Towards Output-Based Mesh Adaptation for High-Order Fluid-Structure Interaction of Flexible Wings

Braden E. Frigoletto*, Krzysztof J. Fidkowski†, and Carlos E. S. Cesnik‡
University of Michigan, Ann Arbor, Michigan, 48109

This paper outlines an approach for output-based mesh adaptation in the simulation of high-order, static fluid-structure interaction of very flexible (high aspect-ratio) wings. A partitioned approach to the fluid-structure system is proposed, wherein the fluid domain is modeled using a high-order discontinuous Galerkin formulation of the Navier-Stokes equations, while the structural domain is modeled with a high-order continuous Galerkin finite element method. Radial basis function interpolation is used to capture the fluid mesh motion due to the structural coupling, and the consequences of its implementation within an Arbitrary Lagrangian-Eulerian formulation are discussed. Fluid-structure coupling is accomplished via a weighted least-squares extrapolation transfer technique for loads and displacements. A coupled adjoint formulation is derived and is used to develop an adjoint-weighted residual technique for output-based error estimation. Implementation details of the coupled adjoint are discussed in depth, as is the error localization technique used for the structural domain. A high aspect-ratio wing model in a cruise flight condition is used to verify the uncoupled, single-discipline output-based error estimation and mesh adaptation methods for the fluid and structural components of the present computational framework.

I. Introduction

In the typical design process for a complex multidisciplinary system like an aircraft, often the complexity of the analysis increases with time as the design evolves and aspects of it are fixed. Simplified, lower-fidelity analyses typically precede those that are higher fidelity. At higher fidelity, multidisciplinary considerations come with significant computational challenges. Additionally, the cost associated with correcting a perceived defect in a design also increases with time, where corrections are cheaper to make when design freedom is still plentiful and aspects of the design have yet to be fixed or frozen. Importantly, for complex multidisciplinary systems, a high level of fidelity can be required to accurately resolve complex coupled physics that are paramount to the performance or health of the design. Not using high fidelity early in the design process bears a significant risk of late discovery of defects, which increases expense or compromises the performance of the final design. The engine aft deck panel on the Northrup-Grumman B-2 is a prime example of this design issue, where an under-resolved analysis of the structural, thermal, and aeroacoustic effects led to a design that experienced crack formation and fatigue failure in orders of magnitude fewer flight hours than initially predicted and necessitated a costly redesign and refurbishment effort [1]. One potential remedy for this issue would be to conduct higher-fidelity, coupled analyses earlier on in the design process to accurately resolve the complex physics and their mutual interaction while the design can still be changed and major cost or performance ramifications can be avoided. Currently, these types of coupled high-fidelity analyses are slow to progress due to a combination of their high computational cost as well as the pre-processing steps necessary for analysis, *e.g.* geometry and mesh generation. Output-based error estimation and mesh adaptation, combined with high-order numerical methods, can alleviate some of these drawbacks in high-fidelity analysis. A design approach combining these two techniques could allow a designer or engineer to start with a relatively coarse discretization of their model and attain a verified output from the analysis in an automated manner that is problem agnostic and more computationally efficient compared to either a uniform refinement or a manual refinement process with lower-order methods as is typical in today's aircraft design environment [2].

High-order numerical methods, for both spatial and temporal discretizations, have received much attention in application to aircraft design [3–7], as they have been shown to accurately resolve solution characteristics for certain classes of problems with far fewer degrees of freedom when compared to low-order methods. However, high-order methods can have high memory requirements, experience poor system conditioning and robustness issues, and may be

*PhD Candidate, Aerospace Engineering

†Professor, Department of Aerospace Engineering, AIAA Associate Fellow

‡Richard A. Auhll Department Chair, Clarence L. "Kelly" Johnson Collegiate Professor of Aerospace Engineering, Fellow, AIAA

slow or difficult to converge for problems with non-smooth features and strong discontinuities. These issues are in part responsible for limiting their widespread adoption within the aircraft design community. Error estimation and adaptive mesh refinement techniques have been shown to be an attractive supplement to high-order methods which address some of these concerns. Error estimation techniques quantify the discretization error in a model, which can be used to define adaptive indicators that signal where to coarsen or refine a mesh locally to more efficiently resolve solution characteristics and evaluate outputs of interest, which can be used for design purposes [2, 8–10]. This is demonstrated well in the first international workshop for high-order methods in computational fluid dynamics [5], where Yano and Darmofal and Burgess and Mavriplis submitted results for several test cases which feature output-based adapted meshes which routinely outperform the low-order analyses as well as other high-order analyses using uniform refinement. In the context of coupled fluid-structure interaction, there have been various developments to enable mesh adaptation. Bathe and Zhang [11] developed an adjoint-free approach to fluid-only h-adaptation for fluid-structure interaction using the flow-condition-based interpolation fluid discretization method, applying it to various problems including nonlinear structural effects like large deformations, nonlinear material behavior, and contact. Borker *et al.* [12] developed a fluid-only h-adaptive technique with a Hessian-based criterion for resolving various flow features for embedded-boundary methods, primarily focused on viscous fluid-structure interaction problems. These studies use low-order discretization methods and do not consider simultaneous adaptation for both fluid and structural domains. They also do not construct error estimates, which are helpful within a design context. Preliminary work by Ojha *et al.* [13–15], has addressed some of these aspects, where a high-order discontinuous Galerkin discretization is used to model the fluid domain, as well as fully-coupled, output-based error estimates and simultaneous mesh adaption for both fluid and structural domains. These studies however are applied to small-scale problems and regimes that are not typical of modern aircraft design. Additionally, other works like Kenway *et al.* [16] have focused on scalable coupled fluid-structure adjoint implementations. However, in these works, the coupled adjoint is used primarily for design optimization applications using lower-order methods and does not incorporate error estimation for analysis. The objective of this paper is to present an approach for output-based error estimation and mesh adaptation for high-order fluid-structure interaction, with applications in aircraft design. The intent is to formulate both the primal and adjoint solutions, as well as discuss various practical implementation details necessary to consider when working with large-scale problems of interest for the aircraft design community.

The outline of the remainder of the paper is as follows. Section II describes the formulation and implementation of the coupled fluid-structure primal problem. Section III outlines the coupled adjoint formulation and discusses implementation details for the partial derivative evaluations. Section IV describes the adjoint-based error estimation technique and adaptive refinement strategy, which are employed independently in each domain. Section V introduces a high-aspect ratio wing model and verifies uncoupled, single-discipline adaptation, which motivates future work towards coupled adaptation on large-scale problems. Finally, Section VI summarizes the contents of the paper and discusses future work.

II. Fluid-Structure Primal Formulation

The present work builds off of the developments made in [17], wherein a detailed discussion can be found on the fluid and structural system governing equations and discretization methods, the fluid-structure coupling procedure for the primal problem, and their computational implementation. A succinct overview of some of these topics will be repeated here for convenience, as well as some new developments. For the remainder of this discussion, the subscript $_f$ denotes a term corresponding to the fluid system, whereas the subscript $_s$ denotes a term corresponding to the structural system.

A. Fluid System

The Navier-Stokes equations describe the governing physics of fluid dynamics and can be augmented by an Arbitrary Lagrangian-Eulerian (ALE) formulation [18] in order to account for moving body-fitted boundaries due to coupling with the structural domain. The governing equations are discretized using a high-order discontinuous Galerkin (DG) finite element method [19]. The discrete nonlinear fluid equations can be written in a residual form as follows

$$\mathbf{R}_f(\mathbf{u}_f) = \mathbf{0}, \tag{1}$$

where $\mathbf{R}_f$ is the discrete fluid residual vector and $\mathbf{u}_f$ is the vector of fluid states. The n_f nonlinear Euler equations are presently implemented and verified in a CFD package called xflow [20]. Viscous solutions are supported but are not used presently to ease the meshing process by avoiding the need for curved anisotropic meshes for the model presented in Section V. For steady solutions, the discrete fluid residual is minimized using Newton's method, where the fluid update, $\Delta\mathbf{u}_f$, is obtained by solving a linear system containing the fluid residual Jacobian matrix, $\frac{\partial\mathbf{R}_f}{\partial\mathbf{u}_f}$, using a preconditioned GMRES iterative linear solver. Once the fluid states have been determined, the fluid pressure and shear loads on the fluid-structure interface can be determined by numerically integrating the inviscid and viscous momentum fluxes on the appropriate boundaries. The resulting fluid load vector, $\vec{\mathbf{F}}_f$, is then transferred to the structural domain for fluid-structure coupling.

B. Structural System

The equilibrium, kinematic, and constitutive equations that describe the governing equations of elasticity are implemented in the Toolkit for the Analysis of Composite Structures (TACS) finite-element package [21]. TACS is an open source, adjoint-enabled, parallel finite-element analysis and design optimization framework. TACS supports the analysis of thin-walled structures, including the effects of geometric nonlinearity, through various implementations of MITC shell elements [22]. TACS is also combined with TMR, a mesh generation and refinement tool for quadrilateral and hexahedral elements [23]. The structural equations are discretized using a continuous Galerkin finite-element method, where the discrete structural equations can be written in a residual form as follows,

$$\mathbf{R}_s(\mathbf{u}_s) = \mathbf{0}, \tag{2}$$

where $\mathbf{R}_s$ is the discrete structural residual vector and $\mathbf{u}_s$ is the vector of structural states. A linear elastic solution is assumed presently, where the n_s structural equations are solved using a preconditioned GMRES iterative linear solver. Once the structural states have been determined, the displacement components of the structural state, $\vec{\mathbf{d}}_s$, are extracted from the full state vector and can be transferred to the fluid domain to complete the fluid-structure coupling procedure. For finite element discretizations which feature rotational degrees of freedom, like beam- or shell-type elements, these degrees of freedom are neglected from the coupling as they are incompatible with the fluid domain boundary definition.

C. Mesh Motion

Due to the coupling with the structural system, not all boundaries in the fluid domain are stationary. The motion of the boundaries arising from the interaction with the structural system needs to be propagated throughout the rest of the fluid domain in a robust and smooth manner to preserve the high-order fluid solution over potentially large deformations. In the ALE approach, the fluid system states and fluxes are mapped analytically between a deforming *physical* space (Lagrangian) and a fixed *reference*[*] space (Eulerian). Edge and interior element contributions to the fluid residual are evaluated using numerical integration with quadrature, evaluated at sets of quadrature points. The analytical one-to-one mapping between the physical and reference space is therefore computed for every quadrature point in the mesh, allowing for the states and fluxes to be mapped for the necessary integrations. The form of this mapping can be provided in different ways; presently it is provided by radial basis function (RBF) interpolation, which can smoothly interpolate the motion from the moving boundary into the interior of the fluid domain and has been used frequently in FSI applications [24]. RBFs interpolate a given field f with the following form,

$$f(\vec{X}) = \sum_{i=1}^{N_c} \vec{\gamma}_i \phi(\|\vec{X} - \vec{X}_{c_i}\|) + \vec{p}(\vec{X}), \tag{3}$$

where $\vec{X}$ is a point spatial coordinate of dimension d, N_c is the number of RBF centers (points where a field quantity is known), $\vec{\gamma}$ is a set of interpolation coefficients, ϕ is a given basis function with respect to the Euclidean distance $\|\cdot\|$, and $\vec{p}(\vec{X})$ is a set of linear polynomials over the spatial dimensions of $\vec{X}$. For the application of mesh motion, the linear polynomial terms, if included, would preserve rigid body translation and rotation, which is not desired and therefore they are neglected. There are many different forms the RBF can take; in the current implementation Wendland's C^2 function with local support is used, which has been used for FSI applications in the past and has been shown to reduce the cost of the interpolation while maintaining accuracy [24]. It has the following form,

$$\phi(R_i) = \begin{cases} 0, & \text{if } R_i > 1 \\ (4R_i + 1)(1 - R_i)^4 & \text{if } R_i \leq 1, \end{cases} \tag{4}$$

[*] Note that this reference space is different from the master element "reference" space.

where $R_i = \|\vec{X} - \vec{X}_{c_i}\|/r_i$, and r_i is the local support radius for the i^{th} RBF center point. Given a set of known fluid surface displacements, $\vec{\mathbf{d}}_f \in \mathbb{R}^{N_c \times d}$, the interpolation coefficients can be determined through the solution of the following linear system

$$\mathbf{\Phi}^{\text{RBF}}\vec{\boldsymbol{\gamma}} = \vec{\mathbf{d}}_f, \tag{5}$$

where $\mathbf{\Phi}^{\text{RBF}} \in \mathbb{R}^{N_c \times N_c}$ is a matrix containing the evaluation of the RBF between each center point on the deforming interface boundaries, and $\vec{\boldsymbol{\gamma}} \in \mathbb{R}^{N_c \times d}$ is a matrix of interpolation coefficients for each center point corresponding to each spatial dimension. In practice, the traditional RBF method is accurate but can become expensive for high-fidelity CFD applications due to the large number of mesh points on a surface, oftentimes leading to a subsampling of the deforming boundaries to offset the cost. Therefore, a more efficient multiscale RBF implementation [25] has been presently implemented, which reduces the computational expense without the need for subsampling the deforming boundaries in the domain. In the multiscale approach, a small subset of the RBF center points, called the base set N_b, use a large support radius, while the rest, called the reduced set N_r, use successively smaller support radii. The effect of the decreasing support radius size leads to a block lower triangular matrix, which can more easily be solved using a forward substitution approach compared to the dense symmetric system in the typical implementation. This block lower triangular system can be seen as

$$\mathbf{\Phi}^{\text{RBF}} = \begin{bmatrix} \mathbf{\Phi}_b & \mathbf{0} \\ \mathbf{\Phi}_r & \mathbf{L}_r \end{bmatrix}, \tag{6}$$

where $\mathbf{\Phi}_b \in \mathbb{R}^{N_b \times N_b}$ is a symmetric matrix containing RBF evaluations between all base set center points, $\mathbf{\Phi}_r \in \mathbb{R}^{N_r \times N_b}$ is a rectangular matrix containing RBF evaluations of the base-set functions at the refined-set centers, and $\mathbf{L}_r \in \mathbb{R}^{N_r \times N_r}$ is a lower-triangular matrix containing the RBF evaluations of all refined-set basis functions at the refined-set centers. Due to the local support of the radial function being used, incrementally reducing the support radius size leads to this lower triangular shape, which is exploited to reduce the computational expense and obviate the need for subsampling.

The ALE formulation requires the evaluation of not only the mesh deformation, but also the mesh velocity. The mesh velocity term augments the fluxes in the mapping between the physical and reference spaces. For steady applications, it is zero and therefore does not participate during the mapping process. When the velocity term is not included, it is equivalent to simply deforming the mesh coordinates directly using the interpolated deformation field from the RBF method and solving the fluid system assuming this new mesh configuration. Therefore, the ALE formulation is only necessary for time-accurate unsteady applications with deforming domains. This is an important distinction to make because the presence of the analytical mapping can influence some of the partial derivative terms that are necessary to evaluate for the coupled adjoint. This point is explained further in Section III. Since only steady applications are presently being pursued, a direct mesh interpolation using the multiscale RBF method described above is used. The residual form of the mesh motion linear system can be written as

$$\mathbf{H}(\vec{\mathbf{d}}_f, \vec{\boldsymbol{\gamma}}) = \mathbf{0}. \tag{7}$$

For each coupled iteration of the primal solution, a set of fluid interface displacements, $\vec{\mathbf{d}}_f$, is provided, from which the interpolation coefficients, $\vec{\boldsymbol{\gamma}}$, are obtained, and the mesh coordinates update, $\Delta\vec{\mathbf{x}}_f$, is determined using the interpolated displacement field,

$$\Delta\vec{\mathbf{x}}_f = \mathbf{\Psi}^{\text{RBF}}\vec{\boldsymbol{\gamma}}, \tag{8}$$

where $\mathbf{\Psi}^{\text{RBF}} \in \mathbb{R}^{N_f \times N_c}$ is a matrix of all RBF functions evaluated at all mesh nodes in the fluid domain.

D. Fluid-Structure Coupling

Within a partitioned approach to high-fidelity FSI, it is common for the computational meshes of the fluid and structural domains to not match on the fluid-structure interface. This is particularly true if h-adaptive mesh refinement is applied to the coupled problem, where a mesh's spacing is being refined. The transferring of loads from the fluid domain to the structural domain, and likewise, the transferring of displacements from the structural domain to the fluid domain, is a necessary and crucial component in the coupled solution. The loads and displacements must be transferred in a conservative and consistent manner. In the present implementation, this process is achieved through the use of the Matching-based Extrapolation of Loads and Displacements (MELD) method [26], implemented in the open-source toolkit for aerostructural analysis and optimization FUNtoFEM [27]. The method requires initializing the surface boundaries in both the fluid and structural meshes, associating each fluid surface RBF center node, $\vec{\mathbf{x}}_{f_c}$, to a specified number of nearest structural surface nodes, $\vec{\mathbf{x}}_s$. The vector of fluid surface mesh displacements, $\vec{\mathbf{d}}_f$, is solved in a weighted least-squares sense preserving rigid-body translation and rotation, where the transfer operation can be written in a residual form as

$$\mathbf{D}(\vec{\mathbf{x}}_s, \vec{\mathbf{x}}_{f_c}, \vec{\mathbf{d}}_s, \vec{\mathbf{d}}_f) = \mathbf{0}. \tag{9}$$

The fluid loads, which are evaluated at the quadrature points of the fluid surface boundaries, $\vec{\mathbf{x}}_{f_q}$, are extrapolated to the structural mesh via a method derived from the principle of virtual work. This procedure is both consistent and conservative and can be written in the following residual form

$$\mathbf{L}(\vec{\mathbf{x}}_s, \vec{\mathbf{x}}_{f_q}, \vec{\mathbf{d}}_s, \vec{\mathbf{F}}_s, \vec{\mathbf{F}}_f) = \mathbf{0}. \tag{10}$$

Note that the MELD implementation within FUNtoFEM features analytical derivatives of the **D** and **L** operators with respect to their various input parameters.

E. Primal Solution Procedure

The combined discrete multidisciplinary system can be written in residual form as

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_f(\mathbf{u}_f; \mathbf{u}_s) \\ \mathbf{R}_s(\mathbf{u}_s; \mathbf{u}_f) \end{bmatrix} = \mathbf{0}, \tag{11}$$

where we seek to find a solution for the fluid and structural states to satisfy this set of coupled equations. The classical method typically employed in the numerical solution of FSI is the nonlinear block Gauss-Seidel fixed-point iterative approach. An iteration of this method begins with a solution on the fluid domain and processing of the loads on the fluid-structure interface. The loads are transferred to the structural domain, where they are then used to solve for the structural states. The structural displacements are transferred back to the fluid surface of the interface, where they are propagated into the volume of the fluid domain with a mesh motion technique. The process then repeats until a coupled convergence criterion is met.

In this approach there are three different sets of tolerances that are specified: ε_f is the tolerance used in the fluid solution procedure, ε_s is the tolerance used in the structural solution procedure, and lastly ε_{FSI}, which is used to check the coupled convergence of the overall system. In the present implementation, the coupled convergence criterion is based on the L_2-norm of the structural update, though coupled convergence can be measured in other ways. Oftentimes, this fixed-point iteration is accelerated using a technique such as Aitken acceleration, which under-relaxes the update that is taken at each iteration in a manner that increases the rate of convergence. For a linear analysis, this scaling factor can be applied to either the fluid loads or the structural displacements, and the result is identical. However, given that the structural analysis could be nonlinear, it is natural to apply the under-relaxation to the fluid loads that are passed to the structural domain. The iterative solution process is described in Algorithm 1.

III. Fluid-Structure Adjoint Formulation

A. Coupled Adjoint

A common goal of computational FSI is to evaluate some kind of scalar output functional, J, which is dependent on the states of both domains

$$J = J(\mathbf{u}_f, \mathbf{u}_s). \tag{12}$$

Given this form, the adjoint fields, $\boldsymbol{\Psi}_f$ and $\boldsymbol{\Psi}_s$ for the fluid and structural domains respectively, are interpreted as the sensitivities of the output functional with respect to infinitesimal perturbations in the fluid and structural residuals, $\delta\mathbf{R}_f$ and $\delta\mathbf{R}_s$ respectively,

$$\delta J = J(\mathbf{u}_f + \delta\mathbf{u}_f, \mathbf{u}_s + \delta\mathbf{u}_s) - J(\mathbf{u}_f, \mathbf{u}_s) = \begin{bmatrix} \boldsymbol{\Psi}_f^T & \boldsymbol{\Psi}_s^T \end{bmatrix} \begin{Bmatrix} \delta\mathbf{R}_f \\ \delta\mathbf{R}_s \end{Bmatrix}. \tag{13}$$

In this equation, the corresponding infinitesimal changes in the fluid and structural states, $\delta\mathbf{u}_f$ and $\delta\mathbf{u}_s$ respectively, enforce the satisfaction of the fluid and structural residual equations in response to the residual perturbations,

$$\begin{Bmatrix} \mathbf{R}_f(\mathbf{u}_f + \delta\mathbf{u}_f; \mathbf{u}_s + \delta\mathbf{u}_s) \\ \mathbf{R}_s(\mathbf{u}_s + \delta\mathbf{u}_s; \mathbf{u}_f + \delta\mathbf{u}_f) \end{Bmatrix} - \begin{Bmatrix} \mathbf{R}_f(\mathbf{u}_f; \mathbf{u}_s) \\ \mathbf{R}_s(\mathbf{u}_s; \mathbf{u}_f) \end{Bmatrix} + \begin{Bmatrix} \delta\mathbf{R}_f \\ \delta\mathbf{R}_s \end{Bmatrix} = \mathbf{0}. \tag{14}$$

Algorithm 1 Nonlinear Block Gauss-Seidel primal solution process

```
Require: $\vec{\mathbf{x}}_s, \vec{\mathbf{x}}_{f_c}, \vec{\mathbf{x}}_{f_q}, k_{\max}, \omega, \omega_{\min}$
  $\vec{\mathbf{u}}_s^{(0)} \leftarrow \mathbf{0}, \vec{\mathbf{F}}_f^{(0)} \leftarrow \mathbf{0}, \Delta\vec{\mathbf{F}}_f^{(0)} \leftarrow \mathbf{0}$    ▷ Initialize transfer and update quantities to 0
  for $k = 1$ to $k_{\max}$ do
    $\vec{\mathbf{F}}_f^{(k)} \leftarrow \mathbf{R}_f(\mathbf{u}_f^{(k)}) \leq \varepsilon_f$    ▷ Solve the fluids and determine surface boundary loads
    $\Delta\vec{\mathbf{F}}_f^{(k)} = \vec{\mathbf{F}}_f^{(k)} - \vec{\mathbf{F}}_f^{(k-1)}$    ▷ Evaluate the fluid update
    $\omega = \omega\left(1 - \frac{\left(\Delta\vec{\mathbf{F}}_f^{(k)} - \Delta\vec{\mathbf{F}}_f^{(k-1)}\right)\cdot\Delta\vec{\mathbf{F}}_f^{(k)}}{\left\|\Delta\vec{\mathbf{F}}_f^{(k)} - \Delta\vec{\mathbf{F}}_f^{(k-1)}\right\|^2}\right)$    ▷ Apply Aitken acceleration update formula
    $\omega = \max(\min(\omega, 1), \omega_{\min})$    ▷ Bound the relaxation factor
    $\vec{\mathbf{F}}_f^{(k)} = \vec{\mathbf{F}}_f^{(k-1)} + \omega\Delta\vec{\mathbf{F}}_f^{(k)}$    ▷ Increment and under-relax the fluid surface loads
    $\vec{\mathbf{F}}_s^{(k)} \leftarrow \mathbf{L}(\vec{\mathbf{x}}_s, \vec{\mathbf{x}}_{f_q}, \vec{\mathbf{d}}_s^{(k-1)}, \vec{\mathbf{F}}_s^{(k)}, \vec{\mathbf{F}}_f^{(k)}) = \mathbf{0}$    ▷ Transfer the fluid surface loads to the structural surface
    $\vec{\mathbf{u}}_s^{(k)} \leftarrow \mathbf{R}_s(\vec{\mathbf{u}}_s^{(k)}) \leq \varepsilon_s$    ▷ Solve for the structural displacements
    $\Delta\vec{\mathbf{u}}_s^{(k)} = \vec{\mathbf{u}}_s^{(k)} - \vec{\mathbf{u}}_s^{(k-1)}$    ▷ Evaluate the structural update
    if $\|\Delta\vec{\mathbf{u}}_s^{(k)}\| \leq \varepsilon_{\mathrm{FSI}}$ then
      break    ▷ Check for coupled convergence
    end if
    $\vec{\mathbf{d}}_f^{(k)} \leftarrow \mathbf{D}(\vec{\mathbf{x}}_s, \vec{\mathbf{x}}_{f_c}, \vec{\mathbf{d}}_s^{(k)}, \vec{\mathbf{d}}_f^{(k)}) = \mathbf{0}$    ▷ Transfer structural displacements to fluid surface boundary
    $\vec{\boldsymbol{\gamma}}^{(k)} \leftarrow \mathbf{H}(\vec{\mathbf{d}}_f^{(k)}, \vec{\boldsymbol{\gamma}}^{(k)}) = \mathbf{0}$    ▷ Solve for mesh motion interpolation coefficients
    $\vec{\mathbf{x}}_f \leftarrow \mathbf{\Psi}^{\mathrm{RBF}}\vec{\boldsymbol{\gamma}}^{(k)}$    ▷ Update the fluid mesh coordinates
  end for
```

Note that the middle term is zero, and it is included to show that if a linearization of the coupled equations is made, the following statement holds

$$\begin{bmatrix} \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f} & \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{u}_s} \\ \dfrac{\partial \mathbf{R}_s}{\partial \mathbf{u}_f} & \dfrac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s} \end{bmatrix} \begin{Bmatrix} \delta\mathbf{u}_f \\ \delta\mathbf{u}_s \end{Bmatrix} + \begin{Bmatrix} \delta\mathbf{R}_f \\ \delta\mathbf{R}_s \end{Bmatrix} = \mathbf{0}, \tag{15}$$

which requires that the discrete coupled residual equations are differentiable. Additionally, by assuming the output functional is also differentiable, substituting Eq. 15 into Eq. 13 yields the following relationship

$$\delta J = \begin{bmatrix} \dfrac{\partial J}{\partial \mathbf{u}_f} & \dfrac{\partial J}{\partial \mathbf{u}_s} \end{bmatrix} \begin{Bmatrix} \delta\mathbf{u}_f \\ \delta\mathbf{u}_s \end{Bmatrix} = -\begin{bmatrix} \mathbf{\Psi}_f^T & \mathbf{\Psi}_s^T \end{bmatrix} \begin{bmatrix} \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f} & \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{u}_s} \\ \dfrac{\partial \mathbf{R}_s}{\partial \mathbf{u}_f} & \dfrac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s} \end{bmatrix} \begin{Bmatrix} \delta\mathbf{u}_f \\ \delta\mathbf{u}_s \end{Bmatrix}. \tag{16}$$

Given that these statements should hold for arbitrary state perturbations $\delta\mathbf{u}_f$ and $\delta\mathbf{u}_s$

$$\begin{bmatrix} \dfrac{\partial J}{\partial \mathbf{u}_f} & \dfrac{\partial J}{\partial \mathbf{u}_s} \end{bmatrix} = -\begin{bmatrix} \mathbf{\Psi}_f^T & \mathbf{\Psi}_s^T \end{bmatrix} \begin{bmatrix} \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f} & \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{u}_s} \\ \dfrac{\partial \mathbf{R}_s}{\partial \mathbf{u}_f} & \dfrac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s} \end{bmatrix}. \tag{17}$$

Rearranging yields the discrete coupled adjoint equations

$$\begin{bmatrix} \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f}^T & \dfrac{\partial \mathbf{R}_s}{\partial \mathbf{u}_f}^T \\ \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{u}_s}^T & \dfrac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s}^T \end{bmatrix} \begin{Bmatrix} \mathbf{\Psi}_f \\ \mathbf{\Psi}_s \end{Bmatrix} = -\begin{Bmatrix} \dfrac{\partial J}{\partial \mathbf{u}_f}^T \\ \dfrac{\partial J}{\partial \mathbf{u}_s}^T \end{Bmatrix}. \tag{18}$$

B. Partial Derivative Evaluation

1. Fluid System Derivatives

The partial derivative of the fluid residual with respect to the fluid states, $\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f}$, is evaluated analytically within xflow for the Navier-Stokes equations and is already employed elsewhere in the solution process, such as in the Newton solver used for computing the fluid state update for each Newton iteration. The other derivative associated with the fluid domain is the partial derivative of the fluid residual with respect to the structural states, $\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_s}$. This partial derivative can be expressed through the chain rule of differentiation of various quantities such as

$$\left(\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_s}\right)^T \mathbf{\Psi}_f = \left(\frac{\partial \vec{\mathbf{d}}_s}{\partial \mathbf{u}_s}\right)^T \left(\frac{\partial \vec{\mathbf{d}}_f}{\partial \vec{\mathbf{d}}_s}\right)^T \left(\frac{\partial \vec{\mathbf{x}}_f}{\partial \vec{\mathbf{d}}_f}\right)^T \left(\frac{\partial \mathbf{R}_f}{\partial \vec{\mathbf{x}}_f}\right)^T \mathbf{\Psi}_f. \tag{19}$$

Working from left to right, the term $\frac{\partial \vec{\mathbf{d}}_s}{\partial \mathbf{u}_s}$ is simply a diagonal matrix with diagonal terms set to unity if that degree of freedom corresponds to a translational degree of freedom, or zero if that degree of freedom corresponds to a rotational degree of freedom. This matrix is not explicitly formed, but the action of it is applied to the structural state. It acts as the enforcement of the compatibility requirement between the structural states and the fluid mesh, as mentioned above. The term $\frac{\partial \vec{\mathbf{d}}_f}{\partial \vec{\mathbf{d}}_s}$ represents the partial derivative of the displacement transfer residuals, **D**, with respect to the structural displacements and is provided analytically in a matrix-free fashion by the MELD implementation. The term $\frac{\partial \vec{\mathbf{x}}_f}{\partial \vec{\mathbf{d}}_f}$ represents the partial derivative of the fluid mesh coordinates with respect to the fluid domain surface displacements. This term can be derived analytically using the definition of the RBF interpolation. The fluid mesh nodal coordinates are provided by the interpolated mesh displacement field, where Eq. 5 is substituted into Eq. 8 such that

$$\begin{aligned} \vec{\mathbf{x}}_f &= \vec{\mathbf{x}}_f + \Delta \vec{\mathbf{x}}_f \\ &= \vec{\mathbf{x}}_f + \mathbf{\Psi}^{\text{RBF}} \left(\mathbf{\Phi}^{\text{RBF}}\right)^{-1} \vec{\mathbf{d}}_f, \end{aligned} \tag{20}$$

from which it becomes clear that the partial derivative term is

$$\frac{\partial \vec{\mathbf{x}}_f}{\partial \vec{\mathbf{d}}_f} = \mathbf{\Psi}^{\text{RBF}} \left(\mathbf{\Phi}^{\text{RBF}}\right)^{-1}. \tag{21}$$

This matrix-matrix product is not explicitly formed in the current implementation. Instead, the action of these two matrices is applied in series to an input vector, where the matrix-vector products are distributed among multiple processors for a parallel simulation according to the decomposition of the fluid domain.

The final partial derivative term that is necessary is $\frac{\partial \mathbf{R}_f}{\partial \vec{\mathbf{x}}_f}$, which represents the change in the fluid residual with respect to the fluid mesh coordinates. The implementation of this partial derivative term is based on a first-order finite-difference approach due to its simplicity. Since the fluid domain uses a DG finite element discretization, the stencil for residual evaluations is compact. A perturbation is applied to a mesh nodal coordinate while maintaining the fluid state, which leads to changes in the residual evaluation of the adjacent elements due to the integration over an element's volume, which is now slightly changed. When viscous effects are not present, a single mesh node's perturbation only affects the elements directly adjacent to the node. However, when viscosity is present, the change in a mesh coordinate for a constant state changes the state gradient in the adjacent elements, which therefore cascades to the face neighbors of the adjacent elements, as depicted in Fig. 1. Since a single mesh coordinate perturbation only affects a small set of neighboring elements, many node coordinates can be perturbed at once to minimize the number of residual evaluations required to approximate the sensitivity entries of $\frac{\partial \mathbf{R}_f}{\partial \vec{\mathbf{x}}_f}$. This is accomplished using a greedy approach in the current implementation, where each perturbation iteration attempts to perturb as many nodes as possible. Even though the fluid system is potentially large, the relative cost of a residual evaluation is small and the evaluation of this partial derivative term remains tractable. When applied to an input vector, this matrix-vector product is also parallelized according to the fluid domain decomposition.

Eq. 19 describes the chain rule considering the direct mesh deformation being applied to the mesh coordinates and not with the analytical mapping which is part of the ALE formulation. If the ALE mapping were to be used, the chain

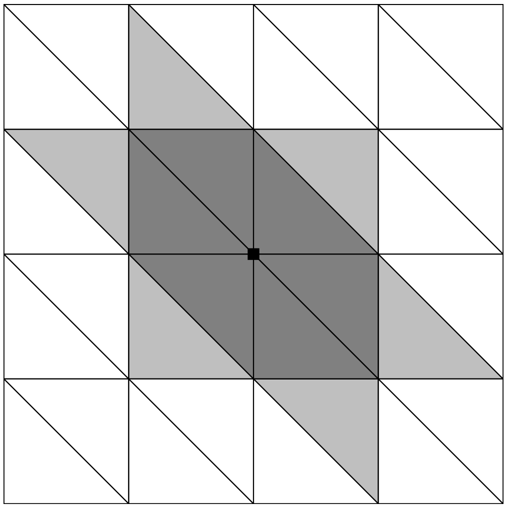


Fig. 1 Two-dimensional example of elements with residuals affected by a nodal coordinate perturbation (black square). Dark gray elements are always affected, light gray elements are affected only when viscosity is modeled, and white elements are not affected.

rule would be modified as

$$\left(\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_s}\right)^T \mathbf{\Psi}_f = \left(\frac{\partial \vec{\mathbf{d}}_s}{\partial \mathbf{u}_s}\right)^T \left(\frac{\partial \vec{\mathbf{d}}_f}{\partial \vec{\mathbf{d}}_s}\right)^T \left(\frac{\partial \vec{\mathbf{x}}_f}{\partial \vec{\mathbf{d}}_f}\right)^T \left(\frac{\partial \vec{\mathbf{X}}_f}{\partial \vec{\mathbf{x}}_f}\right)^T \left(\frac{\partial \mathbf{R}_f}{\partial \vec{\mathbf{X}}_f}\right)^T \mathbf{\Psi}_f, \tag{22}$$

where $\vec{\mathbf{X}}_f$ designates the fluid reference space mesh coordinates. In an ALE formulation, the residual is evaluated on the reference mesh while considering the states and fluxes which have been augmented due to the analytical mapping from the physical space, which contains the deformation effects. Therefore, the partial derivative of the fluid residual is computed with respect to the reference space mesh coordinates, $\frac{\partial \mathbf{R}_f}{\partial \vec{\mathbf{X}}_f}$. An intermediate sensitivity is then needed in the chain of partial derivatives that contains the effect of the analytical map between the reference and physical space, $\frac{\partial \vec{\mathbf{X}}_f}{\partial \vec{\mathbf{x}}_f}$. Assuming the analytical map, $\mathcal{G}$, is given by the RBF interpolation, the map could be evaluated at the mesh nodes in the following manner

$$\vec{\mathbf{x}}_f = \mathcal{G}(\vec{\mathbf{X}}_f) = \vec{\mathbf{X}}_f + \mathbf{\Psi}^{\mathrm{RBF}} \vec{\boldsymbol{\gamma}}. \tag{23}$$

Note however, that both $\mathbf{\Psi}^{\mathrm{RBF}}$ and $\vec{\boldsymbol{\gamma}}$ have an explicit dependence on $\vec{\mathbf{X}}_f$. Therefore, the evaluation of $\frac{\partial \vec{\mathbf{X}}_f}{\partial \vec{\mathbf{x}}_f}$ becomes quite cumbersome and potentially intractable for a mesh typical of high-fidelity CFD applications. It is for this reason that the ALE formulation was dropped, and presently the work focuses on steady applications.

2. Structural System Derivatives

The partial derivative of the structural residual with respect to the fluid states, $\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_f}$, can also be expressed through the chain rule of differentiation from various quantities

$$\left(\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_f}\right)^T \mathbf{\Psi}_s = \left(\frac{\partial \mathbf{u}_{f_q}}{\partial \mathbf{u}_f}\right)^T \left(\frac{\partial \vec{\mathbf{F}}_f}{\partial \mathbf{u}_{f_q}}\right)^T \left(\frac{\partial \vec{\mathbf{F}}_s}{\partial \vec{\mathbf{F}}_f}\right)^T \left(\frac{\partial \mathbf{R}_s}{\partial \vec{\mathbf{F}}_s}\right)^T \mathbf{\Psi}_s. \tag{24}$$

The term $\frac{\partial \vec{\mathbf{F}}_f}{\partial \mathbf{u}_{f_q}}$, represents the partial derivative of the fluid forces with respect to the fluid states located at the quadrature points on the deforming boundary surfaces. Since the evaluation of the surface pressure and shear loads is a boundary surface integral of the momentum fluxes, this term is derived analytically as the weighted sum of derivatives of the inviscid and viscous momentum fluxes with respect to the fluid states. However, since the desired sensitivity is made with respect to the states of the element nodes, the term $\frac{\partial \mathbf{u}_{f_q}}{\partial \mathbf{u}_f}$ is included, which represents the change in the states at the element quadrature points with respect to the states at the element nodes. This can be seen as in the following relation

$$\mathbf{u}_{f_q} = \Phi_f \mathbf{u}_f, \tag{25}$$

where Φ_f contains the evaluation of the element's basis functions at each of the element's quadrature points. Therefore, the partial derivative $\frac{\partial \mathbf{u}_{f_q}}{\partial \mathbf{u}_f}$ consists of the corresponding Φ_f for each element with a face on a deforming boundary surface. The term $\frac{\partial \vec{\mathbf{F}}_s}{\partial \vec{\mathbf{F}}_f}$, represents the partial derivative of the load transfer residuals, $\mathbf{L}$, with respect to the fluid surface loads and is also provided analytically by the MELD implementation. Lastly, the term $\frac{\partial \mathbf{R}_s}{\partial \vec{\mathbf{F}}_s}$ represents the sensitivity of the structural residuals with respect to the structural loads, which, given the form of the linear structural equations, is simply a square identity matrix. This partial derivative is therefore not explicitly formed or applied.

The final partial derivative term required for the coupled adjoint system is the partial derivative of the structural residuals with respect to the structural states, $\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s}$, which at first seems to be a straightforward derivative to evaluate given the linear form of the structural residual equations. However, this term is complicated due to the fact that the structural loads depend on the structural states through the coupling with the fluid domain as discussed in [16]; therefore this term is not simply the structural stiffness matrix. This partial derivative is also computed through the chain rule,

$$\left(\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s}\right)^T \mathbf{\Psi}_s = \left(\mathbf{K}_s - \frac{\partial \vec{\mathbf{F}}_s}{\partial \vec{\mathbf{F}}_f} \frac{\partial \vec{\mathbf{F}}_f}{\partial \vec{\mathbf{d}}_f} \frac{\partial \vec{\mathbf{d}}_f}{\partial \vec{\mathbf{d}}_s} \frac{\partial \vec{\mathbf{d}}_s}{\partial \mathbf{u}_s}\right)^T \mathbf{\Psi}_s. \tag{26}$$

The term $\mathbf{K}_s$ is the structural stiffness matrix. The terms $\frac{\partial \vec{\mathbf{F}}_s}{\partial \vec{\mathbf{F}}_f}$ and $\frac{\partial \vec{\mathbf{d}}_f}{\partial \vec{\mathbf{d}}_s}$ are partial derivatives which are already used in the evaluation of other partial derivative terms and are provided by the MELD implementation. The term $\frac{\partial \vec{\mathbf{F}}_f}{\partial \vec{\mathbf{d}}_f}$ is the partial derivative of the fluid surface load with respect to the fluid surface displacements. However, this is not provided analytically by xflow or through other means like automatic differentiation. Similarly to the term $\frac{\partial \mathbf{R}_f}{\partial \vec{\mathbf{x}}_f}$, it can be evaluated with a first-order finite difference approach. Since the load evaluation only involves a small set of elements on the boundary of the fluid domain, the size of this partial derivative term is typically much smaller than the entire fluid residual, so a direct perturbation approach can be used. This collection of partial derivatives then simplifies to $\frac{\partial \vec{\mathbf{F}}_s}{\partial \mathbf{u}_s}$, which describes the effect of the follower loads on the structural system.

3. *Output Functional Derivatives*

The right-hand side of the adjoint equations is made up of the partial derivatives of the output functional with respect to the fluid and structural states. These partial derivatives change depending on which output is being considered. Typical outputs of interest stemming from the fluid domain are the lift, drag, or moment coefficients of the body, which are evaluated through boundary integrals of the fluid states. The term $\frac{\partial J}{\partial \mathbf{u}_f}$ is determined similarly to $\frac{\partial \vec{\mathbf{F}}_f}{\partial \mathbf{u}_f}$ above, since the boundary integral of the fluid states is identical. As long as the structural node locations, $\vec{\mathbf{x}}_s$, are constant throughout the analysis, the partial $\frac{\partial J}{\partial \mathbf{u}_s}$, would be zero since the output does not explicitly depend on the structural states arising from the shape change of the domain. Typical outputs of interest stemming from the structural domain are the maximum displacement or stress for a given load case, which can be given smoothly and differentiably by Kreisselmeier-Steinhauser aggregation [28]. Likewise, these outputs do not explicitly depend on the fluid states, so the term $\frac{\partial J}{\partial \mathbf{u}_f}$ is zero. The partial derivative $\frac{\partial J}{\partial \mathbf{u}_s}$ on the other hand is provided analytically by TACS/TMR.

C. Adjoint Solution Procedure

With all the partial derivative terms evaluated, the coupled adjoint system described in Eq. 18 can be solved, and much like the primal problem, a block Gauss-Seidel fixed point iterative approach can be used to obtain the coupled adjoint fields. To begin, it is helpful to break up the coupled set of equations into two sets of matrix equations

$$\begin{aligned}\left(\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f}\right)^T \mathbf{\Psi}_f^{(k)} &= -\left(\frac{\partial J}{\partial \mathbf{u}_f}\right) - \left(\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_f}\right)^T \mathbf{\Psi}_s^{(k-1)} \\ &= \mathbf{A}_f^{(k)}\end{aligned} \tag{27}$$

$$\left(\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s}\right)^T \mathbf{\Psi}_s^{(k)} = -\left(\frac{\partial J}{\partial \mathbf{u}_s}\right) - \left(\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_s}\right)^T \mathbf{\Psi}_f^{(k)}, \tag{28}$$

where the superscript k denotes the Gauss-Seidel iteration number. Therefore, from inspection of Eq. 27, the structural adjoint is lagged when solving for the fluid adjoint field. The solution proceeds iteratively until a desired convergence is met on the updates to the adjoint fields. Similarly to the primal solution, under-relaxation can be applied to help with stability. One of the main advantages of this solution procedure is that the existing adjoint solvers currently in xflow and TACS/TMR can be reused with no modification. To do this, one slight adjustment is made to Eq. 28, where the term $\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s}$ is split between its stiffness component, with subscript $_K$, and its force component, with subscript $_F$,

$$\begin{aligned}\left(\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s}\right)_K^T \mathbf{\Psi}_s^{(k)} &= -\left(\frac{\partial J}{\partial \mathbf{u}_s}\right) - \left(\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_s}\right)^T \mathbf{\Psi}_f^{(k)} - \left(\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s}\right)_F^T \mathbf{\Psi}_s^{(k-1)} \\ &= \mathbf{A}_s^{(k)}.\end{aligned} \tag{29}$$

Note that the term associated with the forces also uses the lagged structural adjoint field. The coupled adjoint solution procedure is described in Algorithm 2.

Algorithm 2 Linear Block Gauss-Seidel adjoint solution process

```
Require: $\mathbf{\Psi}_f^0, \mathbf{\Psi}_s^0, k_{\max}, \omega \in (0,1]$
  for $k = 1$ to $k_{\max}$ do
    $\mathbf{A}_f^{(k)} = -\left(\frac{\partial J}{\partial \mathbf{u}_f}\right) - \left(\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_f}\right)^T \mathbf{\Psi}_s^{(k-1)}$    ▷ Assemble fluid adjoint RHS
    $\left(\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f}\right)^T \mathbf{\Psi}_f^{(k)} = \mathbf{A}_f^{(k)}$    ▷ Solve for the fluid adjoint field
    $\Delta \mathbf{\Psi}_f = \mathbf{\Psi}_f^{(k)} - \mathbf{\Psi}_f^{(k-1)}$    ▷ Evaluate the fluid adjoint field update
    $\mathbf{\Psi}_f^{(k)} \leftarrow \mathbf{\Psi}_f^{(k-1)} + \omega \Delta \mathbf{\Psi}_f$    ▷ Under-relax the fluid adjoint field
    $\mathbf{A}_s^{(k)} = -\left(\frac{\partial J}{\partial \mathbf{u}_s}\right) - \left(\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_s}\right)^T \mathbf{\Psi}_f^{(k)} - \left(\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s}\right)_F^T \mathbf{\Psi}_s^{(k-1)}$    ▷ Assemble structural adjoint RHS
    $\left(\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s}\right)_K^T \mathbf{\Psi}_s^{(k)} = \mathbf{A}_s^{(k)}$    ▷ Solve for the structural adjoint field
    $\Delta \mathbf{\Psi}_s = \mathbf{\Psi}_s^{(k)} - \mathbf{\Psi}_s^{(k-1)}$    ▷ Evaluate the structural adjoint field update
    $\mathbf{\Psi}_s^{(k)} \leftarrow \mathbf{\Psi}_s^{(k-1)} + \omega \Delta \mathbf{\Psi}_s$    ▷ Under-relax the structural adjoint field
    if $\|\Delta \mathbf{\Psi}_f\| \leq \varepsilon_{\mathrm{FSI}}$ and $\|\Delta \mathbf{\Psi}_s\| \leq \varepsilon_{\mathrm{FSI}}$ then
      break    ▷ Check for coupled convergence
    end if
  end for
```

IV. Output-Based Error Estimation and Mesh Adaptation

A. Adjoint-Weighted Residual Error Estimate

One advantage of evaluating the adjoint fields is that they can be used to estimate error in the output functional for a given level of discretization in the fluid and structural domains. To do this, one needs to have two different discretization levels in the meshes for both domains, namely a coarse approximation space, denoted by the superscript ^{H} and defined in a vector space $\mathcal{V}^H$, and a fine approximation space, denoted by the superscript ^{h} and defined in a vector space $\mathcal{V}^h$. The exact output error is defined as the difference between an exact output evaluation and an output evaluation computed with approximated states. However, for discretizations with finite-sized elements, an *approximation* of the output error can be made by using the fine-space output evaluation in place of the exact output evaluation,

$$\begin{aligned}\delta J &= J(\mathbf{u}_f, \mathbf{u}_s) - J^H(\mathbf{u}_f^H, \mathbf{u}_s^H) \\ &\approx J^h(\mathbf{u}_f^h, \mathbf{u}_s^h) - J^H(\mathbf{u}_f^H, \mathbf{u}_s^H).\end{aligned} \tag{30}$$

An assumption is made that the fine approximation space entirely contains the coarse approximation space such that $\mathcal{V}^H \subset \mathcal{V}^h$. For finite-element discretizations, this assumption allows for the fine space to be constructed by either a uniform h-refinement applied to the coarse space, or by uniformly incrementing the coarse-space order of approximation. Both methods result in a coarse-to-fine space prolongation operator, $\mathbf{I}^{H\to h}$, that can be used to obtain the fine-space field interpolation, $\bar{\mathbf{q}}^h$, given a coarse-space field, $\mathbf{q}^H$,

$$\bar{\mathbf{q}}^h = \mathbf{I}^{H\to h}\mathbf{q}^H. \tag{31}$$

In the case of both TACS/TMR and xflow, an order approximation increment, or p-prolongation, is used between the coarse and fine spaces, and the prolongation operator is the element-by-element interpolation of the coarse-space field using the coarse-space order element basis functions in the fine space. This prolongation process is valid for both the state field and the adjoint field for a given discipline. Given the lossless manner of the state interpolation, the output evaluated with the coarse-space state and the interpolated fine-space state should be equivalent. A key observation is that if the state field is interpolated on the fine space, it will not satisfy the fine-space residual, since the states were not determined with the higher-order basis at the fine-space level. The coarse-to-fine space prolongation can therefore be interpreted as a perturbation applied to the fine-space residual,

$$\begin{Bmatrix} \mathbf{R}^h_f(\bar{\mathbf{u}}^h_f;\bar{\mathbf{u}}^h_s) \\ \mathbf{R}^h_s(\bar{\mathbf{u}}^h_s;\bar{\mathbf{u}}^h_f) \end{Bmatrix} + \delta\mathbf{R}^h = \mathbf{0}. \tag{32}$$

Recognizing that the residual perturbation is related to output perturbations, *i.e.* error estimates, through Eq. 13, an adjoint-weighted residual is constructed which approximates the error in the output

$$J^h(\mathbf{u}^h_f,\mathbf{u}^h_s) - J^H(\mathbf{u}^H_f,\mathbf{u}^H_s) = \begin{bmatrix} \mathbf{\Psi}^{hT}_f & \mathbf{\Psi}^{hT}_s \end{bmatrix} \begin{Bmatrix} \mathbf{R}^h_f(\bar{\mathbf{u}}^h_f;\bar{\mathbf{u}}^h_s) \\ \mathbf{R}^h_s(\bar{\mathbf{u}}^h_s;\bar{\mathbf{u}}^h_f) \end{Bmatrix}, \tag{33}$$

where directly solving for the states in the fine space is avoided, but the fine-space adjoint field is required. Additionally, a perturbation in the adjoint field can also be defined

$$\begin{Bmatrix} \mathbf{\Psi}^h_f \\ \mathbf{\Psi}^h_f \end{Bmatrix} - \begin{Bmatrix} \bar{\mathbf{\Psi}}^h_f \\ \bar{\mathbf{\Psi}}^h_f \end{Bmatrix} + \begin{Bmatrix} \delta\mathbf{\Psi}^h_f \\ \delta\mathbf{\Psi}^h_s \end{Bmatrix} = \mathbf{0}, \tag{34}$$

which upon substitution into Eq. 33 yields the final adjoint-weighted residual expression, which is split into two main components. The first term on the right-hand side represents the computable correction, which for finite-element methods that satisfy Galerkin orthogonality as in the case of both TACS/TMR and xflow, equates to zero. The second term on the right-hand side represents the remaining error and is the source of error that is targeted for adaptive refinement.

$$J^h(\mathbf{u}^h_f,\mathbf{u}^h_s) - J^H(\mathbf{u}^H_f,\mathbf{u}^H_s) = \begin{bmatrix} \bar{\mathbf{\Psi}}^{hT}_f & \bar{\mathbf{\Psi}}^{hT}_s \end{bmatrix} \begin{Bmatrix} \mathbf{R}^h_f(\bar{\mathbf{u}}^h_f;\bar{\mathbf{u}}^h_s) \\ \mathbf{R}^h_s(\bar{\mathbf{u}}^h_s;\bar{\mathbf{u}}^h_f) \end{Bmatrix} - \begin{bmatrix} \delta\mathbf{\Psi}^{hT}_f & \delta\mathbf{\Psi}^{hT}_s \end{bmatrix} \begin{Bmatrix} \mathbf{R}^h_f(\bar{\mathbf{u}}^h_f;\bar{\mathbf{u}}^h_s) \\ \mathbf{R}^h_s(\bar{\mathbf{u}}^h_s;\bar{\mathbf{u}}^h_f) \end{Bmatrix}. \tag{35}$$

The evaluation of the error estimate then requires the fine-space adjoint field to be known. While the fine-space adjoint solution itself is linear and potentially much more tractable to solve compared to the fine-space state field, directly solving for the adjoint field can potentially become expensive and be of the same order of magnitude in expense as the coarse-space state field. Therefore, its direct solution is often avoided and an approximation is used in its place. One attractive alternative to the full solution is to reconstruct the fine-space adjoint field from the coarse-space adjoint in a localized manner, where neighboring elements can provide information that allow for the high-order field to be approximated locally in a least-squares sense. This is the approach taken in TACS/TMR to approximate $\mathbf{\Psi}^h_s$. Another alternative to exactly solving for the fine-space adjoint field is to prolongate the coarse-space adjoint into the fine space and then run several iterations of an iterative smoother, cutting off the solution early. This is the method used in xflow to approximate $\mathbf{\Psi}^h_f$.

B. Error Localization

With the necessary quantities known, the total error estimate is determined by the inner product of the adjoint perturbation with the fine-space residual, which results in a scalar value. If the total error estimate is summed with the output value of the coarse-space primal solution, it produces a corrected output evaluation. However, in order for the error estimate to be useful for mesh adaptation, the error must be localized to the elements within each domain, where an element in the coarse discretization is referred to as κ^H, and an element in the fine discretization is referred to as κ^h. In the case of both TACS/TMR and xflow where p-prolongation is used between the coarse and fine spaces, the element designations are equivalent, *i.e.* $\kappa^H = \kappa^h = \kappa$. The error can easily be localized to the nodes of each domain by computing the inner product between the adjoint and residual perturbation

$$\epsilon_n = \left\langle \delta\mathbf{\Psi}^{hT}, \mathbf{R}^h(\bar{\mathbf{u}}^h) \right\rangle \Big|_n, \tag{36}$$

where $\Big|_n$ restricts the inner product to the states at node n. For a DG finite element discretization like in xflow, the localization from the n_κ nodes (or more generally basis functions) in an element to the element itself is then the absolute value of the sum of the node-wise errors, *i.e.*,

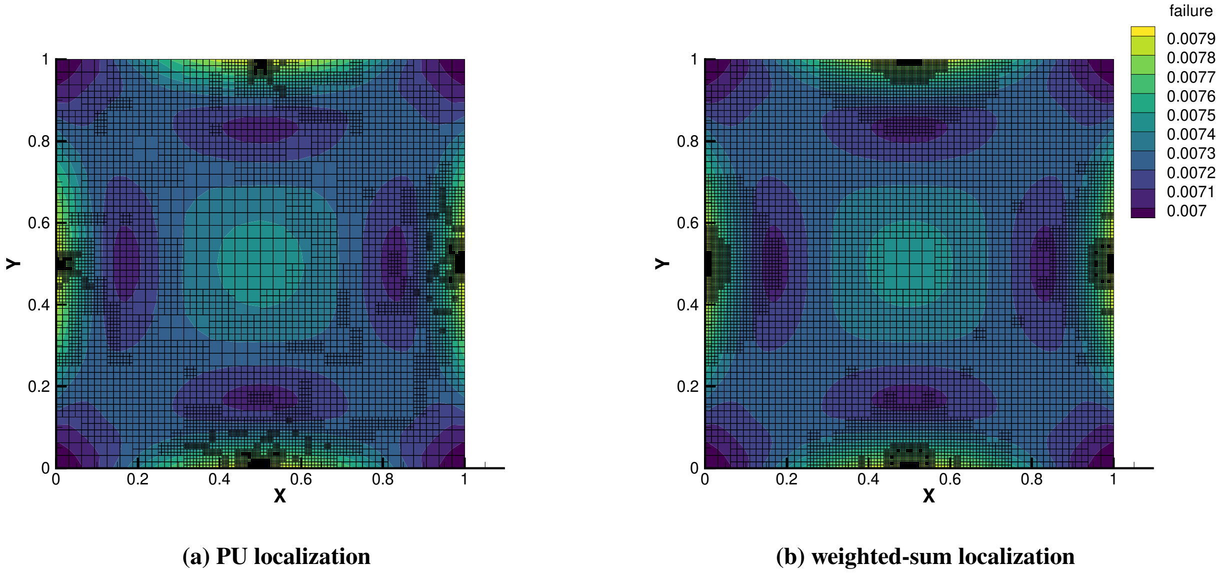
$$\epsilon_\kappa^{\text{DG}} = \left| \sum_{i=1}^{n_\kappa} \epsilon_{n_i} \right|. \tag{37}$$

This is due to the fact that each element in a DG discretization has an independent set of nodes/basis functions, and state continuity across element faces is not strictly enforced. Note that the absolute value is taken outside the sum of nodal errors, which allows for error cancellation due to opposing error signs within the element. The absolute value itself is placed at the element level to signify the equivalence of both positive and negative errors when elements are ordered based on their error indicators. For a continuous finite-element discretization, like in TACS/TMR, state continuity between elements is enforced through the sharing of nodes/basis functions between neighboring elements, which means that the error localization process is not as straightforward compared to the discontinuous method. Richter and Wick developed an error localization technique that uses low-order partition-of-unity (PU) basis functions to integrate the nodal errors and localize them to a reduced set of low-order nodes on each element [10], and is implemented in TACS/TMR [29]. This method leaves an awkward choice to be made however; after the error is localized to the low-order nodes in the mesh, it is still not immediately clear which elements will be marked for refinement. Refinement operations more naturally apply to an element, but the PU approach results in a ranking of errors in the mesh that is based on nodes rather than elements. Richter and Wick propose simply refining all elements attached to the nodes with high errors, but this requires a search process to determine the elements that should be flagged for refinement. Alternatively, a portion of the error at each low-order node can be attributed to each element, but the determination of the weight to use is not immediately clear, and the resulting approach is not as straightforward as the approach used for DG finite elements. Proposed here is a different approach based on a direct weighted-sum of the nodal errors for an element:

$$\epsilon_\kappa^{\text{CG}} = \left| \sum_{i=1}^{n_\kappa} w_i \epsilon_{n_i} \right|, \tag{38}$$

where the weights, w, are determined by the number of elements adjacent to every node. This weighted sum then directly splits the error at the nodes to their adjacent elements, and can also handle cases with hanging nodes.

A comparison between the PU approach and the weighted-sum approach to error localization is applied to a sample structural problem to demonstrate their differences. The sample problem considered presently is the linear elastic analysis of a square, flat plate with isotropic material properties, which is fully fixed along all four edges, and which is subject to a uniform out-of-plane pressure load. The corresponding stress field is symmetric, and the maximum stress occurs at the mid-side point along each edge of the plate. An h-adaptive analysis is carried out with TACS/TMR considering this problem targeting the KS-aggregated failure, which is simply the ratio of the von Mises stress with the material yield stress. The adaptive analysis is conducted once while using the PU error localization and again with the weighted sum error localization approach. The PU approach used a constant weight of $\frac{1}{4}$ for each low-order node error contribution when localizing to the elements, where the weight factor comes from the fact that there are 4 low-order nodes per quadrilateral element in the fine space. The final iteration of the adaptive analysis is shown in Fig. 2, where the contours represent the failure in the plate. The weighted sum localization approach seems to produce a mesh which is more symmetric and regular compared to the PU localization method. The PU method with a constant weight does not yield a symmetric result, since some nodal errors are not accurately counted during localization. Using a varying weight parameter would improve the localization, where the weight is determined by the number of adjacent elements each node has, but doing this makes the initial integration with a PU basis seem like an unnecessary step since the direct weighted sum of nodal errors seems to perform well on its own.



(a) PU localization

(b) weighted-sum localization

Fig. 2 Mesh and stress contours of the final adaptive iteration using two different error localization methods.

Lastly, the total error estimate in the output functional for a given discretization is the sum of the element-wise errors in the domain Ω

$$\epsilon = \sum_{\kappa \in \Omega} \epsilon_\kappa . \tag{39}$$

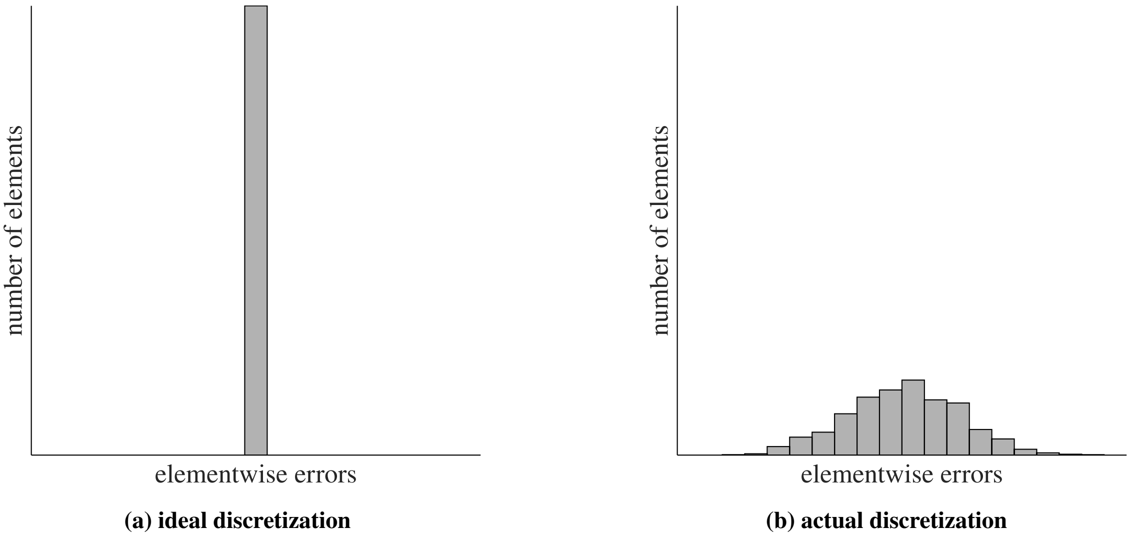
This total error estimate is conservative, since cancellation of error due to differing signs has been stripped out due to the absolute value operation applied to the element-wise error indicators. Note however, that this error estimate does not act as a strict upper bound to the error and can be shown to underreport error if the assumptions made above are not strictly satisfied.

C. Mesh Adaptation Strategies

With error localized to the elements, a mesh adaptation strategy can be used to selectively target elements for updating where it is most beneficial to refine or coarsen. Many adaptation strategies are contextualized by constructing an error histogram, where the element-wise errors are placed in bins sized by error quantities [30]. A typical assumption is that the "ideal" discretization features element errors that are perfectly equidistributed, and that all elements in the histogram lie within the same bin. However, the accuracy for most output functionals of engineering interest is usually determined by elements in a small, localized region of the computational domain. As such, error histograms produced from a fluid or structural analysis will often have a varying distribution of elementwise errors, where a small percentage of elements will carry very large or very small amounts of error. The disparity between an ideal and actual error histogram is visualized in Fig. 3. Adaptation strategies can therefore be devised that seek to iteratively approach this ideal error distribution in the mesh, where refinement operations target elements with large amounts of error and coarsening operations target elements with small amounts of error. Presently, only refinement operations are considered for adaptation. One popular strategy is a decreasing threshold method [31], where the elements containing the highest errors are refined first, and the error threshold is iteratively reduced. One form of this approach that has been shown to be effective is the fixed-fraction growth approach. This method prescribes a fixed fraction of the highest error-containing elements in the mesh to be refined at each adaptive iteration, which effectively makes the refinement threshold a function of the shape of the error histogram. The fixed-fraction growth approach is the adaptation strategy used in both xflow and TACS/TMR.

V. Single-Discipline Adaptation

In working towards enabling output-based mesh adaptation for high-order fluid-structure interaction, the uncoupled, single-discipline adaptive analyses should be verified independently prior to the coupled procedure. What follows is a



(a) ideal discretization

(b) actual discretization

Fig. 3 Sample error histograms considered for mesh adaptation.

demonstration of single-discipline error estimation and adaptation applied to a high-fidelity problem of interest relevant to aircraft design. The structural problem is based on determining the maximum stress in a wingbox design for a given set of loads. The fluid problem is based on determining the lift produced by a wing in its deformed configuration during cruise flight conditions. Based on the current computational framework that has been developed, both the structural and fluid solutions employ the error estimation and localization techniques described above, along with the fixed-growth mesh adaptation strategy. The structural domain refines the mesh based on hanging-node *h*-refinement, where the element size is refined locally and the order of approximation is held constant. The fluid domain refines the mesh based on *p*-refinement, where the element size is held constant and the order of approximation is incremented within an element.

A. uCRM-13.5 Model

The undeflected Common Research Model (uCRM) is the result of an effort to provide publicly available benchmarks for transonic aeroelastic wing analysis and design optimization [32]. The uCRM, a variant of the original NASA Common Research Model (CRM), is sized to be representative of a long-range, twin-aisle transport aircraft and is available in two different configurations: a traditional layout with an aspect ratio of 9, and a variant with an aspect ratio of 13.5 that is more flexible. The 13.5 aspect-ratio model is presently used. However, it is noted that the openly available geometry model for the wingbox contains ill-defined surfaces near the wingtip. As such, the uCRM-13.5 geometry model was sourced from the work of Thelen *et al.* [33] and meshed for both the fluid and structural domains. The wing-only configuration is used presently, where the fuselage and horizontal tail are not modelled in either the fluid or structural domain. Table 1 presents a subset of the general vehicle characteristics for the uCRM-13.5. The details of the transonic flight condition that will be used for testing are described in Table 2. Given that the goal of this study is to demonstrate the benefits of adaptive mesh refinement on this class of problems, it is natural to start with a relatively coarse mesh for both the fluid and structural domains since it is not known a priori where in the domain the solution will be sensitive to the specified outputs. Generally, this is an area where engineering judgement must be made, because it is desirable to start with a very coarse discretization and allow the error estimation and mesh adaptation to discover the optimal discretization through the adaptation process. However, starting with a mesh that is too coarse can violate one of the underlying assumptions made during error estimation, which is that the fine-space solution represents a good approximation to the truth solution. If this assumption is not properly satisfied, the error estimate will be of poor quality and may lead to refinement in areas of the domain which are not effective in increasing accuracy of the output. With this consideration in mind, the uCRM-13.5 was meshed to produce an initial discretization for both the fluid and structural

Table 1 uCRM-13.5 entire vehicle characteristics

Parameter	Value	Units
Aspect Ratio	13.5	-
Span	72.0	m
MAC	5.77	m
Reference Area	383.78	m^2
1/4 Chord Sweep	35	deg
Taper Ratio	0.25	-

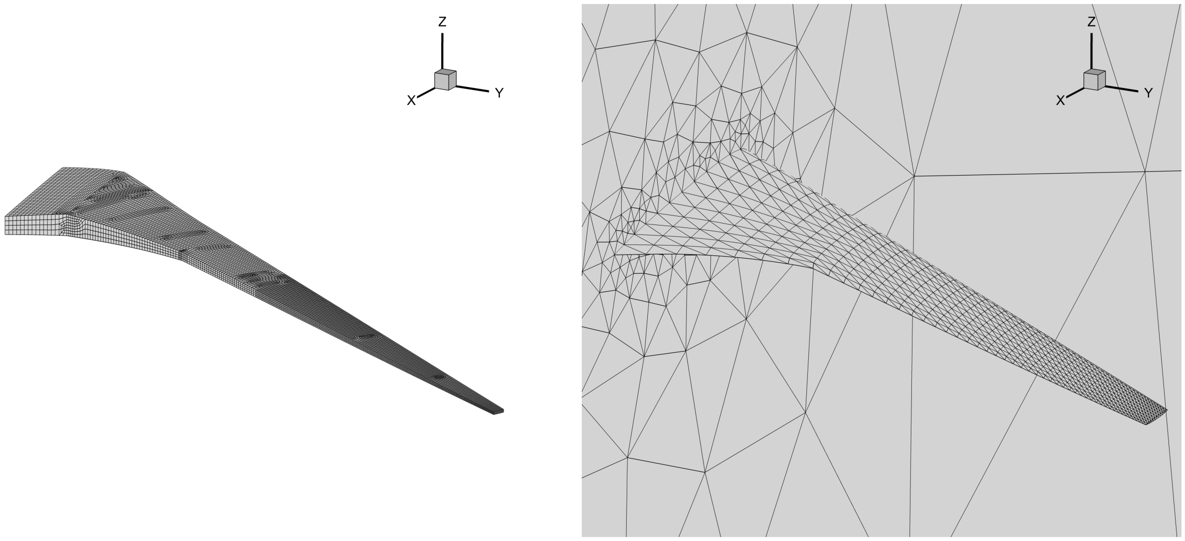
Table 2 uCRM-13.5 reference cruise data

Quantity	Value	Units
Altitude	11,277.6	m
Mach	0.85	-
Re	35,524,500	-
α	2.0	deg

domains for adaptive analysis, details of which can be found in Table 3, as well as depicted in Fig. 4. The fluid domain was meshed using Gmsh [34], and features quadratic, curved tetrahedron elements given the high-order nature of the solution.

Table 3 uCRM-13.5 initial discretization characteristics

Domain	Nodes	Elements	Degrees of Freedom	p
Fluid	35,871	24,180	24,180	0
Structure	18,723	4,971	112,338	2



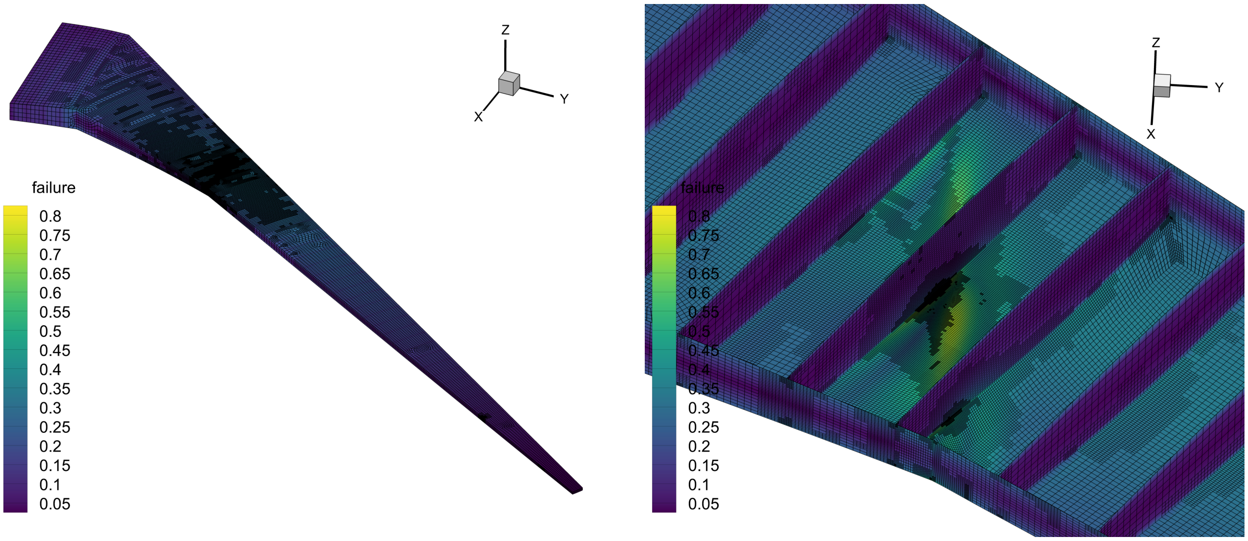
(a) Structural domain **(b) Fluid domain**

Fig. 4 Initial discretizations used to model the uCRM-13.5 wing.

B. Structural h-adaptation

The goal of the h-adaptive structural test case is to determine the maximum stress in the wingbox for a given set of loads. The wingbox itself uses the same boundary conditions that are described in the original reference for the uCRM; the root rib is fully fixed and the fifth rib, which aligns with the fuselage outer mold line, is fixed only in the in-plane and out-of-plane translational degrees of freedom. The model is defined in TACS/TMR with a uniform thickness distribution and isotropic material properties. Since the test case consists of a single discipline and the fluid domain is not modeled, a set of fixed loads is included as a surrogate for the loads coming from the fluid solution once the problem is properly coupled together. The fixed loads consist of a uniform traction applied in the out-of-plane direction on the upper and lower skin surfaces of the wingbox only, and a distributed inertial load is applied to the entire model to simulate the effect of gravity. The magnitude of the uniform traction was increased until the level of deflection at the wingtip was similar to what could occur in flight, approximately 7% of the semispan length. The output is again the KS-aggregated failure index, and therefore we expect the adaptation to refine primarily areas of the mesh containing stress concentrations. Using the fixed-growth adaptation strategy, up to 12% of elements are targeted for refinement at each iteration. The adaptive analysis is set to run until a set number of maximum adaptive iterations is reached, or the output error estimate reaches a tolerance of 0.01. Note that the tolerance is scaled to the units of the specified output.

The adaptive analysis converged in just six adaptive iterations and is compared to uniform refinement and remeshed discretizations to assess the level of convergence and efficacy of the adaptation. The mesh from the final adaptive iteration, as seen in Fig. 5, highlights that the maximum stress occurs where an internal rib connects to the bottom skin panel along the chord-wise line of the Yehudi break. The element-wise refinement indicators, which are based on the element errors, at each adaptive iteration are visualized in Fig. 6, where the error histograms can be seen to incrementally approach a more ideal shape. Fig. 7a compares the output history between the adaptive and uniform refinement analyses, where more than order of magnitude fewer degrees of freedom were necessary for convergence of the adaptive analysis, resulting in a significant saving in terms of both computational time and memory.



(a) Wingbox

(b) Zoom on area of interest – upper skin hidden

Fig. 5 Mesh and failure contour for the final adaptive solution in the structural-only test case.

C. Fluid p-adaptation

The goal of the p-adaptive fluid case is to determine the amount of lift produced by the wing in its deformed configuration at the cruise flight condition described above. Again, since the current analysis is not coupled, the wing begins in its undeformed jig shape, and each adaptive iteration receives a prescribed deformation to the wing surface to more closely mimic the coupled analysis behavior. The spanwise out-of-plane displacement, $\vec{d}_{f_z}(\bar{y})$, is given by the following expression

$$\vec{d}_{f_z}(\bar{y}) = c_z \left(1 - e^{-i}\right) \bar{y}^3, \tag{40}$$

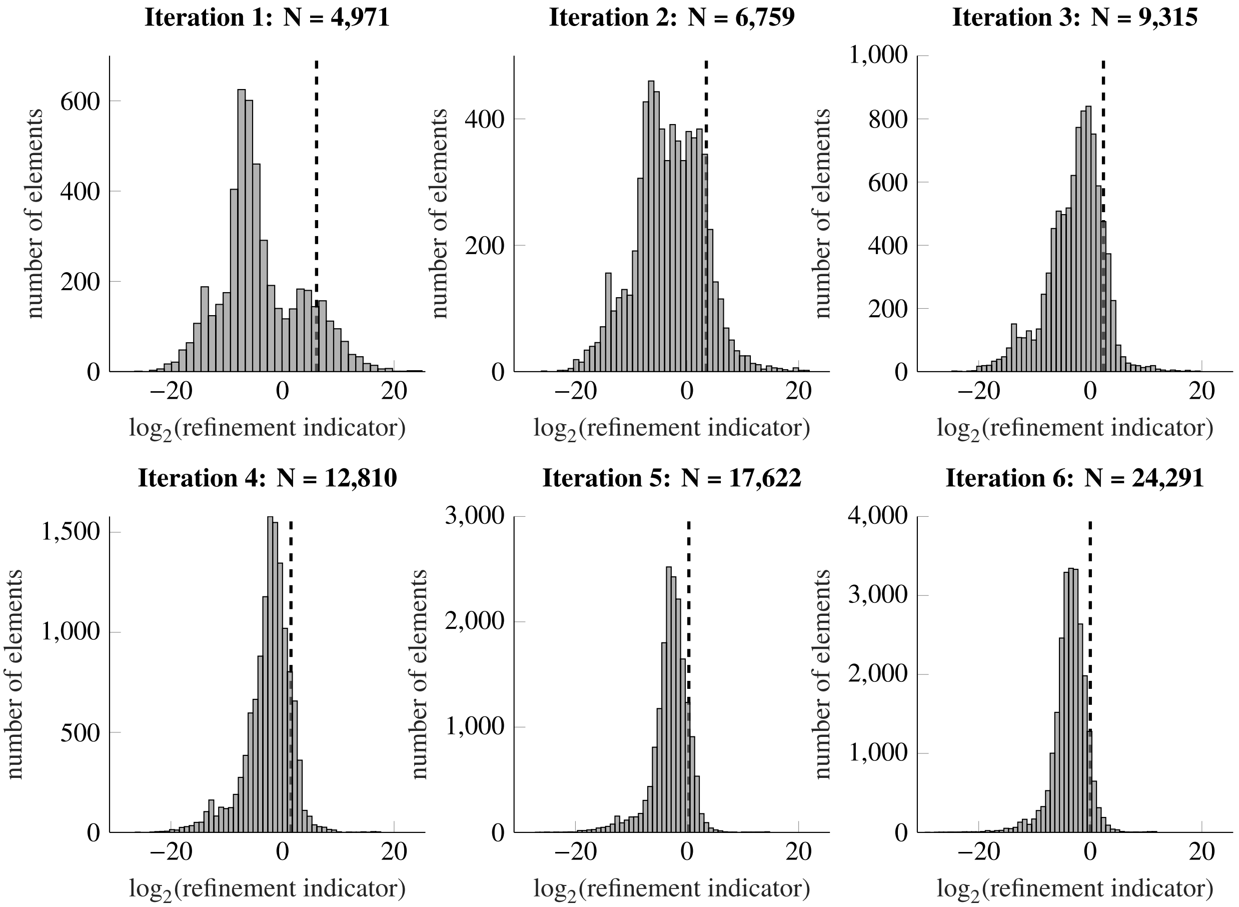
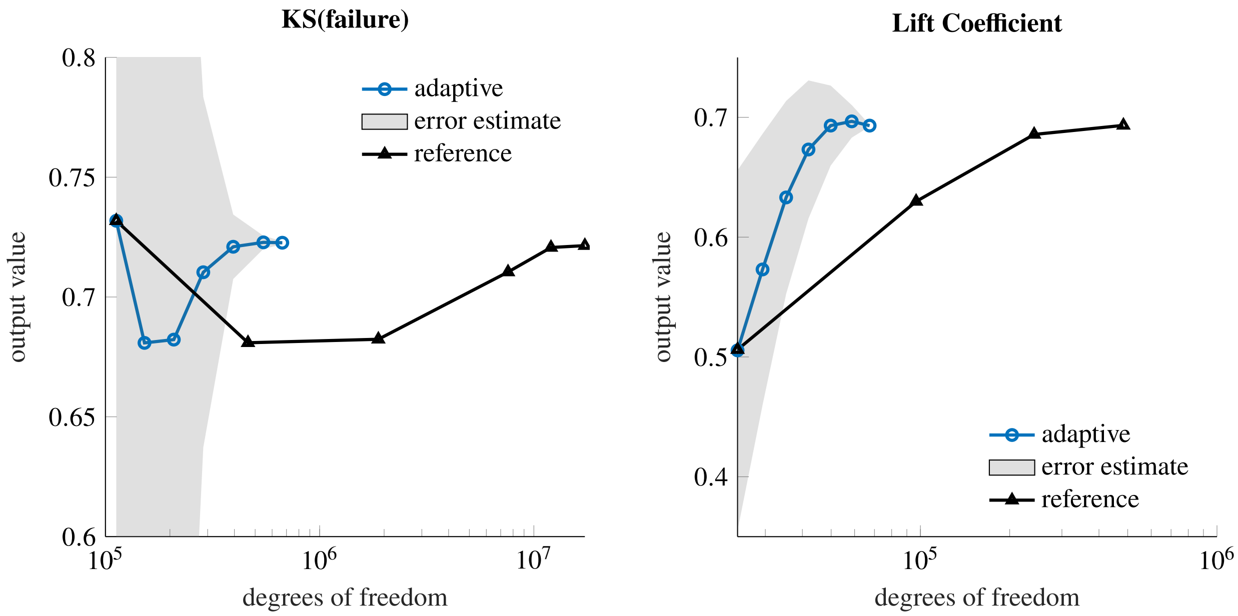


Fig. 6 Progression of error histograms for the structural h-adaptation test case, the dashed vertical line indicates the error value used for the refinement threshold.

where i is the adaptive iteration number, c_z is a constant selected to produce about an 8% tip deflection on the final adaptive iteration, and $\bar{y}$ is the spanwise coordinate normalized by the semispan length of the wing. The deformation is evaluated at the wing surface nodes and interpolated to the interior of the fluid domain using the multiscale RBF interpolation technique described above, where a base set of 500 nodes and support radius of 36 meters are used. The output is the total lift produced by the wing, which is evaluated by computing the boundary flux integral of the momentum states on the wing surface boundaries. The lift is converted from the solution units to a nondimensional coefficient of lift for visualization. A fixed-growth p-adaptation strategy is used with a growth factor of about 7% at each adaptive iteration, applied to the number of degrees of freedom, and an error tolerance of 0.001 in terms of the coefficient of lift. The current implementation of numerical quadrature rules in xflow limits the maximum order of the tetrahedron Lagrange basis functions to $p = 5$, so for adaptation iteration five and beyond, certain elements may be flagged for refinement but may not be allowed to refine further.

Similarly to the structural analysis, only six adaptive iterations were required for convergence within the specified error tolerance. The p-adaptive solution is compared to uniform p-refinement to assess the level of convergence and efficacy of the adaptation. The uniform p-refined mesh is tested at the final deformed configuration for uniform orders of $p = 0$ through $p = 3$. Fig. 7b compares the output history between the uniform and adaptive analyses, where again the adaptive analysis converged in approximately an order of magnitude fewer degrees of freedom when compared to a uniform refinement approach. The adaptive analysis also converged in 68.6% of the wall time required for the uniform analysis, with both cases running on the same 16 Intel® Xeon® E7-4850 v4 2.10GHz CPUs. The fluid solution and order distribution at the final adaptive iteration, as seen in Fig. 8, show that the elements refined the most lie along the leading and trailing edges of the wing as well as above the wing surface and upwind of the wing itself. Refinement at the leading and trailing edges helps to improve the accuracy of the momentum components most critical to the generation



(a) Structural *h*-adaptation

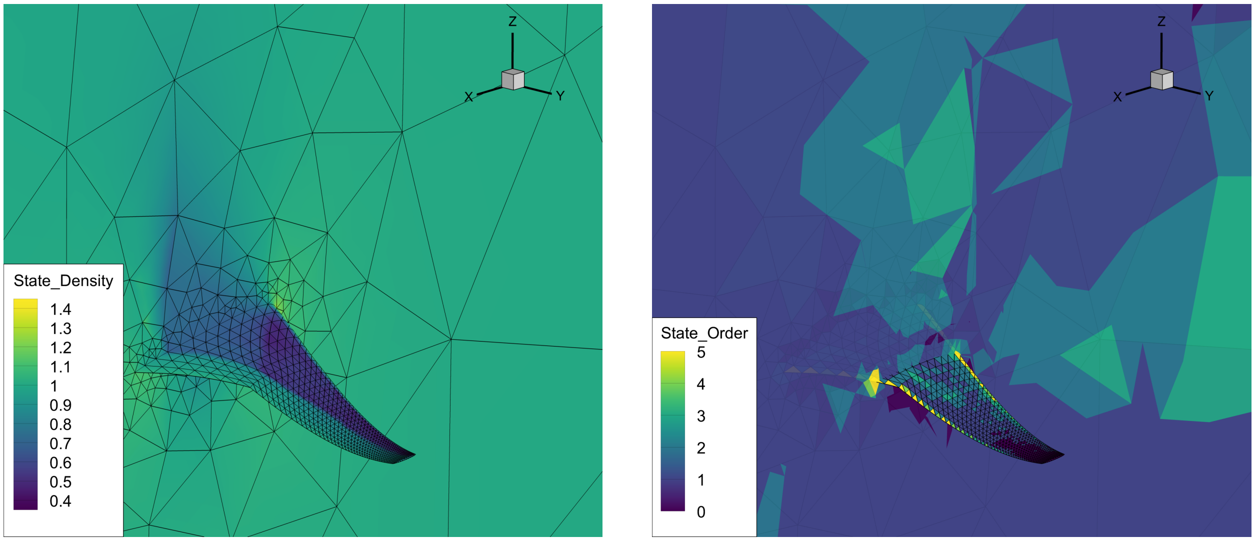
(b) Fluid *p*-adaptation

Fig. 7 Output convergence histories between adaptive and uniform refinement strategies.

of lift. Refinement upstream of the wing reduces errors that can be convected downstream and onto the wing surface boundary, which affect the lift evaluation as well. Note that some elements on the wing surface near the wing tip are kept at order $p = 0$ throughout the entirety of the analysis. This fact highlights one of the great benefits of output-based mesh adaptation: the method is able to identify and exploit the nature of an output definition, in this case the integrated nature of the lift output, which can lead to effective and non-intuitive adaptation patterns.

VI. Conclusions

In this paper, a partitioned approach for output-based error estimation and mesh adaptation for high-order fluid-structure interaction was presented with applications in aircraft design. The fluid-structure primal formulation along with the coupling and solution procedures were discussed. The fluid-structure adjoint formulation was derived and the solution procedure was also discussed. Practical implementation details regarding the evaluation of necessary partial derivative terms for the coupled adjoint were also discussed in depth. The usage of the coupled adjoint with regard to output-based error estimation was also presented, with discussions on error localization as well as an approach for mesh adaptation. While currently the full coupled adjoint solution is still being implemented and verified in the present computational framework, a verification of single-discipline adaptation was made for both the structural and fluid solutions independently. The single-discipline adaptation is applied in the context of high-aspect ratio wing design. The structural *h*-adaptive analysis converged to a solution for the maximum stress for a given set of loads with more than an order of magnitude fewer degrees of freedom when compared to uniform refinement and remeshing. Likewise, the fluid *p*-adaptive analysis converged to a solution for the lift evaluation in a deformed cruise configuration with approximately an order of magnitude fewer degrees of freedom when compared to uniform refinement. Both sets of adaptive analysis demonstrated significant savings in terms of both computational time and memory requirements, and motivates the objective for seeing the same benefits when applied to the fully coupled problem. In future work, the coupled adjoint implementation will be verified and output-based error estimation and mesh adaptation will be applied to the same high-aspect ratio wing design. Ultimately, we seek to demonstrate and quantify the computational benefits of output-based mesh adaptation in its ability to automate the mesh refinement process and leverage the advantages of high-order numerical methods in the high-fidelity analysis of large and complex aeroelastic systems.



(a) Density field **(b) Element p order – with cut plane at 25% semispan**

Fig. 8 State field and element p distribution for the final adaptive solution in the fluid-only test case.

Acknowledgments

This work was supported by the U.S. Air Force Research Laboratory (AFRL) under the Michigan-AFRL Collaborative Center in Aerospace Vehicle Design (CCAVD). The authors would like to thank in particular Dr. Philip Beran and Dr. Nathan Wukie for the technical exchanges. Additionally, the exchanges and insights of the TACS development team, including Prof. Graeme Kennedy (Georgia Institute of Technology) and Dr. Tim Brooks (NASA Glenn Research Center), have been instrumental in the development of TACS/TMR in the pursuit of this work. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

Bibliography

[1] Capaccio, T., "Cracks Found in B-2 Bombers," https://www.latimes.com/archives/la-xpm-2002-mar-20-fi-northrop20-story.html, 2002. Accessed: 2023-05-22.

[2] 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. 673–694. doi:10.2514/1.J050073.

[3] Ekaterinaris, J. A., "High-order accurate, low numerical diffusion methods for aerodynamics," *Progress in Aerospace Sciences*, Vol. 41, No. 3-4, 2005, pp. 192–300. doi:10.1016/j.paerosci.2005.03.003.

[4] Wang, Z. J., "High-order methods for the Euler and Navier–Stokes equations on unstructured grids," *Progress in Aerospace Sciences*, Vol. 43, No. 1-3, 2007, pp. 1–41. doi:10.1016/j.paerosci.2007.05.001.

[5] Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 811–845. doi:10.1002/fld.3767.

[6] Babuska, I., Szabo, B. A., and Katz, I. N., "The p-Version of the Finite Element Method," *SIAM Journal on Numerical Analysis*, Vol. 18, No. 3, 1981, pp. 515–545. doi:10.1137/0718033.

[7] Babuška, I., and Suri, M., "The p and h-p Versions of the Finite Element Method, Basic Principles and Properties," *SIAM Review*, Vol. 36, No. 4, 1994, pp. 578–632. doi:10.1137/1036141.

[8] Becker, R., and Rannacher, R., "An optimal control approach to a posteriori error estimation in finite element methods," *Acta numerica*, Vol. 10, 2001, pp. 1–102. doi:10.1017/S0962492901000010.

[9] Babuvška, I., and Rheinboldt, W. C., "Error estimates for adaptive finite element computations," *SIAM Journal on Numerical Analysis*, Vol. 15, No. 4, 1978, pp. 736–754. doi:10.1137/0715049.

[10] Richter, T., and Wick, T., "Variational localizations of the dual weighted residual estimator," *Journal of Computational and Applied Mathematics*, Vol. 279, 2015, pp. 192–208. doi:10.1016/j.cam.2014.11.008.

[11] Bathe, K.-J., and Zhang, H., "A mesh adaptivity procedure for CFD and fluid-structure interactions," *Computers & Structures*, Vol. 87, No. 11-12, 2009, pp. 604–617. doi:10.1016/j.compstruc.2009.01.017.

[12] Borker, R., Huang, D., Grimberg, S., Farhat, C., Avery, P., and Rabinovitch, J., "Mesh adaptation framework for embedded boundary methods for computational fluid dynamics and fluid-structure interaction," *International Journal for Numerical Methods in Fluids*, Vol. 90, No. 8, 2019, pp. 389–424. doi:10.1002/fld.4728.

[13] Ojha, V., Fidkowski, K., and Cesnik, C. E. S., "Adaptive Mesh Refinement for Fluid-Structure Interaction Simulations," *AIAA SciTech 2021 Forum*, 2021, p. 0731. doi:10.2514/6.2021-0731.

[14] Ojha, V., Fidkowski, K. J., and Cesnik, C. E. S., "Adaptive High-Order Fluid-Structure Interaction Simulations with Reduced Mesh-Motion Errors," *AIAA Journal*, Vol. 59, No. 6, 2021, pp. 2084–2101. doi:10.2514/1.J059730.

[15] Ojha, V., "Goal-Oriented Mesh Adaptation for High-Fidelity Aeroelastic Simulations," Ph.D. thesis, University of Michigan, Ann Arbor, 2022. doi:10.7302/6114.

[16] Kenway, G. K., Kennedy, G. J., and Martins, J. R., "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations," *AIAA Journal*, Vol. 52, No. 5, 2014, pp. 935–951. doi:10.2514/1.J052255.

[17] Frigoletto, B. E., Ojha, V., Fidkowski, K., and Cesnik, C. E. S., "Development of a High-Order Fluid-Structure Interaction Solver for Flexible Wings," *AIAA SciTech 2023 Forum*, 2023, p. 0185. doi:10.2514/6.2023-0185.

[18] Kast, S. M., and Fidkowski, K. J., "Output-based mesh adaptation for high order Navier–Stokes simulations on deformable domains," *Journal of Computational Physics*, Vol. 252, 2013, pp. 468–494. doi:10.1016/j.jcp.2013.06.007.

[19] Cockburn, B., Karniadakis, G. E., and Shu, C.-W., *Discontinuous Galerkin methods: theory, computation and applications*, Vol. 11, Springer Science & Business Media, 2012.

[20] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations," *Journal of Computational Physics*, Vol. 207, No. 1, 2005, pp. 92–113. doi:10.1016/j.jcp.2005.01.005.

[21] Kennedy, G. J., and Martins, J. R., "A parallel finite-element framework for large-scale gradient-based design optimization of high-performance structures," *Finite Elements in Analysis and Design*, Vol. 87, 2014, pp. 56–73. doi:10.1016/j.finel.2014.04.011.

[22] Bathe, K.-J., and Dvorkin, E. N., "A formulation of general shell elements—the use of mixed interpolation of tensorial components," *International Journal for Numerical Methods in Engineering*, Vol. 22, No. 3, 1986, pp. 697–722. doi:10.1002/nme.1620220312.

[23] Chin, T. W., Leader, M. K., and Kennedy, G. J., "A scalable framework for large-scale 3D multimaterial topology optimization with octree-based mesh adaptation," *Advances in Engineering Software*, Vol. 135, 2019. doi:10.1016/j.advengsoft.2019.05.004.

[24] Rendall, T. C., and Allen, C. B., "Unified fluid–structure interpolation and mesh motion using radial basis functions," *International Journal for Numerical Methods in Engineering*, Vol. 74, No. 10, 2008, pp. 1519–1559. doi:10.1002/nme.2219.

[25] Kedward, L., Allen, C. B., and Rendall, T. C., "Efficient and exact mesh deformation using multiscale RBF interpolation," *Journal of Computational Physics*, Vol. 345, 2017, pp. 732–751. doi:10.1016/j.jcp.2017.05.042.

[26] Kiviaho, J. F., and Kennedy, G. J., "Efficient and Robust Load and Displacement Transfer Scheme Using Weighted Least Squares," *AIAA Journal*, Vol. 57, No. 5, 2019, pp. 2237–2243. doi:10.2514/1.J057318.

[27] Kiviaho, J. F., Jacobson, K., Smith, M. J., and Kennedy, G., "A Robust and Flexible Coupling Framework for Aeroelastic Analysis and Optimization," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017, p. 4144. doi:10.2514/6.2017-4144.

[28] Kennedy, G. J., and Hicken, J. E., "Improved constraint-aggregation methods," *Computer Methods in Applied Mechanics and Engineering*, Vol. 289, 2015, pp. 332–354. doi:10.1016/j.cma.2015.02.017.

[29] Kennedy, G., "Adjoint-based Mesh Adaptation Techniques for Aggregation Functionals in Structural Design Optimization," *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2016, p. 4127. doi:10.2514/6.2016-4127.

[30] Aftosmis, M., and Berger, M., "Multilevel error estimation and adaptive h-refinement for cartesian meshes with embedded boundaries," *40th AIAA Aerospace Sciences Meeting & Exhibit*, 2002, p. 863. doi:10.2514/6.2002-863.

[31] Nemec, M., Aftosmis, M., and Wintzer, M., "Adjoint-Based Adaptive Mesh Refinement for Complex Geometries," *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008, p. 725. doi:10.2514/6.2008-725.

[32] Brooks, T. R., Kenway, G. K., and Martins, J. R., "Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings," *AIAA Journal*, Vol. 56, No. 7, 2018, pp. 2840–2855. doi:10.2514/1.J056603.

[33] Thelen, A. S., Bryson, D. E., Stanford, B. K., and Beran, P. S., "Multi-Fidelity Gradient-Based Optimization for High-Dimensional Aeroelastic Configurations," *Algorithms*, Vol. 15, No. 4, 2022, p. 131. doi:10.3390/a15040131.

[34] Geuzaine, C., and Remacle, J.-F., "Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities," *International Journal for Numerical Methods in Engineering*, Vol. 79, No. 11, 2009, pp. 1309–1331. doi:10.1002/nme.2579.