optimal value of $k_1^{\text{analytic}} = 41.7$. Figure 5(b) shows the aspect ratio distribution, and the least-squares fit in the $\log(\mathcal{AR})$-$x_1$ space. The aspect ratio at the boundary obtained using the algorithm is $\mathcal{AR}_0 = 43.9$, which is slightly lower than the analytical result of $\mathcal{AR}_0^{\text{analytic}} = 50.0$; however, the values are still in good agreement. The negative grading away from the boundary of $k_{\mathcal{AR}} = 50.3$ matches closely with that of analytical result, $k_{\mathcal{AR}}^{\text{analytic}} = 50.0$. The comparison of the analytical and numerical mesh parameters is summarized in Table 1.

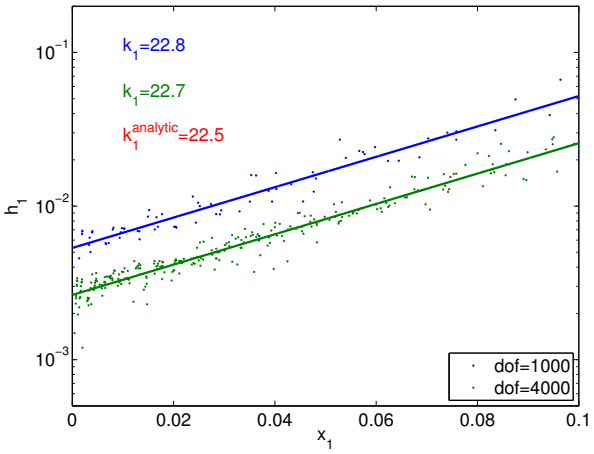
Figure 5(c) and 5(d) show the same $\log(h_1)$-$x_1$ and $\log(\mathcal{AR})$-$x_1$ analysis for $p = 3$ discretization. The grading for $h_1$ and $\mathcal{AR}$ are weaker for $p = 3$ than for $p = 1$, which is consistent with the theory. All parameters of the optimized meshes match well with those of analytical results. Again, without relying on the *a priori* error convergence behavior or the solution Hessian (or a higher derivative equivalent), our algorithm deduces the optimal anisotropic mesh distribution.



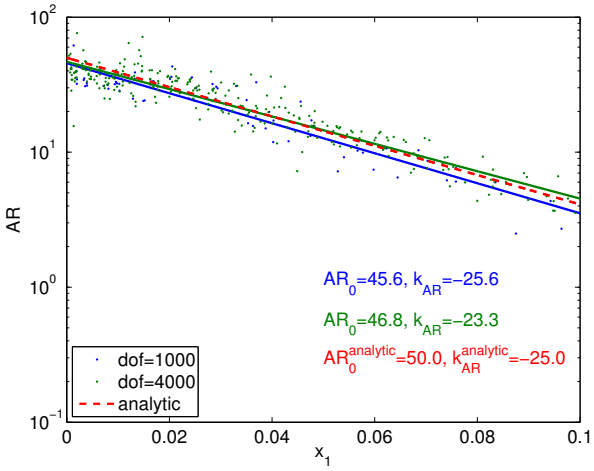

(a) $h_1$, $p = 1$




(b) $\mathcal{AR}$, $p = 1$




(c) $h_1$, $p = 3$




(d) $\mathcal{AR}$, $p = 3$

Figure 5: The element size in the perpendicular direction, $h_1$, and the aspect ratio distribution, $\mathcal{AR} = h_2/h_1$, for the boundary layer problem with $\epsilon = 0.01$ and $\beta = 2^{p+1}$.

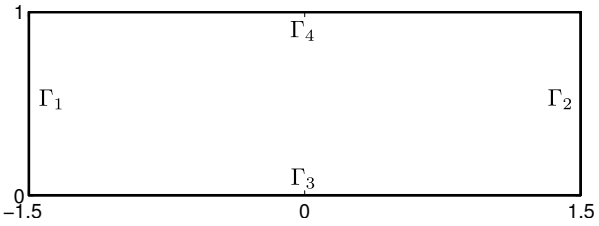



Figure 6: The domain for the advection-diffusion cases.

## 6. Output-Based Adaptation: Advection-Diffusion Problems

### *6.1. Setup*

We consider an application of the proposed adaptation framework to the advection-diffusion equation in a rectangular domain $\Omega \equiv [-1.5, 1.5] \times [0, 1]$ shown in Figure 6. The governing equation is given by

$$\nabla \cdot (\beta u) - \nabla \cdot (\epsilon \nabla u) = f \quad \text{in} \quad \Omega,$$

where $\beta \in \mathbb{R}^2$ defines the advection field, $\epsilon \in \mathbb{R}^+$ is the viscosity, and $f$ is the source function. For all the problems considered, we set $\beta = [1, 0]$ and $\epsilon = 10^{-3}$, so that the Peclet number is $10^3$. The boundary conditions are given by

$$\begin{aligned} -(\beta \cdot n)u + \epsilon \frac{\partial u}{\partial n} &= 0, \quad \text{on } \Gamma_1 \\ \epsilon \frac{\partial u}{\partial n} &= 0, \quad \text{on } \Gamma_2 \\ u &= u_{\Gamma_3}, \quad \text{on } \Gamma_3 \\ u &= 0, \quad \text{on } \Gamma_4, \end{aligned}$$

where the boundaries $\Gamma_i$, $i = 1, \ldots, 4$, are as specified in Figure 6, and $u_{\Gamma_3}$ specifies the solution value on $\Gamma_3$. The general form of the output functional considered is expressed as,

$$\mathcal{J}(u) = \int_\Omega g_\Omega u ds + \int_{\Gamma_3} g_{\Gamma_3} \epsilon \frac{\partial u}{\partial n} ds,$$

where $g_{\Gamma_3}$ and $g_\Omega$ are the two parameters that characterize the output. For the specified form of the output and the boundary conditions, the dual problem is given by

$$-\beta \cdot \nabla \psi - \nabla \cdot (\epsilon \nabla u) = g_\Omega \quad \text{in} \quad \Omega$$

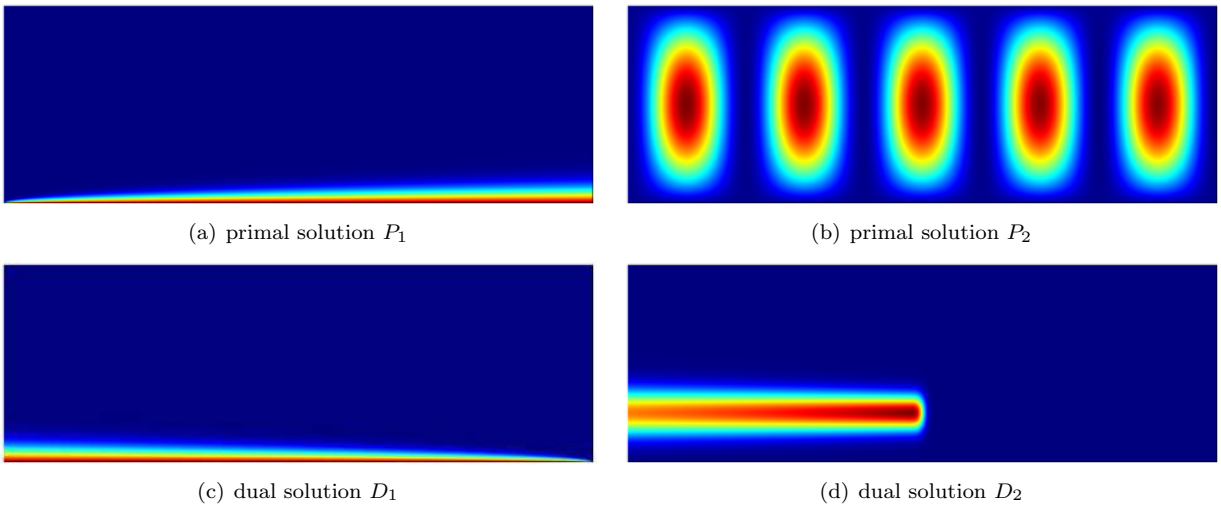
with the boundary conditions

$$\begin{aligned} \epsilon \frac{\partial \psi}{\partial n} &= 0, \quad \text{on } \Gamma_1 \\ (\beta \cdot n)\psi + \epsilon \frac{\partial \psi}{\partial n} &= 0, \quad \text{on } \Gamma_2 \\ \psi &= g_{\Gamma_3}, \quad \text{on } \Gamma_3 \\ \psi &= 0, \quad \text{on } \Gamma_4. \end{aligned}$$

We will consider three different combinations of the source function $f$, the boundary value $u_{\Gamma_3}$, and the output functional parameters $g_\Omega$ and $g_{\Gamma_3}$ to produce primal and dual solutions suitable for assessing the optimization algorithm. The choice of problem parameters and the corresponding primal and dual solutions are summarized in Table 2. A pair of primal solutions, $P_1$ and $P_2$, and a pair of dual solutions, $D_1$ and

| | Primal-Dual | Dual-Only | Primal-Only |
|---|---|---|---|
| $f$ | 0 | $-\sin\left(\frac{10\pi}{3}x_1\right)\sin(\pi x_2)$ | 0 |
| $u_{\Gamma_3}$ | 1 | 0 | 1 |
| $g_\Omega$ | 0 | 0 | $\frac{1}{2\pi(0.0012)}\exp\left(-\frac{1}{2}[\frac{x_1^2}{0.02^2}+\frac{(x_2-0.25)^2}{0.06^2}]\right)$ |
| $g_{\Gamma_3}$ | 1 | 1 | 0 |
| Primal Solution | $P_1$ | $P_2$ | $P_1$ |
| Dual Solution | $D_1$ | $D_1$ | $D_2$ |

Table 2: The set of parameters defining the three advection-diffusion problems. The solution identifications correspond to those in Figure 7.



(a) primal solution $P_1$ (b) primal solution $P_2$

(c) dual solution $D_1$ (d) dual solution $D_2$

Figure 7: Solutions to the boundary layer problems.

$D_2$, are shown in Figure 7. The first problem is called "primal-dual," as the choice of parameters induces a boundary layer in both primal and dual solutions ($P_1$ and $D_1$). The second problem is called "dual-only," as it exhibits a boundary layer in the dual solution ($D_1$) but not in the primal solution ($P_2$). Similarly, the third problem is called "primal-only," as a boundary layer appears only in the primal solution ($P_1$) and not in the dual solution ($D_2$).

### *6.2. Assessment Procedure*

In order to assess the effectiveness of the proposed mesh optimization method, we compare the approach with two different strategies. First is the isotropic refinement based on the DWR error estimate. Second is the anisotropic refinement that uses the DWR error estimate for the area decision and the primal solution for the shape decision. Specifically, the method solves the $p+1$ discretization of the primal problem, takes the first principal direction in the direction of the maximum $p+1$ derivative, selects the second principal length to equidistribute the interpolation error in the two principal directions, and scales the principal lengths to achieve the desired area. The detailed implementation of the algorithm is presented in [28]. We emphasize that both of these approaches use the adjoint-based error estimate; the primary difference in the methods lies in the anisotropy decision process.

For each of the advection-diffusion problems, the solutions are obtained using the $p = 2$ discretization at 500, 1000, and 2000 degrees of freedom. The reference solution is obtained using the $p = 3$ discretization at 40,000 degrees of freedom.

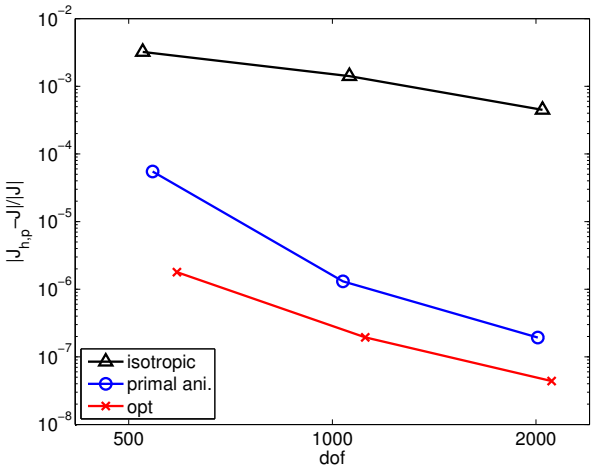
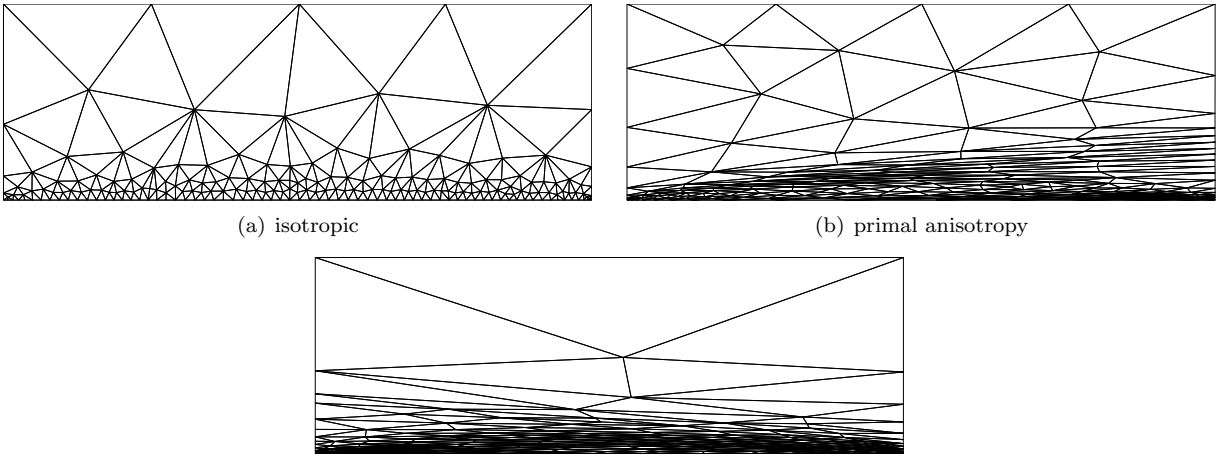



Figure 8: Error convergence to the primal-dual boundary layer problem using $p = 2$ discretization.



(a) isotropic

(b) primal anisotropy

(c) optimization

Figure 9: Adapted meshes for the primal-dual boundary layer problem. ($p = 2$, dof $= 2000$)

### *6.3. Primal-Dual Boundary Layer*

The primal-dual boundary layer problem exhibit boundary layer in both primal ($P_1$) and dual ($D_1$) solution, as shown in Figure 7. The output error convergence for the three adaptation schemes is shown in Figure 8. The proposed anisotropic mesh optimization algorithm reduces the error by over three orders of magnitude compared to isotropic adaptation, even for the moderate Peclet number of $10^3$. The advantage of the anisotropic boundary layer resolution further increases for higher Peclet number cases.

For this problem, the primal-based anisotropy detection is expected to perform well because the region that the primal and dual solutions require high anisotropy match with each other. Thus, targeting the primal anisotropic feature coincidentally results in resolving the dual anisotropic feature. Even then, Figure 8 shows that the optimization based approach outperforms the primal-based anisotropy detection, requiring approximately half of the degrees of freedom for a given output error level.

The final meshes at 2000 degrees of freedom are shown in Figure 9. Because the primal and dual solutions are mirror image of each other about $x_2 = 0$, the isotropic adaptation produces a mesh whose size functions are symmetric about $x_2 = 0$, as shown in Figure 9(a). Recalling that the output error is a (weighted) product of the primal and dual errors, the symmetry of the mesh (and hence the equal level of

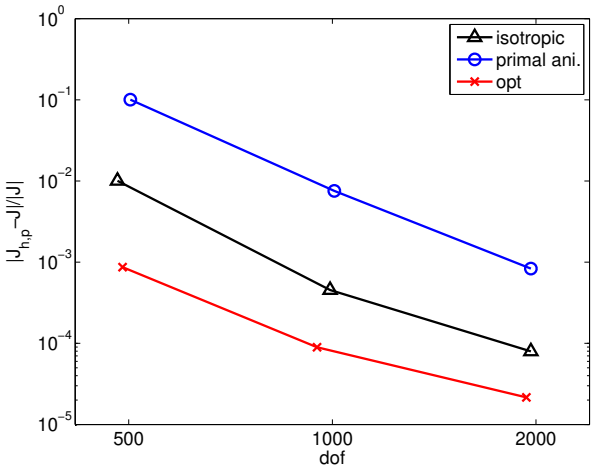



Figure 10: Error convergence to the dual-only boundary layer problem using $p = 2$ discretization.

the resolution of primal and dual solutions) agrees with our intuition. On the other hand, Figure 9(b) shows that the primal-based anisotropy results in a scheme that is biased toward resolving the directional features in the primal solution. The biased-treatment of the primal solution suggests that the element anisotropy is not optimal. However, the primal-based anisotropy detection results in over 60% of the elements having aspect ratio over 10 and 20% having the aspect ratio over 30, contributing to the efficient resolution of the boundary layer and outperforming the isotropic adaptation. Figure 9(c) shows that our optimization algorithm produces a mesh whose size and anisotropy distributions are symmetric about $x_2 = 0$. This is not surprising, as the method is driven completely by the behavior of the *a posteriori* error estimate and automatically balances the influences of the primal and adjoint solutions for this case. On the optimized mesh, over 80% of the elements have aspect ratio of over 10 and 20% have the aspect ratio of over 30.

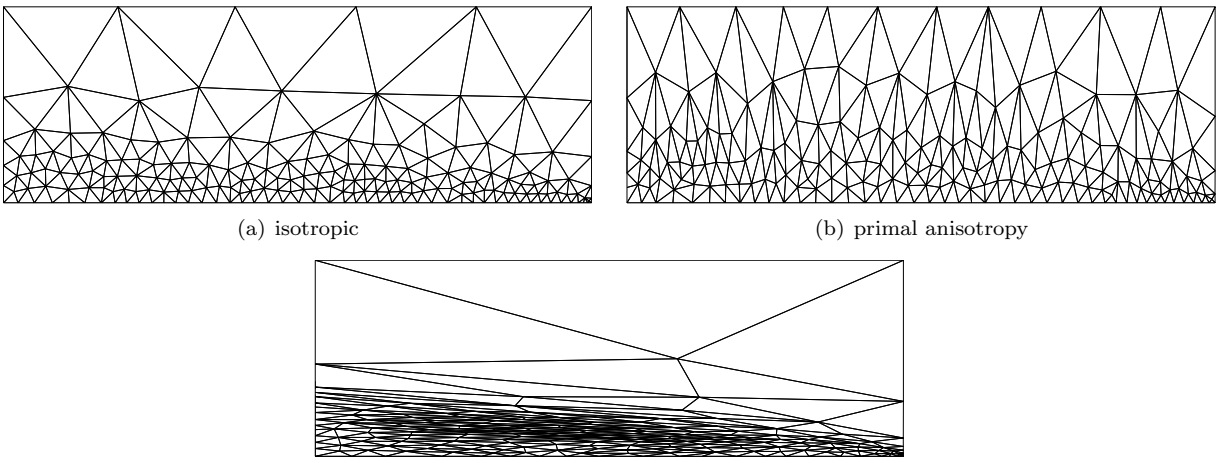
### *6.4. Dual-Only Boundary Layer*

The dual-only boundary layer problem produces a boundary layer in the dual solution ($D_1$) but not in the primal solution ($P_2$), as shown in Figure 7. Figure 10 shows the output error convergence for the three adaptation schemes. The proposed mesh optimization method requires approximately half the degrees of freedom of the isotropic adaptation to achieve a given error tolerance. For this problem, the primal-based anisotropy detection performs worse than the isotropic adaptation.

Figure 11 shows the final meshes at 2000 degrees of freedom. The isotropic adaptation targets the boundary layer in the dual solution; however, its efficiency is limited due to the use of isotropic elements. The primal-based anisotropy detection produces inappropriate anisotropy in the boundary layer, resulting in the method performing worse than the isotropic adaptation. The proposed optimization method targets the dual boundary layer using anisotropic elements. However, because the primal solution is not anisotropic near the bottom wall, the elements are not as anisotropic as those in the primal-dual boundary layer problem. In particular, the fraction of elements with the aspect ratio of over 30 are only 6% for this case, compared to over 20% for the primal-dual boundary layer case. The result again demonstrates that the proposed method automatically balances the resolution of the primal and dual solution to minimize the output error.

### *6.5. Primal-Only Boundary Layer*

The primal-only boundary layer problem considers a regularized line output, and produces a boundary layer in the primal solution ($P_1$) but not in the dual solution ($D_2$), as shown in Figure 7. The output error convergence for the primal-only boundary layer problem is shown in Figure 12. The proposed optimization approach performs significantly better than the isotropic adaptation or the primal-based anisotropy detection.



(a) isotropic

(b) primal anisotropy

(c) optimization

Figure 11: Adapted meshes for the dual-only boundary layer problem. ($p = 2$, dof $= 2000$)

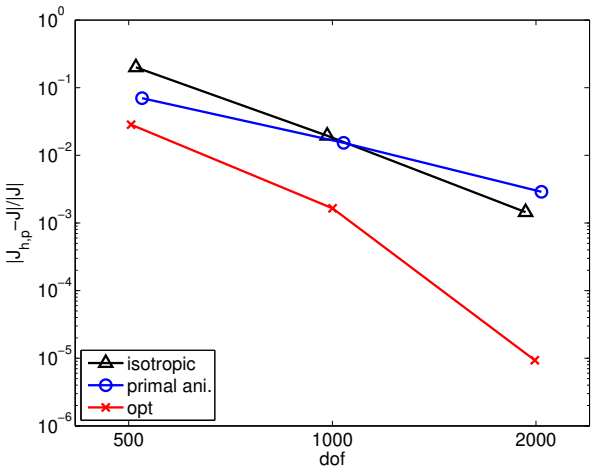
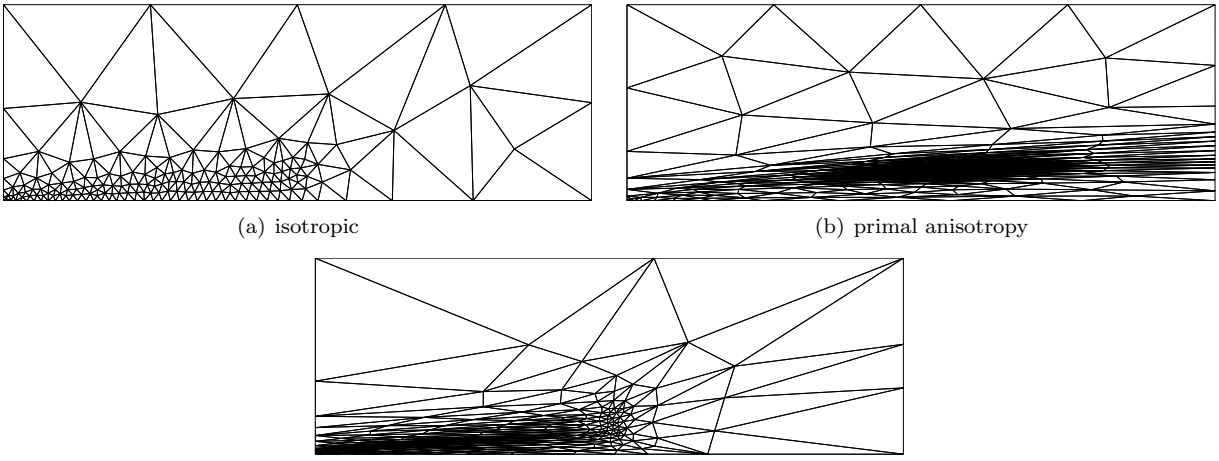



Figure 12: Error convergence to the primal-only boundary layer problem using $p = 2$ discretization.



(a) isotropic

(b) primal anisotropy

(c) optimization

Figure 13: Adapted meshes for the primal-only boundary layer problem. ($p = 2$, dof $= 2000$)

Figure 13 shows the final meshes obtained using the three adaptation strategies. Figure 13(a) shows that the isotropic adaptation targets the region of overlap between the primal boundary layer and dual shear layer; however, similar to the other two cases, its efficiency is limited due to the use of isotropic elements.

Figure 13(b) shows that the primal-based anisotropy detection employs anisotropic elements suitable for resolving the primal boundary layer. Because its sizing decision is based on DWR, smaller elements are employed in vicinity of the dual source term and elements downstream of the source location are large. We note that the apparent refinement in the region downstream the source is due to the use of highly anisotropic elements. However, these anisotropic elements aligned with the primal boundary layer are unsuited for resolving the dual solution that exhibit strong variation in the $x_1$-direction just downstream of the source. The poor performance of the primal-based anisotropy detection shows that anisotropy only suited for resolving the primal solution is in fact worse than no anisotropy at all even for the problem that exhibit strong anisotropic features in the primal solution.

The proposed optimization approach balances the anisotropy requirements for resolving the primal and dual solutions, as shown in Figure 13(c). In the region upstream of the dual source, both the primal and dual solution exhibit strong variation in the $x_2$-direction, and the algorithm resolves these features using highly anisotropic elements. However, in the vicinity of the Gaussian source, the algorithm uses isotropic elements. This is a result of balancing the $x_2$ resolution for the primal solution and $x_1$ resolution for the dual solution.

## 7. Conclusions

We present an optimization framework for anisotropic simplex mesh adaptation. Our method leverages and supplements the recent advances in the discretization and error estimation technologies, extending the DWR-based adaptation to anisotropic problems. The framework builds on a number of concepts recently introduced including: a continuous mesh optimization perspective of Loseille et al. [10]; a local error sampling technique based on local solves introduced by Georgoulis et al. [12]; and an affine-invariant metric tensor manipulation framework of Pennec et al. [11]. A combination of these advances lead to a development of a general adaptation framework based on parameterization of a mesh, selection of error and cost models, identification of the model parameters, and optimization of the surrogate model. Our method inherits the versatility of the underlying discretization and error estimate, handles any discretization order, and adapts to arbitrarily oriented anisotropic features.

A pair of canonical $L^2$ error control problems verify that the proposed method produces optimal meshes, whose parameters match the optimal element size distributions analytically obtained using anisotropic in-

terpolation theory and calculus of variations. For both the isotropic and anisotropic verification cases, the method automatically deduces the optimal meshes for different polynomial orders without an *a priori* knowledge of the error convergence behavior.

Application of the framework to output error control demonstrates the ability of the method to effectively balance the primal and dual errors by appropriately choosing element anisotropy in a fully-automatic manner based on the *a posteriori* error estimate. The proposed method outperforms other DWR-driven adaptation methods with isotropic refinement and primal-based anisotropy detection for all cases considered. Considering the encouraging results for the simple problems, we expect the proposed framework to bring further benefits in more complex problems encountered in engineering and science, where both the primal and dual solutions exhibit complex, anisotropic behaviors that are not known *a priori*.

Lastly, we note that the proposed framework based on mesh parametrization, error and cost model selection, sampling-based model parameter identification, and surrogate-model optimization is a general formulation that can be applied to much wider classes of adaptation problems than those solved in this work. First, our algorithm extends to three-dimensional problems in a straightforward manner. In three dimensions, the metric tensor is described by six entries and the local error sampling would be performed on (at least) six configurations obtained by splitting different edges of a tetrahedron; the error synthesis and optimization procedures are dimension-independent, requiring no modifications. Another extension is *hp* adaptation, in which the mesh is parameterized by not just a metric field but also a polynomial order field. This would require an additional sampling strategy that measures error sensitivity to the solution order. Another practical example is combining the method with structured (or layered) boundary layer mesher that only offers limited mesh control parameters (e.g. grading ratio). In this case, the error and cost sensitivity with respect to the metric field can be mapped to the mesh control parameters via chain rule, and we can perform optimization on the control parameters. Assuming a suitable error estimate is available, the framework also provides an unified treatment of time-dependent problems with varying degrees of space-time mesh flexibility (e.g. method of lines, Rothe method, and fully unstructured). Depending on the type of space-time mesh, the elemental error and cost sensitivity can be mapped to an appropriate set of mesh control parameters. Extension of the framework, such as those mentioned above, are the areas of ongoing research.

**Acknowledgements**

We would like to thank the ProjectX team for the many contributions during the course of this work. This work was supported by the Singapore-MIT Alliance Fellowship in Computational Engineering and The Boeing Company with technical monitor Dr. Mori Mani.

## A. Properties of the Adaptation Algorithm

### *A.1. Relationship between Step Matrix and the Change in Approximability*

In designing our surrogate error model and optimization algorithm, we advocated the use of the step matrix $S$ (either elemental or vertex) rather than using the metric tensor $\mathcal{M}$ directly. This is because the magnitude of the entries of a step matrix $S$ is closely related to the change in the anisotropic approximability of the space associated with $\mathcal{M}_0$ and $\mathcal{M}(S) \equiv \mathcal{M}_0^{1/2} \exp(S) \mathcal{M}_0^{1/2}$, as stated in Section 3.2. Here we prove the relationship Eq. (5) between the change in the anisotropic approximability and the entries of the step matrix $S$.

The change in the approximability in a given direction, or the ratio of the directional lengths between the configurations induced by $\mathcal{M}_0$ and $\mathcal{M}(S)$, is

$$\frac{h(e;\mathcal{M}(S))}{h(e;\mathcal{M}_0)} = \left( \frac{e^T \mathcal{M}_0 e}{e^T \mathcal{M}_0^{1/2} \exp(S) \mathcal{M}_0^{1/2} e} \right)^{1/2}.$$

The lower bound of the ratio, i.e. the maximum increase in the approximability, is related to the eigenvalues of $S$ by

$$\min_{e\in\mathbb{R}^d\setminus 0}\frac{h(e;\mathcal{M}(S))}{h(e;\mathcal{M}_0)} = \min_{e\in\mathbb{R}^d\setminus 0}\left(\frac{e^T\mathcal{M}_0 e}{e^T\mathcal{M}_0^{1/2}\exp(S)\mathcal{M}_0^{1/2}e}\right)^{1/2} = \min_{f\in\mathbb{R}^d\setminus 0}\left(\frac{f^T f}{f^T\exp(S)f}\right)^{1/2}$$
$$= (\lambda_{\max}(\exp(S)))^{-1/2} = \exp\left(-\frac{1}{2}\lambda_{\max}(S)\right),$$

where $\lambda_{\max}(S)$ denotes the maximum eigenvalue of $S$. Similarly, the upper bound of the ratio can be expressed as

$$\max_{e\in\mathbb{R}^d\setminus 0}\frac{h(e;\mathcal{M}(S))}{h(e;\mathcal{M}_0)} = \exp\left(-\frac{1}{2}\lambda_{\min}(S)\right),$$

where $\lambda_{\min}(S)$ denotes the minimum eigenvalue of $S$. Thus, we can control the maximum increase or decrease in the approximability by controlling the maximum and minimum eigenvalue of $S$, respectively. In particular, because

$$\lambda_{\min}^2(S) \le \|S\|_F^2 \quad \text{and} \quad \lambda_{\max}^2(S) \le \|S\|_F^2,$$

the magnitude of the entries in $S$ is a good indicator of the maximal change in the approximability in moving from $\mathcal{M}_0$ to $\mathcal{M}(S)$. Thus, expressing the manipulation in terms of the step tensor $S \in Sym_d$ and mapping the tensor to $\mathcal{M}(S) \in Sym_d^+$ via the exponential map not only eliminates the potential of generating a null-tensor but also provides a convenient means of controlling the change in the anisotropic approximability.

*A.2. Invariance of the Sampling Quality*

One of the important features of the proposed error model and sampling strategy is that the quality of the error reconstruction does not degrade on highly anisotropic elements. Recalling the error reconstruction operates on the step matrices $\{S_{\kappa_i}\}_{i=1}^{n_{\text{config}}}$ in the tangent space, the property requires that the set of step matrices does not become degenerate on a highly anisotropic configuration. In fact, we will show that $\{S_{\kappa_i}\}_{i=1}^{n_{\text{config}}}$ are invariant with respect to the current configuration $\mathcal{M}_{\kappa_0}$ up to orthogonal transformation, which does not influence the quality of reconstruction. The invariance is a consequence of the local coordinate system induced by the affine-invariant metric. We note that, if the error reconstruction is performed directly using the coefficients of the metric tensor, the error reconstruction would become ill-posed as the step tensors $\mathcal{M}_{\kappa_0} - \mathcal{M}_{\kappa_i}$ becomes degenerate on highly anisotropic elements.

Let us denote the metric tensor associated with the unit reference element by $\widehat{\mathcal{M}}_0$. By definition, $\widehat{\mathcal{M}}_0 = I$. Let us denote the mapping of the unit reference element to an element obtained by the $i$-th local mesh operation of the reference element by $\widehat{J}_i$. The tensor corresponding to the split reference element is

$$\widehat{\mathcal{M}}_i = \widehat{J}_i^{-T}\widehat{\mathcal{M}}_0\widehat{J}_i^{-1}.$$

The step tensor from $\widehat{\mathcal{M}}_0$ to $\widehat{\mathcal{M}}_i$ is

$$\widehat{S}_i = \log(\widehat{\mathcal{M}}_0^{-1/2}\widehat{\mathcal{M}}_i\widehat{\mathcal{M}}_0^{-1/2}) = \log(\widehat{\mathcal{M}}_i)$$

Let us now consider the step tensor from an arbitrary configuration $\mathcal{M}_{\kappa_0}$ to the configuration obtained by the $i$-th local mesh operation, $\mathcal{M}_{\kappa_i}$. Let us denote the mapping from the unit reference triangle, $\widehat{\mathcal{M}}_0$, to $\mathcal{M}_{\kappa_0}$ by $J$ and the singular value decomposition of $J$ by $J = U\Sigma V^T$. Then, $\mathcal{M}_{\kappa_0}$ can be expressed as

$$\mathcal{M}_{\kappa_0} = J^{-T}\widehat{\mathcal{M}}_0 J^{-1} = (U\Sigma^{-1}V^T)I(V\Sigma^{-1}U^T) = U\Sigma^{-2}U^T.$$

Similarly, using the mapping $J$, we can express the configuration obtained by $i$-th mesh operation as

$$\mathcal{M}_{\kappa_i} = J^{-T}\widehat{\mathcal{M}}_i J^{-1} = U\Sigma^{-1}V^T\widehat{\mathcal{M}}_i V\Sigma^{-1}U^T.$$

The step matrix from $\mathcal{M}_{\kappa_0}$ to $\mathcal{M}_{\kappa_i}$ is

$$\begin{aligned} S_{\kappa_i} &= \log(\mathcal{M}_{\kappa_0}^{-1/2}\mathcal{M}_{\kappa_i}\mathcal{M}_{\kappa_0}^{-1/2}) \\ &= \log((U\Sigma^{-2}U^T)^{-1/2}(U\Sigma^{-1}V^T\widehat{\mathcal{M}}_i V\Sigma^{-1}U^T)(U\Sigma^{-2}U^T)^{-1/2}) \\ &= UV^T\log(\widehat{\mathcal{M}}_i)VU^T = (VU^T)^T\widehat{S}_i(VU^T) \end{aligned}$$

The step matrix from $\mathcal{M}_{\kappa_0}$ to $\mathcal{M}_{\kappa_i}$ is related to the step matrix from $\widehat{\mathcal{M}}_0$ to $\widehat{\mathcal{M}}_i$ by the orthogonal transformation induced by $VU^T$. Thus, as long as the samples $\{\widehat{\mathcal{M}}_i\}_{i=1}^{n_{\text{config}}}$ are chosen such that the linear error reconstruction problem is well-posed on the reference element, the linear fitting problem on $\mathcal{M}_{\kappa_0}$ is well-posed. In other words, the quality of the error model reconstruction is preserved even on high aspect ratio elements encountered in anisotropic adaptation.

*A.3. Inclusion of the Isotropic Error Model*

As mentioned in Section 4.4, our anisotropic error model $\eta_\kappa(S_\kappa) = \eta_{\kappa_0}\exp(\text{tr}(R_\kappa S_\kappa))$ is a generalization of the familiar isotropic error relationship based on the power law,

$$\eta_\kappa^{\text{iso}}(h) = \eta_{\kappa_0}\left(\frac{h}{h_0}\right)^{r_\kappa^{\text{iso}}}, \tag{18}$$

where $r_\kappa^{\text{iso}}$ is the convergence rate. In particular, the behavior of the error model under isotropic scaling is consistent with that of the isotropic error model in the following sense. The isotropic metric $\mathcal{M}$ for mesh size $h$ is given by $\mathcal{M} = h^{-2}I$. The step tensor required to change from an isotropic tensor $\mathcal{M}_0 = h_0^{-2}I$ to $\mathcal{M} = h^{-2}I$ is

$$S_\kappa = \log\left(\mathcal{M}_0^{-1/2}\mathcal{M}\mathcal{M}_0^{-1/2}\right) = \log\left(h_0^2h^{-2}I\right) = -2\log\left(\frac{h}{h_0}\right)I$$

We note that the trace-free part $\tilde{S}_\kappa$ vanishes as expected, and the isotropic part is $s_\kappa = -2\log(h/h_0)$. Substitution of the step tensor into the local error model yields

$$\eta_\kappa\left(S_\kappa\right) = \eta_\kappa\left(-2\log\left(\frac{h}{h_0}\right)I\right) = \eta_{\kappa_0}\exp\left(-2dr_\kappa\log\left(\frac{h}{h_0}\right)\right) = \eta_{\kappa_0}\left(\frac{h}{h_0}\right)^{-2dr_\kappa}.$$

If we define $r_\kappa^{\text{iso}} = -2dr_\kappa$, then we recover the isotropic error relationship Eq. (18). Thus, our error model can be thought of as an extension of the scalar error model to anisotropic deformations.

*A.4. Invariance under Coordinate Transformation*

In this section, we show that the tensor field optimization algorithm presented is independent of the particular coordinate representation of the tensors. The property means that the same physical problem represented in two different coordinate systems would produce the identical sequences of the tensor fields with respect to the physical problem.

Let us consider two coordinate systems, $x$ and $\bar{x}$, that are related by the mapping

$$\bar{x} = g(x) = \alpha Ux + \bar{x}_0,$$

where $U$ is a $d \times d$ orthogonal matrix, $\alpha > 0$ is the coordinate scaling factor, and $\bar{x}_0 \in \mathbb{R}^d$ is the coordinate shift. A mesh defined in terms of $x$ can be represented in the coordinate system $\bar{x}$ by mapping each nodal

coordinate according to $\bar{x} = g(x)$. Then, the tensor field represented in the coordinate system $\bar{x}$, $\{\bar{\mathcal{M}}\}_{\bar{x}\in\Omega}$, is related to that of the coordinate system $x$, $\{\mathcal{M}\}_{x\in\Omega}$, by

$$\bar{\mathcal{M}}(\bar{x}) = \alpha^{-2} U \mathcal{M}(x) U^T.$$

Now let us work through the adaptation procedure and show that it is invariant under coordinate transformation.

The first step of adaptation is local sampling. The elemental step tensor in $\bar{x}$, $\bar{S}_{\kappa_i}$, is related to that in $x$, $S_{\kappa_i}$, by

$$\begin{aligned}
\bar{S}_{\kappa_i} &= \log\left(\bar{\mathcal{M}}_{\kappa_0}^{-1/2}\bar{\mathcal{M}}_{\kappa_i}\bar{\mathcal{M}}_{\kappa_0}^{-1/2}\right) \\
&= \log\left(\left(\alpha^{-2}U\mathcal{M}_{\kappa_0}U^T\right)^{-1/2}\left(\alpha^{-2}U\mathcal{M}_{\kappa_i}U^T\right)\left(\alpha^{-2}U\mathcal{M}_{\kappa_0}U^T\right)^{-1/2}\right) \\
&= U\log\left(\mathcal{M}_{\kappa_0}^{-1/2}\mathcal{M}_{\kappa_i}\mathcal{M}_{\kappa_0}^{-1/2}\right)U^T = US_{\kappa_i}U^T,
\end{aligned}$$

where we have identified the step matrix in $x$ coordinate system as $S_{\kappa_i} = \mathcal{M}_{\kappa_0}^{-1/2}\mathcal{M}_{\kappa_i}\mathcal{M}_{\kappa_0}^{-1/2}$. We also map the change in the error to the logarithmic space, i.e. $f_{\kappa_i} = \log(\eta_{\kappa_i}/\eta_\kappa)$. Here, because the two coordinate systems represent the same physical system, we assume that $\eta_{\kappa_i}/\eta_{\kappa_0}$ evaluates to the same value for all $i = 1, \ldots, n_{\text{config}}$ and $\kappa \in \mathcal{T}_h$, resulting in the same $\{f_{\kappa_i}\}_{i=1}^{n_{\text{config}}}$ for both coordinate systems. To identify the rate matrix in the transformed coordinate system, $\bar{R}_\kappa$, we solve the minimization problem

$$\bar{R}_\kappa = \underset{\bar{Q}\in Sym_d}{\arg\min} \sum_{i=1}^{n_{\text{config}}} \left(f_{\kappa_i} - \text{tr}(\bar{Q}\bar{S}_{\kappa_i})\right) = \underset{\bar{Q}\in Sym_d}{\arg\min} \sum_{i=1}^{n_{\text{config}}} \left(f_{\kappa_i} - \text{tr}\left(\bar{Q}US_{\kappa_i}U^T\right)\right).$$

Recalling $R_\kappa$ in the original coordinate system is the solution to

$$R_\kappa = \underset{Q\in Sym_d}{\arg\min} \sum_{i=1}^{n_{\text{config}}} \left(f_{\kappa_i} - \text{tr}\left(QS_{\kappa_i}\right)\right).$$

and noting that similarity transforms do not alter the value of trace, we immediately recognize the solution to the minimization problem on the transformed coordinate is related to that of the original coordinate by

$$\bar{R}_\kappa = UR_\kappa U^T.$$

As a result, the two error models are identical in the sense that

$$\bar{\eta}_\kappa(\bar{S}_\kappa) = \eta_{\kappa_0}\exp\left(\text{tr}\left(\bar{R}_\kappa\bar{S}_\kappa\right)\right) = \eta_{\kappa_0}\exp\left(\text{tr}\left(UR_\kappa U^T US_\kappa U\right)\right) = \eta_{\kappa_0}\exp\left(\text{tr}\left(R_\kappa S_\kappa\right)\right) = \eta_\kappa(S_\kappa).$$

Similarly, the cost model is identical because

$$\bar{\rho}_\kappa(\bar{S}_\kappa) = \rho_{\kappa_0}\exp\left(\frac{1}{2}\text{tr}(\bar{S}_\kappa)\right) = \rho_{\kappa_0}\exp\left(\frac{1}{2}\text{tr}(US_\kappa U^T)\right) = \rho_{\kappa_0}\exp\left(\frac{1}{2}\text{tr}(S_\kappa)\right) = \rho_\kappa(S_\kappa).$$

Finally, to create the new vertex representation of the metric field, we solve the optimization problem on the surrogate model. Recall that the optimization algorithm relies entirely on the gradient of the surrogate error and cost functions. Let us denote the surrogate error and cost functions in the transformed space by $\bar{\mathcal{E}}(\{\bar{S}_\nu\})$ and $\bar{\mathcal{C}}(\{\bar{S}_\nu\})$. Because the error and cost models are invariant under the coordinate transformation, their derivatives are related by simple coordinate transformations,

$$\frac{\partial\bar{\mathcal{E}}}{\partial\bar{S}_\nu} = U\frac{\partial\mathcal{E}}{\partial S_\nu}U^T \quad \text{and} \quad \frac{\partial\bar{\mathcal{C}}}{\partial\bar{S}_\nu} = U\frac{\partial\mathcal{C}}{\partial S_\nu}U^T.$$

Consequently, the proposed gradient descent algorithm produces the vertex step matrices in the transformed coordinate, $\{\bar{S}_\nu\}$, which are related to that solved in the original coordinate by

$$\bar{S}_\nu = U S_\nu U^T.$$

The exponential map of the step matrices in the transformed coordinate yields

$$\begin{aligned}\bar{\mathcal{M}}_\nu\left(\bar{S}_\nu\right) &= \bar{\mathcal{M}}_{\nu_0}^{1/2} \exp\left(\bar{S}_\nu\right) \bar{\mathcal{M}}_{\nu_0}^{1/2} \\ &= \left(\alpha^{-2} U \mathcal{M}_{\nu_0} U^T\right)^{1/2} \exp\left(U S_\nu U^T\right) \left(\alpha^{-2} U \mathcal{M}_{\nu_0} U^T\right)^{1/2} \\ &= \alpha^{-2} U \mathcal{M}_{\nu_0}^{1/2} \exp\left(S_\nu\right) \mathcal{M}_{\nu_0}^{1/2} U^T = \alpha^{-2} U \mathcal{M}\left(S_\nu\right) U^T.\end{aligned}$$

Because the relationship between $\bar{\mathcal{M}}_\nu\left(\bar{S}_\nu\right)$ and $\mathcal{M}\left(S_\nu\right)$ is identical to the transformation of the tensor for the two coordinate systems, the two updated tensors $\{\mathcal{M}_\nu\}_{\nu\in\mathcal{V}}$ and $\{\bar{\mathcal{M}}_\nu\}_{\nu\in\mathcal{V}}$ represent the same physical tensor fields. Thus, our adaptation algorithm is invariant under coordinate transformation.

## B. Comparison of Discrete and Continuous Optimization

## References


[1] M. Ainsworth, J. T. Oden, A Posteriori Error Estimation in Finite Element Analysis, John Wiley and Sons, 2000.

[2] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, in: A. Iserles (Ed.), Acta Numerica, Cambridge University Press, 2001.

[3] J. Peraire, M. Vahdati, K. Morgan, O. C. Zienkiewicz, Adaptive remeshing for compressible flow computations, J. Comput. Phys. 72 (1987) 449–466.

[4] M. J. Castro-Díaz, F. Hecht, B. Mohammadi, O. Pironneau, Anisotropic unstructured mesh adaptation for flow simulations, Internat. J. Numer. Methods Fluids 25 (1997) 475–491.

[5] L. Formaggia, S. Perotto, P. Zunino, An anisotropic a-posteriori error estimate for a convection-diffusion problem, Comput. Visual Sci. 4 (2001) 99–104.

[6] L. Formaggia, S. Micheletti, S. Perotto, Anisotropic mesh adaptation with applications to CFD problems, in: H. A. Mang, F. G. Rammerstorfer, J. Eberhardsteiner (Eds.), Fifth World Congress on Computational Mechanics, Vienna, Austria, 2002.

[7] D. A. Venditti, D. L. Darmofal, Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows, J. Comput. Phys. 187 (1) (2003) 22–46.

[8] K. J. Fidkowski, D. L. Darmofal, A triangular cut-cell adaptive method for higher-order discretizations of the compressible Navier-Stokes equations, J. Comput. Phys. 225 (2007) 1653–1672.

[9] T. Leicht, R. Hartmann, Anisotropic mesh refinement for discontinuous Galerkin methods in two-dimensional aerodynamic flow simulations, Internat. J. Numer. Methods Fluids 56 (2008) 2111–2138.

[10] A. Loseille, F. Alauzet, Continuous mesh model and well-posed continuous interpolation error estimation, INRIA RR-6846 (2009).

[11] X. Pennec, P. Fillard, N. Ayache, A Riemannian framework for tensor computing, Int. J. Comput. Vision 66 (1) (2006) 41–66.

[12] E. H. Georgoulis, E. Hall, P. Houston, Discontinuous Galerkin methods on *hp*-anisotropic meshes II: A posteriori error analysis and adaptivity, Appl. Numer. Math. 59 (2009) 2179–2194.

[13] M. Ceze, K. J. Fidkowski, Output-driven anisotropic mesh adaptation for viscous flows using discrete choice optimization, AIAA 2010–170 (2010).

[14] T. Richter, *A posteriori* error estimation and anisotropy detection with the dual-weighted residual method, Internat. J. Numer. Methods Fluids 62 (2010) 90–118.

[15] T. Leicht, R. Hartmann, Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations, J. Comput. Phys. 229 (2010) 7344–7360.

[16] M. A. Park, Anisotropic output-based adaptation with tetrahedral cut cells for compressible flows, PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics (2008).

[17] H. Sun, Impact of triangle shapes using high-order discretizations and direct mesh adaptation for output error, Masters thesis, Massachusetts Institute of Technology, Computation for Design and Optimization Program (2009).

[18] A. Brandt, Multi-level adaptive solutions to boundary-value problems, Math. Comp. 31 (138) (1977) 333–390.

[19] R. Rannacher, Adaptive Galerkin finite element methods for partial differential equations, Journal of Computational and Applied Mathematics 128 (2001) 205–233.

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

[21] J. Lu, An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts (2005).

[22] R. Hartmann, Adjoint consistency analysis of discontinuous Galerkin discretizations, SIAM J. Numer. Anal. 45 (6) (2007) 2671–2696.
[23] T. A. Oliver, A higher-order, adaptive, discontinuous Galerkin finite element method for the Reynolds-averaged Navier-Stokes equations, PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics (Jun. 2008).
[24] R. Becker, R. Rannacher, A feed-back approach to error control in finite element methods: Basic analysis and examples, East-West J. Numer. Math. 4 (1996) 237–264.
[25] F. Hecht, Bamg: Bidimensional anisotropic mesh generator, `http://www-rocq1.inria.fr/gamma/cdrom/www/bamg/eng.htm` (1998).
[26] M. Yano, D. Darmofal, An optimization framework for anisotropic simplex mesh adaptation: application to aerodynamic flows, AIAA 2012–0079 (Jan. 2012).
[27] C. Schwab, *p*- and *hp*- Finite Element Methods, Oxford Science Publications, Great Clarendon Street, Oxford, UK, 1998.
[28] M. Yano, J. M. Modisette, D. Darmofal, The importance of mesh adaptation for higher-order discretizations of aerodynamic flows, AIAA 2011–3852 (Jun. 2011).