Improving High-Order Finite Element Approximation Through Geometrical Warping

Devina P. Sanjaya* and Krzysztof J. Fidkowski†

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA

Polynomial basis functions are the ubiquitous workhorse of high-order finite element methods, but their generality comes at a price of high computational cost and fragility in the face of under-resolution. In this paper we present a method for constructing a posteriori tailored, generally non-polynomial, basis functions for approximating a solution. This method is similar to solution-based adaptation, in which elements of the computational mesh are sized and oriented based on characteristics of the solution. The method takes advantage of existing infrastructure in high-order methods: the reference-to-global mapping used in constructing curved elements. By optimizing this mapping, we "warp" elements to make them ideally suited for representing a target solution or computing a scalar output from the solution. For scalar advection-diffusion and compressible Navier-Stokes problems, we show that such warped elements can offer significant accuracy benefits without increasing degrees of freedom in the system.

I. Introduction

High-order finite element methods, such as discontinuous Galerkin, offer accuracy benefits for many problems due to their reliance on high-order polynomial functions for representing the solution. Polynomials have excellent approximation properties, at least for smooth functions, and when accuracy is important, high-order approximation can beat low-order approximation in terms of degrees of freedom and even computational cost.[1]

However, polynomial approximation is not always the best choice. High accuracy requirements may necessitate very high polynomial orders that make solutions computationally intractable. In addition, certain features of the solution may be too under-resolved on a given mesh for robust polynomial approximation. One solution is adaptation, in particular of the hp variety in which mesh elements (h) are refined where high order is not advantageous.[2,3,4,5] Another option is to tailor finite element basis functions to the problem at hand. This idea has been recognized in numerous previous works, including the partition of unity method,[6] the extended finite element method,[7] isogeometric analysis,[8] and the discontinuous enrichment method.[9] Tailoring basis functions a priori is possible for some problems but it is hard in general, especially for complex flows in which the locations of features such as shocks and shear layers is not known ahead of time. On the other hand, tailoring basis functions a posteriori is a more robust alternative, one that we pursue in this work.

Our proposed approach uses basis functions parametrized by reference-to-global mappings used in the definition of curved elements. Indeed, many high-order methods already do not employ global-space high-order polynomials. Polynomials are used on a reference element, but the reference-to-global mapping distorts the approximation. This distortion was recognized previously and an approach was designed to correct it via a "shadow map".[10] In this work we take an alternate position and embrace the distortion produced by the mapping. That is, we attempt to tune the reference-to-global coordinate maps in a mesh to produce elements that are customized for representing a particular solution. We refer to this process as element "warping" because we are changing the (internal) shape of an element. The mapped basis functions will no longer constitute a complete polynomial set in global space; instead for a given solution order p, the basis functions will contain certain high-order modes that enable accurate approximation of the target solution.

*Graduate Research Assistant, AIAA Member

†Associate Professor, AIAA Senior Member

This is not the only way to create parametrized basis functions, but it is one that uses machinery (i.e. curved element mappings) already available in many high-order codes. Our eventual goal is to combine this method with *hp* output-based adaptation to create customized approximation spaces geared for predicting a desired output to high accuracy.

The outline for the remainder of this paper is as follows. Section II reviews the discontinuous Galerkin (DG) finite element discretization, with particular emphasis on solution approximation and curved elements. Section III introduces the idea of intentionally curving the interior structure of an element to improve approximation for a given solution order, and Section IV presents our approach for optimizing the associated reference-to-global coordinate transformation. Section V shows results obtained from this method, and Section VI presents conclusions and plans for further work.

II. A Discontinuous Finite Element Discretization

The idea of warping an element to improve its approximation properties can be applied to any method that supports high-order curved elements. We focus on the discontinuous Galerkin (DG) method because we have experience with it and because it is a relatively mature high-order method suitable for convection-dominated flows that are prevalent in aerospace engineering, our target application. In this section, we present the discretization, with particular attention to the curved-element treatment.

II.A. Conservation Law

Consider a conservation law given by the partial differential equation (PDE)

$$\partial_t \mathbf{u} + \nabla \cdot \vec{\mathbf{H}}(\mathbf{u}, \nabla \mathbf{u}) \quad = \quad \mathbf{0}, \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^s$ is the state vector, $\vec{\mathbf{H}} \in \mathbb{R}^{d \times s}$ is the total flux, s is the state rank, and d is the spatial dimension. We decompose the flux into convective and diffusive parts via $\vec{\mathbf{H}} = \vec{\mathbf{F}}(\mathbf{u}) + \vec{\mathbf{G}}(\mathbf{u}, \nabla \mathbf{u})$, where $\vec{\mathbf{G}}(\mathbf{u}, \nabla \mathbf{u}) = -\underline{\mathbf{K}}(\mathbf{u}) \nabla \mathbf{u}$ is the viscous flux and $\underline{\mathbf{K}} \in \mathbb{R}^{d^2 \times s^2}$ is the viscous diffusivity tensor.

II.B. Solution Approximation

DG,[11,12,10] as a finite element method, approximates the state $\mathbf{u}$ in functional form using linear combinations of basis functions on each element. No continuity constraints are imposed on the approximations on adjacent elements. Denoting by T_h the set of N_e elements in a non-overlapping tessellation of the domain Ω, as illustrated in Figure 1, the state on element e is approximated as

$$\mathbf{u}_h(\vec{x})\Big|_{\Omega_e} = \sum_{n=1}^{N_p} \mathbf{U}_{en} \phi_{en}^{\text{glob}}(\vec{x}). \tag{2}$$

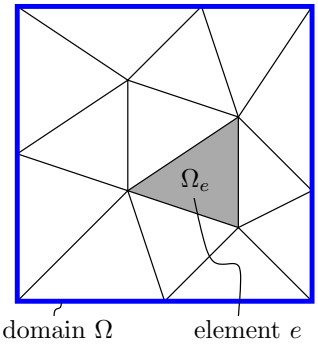


Figure 1. A partition of a square domain into triangular elements.

In this equation, N_p is the number of basis functions per element, $\mathbf{U}_{en}$ is the vector of s coefficients for the n^{th} basis function on element e: $\phi_{en}^{\text{glob}}(\vec{x})$. Formally, we write $\mathbf{u}_h \in \boldsymbol{\mathcal{V}}_h = [\mathcal{V}_h]^s$, where, if the elements are not curved,

$$\mathcal{V}_h = \{u \in L_2(\Omega) : u|_{\Omega_e} \in \mathcal{P}^p \ \forall \Omega_e \in T_h\},$$

and $\mathcal{P}^p$ denotes polynomials of order p on the element. A caveat here is that for elements that are curved, the polynomial approximation is usually performed on a master reference element, so that following the reference-to-global mapping, the state approximation on curved elements is not strictly of order p. We will take advantage of this observation when we optimize the curved element shape to yield better approximation properties compared to polynomials.

II.C. The Weak Form

We obtain a weak form of Eqn. 1 by multiplying the PDE by test functions $\mathbf{v}_h \in \boldsymbol{\mathcal{V}}_h$ and integrating by parts to couple elements via interface fluxes. We use the Roe scheme[13] for convective fluxes and the second

form of Bassi and Rebay (BR2)[14] for viscous fluxes. We can write the final semilinear weak form as

$$\mathcal{R}_h(\mathbf{u}_h, \mathbf{v}_h) = \sum_{e=1}^{N_e} \mathcal{R}_h(\mathbf{u}_h, \mathbf{v}_h|_{\Omega_e}) = 0, \quad \forall \mathbf{v}_h \in \boldsymbol{\mathcal{V}}_h, \tag{3}$$

where

$$\begin{aligned}\mathcal{R}_h(\mathbf{u}_h, \mathbf{v}_h|_{\Omega_e}) \;=\; & \int_{\Omega_e} \mathbf{v}_h^T \partial_t \mathbf{u}_h \, d\Omega - \int_{\Omega_e} \nabla \mathbf{v}_h^T \vec{\mathbf{H}} \, d\Omega + \int_{\partial\Omega_e \setminus \partial\Omega} \mathbf{v}_h^{+T} \left(\widehat{\mathbf{F}} + \widehat{\mathbf{G}} \right) ds + \int_{\partial\Omega_e \cup \partial\Omega} \mathbf{v}_h^{+T} \left(\widehat{\mathbf{F}}^b + \widehat{\mathbf{G}}^b \right) ds \\ & - \int_{\partial\Omega_e \setminus \partial\Omega} \nabla \mathbf{v}_h^{+T} \underline{\mathbf{K}}^+ \left(\mathbf{u}_h^+ - \widehat{\mathbf{u}}_h \right) \vec{n} \, ds - \int_{\partial\Omega_e \cup \partial\Omega} \nabla \mathbf{v}_h^{+T} \underline{\mathbf{K}}^b \left(\mathbf{u}_h^+ - \mathbf{u}_h^b \right) \vec{n} \, ds,\end{aligned} \tag{4}$$

where $(\cdot)^T$ denotes transpose, $\widehat{\mathbf{u}}_h = (\mathbf{u}_h^+ + \mathbf{u}_h^-)/2$ is the unique state on an interior face, and on the element boundary $\partial\Omega_e$ the notations $(\cdot)^+, (\cdot)^-, (\cdot)^b$ denote quantities taken from the element interior, neighbor element, and domain boundary, respectively. The last two terms symmetrize the weak form for adjoint consistency. Finally, we note that on boundary faces, fluxes are computed directly from the boundary state, $\mathbf{u}^b$, which is a projection of the interior state and the boundary-condition data, $\mathbf{u}_h^b = \mathbf{u}_h^b(\mathbf{u}_h^+, \mathrm{BC})$.

II.D. Curved Elements

An element is geometrically linear if its shape is defined by the location of its primary vertices. For example, in two dimensions, three points define a triangle and four points define a quadrilateral. In between the vertices the geometry is interpolated (bi/tri)-linearly. Such elements are simple to work with but, when used on curved domain boundaries, they do not approximate the boundary well enough for use with high-order solution approximation in DG.[15] A remedy is to curve the elements by equipping each element with additional geometry information, typically in the form of extra "high-order" geometry nodes.

A standard and relatively simple way to curve elements is to use a polynomial mapping from the reference element to the global element, as illustrated in Figure 2. The formula for the mapping function is given in

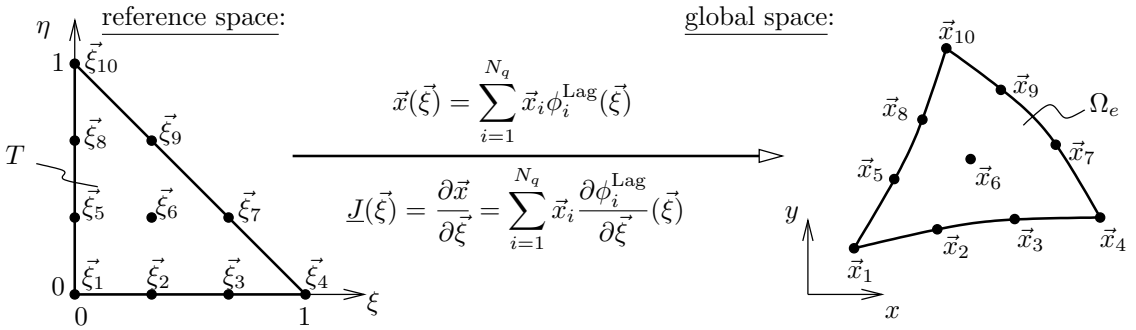


Figure 2. Example of a mapping from a unit reference triangle to a curved-element in global space. The mapping functions for the global coordinates x, y (rolled into $\vec{x}$) are polynomials of order $q = 3$ in this case, for a total of $N_q = 10$ geometry nodes.

the figure, where q is the order of this polynomial, $N_q = (q+1)(q+2)/2$ is the total number of degrees of freedom in the mapping, $\vec{\xi} = [\xi, \eta]^T$ is the coordinate in reference space, and $\vec{x} = [x, y]^T$ is the coordinate in global space. Using Lagrange basis functions, $\phi_i^{\mathrm{Lag}}(\vec{\xi})$, in the mapping allows for an intuitive specification of the high-order element: the coordinates of the N_q nodes $\vec{x}_i$ fully define the mapping function, and $\vec{x}(\vec{\xi}_i) = \vec{x}_i$.

The coordinates $\vec{x}_i$ should be chosen consistently with the corresponding reference-space nodes, $\vec{\xi}_i$, which are equally spaced on the reference element. For example, in Figure 2, $\vec{\xi}_6$ is the centroid of the reference triangle, so $\vec{x}_6$ should be located somewhere in the middle of the curved element. On edges/faces that are on domain boundaries, these nodes are typically on the geometry. However, these requirements do not pin down their locations, and heuristics or quality metrics such as maximizing the Jacobian determinant are often used in high-order node placement. In the next section we discuss another choice: a figure of merit based on accurate solution approximation.

III. Warping High-Order Curved Elements

Curved elements are primarily used on domain boundaries to accurately define a geometry for use with high-order solution approximation. For highly-anisotropic boundary-layer meshes, curved elements are generally also needed inside the domain to prevent elements from "popping through" each other and creating negative volumes. Such curving is performed out of necessity in creating a valid mesh, driven ultimately by geometry representation requirements on the domain boundary.

Typically not much attention is paid to the precise location of the high-order nodes, with the exception of those that have to lie on the boundary. Instead, heuristics often dictate locations that in some sense maximize the validity of the element, i.e. smoothly-varying coordinates with no clustering of nodes. In this section we present the idea of deliberately "warping" an element by moving high-order nodes to possibly-clustered locations to optimize solution approximation.

Figure 3 illustrates the concept of element warping. Of interest is the location of a curved element's

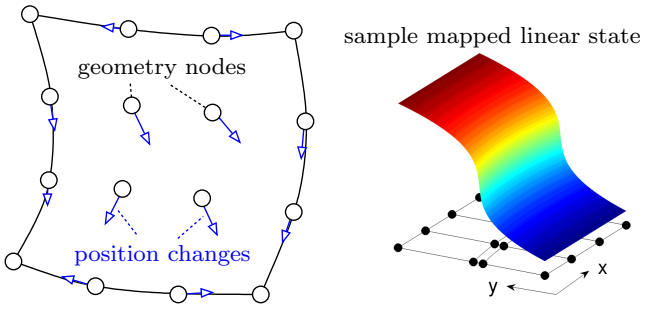


Figure 3. Schematic of high-order element *warping*, which consists of intentional placement of high-order interior nodes to improve an element's approximation power for a particular solution.

geometry nodes, which dictate the mapping from reference space to global space. By moving these nodes, we can change the behavior of an approximated function, i.e. the state, in the global space. For example, consider a function that is a linear polynomial in the reference space coordinates. This is what we typically refer to as a $p = 1$ solution approximation, since basis functions are most easily defined in reference space. If an element is geometrically linear, so that the reference-to-global mapping is affine, the $p = 1$ function remains linear in global space. However, if the element is curved, the mapped function will not necessarily remain linear in the global coordinates. Figure 3 illustrates this schematically for a $q = 3$ quadrilateral in which the middle nodes are placed close to each other, so that a $p = 1$ function in reference space develops a shear-layer type of structure in global space.

In general, for arbitrary curved elements, a function that is an order p polynomial in reference space does not remain an order p polynomial, or even a polynomial at all, in the global space coordinates. Specifically, a polynomial basis function in reference space, $\phi^{\text{ref}}(\vec{\xi})$, maps to a global basis function according to

$$\phi^{\text{glob}}(\vec{x}(\vec{\xi})) = \phi^{\text{ref}}(\vec{\xi}),$$

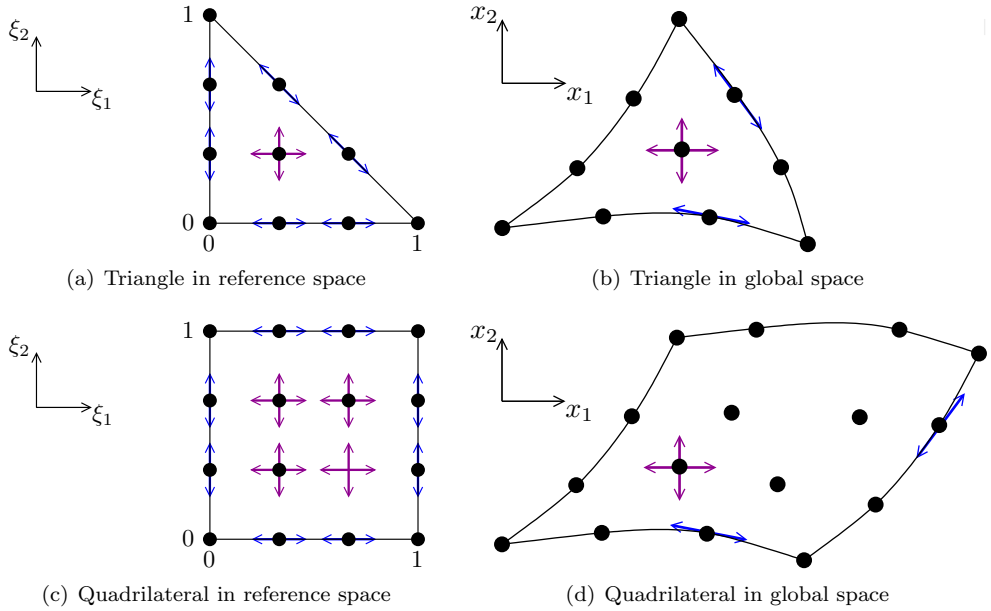
where $\vec{x}(\vec{\xi})$ is the geometry mapping given in Figure 2. Moving an element's high order geometry nodes changes this mapping and gives us control over the appearance of the high-order basis functions. Our goal is to optimize these global basis functions for the approximation of a particular solution, and we describe this optimization in the next section. Prior to moving on, however, we note that we are effectively working with a parametrized set of basis functions, where the parameters are the high-order geometry nodes. For a large q, we have many parameters, and we expect to be able to design "custom" basis functions that will allow us to accurately represent a solution even with low order p. We expect increasing q to be computationally more desirable compared to increasing p, since the size of the system of equations is independent of q.

IV. Warp Optimization

In the previous section, we introduced the idea that warping an element can change its approximation capabilities. In this section, we describe our approach to optimize the warp of an element by moving its high-order geometry nodes to optimal locations.

IV.A. Design Variables

In order to keep computational costs low, we presently make the optimization problems local to each element. To minimize the influence of one element's optimization on its neighbor elements, we constrain the movement of the high-order geometry nodes so as not to affect the element shape (much). Thus, we do not move an element's primary vertices (3 for a triangle) and we do not move edge nodes perpendicular to the edge. Figure 4 illustrates the allowable motions of nodes in $q = 3$ triangles and quadrilaterals.



(a) Triangle in reference space

(b) Triangle in global space

(c) Quadrilateral in reference space

(d) Quadrilateral in global space

Figure 4. Example of triangular and quadrilateral elements in reference and global spaces with geometry order $q = 3$. In order to keep the element shape mostly unchanged during optimization of the mapping, vertex nodes are non-movable, edge nodes can move in one direction, and element-interior nodes can move in two directions.

For optimization we need to choose the design variables. The global coordinates $\vec{x}_i$ of the mapped nodes are an obvious choice, but they are not ideal because they allow for arbitrary deformation. We would still have to impose the constraints that, for example, edge nodes move only along the edge, and this is hard to do in global space for curved elements. Instead, we turn to the reference element: we hold the global nodes $\vec{x}_i$ fixed, but we vary/optimize the *reference space* coordinates, $\vec{\xi}_i$, corresponding to these nodes. Normally, using an equally-spaced nodal Lagrange basis for the reference-to-global mapping, the $\vec{\xi}_i$ are just evenly distributed on the reference element, with horizontal/vertical spacing of $1/(q+1)$. During optimization, we change the positions of the $\vec{\xi}_i$ in reference space, where imposing the edge motion constraint is trivial. As these $\vec{\xi}_i$ still map to the fixed $\vec{x}_i$, the element must warp.

The design variables are the allowable displacements of each $\vec{\xi}_i$ in reference space. Call $\boldsymbol{\delta}$ the vector of allowable displacements. The size of this vector is $q^2 - 1$ for triangles and $2(q^2 - 1)$ for quadrilaterals. By the end of optimization, and during it for convenience of code-reuse when calling certain functions, we must express the shape of the element using the standard equally-spaced Lagrange basis. We do this by solving the following linear system,

$$\boldsymbol{\Phi}\, \mathbf{x}^{\text{new}} = \mathbf{x}^{\text{old}},$$

where $\boldsymbol{\Phi}$ is an $N_q \times N_q$ matrix with entries $\Phi_{ij} = \phi_j^{\text{Lag}}(\vec{\xi}_i')$, $\vec{\xi}_i'$ are the displaced reference-space coordinates of all the nodes, $\mathbf{x}^{\text{old}}$ is an $N_q \times d$ matrix of the original global-space node coordinates (one node per row), and $\mathbf{x}^{\text{new}}$ is an $N_q \times d$ matrix of the desired new global-space coordinates.

For multiple elements, we perform the optimization on each element independently and then average the (global-space) displacements of nodes that are shared between elements. In practical cases, optimization is not applied to every element, but rather only to those elements with the largest errors.

IV.B. Objective Function

Our goal here is to create a metric for measuring an element's approximation power, for use in optimization and as an error indicator telling us which elements in a mesh need to be warped. We consider the following two objective functions.

IV.B.1. Least-Squares Error

Suppose that the exact solution, $u^{\text{exact}}(\vec{x})$, were known. This is the case when testing with manufactured solutions, while for practical cases we could consider a solution or reconstruction in a higher-order space (e.g. $p+1$). For a scalar problem ($s=1$), the least-squares error, ε_e, on element Ω_e is defined via

$$(\varepsilon_e^{LS})^2 = \min_{u_h \in \mathcal{V}_h} \int_{\Omega_e} \left[u_h(\vec{x}) - u^{\text{exact}}(\vec{x})\right]^2 dA = \min_{u_h \in \mathcal{V}_h} \int_{E} \left|\underline{J}(\vec{\xi})\right| \left[u_h(\vec{\xi}) - u^{\text{exact}}(\vec{x}(\vec{\xi}))\right]^2 dE, \tag{5}$$

where $\underline{J}(\vec{\xi})$ is the mapping Jacobian matrix, E is the reference-space element, and the minimization is taken over all possible u_h in the solution approximation space $\mathcal{V}_h$. The integral in reference space is evaluated using Gauss quadrature with N_g Gauss points, $\vec{\xi_g}$, and weights, w_g. Note that integrating in reference space allows us to pre-compute and reuse evaluations of the basis functions at the quadrature points. Furthermore, negative Jacobian determinants encountered during integration indicate infeasible regions of the design space to the optimizer. Finally, for systems of equations, the least-squares error can be computed for each state component separately, and summed if a single number is desired, though this introduces dependence on arbitrary variable scaling.

IV.B.2. Computing Outputs of a System

In aerospace applications, we often deal with systems of equations ($s>1$) and we generally only care about one or several outputs instead of the solution everywhere. In this case, reducing the state approximation error everywhere in the domain via the least-squares error metric can be inefficient since the state may not need to be accurate everywhere to predict accurate outputs. A more efficient approach, and one that naturally handles systems, is to use an output-based error measurement, as described in this section.

Output-based methods rely on the solution of an output adjoint, which acts as a weight on the residual to produce the error estimate and adaptive indicator.[3] The discrete adjoint solution, $\mathbf{\Psi}$, satisfies

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]^T \mathbf{\Psi} + \frac{\partial \mathcal{J}}{\partial \mathbf{U}}^T = \mathbf{0}, \tag{6}$$

where $\mathcal{J}$ is the scalar output of interest. Solving this equation yields coefficients of the adjoint solution, which can then be used to approximate the continuous adjoint, $\boldsymbol{\psi}(\vec{x})$.

A simple approach for incorporating output-based adjoint information into the objective function is to modify the least-squares error estimate in Eqn. 5 to use a weighted combination of primal and adjoint errors,

$$\varepsilon_e^{\mathcal{J}} = \frac{1}{2}\|\mathbf{R}\|_e^T \, \|\boldsymbol{\psi} - \boldsymbol{\psi}^{\text{fine}}\|_{LS} + \frac{1}{2}\|\mathbf{R}^{\psi}\|_e^T \, \|\mathbf{u} - \mathbf{u}^{\text{fine}}\|_{LS}, \tag{7}$$

where $\boldsymbol{\psi}^{\text{fine}}$ and $\mathbf{u}^{\text{fine}}$ are approximate/reconstructed fine-space ($p+1$) solutions, $\|\mathbf{R}\|_e$ is the fine-space primal residual norm on element e, and $\|\mathbf{R}^{\psi}\|_e$ is the fine-space adjoint residual norm on element e. The residuals are computed using primal and adjoint states prior to the optimization (i.e. $\boldsymbol{\delta} = \mathbf{0}$). Norms are taken separately for each equation for systems, and the subscript LS indicates the least-squares error – i.e. $\|\boldsymbol{\psi} - \boldsymbol{\psi}^{\text{fine}}\|_{LS}$ is error between the "truth" (fine) adjoint and its projection into the order p approximation space of the current warped element.

Eqn. 7 is motivated by the observation that low output error is achieved when both the primal and adjoint solutions are approximated well in an element, and the primal/ adjoint residuals indicate the relative importance of each and make the combination dimensionally consistent.

IV.C. Constraints

As mentioned previously, we need all Jacobian determinants to be non-negative to ensure that all elements in the mesh are valid and to obtain a physical solution. We find it more robust to go a step further and enforce a minimum Jacobian determinant over an element (measured at integration points). Thus, we impose the following constraint on element e:

$$c_e \equiv \frac{\min(|\underline{J}(\vec{\xi}_g)|)}{V_0} - \eta_V > 0, \tag{8}$$

where V_0 is the initial element volume and η_V is the prescribed non-dimensional minimum determinant of Jacobian as a fraction of the element volume.

IV.D. Optimization Problem

Now, we can formulate our constrained optimization problem on an element as follows:

$$\begin{aligned} &\text{minimize} && \frac{\varepsilon_e(\boldsymbol{\delta})}{\varepsilon_{e0}} \\ &\text{w.r.t.} && \boldsymbol{\delta} \\ &\text{subject to} && c_e, \end{aligned} \tag{9}$$

where we explicitly indicate the dependence of the error on the design variables $\boldsymbol{\delta}$, and where $\varepsilon_{e0} = \varepsilon_e(\boldsymbol{\delta}=\mathbf{0})$ is the initial error on the element. We solve this constrained optimization problem via an interior penalty method, by using an inverse barrier function with μ_b as the non-dimensional barrier penalty factor. This turns Eqn. 9 into the following unconstrained optimization problem on element e:

$$\begin{aligned} &\text{minimize} && \pi_e(\vec{\xi}(\boldsymbol{\delta}), \mu_b) = \frac{\varepsilon_e(\boldsymbol{\delta})}{\varepsilon_{e0}} + \mu_b \frac{V_0}{\min(|\underline{J}(\vec{\xi}_g)|) - \eta_V V_0} \\ &\text{w.r.t.} && \boldsymbol{\delta}. \end{aligned} \tag{10}$$

The solution to Eqn. 10 approaches that to Eqn. 9 as μ_b approaches zero. To keep computational cost low, optimization is only performed on a fraction, f^{adapt}, of elements with the largest error indicator. The optimization problem on each element is solved using a gradient-based method: the Broyden-Fletcher-Goldfarb-Shanno (BFGS)[16] algorithm with a line search. At each iteration of BFGS, a line search with strong Wolfe conditions is performed and μ_b is reduced until we find the optimum node locations. Currently, all gradients are calculated using a finite-difference approximation.

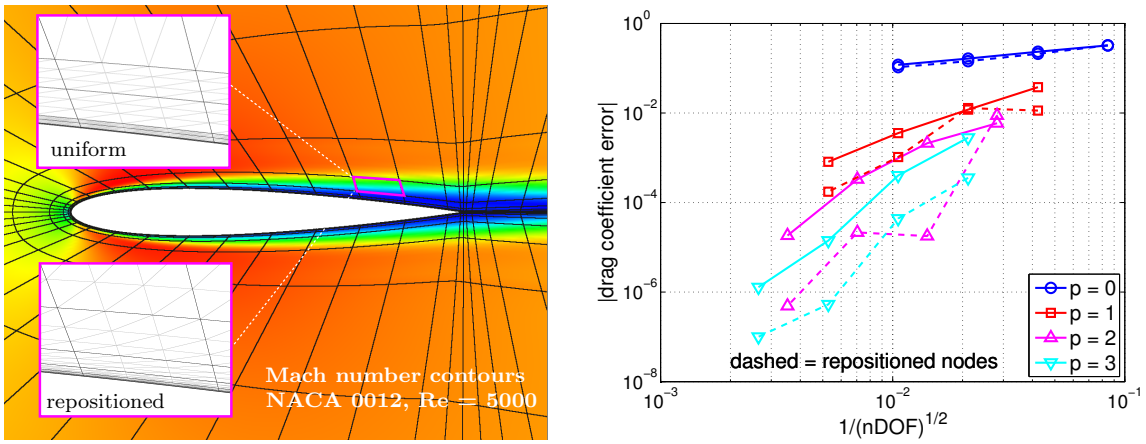
V. Results

V.A. Boundary Layer Approximation for a Laminar Airfoil

Before presenting the results of the node movement optimization, we offer a heuristic example of the potential benefit of moving high-order nodes. Consider a NACA 0012 airfoil in $M = 0.5$, $Re = 5000$ flow. In these conditions, a boundary layer (albeit not a very thin one) develops near the airfoil wall. Within this boundary layer, several flow properties change rapidly in the wall-normal direction, and an accurate representation of this boundary layer flow is important for predicting the drag.

We investigate two types of $q = 3$ meshes for calculating drag at several different p. The first mesh ("uniform") is one in which the high-order nodes are spaced uniformly in the elements. The second mesh ("repositioned") is one in which the high-order nodes are heuristically clustered towards the airfoil, which improves the ability of basis functions to accurately capture the rapid variation of flow quantities near the airfoil.

Figure 5 shows a sample flowfield and the convergence of drag for uniform refinements of the two families of meshes considered. For approximation orders $p = 1, 2, 3$ we see a benefit of using the meshes with the repositioned nodes. Specifically, the drag coefficient error drops by one or more orders of magnitude compared to the meshes with the uniformly-distributed high-order geometry nodes, for the same computational cost. These results are for uniform refinements of a single guessed repositioning: we expect further improvements from an optimization algorithm.



(a) Two types of curved elements

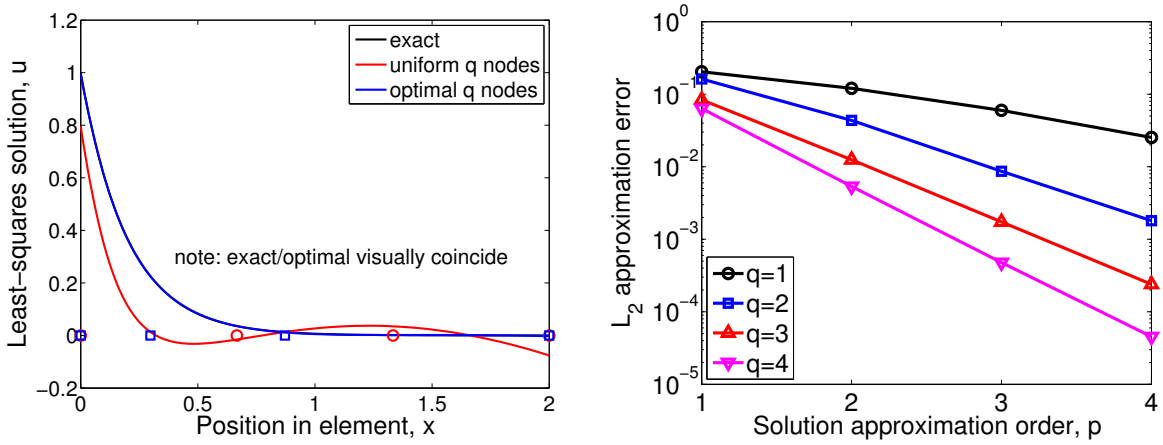
(b) Drag convergence for two mesh types

Figure 5. Motivating result of effect of interior node placement in high-order curved elements, using a DG discretization of compressible Navier-Stokes flow over an airfoil. Repositioning high-order nodes within each element towards the airfoil yields better approximation of the boundary layer and significantly improved drag estimates.

V.B. Single Element, One-dimensional Geometry Node Optimization

Next, we consider an L_2 error minimization problem in which we try to approximate an exact function $u^{\text{exact}}(x) = e^{-5x}$ over the interval $x \in [0, 2]$ using various solution orders p and geometry orders q. For each combination of p and q, we run an optimization algorithm that tunes the positions of the $q-1$ internal nodes in order to minimize the least squares error ε_e^{LS} in Eqn. 5. Note, for $q = 1$, the element cannot be warped and so there is nothing to tune.

Figure 6 shows the results of the optimization for all combinations of p and q tested. The $q = 1$ result can be taken as the baseline, unwarped, case. Relative to this, we see that for a given p, increasing q has a strong effect on the error. The benefit in error reduction improves with increasing p: for example, for $p = 4$,



(a) Sample approximations

(b) Convergence with p and q

Figure 6. L_2 error minimization for a single-element in one spatial dimension by optimal positioning of the high-order geometry nodes.

warping the element with $q = 4$ geometry versus the baseline $q = 1$ geometry decreases the L_2 error by over five orders of magnitude.

V.C. Single Element, Two-dimensional Geometry Node Optimization

Before we try to approximate a two-dimensional exact solution, we first try to approximate the same one-dimensional exact solution as in the previous section with two-dimensional geometry node movement on a

quadrilateral element. Figure 7 shows that even when the exact solution is strictly one dimensional, allowing some y-movement of the nodes results in further decrease in the L_2 error.

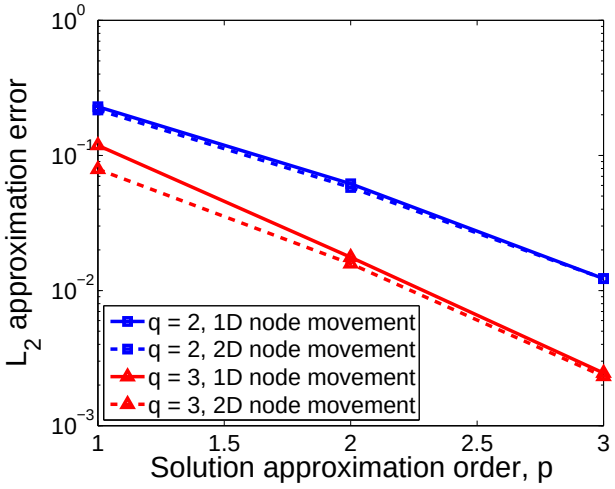


Figure 7. L_2 **error minimization for a function only of** x **on a single-element in two dimensions. The results compare one-dimensional node movement to two-dimensional node movement.**

Second, we consider a convective equation, $\nabla \cdot (\vec{V}u) + S = 0$, where u is a manufactured boundary layer solution $u^{\text{manufactured}}(x,y) = 1 - e^{-\frac{y}{\delta(x)}}$ with $\delta(x) = \sqrt{\varepsilon_{bl} x}$, and S is a source term that makes $u^{\text{manufactured}}(x,y)$ a solution to the PDE. For the result presented here, we set $\vec{V} = (1,0)$, $\varepsilon_{bl} = 0.1$, and we limit the number of iterations for $q = 2$ to 4 iterations for $q = 3$ to 24. We allocate more iterations for $q = 3$ cases since there are more design variables involved. For these cases, the elements do not become too skewed and we can set $\eta_V = 0$ in imposing the volume constraint.

Figure 8 shows similar benefits as in the previous section: error reduction improves with increasing p and for a given p, increasing q reduces the error significantly. More importantly, this figure shows that it is more favorable to increase q than to increase p, considering the improvement gained from increasing p or q is on the same order of magnitude but that increasing p increases the size of the system to be solved. At low orders, p, the benefit of increasing q is not always easily realized, as shown for $p = 1$ in Figure 8. This is likely due to significant under-resolution of the solution approximation at low orders.

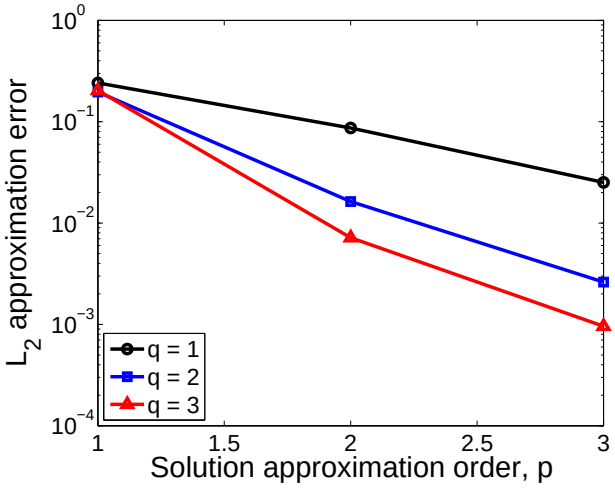
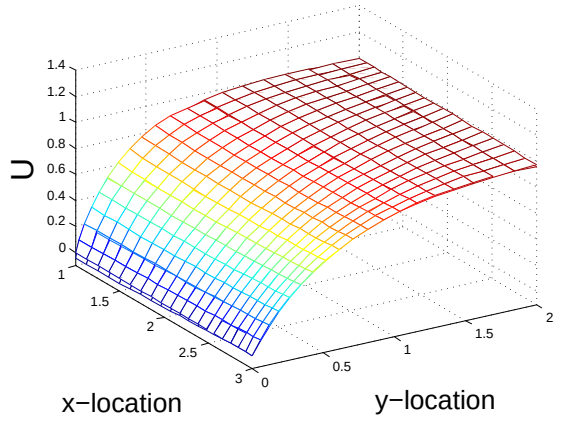


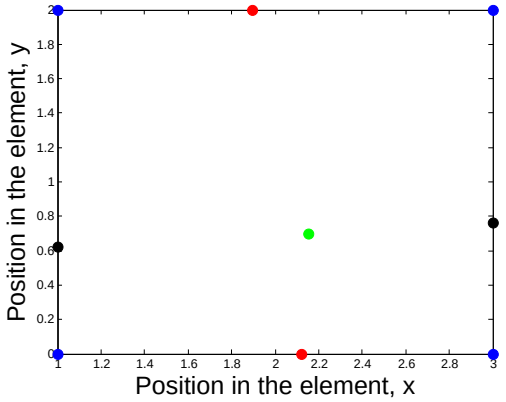
Figure 8. L_2 **error minimization for a single quadrilateral element in two spatial dimensions by optimal two-dimensional positioning of the high-order geometry nodes.**

Figure 9 shows a sample of the approximated boundary layer solution of interest (overlaid with the exact solution) along with the corresponding optimal node locations. Here, the difference is most obvious for

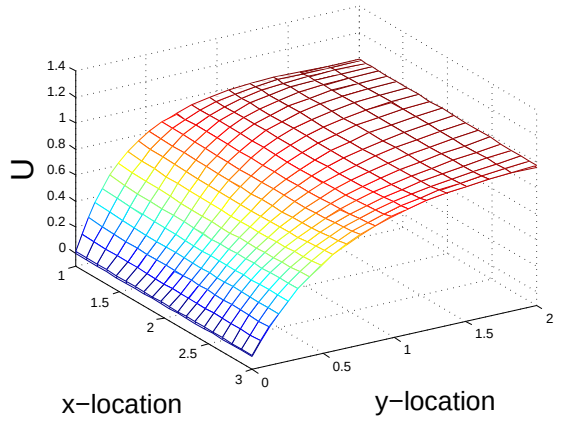
solutions generated using $p = 2$ and $q = 2$, and the difference becomes visually less discernible as we increase p or q. Furthermore, we can see that the high-order nodes cluster around the area where the solution is changing the most, and this results in the reduction of the L_2 approximation error.



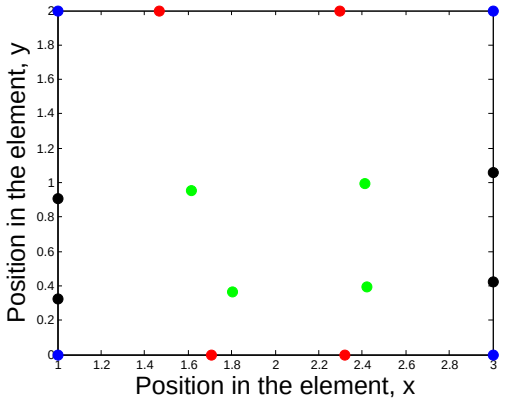
(a) Solution approximation for $p = 2$, $q = 2$ ($\varepsilon = 0.016299$)



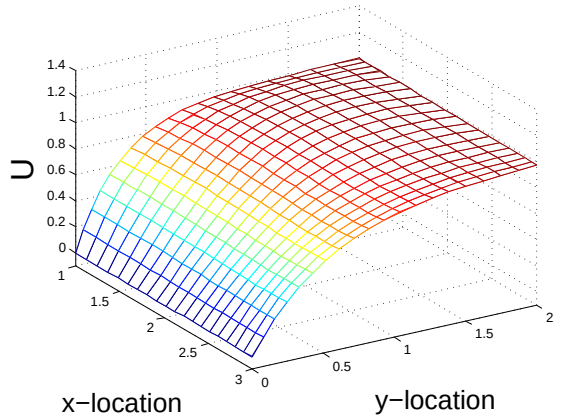
(b) Optimal node locations for $p = 2$, $q = 2$



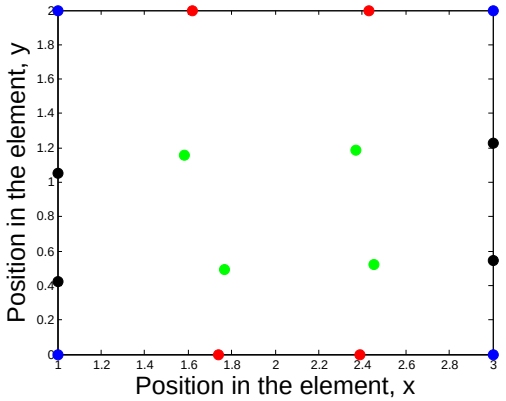
(c) Solution approximation for $p = 2$, $q = 3$ ($\varepsilon = 0.007173$)



(d) Optimal node locations for $p = 2$, $q = 3$



(e) Solution approximation for $p = 3$, $q = 3$ ($\varepsilon = 0.000959$)



(f) Optimal node locations for $p = 3$, $q = 3$

Figure 9. Example of approximated solutions for a two-dimensional least squares minimization problem, along with the corresponding optimal node locations. The approximated solutions are overlaid with the exact solution. The initial error (ε_0) with $p = 2$ is 0.08677 and with $p = 3$ is 0.02521.

V.D. Single Element, Manufactured Solution

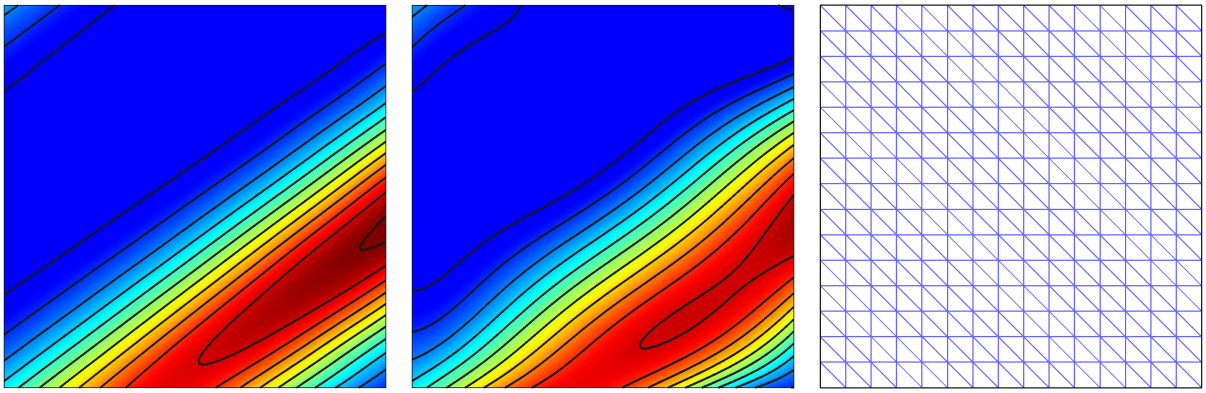
We consider a two-dimensional diffusion equation with source on a $[0,1]^2$ domain,

$$\nabla^2 u + S(\vec{x}) = 0,$$

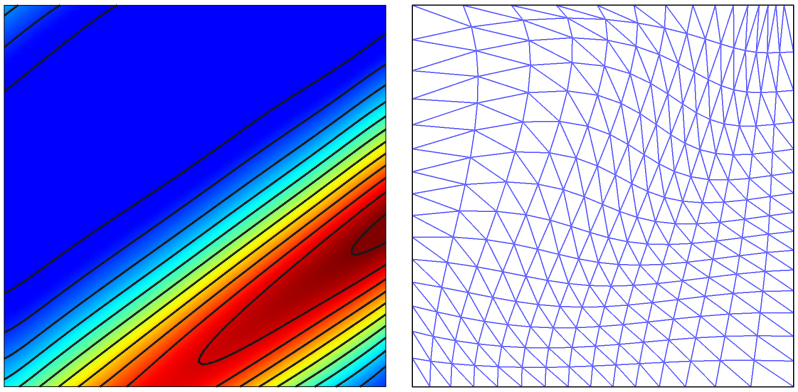
where $S(\vec{x})$ is a source term that makes the following function a manufactured solution:

$$u^{\text{manufactured}}(x,y) = \exp[a_1 \sin(a_2 x + a_3 y) + a_4 \cos(a_5 x + a_6 y)],$$

with $[a_1, a_2, a_3, a_4, a_5, a_6] = [1, 3, -4, .5, -2, 3.5]$. We use one $q = 4$ element, $p = 4$ solution approximation, and least squares error optimization with parameters $\eta_V = 0.1$ and $\mu_b = 0.1$. Figure 10 begins with the exact (manufactured solution) and the baseline $p = 4$ solution on the unoptimized, equal node-spacing element, where the least-squares error is $\varepsilon^{LS} = .0985$. The figure then shows the optimized-element solution, which is visually closer to the exact solution and which has a much lower least-squares error: $\varepsilon^{LS} = .0064$. The triangular mesh plots in Figure 10 show the initial and optimized "shape" of the single element obtained by subdividing the reference element into many, $2 \times (15 \times 15)$, equally-spaced triangles and plotting the mapped positions of these triangles in global space. We see that the optimized element shape shows marked stretching and twisting in the reference-to-global mapping; it is this distortion that is responsible for the improved approximation ability of the element even when using the same $p = 4$ space on the reference element.



(a) Exact solution (b) Initial solution ($\varepsilon^{LS} = .0985$) (c) Initial shape



(d) Optimized solution ($\varepsilon^{LS} = .0064$) (e) Optimized shape

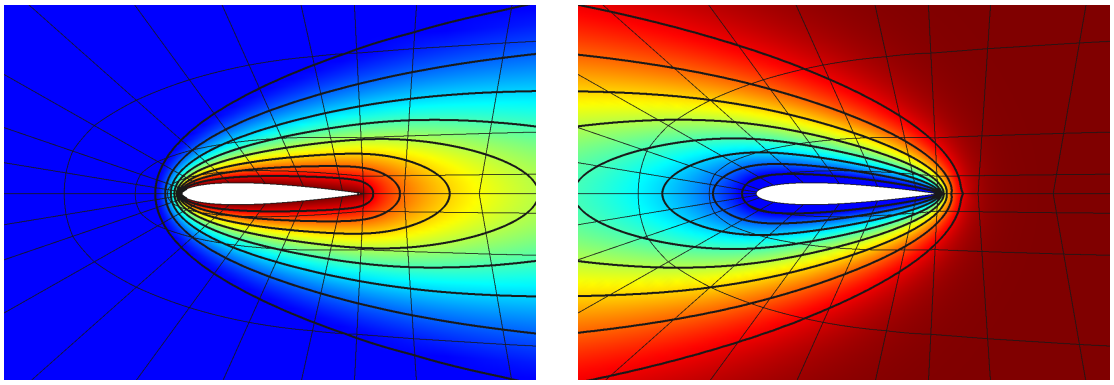
Figure 10. Single element manufactured solution: initial and optimized element shapes and least-squares approximated solutions. The least-squares error decreases from $\varepsilon^{LS} = .0985$ on the initial element to $\varepsilon^{LS} = .0064$ on the optimized element.

V.E. Multiple Element Optimization: Scalar Advection-Diffusion

We now consider a scalar advection-diffusion problem on a unit-chord ($c = 1$) NACA 0012 airfoil,

$$\nabla \cdot (\vec{V} u) - \nu \nabla^2 u = 0, \tag{11}$$

where $\vec{V} = (1, 0)$, and $Pe \equiv |\vec{V}| c/\nu = 10$. Dirichlet boundary conditions are applied: $u = 1$ on the airfoil surface and $u = 0$ on the far-field. In this problem we use the output-based optimization metric, where the output of interest ($\mathcal{J}$) is the integrated flux of u through the airfoil surface. Figure 11 shows the fine-space primal and adjoint solutions for this output on a quadrilateral mesh.



(a) Fine-space primal (b) Fine-space adjoint

Figure 11. Scalar advection-diffusion, $Pe = 10$: fine-space primal and adjoint solutions for an integrated flux output.

We run our high-order node optimization algorithm for the quadrilateral mesh shown in Figure 11, using $p = 2$, $q = 4$, $f^{\text{adapt}} = 0.2$, $\eta_V = 0.15$, and $\mu_b = 0.2$. Note that only 20% of the elements are optimized – the remaining 80% with low error are skipped. Following optimization of the targeted elements, we solve the problem again on the optimized mesh and compute the new output. Denote by $\delta\mathcal{J}$ the error in the output relative to a truth ($p = 4$) solution. We find that this output error reduces from $|\delta\mathcal{J}| = 9.0 \times 10^{-4}$ on the initial mesh to $|\delta\mathcal{J}| = 1.3 \times 10^{-5}$ on the mesh with optimized element shapes.

Figure 12 shows the error indicator, i.e. $\varepsilon_e^{\mathcal{J}}$ for each element in the baseline mesh. We see that the area near the trailing edge of the airfoil has the largest error, and thus, elements near the trailing edge will be targeted for warping. In addition, elements near the leading edge and away from the airfoil above and below it will be targeted.

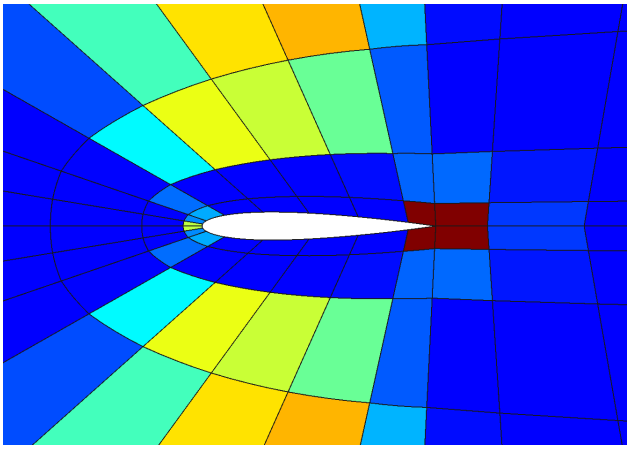
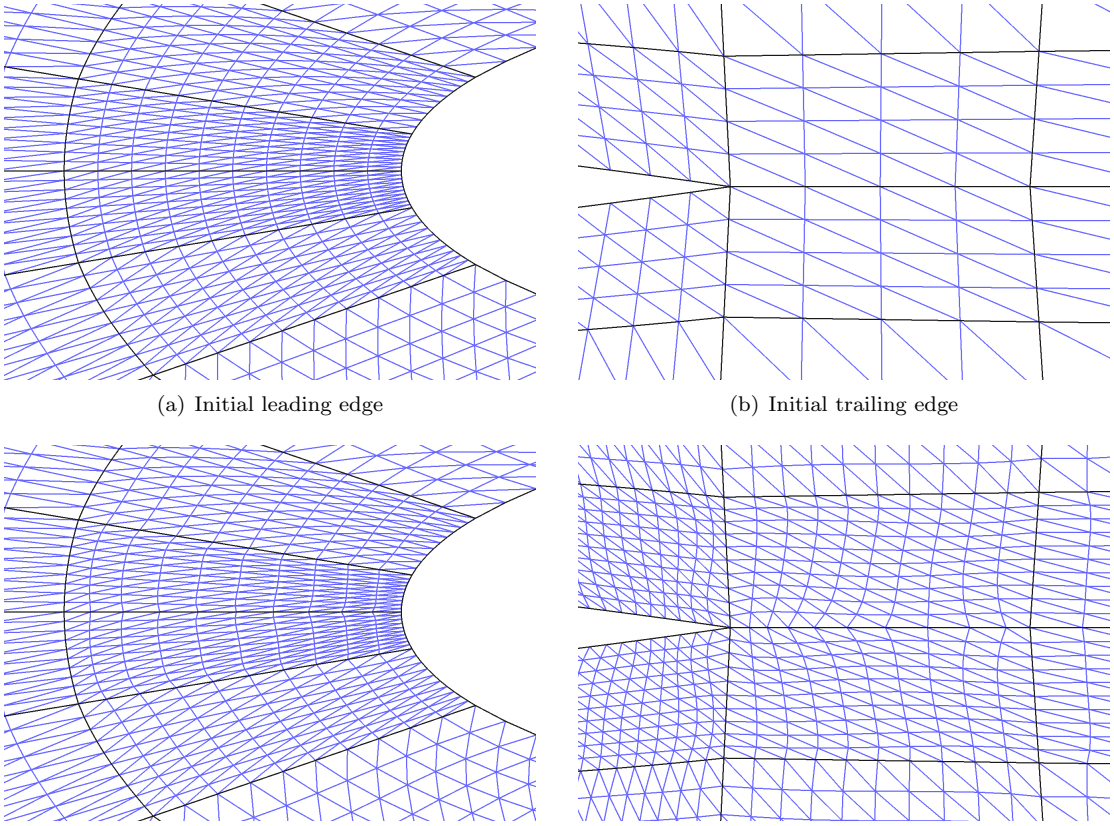


Figure 12. Scalar advection-diffusion, $Pe = 10$: adaptive indicator on a quadrilateral mesh. This shows that the elements around the trailing edge of the airfoil have large error and will be warped.

Figure 13 shows the initial and optimized quadrilateral element shapes in the leading edge and trailing edge regions of the airfoil. We see pronounced distortion of the trailing-edge elements and some distortion of the leading-edge ones.



(a) Initial leading edge

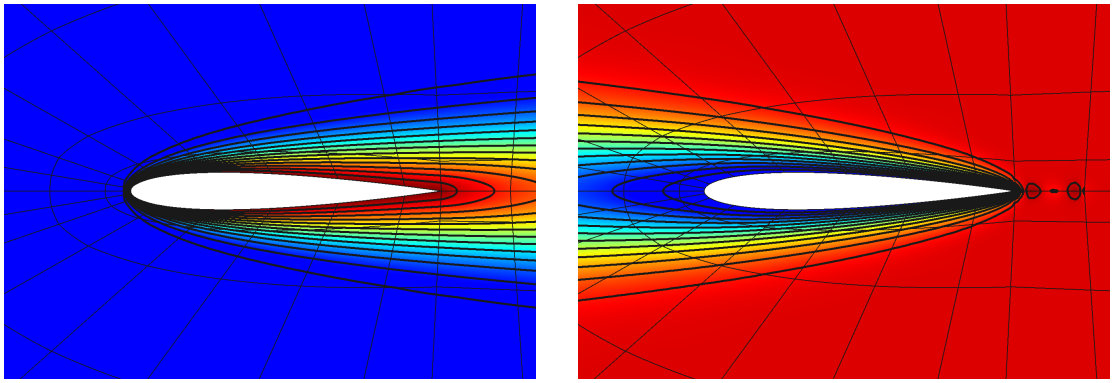
(b) Initial trailing edge

(c) Optimal leading edge

(d) Optimal trailing edge

Figure 13. Scalar advection-diffusion, $Pe = 10$: initial and optimized quadrilateral element shapes around the leading edge and trailing edge of the NACA 0012 airfoil.

Next, using the same problem setup but with lower viscosity, $Pe = 100$, we also observe a dramatic output error reduction: $|\delta\mathcal{J}| = 1.5 \times 10^{-4}$ on the baseline mesh, and $|\delta\mathcal{J}| = 9.4 \times 10^{-6}$ on the mesh with optimized element shapes. Figure 14 shows the fine-space primal and adjoint solutions. Figure 15 shows



(a) Fine-space primal (b) Fine-space adjoint

Figure 14. Scalar advection-diffusion, $Pe = 100$: fine-space primal and adjoint solutions.

the adaptive indicator for this case: close to the airfoil surface, the area near the leading edge and trailing edge of the airfoil has large error, and thus, elements there will be warped. Figure 16 shows the optimized element shapes around the leading and trailing edges of the airfoil. The distortion is again evident, though it is not as pronounced. Yet, we observe that even such small (and non-intuitive) changes in the element shapes can have a dramatic impact on the output of interest.

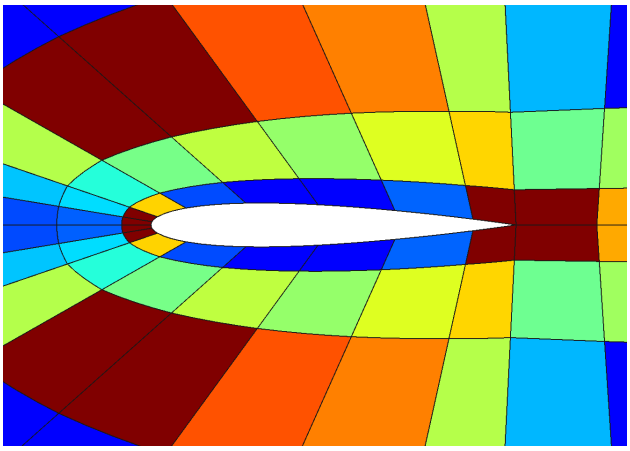


Figure 15. Scalar advection-diffusion, $Pe = 100$: adaptive indicator showing that the elements around the leading edge and trailing edge of the airfoil have the largest error and will be warped.

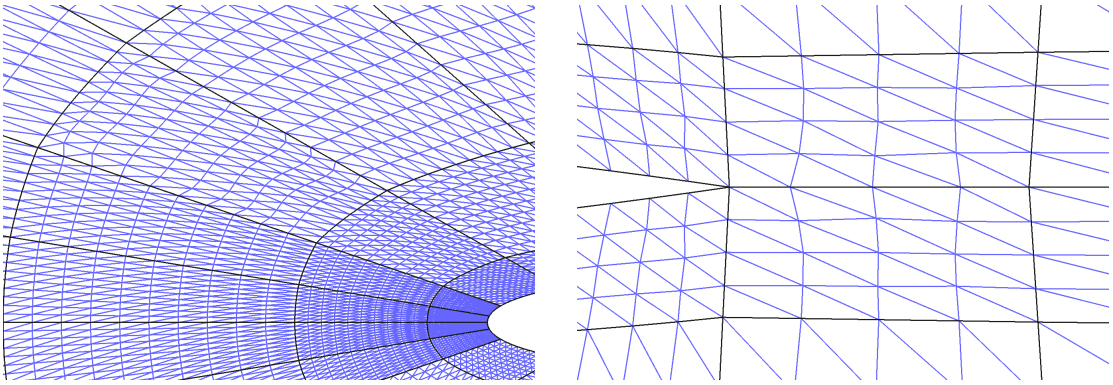
Finally, we consider the same $Pe = 100$ case but on a triangular mesh. Figure 17 shows the mesh, primal solution, and the error indicator. Elements targeted for optimization include those at the trailing edge and a small distance above and below the airfoil. Following optimization of the top 20% largest-error elements, the output error reduces from $|\delta\mathcal{J}| = 6.2 \times 10^{-5}$ on the baseline triangular mesh to $|\delta\mathcal{J}| = 3.4 \times 10^{-6}$ on the optimized mesh. Figure 18 shows the optimized mesh on the surface above/below the airfoil and around trailing edge of the airfoil.

V.F. Multiple Elements Optimization: Two-dimensional Navier-Stokes

Finally, we consider a system of equations, compressible Navier-Stokes,

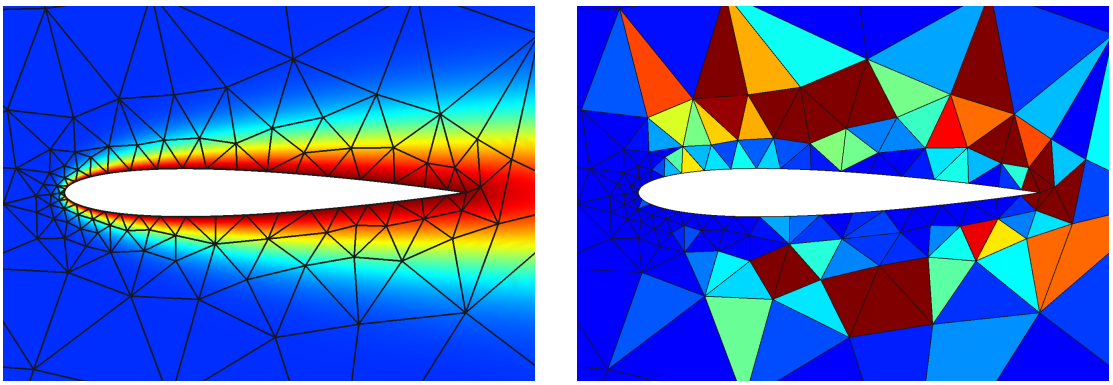
$$\nabla \cdot \vec{\mathbf{F}}(\mathbf{u}) + \nabla \cdot \vec{\mathbf{G}}(\mathbf{u}, \nabla\mathbf{u}) = \mathbf{0}, \tag{12}$$

where $\vec{\mathbf{F}}$ and $\vec{\mathbf{G}}$ are respectively the inviscid and viscous fluxes. The geometry is a NACA 0012 airfoil with a closed trailing edge, and the flow conditions are $M = 0.5$ and $Re = |\vec{V}|c/\nu = 1000$. A Prandtl number of



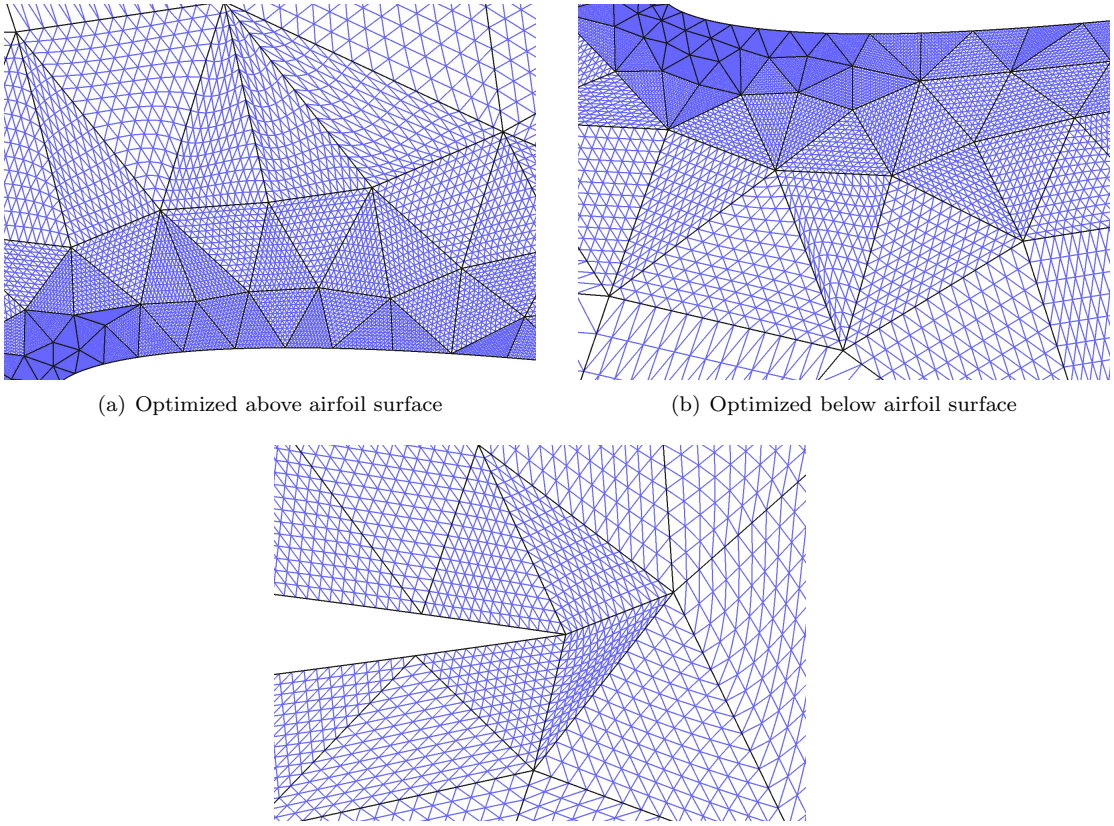
(a) Optimized leading edge (b) Optimized trailing edge

Figure 16. Scalar advection-diffusion, $Pe = 100$: initial and optimized quadrilateral element shapes around the leading edge and trailing edge.



(a) Fine-space primal (b) Error indicator

Figure 17. Scalar advection-diffusion, $Pe = 100$, triangular mesh: fine-space primal solution and error indicator.



(a) Optimized above airfoil surface

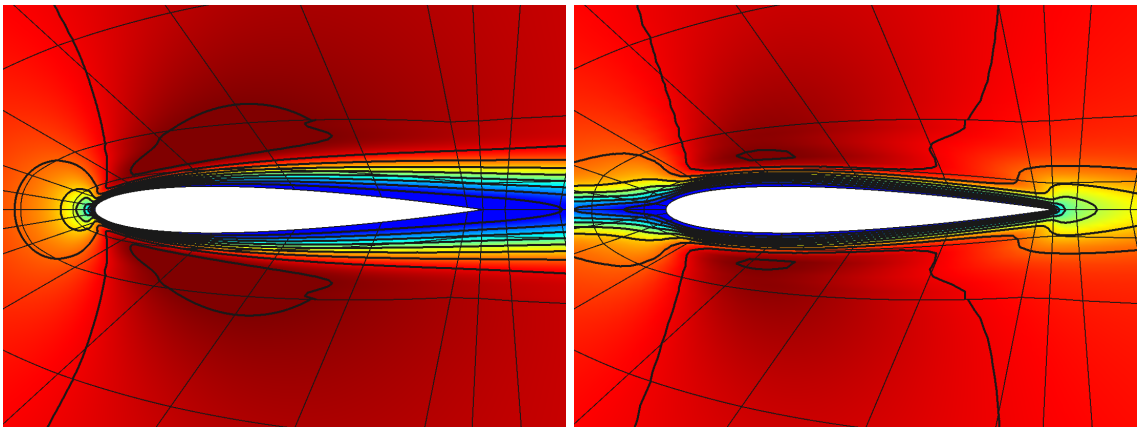
(b) Optimized below airfoil surface

(c) Optimized trailing edge

Figure 18. Scalar advection-diffusion, $Pe = 100$, triangular mesh: initial and optimized element shapes above and below the surface of the airfoil and near the trailing edge.

$Pr = 0.71$ is used, and the boundary conditions are adiabatic walls on the airfoil and free-stream conditions in the far-field. The output of interest is the drag.

Figure 19 shows the primal Mach contours and the x-momentum component of the drag adjoint computed with high-order ($p = 4$) approximation on the baseline quadrilateral mesh. We see boundary-layer structures in both the primal and adjoint.



(a) Fine-space primal (b) Fine-space adjoint

Figure 19. Compressible Navier-Stokes: fine-space primal and adjoint solutions.

Following the baseline solution, we run an optimization using $p = 2$ solution approximation, $f^{\text{adapt}} = 0.5$, $\eta_V = 0.1$, and $\mu_b = 0.2$. Figure 20 shows the error indicator: regions targeted for adaptation include above and below the airfoil, whereas the trailing edge has relatively low error in this case, possibly due to the already small elements there. Note that in this case we optimize half of the element shapes.

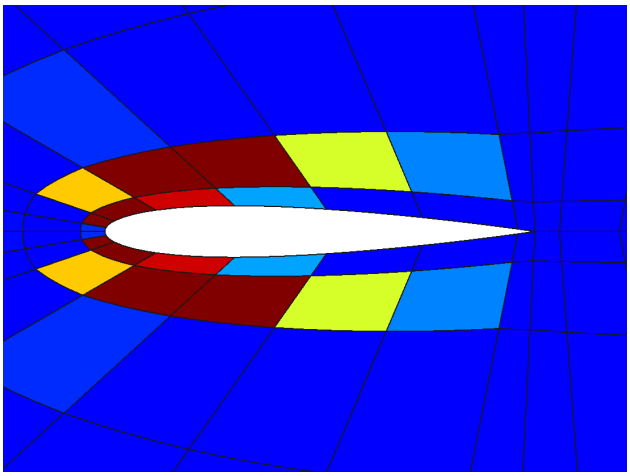
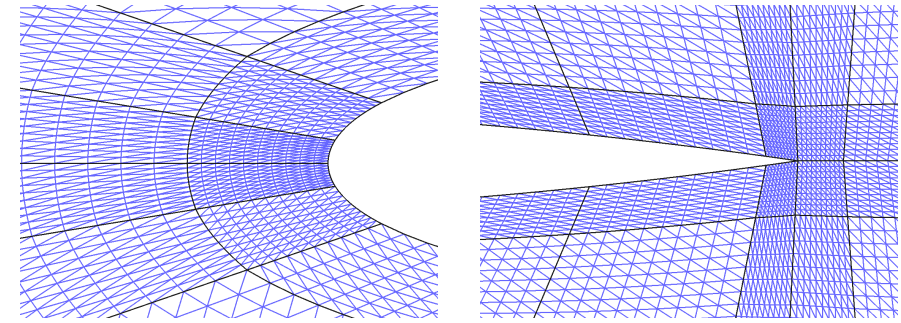
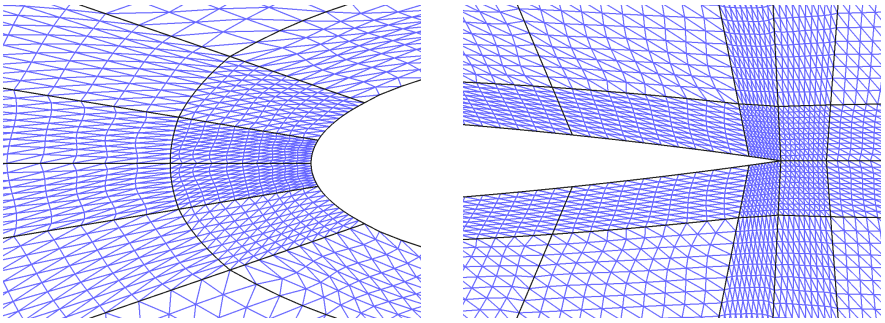


Figure 20. Compressible Navier-Stokes: adaptive indicator.

Figure 21 shows the initial and optimized element shapes around the leading edge and trailing edge of the airfoil. We see discernible and non-intuitive node movement, mostly near the leading edge. After solving again on the optimized mesh, we find that the error in the drag reduces from $|\delta\mathcal{J}| = 1.1 \times 10^{-3}$ on the baseline mesh to $|\delta\mathcal{J}| = 4.8 \times 10^{-4}$ on the mesh with optimized element shapes. The reduction in output error is not as large in this case as in the previous scalar cases. A possible reason for this is that this example is a system of equations, and different components of the system may impose different demands on the optimal element shape. In addition, we are not allowing cancellation of errors between system components in our error indicator, since we compute the least-squares errors in Eqn. 7 component-wise. Relaxing this conservative calculation may lead to larger error drops.



(a) Initial leading edge (b) Initial trailing edge



(c) Optimal leading edge (d) Optimal trailing edge

Figure 21. Compressible Navier-Stokes: initial and optimized mesh around the leading edge and trailing edge.

VI. Conclusions and Future Work

In this paper, we present a method for tailoring basis functions in a finite element method to better approximate a solution. This tailoring requires virtually no additional infrastructure beyond that already available to support curved elements in a high-order discretization, in this case DG. Instead, the locations of high-order geometry nodes become tunable parameters that warp the reference-to-global coordinate mapping and allow for accurate approximation of high-order solution features using low-order polynomials in reference space. We introduce an element-local optimization algorithm for determining the ideal positions of these nodes, driven by both least-squares and output-based error metrics. For scalar problems we observe at least a ten-fold reduction in the error, both in least-squares and in outputs. For the Navier-Stokes equations, we were able to reduce the error by a factor of two with the current approach. We note that the proposed element shape optimization does not add any degrees of freedom to the system of equations. It does require fine-space information, though this could be re-used from output-based $h/p/hp$ adaptation. Future work includes implementation of metric-based node placement, combination with hp adaptation, and globalizing the optimization problem.

Acknowledgments

The authors acknowledge the financial support of the University of Michigan and the Department of Energy, grant DE-FG02-13ER26146/DE-SC0010341.

References

[1]Wang, Z., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., "High-Order CFD Methods: Current Status and Perspective," *International Journal for Numerical Methods in Fluids*, 2013, DOI: 10.1002/fld.3767.

[2]Houston, P., Senior, B., and Süli, E., "hp-Discontinuous Galerkin Finite Element Methods for Hyperbolic Problems: Error Analysis and Adaptivity," *International Journal for Numerical Methods in Fluids*, Vol. 40, 2002, pp. 153–169.

[3]Fidkowski, K. J. and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *American Institute of Aeronautics and Astronautics Journal*, Vol. 49, No. 4, 2011, pp. 673–694.

[4]Wang, L. and Mavriplis, D., "Adjoint-based $h-p$ adaptive discontinuous Galerkin methods for the 2D compressible Euler equations," *Journal of Computational Physics*, Vol. 228, 2009, pp. 7643–7661.

[5]Fidkowski, K., "High-Order OutputBased Adaptive Methods for Steady and Unsteady Aerodynamics," *37^{th} Advanced CFD Lectures series; Von Karman Institute for Fluid Dynamics (December 9–12, 2013)*, edited by H. Deconinck and R. Abgrall, von Karman Institute for Fluid Dynamics, 2013.

[6]Melenk, J. and Babuka, I., "The partition of unity finite element method: Basic theory and applications," *Computer Methods in Applied Mechanics and Engineering*, Vol. 139, No. 14, 1996, pp. 289 – 314.

[7]Dolbow, J. and Belytschko, T., "A finite element method for crack growth without remeshing," *International Journal for Numerical Methods in Engngineering*, Vol. 46, No. 1, 1999, pp. 131–150.

[8]Benson, D., Bazilevs, Y., De Luycker, E., Hsu, M.-C., Scott, M., Hughes, T., and Belytschko, T., "A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM," *International Journal for Numerical Methods in Engineering*, Vol. 83, No. 6, 2010, pp. 765–785.

[9]Farhat, C., Harari, I., and Franca, L. P., "The discontinuous enrichment method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, No. 48, 2001, pp. 6455–6479.

[10]Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "p-Multigrid solution of high–order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 207, 2005, pp. 92–113.

[11]Reed, W. and Hill, T., "Triangular Mesh Methods for the Neutron Transport Equation," Los Alamos Scientific Laboratory Technical Report LA-UR-73-479, 1973.

[12]Cockburn, B. and Shu, C.-W., "Runge-Kutta discontinuous Galerkin methods for convection-dominated problems," *Journal of Scientific Computing*, Vol. 16, No. 3, 2001, pp. 173–261.

[13]Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

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

[15]Bassi, F. and Rebay, S., "High–order accurate discontinuous finite element solution of the 2-D Euler equations," *Journal of Computational Physics*, Vol. 138, 1997, pp. 251–285.

[16]J. E. Dennis, J. and More, J. J., "Quasi-Newton Methods, Motivation and Theory," *Society for Industrial and Applied Mathematics Review*, Vol. 19, No. 1, 1977, pp. 46 – 89.