Adjoint-based Local Reanalysis of Nonlinear PDEs

Jared Crean*, Kinshuk Panda†, and Jason E. Hicken‡
Rensselaer Polytechnic Institute

During the design process, an engineer may need to predict the performance of many alternative geometries. A reanalysis method is attractive in this context, because it uses information at a baseline geometry to accelerate predictions of other geometries. In this work, we use an adjoint-based indicator to identify regions of the domain where perturbations to the solution will have the largest effect on an output of interest. The governing equations can then be re-solved on only the most influential regions, reducing computational cost. We refer to this method as *adjoint-based* local reanalysis. Computational results compare the output computed by reanalysis against re-solving the governing equation on the entire domain.

I. Introduction

High-fidelity computational fluid dynamics (CFD) can make accurate aerodynamic predictions over a wide range of operating conditions. However, the expense of CFD precludes its application to every potential configuration an engineer may wish to consider during the design process. This motivates the idea of reanalysis, whose goal is to leverage the known solution of one design to accelerate the analysis of subsequent designs.

This paper introduces a new approach to reanalysis that draws on several results from the adjoint-weighted-residual (AWR) method, which is used in both output-based error estimation and mesh adaptation. The AWR method defines an output quantity of interest and then computes the associated adjoint. The discretization error can then be estimated using the product of the adjoint and the residual of the governing equation evaluated on an enriched solution space. The product between the adjoint and the enriched residual can also be localized and used to drive mesh adaptation that targets the error in the output of interest.

In this paper, the AWR method is generalized to estimate the change in the output caused by a change in the design variables rather than a change in the discretization. This change in output is localized in a similar way to AWR error estimates.

An important observation from adjoint-based mesh refinement studies (see the review [1]) is that the output errors are focused on small regions of the domain. When considering several functionals, such as lift, drag, and pitching moment, the refined meshes are characterized by distinct patterns of elements near the leading and trailing edges, and along the stagnation streamline, depending on the functional, see, e.g., [2]. This localization of error suggests an approach to reanalysis. When the baseline and new configurations are similar, we can expect the (output-relevant) difference between the existing solution and the new solution to be localized in space. As a result, only a small portion of the domain must be re-solved in order to obtain an accurate value of the output quantity of interest on the new configuration.

The remainder of this paper describes an adjoint-based reanalysis technique for nonlinear partial differential equations (PDEs) based on the above ideas. Specifically, we will develop an adjoint-based reanalysis indicator in Section II, describe the method of re-solving the governing equations on the sub-domain in Section III, and apply the method in Section IV to two test problems. The values produced by reanalysis will be compared with the values computed by re-solving the PDE on the entire domain. Concluding remarks are given in Section V.

II. Reanalysis Theory

To begin, consider the discretization of a (general) governing PDE on a domain Ω, with boundary $\partial\Omega$. The discretized equation can be written in residual form as

$$\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x}) = \boldsymbol{0}, \tag{1}$$

*Graduate Student, Department of Mechanical, Aerospace, and Nuclear Engineering, and AIAA Student Member (creanj@rpi.edu)

†Graduate Student, Department of Mechanical, Aerospace, and Nuclear Engineering, and AIAA Student Member (pandak@rpi.edu)

‡Assistant Professor, Department of Mechanical, Aerospace, and Nuclear Engineering, and AIAA Member (hickej2@rpi.edu)

where $\boldsymbol{R} : \mathbb{R}^n \times \mathbb{R}^{n_d} \to \mathbb{R}^n$ is the residual, $\boldsymbol{u} \in \mathbb{R}^n$ is the solution, and $\boldsymbol{x} \in \mathbb{R}^{n_d}$ are the design variables. The number of degrees-of-freedom in the discretization is n, and the number of design variables is n_d. The design variables could represent, for example, the geometric parameterization of an airfoil. Suitable boundary conditions on $\partial\Omega$ are required for a well-posed problem and are assumed to be incorporated in (1). Note that the solution $\boldsymbol{u}$ can be defined as a function of $\boldsymbol{x}$ using (1) based on the implicit function theorem. Thus, for the remainder of the paper $\boldsymbol{u}(\boldsymbol{x})$ denotes the solution $\boldsymbol{u}$ obtained by solving (1) at a given $\boldsymbol{x}$.

We also introduce an output functional of interest

$$J(\boldsymbol{u}, \boldsymbol{x}) : \mathbb{R}^n \times \mathbb{R}^{n_d} \to \mathbb{R}, \tag{2}$$

which can, for example, be the lift on an airfoil. For the output (2), the output-based reanalysis problem can be formally stated as

$$\begin{aligned} &\text{Given} && \boldsymbol{u}, \boldsymbol{x}, \Delta\boldsymbol{x} \\ &\text{Find} && J(\boldsymbol{u}_{\text{new}}, \boldsymbol{x} + \Delta\boldsymbol{x}) \approx J(\boldsymbol{u}(\boldsymbol{x} + \Delta\boldsymbol{x}), \boldsymbol{x} + \Delta\boldsymbol{x}) \\ &\text{Such that} && \boldsymbol{R}(\boldsymbol{u}_{\text{new}}, \boldsymbol{x} + \Delta\boldsymbol{x}) \approx \boldsymbol{0}. \end{aligned} \tag{3}$$

After the governing equation has been solved at an initial design, $\boldsymbol{x}$, and the update to the design, $\Delta\boldsymbol{x}$, has been specified, the reanalysis problem is to compute a new functional value $J(\boldsymbol{u}_{\text{new}}, \boldsymbol{x} + \Delta\boldsymbol{x})$. This functional value must be a good approximation to the functional value computed using the exact solution at the new design, that is, the value $J(\boldsymbol{u}(\boldsymbol{x} + \Delta\boldsymbol{x}), \boldsymbol{x} + \Delta\boldsymbol{x})$. An additional constraint is that the solution $\boldsymbol{u}_{\text{new}}$ approximately satisfies the governing equation. This constraint distinguishes reanalysis from more general approximation theory methods, such as Taylor series approximations of J, which do not enforce the condition $\boldsymbol{R}(\boldsymbol{u}_{\text{new}}, \boldsymbol{x} + \Delta\boldsymbol{x}) \approx \boldsymbol{0}$.

The approach developed below computes the solution $\boldsymbol{u}_{\text{new}}$ by solving $\boldsymbol{R}(\boldsymbol{u}_{\text{new}}, \boldsymbol{x} + \Delta\boldsymbol{x}) = \boldsymbol{0}$ on a subset of the domain identified by an adjoint-based indicator. The approximate functional value is then computed using the approximate solution $\boldsymbol{u}_{\text{new}}$.

A. Steady Adjoint

The adjoint plays a vital role in the proposed reanalysis method, so this section reviews the steady adjoint equation and its use in evaluating total derivatives. Readers familiar with the adjoint may safely proceed to the following section.

Adjoints are used in numerous scientific and engineering disciplines to compute sensitivities of constrained functions [3–5]. In aerodynamic shape optimization, for example, the adjoint is used to compute the derivative of an aerodynamic force or moment with respect to changes in the geometry, while treating the governing equations as a constraint.

To illustrate the usefulness of the adjoint, consider computing the derivative of J with respect to changes in the design variables This derivative can be expressed as

$$\frac{\mathrm{d}J(\boldsymbol{u}, \boldsymbol{x})}{\mathrm{d}\boldsymbol{x}} = \frac{\partial J(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{x}} + \frac{\partial J(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{u}} \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{x}}, \tag{4}$$

where the total derivative $\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{x}}$ takes into account dependence of $J(\boldsymbol{u}, \boldsymbol{x})$ on $\boldsymbol{x}$, both directly and indirectly through $\boldsymbol{u}(\boldsymbol{x})$. The direct sensitivity $\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{x}}$ is found by solving

$$\frac{\mathrm{d}\boldsymbol{R}(\boldsymbol{u}(\boldsymbol{x}), \boldsymbol{x})}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{u}} \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{x}} + \frac{\partial \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{x}} = \boldsymbol{0}. \tag{5}$$

Substituting $\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{x}}$ from (5) into (4), we find

$$\begin{aligned} \frac{\mathrm{d}J(\boldsymbol{u}, \boldsymbol{x})}{\mathrm{d}\boldsymbol{x}} &= \frac{\partial J(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{x}} - \frac{\partial J(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{u}} \left(\frac{\partial \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{u}} \right)^{-1} \frac{\partial \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{x}} \\ &= \frac{\partial J(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{x}} + \boldsymbol{\psi}^T \frac{\partial \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})}{\partial \boldsymbol{x}}, \end{aligned} \tag{6}$$

where the adjoint $\boldsymbol{\psi}$ is the solution of

$$\frac{\partial J(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{u}} + \boldsymbol{\psi}^T \frac{\partial \boldsymbol{R}(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{u}} = \boldsymbol{0}. \tag{7}$$

Thus we can use the adjoint to compute derivatives of J, taking into account that $\boldsymbol{u}$ is constrained to satisfy (1). The evaluation of (6) is efficient for large numbers of design variables because it requires only partial derivatives and transposed Jacobian-vector products, and the cost of solving for the adjoint $\boldsymbol{\psi}$ is independent of the number of design variables.

B. Reanalysis

We now derive an adjoint-based expression for $\Delta J = J(\boldsymbol{u}(\boldsymbol{x}+\Delta\boldsymbol{x}), \boldsymbol{x}+\Delta\boldsymbol{x}) - J(\boldsymbol{u}(\boldsymbol{x}), \boldsymbol{x})$, that is, the exact change in the output functional given a prescribed change in the design variables. This expression for ΔJ will form the basis of a localization strategy that indicates where to re-solve. The derivation is similar to adjoint-based error estimation [1]; however, unlike error estimation, the discretization is fixed and the change in J is caused by $\Delta\boldsymbol{x}$, with $\boldsymbol{u}(\boldsymbol{x})$ treated as a dependent variable.

We begin by expressing the change in the output as an integral with respect to the parameter θ, which controls the progression of the design from $\boldsymbol{x}$ to $\boldsymbol{x}+\Delta\boldsymbol{x}$:

$$\begin{aligned} J(\boldsymbol{u}(\boldsymbol{x}+\Delta\boldsymbol{x}), \boldsymbol{x}+\Delta\boldsymbol{x}) - J(\boldsymbol{u},\boldsymbol{x}) &= \int_0^1 \frac{\mathrm{d}J(\boldsymbol{u}(\boldsymbol{x}),\boldsymbol{x})}{\mathrm{d}\theta}\bigg|_{\boldsymbol{x}=\boldsymbol{x}+\theta\Delta\boldsymbol{x}} \mathrm{d}\theta \\ &= \int_0^1 \left(\frac{\partial J(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} + \frac{\partial J(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{u}} \frac{\mathrm{d}\boldsymbol{u}(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}}\Delta\boldsymbol{x} \right)\bigg|_{\boldsymbol{x}=\boldsymbol{x}+\theta\Delta\boldsymbol{x}} \mathrm{d}\theta \end{aligned} \tag{8}$$

where $\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{x}}\Delta\boldsymbol{x}$ is found by right-multiplying (5) by $\Delta\boldsymbol{x}$ and solving:

$$\frac{\partial \boldsymbol{R}(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{u}} \left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{x}}\Delta\boldsymbol{x} \right) = -\left(\frac{\partial \boldsymbol{R}(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} \right). \tag{9}$$

Substituting the resulting expression for $\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{x}}\Delta\boldsymbol{x}$ into (8) gives

$$J(\boldsymbol{u}(\boldsymbol{x}+\Delta\boldsymbol{x}), \boldsymbol{x}+\Delta\boldsymbol{x}) - J(\boldsymbol{u},\boldsymbol{x}) = \int_0^1 \left(\frac{\partial J(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} - \frac{\partial J(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{u}} \left(\frac{\partial \boldsymbol{R}(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{u}} \right)^{-1} \frac{\partial \boldsymbol{R}(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} \right)\bigg|_{\boldsymbol{x}=\boldsymbol{x}+\theta\Delta\boldsymbol{x}} \mathrm{d}\theta. \tag{10}$$

Using the definition of the adjoint, (7), gives the final expression

$$J(\boldsymbol{u}(\boldsymbol{x}+\Delta\boldsymbol{x}), \boldsymbol{x}+\Delta\boldsymbol{x}) - J(\boldsymbol{u},\boldsymbol{x}) = \int_0^1 \left(\frac{\partial J(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} + \boldsymbol{\psi}^T \frac{\partial \boldsymbol{R}(\boldsymbol{u},\boldsymbol{x})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} \right)\bigg|_{\boldsymbol{x}=\boldsymbol{x}+\theta\Delta\boldsymbol{x}} \mathrm{d}\theta. \tag{11}$$

Note that the adjoint in the integrand of (11) must be considered an implicit function of $\boldsymbol{x}$ just as the solution $\boldsymbol{u}$ is.

C. Approximations

Evaluating (11) gives the exact change in the functional and, in principle, solves the reanalysis problem (3). In practice, evaluating the integral requires the solution of the governing equation and adjoint along the path from $\boldsymbol{x}$ to $\boldsymbol{x}+\Delta\boldsymbol{x}$, defeating the purpose of reanalysis. To avoid this, an approximation to (11) can be used to identify the elements where changing the design variables will also change the solution variables in a way that affects the functional value. The governing equation can then be solved on this subset of elements for $\boldsymbol{u}_{\text{new}}$ and a new functional value, $J(\boldsymbol{u}_{\text{new}}, \boldsymbol{x}+\Delta\boldsymbol{x})$, computed.

First, we introduce the approximate change in solution $\Delta\tilde{\boldsymbol{u}}$

$$\Delta\tilde{\boldsymbol{u}} = \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{x}}\Delta\boldsymbol{x}, \tag{12}$$

which can be computed by solving (9). Solving for $\Delta\tilde{\boldsymbol{u}}$ requires information which is already available, namely the baseline solution $\boldsymbol{u}(\boldsymbol{x})$.

To approximate the integral in (11), the integrand is evaluated at the approximate midpoint $\boldsymbol{u}_{\text{mid}} = \boldsymbol{u} + 0.5\Delta\tilde{\boldsymbol{u}}$, $\boldsymbol{x}_{\text{mid}} = \boldsymbol{x} + 0.5\Delta\boldsymbol{x}$:

$$J(\boldsymbol{u}(\boldsymbol{x}+\Delta\boldsymbol{x}), \boldsymbol{x}+\Delta\boldsymbol{x}) - J(\boldsymbol{u},\boldsymbol{x}) \approx \frac{\partial J(\boldsymbol{u}_{\text{mid}}, \boldsymbol{x}_{\text{mid}})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} + \boldsymbol{\psi}_{\text{mid}}^T \frac{\partial \boldsymbol{R}(\boldsymbol{u}_{\text{mid}}, \boldsymbol{x}_{\text{mid}})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x}. \tag{13}$$

For nonlinear problems, $\boldsymbol{u} + 0.5\Delta\tilde{\boldsymbol{u}}$ is not the solution of (1) at $\boldsymbol{x} + 0.5\Delta\boldsymbol{x}$; however, finding the true solution would require re-solving the governing equation on the entire mesh, which would make reanalysis more expensive than solving (1) at the new design. Using the solution $\boldsymbol{u} + 0.5\Delta\tilde{\boldsymbol{u}}$ incurs some approximation error, but significantly reduces the cost of the computing the reanalysis indicator. The quantity $\boldsymbol{\psi}_{\text{mid}}$ is the adjoint computed at the approximate midpoint. Evaluating (7) at the midpoint gives

$$\frac{\partial J(\boldsymbol{u}_{\text{mid}}, \boldsymbol{x}_{\text{mid}})}{\partial \boldsymbol{u}} + \boldsymbol{\psi}_{\text{mid}}^T \frac{\partial \boldsymbol{R}(\boldsymbol{u}_{\text{mid}}, \boldsymbol{x}_{\text{mid}})}{\partial \boldsymbol{u}} = \boldsymbol{0}. \tag{14}$$

In addition to the midpoint approximation, we also neglect the first term in (13). Because this term is a partial derivative with respect to the design variables $\boldsymbol{x}$, it is not expected to provide a good indicator of where updating the solution $\boldsymbol{u}$ will have the greatest effect on the functional J.

With these simplifications, the reanalysis indicator is

$$\boldsymbol{\psi}_{\text{mid}}^T \frac{\partial \boldsymbol{R}(\boldsymbol{u}_{\text{mid}}, \boldsymbol{x}_{\text{mid}})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x}. \tag{15}$$

This indicator is similar to the error estimate used to drive adjoint-based mesh refinement, $\boldsymbol{\psi}^T\boldsymbol{R}(\boldsymbol{u}_H^h)$; see, for example, [1]. For mesh refinement, $\boldsymbol{R}(\boldsymbol{u}_H^h)$ measures solution inadequacy by evaluating the fine-space residual with the coarse space solution $\boldsymbol{u}_H^h$. For reanalysis, $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}\Delta\boldsymbol{x}$ is the approximate change in residual as the design changes from $\boldsymbol{x}$ to $\boldsymbol{x} + \Delta\boldsymbol{x}$. In both cases the adjoint weighting indicates where the perturbations to the residual has the greatest impact on the output.

D. Localization of the Reanalysis Indicator

To localize the contribution of a given element* i to the change in the output, we consider a non-overlapping triangulation $\mathcal{T}$ with k elements. Similar to adjoint-based mesh refinement [1], the reanalysis indicator $\boldsymbol{\epsilon} \in \mathbb{R}^k$ on the ith element is computed as

$$\epsilon_i = \sum_{j\in\mathcal{T}_i} \left| (\boldsymbol{\psi}_{\text{mid}})_j^T \left(\frac{\partial \boldsymbol{R}(\boldsymbol{u}_{\text{mid}}, \boldsymbol{x}_{\text{mid}})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} \right)_j \right|. \tag{16}$$

A subscript denotes accessing a particular element of a vector; for example, in the equation above $(\boldsymbol{\psi}_{\text{mid}})_j$ indicates the element of the adjoint vector for node j of element i, and $\boldsymbol{\epsilon}_i$ is the ith element of $\boldsymbol{\epsilon}$.

E. The Linear Case

To investigate the behavior of (13), we consider the case where the governing equation and functional are linear in both the solution and design variables. We will show that, for linear problems, (13) produces an exact error estimate for the functional.

The residual and functional are written as general linear functions

$$\begin{aligned} \boldsymbol{R}(\boldsymbol{u},\boldsymbol{x}) &= \mathsf{A}\boldsymbol{u} + \mathsf{B}\boldsymbol{x} + \boldsymbol{c} \\ J(\boldsymbol{u},\boldsymbol{x}) &= \mathsf{C}\boldsymbol{u} + \mathsf{D}\boldsymbol{x} + \boldsymbol{e} \end{aligned} \tag{17}$$

*Here we assume a finite-element-type discretization, but the method can be adapted to other types of discretizations where A, B, C, and D are constant matrices and $\boldsymbol{c}$ and $\boldsymbol{e}$ are constant vectors. With these definitions, it is easily seen that $\boldsymbol{\psi}_{\text{mid}} = \boldsymbol{\psi}$.

Evaluating the right-hand side of (13) gives the estimated change in the output ΔJ_{est}:

$$\begin{aligned}\Delta J_{\text{est}} &= \frac{\partial J}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} + \boldsymbol{\psi}^T\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}\Delta\boldsymbol{x}\\ &= \frac{\partial J}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} - \frac{\partial J}{\partial \boldsymbol{u}}\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right)^{-1}\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}\Delta\boldsymbol{x}\\ &= \mathsf{D}\Delta\boldsymbol{x} - \mathsf{C}\mathsf{A}^{-1}\mathsf{B}\Delta\boldsymbol{x}.\end{aligned} \tag{18}$$

It is easily shown from (17) that the change in solution $\Delta\boldsymbol{u}$ for a given change in the design variables $\Delta\boldsymbol{x}$ is $\Delta\boldsymbol{u} = -\mathsf{A}^{-1}\mathsf{B}\Delta\boldsymbol{x}$. Using this result, the exact change in functional value (i.e. the left-hand side of (13)) is

$$\begin{aligned}\Delta J &= J(\boldsymbol{u}+\Delta\boldsymbol{u}, \boldsymbol{x}+\Delta\boldsymbol{x}) - J(\boldsymbol{u},\boldsymbol{x})\\ &= \mathsf{C}(\boldsymbol{u}+\Delta\boldsymbol{u}) + \mathsf{D}(\boldsymbol{x}+\Delta\boldsymbol{x}) + \boldsymbol{e} - (\mathsf{C}\boldsymbol{u} + \mathsf{D}\boldsymbol{x} + \boldsymbol{e})\\ &= \mathsf{C}\Delta\boldsymbol{u} + \mathsf{D}\Delta\boldsymbol{x}\\ &= -\mathsf{C}\mathsf{A}^{-1}\mathsf{B}\Delta\boldsymbol{x} + \mathsf{D}\Delta\boldsymbol{x}.\end{aligned} \tag{19}$$

This is equal to ΔJ_{est} in (18), thus the error indicator (13) is exact for linear problems.

III. Re-solve Methodology

A. Sub-domain Re-solve

This section describes how (16) is used to identify the sub-domain in which the governing equations are re-solved. The new solution in the sub-domain is then combined with a linear approximation to the solution in the original domain to compute a new functional value.

Recall that Ω is the computational domain. The governing equation is re-solved on a subset of the original domain; we will denote this subset with Ω'. The subset will be identified using the indicator (16). Elements from the original domain Ω are added to Ω' in order of decreasing $\boldsymbol{\epsilon}_i$ value until Ω' contains 99% of the total reanalysis indicator $\sum_i \boldsymbol{\epsilon}_i$.

This simple heuristic is not unique and differs from the fixed-fraction rule often used in adjoint-based mesh adaptation [1]. The benefit of this heuristic is that it allows the algorithm to decide how much of the domain to re-solve on depending on the distribution of the indicator. In cases where the sensitivity is localized, the algorithm can select only a small region. If the output is sensitive to the solution in large areas of the domain, the algorithm can detect this and re-solve accordingly. In the case of reanalysis, there is a fixed upper bound on the cost of re-solving, namely the cost of the original solve plus the cost of computing the indicator. As a result, the excessive refinement problem, described in [1], that can occur in adjoint-based mesh refinement without the fixed-fraction rule is not as severe. The results in Section IV confirm this heuristic does not select an excessive number of elements.

The new domain Ω' requires additional boundary conditions for the governing equation to be well-posed. Let the boundary of Ω' be denoted $\partial\Omega' = \partial\Omega'_{\text{orig}} \cup \partial\Omega'_{\text{new}}$, where $\partial\Omega'_{\text{orig}}$ is the part of the boundary common to Ω and Ω', and $\partial\Omega'_{\text{new}}$ is the part of the boundary that was previously on the interior of Ω. On $\partial\Omega'_{\text{orig}}$, the original boundary conditions are applied. On $\partial\Omega'_{\text{new}}$, new boundary conditions are introduced to enforce consistency with the solution on Ω.

For the discontinuous-Galerkin (DG) discretizations used in this work, the natural approach for imposing both boundary conditions and inter-element coupling is to use a numerical flux function $f^*(\boldsymbol{u}_\kappa, \boldsymbol{u}_\nu)$. For element interfaces, $\boldsymbol{u}_\kappa$ and $\boldsymbol{u}_\nu$ are the solutions on the two sides of the interface. For boundary faces, $\boldsymbol{u}_\nu$ is the boundary state and $\boldsymbol{u}_\kappa$ is the trace of the numerical solution at the boundary of element κ.

On $\partial\Omega'_{\text{new}}$, the numerical solution from Ω is weakly imposed as a boundary condition. Specifically, if γ is the face of an element on $\partial\Omega'_{\text{new}}$ shared by elements κ and ν, where $\kappa \subset \Omega'$ and $\nu \not\subset \Omega'$, then the boundary flux is $g(\boldsymbol{u}_\kappa) = f^*(\boldsymbol{u}_\kappa, \boldsymbol{u}_\nu)$. Because element ν is not in the sub-domain Ω', $\boldsymbol{u}_\nu$ remains constant while $\boldsymbol{u}_\kappa$, $\kappa \subset \Omega'$ is updated.

One choice for the boundary state on $\partial\Omega'_{\text{new}}$ is the original solution $\boldsymbol{u}(\boldsymbol{x})$; however, preliminary results showed this produces inaccurate functional values. For example, the baseline state does not reflect changes in circulation that result when lift increases or decreases. Instead, the approximate solution $\boldsymbol{u} + \Delta\tilde{\boldsymbol{u}}$ is imposed. This approximation comes at no additional cost, since $\Delta\tilde{\boldsymbol{u}}$ is computed in the course of evaluating (16).

With these boundary conditions, (1) can be solved for the new solution $\boldsymbol{u}_{\text{new}}$ on the subset Ω', leaving the linear solution $\boldsymbol{u} + \Delta\tilde{\boldsymbol{u}}$ on the remainder of the domain. The updated solution can be used to compute a new value of the functional, thereby solving the reanalysis problem (3). The total cost of reanalysis is two linear solves, one for $\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{x}}\Delta\boldsymbol{x}$ and one for $\boldsymbol{\psi}_{\text{mid}}$, plus the cost of the nonlinear solve on the sub-domain Ω'.

An astute reader may notice that, starting from (8), the reanalysis indicator (15) can be written in terms of $\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{x}}\Delta\boldsymbol{x}$, rather than the adjoint $\boldsymbol{\psi}$. Given that $\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}\boldsymbol{x}}\Delta\boldsymbol{x}$ must be computed to impose the boundary conditions, this suggests the adjoint solve could be avoided. While the estimated change in J is the same for (8) and (11), the localizations of these equations into the form of (16) are not. Consider the example of computing the lift on an airfoil in an inviscid compressible flow, which will be explored in Section IV. The $\frac{\partial J}{\partial \boldsymbol{u}}$ term in (8) is non-zero only in the elements on the surface of the airfoil. The adjoint, however, is non-zero throughout the domain, in general. Because of the hyperbolic nature of the Euler equations, changes in the solution throughout the domain can propagate and affect the solution on the surface of the airfoil. We therefore expect (11) to provide a better basis for selecting elements to re-solve.

B. Reanalysis Effectiveness Estimate

Given that the approximation $\boldsymbol{u} + \Delta\tilde{\boldsymbol{u}}$ is available, we can compute an approximate functional value

$$J(\boldsymbol{u}(\boldsymbol{x} + \Delta\boldsymbol{x}), \boldsymbol{x} + \Delta\boldsymbol{x}) \approx J(\boldsymbol{u} + \Delta\tilde{\boldsymbol{u}}, \boldsymbol{x} + \Delta\boldsymbol{x}). \tag{20}$$

From this, we can estimate the benefit of re-solving the governing equation over the approximation provided by $\Delta\tilde{\boldsymbol{u}}$. As shown in Section II.E, the error estimate is exact for linear problems, and is presumably quite accurate for weakly nonlinear problems. In these cases, the approximate solution $\boldsymbol{u} + \Delta\tilde{\boldsymbol{u}}$ can be returned rather than re-solving the governing equation. In order to detect this condition, we compute the difference between the approximate functional (20) and the approximate J from (13)

$$\Delta J_{\text{re-solve}} = J(\boldsymbol{u} + \Delta\tilde{\boldsymbol{u}}, \boldsymbol{x} + \Delta\boldsymbol{x}) - \left(J(\boldsymbol{u}, \boldsymbol{x}) + \frac{\partial J(\boldsymbol{u}_{\text{mid}}, \boldsymbol{x}_{\text{mid}})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} + \boldsymbol{\psi}^T_{\text{mid}}\frac{\partial \boldsymbol{R}(\boldsymbol{u}_{\text{mid}}, \boldsymbol{x}_{\text{mid}})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x} \right). \tag{21}$$

This expression provides an estimate of the benefit to re-solving the governing equation rather than evaluating the functional with the linear approximation of the solution, $J(\boldsymbol{u} + \Delta\tilde{\boldsymbol{u}}, \boldsymbol{x} + \Delta\boldsymbol{x})$. With a suitable tolerance, $\tau_{\text{re-solve}}$, this estimate can be used to avoid re-solving if the linear approximation is sufficiently accurate.

C. Algorithm Summary

The reanalysis algorithm is shown in Algorithm 1. The algorithm sets the new solution $\boldsymbol{u}_{\text{new}}$ to $\boldsymbol{u} + \Delta\tilde{\boldsymbol{u}}$ and re-solves only if the tolerance on (21) indicates it will be beneficial. If a re-solve is not beneficial, the approximate solution $\boldsymbol{u} + \Delta\tilde{\boldsymbol{u}}$ is used to compute the new functional value. For this work, we set $\tau_{\text{re-solve}} = 0.01$.

Algorithm 1: Reanalysis algorithm

```
Input: $\boldsymbol{u}(\boldsymbol{x}), \boldsymbol{x}, \Delta\boldsymbol{x}$
Solve for $\Delta\tilde{\boldsymbol{u}}$ using (9)
Calculate $\boldsymbol{u}_{\text{mid}} = \boldsymbol{u}(\boldsymbol{x}) + 0.5\Delta\tilde{\boldsymbol{u}}$ and $\boldsymbol{x}_{\text{mid}} = \boldsymbol{x} + 0.5\Delta\boldsymbol{x}$
Solve for $\boldsymbol{\psi}_{\text{mid}}$ using (14)
Compute $\Delta J_{\text{re-solve}}$ with (21)
Set $\boldsymbol{u}_{\text{new}} = \boldsymbol{u}(\boldsymbol{x}) + \Delta\tilde{\boldsymbol{u}}$ on $\Omega$
if $|\Delta J_{re\text{-}solve}| > \tau_{re\text{-}solve}$ then
    Use indicator (16) to identify sub-domain $\Omega'$
    Solve (1) with additional boundary conditions from $\boldsymbol{u}(\boldsymbol{x}) + \Delta\tilde{\boldsymbol{u}}$ to update $\boldsymbol{u}_{\text{new}}$ on $\Omega'$
end
return $J(\boldsymbol{u}_{new}, \boldsymbol{x} + \Delta\boldsymbol{x})$
```

IV. Results

This section presents two examples of reanalysis: a steady isentropic vortex problem and an airfoil lift-prediction problem. Both cases are discretized using Summation-by-Parts (SBP) operators and a DG scheme. The governing equations are the steady Euler equations in two dimensions:

$$\frac{\partial \boldsymbol{F}_x}{\partial x} + \frac{\partial \boldsymbol{F}_y}{\partial y} = 0, \tag{22}$$

where the state is the vector of conservative variables, $[\rho, \rho u, \rho v, e]^T$, and the flux vectors are

$$\boldsymbol{F}_x = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (e+p)u \end{bmatrix}, \qquad \text{and} \qquad \boldsymbol{F}_y = \begin{bmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ (e+p)v \end{bmatrix}. \tag{23}$$

The calorically perfect ideal gas law is used to define the pressure and close the system. The equations are discretized with a $p = 1$ discontinuous-Galerkin (DG) method, which is described in greater detail in Appendix A. The no-penetration wall boundary condition is discretized in an adjoint-consistent manner to ensure a smooth adjoint field [6].

A. Steady Isentropic Vortex

The first test problem is a steady isentropic vortex. This problem will examine the behavior of the reanalysis indicator (16) when the inlet state is perturbed and the functional is measured on the outlet.

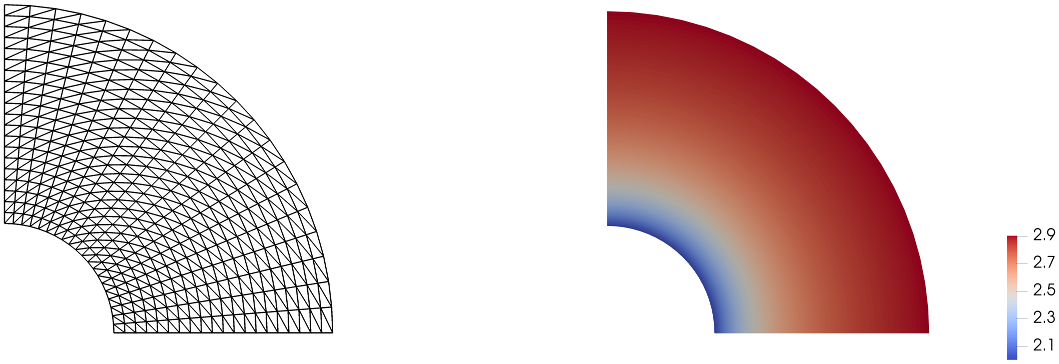
The steady isentropic vortex is defined on the quarter annulus computational domain shown in Figure 1a, with inner radius $r_{\text{in}} = 1$ and outer radius 3. The mesh has 20 quadrilaterals in each direction, and each quadrilateral is subdivided into 2 triangles. The analytical value of the density for the steady vortex is

$$\rho(r) = \rho_{\text{in}} \left[1 + \frac{\gamma - 1}{2} M_{\text{in}}^2 \left(1 - \frac{r_{\text{in}}^2}{r^2} \right) \right]^{\frac{1}{\gamma - 1}}.$$

The density at the inner radius is $\rho_{\text{in}} = 2$ and the Mach number at the inner radius is $M_{\text{in}} = 0.95$. The remaining variables can be computed from isentropic relations.

Figure 1b plots the density. The vortex rotates clockwise, with the vertical boundary as the inlet and the horizontal boundary as the outlet. At the inlet and outlet boundaries, the analytical solution is weakly imposed using the numerical flux function $\boldsymbol{f}^*$, and no penetration boundary conditions are applied along the inner and outer radii.

We consider a single design variable, the x-momentum added uniformly across the inlet region $y \in [2.5, 3.0]$. The added momentum, $\Delta\boldsymbol{x}$, is 0.45 which is an increase of approximately 64% compared to the initial momentum at the



(a) Mesh

(b) Initial density

Figure 1 Mesh and initial condition

specified section of the inlet. The functional is the mass flow rate per unit span over $x \in [2.5, 3.0]$ along the horizontal outlet boundary; that is

$$J(\boldsymbol{u}, \boldsymbol{x}) = \int_{2.5}^{3.0} \rho v \mathrm{d}x. \tag{24}$$

Initially, the streamlines are concentric circles around the origin. Consequently, the added momentum propagates through the domain and exits approximately through the section of the outlet where the functional is computed. We also note that, for this problem, J depends on the design variables only through the solution $\boldsymbol{u}$, thus $\frac{\partial J}{\partial \boldsymbol{x}}$ is zero.

The $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}} \Delta \boldsymbol{x}$ term and adjoint used to compute the reanalysis indicator (16) are shown in Figures 2a and 2b, respectively. The design variables affect the residual only through the inlet boundary condition in this problem, so $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}} \Delta \boldsymbol{x}$ is non-zero only for the elements along the inlet region. The adjoint shows the sensitivity of the functional to residual perturbations along the band $r \in [2.5, 3.0]$ from the outlet to the inlet. This is consistent with physical intuition that the streamlines form concentric circles, traversing from the inlet to the outlet.

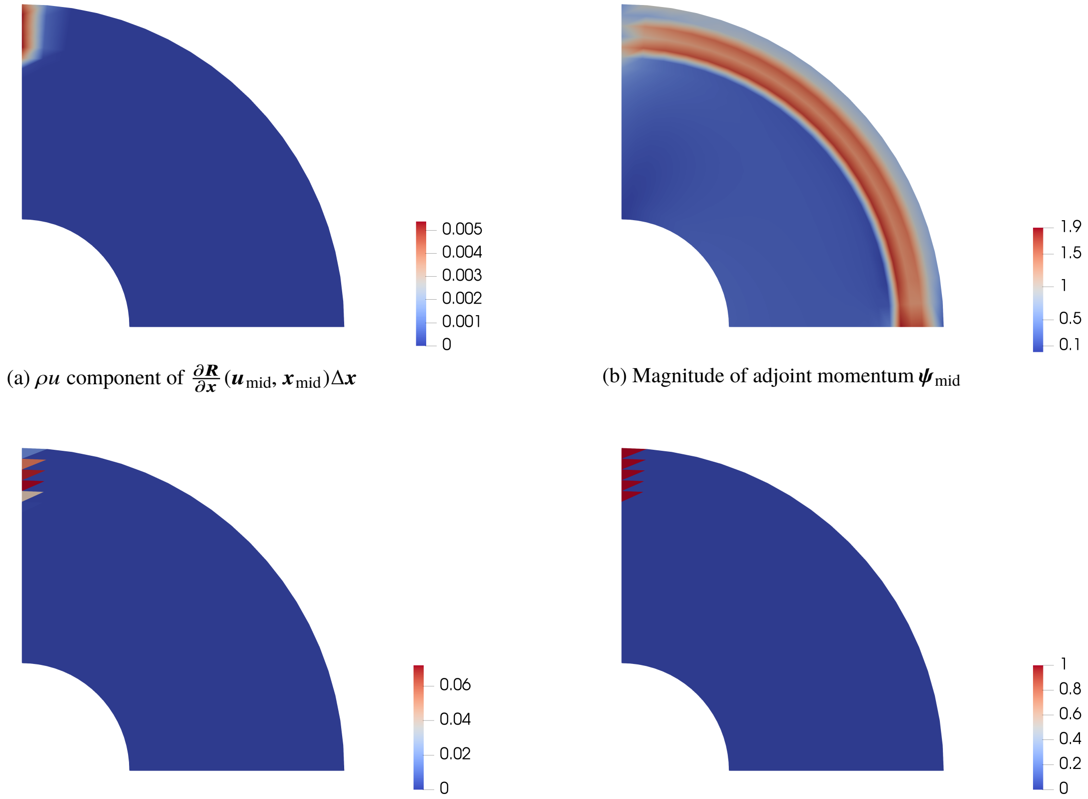
The indicator values (16) are shown in Figure 2c. Because $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}} \Delta \boldsymbol{x}$ is localized, only elements along the inlet have non-zero values. The expected benefit to re-solving is greater than $\tau_{\text{re-solve}}$, so a re-solve is performed. Because no elements along the outlet are selected, re-solving does not change the functional value, so the approximate solution $\boldsymbol{u} + \Delta \tilde{\boldsymbol{u}}$ is used to evaluate the functional.

The functional values and errors corresponding to the approximations introduced in Section II are shown in Table 1. The errors shown are relative to the functional value produced by a solve on the entire domain, $J(\boldsymbol{u}(\boldsymbol{x} + \Delta \boldsymbol{x}), \boldsymbol{x} + \Delta \boldsymbol{x})$. This full-solve functional value is considered the exact, or ideal value. The results show that reanalysis is able to estimate the functional to within 3.65% of the ideal value using the linear-state approximation $J(\boldsymbol{u} + \Delta \tilde{\boldsymbol{u}}, \boldsymbol{x} + \Delta \boldsymbol{x})$. It is possible to detect cases when no elements that are used in the computation of the functional are selected and avoid re-solving in this case. We leave the development of such heuristics for future work.

B. Airfoil

This test problem considers a case often encountered in aerodynamic shape optimization: recalculating a force coefficient on an airfoil after the geometry is modified. Here, the initial geometry is a NACA 0012 airfoil with chord length of one unit. The far-field boundary is a circle of diameter 20 chord lengths. The Mach number is 0.5 and the angle of attack is 2 degrees.

To manipulate the geometry, a free-form deformation (FFD) box [7] is defined around the airfoil and parameterized using B-splines. There are two rows of 5 control points each, equally spaced along the length of the box. The B-splines are cubic in the x direction and linear in the y direction. For this case, the design variables $\boldsymbol{x}$ are the coordinates of the FFD control points. To define the new geometry, the design variables are perturbed according to a uniform random



(a) ρu component of $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}(\boldsymbol{u}_{\text{mid}}, \boldsymbol{x}_{\text{mid}})\Delta \boldsymbol{x}$

(b) Magnitude of adjoint momentum $\boldsymbol{\psi}_{\text{mid}}$

(c) Indicator value

(d) Elements marked red are re-solved

Figure 2 Vortex Results

Table 1 Vortex results

	Value	% Error	# Elements	Refer to
$J(\boldsymbol{u}(\boldsymbol{x}), \boldsymbol{x})$	0.2863	-	800	(24)
$J(\boldsymbol{u}(\boldsymbol{x}+\Delta\boldsymbol{x}), \boldsymbol{x}+\Delta\boldsymbol{x})$	0.3770	0	800	(24)
$J(\boldsymbol{u}+\Delta\tilde{\boldsymbol{u}}, \boldsymbol{x}+\Delta\boldsymbol{x})$	0.3632	3.650	0	(20)
J estimate	0.3769	0.03133	0	(13)
$\Delta J_{\text{re-solve}}$	0.01364	-	-	(21)
$J(\boldsymbol{u}_{\text{new}}, \boldsymbol{x}+\Delta\boldsymbol{x})$	0.3632	3.650	5	

Table 2 Airfoil lift coefficient

	Value	% Error	# Elements	Refer to
Initial	0.2705	-	2,671	
$J(\boldsymbol{u}(\boldsymbol{x}+\Delta\boldsymbol{x}),\boldsymbol{x}+\Delta\boldsymbol{x})$	0.9485	0	2,671	
$J(\boldsymbol{u}+\Delta\tilde{\boldsymbol{u}},\boldsymbol{x}+\Delta\boldsymbol{x})$	0.9322	-1.726	0	(20)
J estimate	0.9469	-0.1703	0	(13)
$\Delta J_{\text{re-solve}}$	-0.01475	-	-	(21)
$J(\boldsymbol{u}_{\text{new}},\boldsymbol{x}+\Delta\boldsymbol{x})$	0.9531	0.4814	953	

distribution in the range $[0, 0.1]$.

In order to avoid generating invalid elements, the FFD box is used only to update the coordinates of the mesh vertices on the surface of the airfoil. The mesh volume coordinates are then updated with an inverse-distance weighted explicit mesh movement algorithm based on [8]. This algorithm has been successfully applied to several constrained optimization problems [9, 10]. The initial and warped meshes are shown in Figures 3a and 3b, respectively.

The x-momentum plots for the initial geometry, reanalysis, and the full solve are shown in Figures 3c, 3d, and 3e, respectively. The plots show the x-momentum is increased above the airfoil and slightly decreased below, and that this change is captured by reanalysis. Figure 3f shows the difference between the full-solve solution and the reanalysis solution.

Figures 4a and 4b show $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}\Delta\boldsymbol{x}$ and the adjoint $\boldsymbol{\psi}_{\text{mid}}$ used to compute the reanalysis indicator (16). The $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}\Delta\boldsymbol{x}$ term is largest near the surface, where the surface normal vectors change significantly from the original design to the new design. The adjoint is largest in magnitude near the surface of the airfoil as well, particularly near the trailing edge. Figure 4c shows the elements on the surface of the airfoil have large indicator values, and therefore a significant effect on the functional. Elements near the leading and trailing edges also have an appreciable impact on the functional.

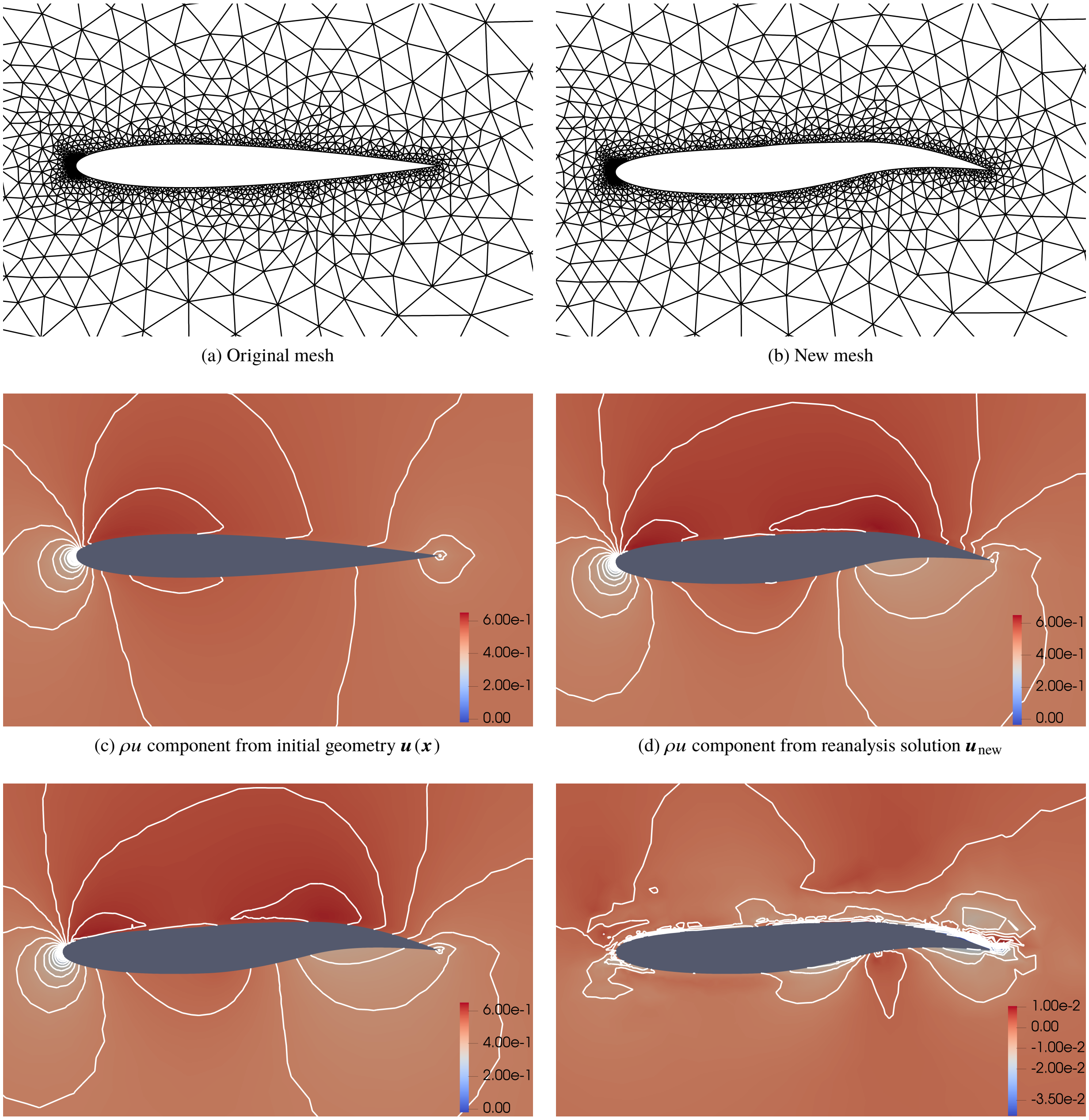
The elements selected for re-solving are shown in Figure 4d. Elements near the leading and trailing edges are selected, as well as those on the surface of the airfoil. In addition, a large group of elements above the downstream half of the airfoil are selected. This group of elements is disjoint from the elements connected to the surface, and hence have no effect on the functional value. It is possible to develop heuristics to remove disjoint sets of elements; however for this preliminary work we solve on the elements identified by the indicator.

The functional values and errors are listed in Table 2. The expected benefit to re-solving is greater than $\tau_{\text{re-solve}}$, so a re-solve is performed. As in the vortex case, errors are computed relative to the functional value produced by a solve on the entire domain. This full-solve value is considered the exact, or ideal value. The approximate functional value from the right-hand side of (20) is within 1.7% of the exact value, however, re-solving on the subset of 953 elements reduces the error from $J(\boldsymbol{u}+\Delta\tilde{\boldsymbol{u}},\boldsymbol{x}+\Delta\boldsymbol{x})$ by more than a factor of 3. We note that the functional value estimated by (13) is more accurate than the functional value computed from $\boldsymbol{u}_{\text{new}}$; however obtaining an approximate solution can provide valuable information during the design process, enabling both visualization and computation of derivatives.

V. Conclusions

We have presented a new adjoint-based method for local reanalysis of partial differential equations. This method uses an adjoint-based reanalysis indicator to identify regions where the corresponding output is sensitive to perturbations in the residual. The governing equations are re-solved only in regions of high sensitivity. This method considers both the relevance of the solution to the output functional and the full physics of the problem when re-solving, ignoring solution features that are unimportant. As a result, computational resources are expended in an efficient manner. In cases where new flow features develop at the perturbed design, for example boundary-layer separation, reanalysis has the potential to be significantly more accurate than linear approximations.

Several questions remain, the most pressing of which is the impact of the boundary condition on $\partial\Omega'_{\text{new}}$ In the airfoil case, the $\boldsymbol{u}+\Delta\tilde{\boldsymbol{u}}$ solution was sufficiently accurate, but for problems with stronger nonlinearities this may not be the case, leading to an inaccurate solution on the sub-domain. For nonlinear cases where the change in geometry is



(a) Original mesh

(b) New mesh

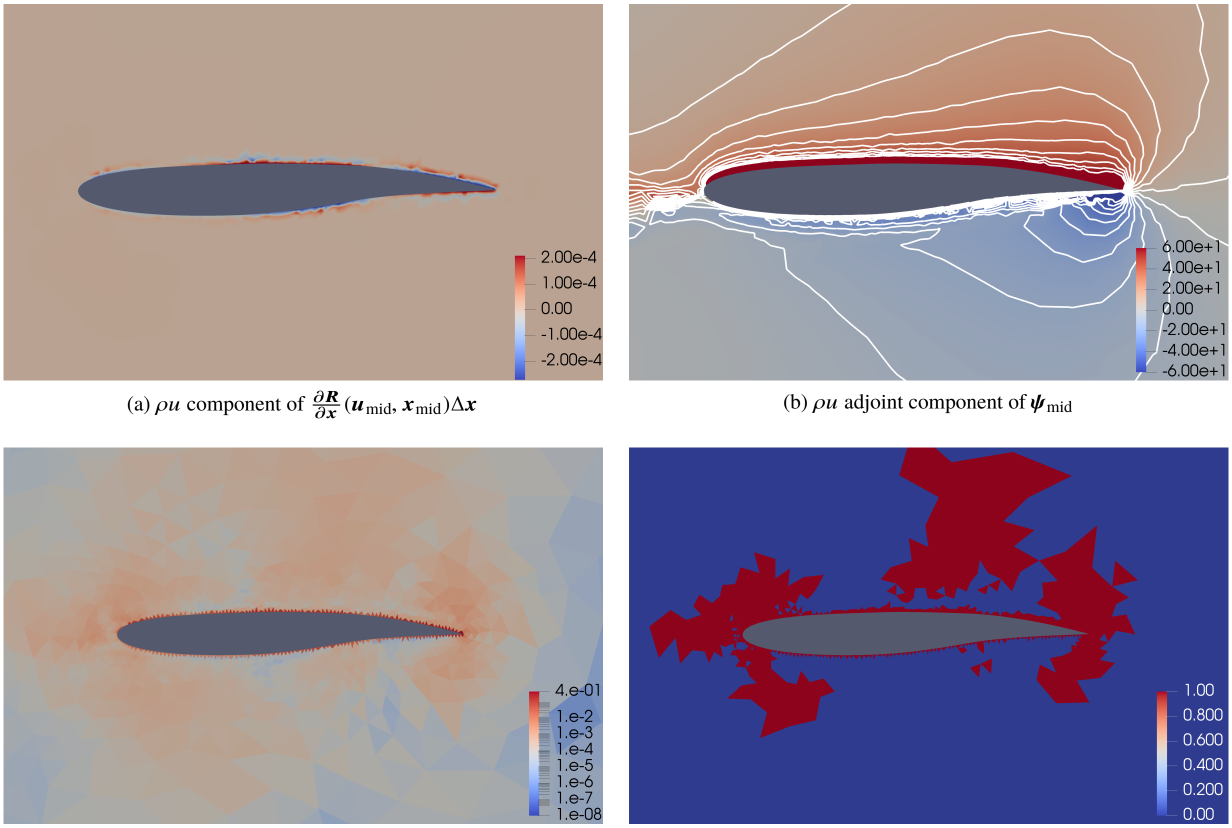
(c) ρu component from initial geometry $\boldsymbol{u}(\boldsymbol{x})$

(d) ρu component from reanalysis solution $\boldsymbol{u}_{\text{new}}$

(e) ρu component from full solve $\boldsymbol{u}(\boldsymbol{x} + \Delta\boldsymbol{x})$

(f) ρu component difference $\boldsymbol{u}(\boldsymbol{x} + \Delta\boldsymbol{x}) - \boldsymbol{u}_{\text{new}}$

Figure 3   Airfoil mesh and solutions



(a) ρu component of $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}(\boldsymbol{u}_{\text{mid}}, \boldsymbol{x}_{\text{mid}})\Delta \boldsymbol{x}$

(b) ρu adjoint component of $\boldsymbol{\psi}_{\text{mid}}$

(c) Indicator value

(d) Elements marked red are re-solved

Figure 4 Airfoil indicator

localized, the approximate solution several distance units away from the change may be sufficiently accurate. If the geometry change is global, then the boundary of the sub-domain may need to be so far away from the body that the number of elements in the sub-domain is not significantly smaller than in the original domain.

Additional challenges include parallel load balancing of the sub-domain solve and accuracy in the presence of discontinuities. Even if the original mesh is load balanced, the sub-domain mesh will not be well balanced, in general. Dynamic load balancing will be required, and the cost must be weighed against the cost of the nonlinear solve on the sub-domain. When solving on the sub-domain, shocks may arise that were not present in the the original solution. This could make the nonlinear solve on the sub-domain difficult to converge, and the accuracy of the resulting solution could be strongly dependent on the boundary condition applied to $\partial\Omega'_{\text{new}}$.

A. PDE Discretization

The governing equations in (22) are discretized using diagonal-norm Summation-by-Parts operators on simplex elements [11] using a weak form discontinuous-Galerkin formulation. Degree $p = 1$ SBP-Ω type elements are used, which contain degrees-of-freedom on the interior of the element but none on the faces [12]. In the $p = 1$ case, the SBP-DG scheme is equivalent to a collocation DG finite element method. The final discretization on each element κ is

$$\bar{\mathsf{Q}}_x^T \boldsymbol{f}_x(\boldsymbol{u}_\kappa) + \bar{\mathsf{Q}}_y^T \boldsymbol{f}_y(\boldsymbol{u}_\kappa) - \sum_{\gamma \subset \partial\Omega_\kappa} (\mathsf{R}_{\gamma\kappa}^T \mathsf{B}_\gamma \otimes \mathsf{I}_4) \boldsymbol{f}^\star(\mathsf{R}_{\gamma\kappa}\boldsymbol{u}_\kappa, \mathsf{R}_{\gamma\nu}\boldsymbol{u}_\nu) = 0, \tag{25}$$

where $\bar{\mathsf{Q}}_x = (\mathsf{H}_\kappa \mathsf{D}_x) \otimes \mathsf{I}_4$. The D_x matrix is the x direction differentiation operator for the element, and H_κ is the diagonal mass matrix. The Kronecker product is used to apply the operator to all 4 components of the Euler equations simultaneously, and the transposes on $\bar{\mathsf{Q}}_x$ and $\bar{\mathsf{Q}}_y$ are the result of applying integration-by-parts [11] [12]. $\boldsymbol{f}_x(\boldsymbol{u}_\kappa)$ is the concatenation of the vector of $\boldsymbol{F}_x$ evaluated at each node of the element. At each cubature node on the interface, the numerical flux function $\boldsymbol{f}^*$ is evaluated using the solution from elements κ and ν on either side of the interface. The operators $\mathsf{R}_{\gamma\kappa}$ and $\mathsf{R}_{\gamma\nu}$ interpolate the solution from the volume nodes to the face cubature points where $\boldsymbol{f}^*$ is evaluated. B_γ is a diagonal matrix, consisting of integration weights on the face γ, and $\mathsf{R}_{\gamma\kappa}^T$ applies a test function that weights each face cubature point against the volume nodes.

The numerical flux function $\boldsymbol{f}^*$ is the Roe flux in the face normal direction, and is used for both element interfaces and weakly imposing boundary conditions, except for no-penetration boundaries. For faces on a no-penetration boundary, the boundary flux $\boldsymbol{f}^*$ is replaced by

$$n_x \boldsymbol{f}_x(\mathsf{P}\mathsf{R}_{\gamma\kappa}\boldsymbol{u}_\kappa) + n_y \boldsymbol{f}_y(\mathsf{P}\mathsf{R}_{\gamma\kappa}\boldsymbol{u}_\kappa), \tag{26}$$

where n_x and n_y are the x and y components of the face-normal vector, and P is a matrix that projects the velocity onto the direction tangent to the face, which is important for achieving smooth adjoint fields [6].

Acknowledgments

J. Crean and J. Hicken were supported by the National Science Foundation under Grant No. 1554253. The authors gratefully acknowledge this support. We also thank RPI's Scientific Computation Research Center for the use of computer facilities.

References

[1] Fidkowski, K. J., and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 674–694.

[2] Fidkowski, K. J., and Roe, P. L., "An Entropy Adjoint Approach to Mesh Refinement," *SIAM Journal on Scientific Computation*, Vol. 32, No. 4, 2010, pp. 1261–1287.

[3] Lions, J.-L., *Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles*, Etudes Mathématiques, Dunod; Gauthier-Villars, Collier-Macmillan Ltd., 1968.

[4] Pironneau, O., "On optimum design in fluid mechanics," *Journal of Fluid Mechanics*, Vol. 64, No. 1, 1974, pp. 97–110.

[5] Jameson, A., “Aerodynamic design via control theory,” *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260. doi:10.1007/BF01061285, URL https://doi.org/10.1007/BF01061285.

[6] Lu, J., “An a posteriori Error Control Framework for Adaptive Precision Optimization using Discontinuous Galerkin Finite Element Method,” Ph.D. thesis, Massachusetts Institute of Technology, 2005.

[7] Sederberg, T. W., and Parry, S. R., “Free-form deformation of solid geometric models,” *SIGGRAPH ’86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 1986, pp. 151–160. doi:10.1145/15922.15903, URL http://dx.doi.org/10.1145/15922.15903.

[8] Luke, E., Collins, E., and Blades, E., “A fast mesh deformation method using explicit interpolation,” *Journal of Computational Physics*, Vol. 231, No. 2, 2012, pp. 586 – 601. doi:https://doi.org/10.1016/j.jcp.2011.09.021, URL http://www.sciencedirect.com/science/article/pii/S0021999111005535.

[9] Kenway, G. K., Mishra, A., Secco, N. R., Duraisamy, K., and Martins, J., “An Efficient Parallel Overset Method for Aerodynamic Shape Optimization,” *Proceedings of the 58th AIAA Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, 2017. doi:10.2514/6.2017-0357, URL https://doi.org/10.2514/6.2017-0357.

[10] He, P., Mader, C. A., Martins, J. R., and Maki, K. J., “An aerodynamic design optimization framework using a discrete adjoint approach with OpenFOAM,” *Computers and Fluids*, Vol. 168, 2018, pp. 285 – 303. doi:https://doi.org/10.1016/j.compfluid.2018.04.012, URL http://www.sciencedirect.com/science/article/pii/S0045793018302020.

[11] Hicken, J. E., Del Rey Fernández, D. C., and Zingg, D. W., “Multi-dimensional summation-by-parts Operators: General theory and application to simplex elements,” *SIAM Journal on Scientific Computing*, Vol. 38, No. 4, 2016, pp. A1935–A1958.

[12] Del Rey Fernández, D. C., Hicken, J. E., and Zingg, D. W., “Simultaneous Approximation Terms for Multi-dimensional Summation-by-Parts Operators,” *Journal of Scientific Computing*, Vol. 75, No. 1, 2018, pp. 83–110. doi:10.1007/s10915-017-0523-7, URL https://doi.org/10.1007/s10915-017-0523-7.