Development of a High-Order Fluid-Structure Interaction Solver for Flexible Wings

Braden E. Frigoletto[*], Vivek Ojha[†], Krzysztof J. Fidkowski[‡], and Carlos E. S. Cesnik[§]
University of Michigan, Ann Arbor, Michigan, 48104

This paper demonstrates an approach for creating a coupled partitioned solver for the high-order simulation of static fluid-structure interaction. A high-order discontinuous Galerkin formulation of the Navier-Stokes equations is used to model the fluid domain in addition to a high-order finite element method used to model the elastic structural domain. An Arbitrary Lagrangian-Eulerian formulation is used to facilitate mesh deformation in the fluid domain, using radial basis function interpolation. Fluid-structure coupling is also discussed, where spatial coupling is accomplished via a weighted least-squares extrapolation transfer technique for loads and displacements. A nonlinear block Gauss-Seidel iteration process is used to converge the coupled solution for static scenarios. The high-order coupled solution is verified for a three-dimensional test case of a flexible half-aircraft configuration at a reference cruise flight condition. A discussion of mesh refinement reviews the merits of both p- and h-refinement strategies applied to the current test case.

I. Introduction

There has been a growing interest in using high-order discretization methods for complex and challenging problems such as free-surface flows and fluid-structure interaction (FSI). Numerical simulation of FSI is an active area of research that spans many different disciplines. The focus of this paper is on the aeronautical applications of FSI where numerical simulations can be used to improve the understanding and prediction of coupled aeroelastic responses and instabilities observed in aircraft [1]. The interaction between the fluids and structures is nonlinear and involves multiple scales, thereby making the coupled system challenging to solve. Many approaches have been suggested for simulating fluid-structure interaction [2]. Numerical approaches for solving the coupled fluid-structure system can be broadly divided into two categories: monolithic and partitioned. The monolithic approach [3] is a fully-coupled approach where the two systems are solved simultaneously. This approach combines both subsystems into one large system of equations, which often leads to accurate results but requires significant implementation effort and uses less efficient solution techniques. The second approach, generally referred to as the partitioned approach [4], uses multiple solvers, each responsible for a single discipline in the problem, and couples the domains both spatially and temporally with a specified communication protocol or algorithm. This method facilitates software modularity and mathematical modelling and is what is used presently in this work.

One challenging aspect of the partitioned approach to FSI involves how to transfer necessary information between different disciplines or domains. The fluid-structure interface for the partitioned approach is often incongruous between each mesh, so oftentimes the nodes belonging to each domain on the interface are not shared between them, and the information at the nodes cannot be directly applied to the coupled domains. Additionally, large gaps can also exist between meshes at the interface, for example at the leading or trailing edge of a wing section outer-mold line and the internal wingbox structure. Transferring a field quantity such as loads or displacements across a non-matching interface can therefore require both interpolation and extrapolation of the field quantity, which can be difficult to achieve in a robust and efficient manner. Typical approaches to this problem include projection-based methods such as those provided by Brown [5] and Farhat *et al.* [6], as well as interpolation-based methods such as those based on radial basis functions (RBFs) provided by Rendall and Allen [7]. Projection-based methods typically require access to the underlying element shape functions and a connectivity between every interface node and nearest element, which complicates implementation. Interpolation-based methods on the other hand generally avoid these requirements and operate on sets of point clouds, however they tend to scale poorly with model size and require sub-sampling which can

[*] PhD. Candidate, Department of Aerospace Engineering

[†] PhD. Candidate, Department of Aerospace Engineering

[‡] Professor, Department of Aerospace Engineering, AIAA Associate Fellow

[§] Clarence L. "Kelly" Johnson Collegiate Professor, Department of Aerospace Engineering, AIAA Fellow lead to fictitious load and stress distributions when applied to aeroelastic systems. The presently adopted approach is the matching-based extrapolation of loads and displacements (MELD) method [8]. MELD is based on a weighted least-squares solution and is able to achieve localized information transfer similar to projection-based methods, with the modularity and non-intrusiveness typically associated with interpolation-based methods.

Another challenging aspect of the numerical simulation of FSI is how deformable domains are handled between the different disciplines. A common way of simulating problems involving deformable domains in computational fluid dynamics (CFD) is by using Arbitrary Lagrangian-Eulerian (ALE) methods [9]. In the ALE framework, the fluid mesh moves, but at a velocity different from that of the flow, which is useful for modeling problems in which boundaries of the mesh move or deform. The ALE method uses a map between the deforming physical domain and a static reference domain and solves transformed equations on the reference domain [10]. Fluid simulations based on the ALE formulation can use $r-$adaptation to obtain an optimized mesh close to solid boundaries [11] or use a mesh deformation technique to conform the fluid mesh to the moving boundaries. Several mesh deformation methods exist in the literature [12], and these can be classified into two main categories: 1) physical-analogy-based techniques and 2) interpolation-based techniques. A physical analogy method [13] may consider each edge of the mesh to behave as a spring, which has its own stiffness value. On the other hand, an interpolation-based method computes the movement of grid nodes as a function of boundary nodes, without any particular physical meaning. RBF interpolation [14] and inverse distance methods [15] are some examples of interpolation-based techniques. This paper considers primarily the mesh deformation techniques based on RBFs.

In this paper, the MELD transfer technique and RBF mesh motion interpolation are combined along with high-order formulations for both fluid and structural disciplines to create a high-order, parallel, computational FSI framework for static conditions. This framework is developed and verified for a three-dimensional test case featuring a half-aircraft model at a cruise flight condition. The outline of the remainder of the paper is as follows. Section II reviews the governing equations of the fluid and structural subsystems. Section III reviews the spatial coupling algorithm and solution process for the partitioned approach. Section IV outlines the test case details and the results generated using these methods. Finally, section V summarizes the approach and results of the coupled FSI solver presented and discusses future developments.

II. Governing Equations

A. Compressible Flow

The fluid system is governed by the Navier-Stokes equations, given by

$$\left.\frac{\partial \mathbf{u}_f}{\partial t}\right|_x + \nabla \cdot \vec{\mathbf{F}}(\mathbf{u}_f, \nabla \mathbf{u}_f) = \mathbf{0}, \quad \vec{\mathbf{F}} = \vec{\mathbf{F}}^i(\mathbf{u}_f) - \vec{\mathbf{F}}^v(\mathbf{u}_f, \nabla \mathbf{u}_f), \tag{1}$$

where $\mathbf{u}_f(\vec{x}, t) \in \mathbb{R}^{n_f}$ is the conservative state vector, $\vec{x} \in \mathbb{R}^d$ is the spatial coordinate, $t \in \mathbb{R}$ is time, and $\vec{\mathbf{F}}^i$ and $\vec{\mathbf{F}}^v$ are the inviscid and viscous fluxes, respectively. Additionally, n_f is the number of governing equations and d is the number of spatial dimensions. In the case of a non-deformable domain, the fluid equations are solved numerically in the Eulerian frame of reference, where the computational grid is fixed relative to the fluid. However, numerical simulation of fluid dynamics involving a deforming domain, such as in the case of FSI, faces issues due to the lack of a precise interface definition and under-resolved flow features when solved in the Eulerian frame of reference. Alternatively, the Lagrangian approach, in which each node in the fluid mesh follows the material particle during motion faces problems dealing with large distortions of the computational domain. To resolve these issues, an alternative method, the Arbitrary Lagrangian-Eulerian approach, has been introduced and is applied in the present work.

B. Arbitrary Lagrangian-Eulerian Formulation

The Arbitrary Lagrangian-Eulerian (ALE) approach combines advantages of both the Eulerian and Lagrangian approaches. In this method, the deformable physical domain is mapped to a fixed reference domain by a time-dependent mapping. A simple and effective ALE method for the discontinuous Galerkin discretization was introduced by Persson *et al.* [10] and a similar approach [16] is followed in this work.

Let the deforming physical space be defined by $v(t)$, the static reference space by V, and let $\mathcal{G}(\vec{X}, t)$ represent the one-to-one time-dependent mapping between the two spaces. Each point $\vec{X}$ in the static reference space is mapped to a corresponding point $\vec{x}(\vec{X}, t)$ in the physical space, based on the specified deformation of the mesh. The Jacobian of the mapping, represented by $\underline{G}$, and the mapping velocity, $\vec{v}_X$, are given by

$$\underline{G} = \nabla_X \mathcal{G}, \quad \vec{v}_X = \left.\frac{\partial \mathcal{G}}{\partial t}\right|_X. \tag{2}$$

Let $g = \det(\underline{G})$. The Navier-Stokes equations mapped to the reference frame can be written as [10]

$$\left.\frac{\partial \mathbf{u}_{Xf}}{\partial t}\right|_X + \nabla_X \cdot \vec{\mathbf{F}}_X(\mathbf{u}_{Xf}, \nabla_X \mathbf{u}_{Xf}) = \mathbf{0}, \quad \vec{\mathbf{F}}_X = \vec{\mathbf{F}}^i_X(\mathbf{u}_{Xf}) - \vec{\mathbf{F}}^v_X(\mathbf{u}_{Xf}, \nabla_X \mathbf{u}_{Xf}), \tag{3}$$

where the transformed state, derivatives, and fluxes in the reference frame are given by

$$\mathbf{u}_{Xf} = g\mathbf{u}_f, \tag{4}$$

$$\nabla_x \mathbf{u}_f = \nabla_X(g^{-1}\mathbf{u}_{Xf})\underline{G}^{-T} = (g^{-1}\nabla_X \mathbf{u}_{Xf} - \mathbf{u}_{Xf}\nabla_X(g^{-1}))\underline{G}^{-T}, \tag{5}$$

$$\vec{\mathbf{F}}^i_X = g\underline{G}^{-1}\vec{\mathbf{F}}^i - \mathbf{u}_{Xf}\underline{G}^{-1}\vec{v}_X, \qquad \vec{\mathbf{F}}^v_X = g\underline{G}^{-1}\vec{\mathbf{F}}^v. \tag{6}$$

C. Fluid System Spatial Discretization

To discretize the state equations, given in Eqn. 3, a discontinuous Galerkin (DG) finite-element method [17–19] is used in space. As a finite-element method, DG approximates the state $\mathbf{u}_f$ in functional form using linear combinations of basis functions on each element, where no continuity constraints are imposed between adjacent elements. Given T_h, a non-overlapping tessellation of the domain Ω into N_e elements, the state on an element, Ω_e, is approximated as

$$\left.\mathbf{u}(\vec{x}(\vec{\xi}))\right|_{\Omega_e} = \sum_{n=1}^{N_p} \mathbf{U}_{en}\phi_{en}(\vec{x}(\vec{\xi})). \tag{7}$$

In this equation, N_p is the number of basis functions per element, $\mathbf{U}_{en}$ is the vector of s coefficients for the n^{th} basis function on element e, $\phi_{en}(\vec{x}(\vec{\xi}))$, and n_f is the state rank. $\vec{X}$ denotes the global coordinates, and $\vec{\xi}$ denotes the reference-space[*] coordinates in a master element. Formally, $\mathbf{u} \in \boldsymbol{\mathcal{V}}_h = [\mathcal{V}_h]^{n_f}$, where, if the elements are not curved, $\mathcal{V}_h = \left\{u \in L_2(\Omega) : u|_{\Omega_e} \in \mathcal{P}^p \ \forall\ \Omega_e \in T_h\right\}$, and $\mathcal{P}^p$ denotes polynomials of order p on the element. With the spatial discretization described above, the governing equations can written in abbreviated form as

$$\mathbf{R}_f(\mathbf{u}_f, \vec{\mathbf{u}}_s) = \mathbf{0}, \tag{8}$$

where $\mathbf{R}_f$ is the discrete spatial residual vector. Each aeroelastic iteration requires the minimization of $\mathbf{R}_f$ below a specified tolerance, ϵ_f, after which the fluid pressure and shear forces can be post-processed on the surface boundaries that are marked as part of the fluid-structure interface. The forces are proportional to the fluid stresses determined at each element quadrature point on the boundary, where the pressure, $\vec{p}_q$, and shear, $\vec{s}_q$, stresses for each quadrature point q can be expressed as the following

$$\begin{aligned} \vec{p}_q &= \mathbf{I}^{\text{mom}}\mathbf{F}^i_q, \\ \vec{s}_q &= \mathbf{I}^{\text{mom}}\mathbf{F}^v_q, \end{aligned} \tag{9}$$

where $\mathbf{F}^i_q$ and $\mathbf{F}^v_q$ are respectively the normal inviscid and viscous fluxes evaluated at the quadrature points, and $\mathbf{I}^{\text{mom}} \in \mathbb{R}^{d \times n_f}$ is a matrix that isolates the d momentum components from the n_f entries in the conservative flux. The total fluid stress defined at a single quadrature point, $\vec{f}_q$, is then just the sum of the pressure and shear stresses. The total fluid force vector can be obtained by integrating the quadrature point stresses with the appropriate quadrature weights w_q such as

$$\vec{F}_f = \sum_{q=1}^{n_q} \vec{F}_{fq}, \quad \vec{F}_{fq} = w_q \vec{f}_q = w_q(\vec{p}_q + \vec{s}_q), \tag{10}$$

where $\vec{F}_f \in \mathbb{R}^d$ is the total fluid force on the boundary, and n_q is the number of quadrature points on the boundary. The quadrature-point contributions to the total fluid force vector, $\vec{F}_{fq}$, are assembled into a global array, $\vec{\mathbf{F}}_f \in \mathbb{R}^{n_q \times d}$, and passed to the structural subsystem for aeroelastic analysis. The fluid physics and DG finite-element method are implemented in the CFD code used presently called xflow.

[*] Reference-space here pertains to the master element, not the ALE reference domain.

D. Linear Elasticity

For the case of linear elasticity, the structural residual can be written as the following, where the subscript s denotes terms related to the structural subsystem

$$\begin{aligned} \mathbf{R}_s = \mathbf{K}\vec{\mathbf{u}}_s - \vec{\mathbf{F}}_s = \mathbf{0}, \\ \mathbf{R}_s(\vec{\mathbf{u}}_s, \mathbf{u}_f) = \mathbf{0}, \end{aligned} \tag{11}$$

where $\mathbf{K}$ is the structural stiffness matrix of the undeformed structure, $\vec{\mathbf{u}}_s$ is the vector of structural states, and $\vec{\mathbf{F}}_s$ is the structural load vector. Note that a spatial coupling mechanism is necessary to transfer $\vec{\mathbf{F}}_f$ to $\vec{\mathbf{F}}_s$ in a manner that is both consistent and conservative. These equations are implemented in the Toolkit for the Analysis of Composite Structures (TACS) [20, 21]. TACS is an open source, adjoint-enabled, parallel finite-element analysis framework typically used for gradient-based multidisciplinary design optimization. TACS has previously been coupled with various fluid solvers for aerostructural optimization applications [22–24]. Once the structural displacements are solved for, the spatial coupling mechanism is used to transfer $\vec{\mathbf{u}}_s$ to the set of fluid boundary surface displacements, $\vec{\mathbf{w}}_f$.

E. Structural System Spatial Discretization

The structural domain is discretized in space with a finite element method. The kinetic and potential energies are integrated over each element in the structural domain based on the type of element used and its corresponding shape functions. TACS most commonly employs a general-purpose high-order shell element based on the Mixed Interpolation of Tensorial Components (MITC) [25] formulation, although other element types are supported as well. These shell elements are well-suited for thin-walled structures typically associated with the structures of aircraft. Note that while nodal rotations are included in the structural states for MITC shell elements, they are not included when passing structural information to the fluid interface. Only the structural displacements are transferred.

III. Fluid-Structure Coupling

A. Spatial Coupling

The spatial coupling between the fluid and structural subsystem involves information transfer across the fluid-structure interface. In general, the fluid-structure interface is non-matching between the two domains, so the transfer of loads from the fluid mesh to the structural mesh and the displacement and velocity of the structural mesh to the fluid mesh is not directly one-to-one. Various methods for the information transfer across non-matching sets exist [26–28], one such method being the Matching-based Extrapolation of Loads and Displacements (MELD) [8]. MELD has previously been implemented within the open-source toolkit for aeroelastic analysis and optimization called FUNtoFEM [24], which is used presently.

MELD works by first initializing the surface boundaries in both the fluid and structural meshes, associating each fluid surface mesh node, $\vec{\mathbf{x}}_f$, to a specified number of nearest structural surface nodes, $\vec{\mathbf{x}}_s$. The vector of fluid surface mesh displacements, $\vec{\mathbf{w}}_f$, is solved for in a weighted least-squares sense preserving rigid-body translation and rotation, where the transfer operation can be written in a residual form and is dependent on the surface node locations and structural displacements

$$\mathbf{D}(\vec{\mathbf{x}}_s, \vec{\mathbf{x}}_f, \vec{\mathbf{u}}_s, \vec{\mathbf{w}}_f) = \mathbf{0}. \tag{12}$$

The fluid loads, which are evaluated at the quadrature points of the fluid surface boundaries, 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, which is dependent on the surface node locations, structural displacements, and fluid loads

$$\mathbf{L}(\vec{\mathbf{x}}_s, \vec{\mathbf{x}}_f, \vec{\mathbf{u}}_s, \vec{\mathbf{F}}_s, \vec{\mathbf{F}}_f) = \mathbf{0}. \tag{13}$$

An advantage of MELD over other methods is that it operates entirely with sets of point clouds, not requiring any underlying connectivity information for the structural and fluid surface meshes as well as not requiring any element integration of shape functions. Additionally, when compared to an interpolation-based RBF transfer scheme where sub-sampling is necessary to reduce computational time, the load distribution from MELD avoids fictitious load concentrations and leads to more accurate and realistic load and resulting stress distributions in the structure. MELD offers a few parameters that determine the quality of the information transfer and that can be freely adjusted. The decay parameter, β, controls the amount of influence a node has on the transferred quantity based on spatial proximity. The number of structural nodes, N, used to link to each fluid node affects how well loads are conserved and displacements are localized in the resulting transfer. Note that in the current implementation, two different sets of point clouds are used for each direction of information transfer across the fluid-structure interface. The structural surface nodes are used in both sets. However, the fluid surface mesh nodes are used to receive displacements and velocities, whereas the fluid surface quadrature points are used to send fluid loads.

B. Mesh Motion

For deforming domains, the ALE formulation of the Navier-Stokes equations requires a mapping between the reference and deformed physical mesh. In this work, RBFs provide that mapping and are used for deforming the interior fluid mesh to conform to the flexible structure that define the fluid-structure interface on the boundaries of the fluid mesh. To determine a specific fluid volume node displacement or velocity, it is interpolated based on the interpolation function taking the form

$$\vec{q}(\vec{X}) = \sum_{i=1}^{n_c} \vec{\gamma}_i \phi(\|\vec{X} - \vec{X}_{c_i}\|) + \vec{p}(\vec{X}), \tag{14}$$

where $\vec{q}(\vec{X})$ is the interpolation function, n_c is the number of RBF centers (nodes where a field quantity is known, *i.e.* the number of fluid surface nodes), $\vec{\gamma}$ is a set of interpolation coefficients (one for each spatial dimension), ϕ is a given basis function with respect to the Euclidean distance $\|\cdot\|$, and $\vec{p}(\vec{X})$ is a linear polynomial. Considering the application of mesh motion, the inclusion of the linear polynomial term dictates whether the resulting interpolation function preserves rigid body translations and rotations in the mesh. The preservation of rigid body translations and rotations is not desired since the motion of the boundary should be diffused and smoothed throughout the interior of the domain. Therefore, the linear polynomial terms are neglected. For a given deformed fluid surface boundary, a requirement is placed on the interpolation function to return the known states at the RBF centers, *i.e.* the fluid surface boundary nodes, which leads to the following linear system that can be solved to determine the unique set of interpolation coefficients

$$\vec{q}(\vec{X}_c) = \vec{\mathbf{w}}_f, \tag{15}$$

$$\mathbf{\Phi}\vec{\boldsymbol{\gamma}} = \vec{\mathbf{w}}_f, \tag{16}$$

where $\mathbf{\Phi} \in \mathbb{R}^{n_c \times n_c}$ is a symmetric matrix containing the evaluation of the RBF between each center point, and $\vec{\boldsymbol{\gamma}} \in \mathbb{R}^{n_c \times d}$ is a matrix of interpolation coefficients for each point corresponding to each spatial dimension. The same process can be followed for the fluid surface velocity, $\dot{\vec{\mathbf{w}}}_f$, to determine its own unique set of interpolation coefficients. For steady FSI, however, the mesh velocity is defined to be zero and can be neglected in the solution. The residual form of the mesh motion linear system can be written as

$$\mathbf{H}(\vec{\mathbf{w}}_f, \vec{\boldsymbol{\gamma}}) = \mathbf{0}. \tag{17}$$

The interpolation coefficients are used to determine the new mapping, $\mathcal{G}(\vec{X}, t)$, which is unique for each quadrature point involved in an evaluation of the fluid residual. The computational cost of RBF interpolation methods scales poorly as the number of center points increases. However, for the application of mesh motion, only the general shape and geometry of the deforming boundaries needs to be represented for the mesh displacement and velocity to be interpolated well and mesh quality to be preserved. Therefore, sub-sampling of the fluid surface mesh is typically satisfactory in preserving mesh quality while keeping the computational cost low. Wendland's C^2 basis function,

$$\phi(R) = \begin{cases} 0, & \text{if } R > 1 \\ (4R+1)(1-R)^4, & \text{if } R \leq 1 \end{cases} \tag{18}$$

where $R = \|\vec{X} - \vec{X}_c\|/r$, and r is the support radius, is used to achieve improved efficiency [29]. Compact supports restrict the motion in the mesh to the local vicinity of the moving boundary specified by the support radius. Points outside of the support radius have no influence and do not participate in the interpolation of motion in the mesh, thereby reducing the computational cost when applying the mapping between the physical and reference domains. Wendland's family of RBFs come in varying levels of smoothness as well (C^0, C^2, C^4, C^6, etc.), which can be advantageous to use for fluid meshes with curved elements and high-order solutions.

With the interpolation coefficients known, the ALE mapping then becomes the following, which is unique for each quadrature point located at position $\vec{X}$ at time t in the mesh

$$\vec{x} = \mathcal{G}(\vec{X}, t) = \vec{X} + \sum_{i=1}^{n_c} \vec{\gamma}_i \phi(R_i). \tag{19}$$

The Jacobian of the mapping for each point, $\underline{G} \in \mathbb{R}^{d \times d}$, is determined through the chain rule of differentiation

$$\underline{G}_{jk} = \underline{I}_{jk} + \sum_{i=1}^{n_c} \gamma_{ij} \frac{\partial \phi}{\partial R_i} \frac{\partial R_i}{\partial X_k} \text{ where, } j, k = 1 \ldots d. \tag{20}$$

C. Static FSI Solution Process

The nonlinear block Gauss-Seidel process for aeroelastic iterations is used to converge the coupled solver for a static condition. Each coupled iteration begins with a fluid evaluation to determine fluid loads. To aid in the stability of the coupled solution, a relaxation factor, ω, can be applied to either the fluid loads or the structural update prior to being transferred across the fluid-structure interface. In the case where geometrically-nonlinear effects are included in the structural solution, the relaxation is traditionally applied to the fluid loads rather than the structural displacements. However, for a linear structural analysis, both choices are equivalent. In the current implementation, the relaxation, which is determined at each iteration with the Aitken acceleration update formula, is applied to the fluid loads. After the fluid loads are relaxed, they are transferred from the fluid surface quadrature points to the structural surface nodes via the MELD transfer scheme. Following this, the structural solution is obtained. The coupled convergence is then evaluated and compared to a convergence criterion. If convergence has not yet been reached, the structural displacements are then transferred to the fluid surface mesh via MELD. The RBF mesh motion interpolation coefficients are then solved prior to the next fluid evaluation, where the mesh motion mapping occurs within the ALE formulation. The cycle repeats until the convergence criterion is met or a maximum number of coupled iterations is reached. The process assumes that each solver independently solves its own set of governing equations for a particular deformed configuration within a specified tolerance, ϵ_f and ϵ_s for the fluid and structural solvers, respectively. The L_2 norm of the structural update is compared with ϵ_{FSI} to determine the convergence of the coupled system. Algorithm 1 describes the iterative solution process.

Algorithm 1 Nonlinear Block Gauss-Seidel solution process.

```
Require: $\vec{\mathbf{x}}_s, \vec{\mathbf{x}}_f, k_{\max}, \omega, \omega_{\min}$
  $\vec{\mathbf{u}}_s^{(0)} \leftarrow \mathbf{0}, \vec{\mathbf{F}}_f^{(0)} \leftarrow \mathbf{0}, \Delta\vec{\mathbf{F}}_f^{(0)} \leftarrow \mathbf{0}$    ▷ Initialize transfer and update quantities to 0
  for $k = 1$ to $k_{\max}$ do
      $\vec{\mathbf{F}}_f^{(k)} \leftarrow \mathbf{R}_f(\mathbf{u}_f^{(k)}, \vec{\mathbf{u}}_s^{(k-1)}) \leq \epsilon_f$    ▷ 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
      $\vec{\mathbf{F}}_s^{(k)} \leftarrow \mathbf{L}(\vec{\mathbf{x}}_s, \vec{\mathbf{x}}_f, \vec{\mathbf{u}}_s^{(k-1)}, \vec{\mathbf{F}}_s^{(k)}, \vec{\mathbf{F}}_f^{(k)}) = \mathbf{0}$    ▷ Transfer the fluid surface loads to the structural surface
      $\vec{\mathbf{u}}_s^{(k)} \leftarrow \mathbf{R}_s(\vec{\mathbf{u}}_s^{(k)}, \mathbf{u}_f^{(k)}) \leq \epsilon_s$    ▷ 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 $\|\Delta\vec{\mathbf{u}}_s^{(k)}\| \leq \epsilon_{\text{FSI}}$ then
          break    ▷ Check for coupled convergence
      end if
      $\vec{\mathbf{w}}_f^{(k)} \leftarrow \mathbf{D}(\vec{\mathbf{x}}_s, \vec{\mathbf{x}}_f, \vec{\mathbf{u}}_s^{(k)}, \vec{\mathbf{w}}_f^{(k)}) = \mathbf{0}$    ▷ Transfer structural displacements to fluid surface boundary
      $\vec{\boldsymbol{\gamma}}^{(k)} \leftarrow \mathbf{H}(\vec{\mathbf{w}}_f^{(k)}, \vec{\boldsymbol{\gamma}}^{(k)}) = \mathbf{0}$    ▷ Solve for mesh motion interpolation coefficients
  end for
```

Both the TACS and FUNtoFEM libraries are written in C++, but also include a Python interface that simplifies their setup and usage. A similar Python interface was developed for xflow, which is written in C. A common Python interface allows for parallel coupled FSI solutions to be done using these three components and facilitates the data passing between them. Each solver is initialized via its respective input files and performs its solution on its own MPI communicator, consisting of its own allotment of processors. Data are transferred via the MELD transfer scheme implemented in FUNtoFEM, which handles the data across the different communicators and domains modeled in the problem. All of the present results were generated using an allotment of 24 Intel® Xeon CPU E7-4850 v4 running at 2.10 GHz.

IV. Numerical Results

A. uCRM-9 Test Case

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 [30]. 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. In the interest of avoiding geometric nonlinearities in the structural response in this preliminary work, the aspect ratio 9 configuration is used. Table 1 presents a subset of the general vehicle characteristics. The reference condition used for verification is a cruise flight condition, the details of which can be found in Table 2. This reference condition includes data for both fluid and structural outputs for varying levels of mesh fidelity. In the interest of computational time, the coarsest of structural and fluid meshes is used for this verification. Note that the Reynolds number for this reference condition uses the uCRM-9 mean aerodynamic chord (MAC) for the characteristic length. Fig. 1 shows both the structural and fluid models for the uCRM-9.

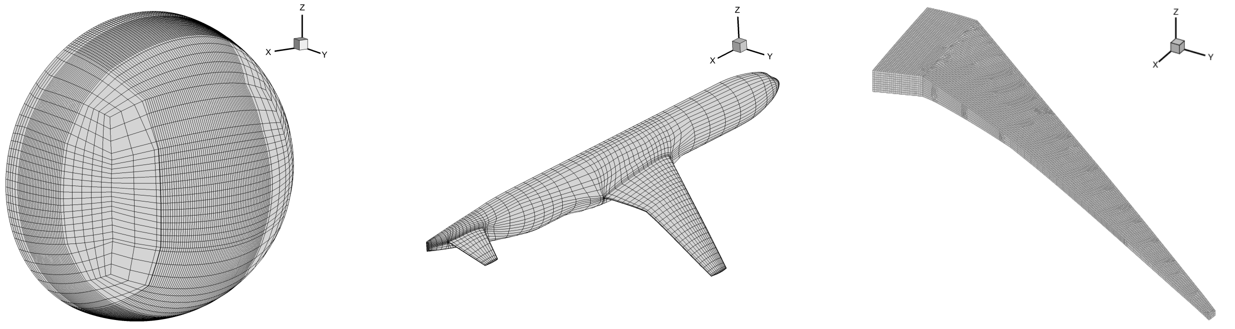
Table 1 uCRM-9 vehicle characteristics.

Parameter	Value	Units
Aspect Ratio	9	-
Span	58.76	m
MAC	7.01	m
Reference Area	383.74	m^2
1/4 Chord Sweep	35	deg
Taper Ratio	0.275	-

Table 2 uCRM-9 reference cruise data.

Quantity	Value	Units
Altitude	11,277.6	m
Mach	0.85	-
Re	43,130,072	-
α	2.014	deg
C_L	0.5	-
$\max(\delta_z)$	2.594	m

The structural model used for this study is identical to the publicly available coarse model. Note that only the wingbox of the half-aircraft is considered flexible, whereas the fuselage and horizontal tail members are considered rigid and are not modeled structurally. As such, they do not directly contribute to the loads applied to the flexible wingbox. Even though the coarse structural mesh is used presently, the discretization of this mesh is relatively fine and features are well resolved. When compared to the finest discretization available, a structural solution featuring a gravitational load showed a difference of only 0.05% between the maximum displacement in the coarse and fine meshes, giving confidence that the coarse structural mesh is well within the asymptotic range of convergence for a linear problem. The



(a) Farfield fluid volume mesh **(b) Surface fluid mesh** **(c) Wingbox structural mesh**

Fig. 1 Fluid and structural discretizations for the uCRM-9.

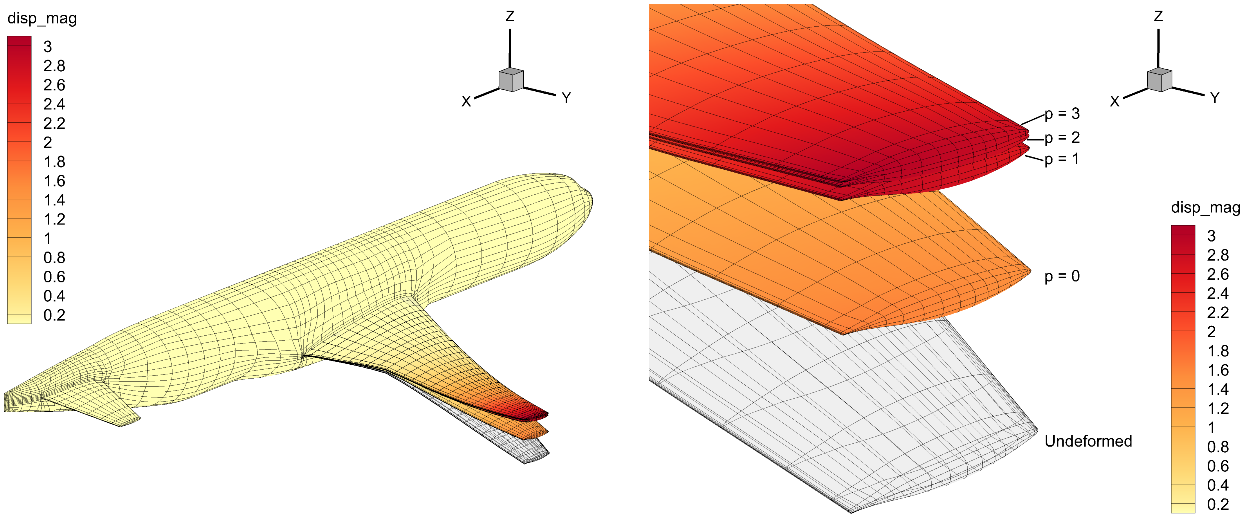
multi-block fluid mesh provided for the coarse fluid discretization featured nearly 300 cell blocks with 1.23 million cells in total. This mesh is not directly compatible with the mesh format required for the fluid solution with xflow. Therefore, the mesh was processed to adhere to a usable format, which removed duplicate nodes and merged cell blocks into a single element group. This mesh preprocessing also included a step to agglomerate linear elements into second-order, curved hexahedral elements on the vehicle boundaries. The final mesh used for the fluid solution is relatively coarse, particularly in the normal direction of the vehicle surface boundaries. Therefore, the final mesh is not well suited for the inclusion of viscous effects with a Reynolds-Averaged Navier-Stokes (RANS) solution given the lack of resolution in the boundary-layer discretization. Instead, an inviscid Euler solution is carried out, which leads to expected differences in the results when compared to the reference RANS solution and is discussed further below. Table 3 presents discretization details of both the coarse structural and coarse fluid meshes used in this study.

Table 3 uCRM-9 coarse mesh discretization information.

Mesh Type	No. Nodes	No. Elements
Fluid	161,726	72,532
Structural	23,886	25,129

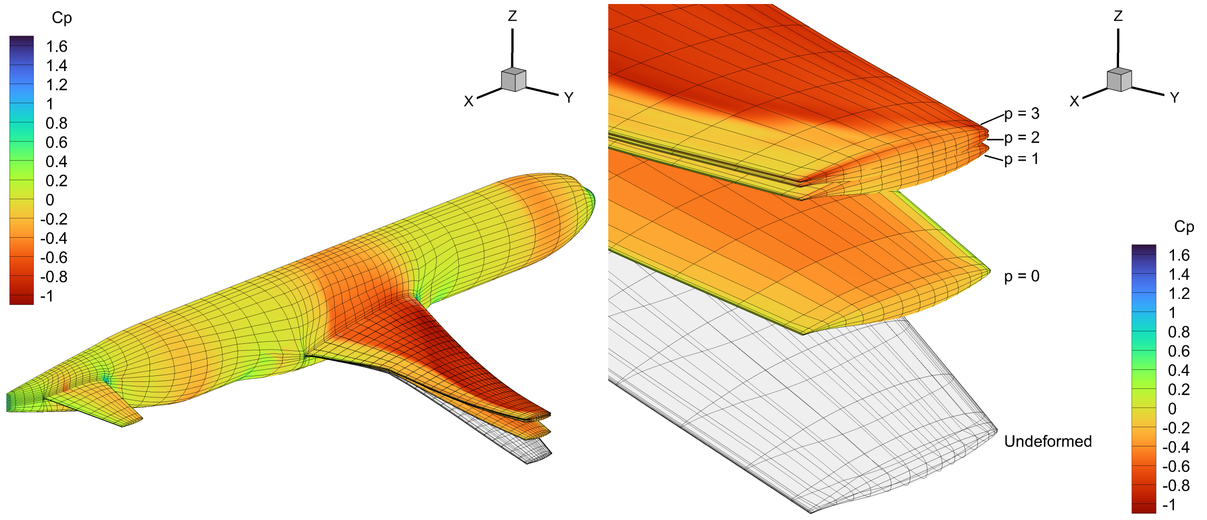
B. Coupled Response

The static coupling procedure in Algorithm 1 has been applied to the uCRM-9 at the reference cruise flight condition. Given the coarse discretization of the fluid domain, a spatial refinement study was conducted to achieve a true converged result with which a comparison to the reference data could be made. A uniform refinement study on the order of elemental polynomial basis functions, p, was conducted. This is one of the advantages of the DG finite-element formulation implemented in xflow, which easily allows for spatial refinement without the need for re-meshing. The order of approximation in all fluid elements was incremented from $p = 0$ to $p = 3$. Note that when constant element basis functions are used, i.e. $p = 0$, in a DG FEM, the solution degenerates to a first-order finite-volume method solution. The structural mesh was kept unchanged throughout each of the cases given its relatively fine discretization and agreement with the reference data when checked with a reference load case not shown here. Among each of the cases, the coupling parameters used to define the MELD transfer schemes were held constant along with the mesh motion parameters. This is due to the fact that both the fluid and structural mesh nodal information were not changing between cases. Table 4 summarizes the coupling, mesh motion parameters, and tolerances used for the coupled solutions. Note, however, that the metric used to check convergence of the fluid residual is an absolute tolerance. This absolute tolerance should be sized appropriately based on the size/scaling of the mesh geometry as well as the spatial order of approximation used in the solution. Therefore, while $\epsilon_f = 1 \cdot 10^{-5}$ was suitable for the low-order solutions on the given mesh/geometry, the convergence tolerance was increased for the high-order solutions.



(a) Vehicle **(b) Wingtip**

Fig. 2 TACS-xflow displacement magnitude convergence history for $p = 0, 1, 2, 3$.



(a) Vehicle **(b) Wingtip**

Fig. 3 TACS-xflow C_p convergence history for $p = 0, 1, 2, 3$.

Table 4 Coupling parameters used for uCRM-9 TACS-xflow solutions.

Parameter	Description	Value
β_{s2f}	Decay parameter for the MELD transfer from structural to fluid domain	0.5
N_{s2f}	Number of nearest structural nodes to link to each fluid surface node for the MELD transfer from structural to fluid domain	250
β_{f2s}	Decay parameter for the MELD transfer from fluid to structural domain	0.5
N_{f2s}	Number of nearest structural nodes to link to each fluid surface quadrature point for the MELD transfer from fluid to structural domain	250
-	Compact RBF to use for the mesh motion interpolation	C^2
r	Support radius (m) of the selected compact RBF	36
n_c	Number of fluid surface nodes (RBF centers) that define the mesh motion	~1000
ϵ_s	Relative tolerance used for the solution of structural states	$1 \cdot 10^{-12}$
ϵ_f	Absolute tolerance used for the solution of fluid states	$1 \cdot 10^{-5}$
ϵ_{FSI}	Absolute tolerance used for the coupled convergence criterion	$1 \cdot 10^{-3}$

In all four of the coupled solutions produced with the TACS-xflow framework, each coupled solution converged in 10 or fewer coupled static iterations for the specified tolerances. The time required for each coupled static iteration was dominated by the fluid solution, which scaled with the order of approximation used. Sample results of the final converged configurations can be seen in Fig. 2 and 3. The total coefficient of lift, C_L, and the maximum wingtip vertical deflection, $\max(\delta_Z)$, were the outputs selected to make comparisons against the reference uCRM-9 data. The TACS-xflow convergence study for these outputs is shown in Fig. 4, where they are compared to the reference data. In both the structural and fluid outputs shown, there is an over-prediction in the final converged values, corresponding to 13.6% error in $\max(\delta_Z)$, and 17.7% error in C_L. This discrepancy is expected given the difference in the physics modeled in xflow versus the reference solution. The lack of viscosity in the Euler solution of xflow leads to a fluid solution that lacks a boundary layer and related boundary layer effects. This difference is made clear by comparing the coefficient of pressure, C_p, from the TACS-xflow Euler solution to the RANS reference at four span-wise locations along the wing, as shown in Fig. 5. The sectional views indicate that the Euler solution predicts the shock further towards the trailing edge of the upper surface of the wing compared to the RANS solution, leading to an increased suction on the upper surface of the wing and an overall higher lift condition. The inclusion of viscous effects produces a boundary layer, which can act as an effective thickness added to the wing profile along the chord direction of the section. This additional thickness is likely to move the shock closer to the leading edge of the wing, which would lead to an increase in total drag and a decrease in total lift. The decrease in lift would then lead to a decrease in the wing displacement and both outputs being compared would show a reduction in error with the reference. Therefore, the error in the current comparisons is most likely attributed to the discrepancy in the physics being modelled and is unlikely a result of the coupling procedure implemented in the TACS-xflow framework. Given the expected differences in physics being modelled, the overall agreement between the reference and TACS-xflow is acceptable.

C. Mesh Refinement

For the selected test case, it has been demonstrated here that the overall accuracy of the coupled outputs is highly sensitive to the accuracy of the fluid-domain solution. The current linear structural solution is computationally inexpensive and even a finely-discretized structural domain can be solved for in a fraction of the time required for the fluid-domain solution. Likewise, the transfer of data across the fluid-structure interface is on the same order of expense as the structural solution. Therefore, in practice, a convergence study for the coupled problem should prioritize the refinement of the fluid-domain solution when complex and nonlinear flow features like shocks are present like in the test case used presently. Given the computational expense of the present fluid-domain solution, the refinement should be done in a way to achieve the greatest improvement in the output of interest for the least amount of additional computational effort. For a finite-element solution, spatial refinement is accomplished either through increasing the spatial order of the element basis functions, p-refinement (shown above), or through a reduction in element size, h-refinement. An advantage of the DG formulation is that p-refinement can be applied on a per-element basis given that state continuity between adjacent elements is not strictly enforced. This feature makes DG finite elements capable of

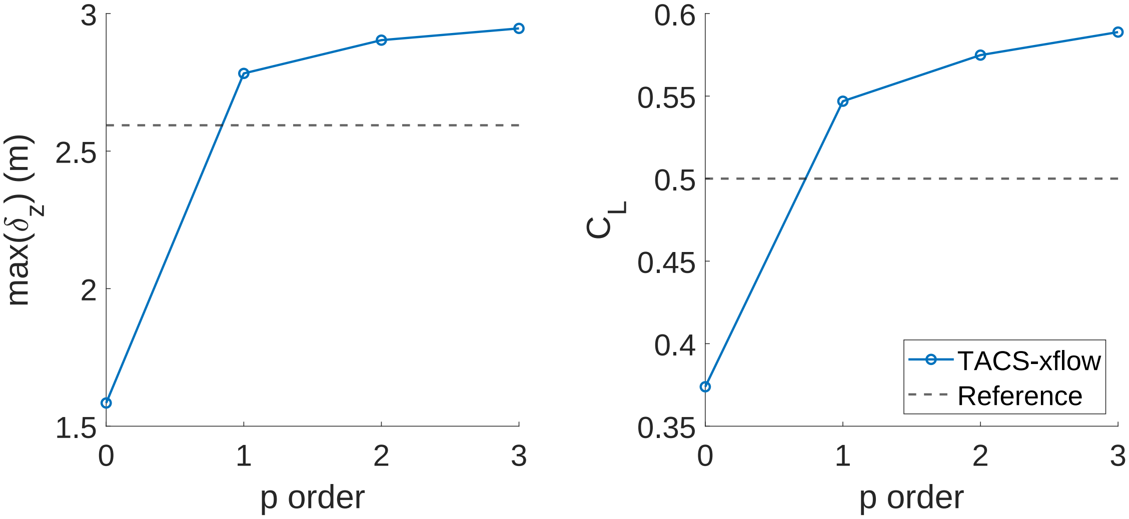


Fig. 4 TACS-xflow p-refined coupled static convergence for max(δ_Z) (left) and C_L (right) shows slight overpredictions due to lack of viscous effects.

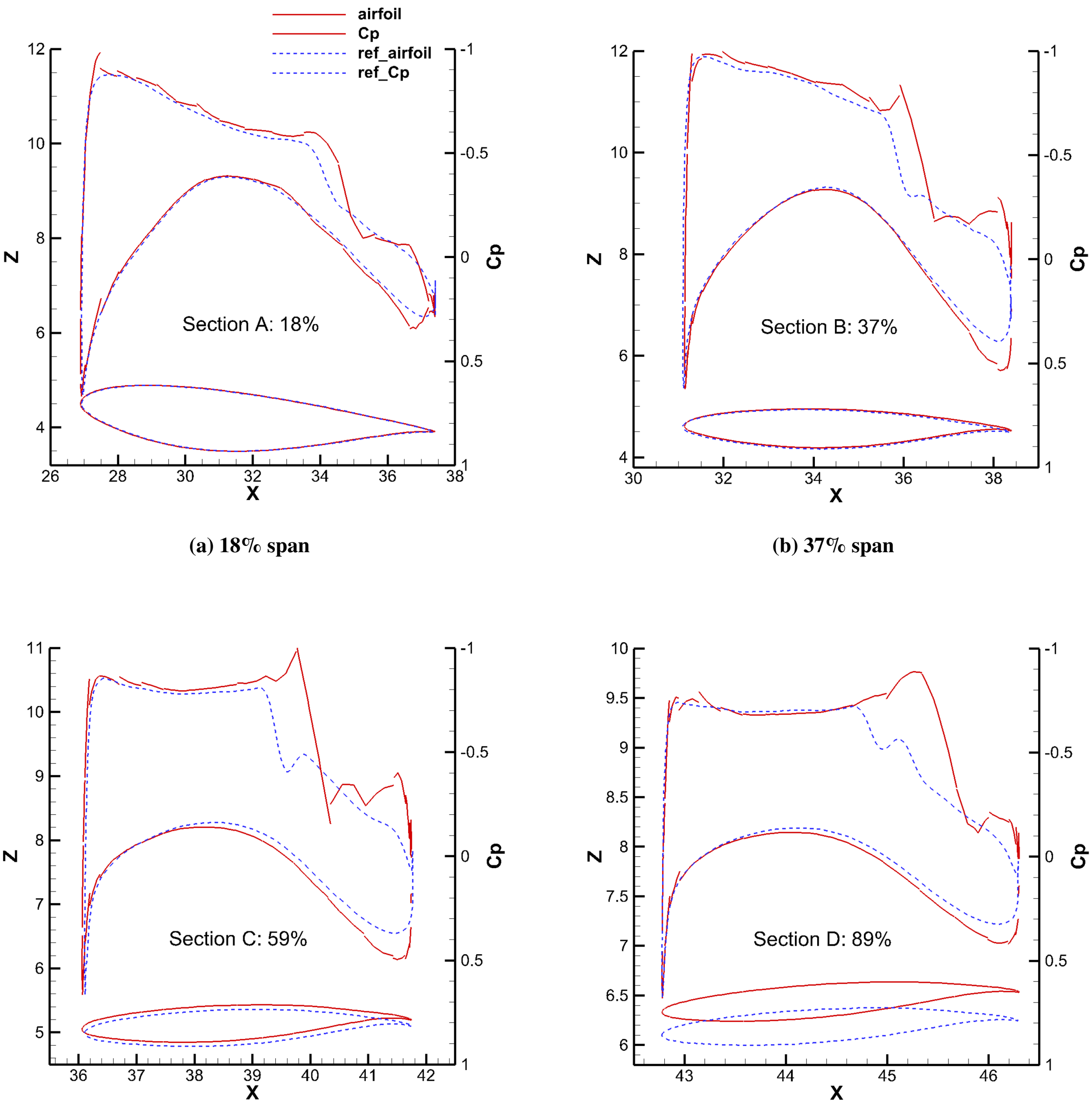
applying output-based adaptation techniques to their full effect.

The availability of two refinement methods raises the question as to which method is more effective at improving accuracy in the coupled solution. To demonstrate the difference between the two, we consider the fluid-only Euler solution to the uCRM-9 cruise condition about its undeformed configuration. Given that the fluid solution about the undeformed vehicle is simply the first iteration to the coupled problem, assessing the convergence behavior of the fluid solution about this undeformed configuration informs the convergence behavior of the entire coupled problem as well. Starting with the initial coarse fluid mesh used above, uniform p- and h- refinement will be applied to the mesh and the C_L output will be compared between all cases. Uniform p-refinement is applied to the uCRM-9 as done previously, where the mesh size is held constant for each case. Hanging-node h-refinement is applied to uniformly decrease the element size in the mesh. Given the hexahedral element shape used presently, this process isotropically splits all elements in the mesh into eight smaller elements. Table 5 summarizes the resulting fluid discretizations being compared, where the p-level is the order of polynomial used for element integration, and the h-level is the number of times hanging-node refinement has been applied to the original coarse mesh described in Table 3.

Table 5 uCRM-9 fluid mesh refinement information.

Mesh Discretization	No. Nodes	No. Elements	No. DOF
p0, h0	161,726	72,532	72,532
p1, h0	161,726	72,532	290,128
p2, h0	161,726	72,532	725,320
p3, h0	161,726	72,532	1,450,640
p0, h1	1,085,726	580,256	580,256
p1, h1	1,085,726	580,256	2,321,024

Fig. 6a shows the convergence history for the C_L output for each of the fluid solutions. For the original coarse mesh and a given level of accuracy in the C_L output, uniform p-refinement requires fewer degrees of freedom compared to uniform h-refinement. However, the number of degrees of freedom in the problem is not the only factor that influences computational expense. Fig. 6b shows the wall time comparison between the two approaches, normalized by the time of the p0, h0 solution. With high-order approximations, numerical conditioning and stability can cause difficulties in the nonlinear solution process and necessitate the use of additional numerical techniques to aid in producing a valid



(a) 18% span

(b) 37% span

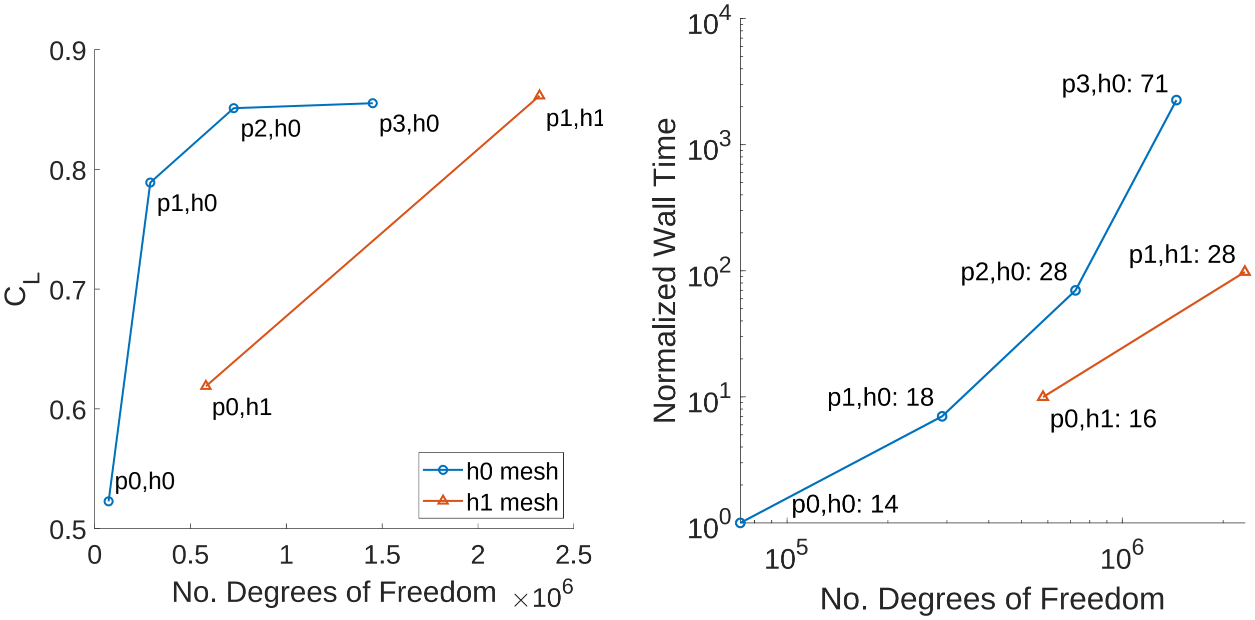
(c) 59% span

(d) 89% span

Fig. 5 C_p **comparisons between the reference (dashed blue) and the TACS-xflow** $p = 2$ **coupled solution (red) show a clear difference in shock location on the upper surface of the wing between viscous and inviscid solutions.**

solution. This process is implementation-dependent and can greatly impact computation time. As indicated in Fig. 6b, the number of nonlinear iterations required by the fluid solver increased with each increment in p, leading to the increase in computational time. Therefore, the p3 ,h0 solution took more time to complete compared to the p1, h1 solution, even though the high-order solution involved fewer total degrees of freedom. However, further improvements in the algorithmic approach taken for the numerical solution to this problem could potentially alleviate this issue.

For a finite element solution, uniform refinement is guaranteed to converge to the truth solution in the limit of small element size or high approximation order. However, in practice, the solution error is generally sensitive to a small number of elements that approximate specific local features of the solution in the domain. Refining in these areas alone



(a) p-refinement vs. h-refinement

(b) wall time comparison, label indicates number of nonlinear iterations required for convergence

Fig. 6 p-refinement on a coarse mesh achieves a greater accuracy in the output for fewer degrees of freedom when compared to uniform h-refinement, but at a cost of longer compute time.

greatly improves the accuracy of the approximate solution with only a small increase in computational expense. However, it is difficult to know a priori how and where to refine within the domain, especially for coupled multidisciplinary systems with various sources of error, whether it's discretization error, mesh motion error, etc. Output-based mesh adaptation automates this process, producing the optimal mesh refinement that minimizes discretization error subject to a constraint on the increase of computational expense for a given output. In the future, coupled output-based mesh adaptation will be pursued within the current FSI computational framework.

V. Concluding Remarks

In this paper, a partitioned computational FSI framework was described and implemented. A DG FEM discretization was applied to an ALE formulation of the Navier-Stokes equations and was coupled with a FEM discretization of the linear elastic structural equations. The MELD method was employed to efficiently and robustly transfer the loads and displacements between each discipline, and the mesh motion within the fluid domain interior was handled by RBF interpolation. The block Gauss-Seidel static coupling iterative procedure is described and was applied to a large-scale three-dimensional test case featuring the uCRM-9 aircraft at a cruise condition, which is verified against reference data. Finally, a mesh refinement study was carried out which demonstrates the trade off between accuracy and computation time between p- and h-refinement strategies. In future work, we seek to improve the coupled framework by expanding it to unsteady aeroelastic problems, as well as implementing coupled adjoint capabilities. Ultimately we seek to implement both steady and unsteady output-based mesh adaptation 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.

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. Phil Beran and Dr. Nathan Wukie for the techncial exchanges. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

References

[1] Bisplinghoff, R. L., Ashley, H., and Halfman, R. L., *Aeroelasticity*, Courier Corporation, 2013.

[2] Hou, G., Wang, J., and Layton, A., "Numerical methods for fluid-structure interaction—a review," *Communications in Computational Physics*, Vol. 12, No. 2, 2012, pp. 337–377.

[3] Hübner, B., Walhorn, E., and Dinkler, D., "A monolithic approach to fluid–structure interaction using space–time finite elements," *Computer methods in applied mechanics and engineering*, Vol. 193, No. 23-26, 2004, pp. 2087–2104.

[4] Piperno, S., Farhat, C., and Larrouturou, B., "Partitioned procedures for the transient solution of coupled aroelastic problems Part I: Model problem, theory and two-dimensional application," *Computer methods in applied mechanics and engineering*, Vol. 124, No. 1-2, 1995, pp. 79–112.

[5] Brown, S., and Brown, S., "Displacement extrapolations for CFD+ CSM aeroelastic analysis," *38th Structures, Structural Dynamics, and Materials Conference*, 1997, p. 1090.

[6] Farhat, C., Lesoinne, M., and Le Tallec, P., "Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity," *Computer methods in applied mechanics and engineering*, Vol. 157, No. 1-2, 1998, pp. 95–114.

[7] 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.

[8] Kiviaho, J., and Kennedy, G. J., "Efficient and Robust Load and Displacement Transfer Scheme Using Weighted Least-Squares," *AIAA Journal*, Vol. 57, No. 5, 2019. https://doi.org/10.2514/1.J057318, URL http://gkennedy.gatech.edu/wp-content/uploads/2019/03/technical_note.pdf.

[9] Donea, J., Giuliani, S., and Halleux, J.-P., "An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions," *Computer methods in applied mechanics and engineering*, Vol. 33, No. 1-3, 1982, pp. 689–723.

[10] Persson, P.-O., Bonet, J., and Peraire, J., "Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 17-20, 2009, pp. 1585–1595.

[11] Budd, C. J., Huang, W., and Russell, R. D., "Adaptivity with moving grids," *Acta Numerica*, Vol. 18, 2009, pp. 111–241.

[12] Selim, M., Koomullil, R., et al., "Mesh deformation approaches–a survey," *Journal of Physical Mathematics*, Vol. 7, No. 2, 2016.

[13] Batina, J. T., "Unsteady Euler airfoil solutions using unstructured dynamic meshes," *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1381–1388.

[14] 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.

[15] Witteveen, J., "Explicit and robust inverse distance weighting mesh deformation for CFD," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010, p. 165.

[16] 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, No. 1, 2013, pp. 468–494. https://doi.org/10.1016/j.jcp.2013.06.007.

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

[18] Reed, W. H., and Hill, T., "Triangular mesh methods for the neutron transport equation," Tech. rep., Los Alamos Scientific Lab., N. Mex.(USA), 1973.

[19] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "p-Multigrid solution of high–order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 207, 2005, pp. 92–113. https://doi.org/10.1016/j.jcp.2005.01.005.

[20] 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.

[21] Boopathy, K., and Kennedy, G. J., "Parallel Finite Element Framework for Rotorcraft Multibody Dynamics and Discrete Adjoint Sensitivities," *AIAA Journal*, Vol. 57, No. 8, 2019, pp. 3159–3172.

[22] Kennedy, G., and Martins, J., "A comparison of metallic and composite aircraft wings using aerostructural design optimization," *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012, p. 5475.

[23] 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.

[24] 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.

[25] 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.

[26] Smith, M. J., Cesnik, C. E. S., Hodges, D. H., and Moran, K., "An evaluation of computational algorithms to interface between CFD and CSD methodologies," *37th Structure, Structural Dynamics and Materials Conference*, 1996, p. 1400.

[27] Smith, M. J., Hodges, D. H., and Cesnik, C. E. S., "Evaluation of Computational Algorithms Suitable for Fluid-Structure Interactions," *Journal of Aircraft*, Vol. 37, No. 2, 2000, pp. 282–294. https://doi.org/10.2514/2.2592.

[28] de Boer, A., van Zuijlen, A. H., and Bijl, H., "Review of coupling methods for non-matching meshes," *Computer methods in applied mechanics and engineering*, Vol. 196, No. 8, 2007, pp. 1515–1525.

[29] De Boer, A., Van der Schoot, M., and Bijl, H., "Mesh deformation based on radial basis function interpolation," *Computers & structures*, Vol. 85, No. 11-14, 2007, pp. 784–795.

[30] 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.