AIAA SciTech Forum
4-8 January 2016, San Diego, California, USA
54th AIAA Aerospace Sciences Meeting

10.2514/6.2016-0862

High-Order Output-Based Adaptive Simulations of Turbulent Flow Over a Three Dimensionsional Bump

Krzysztof J. Fidkowski*
Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA
Marco A. Ceze†
NASA Ames Research Center, Moffett Field, CA, USA

This paper presents output-based high-order adaptive simulations of a three-dimensional benchmark problem for turbulence model verification. The flow over a smooth bump is modeled using the Reynolds-averaged compressible Navier-Stokes equations, with a negative-viscosity formulation of the Spalart-Allmaras (SA) one-equation closure. The initial high-order, curved, hexahedral mesh for the adaptive runs is generated manually with sufficient boundary layer resolution to enable robust convergence at the approximation orders tested. Thereafter, resolution is added using fixed-fraction, hanging-node refinement driven by an output error indicator, calculated from a discrete adjoint-weighted residual. A comparison of the output convergence results to previous data verifies the methods used and demonstrates the benefit of high-order approximation for this case. In addition, several implementation details are discussed, including quality curved-mesh generation, wall distance calculation, hanging-node refinement, snapping of boundaries to the geometry, and a nonlinear Newton continuation strategy.

I. Introduction

The Reynolds-averaged Navier-Stokes (RANS) equations remain an invaluable model routinely used in analysis and design of aerospace vehicles. RANS simulations are computationally cheap compared to other options for simulation turbulence because they take advantage of anisotropic meshes that reduce the degrees of freedom required to accurately resolve thin boundary and shear layers. However, realizing this advantage is not always straightforward, especially for high-order methods that use curved elements. Maintaining mesh validity in these cases, particularly in adaptive three-dimensional simulations, remains a challenging task and a subject of ongoing research.

The combination of high-order approximation and mesh adaptation offers an attractive solution strategy for RANS simulations, which generally contain both smooth and singular features.[1] In addition, output-based adaptive methods[2–5] offer a systematic approach for identifying regions of the domain that require more resolution for the prediction of scalar outputs of interest. These methods also return error estimates that can improve the robustness of solution verification and the efficiency uncertainty quantification studies. It is for these reasons that we consider output-based methods in the present study.

In this paper, we apply a high-order adaptive solution technique to a three-dimensional benchmark test case modeled using the RANS equations, closed with a negative-turbulent-viscosity modification of the Spalart-Allmaras (SA) one-equation model.[6] Many previous works have investigated

*Associate Professor, AIAA Senior Member, kfid@umich.edu
†Postdoctoral Fellow, Oak Ridge Associated Universities, marco.a.ceze@nasa.gov

the RANS-SA equations, including in a high-order adaptive setting.[1,7–10] The majority of the latter work has focused on demonstrating benefits of adaptive refinement and/or high-order over uniform or heuristic refinement for such flows. More recently, the authors and several other groups have compared RANS results across discretizations and mesh types/refinement techniques, for several two-dimensional benchmark problems.[11] In the present work we extend the verification to three dimensions, with an in-depth study of one simulation: flow over a smooth three-dimensional bump.

The remainder of this paper is organized as follows. Section II presents the compressible Navier-Stokes equations closed with the RANS-SA model, and Section III discusses their discretization with a high-order discontinuous finite-element method. Section IV details several implementation aspects specific to three-dimensional RANS simulations. Section V describes the output error estimation and adaptation techniques, and Section VI presents results for the several benchmark cases considered. Section VII concludes with a summary and a discussion of possible future directions.

II. The Reynolds-Averaged Compressible Navier-Stokes Equations

The model equations in this work are compressible Navier-Stokes, Reynolds-averaged with a version of the Spalart-Allmaras turbulence model that is modified for improved stability for negative values of the turbulence working variable, $\tilde{\nu}$.[6] The resulting Reynolds-averaged Navier-Stokes (RANS) equations are, using index notation with implied summation on repeated indices,

$$\begin{aligned}
&\partial_t \rho && + \partial_j(\rho u_j) && && = 0\\
&\partial_t(\rho u_i) && + \partial_j(\rho u_j u_i + p\delta_{ij}) && - \partial_j \tau_{ij} && = 0\\
&\partial_t(\rho E) && + \partial_j(\rho u_j H) && - \partial_j(u_i \tau_{ij} - q_j) && = 0\\
&\underbrace{\partial_t(\rho\tilde{\nu})}_{\text{unsteady}} && + \underbrace{\partial_j(\rho u_j \tilde{\nu})}_{\text{convective}} && - \underbrace{\partial_j\left[\frac{1}{\sigma}\rho(\nu+\tilde{\nu}f_n)\partial_j\tilde{\nu}\right]}_{\text{diffusive}} + \underbrace{S_{\tilde{\nu}}}_{\text{source}} && = 0
\end{aligned} \tag{1}$$

where the terms have been split according to their treatment in the discretization, and where source term for the $\tilde{\nu}$ equation is

$$S_{\tilde{\nu}} = \frac{1}{\sigma}(\nu+\tilde{\nu}f_n)\partial_j\rho\partial_j\tilde{\nu} - \frac{c_{b2}\rho}{\sigma}\partial_j\tilde{\nu}\partial_j\tilde{\nu} - P + D.$$

In the above equations, ρ is the density, ρu_j is the momentum, E is the total energy, $H = E + p/\rho$ is the total enthalpy, $p = (\gamma-1)\left(\rho E - \frac{1}{2}\rho u_i u_i\right)$ is the pressure, γ is the ratio of specific heats, P is the turbulence production, D is the turbulence destruction, and i, j index the spatial dimension, dim. The Reynolds stress, τ_{ij}, is

$$\tau_{ij} = 2(\mu+\mu_t)\bar{\epsilon}_{ij}, \qquad \bar{\epsilon}_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i) - \frac{1}{3}\partial_k u_k \delta_{ij}.$$

μ is the laminar dynamic viscosity, obtained using Sutherland's law,

$$\mu = \mu_{\text{ref}}\left(\frac{T}{T_{\text{ref}}}\right)^{1.5}\left(\frac{T_{\text{ref}}+T_{\text{s}}}{T+T_{\text{s}}}\right), \tag{2}$$

where $T = p/(\rho R)$ is the temperature, R is the gas constant for air (the difference in specific heats), and the eddy viscosity, μ_t, is

$$\mu_t = \begin{cases} \rho\tilde{\nu}f_{v1} & \tilde{\nu} \geq 0 \\ 0 & \tilde{\nu} < 0 \end{cases} \qquad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu}.$$

The heat flux, q_j, is given by

$$q_j = (\kappa + \kappa_t)\partial_i T, \qquad \kappa = C_p\mu/Pr, \qquad \kappa_t = C_p\mu_t/Pr_t,$$

where Pr and Pr_t are the laminar and turbulent Prandtl numbers, and C_p is the specific heat at constant pressure. The production term, P, is

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

where the modified vorticity $\tilde{S}$ is written as

$$\tilde{S} = \begin{cases} S + \overline{S} & \overline{S} \geq -c_{v2}S \\ S + \dfrac{S(c_{v2}^2 S + c_{v3}\overline{S})}{(c_{v3} - 2c_{v2})S - \overline{S}} & \overline{S} < -c_{v2}S \end{cases}, \quad \overline{S} = \frac{\tilde{\nu} f_{v2}}{\kappa^2 d^2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}. \tag{3}$$

In Eqn. 3, $S = \sqrt{2\Omega_{ij}\Omega_{ij}}$ is the vorticity magnitude (summation implied on i, j), and $\Omega_{ij} = \frac{1}{2}(\partial_i v_j - \partial_j v_i)$ is the vorticity tensor. d is the distance to the closest wall. The destruction term, D, is given by

$$D = \begin{cases} c_{w1} f_w \dfrac{\rho\tilde{\nu}^2}{d^2} & \chi \geq 0 \\ -c_{w1}\dfrac{\rho\tilde{\nu}^2}{d^2} & \chi < 0 \end{cases}, \quad f_w = g\left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{1/6}, \quad g = r + c_{w2}(r^6 - r), \quad r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}.$$

Finally, in Eqn. 1, $f_n = 1$ for positive $\tilde{\nu}$ and

$$f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}, \quad \text{when } \chi < 0. \tag{4}$$

Relevant closure coefficients are

c_{b1}	=	0.1355	c_{w1}	=	$\frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}$	c_{v1}	=	7.1
c_{b2}	=	0.622	c_{w2}	=	0.3	κ	=	0.41
σ	=	2/3	c_{w3}	=	2	Pr_t	=	0.9
c_{n1}	=	16	c_{v2}	=	0.7	c_{v3}	=	0.9

III. Discontinuous Galerkin Discretization

We discretize Eqn. 1 using a discontinuous Galerkin (DG) finite element method.[9,12] Defining the state vector as $\mathbf{u} = [\rho, \rho u_i, \rho E, \rho\tilde{\nu}]^T$, we write Eqn. 1 in compact conservative form,

$$\partial_t \mathbf{u} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{u}, \nabla\mathbf{u}) + \mathbf{S}(\mathbf{u}, \nabla\mathbf{u}) \quad = \quad \mathbf{0}, \tag{5}$$

where $\vec{\mathbf{F}}$ is the combined inviscid/viscous flux vector, and $\mathbf{S}$ is the source term associated with the turbulence closure equation. We approximate the state in a finite dimensional space, $\mathbf{u}_h \in \boldsymbol{\mathcal{V}}_h$, where $\boldsymbol{\mathcal{V}}_h$ is the space of element-wise discontinuous polynomials of order p. Choosing a basis for $\boldsymbol{\mathcal{V}}_h$ yields the following state approximation on element k,

$$\mathbf{u}(\vec{x})\big|_k = \sum_{j=1}^{n(p)} \mathbf{U}_{kj}\phi_j(\vec{x}),$$

where $\mathbf{U}_{kj}$ are the six unknowns associated with basis function j on element k. We denote by $\mathbf{U} = \{\mathbf{U}_{kj}\}$ all of these unknowns rolled into one vector. By virtue of the discontinuous approximation space inherent to DG, the basis functions used to approximate the state need not correspond to the geometrical shape of the element. For example, whereas tensor product basis functions are typically used on hexahedral elements, in DG one can also use full-order "tetrahedral" basis functions. This results in a significant savings in degrees of freedom for the same nominal order of convergence: for example, a $p = 4$ tensor product basis requires 125 unknowns per element, while a $p = 4$ full-order basis only requires 35 unknowns per element. This yields storage savings factors of over 3.5 for the state and 12 for the residual Jacobian matrix.

Multiplying Eqn. 5 by test functions in $\boldsymbol{\mathcal{V}}_h$, which are the same as the basis functions for DG, integrating by parts on each element, and using the Roe[13] convective flux and the second form of Bassi and Rebay (BR2)[14] for the viscous treatment, we obtain the following system of nonlinear equations,

$$\mathbf{R}(\mathbf{U}) \quad = \quad \mathbf{0}. \tag{6}$$

III.A. Symmetry boundary conditions

The bump case presently considered requires symmetry boundary conditions on several boundaries. In the continuous limit, symmetry requires vanishing normal state derivatives. A finite-dimensional solution will generally violate these requirements pointwise, so we must enforce the BC weakly. This enforcement involves transforming the state and gradient, similarly to methods in previous works,[15] though we construct a state/gradient on the boundary instead of employing a ghost cell. Starting with the state, we require that at a symmetry boundary all *vectors* in the state (e.g. a velocity) have their normal components zeroed out. This results in a linear transformation from the interior ($\mathbf{u}^+$) to the boundary ($\mathbf{u}^b$) state vector, which reads $\mathbf{u}^b = \mathbf{A}\mathbf{u}^+$. $\mathbf{A}$ is the identity matrix for all states except the momentum, which transforms as $(\rho\vec{v})^b = \underline{V}(\rho\vec{v})^+$, where $\underline{V} = \underline{I} - \vec{n} \otimes \vec{n} = \delta_{ij} - n_i n_j$. $\vec{n}$ is the outward-pointing normal, and $\underline{I} = \delta_{ij}$ is the dim $\times$ dim identity matrix.

The state gradient transformation must account for possibly nonzero normal velocity components. We first consider a hypothetical *ghost* state ($\mathbf{u}^-$) and gradient ($\nabla\mathbf{u}^-$), obtained by *reflecting* the velocity about the symmetry plane. Specifically, $\mathbf{u}^- = \mathbf{B}\mathbf{u}^+$, where $\mathbf{B}$ is an identity matrix for all states except the momentum, which transforms as $(\rho\vec{v})^- = \underline{W}(\rho\vec{v})^+$, where $\underline{W} = \underline{I} - 2\vec{n} \otimes \vec{n} = \delta_{ij} - 2n_i n_j$. Note that $\mathbf{B} = 2\mathbf{A} - \mathbf{I}$ and that $\underline{W} = 2\underline{V} - \underline{I}$. Differentiating the expression for $\mathbf{u}^-$ in space gives the gradient, which we must reflect by applying $\underline{W}$, so that $\nabla\mathbf{u}^- = \mathbf{B}\nabla\mathbf{u}^+\underline{W}^T$. Finally, we obtain the gradient *at* the boundary, $\nabla\mathbf{u}^b$, by averaging the interior and exterior gradients – this is consistent with what would happen in the viscous flux calculation if there were actually a symmetrical mesh on the other side of the symmetry line. So we have

$$\begin{aligned} \nabla\mathbf{u}^b &= \frac{1}{2}\left(\nabla\mathbf{u}^+ + \nabla\mathbf{u}^-\right) = \frac{1}{2}\left(\nabla\mathbf{u}^+ + \mathbf{B}\nabla\mathbf{u}^+\underline{W}^T\right) = \frac{1}{2}\left(\nabla\mathbf{u}^+ + (2\mathbf{A} - \mathbf{I})\nabla\mathbf{u}^+(2\underline{V}^T - \underline{I})\right) \\ &= \nabla\mathbf{u}^+ + \mathbf{A}\nabla\mathbf{u}^+(2\underline{V}^T - \underline{I}) - \nabla\mathbf{u}^+\underline{V} = \nabla\mathbf{u}^+\vec{n} \otimes \vec{n} + \mathbf{A}\nabla\mathbf{u}^+(\underline{I} - 2\vec{n} \otimes \vec{n}). \end{aligned}$$

III.B. Scaling of $\tilde{\nu}$

The SA working variable, $\tilde{\nu}$, will generally be orders of magnitude smaller than the other state components. We use scaling or "non-dimensionalization" of $\tilde{\nu}$ to make its range of numerical values similar to the other state components. This proves to be effective in improving the performance of the linear and nonlinear solvers.[12] We store the scaled quantity, $\rho\tilde{\nu}'$, given by

$$\rho\tilde{\nu}' = \frac{\rho\tilde{\nu}}{\kappa_{SA}\mu_\infty},$$

where κ_{SA} is a scaling factor, typically $\mathcal{O}(\sqrt{Re})$, and μ_∞ is the free-stream laminar dynamic viscosity. In addition, the SA $\tilde{\nu}$ equation is divided by $\kappa_{SA}\mu_\infty$.

IV. Implementation

IV.A. Mesh Generation

A curved, hexahedral mesh was generated manually as a starting mesh for the adaptive runs. High-order curved elements provide geometric fidelity that is essential for robust and accurate solutions via the discontinuous Galerkin method, even at solution approximation orders of $p = 1$.[16] In this work, we employ curved elements throughout the domain, as the presence of highly-anisotropic elements near the wall precludes the use of just one layer of curved elements. The anisotropy is motivated by efficiency: RANS simulations require much higher resolution, i.e. smaller length scales, perpendicular to the wall compared to parallel to the wall, so that anisotropic "pancake" elements reduce the total degrees of freedom without sacrificing accuracy.

In this work, each hexahedron in physical space is obtained by mapping a unit reference cube via tensor-product polynomials of order q, as shown in Figure 1. Nodal Lagrange basis functions, with equal node spacing in reference space, yield curved elements that interpolate the provided nodes, $(q+1)^3$ total. This property is useful for defining curved hexahedra via the coordinates of their $(q+1)^3$ nodes.

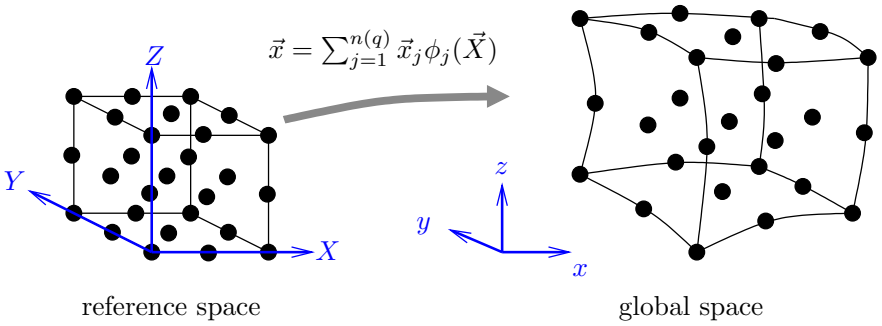


Figure 1. Reference-to-global mapping for a hexahedral elements, shown for $q = 2$ geometry approximation with 27 nodes.

The high-order nodes inside the elements should be spaced roughly uniformly in physical space, unless attempting to optimize the elements' approximation power,[17] to minimize skewness and avoid the risk of negative mapping Jacobians. However, the elements themselves should ideally be distributed non-uniformly over the domain to provide efficient solution approximation. In this work we use logarithmic spacing for the element corner nodes, and linear spacing for the in-between high-order nodes, as illustrated in Figure 2. For the present bump problem, these spacings are prescribed for a uniform lattice in a featureless duct, which is then deformed and blended to fit the given geometry.

Several remarks about the high-order meshes are in order. First, the meshes are watertight by virtue of the uniqueness of edge and face approximations. That is, each edge is fully defined by the nodes on that edge, regardless of the positions of the other nodes in the element. The same is true for the faces. Second, although we use high-order curved elements, slope/normal continuity is not explicitly enforced at inter-element boundaries. However, the mismatch in the slope, along with the geometry errors in general, diminish as the mesh is refined. Third, curved elements require non-canonical mass matrices, and these are computed via quadrature and stored for the duration of the run. Finally, when a mesh is adapted, new nodes placed on the curved boundary are snapped to the geometry. This demands sufficient resolution from the initial mesh to prevent element inversion,

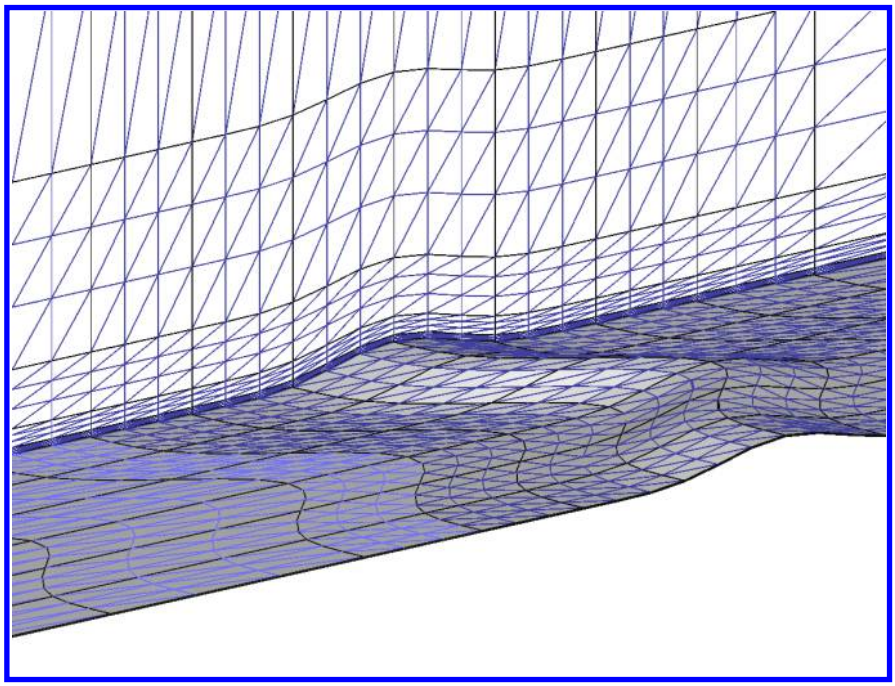


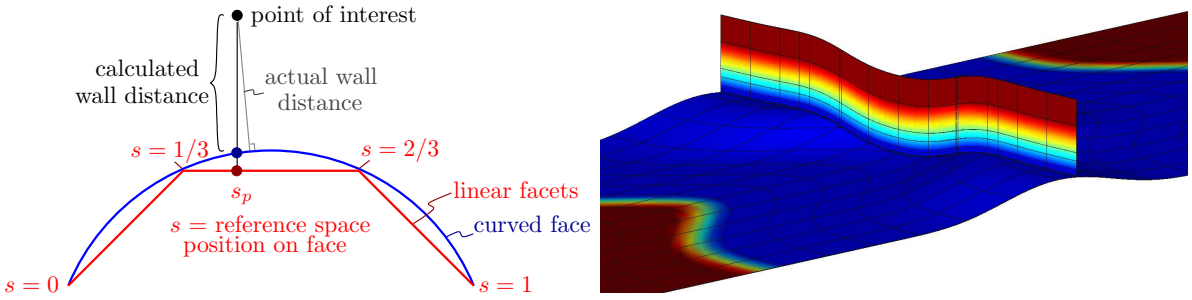
Figure 2. To increase resolution near the bump wall, elements (the boundaries of which are denoted by black lines) are spaced logarithmically away from the wall. However, to minimize element skewness, the nodes inside each element (identified by the blue triangulation) are spaced uniformly.

as only the first layer of elements adjacent to the boundary is affected by the snapping.

IV.B. Wall Distance Calculation

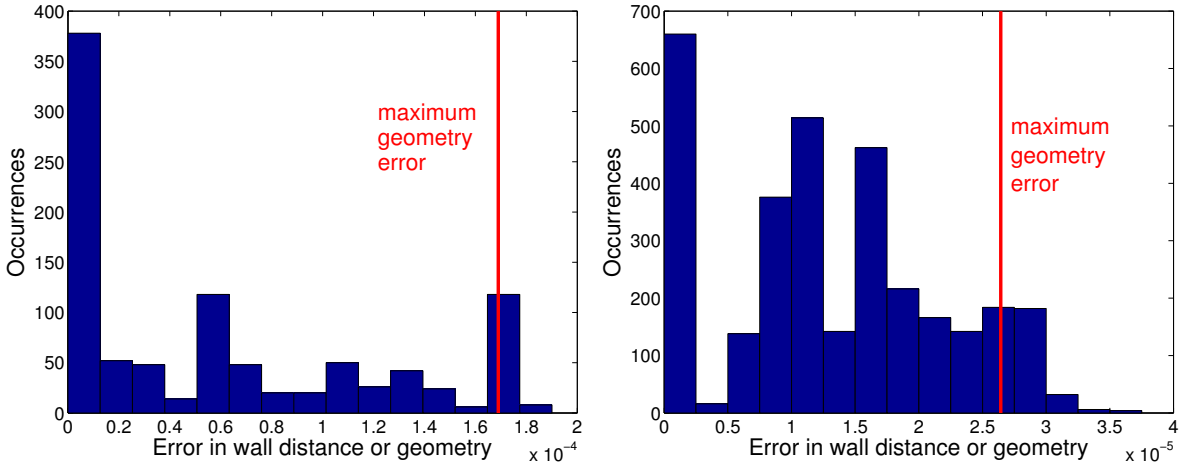
The distance to the closest wall, d, is required for the Spalart-Allmaras turbulence model. This distance is used at every quadrature point, during the construction of the residual and residual Jacobian matrix. Rather than separately storing the distance at every quadrature point, the wall distance is approximated by a polynomial of order p_{wd} on each element. This polynomial is constructed by evaluating the wall distance at each Lagrange node used in the polynomial approximation on an element.

The wall distance is calculated at each order p_{wd} Lagrange node of each element using the following procedure. First, the distance to each boundary node is calculated via a brute-force procedure to pre-select the closest boundary faces, which are those adjacent to the closest node. For each of these boundary faces, which can generally be curved, the wall distance is computed by first subdividing the high-order face into linear triangles, with $2(q+1)$ subdivisions along each edge. Thus, a quadrilateral boundary face would be split into $2[2(q+1)]^2 = 8(q+1)^2$ triangles. The distance to each of these triangles is computed by projection, and the closest triangle is selected. The minimum distance to the linear triangles could be used as an estimate of the wall distance, though it could be insufficiently accurate for highly-curved faces. Rather, the projection of the point in question onto the closest triangle defines reference-space coordinates on the original face that are used to "snap" the projection to the curved face geometry via the reference-to-global geometry mapping, as illustrated in Figure 3. The distance to this snapped projection then defines the wall distance. For further accuracy, the projection could serve as an initial guess for Newton-Raphson iterations to minimize the distance on the curved face, though this extra step has not been found to produce additional accuracy gains above the geometry approximation error of the curved faces, as indicated by the wall distance error study in Figure 4.



(a) Wall-distance calculation (b) Contours of the wall-distance function

Figure 3. Schematic showing the process by which the wall distance is calculated on curved faces. Projection to a linear panel representation of the face defines the reference-space face coordinates, which are then used to calculate a more accurate projection on the true curved face. The wall distance is stored using a polynomial representation on each element.



(a) (b)

Figure 4. Histograms of error in the wall distance calculation at quadrature points in the first layer of elements adjacent to the wall, shown for a baseline mesh (left) and a uniform refinement of the baseline mesh (right). The error is calculated relative to a true wall distance obtained by a Newton-Raphson projection to the true geometry. For most of the points, the error in the wall distance is dominated by the maximum error in the geometry representation (via high-order curved elements), indicated by the vertical red line in each plot. In addition, the error decreases with mesh refinement as expected.

IV.C. Nonlinear Solver

The DG discretization yields a system of nonlinear equations, $\mathbf{R}(\mathbf{U}) = \mathbf{0}$, that must be solved for the unknown solution approximation coefficients, $\mathbf{U}$. We use a Newton-Raphson method with pseudo-transient continuation for robustness when not close to the root. Specifically, at each Newton iteration, the following linear system is solved (via the restarted generalized minimal residual method – GMRES):

$$\left(\frac{\mathbf{M}}{\Delta t_a} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\Big|_{\mathbf{U}_0}\right)\Delta\mathbf{U} + \mathbf{R}(\mathbf{U}_0) \quad = \quad \mathbf{0}, \tag{7}$$

where $\mathbf{U}_0$ is the solution guess, $\mathbf{M}$ is the block-diagonal mass matrix, and Δt_a is an element-specific artificial time step given by

$$\Delta t_a = \mathrm{CFL}\frac{h}{c_{\max}},$$

with h = volume/(surface area), and $c_{\max}$ the maximum characteristic speed over the quadrature points of the element. CFL is a global Courant-Friedrichs-Lewy number that starts low and progressively increases according to an evolution strategy, described next, as the solution converges.

When not close to the root, the solution to Eqn. 7 may produce an update that makes $\mathbf{U}_0 + \Delta\mathbf{U}$ non-physical. We therefore under-relax the update according to the following line-search strategy:

1. Given: $\mathbf{U}_0$ and $\Delta\mathbf{U}$, the solution to Eqn. 7.
2. Compute ω^{phys} = maximum fraction such that $\mathbf{U}_0 + \omega^{\text{phys}}\Delta\mathbf{U}$ remains physical. This involves checks at quadrature points of each element.
3. Set $\omega = \min(1, \omega^{\text{phys}})$. If $\omega < 1$, set $\omega = \omega\beta^{\text{phys}}$, where $\beta^{\text{phys}} < 1$ is a buffer reduction factor that keeps the solution from being borderline non-physical.
4. While $\omega > \omega^{\min}$ and $\|\mathbf{R}(\mathbf{U}_0 + \omega\Delta\mathbf{U}\| > \beta^{\text{residual}}\|\mathbf{R}(\mathbf{U}_0)\|$: set $w = w\beta^{\text{line}}$, where $\beta^{\text{line}} < 1$.
5. If $\omega < \omega^{\min}$, do not take the update. Instead, set $\mathrm{CFL} = \mathrm{CFL}\beta^{\text{CFL,decrease}}$ and return to the first step.
6. If $\omega \geq \omega^{\min}$, take the update: $\mathbf{U} = \mathbf{U}_0 + \omega\Delta\mathbf{U}$. Furthermore, if $\omega = 1$, raise the CFL number: $\mathrm{CFL} = \mathrm{CFL}\beta^{\text{CFL,increase}}$. Return to the first step.

The parameters do not need much tuning, and the ones below used in this study also work for many other problems.

$$\beta^{\text{phys}} = 0.5,\ \beta^{\text{residual}} = 2.0,\ \beta^{\text{line}} = 0.5,\ \omega^{\min} = 0.24,\ \beta^{\text{CFL,increase}} = 1.2,\ \beta^{\text{CFL,decrease}} = 0.1.$$

To accelerate the solver during adaptive iterations, when a good guess is available from the solution on the previous mesh, the CFL increase factor is raised to $\beta^{\text{CFL,increase}} = 5.0$. In all runs, the starting CFL number is 1.0.

V. Mesh Adaptation

V.A. Error Estimation

We use an adjoint-based output error estimate to drive mesh adaptation. The discrete system of nonlinear equations reads $\mathbf{R}(\mathbf{U}) = \mathbf{0}$, where the state and residual vectors are both in $\mathbb{R}^N$. For a scalar output, $J(\mathbf{U})$, the discrete adjoint vector, $\mathbf{\Psi} \in \mathbb{R}^N$, is the sensitivity of J to perturbations in $\mathbf{R}$.[5] It satisfies the following linear equation,

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^T \mathbf{\Psi} + \left(\frac{\partial J}{\partial \mathbf{U}}\right)^T = \mathbf{0}. \tag{8}$$

The adjoint vector provides an estimate of the error in the output when computing on a finite-dimensional approximation space. Consider two finite-dimensional spaces: a coarse approximation space, $\boldsymbol{\mathcal{V}}_H$, on which we calculate the state and output, and a fine space, $\boldsymbol{\mathcal{V}}_h$ (obtained by incrementing the approximation order by 1), on which we compute the adjoint and relative to which we estimate the error. We would like to measure the output error in the coarse solution relative to the fine space,

$$\text{output error: } \delta J \equiv J_H(\mathbf{U}_H) - J_h(\mathbf{U}_h). \tag{9}$$

We assume that the fine approximation space contains the coarse approximation space, so that a lossless state injection, $\mathbf{U}_h^H \equiv \mathbf{I}_h^H \mathbf{U}_H$, exists, where $\mathbf{I}_h^H$ is the coarse-to-fine state injection (prolongation) operator. The fine-space solution, $\mathbf{U}_h \in \mathbb{R}^{N_h}$, solves $\mathbf{R}_h(\mathbf{U}_h) = \mathbf{0}$, but the injected state will generally not give zero fine-space residuals, $\mathbf{R}_h(\mathbf{U}_h^H) \neq \mathbf{0}$. Instead, the injected coarse state solves a *perturbed* fine-space problem, $\mathbf{R}_h(\mathbf{U}'_h) - \mathbf{R}_h(\mathbf{U}_h^H) = \mathbf{0}$, and the fine-space adjoint, $\boldsymbol{\Psi}_h$, tells us to expect an output perturbation given by the inner product between the adjoint and the residual perturbation,

$$\delta J \approx -\boldsymbol{\Psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H). \tag{10}$$

This estimate assumes small perturbations in the state when the output or equations are nonlinear. It does not require the fine-space primal solution, $\mathbf{U}_h$, but it does require the fine-space adjoint. In this work, we fully converge the fine-space adjoint about the injected state, $\mathbf{U}_h^H$, storing the fine-space Jacobian and using $\boldsymbol{\Psi}_h^H \equiv \mathbf{I}_h^H \boldsymbol{\Psi}_H$ as a initial guess in the GMRES iterative solver for $\boldsymbol{\Psi}_h$. For the three-dimensional problem considered in this work, this does add non-trivial additional computational cost and memory overhead. However, we do this to minimize additional sources of error. In practice, techniques such as iterative smoothing or reconstruction can be used to approximate the fine-space adjoint and reduce the cost.[5,9,12]

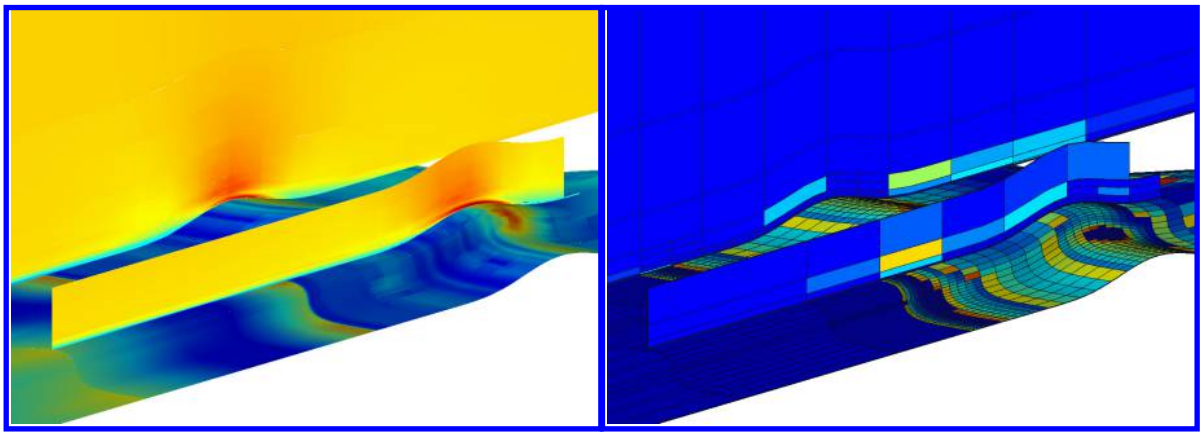
In an adjoint-consistent formulation, the discrete adjoint vector $\boldsymbol{\Psi}$, consists of expansion coefficients that, when combined with the finite element basis vectors, approximate the continuous adjoint vector. We can therefore visualize the adjoint in the same way as the state. Figure 5(a) shows one of the adjoint components for the three-dimensional bump problem under consideration. This field quantity should be interpreted as, at every point in space, the sensitivity of the output, in this case drag, to residual perturbations in one of the conservation equations, in this case x momentum. We note that in this work, the discretization is not strictly adjoint consistent, as the state gradients required for the turbulence source term are computed directly from the element-interior state, not accounting for the inter-element jumps. However, the impact of this inconsistency on the adaptive refinement has previously been shown to not be very large.[7]

V.B. Error Localization

The adjoint-weighted residual error estimate in Eqn. 10 can be localized to the elements, indexed by k, according to

$$J_H(\mathbf{U}_H) - J_h(\mathbf{U}_h) \approx -\boldsymbol{\Psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H) = -\sum_k \boldsymbol{\Psi}_{hk}^T \mathbf{R}_{hk}(\mathbf{U}_h^H) \quad \Rightarrow \quad \epsilon_k = \left| \boldsymbol{\Psi}_{hk}^T \mathbf{R}_{hk}(\mathbf{U}_h^H) \right|,$$

where the subscript k indicates degrees of freedom associated with element k, and the adaptive indicator ϵ_k is the absolute value of the elemental contribution. Figure 5(b) shows the distribution of error indicators for one of the adaptive iterations in the bump case. High values of the error indicator denote elements that contribute the most to the output error, and these should be targeted for refinement to reduce their residual and to improve their approximation of the adjoint.



(a) Drag adjoint (b) Adaptive indicator

Figure 5. Flow over a three-dimensional bump: the conservation of x-momentum component of the drag adjoint on a medium resolution mesh, and the localized output error which serves as the adaptive indicator.

V.C. Hanging-Node Hexahedral Mesh Refinement

The mesh refinement strategy used in this work is hanging-node subdivision of elements in an initially structured hexahedral mesh.[9,12,18] In this strategy, a fixed fraction, $f^{\text{frac}} = .03$, of elements with the highest error indicator is flagged for refinement. For the present results, we consider isotropic refinement in which each hexahedron is subdivided uniformly into eight hexahedra, as illustrated in Figure 6. This refinement is done in each element's reference space by employing the reference-to-global coordinate mapping in calculating the refined elements' geometry node coordinates. The refined elements inherit the same geometry approximation order and quadrature rules as the parent coarse element.

Elements created in a hanging-node refinement can be marked for h-refinement again in subsequent adaptation iterations. In this case, neighbors are divided in the minimal possible fashion, generally anisotropically, to keep one level of refinement difference between adjacent cells, as illustrated in Figure 6.

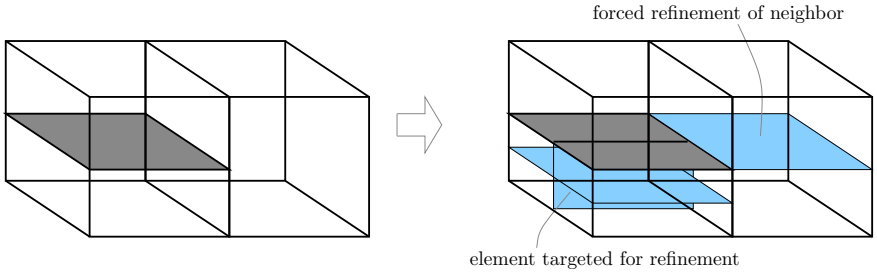


Figure 6. Hanging-node refinement of hexahedral elements, with a maximum of one level of refinement separating two elements. Following refinement of the bottom-left hexahedron, the neighbor element is refined in a minimal way to maintain a maximum of one level refinement difference between the elements.

On curved boundaries, new nodes are snapped to the geometry, a perturbation that is usually small when high-order curved elements are used. This perturbation is propagated into the interior of the first layer of elements adjacent to the boundary by a linear weight that drops to zero on the face opposite the boundary. A caveat of snapping to geometry with three-dimensional hanging-node meshes is that it is easy to generate invalid, non water-tight, meshes. Figure 7 illustrates this problem at a hanging node face when all boundary nodes are snapped to the geometry.

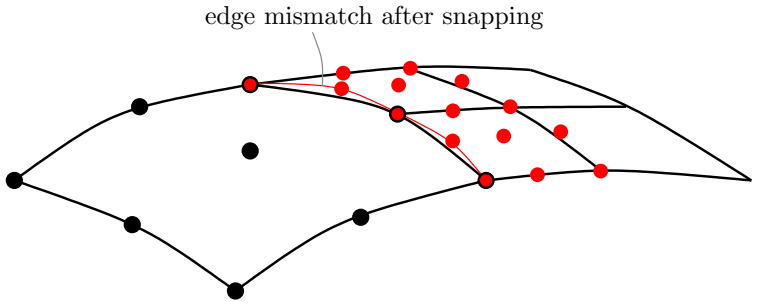


Figure 7. Snapping all boundary nodes to the true geometry can create non water-tight meshes for hanging-node meshes in three dimensions. Shown are adjacent boundary faces, the one on the right refined one more level compared to the one on the left.

The issue here is that when the nodes of the small elements on the refined side of the hanging-node face get snapped to the geometry, their edges and faces may no longer match the edges and face of the coarse-side element. One could constrain the perturbations of these fine-side nodes a priori to follow the perturbations of the coarse side, a constraint that would have to propagate to the affected faces as well. However, we employ a somewhat simpler a posteriori water-tightness fix to the mesh: following the snapping of boundary nodes and propagation to element interiors, we loop over all hanging node faces in the mesh and set all nodes of the fine-side elements on the face to match the geometry approximation on the coarse-side neighbors. A free-stream residual test, which would not pass for a non-water-tight mesh, verifies the validity of this fix.

VI. Results

VI.A. Problem Description

Figure 8 illustrates the setup of the present problem, flow over a three-dimensional bump. The free-stream Mach number is 0.2, the Reynolds number based on unit length, $L = 1$, is $Re_L = 3\times10^6$, and the Prandtl number is $Pr = 0.72$. A temperature-dependent viscosity is used, with the Sutherland reference temperature $T_{\text{ref}} = 300K$ and temperature constant $T_s = 110.33K$ in Eqn. 2. The free-stream turbulent viscosity level is set to 3 times the free-stream laminar viscosity.

The boundary conditions consist of total temperature and pressure at the inlet and static pressure at the outlet. An adiabatic no-slip condition is imposed on the wall boundary, and symmetry conditions are enforced on all other boundaries. The bottom wall extends horizontally from $x_{\text{LE}}(y)$ to $1.5x_{\text{LE}}(y)$, where $x_{\text{LE}}(y) = 0.3[\sin(\pi y)]^4$. That is, the leading and trailing edges of the wall vary in y. The vertical displacement of the bump is given by

$$z(x,y) = \begin{cases} .05\left[\sin\left(\pi(x - x_{\text{LE}}(y) - 0.3)/.9\right)\right]^4 & 0.3 < x - x_{\text{LE}}(y) < 1.2 \\ 0 & \text{otherwise} \end{cases}$$

The outputs of interest are the lift and drag coefficients on the bump wall, which are calculated as the force normalized by the free-stream dynamic pressure and the reference area, $S_{\text{ref}} = 1.5$.

Figure 9 shows the initial mesh of the bump used for the adaptive runs. It consists of 864 $q = 3$ elements, with uniform spacing in the y direction, uniform spacing in the x direction across the bump wall, logarithmic spacing in x in the pre-wall and post-wall regions extending to the farfield, and logarithmic spacing in the z direction. The first layer of elements adjacent to the wall has a Δz of 0.00108.

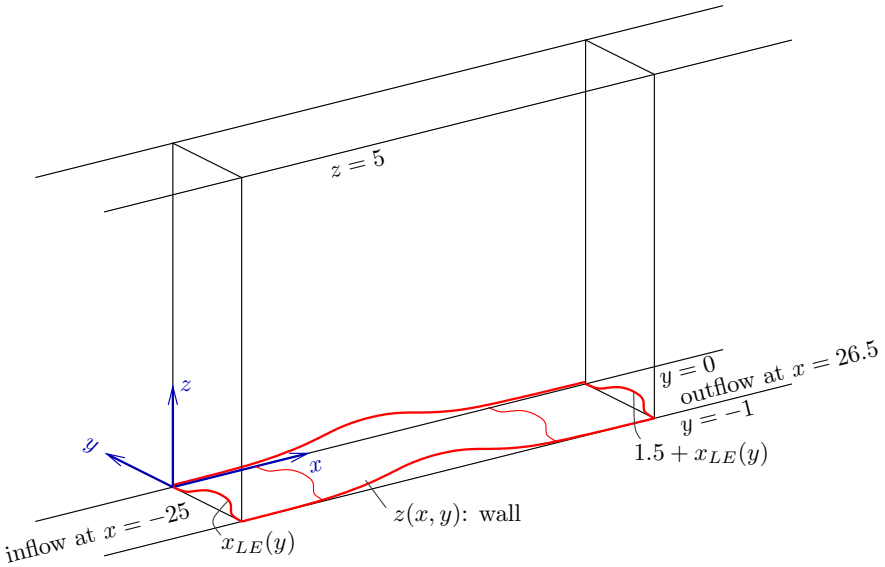
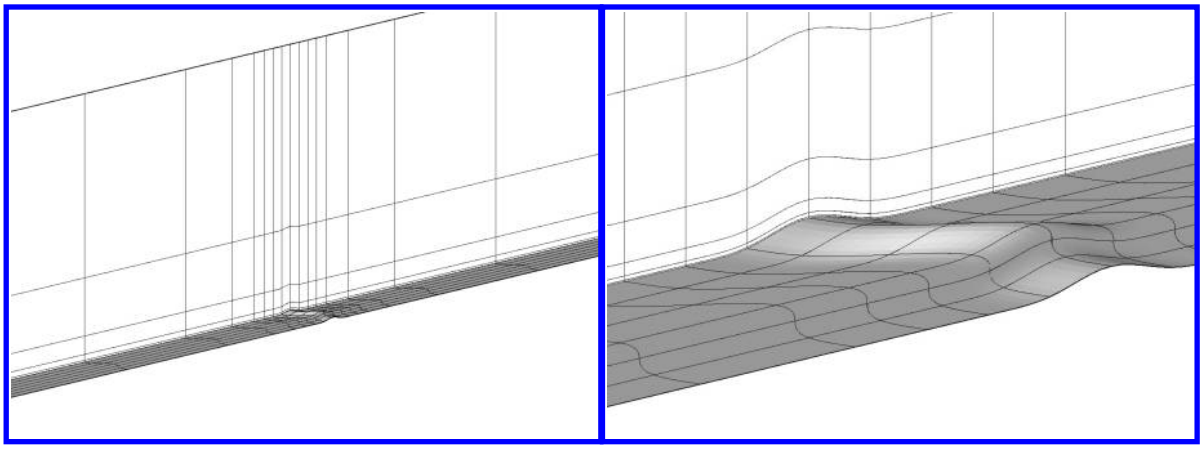


Figure 8. Setup for the bump problem.



(a) Zoomed-out view (b) Zoomed-in view

Figure 9. Initial mesh used for the adaptive simulations, consisting of 864 $q = 3$ elements.

VI.B. Output Convergence

Starting from the initial mesh, solutions at prescribed orders, p, were converged using the nonlinear solver outlined in Section IV.C with $\beta^{\text{CFL,increase}} = 1.2$. A full-order "tetrahedral" basis was used to reduce the size of the systems at high orders compared to the tensor-product "hexahedral" basis. Order continuation was used, where the solution at order p was used to initialize the state at order $p+1$. Free-stream initial conditions were used for $p = 1$. Following convergence on the initial mesh, adaptive refinements were performed at each order p using the hanging node strategy described in Section V.C with $f^{\text{frac}} = .03$. Both lift and drag output adjoints were used to drive separate refinement sequences. Since the solution was not expected to change much between the adaptive iterations, a more aggressive CFL increase factor of $\beta^{\text{CFL,increase}} = 5$ was used to speed up convergence.

Figure 10 shows the convergence of the drag and lift coefficients with adaptive refinement. The data are shown as output versus h, which is a measure of the mesh size defined as $h = (\text{dof})^{-1/3}$. For a mesh with isotropic elements, all of the same size, this would correspond to the mesh diameter. For our adapted and anisotropic meshes, this is no longer the case, and instead h is a surrogate measure of the number of degrees of freedom: the larger the h the fewer the degrees of freedom, the cheaper the cost. In addition to the raw output data at each order p, Figure 10 also shows the "corrected" outputs, which are obtained by subtracting the adjoint-weighted residual output error estimate, δJ in Eqn. 10, from the computed output. In the asymptotic regime and when the adjoint is solved to sufficient accuracy, this correction should yield a faster converging output.

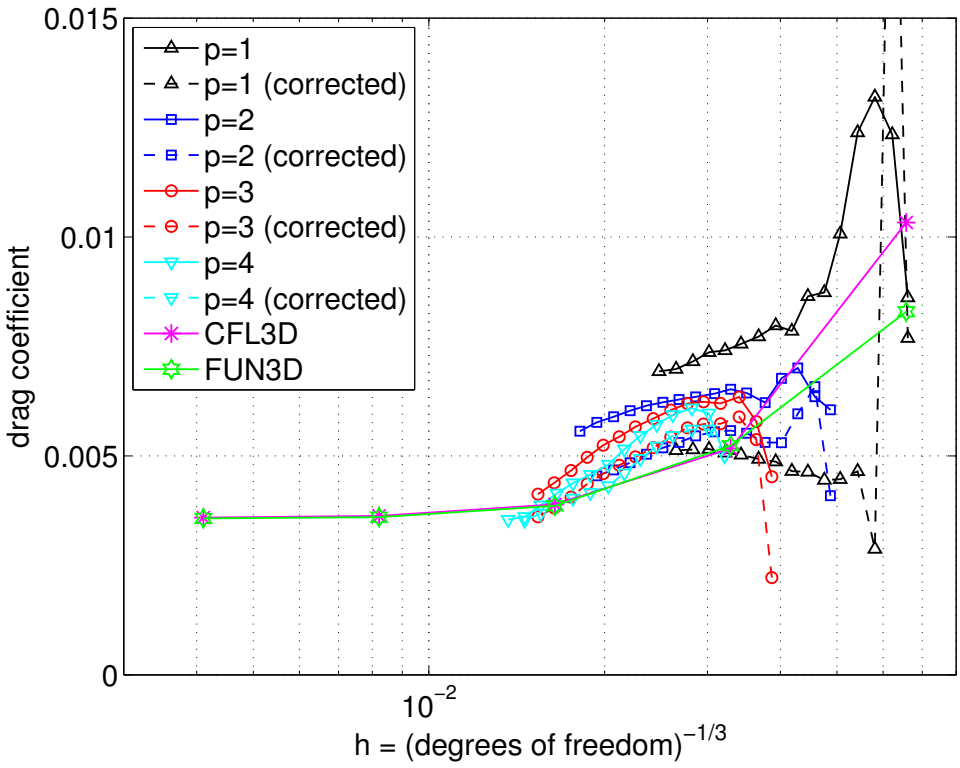
As shown in Figure 10 the DG results in this work appear to eventually, for fine meshes and high orders, converge to the same results obtained using the CFL3D and FUN3D codes, the data for which was provided on the NASA turbulence modeling resource website. At order $p = 1$, the convergence is not yet apparent and from the trend would require many more iterations (a much smaller h) to converge. The corrected $p = 1$ result is closer to the expected true value, but its convergence is also not very rapid. Order $p = 2$ performs better, with a more rapid convergence, especially for the lift output at the finer meshes. In addition, the corrected $p = 2$ result appears competitive with the CFL3D and FUN3D data at similar degrees of freedom. As the order increases, the convergence improves further, with the $p = 4$ raw and corrected data showing very good convergence to the results predicted by the fine CFL3D and FUN3D runs.

VI.C. Adapted Meshes

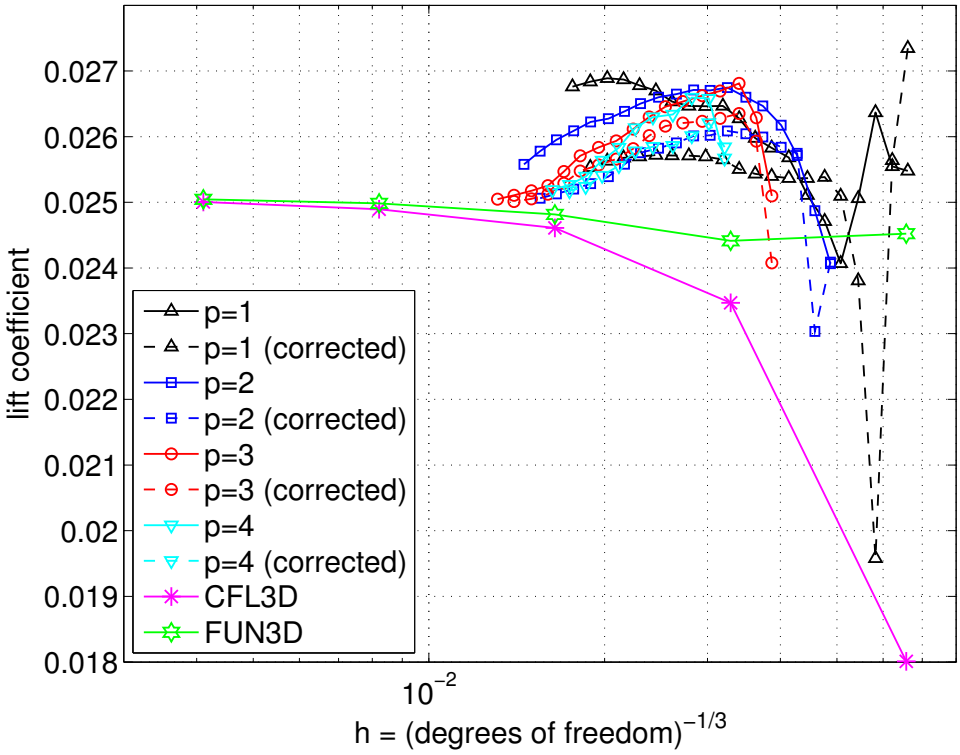
The adaptive refinement strategy selectively increases resolution in areas where nonzero residuals are present and affect the output of interest the most. The resulting adapted meshes are specific to the output calculated and to the order of approximation used. We take a close look at the meshes obtained at the 10^{th} adaptive iteration, for the drag and lift outputs, using approximation orders $p = 1$ an $p = 4$.

Figures 11 and 12 show the meshes adapted to the drag and lift outputs respectively, at $p = 1$ approximation order. We see similar, though not identical, refinement patterns. Both outputs are sensitive to the leading-edge region and to the crest of the bump. The regions before and after the bump are refined in slightly different fashions: the drag-adapted mesh shows increased resolution in these regions, especially near the wall at the $x = 1.2$ cut.

Figures 13 and 14 show the drag- and lift-adapted meshes for order $p = 4$. These are also at the 10^{th} adaptive iteration, and given the same adaptive refinement fraction, they have a number of elements similar to the $p = 1$ meshes. However, the distribution of resolution at $p = 4$ differs markedly from that at $p = 1$. We see much less resolution added away from the wall at the higher order, indicating that this region is relatively well-resolved compared to the region near the wall, where the density of elements is high for $p = 4$. This translates into more surface refinement and

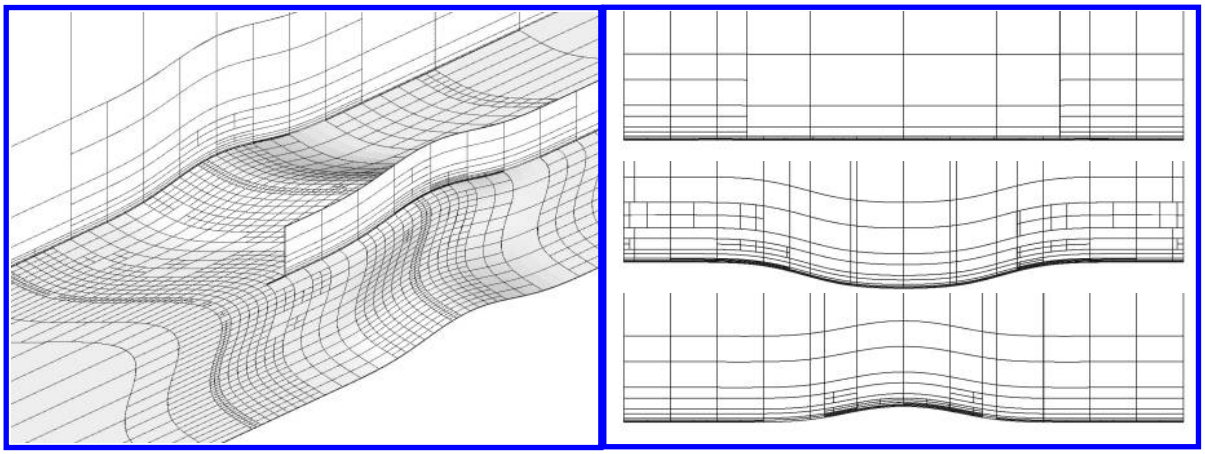


(a) Drag convergence



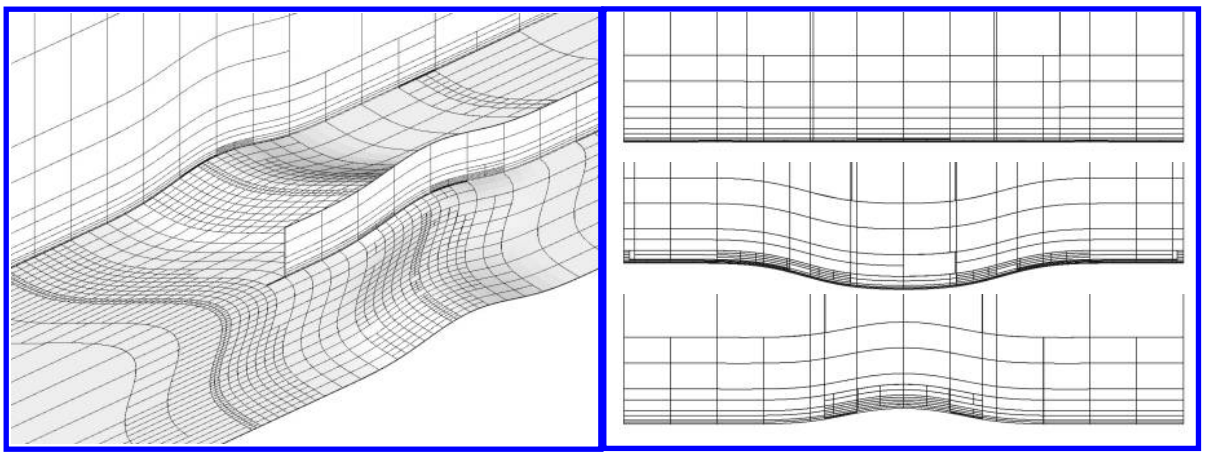
(b) Lift convergence

Figure 10. Convergence of drag and lift coefficients with mesh size, measured as the inverse third root of the degrees of freedom, for adaptive refinements on drag and lift outputs at various orders p. "Corrected" results denote outputs adjusted using the adjoint-weighted residual error estimate. CFL3D and FUN3D convergence histories are shown for comparison.



(a) 3D view (b) Cuts at $x = 0.3$ (top), $x = 0.75$, and $x = 1.2$

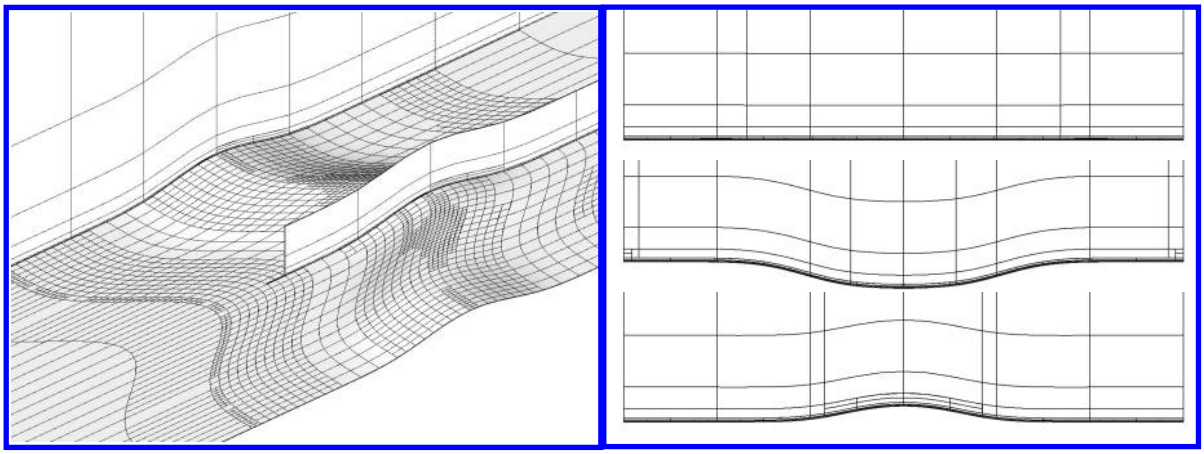
Figure 11. Mesh at the 10^{th} drag-based adaptive iteration (6213 elements) using $p = 1$ approximation.



(a) 3D view (b) Cuts at $x = 0.3$ (top), $x = 0.75$, and $x = 1.2$

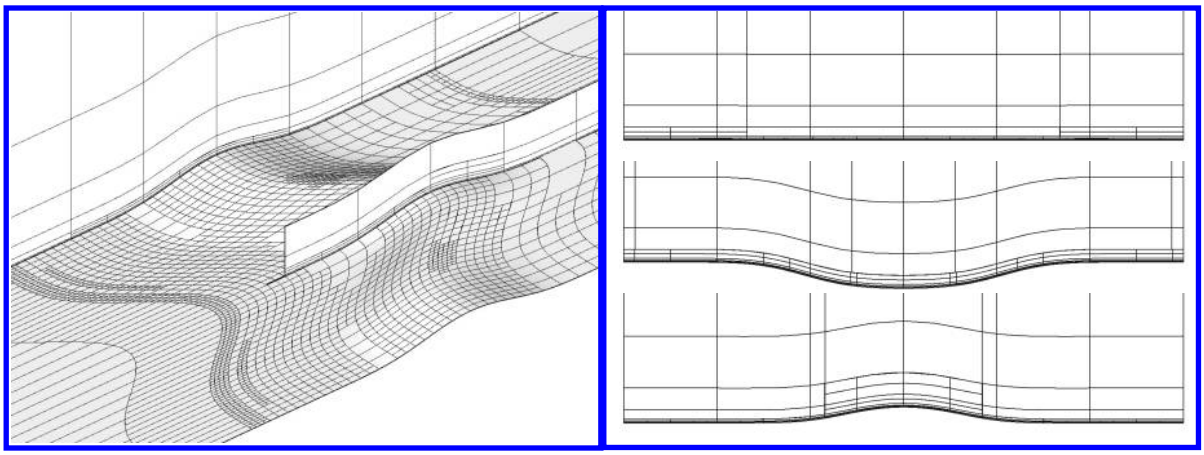
Figure 12. Mesh at the 10^{th} lift-based adaptive iteration (6357 elements) using $p = 1$ approximation.

more refinement of the boundary layer just off the wall at $p = 4$. Also, comparing the drag- and lift-adapted meshes, we see that the drag output requires more resolution on the bump crest and aft of the bump, whereas the lift output requires more resolution at the leading-edge region and in regions further away from the wall. The refinement patterns are relatively smooth, suggesting that the effect of the adjoint inconsistency in the turbulent source term treatment on the mesh refinement is not strong.



(a) 3D view (b) Cuts at $x = 0.3$ (top), $x = 0.75$, and $x = 1.2$

Figure 13. Mesh at the 10^{th} drag-based adaptive iteration (6280 elements) using $p = 4$ approximation.



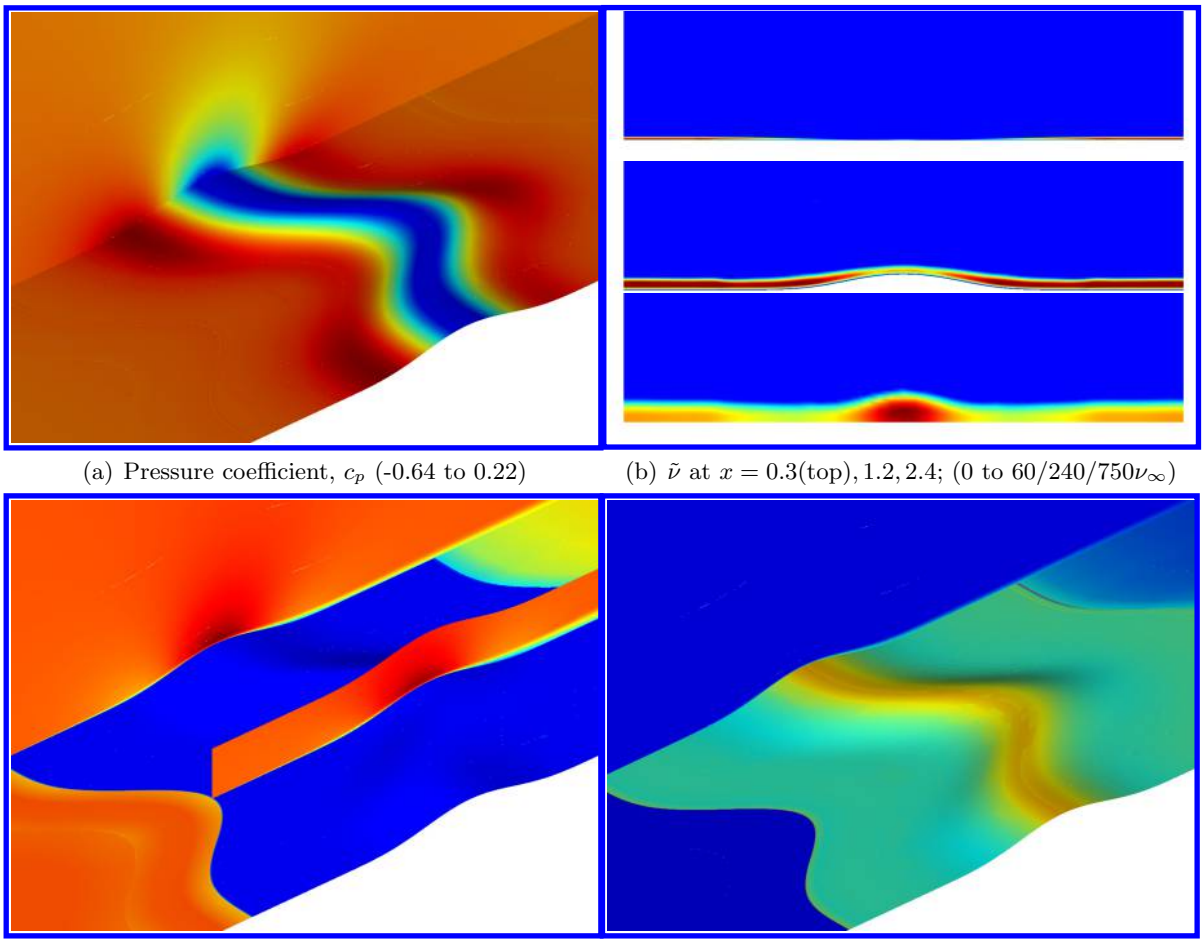
(a) 3D view (b) Cuts at $x = 0.3$ (top), $x = 0.75$, and $x = 1.2$

Figure 14. Mesh at the 10^{th} lift-based adaptive iteration (6627 elements) using $p = 4$ approximation.

VI.D. Field Plots

Figure 15 shows a selection of field quantities for a relatively well-resolved mesh, using $p = 4$ approximation order. The pressure contours are smooth and show the expected trend of low pressure over the crest of the bump and pressure buildup near the leading-edge regions. The turbulent viscosity contours at various x locations show an increasingly thicker turbulent viscosity layer, with three-dimensional structure – a strong core in the center at $x = 2.4$ downstream of the bump wall. The Mach number contours show the expected no-slip boundary condition at the bump

wall, and a thin boundary layer region that increases in thickness aft of the bump. Finally, the entropy contours show generation of entropy over the surface of the wall, with larger levels on the bump crest.



(a) Pressure coefficient, c_p (-0.64 to 0.22)

(b) $\tilde{\nu}$ at $x = 0.3(\text{top}), 1.2, 2.4$; (0 to $60/240/750\nu_\infty$)

(c) Mach contours (0 to 0.25)

(d) Entropy contours

Figure 15. Plots of several field quantities on a fine mesh, using $p = 4$ approximation order.

VII. Conclusions

This paper has presented the results of adaptive simulations for Reynolds-averaged turbulent flow over a three-dimensional bump, using a high-order discontinuous Galerkin finite element discretization. Details on the fluid model and the discretization were presented, notably regarding generation of curved meshes, calculation of the wall distance, and solution of the nonlinear system via pseudo-time stepping. Adaptive refinement was driven by an adjoint-weighted residual, with the output-specific discrete adjoint solution computed on a finer space providing residual sensitivity information. Hanging-node adaptation was employed to increase resolution, with a water-tightness enforcement to ensure mesh validity when snapping boundary points to the true geometry. The three-dimensional runs were performed on hexahedral elements but using a tetrahedral, full-order, basis in order to significantly reduce the size of the system for the same formal convergence order.

The adaptive results show that when using high-order approximation, both the drag and lift outputs converge to the asymptotic values predicted by previous CFL3D and FUN3D runs. De-

pending on the accuracy tolerance, there is an advantage of the high-order DG simulations of up to an order of magnitude in degrees of freedom (slightly more than a factor of two in h) over the CFL3D and FUN3D runs. However, this advantage does not directly extend to computational time, as degrees of freedom in high-order DG simulations are generally more computationally expensive than those for second-order finite volume simulations. We also note that the incorporation of the output error estimate improves the convergence of the output, but that there is still a large pre-asymptotic region in which the output is far away from the truth value and the correction, based on linearized theory, does not uncover all of the error. This effect is strongest at the lower orders, suggesting another reason for using high-order approximation.

Further benefits in adaptive three-dimensional simulations will likely be achieved by more general adaptive mechanics, e.g. using fully-unstructured tetrahedral meshes, or hybrid prismatic and tetrahedral meshes, for which the impact of the initial mesh quality is not as limiting. In addition, even for hanging-node refinement, anisotropic h refinement and combined h and p refinement should provide noticeable improvements compared to the isotropic h strategy employed in this work. Nevertheless, this study verified both the fluid model and the ability of adjoint-based refinement to successfully target the desired output with incremental resolution in small portions of the three-dimensional domain.

References

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

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

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

[4]Hartmann, R. and Houston, P., "Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations," *Journal of Computational Physics*, Vol. 183, No. 2, 2002, pp. 508–532.

[5]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.

[6]Allmaras, S., Johnson, F., and Spalart, P., "Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model," Seventh International Conference on Computational Fluid Dynamics (ICCFD7) 1902, 2012.

[7]Oliver, T. A., *A High–order, Adaptive, Discontinuous Galerkin Finite Elemenet Method for the Reynolds-Averaged Navier-Stokes Equations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2008.

[8]Oliver, T. A. and Darmofal, D. L., "Impact of turbulence model irregularity on high–order discretizations," AIAA Paper 2009-953, 2009.

[9]Ceze, M. A. and Fidkowski, K. J., "An anisotropic hp-adaptation framework for functional prediction," *American Institute of Aeronautics and Astronautics Journal*, Vol. 51, 2013, pp. 492–509.

[10]Yano, M., *An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2012.

[11]Ceze, M. A. and Fidkowski, K. J., "High-order output-based adaptive simulations of turbulent flow in two dimensions," AIAA Paper 2015–1532, 2015.

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

[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., "Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations," *International Journal for Numerical Methods in Fluids*, Vol. 40, 2002, pp. 197–207.

[15]Leicht, T. and Hartmann, R., "Error estimation and anisotropic mesh refinement for 3D laminar aerodynamic flow simulations," *Journal of Computational Physics*, Vol. 229, 2010, pp. 7344–7360.

[16]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.

[17]Sanjaya, D. P. and Fidkowski, K. J., "Improving high-order finite element approximation through geometrical warping," AIAA Paper 2015–2605, 2015.

[18]Ceze, M. A. and Fidkowski, K. J., "Output-driven anisotropic mesh adaptation for viscous flows using discrete choice optimization," AIAA Paper 2010-0170, 2010.