AIAA SciTech Forum
8-12 January 2024, Orlando, FL
AIAA SCITECH 2024 Forum

10.2514/6.2024-2411

Enforcing Work Conservation in Modular Aeroelastic Coupling For Multidisciplinary Design Optimization

Sebastiaan P. C. van Schie *, Michael A. P. Warner †, Andrew H. Fletcher ‡, John T. Hwang §
University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093

There is significant interest in solving Multidisciplinary Design Analysis and Optimization (MDAO) problems, in which models of various disciplines are used simultaneously. As mentioned in NASA's CFD Vision 2030 report this has often happened at the cost of significant software development efforts and one-off model coupling implementations. This work investigates a methodology for modular aeroelastic coupling for use in MDAO problems by using a Solver-Independent Field Representation (SIFR) as intermediate configuration between solid and fluid solvers. Each solver is integrated into SIFR through a standardized approach that allows for force consistency and aeroelastic work conservation. The aim of leveraging such an approach for aeroelasticity is to significantly reduce the development and implementation cost of coupled aeroelastic analysis for MDAO. In this work we implement a coupled aeroelastic approach through SIFR and use it to carry out static simulations of an aircraft wing. We derive various sets of algebraic constraints for work conservation that can inform model developers and analysts when defining coupled aeroelastic analyses. We apply the coupling methodology to a Vortex Lattice Method model and FEniCS-based Finite Element and carry out coupled simulations of an aircraft wing, in which it's shown that we can indeed obtain force consistency and work conservation with SIFR.

I. Introduction

Large-scale Multidisciplinary Design Analysis & Optimization (MDAO) has gained significant interest over the last decade, to a large extent because of the popularization of and support for platforms that support the calculation of derivatives for gradient-based design optimization. These include the seminal paper by Martins and Hwang on unification of the various methods for derivative computations [1] and NASA's open-source OpenMDAO platform [2]. Various toolkits have been developed to carry out gradient-based aerostructural MDAO, as have various solvers that allow for derivative computations. Low-to-mid-fidelity examples include SHARPy [3], OpenAeroStruct [4] and UM/NAST [5], while MACH [6] and FUNtoFEM [7] are examples of high-fidelity implementations. We highlight the recent work on MPhys*, a multiphysics extension of the OpenMDAO toolkit that focuses on providing standardized software interfaces between high-fidelity physics solvers. As far as we are aware these platforms all directly couple fluid and solid solvers to each other, which can make it cumbersome to change either solver due to the lack of universal interface. Furthermore, as mentioned in NASA's CFD Vision 2030 report [8], there is a clear need for standardizing CFD solver interfaces and integration in multidisciplinary analysis problems.

The aim of this work is to establish a modular solver-independent aeroelastic coupling methodology for use in large-scale MDAO problems that conserves forces (also known as 'force consistency') and aeroelastic work between solvers. We leverage the approach that was proposed by Van Schie et al. in [9]. Although Van Schie et al. present the foundation of a general approach with fluid and solid solvers interacting through an intermediate, solver-independent reference configuration, the presented implementation examples still directly couple the fluid and solid solvers. Anderson et al. in [10] use the same approach for dynamic aeroelasticity, with a Vortex Particle Method implementation coupled to a FEniCS-based Reissner–Mindlin shell model.

In this work we generalize the aforementioned aeroelastic coupling approach that was introduced in [9]. We use the Solver-Independent Field Representation (SIFR) for integrating and coupling solvers, in a modular fashion, that

*Ph.D. Student, Department of Mechanical and Aerospace Engineering, AIAA student member
†Ph.D. Student, Department of Mechanical and Aerospace Engineering, AIAA student member
‡Ph.D. Student, Department of Mechanical and Aerospace Engineering, AIAA student member
§Assistant Professor, Department of Mechanical and Aerospace Engineering, AIAA senior member
*`https://openmdao.github.io/mphys/`

compute physically meaningful field quantities. The SIFR methodology is presented in a companion paper [11]. Here, we focus on the conservation of force and energy in aeroelastic coupling using the SIFR methodology.

We have two specific contributions. First, we present a principled approach for constructing the solver coupling maps within the SIFR methodology. Second, we investigate the effects of enforcing aeroelastic work conservation on the predictions of a coupled simulation of an eVTOL aircraft wing. Therefore, whereas this work focuses on aeroelastic coupling in the context of SIFR, the work by Warner et al. [11] focuses on the more general application of SIFR to MDAO problems for representing both the system state and material parameters. The solver-independent aeroelasticity methods presented here are implemented in the CADDEE framework [12] and are planned to be used in future works as well.

The structure of this work is as follows. Sec. II gives a short overview of aeroelastic coupling. A brief literature survey is included in order to contextualize the contribution of this work, after which the algebraic conditions for consistent and conservative static aeroelastic coupling methods are introduced. Sec. III then applies these methods to computational frameworks that use SIFR and details the possibilities and constraints for using such an aeroelastic coupling methodology. Lastly Sec. IV.D applies this methodology to a coupled simulation of an eVTOL aircraft wing; we detail the coupling implementation that is used for both fluid and solid solvers, after which we compare the predictions of a non-conservative and conservative coupling approach.

II. Computational Aeroelastic Coupling

Solving Fluid-Structure Interaction (FSI) problems entails solving coupled fluid and solid subproblems, each on their own (sub)domain. Loads exerted by the fluid on the solid structure result in structural deformations, which in turn affect the fluid's flow field and hence the fluid's loads on the structure. In computational environments this feedback loop manifests itself through two transfer mechanisms:

1) Fluid loads need to be transferred to the (boundary of the) solid domain;
2) Solid deformations need to be transferred to the boundary of the fluid domain.

A variety of approaches exist for resolving this two-way coupling. In this work we focus on coupling solid and fluid solvers in a partitioned way, in which linear(ized) systems of solid and fluid equations are solved separately. This in contrast to monolithic coupling, in which a single linear(ized) system is set up that simultaneously solves the coupled fluid and solid subproblems. In this way monolithic approaches impose strict compatibility of the solid & fluid deformations and aerodynamic loads. While this is an attractive way of resolving the coupled nature of aeroelastic problems, having to assemble a single coupled system of equations that computes both fluid and solid solutions necessitates low-level solver access. It moreover requires a significant amount of effort to change either solver and is restrictive in the types of solvers that can be coupled. Within the current context solver modularity is paramount, and hence we instead opt for a partitioned approach. We also focus on coupled simulations in which the fluid and solid meshes do not match along their respective boundaries. Solutions computed with a monolithic approach or with matching meshes can be regarded as special cases of this more general partitioned and non-matching treatment. We include a brief literature survey in Sec. II.A, after which we go over the algebraic requirements for force consistency and aeroelastic work conservation in Sec. II.B.

A. Coupling Approaches in Literature

Since the mathematical models that govern solid and fluid continua are generally conservation laws, their corresponding numerical solvers are often constructed to retain (parts of) this conservation behavior in their discrete solutions. This motivates to construct the displacement and load data transfer maps D and L such that they retain one or more of such invariants as well. For aeroelasticity two relevant invariants can be identified:

- Total aerodynamic force (in each principal direction)
- Aeroelastic work (i.e. 'force times distance')

The use of these invariants is ubiquitous throughout the relevant literature. An advantage of approaches that are explicitly constructed with such conservation properties in mind is that they generalize to non-matching fluid and solid domains. This is relevant for both low- and high-fidelity coupled solvers due to possibly different approaches for domain discretization. Retaining the force invariants is generally referred to as 'force consistency', whereas retaining the work invariant is called 'conservation'.

Maman and Farhat et al. [13, 14] focus on constructing suitable coupling methods that have consistent loads and conserve aeroelastic work under mappings between non-matching domains. They also analyze the relevance of this with simplified problems. They focus on projection-based methods, in which quantities on the fluid and solid meshes are related to one another through (orthogonal) projection. This however requires combining information of the connectivity of both meshes, which is invasive and requires access to detailed mesh information. Interpolation-based methods take an alternative approach, where fields on one mesh are sampled in various locations before being interpolated onto the other mesh. Examples of works that introduce and apply such methods include Smith et al. [15], Beckert and Wendland [16] and Rendall and Allen [17]. While this does not require detailed mesh information, the resulting interpolation problem needs to be solved, which could scale poorly with problem size and requires a well-conditioned interpolation matrix. De Boer et al. [18, 19] provide an overview and compare the performance of various projection and interpolation methods when applied to exact and quasi-1D problems. Gerstenberger and Wall [20] propose to decouple the motion of the fluid mesh boundary from the fluid and solid problems and enforce the interaction conditions between the fluid and solid with Lagrange multipliers. However, they don't explicitly consider work conservation.

Brown [21] proposed using rigid links to map displacements and forces between fluid and solid boundaries in a way that leads to force consistency and aeroelastic work conservation. This approach has been used in various works throughout the years, such as by Kennedy [22] and Kenway [23], both with Martins. A more recent contribution is the Matching-based Extrapolation of Moments and Loads (MELD) by Kiviaho and Kennedy [24], which also retains force consistency and work conservation.

B. Consistency and Conservation in Aeroelastic Coupling

We consider static aeroelastic coupling cases. In order to keep the treatment of aeroelastic coupling as general as possible we consider the (discretized) solid and fluid solvers as general functions. Let $\boldsymbol{u}$ and $\boldsymbol{w}$ denote displacement and force fields. We denote the solid solver as $\mathcal{S} : \mathbb{R}^{n_{s'}} \times \mathbb{R}^{m_{s'}} \longrightarrow \mathbb{R}^{n_{s'}}$ with $\mathcal{S} = \mathcal{S}(\boldsymbol{u}, \boldsymbol{w})$, whereas the fluid solver is denoted as $\mathcal{A} : \mathbb{R}^{n_{a'}} \times \mathbb{R}^{m_{a'}} \longrightarrow \mathbb{R}^{m_{a'}}$ with $\mathcal{A} = \mathcal{A}(\boldsymbol{u}, \boldsymbol{w})$. Here $n_{s'}$, $m_{s'}$ and $n_{a'}$, $m_{a'}$ denote the number of degrees of freedom in the solid and fluid solvers respectively, n and m reflect the numbers of displacement and force degrees of freedom. Note that both solvers depend on their state variables as well (i.e. $\mathcal{S}$ depends on $\boldsymbol{u}_{s'}$ and $\mathcal{A}$ on $\boldsymbol{w}_{a'}$). By doing this we can include nonlinear models into our analysis.

Solving the static aeroelastic problem amounts to finding a consistent solution that satisfies the following partitioned solid-fluid system of equations:

$$\begin{aligned}
\boldsymbol{u}_{s'} &= \mathcal{S}(\boldsymbol{u}_{s'}, \boldsymbol{w}_{s'}) \\
\boldsymbol{w}_{a'} &= \mathcal{A}(\boldsymbol{u}_{a'}, \boldsymbol{w}_{a'}) \\
\boldsymbol{u}_{a'} &= D\boldsymbol{u}_{s'} \\
\boldsymbol{w}_{s'} &= L\boldsymbol{w}_{a'}
\end{aligned} \tag{1}$$

Where $D : \mathbb{R}^{n_{s'}} \longrightarrow \mathbb{R}^{n_{a'}}$ and $L : \mathbb{R}^{m_{a'}} \longrightarrow \mathbb{R}^{m_{s'}}$ are linear maps that transform the outputs of each solver into an appropriate input for the other solver. This is because the fluid and solid solvers will in general have different interpretations for their degrees of freedom. These arise from (among other things) possible differences in their numerical discretization approaches, mesh sizes and domain non-conformity. Instead of using linear maps, they can be constructed as linearizations of some underlying nonlinear projection as well. For the sake of simplicity we focus here on linear maps, without loss of generality.

Eq. (1) can be solved in an iterative fashion. Fixed-point iteration is the usual approach for partitioned systems owing to its ease of implementation and minimal invasiveness, which is especially relevant when dealing with commercial or otherwise closed-source solvers. Such an iterative solution approach is necessary even when both coupled solvers are linear, due to their implicit dependence on each other's outputs. Fig. 1 shows what fixed-point iteration for this system looks like: We start with an initial displacement estimate $\boldsymbol{u}_{a'}^{(0)}$ (and for nonlinear fluid models an initial force vector $\boldsymbol{w}_{a'}^{(0)}$) and compute $\boldsymbol{w}_{a'}^{(1)}$. This force vector is then mapped to the solid solver with L and used to compute $\boldsymbol{u}_{s'}^{(1)}$. That solid displacement is mapped to the fluid solver with D, after which the fixed-point cycle starts again. Iterations are carried out until a predefined convergence tolerance is reached. The norm of the difference between successive iterations of any quantity can be used to measure convergence of the fixed-point iteration approach.

$$
\begin{tikzcd}
 & \boldsymbol{w}_{s'}^{(1)} \arrow[r, bend left, "\mathcal{S}"] & \boldsymbol{u}_{s'}^{(1)} \arrow[d, "D"] & \boldsymbol{w}_{s'}^{(2)} \arrow[r, bend left, "\mathcal{S}"] & \boldsymbol{u}_{s'}^{(2)} \arrow[d] & & \boldsymbol{w}_{s'}^{(k)} \arrow[r, bend left, "\mathcal{S}"] & \boldsymbol{u}_{s'}^{(k)} \\
\boldsymbol{u}_{a'}^{(0)} \arrow[r, bend right, "\mathcal{A}"'] & \boldsymbol{w}_{a'}^{(1)} \arrow[u, "L"] & \boldsymbol{u}_{a'}^{(1)} \arrow[r, bend right, "\mathcal{A}"'] & \boldsymbol{w}_{a'}^{(2)} \arrow[u, "L"] & \cdots \arrow[r, bend right] & \boldsymbol{u}_{a'}^{(k-1)} \arrow[r, bend right, "\mathcal{A}"'] & \boldsymbol{w}_{a'}^{(k)} \arrow[u, "L"] &
\end{tikzcd}
$$

Fig. 1 Static fixed-point iteration for coupled fluid solver $\mathcal{A}$ and solid solver $\mathcal{S}$

Solid and fluid solvers $\mathcal{S}$ and $\mathcal{A}$ are provided at runtime; we make no further assumptions on their specific structure. This leaves open the question of how to define the linear maps D and L that convert displacement and force between both solvers. We do this in a solver-independent way and recall here the approach that was covered in [9]. That approach forms a generalization of the approach of De Boer et al. [18, 19]. We first look at force consistency, after which comes aeroelastic work conservation.

1. Force consistency

Enforcing total aerodynamic force consistency involves only L, $\boldsymbol{w}_{s'}$ and $\boldsymbol{w}_{a'}$. Suppose that we explicitly separate the components of force vectors $\boldsymbol{w}_{s'}$ and $\boldsymbol{w}_{a'}$ into separate columns, one corresponding to each principal/coordinate direction:

$$
\begin{aligned}
\boldsymbol{w}_{s'} &= \begin{bmatrix} \boldsymbol{w}_{s'}^{(x)} & \boldsymbol{w}_{s'}^{(y)} & \boldsymbol{w}_{s'}^{(z)} \end{bmatrix} \\
\boldsymbol{w}_{a'} &= \begin{bmatrix} \boldsymbol{w}_{a'}^{(x)} & \boldsymbol{w}_{a'}^{(y)} & \boldsymbol{w}_{a'}^{(z)} \end{bmatrix}
\end{aligned}
$$

We proceed the following discussion with $\boldsymbol{w}_{s'}^{(x)}$ and $\boldsymbol{w}_{a'}^{(x)}$, and assume that the other force vector components are expressed in the same bases, meaning that the conclusions drawn generalize to them as well. The entries of vectors $\boldsymbol{w}_{s'}^{(x)}$ and $\boldsymbol{w}_{a'}^{(x)}$ are associated to corresponding basis elements, no matter whether Finite Element, Finite Volume, Finite Difference or other discretization approaches are used. The only requirement is that the basis elements are functions of spatial coordinates. We can then write the expansions of $\boldsymbol{w}_{s'}^{(x)}$ and $\boldsymbol{w}_{a'}^{(x)}$ as follows:

$$
\begin{aligned}
w_{s'}^{(x)} &= \sum_{i=1}^{m_{s'}^{(x)}} w_{s',i}^{(x)} \phi_i \\
w_{a'}^{(x)} &= \sum_{j=1}^{m_{a'}^{(x)}} w_{a',j}^{(x)} \psi_j
\end{aligned}
$$

Where (with some abuse of notation) $w_{s'}^{(x)}$, $w_{a'}^{(x)}$ denote the expansions of the force vectors $\boldsymbol{w}_{s'}^{(x)}$, $\boldsymbol{w}_{a'}^{(x)}$, and $\{\phi_i\}_{i=1}^{m_{s'}^{(x)}}$, $\{\psi_j\}_{j=1}^{m_{a'}^{(x)}}$ are their corresponding sets of basis functions. For Finite Element solvers these correspond to the basis in which each force vector is expanded, whereas they are piecewise constant over each cell in Finite Volume solvers. In case the entries of the corresponding vector $\boldsymbol{w}_{s'}$ or $\boldsymbol{w}_{a'}$ are nodal/pointwise force vectors, they are Dirac delta functions centered around the point of force application. Force consistency in x-direction then dictates that:

$$
\int_{\Omega_{\mathcal{S}}} w_{s'}^{(x)} \, d\Omega = \sum_{i=1}^{m_{s'}^{(x)}} w_{s',i}^{(x)} \int_{\Omega_{\mathcal{S}}} \phi_i \, d\Omega = \sum_{j=1}^{m_{a'}^{(x)}} w_{a',j}^{(x)} \int_{\Omega_{\mathcal{A}}} \psi_j \, d\Omega = \int_{\Omega_{\mathcal{A}}} w_{a'}^{(x)} \, d\Omega
$$

We can write this as two vector products as well, with the integrals denoting elementwise integration of the basis function vectors:

$$
\sum_{i=1}^{m_{s'}^{(x)}} w_{s',i}^{(x)} \int_{\Omega_{\mathcal{S}}} \phi_i \, d\Omega = \left(\boldsymbol{w}_{s'}^{(x)}\right)^T \int_{\Omega_{\mathcal{S}}} \boldsymbol{\phi} \, d\Omega = \left(\boldsymbol{w}_{a'}^{(x)}\right)^T \int_{\Omega_{\mathcal{A}}} \boldsymbol{\psi} \, d\Omega = \sum_{j=1}^{m_{a'}^{(x)}} w_{a',j}^{(x)} \int_{\Omega_{\mathcal{A}}} \psi_j \, d\Omega \tag{2}
$$

Where $\Omega_{\mathcal{S}}, \Omega_{\mathcal{A}}$ are the discretized domains of the solid and fluid solver respectively. Recall from Eq. (1) that $\boldsymbol{w}_{s'} = L\boldsymbol{w}_{a'}$. Specifically for the force vector components in x-direction, $\boldsymbol{w}_{s'}^{(x)} = L^{(x)}\boldsymbol{w}_{a'}^{(x)}$, such that:

$$w_{s'}^{(x)} = \sum_{i=1}^{m_{s'}^{(x)}} w_{s',i}^{(x)}\phi_i = \sum_{i=1}^{m_{s'}^{(x)}} \phi_i \underbrace{\sum_{j=1}^{m_{a'}^{(x)}} L_{ij}^{(x)} w_{a',j}^{(x)}}_{=w_{s',i}^{(x)}}$$

Where $L_{ij}^{(x)}$ are the entries of $L^{(x)}$. We apply this mapped formulation of $w_{s'}^{(x)}$ to Eq. (2) and reorder the terms. This shows that:

$$\left(\boldsymbol{w}_{a'}^{(x)}\right)^T \left(L^{(x)}\right)^T \int_{\Omega_{\mathcal{S}}} \boldsymbol{\phi}\ d\Omega = \left(\boldsymbol{w}_{a'}^{(x)}\right)^T \int_{\Omega_{\mathcal{A}}} \boldsymbol{\psi}\ d\Omega$$

From which the following equivalent condition can be derived, if this is to hold for all vectors $\boldsymbol{w}_{a'}^{(x)}$:

$$\left(\boldsymbol{w}_{a'}^{(x)}\right)^T \left[\left(L^{(x)}\right)^T \int_{\Omega_{\mathcal{S}}} \boldsymbol{\phi}\ d\Omega - \int_{\Omega_{\mathcal{A}}} \boldsymbol{\psi}\ d\Omega\right] = 0 \tag{3}$$

We conclude that force consistency implies that the entries of $L^{(x)}$ must satisfy the condition posed by Eq. (3). In summation notation the same condition for each ψ_i is:

$$\int_{\Omega_{\mathcal{A}}} \psi_i\ d\Omega = \sum_{j=1}^{m_{s'}^{(x)}} L_{ji}^{(x)} \int_{\Omega_{\mathcal{S}}} \phi_j\ d\Omega \tag{4}$$

To clarify this condition we give an example: Suppose that $w_{a'}^{(x)}$ and $w_{s'}^{(x)}$ are expressed in nodal bases. Since all basis functions are Dirac delta functions this implies that:

$$\int_{\Omega_{\mathcal{S}}} \phi_i\ d\Omega = 1\ \forall\ i \in 1, 2, \ldots, m_{s'}^{(x)}$$
$$\int_{\Omega_{\mathcal{A}}} \psi_i\ d\Omega = 1\ \forall\ i \in 1, 2, \ldots, m_{a'}^{(x)}$$

It follows from Eq. (4) that:

$$\sum_{j=1}^{m_{s'}^{(x)}} L_{ji}^{(x)} = 1 \tag{5}$$

In other words: Each column sum of $L^{(x)}$ must be equal to 1, and thus each entry of $\boldsymbol{w}_{a}^{(x)}$ is partitioned exactly over the entries of $\boldsymbol{w}_{s'}^{(x)}$. A more general formulation of $L^{(x)}$ that satisfies Eq. (4) for arbitrary sets of basis functions can be given as:

$$L_{ji}^{(x)} = \alpha_{ji} \frac{\int_{\Omega_{\mathcal{A}}} \psi_i\ d\Omega}{\int_{\Omega_{\mathcal{S}}} \phi_j\ d\Omega} \tag{6}$$

$$L_{ij} = \alpha_{ij} \frac{\int_{\Omega_{\mathcal{A}}} \psi_j\ d\Omega}{\int_{\Omega_{\mathcal{S}}} \phi_i\ d\Omega} \qquad \sum_i \alpha_{ij} = 1\ \forall\ i \in 1, 2, \ldots, m_{s'}^{(x)} \tag{7}$$

$$(D_{a'a})_{ij} = \frac{f_\epsilon(r_{ij})}{\sum_k f_\epsilon(r_{ik})}$$

$$(L_{aa'})_{ij} = \alpha_{ij} \int b_j$$

$$\alpha_{ij} = \frac{f_\epsilon(r_{ij})}{\sum_k f_\epsilon(r_{kj})}$$

Where the coefficients α_{ji} can be chosen arbitrarily or according to some (physics-motivated) selection criterion, as long as they satisfy the following condition:

$$\sum_j \alpha_{ji} = 1 \;\forall\; j \in 1, 2, \ldots, m_{s'}^{(x)}$$

Note that this is consistent with the example given earlier, in which the integral of each basis function ϕ_j, ψ_i was equal to 1. Constructing $L^{(x)}$ according to Eq. (6) results in a force map that retains consistency in x-direction. If we circle back to the assumption made at the start of this section (the same bases being used in each direction) we can use $L^{(x)} = L$ and project the forces in each direction in a consistent manner through a straightforward matrix multiplication:

$$\boldsymbol{w}_{s'} = \begin{bmatrix} \boldsymbol{w}_{s'}^{(x)} & \boldsymbol{w}_{s'}^{(y)} & \boldsymbol{w}_{s'}^{(z)} \end{bmatrix} = L \begin{bmatrix} \boldsymbol{w}_{a'}^{(x)} & \boldsymbol{w}_{a'}^{(y)} & \boldsymbol{w}_{a'}^{(z)} \end{bmatrix} = L\boldsymbol{w}_{a'}$$

2. Aeroelastic work conservation

Whereas force consistency involves only the force vectors, aeroelastic work conservation involves both $\boldsymbol{u}$ and $\boldsymbol{w}$. Work W can be defined in a general (continuous) field as the integrated dot product of (continuous) displacement $\boldsymbol{u}$ and force $\boldsymbol{w}$:

$$W = \int_\Omega \boldsymbol{u} \cdot \boldsymbol{w} \; d\Omega = \int_\Omega (\boldsymbol{u}, \boldsymbol{w}) \; d\Omega \tag{8}$$

An analogous definition in discretized settings can be given as follows:

$$W = \mathrm{tr}\left(\boldsymbol{u}^T N \boldsymbol{w}\right) \tag{9}$$

Where $\boldsymbol{u}$, $\boldsymbol{w}$ are again arrays of degrees of freedom, arranged such that each column corresponds to one principal direction. The trace operator selects precisely the diagonal components of the resulting matrix, i.e. those that correspond to the dot product in Eq. (8). We call N an 'invariant matrix', which encodes the integral in Eq. (8). It serves a role similar to inner product matrices and contains similar basis function expansions to those that were used in the previous discussion of force consistency. Note that in Eq. (9) we assume that the same discrete representations are used for each component of $\boldsymbol{u}$ and $\boldsymbol{w}$. In general N is solver- and discretization-agnostic and, as mentioned, depends on the discrete representations that are used for $\boldsymbol{u}$ and $\boldsymbol{w}$. Some examples of this are given in [9]. Conservation of aeroelastic work can be obtained in discrete settings by defining Eq. (9) for each solver and equating the resulting discrete inner products:

$$W_{\mathcal{S}} = \mathrm{tr}\left(\boldsymbol{u}_{s'}^T N_{\mathcal{S}} \boldsymbol{w}_{s'}\right) = \mathrm{tr}\left(\boldsymbol{u}_{a'}^T N_{\mathcal{A}} \boldsymbol{w}_{a'} = W_{\mathcal{A}}\right)$$

$N_{\mathcal{S}}$ and $N_{\mathcal{A}}$ are the solver-specific invariant matrices of the solid and fluid solvers. We recall the linear maps given in Eq. (1) that link the displacements and forces in both solvers, and apply these:

$$\mathrm{tr}\left(\boldsymbol{u}_{s'}^T N_{\mathcal{S}} \boldsymbol{w}_{s'}\right) = \mathrm{tr}\left(\boldsymbol{u}_{s'}^T N_{\mathcal{S}} L \boldsymbol{w}_{a'}\right) = \mathrm{tr}\left(\boldsymbol{u}_{s'}^T D^T N_{\mathcal{A}} \boldsymbol{w}_{a'}\right) = \mathrm{tr}\left(\boldsymbol{u}_{a'}^T N_{\mathcal{A}} \boldsymbol{w}_{a'}\right)$$

Which results in the following relation when moving all terms to one side:

$$\mathrm{tr}\left(\boldsymbol{u}_{s'}^T \left[N_{\mathcal{S}} L - D^T N_{\mathcal{A}}\right] \boldsymbol{w}_{a'}\right) = 0 \tag{10}$$

Since we want Eq. (10) to hold for all vectors $\boldsymbol{u}_{s'}$ and $\boldsymbol{w}_{a'}$, we thus require that a transpose relationship exists between both linear maps, with the exact relationship depending on the inner product matrices of both solvers:

$$N_{\mathcal{S}} L = D^T N_{\mathcal{A}} \tag{11}$$

With this we can define one of the maps and impose conservation of aeroelastic work to derive the other linear map. We have two resulting options, depending on which map is defined:

- Deriving L from the definition of D gives:

$$L = \underbrace{\left(N_{\mathcal{S}}^T N_{\mathcal{S}}\right)^{-1} N_{\mathcal{S}}^T}_{=N_{\mathcal{S}}^\dagger} D^T N_{\mathcal{A}} \tag{12}$$

- Whereas deriving D from a definition of L results in the following:

$$D^T = N_{\mathcal{S}} L \underbrace{N_{\mathcal{A}}^T \left(N_{\mathcal{A}} N_{\mathcal{A}}^T\right)^{-1}}_{=N_{\mathcal{A}}^{\ddagger}} \Longrightarrow D = \left(N_{\mathcal{A}}^{\ddagger}\right)^T L^T N_{\mathcal{S}}^T \tag{13}$$

Where the superscripts † and ‡ respectively denote the left and right Moore-Penrose pseudo-inverses. Note that defining the pseudo-inverses in this way is only valid if $\left(N_{\mathcal{S}}^T N_{\mathcal{S}}\right)^{-1}$ and $\left(N_{\mathcal{A}} N_{\mathcal{A}}^T\right)^{-1}$ exist, i.e. when the products $N_{\mathcal{S}}^T N_{\mathcal{S}}$ and $N_{\mathcal{A}} N_{\mathcal{A}}^T$ are of full rank. This can only happen when $N_{\mathcal{S}}$ has more rows than columns, i.e. $n_{s'} \geq m_{s'}$, and when $N_{\mathcal{A}}$ has more columns than rows, i.e. $m_{a'} \geq n_{a'}$.

In case either of these requirements are not satisfied the corresponding pseudo-inverses can be computed through the Singular Value Decomposition (SVD). For example, suppose that $N_{\mathcal{S}} = U\Sigma V^T$ is a valid SVD with the singular values σ_i on the diagonal of Σ. Then $N_{\mathcal{S}}^{\dagger} = V\Sigma^{\dagger} U$, with $\Sigma^{\dagger}$ the transpose of Σ with the reciprocal non-zero singular values on its diagonal:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \sigma_p & 0 \end{bmatrix} \Longrightarrow \Sigma^{\dagger} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{\sigma_p} \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

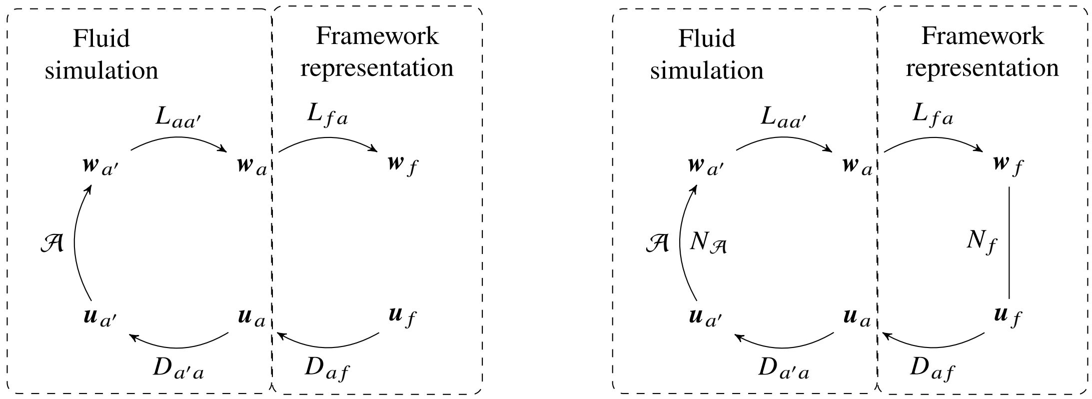
Unlike before we do not have $N_{\mathcal{S}}^{\dagger} N_{\mathcal{S}} = I$ in this case. Using Eq. (12) as $L = N_{\mathcal{S}}^{\dagger} D^T N_{\mathcal{A}}$ then leads to a projection map L that does not exactly conserve aeroelastic work. Lastly we point out that pseudo-inverses are sensitive to the condition number of a matrix similarly to the regular matrix inverse.

III. Solver-Independent Data Transfer

Sec. II covered algebraic conditions for consistent and conservative static aeroelastic coupling approaches. Next we show how these can be combined with a Solver-Independent Field Representation to result in a suitable modular aeroelastic coupling approach for use in single- and multi-fidelity MDO environments. Warner et al. [11] give more details about the implementation of this solver-independent state and how it can be used for geometry and material parameterization in MDAO problems. The aforementioned displacement and force maps D and L are modified to allow for indirect coupling of the fluid and solid solvers. We first cover the solver-independent data transfer approach for a single solver in Sec. III.A. Following this Sec. III.C covers how force consistency and work conservation can be retained for aeroelasticity problems that use the SIFR methodology.

A. Solver-Independent Field Representation and Individual Solver Coupling

Following Warner et al. [11], we define the Solver-Independent Field Representation (SIFR) as a geometry and state description that is independent of either solver. We refer to SIFR as the ‘framework’ representation. SIFR defines the geometry as named sets of B-spline surfaces and volumes, on which state and solution fields can be defined. Both continuous fields (discretized as B-splines) and discrete fields (as sets of nodal values) are supported. In this way SIFR can be used to evaluate optimization objective functions and constraints without relying on individual solvers. Each solver exchanges data with only SIFR. Couplings between solvers are thus handled indirectly, with each solver exchanging their inputs and outputs with SIFR instead of with one another directly. Intermediate states between SIFR and each solver are defined (separately for each solver) for each field in order to facilitate straightforward framework-solver data transfer. We will refer to these representations as the Outer Mold Line (OML) representations, even though they are defined on internal components as well. In the OML configuration each state and field is represented by a set of point values. Integrating a solver into SIFR requires defining the projection maps that convert these sets of point values to the solver's inputs and outputs. Fig. 2a shows an example of this methodology for integrating fluid solver $\mathcal{A}$ into SIFR. The displacement and force maps D_{af} and L_{fa} are internal to the framework and relate its SIFR to the solver-specific displacement and force OML point sets $\boldsymbol{u}_a$ and $\boldsymbol{w}_a$. Solver-specific projection maps $D_{a'a}$ and $L_{aa'}$ have to be defined in order to establish the coupling between solver $\mathcal{A}$ and the OML representations.



(a) Integrating fluid solver $\mathcal{A}$ into SIFR as covered by Warner et al. [11]

(b) Including solver and framework invariant matrices to enforce aeroelastic work conservation

Fig. 2 Integrating fluid solver $\mathcal{A}$ into SIFR with and without considering aeroelastic work conservation

B. Consistent & Conservative Individual Solver Coupling

Now we apply to the coupling introduced in the previous section the concepts that were used in Sec. II in order to establish force consistency and aeroelastic work conservation. For this we introduce the fluid solver invariant matrix $N_{\mathcal{A}}$ and the framework invariant matrix N_f. Since the derivation here is largely identical to that presented in Sec. II.B, we can reuse earlier results. In particular, we insert into Eq. (11) the linear maps shown in Fig. 2b that link fluid solver $\mathcal{A}$ to the framework representation:

$$N_f \underbrace{L_{fa} L_{aa'}}_{=L_a} = \underbrace{D_{af}^T D_{a'a}^T}_{=D_a^T} N_{\mathcal{A}} \tag{14}$$

The linear maps L_a, D_a have thus each been broken up into a composition of two maps. Conserving aeroelastic work under these maps implies that one of the following two things can be done:

1) Three of the total of four maps can be specified, with the fourth following from Eq. (14);
2) One of the compositions of two maps in Eq. (14) can be computed based on the definition of the other two maps.

This gives six options in total per solver. Regardless of which map or composition of maps is chosen to follow from Eq. (14), a closed-form solution for it can be derived on paper. The choice for a specific option can be made with various considerations in mind if exact aeroelastic work conservation is desired:

- Condition numbers of the projection maps and invariant matrices;
- Shapes of the projection maps and invariant matrices;
- Ease of implementation;
- Solver structure.

The first of the two points above follow from the need to compute Moore-Penrose pseudo-inverses. For example, suppose that we construct L_{fa} such that aeroelastic work is conserved. Then from Eq. (14) we see that:

$$L_{fa} = N_f^{\dagger} D_{af}^T D_{a'a}^T N_{\mathcal{A}} L_{aa'}^{\ddagger}$$

N_f and $L_{aa'}$ need to be reasonably well-conditioned in order for their pseudo-inverses to be constructed accurately. Moreover, we only obtain exact aeroelastic work conservation if $N_f^T N_f$ and $L_{aa'} L_{aa'}^T$ are of full rank, i.e. if $n_f \geq m_f$ and $m_{a'} \geq m_a$.

As mentioned before, any one of the four projection maps or their compositions (L_a, D_a) can be determined from Eq. (14) when the other maps are defined. Table 1 gives an overview for each option of which pseudo-inverses need to be computed and which matrix shape constraints must be satisfied in order to have exact conservation of aeroelastic work.

C. Consistent & Conservative Modular Aeroelastic Coupling

We now couple solid solver $\mathcal{S}$ to the framework representation as well. This results in the coupling approach shown in Fig. 3. Introducing a framework configuration between the fluid and solid solvers does not negatively affect

Computed map	**Pseudo-inverses**	**Shape constraints**
L_{fa}	N_f $L_{aa'}$	$n_f \geq m_f$ $m_{a'} \geq m_a$
$L_{aa'}$	$N_f L_{fa}$	$n_f \geq m_a$
D_{af}	$N_{\mathcal{A}}^T D_{a'a}$	$m_{a'} \geq n_a$
$D_{a'a}$	D_{af} $N_{\mathcal{A}}^T$	$n_f \geq n_a$ $m_{a'} \geq n_{a'}$
L_a	N_f	$n_f \geq m_f$
D_a	$N_{\mathcal{A}}^T$	$m_{a'} \geq n_{a'}$

Table 1 Required pseudo-inverses and exact conservation shape constraints for each map that can be computed through Eq. (14) **once the other maps are defined**

the coupling methodology's consistency and conservation properties, as long as the coupling schemes between the framework and both solvers are consistent and conservative. This was already shown by Van Schie et al. in [9]. The data streams between the framework and either solver can be considered entirely independent from one another. This facilitates the modularity of the proposed coupling methodology: Either solver can be changed without directly affecting the other.

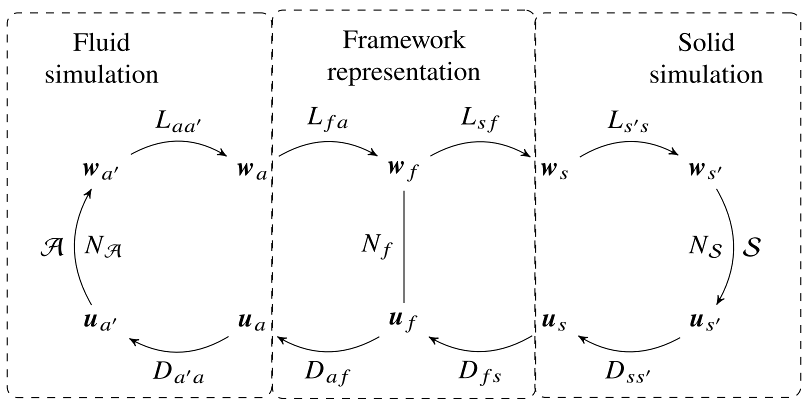


Fig. 3 Solver-Independent Field Representation as intermediate state between fluid and solid solvers

We can derive the equivalent of Eq. (14) for the solid solver-framework coupling:

$$N_{\mathcal{S}} \underbrace{L_{s's} L_{sf}}_{=L_s} = \underbrace{D_{ss'}^T D_{fs}^T}_{=D_s^T} N_f \tag{15}$$

IV. Application to Passenger Air Vehicle

We apply the consistent and conservative modular aeroelastic coupling methodology to a realistic test model. This work was done in collaboration with Aurora Flight Sciences, leveraging their expertise in eVTOL vehicles. A key aspect of this collaboration has been the development of a test case model inspired by Aurora Flight Sciences' Passenger Air Vehicle (PAV). It is important to note that our model does not replicate the actual PAV, but rather draws inspiration from it. To ensure the protection of Aurora Flight Sciences' proprietary information, our approach has been to utilize publicly available data and images of the PAV as a foundation for creating an analogous yet distinct model. This approach allows us to explore design and optimization strategies while respecting intellectual property considerations. A detailed description of the PAV is given in [25].

Computed map	Pseudo-inverses	Shape constraints
D_{fs}	N_f^T $D_{ss'}$	$m_f \geq n_f$ $n_s \geq n_{s'}$
$D_{ss'}$	$N_f^T D_{fs}$	$n_s \geq m_f$
L_{sf}	$N_S L_{s's}$	$m_s \geq n_{s'}$
$L_{s's}$	N_S L_{sf}	$m_{s'} \geq n_{s'}$ $m_f \geq m_s$
D_s	N_f^T	$m_f \geq n_f$
L_s	N_S	$m_{s'} \geq n_{s'}$

Table 2 Required pseudo-inverses and exact conservation shape constraints for each map that can be computed through Eq. (15) **once the other maps are defined**

In this work we focus on simulating its main wing. We start by describing the framework representation in Sec. IV.A. Following that we introduce the fluid solver and the coupling approaches that connects it to the framework representation in Sec. IV.B. Section IV.C covers the same topics for the solid solver, after which the coupled simulation results are given in Sec. IV.D for various conditions. Note that for both solvers we first introduce a coupling approach that does not conserve aeroelastic work. Afterwards those approaches are modified to allow for work conservation.

A. Framework Representation

In the framework representation we represent the aerodynamic loads as point forces, whereas the solid displacements are represented as B-splines. In Fig. 3 the map D_{af} (which map displacements from the framework to OML representations) is a straightforward evaluation, while the displacement map D_{fs} into the framework consists of a B-spline interpolation. The force maps into (L_{fa}) and out of (L_{sf}) the framework both use inverse distance weighting to map between sets of nodes. A brief general description of such approaches is given in Sec. IV.B, for a more detailed description the interested reader is referred to [11]. Since the aerodynamic loads in the framework representation are modeled as point forces, invariant matrix N_f corresponds to the evaluation map of the displacement B-spline basis functions in each force node. We note here that technically we are using two invariant matrices for the framework representation: One that covers displacements and forces on both wings, and one that only covers the left wing. The first of these is used to exchange data with the fluid solver, whereas the latter is used to interface with the solid solver. Grids of 10 by 10 degrees of freedom are used to represent quantities on each surface present in the framework representation, over a total of 32 surfaces for the entire wing and internal wing structure.

B. Fluid Solver and Coupling Approach

We use a Vortex Lattice Method (VLM) solver to compute an aerodynamic force distribution over the wing. VLM represents the wing geometry as a set of panels that cover the wing camber plane. Its displacement inputs thus corresponds to the motion of the corner points of these panels. Similarly the aerodynamic forces are modeled as a set of point force vectors, with one force vector per panel. For more in-depth information on VLM solvers the interested reader is referred to the textbook by Plotkin & Katz [26]. Sarojini et al. verified the accuracy of this VLM implementation in [25]. We first establish a non-conservative coupling approach. For this we define the displacement and force projection maps $D_{a'a}$ and $L_{aa'}$ that relate the solver inputs and outputs to the OML representations of nodal displacement and force values $\boldsymbol{u}_a$ and $\boldsymbol{w}_a$, as indicated in Fig. 2a and Fig. 3. For both displacement and force we use an inverse distance weighting approach. This approach is covered in [9], we briefly repeat it here for convenience. We point out that various elements of this inverse distance weighting procedure are used in the MELD load and displacement transfer scheme [24] as well.

We start by defining a set of target nodes and a set of source nodes in physical space. As their names imply the resulting map will project from the source nodes to the target nodes. We compute the distances between each target node and the nodes of its corresponding source representation. Denote the physical distance between solver target node

i and source node j with d_{ij}. Then we plug this distance into weight function $f(d_{ij})$. This can be any function whose value is monotonically decreasing as input d_{ij} increases, such as most function with negative exponents. In this work we use a weighted Gaussian function:

$$f_\epsilon(d_{ij}) = ||\Omega_i|| \exp\left(-(\epsilon d_{ij})^2\right)$$

Where ϵ is a sensitivity parameter that defines the domain of dependence of a given target node. $||\phi_i||$ is the integral of the basis function associated with target node i. Since in this case we're working with nodal values this integral is equal to 1 for all i; this is not the case later in this work. The weights g_{ij} of the inverse distance weighting map can then be computed by dividing each $f_\epsilon(d_{ij})$ by summing the weights $f_\epsilon(d_{ij})$ over i:

$$g_{ij} = \frac{f_\epsilon(d_{ij})}{\sum_k f_\epsilon(d_{kj})} \tag{16}$$

Using the g_{ij} as matrix entries of a linear map, we note that the column sums of said matrix are all equal to 1, which is in line with the derivation of force consistency in Sec. II.B. For displacement map $D_{a'a}$ the source nodes are the OML displacement nodes and the target nodes are the VLM panel corners. For force map $L_{aa'}$ the source nodes are the VLM force nodes on the panels, while the target nodes are the OML force nodes. This fully defines the non-conservative integration VLM into SIFR, since the maps L_{fa} and D_{af} between the OML nodes and the framework representation are solver-independent. The resulting coupling scheme is shown in Fig. 4a.

Now we derive a conservative approach that couples the VLM solver to SIFR. The VLM invariant matrix $N_{\mathcal{A}}$ is the only extra piece of solver information that is needed to implement our consistent and conservative coupling approach. As is covered in more detail in [9] the fluid invariant matrix is formulated as the linear interpolation matrix that projects the forces to the displacement nodes as a partition of unity. The force node on each panel is located in the middle of the panel at 25% of its chord length, and hence we project 37.5% of each force vector to the front two corners of the corresponding panel, and 12.5% to each of the rear two corners. Encoding this in a matrix then gives our invariant matrix $N_{\mathcal{A}}$.

We are left to determine which projection map(s) will be constructed from work conservation relation Eq. (14). It was found that the aforementioned force and displacement projection maps are ill-conditioned, thus we want to avoid having to use their pseudo-inverses. Taking a look at table 1 we see that this means that either the composite displacement map D_a or composite force map L_a (which directly map between VLM and framework representation) should be inferred from Eq. (14). However, for the mesh of the VLM solver it will never happen that $m_{a'} \geq n_{a'}$, since there will always be more panel corner nodes (up to four unique corner nodes per panel) than force nodes (always one per panel). This leaves the possibility of constructing L_a such that aeroelastic work is conserved. We rewrite Eq. (14) to find that:

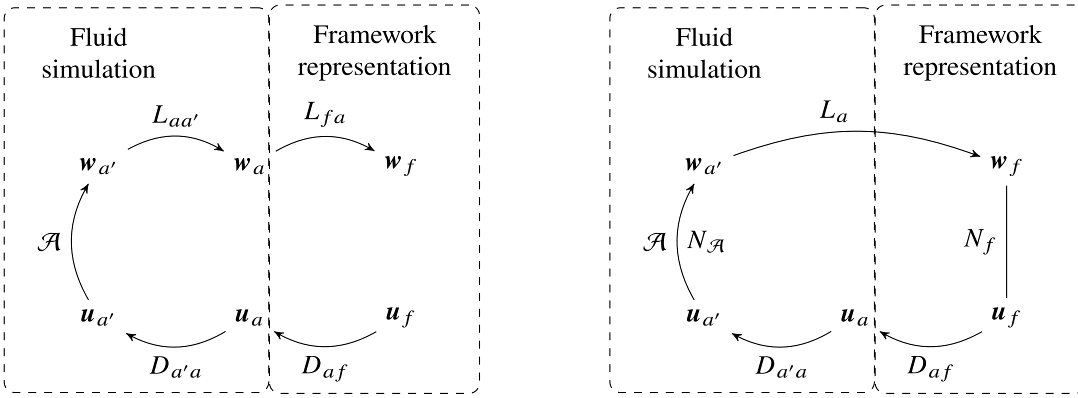
$$L_a = N_f^\dagger D_{af}^T D_{a'a}^T N_{\mathcal{A}}$$

Which then defines our conservative aeroelastic coupling approach for the VLM solver.

C. Solid Solver and Coupling Approach

A Finite Element shell model is used to compute the displacements of the PAV wing due to the aerodynamic loads predicted by the VLM model. We use a static implementation of a linear Reissner–Mindlin shell model through the Finite Elements in PDE-constrained Multidisciplinary Optimization (FEMO) toolkit [27]. FEMO is based on the FEniCSx Finite Element platform [28]. This shell model implementation has been used before for two-way static aeroelastic coupling in [9]. Furthermore, Sarojini et al. [25] compare its predictions with various other shell and beam models under an applied 3g aerodynamic load (without two-way coupling) for the PAV wing. Figure 5 shows the PAV wing box model and its internal rib structure as used in the solid solver. The force inputs are modeled as a linear vector-valued function on the wing box model, while the displacements are expressed in a vector-valued quadratic function basis. The mesh consists of 2,303 vertices and 2,472 elements. Also present in the Reissner–Mindlin shell model are local rotations. In the current implementation these are not used in the coupling model of the solid solver.

We again start by establishing the non-conservative coupling approach, of which a diagram is shown in Fig. 6a. The maps L_{sf} and D_{fs} that connect the framework representation to the sets of displacement and force OML nodes are as covered in Sec. IV.A, leaving us to define the maps that connect quantities in the OML nodes to the solver inputs and outputs. We note that there is a one-to-one correspondence between the nodes of the solid mesh and the



(a) Non-conservative coupling **(b) Conservative coupling approach**

Fig. 4 Conservative and non-conservative approaches for connecting the fluid solver to the framework representation

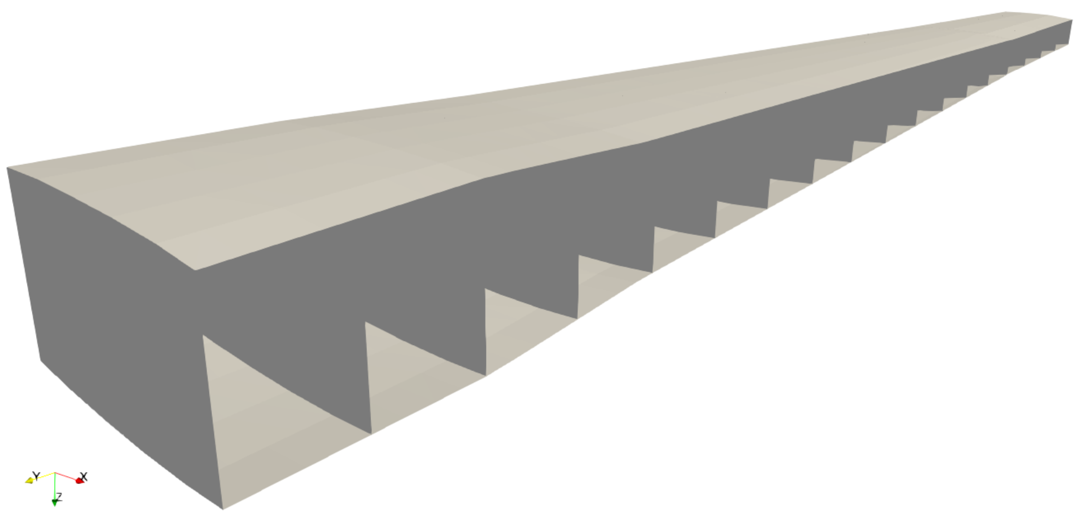


Fig. 5 Cutaway view of PAV wing box geometry used in the solid solver

linear basis functions in which the force is expressed in the solid solver. We associate each force basis function with a specific mesh node, which then allows us to use the g_{ij} in Eq. (16) to construct force map $L_{s's}$. In this case the $||\phi_i||$ correspond to the integrals of force basis function ϕ_i. This defines $L_{s's}$, leaving us to construct displacement map $D_{ss'}$. The solid displacements are expressed in a quadratic function basis. We want to use an approach that is similar to how $L_{s's}$ is constructed, and hence we define $D_{ss'}$ as a composition of two maps: First we interpolate the quadratic displacement functions to a linear function basis (to establish an association with the mesh nodes), after which we use the matrix entries from Eq. (16). Suppose that we denote the aforementioned interpolation matrix with Q and the matrix constructed from Eq. (16) with G, then $D_{ss'} = GQ$.

Next we define the conservative solid solver coupling approach. We start by defining solid invariant matrix $N_{\mathcal{S}}$. One way in which this can be done is to model $N_{\mathcal{S}}$ by numerically integrating products of the quadratic displacement basis functions and linear force basis functions. Taking a look at table 2 we can see that this would prevent us from being able to compute force map L_s through Eq. (15), since $n_{s'} > m_{s'}$. Instead we follow the approach that was already used in [9] and define $N_{\mathcal{S}}$ as a composition of two matrices: The left matrix Q^T projects the quadratic displacements to the linear function basis, while the right matrix M is constructed by integrating all combinations of the linear basis functions (identical to mass matrices in regular Finite Element methods). This gives us a workaround for the shape constraint $m_{s'} \geq n_{s'}$: Since in Eq. (15) both invariant matrix $N_{\mathcal{S}}$ and displacement map $D_{ss'}^T$ contain the matrix Q^T on their left side, we can remove this interpolation term from Eq. (15). This is shown in Eq. (17) below. The result of this is that we do not have to compute the (pseudo-)inverse of the composition $N_{\mathcal{S}} = Q^T M$; instead we can just invert M, which is well-conditioned and symmetric. We note that defining $N_{\mathcal{S}}$ in this way is a modeling choice. Alternative definitions are possible, depending on the shape and conditioning constraints that need to be satisfied.

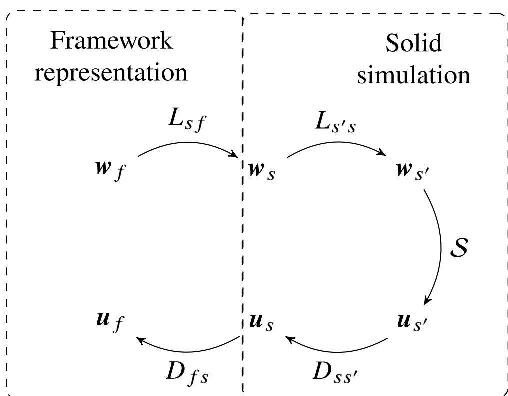
As was the case for the fluid solver coupling, we can compute any of the (compositions of) projection maps with Eq. (15) in order to enforce aeroelastic work conservation between the framework and the solid solver. Again the force and displacement maps are ill-conditioned, we thus avoid (pseudo-)inverting them. From table 2 it follows that either D_s or L_s can be computed with Eq. (15). We elect in this work to compute L_s. Alternatively D_s can be computed; this would avoid the issues mentioned above with the shape constraints on $N_{\mathcal{S}}$, since N_f^T would instead be (pseudo-)inverted. Recall that $D_{ss'} = GQ$ and $N_{\mathcal{S}} = Q^T M$. We first restate Eq. (15) with these two definitions applied:

$$Q^T M L_s = Q^T G^T D_{fs}^T N_f \tag{17}$$

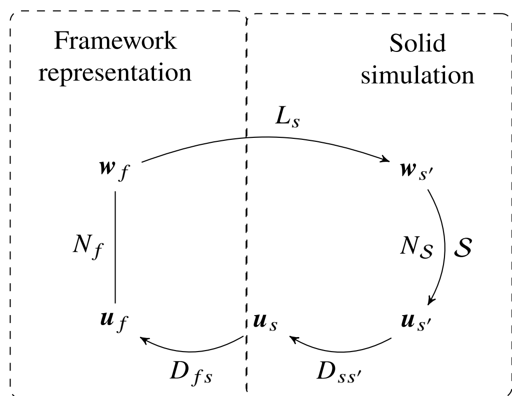
Eliminating Q^T from both sides and rearranging gives:

$$L_s = M^{-1} G^T D_{fs}^T N_f$$

Which is the force map that conserves aeroelastic energy when mapping between the solid solver and the framework; it maps directly from the framework to the solid solver, without explicitly projecting to the OML force nodes.



(a) Non-conservative coupling



(b) Conservative coupling

Fig. 6 Conservative and non-conservative approaches for connecting the solid solver to the framework representation

D. Coupled Simulation Results

We run a series of coupled aeroelastic simulations of the main wing of the PAV while varying the pitch angle. A fixed-point iteration approach is used, with the fluid and models being run in an alternating way until the maximum difference between successive iterations is smaller than 10^{-12}. As mentioned earlier, detailed geometric and material information of the PAV is given in [25].

1. Forces

Fig. 7 shows the lift and induced drag forces (and corresponding force coefficients) obtained with the conservative and non-conservative coupling approaches. Also shown are the forces computed with a rigid wing. As can be seen the the three sets of results are nearly visually indistinguishable. Only at higher pitch angles does the rigid wing produce slightly more lift, due to its lack of wing bending. This happens because the wing is too stiff for aeroelastic effects to have a noticeable impact on its behavior at these flow conditions: As will be covered later the tip displacements at the highest pitch angles are only around 1% of the wing's semi-span. The difference between the two aeroelastic

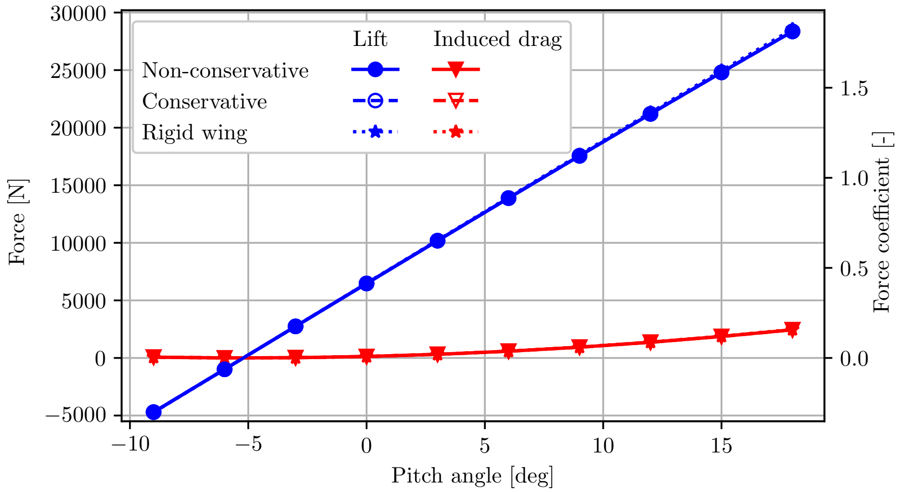


Fig. 7 Lift and induced drag (and associated coefficients) versus pitch angle for the two coupling approaches and a rigid wing

coupling approaches is between 1 and 10 Newtons for all pitch angles. In Fig. 8 we see that both coupling approach have (approximate) force consistency, with the maximum non-consistency being $O(10^{-12})$. Shown are the 2-norms differences of the total force vectors when mapping between configurations, normalized with the 2-norm of the total framework force vector.

2. Aeroelastic Work

Next we look at the (non-)conservation of aeroelastic work. Figure 9 shows the work conservation error for mapping between either model and the framework, normalized with the framework's total work. Figure 9a covers conservative coupling, Fig. 9b the non-conservative coupling approach. As its name implies the conservative approach retains aeroelastic work to within $O(10^{-15})$, whereas the non-conservative approach has conservation errors of around 0.01 or less for most of the pitch angle range. There is a large peak in relative non-conservation in Fig. 9b since the forces and displacements are small near the pitch angle of −6 degrees (and hence the aeroelastic work is near-zero). Interestingly the conservation errors of the fluid and solid coupling always seem to be of opposite sign.

3. Tip displacements

Lastly we look at the tip displacements. These change in a linear way as the pitch angle is increased, just like the lift forces. This is shown in Fig. 10. As mentioned earlier the tip displacements at the highest pitch angles are only 1% of the wing semi-span. Only small differences exist between the various configurations. The non-conservative coupling approach has slightly higher maximum displacements. Fig. 11 shows the tip displacement errors between FEniCS on the one hand and VLM and the framework configuration on the other hand, expressed as a fraction of the FEniCS tip displacement. For most of the pitch angle range the relative tip displacement errors are seemingly constant except for a single outlier at a negative pitch angle of −3 degrees. The differences between both coupling approaches appear negligible, which is to be expected since they utilize the same . We note that the current approach to projecting displacements to the framework representation only induces an error of 1.7% in the maximum displacement for most of the pitch angle range.

V. Conclusions

In this work we presented a methodology for modular aeroelastic coupling that can be made both consistent and conservative. We presented a brief overview of aeroelastic coupling approaches from literature, followed by a treatment of static aeroelastic coupling. We covered the algebraic conditions for both force consistency and work conservation. Following that we introduced the Solver-Independent Field Representation (SIFR) as an intermediate configuration that

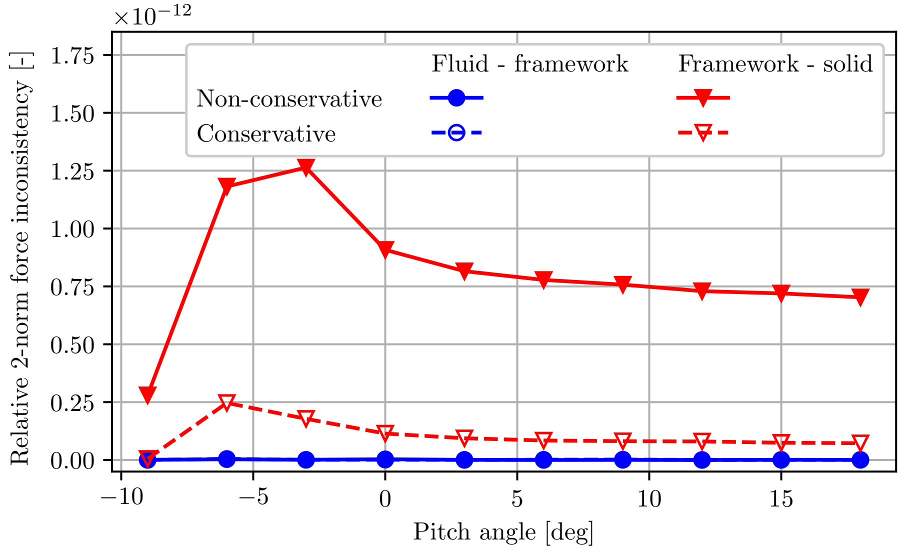
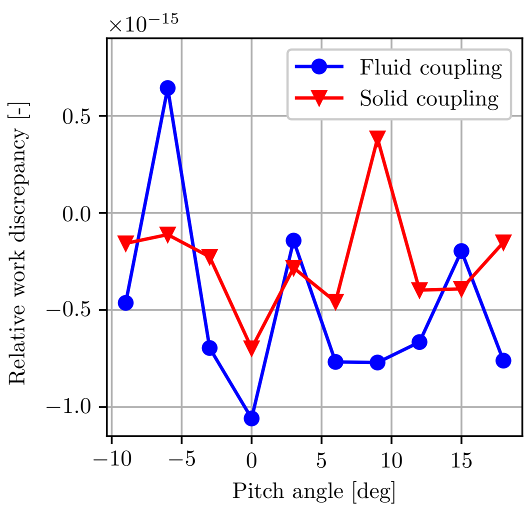
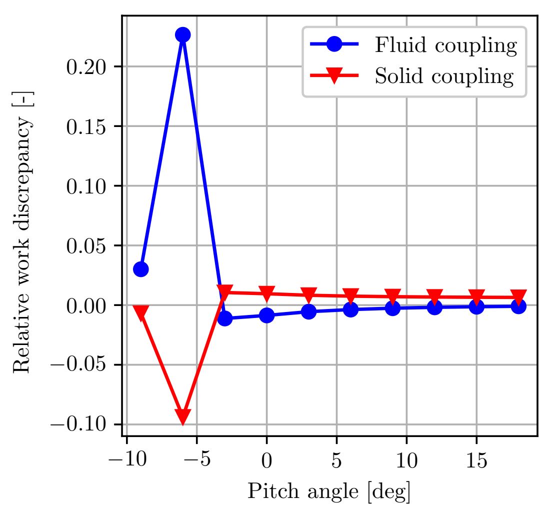


Fig. 8 Relative 2-norm force consistency errors for the two coupling approaches



(a) Conservative coupling



(b) Non-conservative coupling

Fig. 9 Relative work conservation errors for both models coupled to the framework configuration

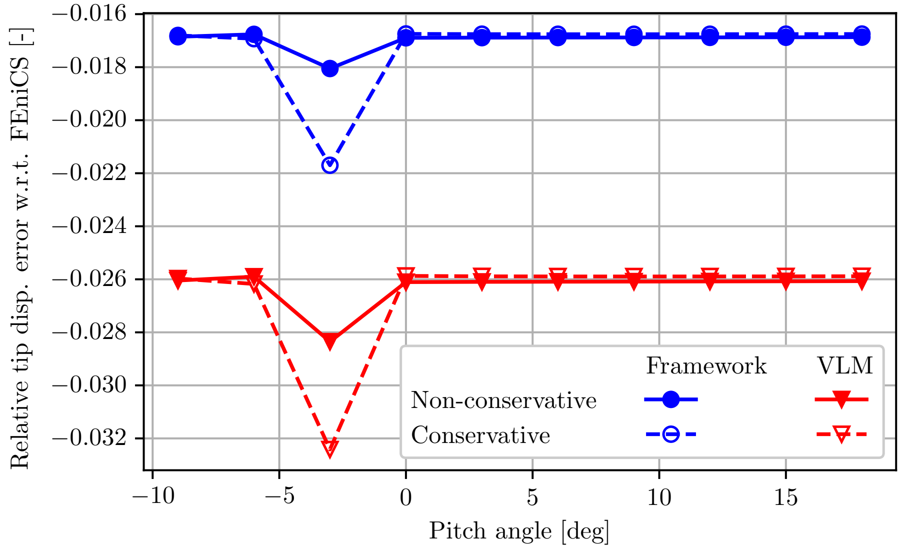


Fig. 10 Absolute maximum displacements of both approaches for all three configurations

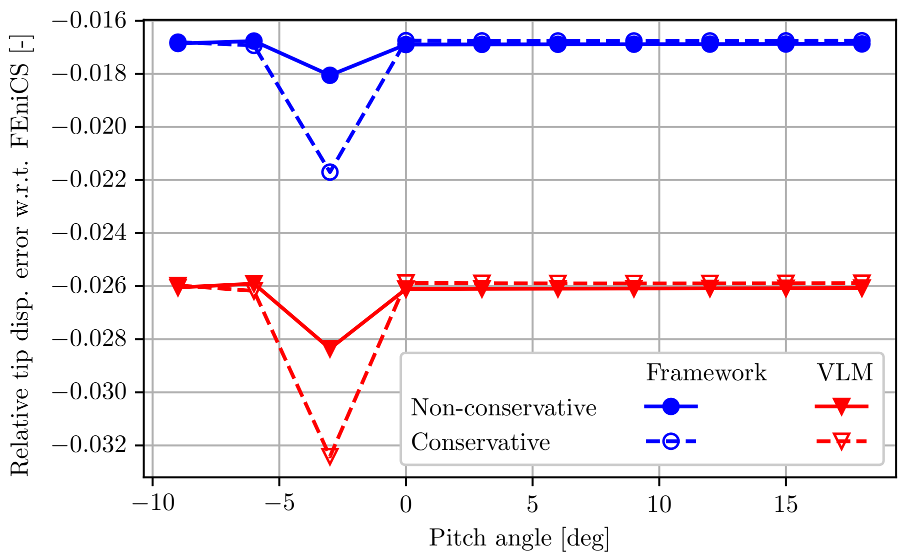


Fig. 11 Relative tip displacement errors of VLM and framework configurations w.r.t. FEniCS tip displacement for both coupling approaches

fluid and solid models can be integrated into for aeroelastic simulations. The intention of this arrangement is to facilitate single- and multi-fidelity Multidisciplinary Design Analysis and Optimization (MDAO). We detailed the approach with which models exchange information with SIFR: This approach consists of compositions of two maps for each solution field, with one map transforming model inputs and outputs into a nodal representation (or vice versa) and the other map projecting the nodal representation to the SIFR geometry. Each one of these maps (or any composition of them) can be constructed such that aeroelastic work is conserved if certain matrix shape constraints are satisfied. We have detailed the shape constraints of all possibilities for both fluid and solid solvers.

The modular aeroelastic coupling methodology was applied to a reconstruction (based on public information) of Aurora Flight Sciences' Passenger Air Vehicle (PAV). Two modular aeroelastic coupling approaches were tested, with one such that aeroelastic work is conserved. A Vortex Lattice Model (VLM) was used as fluid model, and a FEniCS-based Finite Element shell model is the solid model. We found that the differences between both approaches were minor, likely because the aeroelastic effects are limited for the PAV wing due to its stiffness and because static aeroelasticity does not stress the model coupling approaches to their limits. We intend to extend this methodology to dynamic aeroelasticity in the future, which should make the differences between both approaches more pronounced.

VI. Future Work

The focus in this work is on a methodology for consistent and conservative modular static aeroelastic coupling. In future work we intend to apply this approach to dynamic aeroelasticity as well. We expect the differences between conservative and non-conservative coupling to be more significant in such dynamic cases. We furthermore intend to apply this modular coupling methodology to established aeroelastic benchmarks. We envision comparisons that can be made with experimental [29] and computational [30] data of the Pazy wing, the Common Research Model [31] and other suitable data sets in order to characterize the performance of the modular aeroelastic coupling approach in general and the conservative coupling method in particular.

Acknowledgments

This research has been funded through NASA award No. 80NSSC21M0070.

References

[1] Martins, J. R. R. A., and Hwang, J. T., "Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models," *AIAA Journal*, Vol. 51, No. 11, 2013, pp. 2582–2599.

[2] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, 2019, pp. 1075–1104.

[3] Del Carre, A., Muñoz Simón, A., Goizueta, N., and Palacios, R., "SHARPy: A dynamic aeroelastic simulation toolbox for very flexible aircraft and wind turbines," *Journal of Open Source Software*, Vol. 4, No. 44, 2019, p. 1885.

[4] Jasa, J. P., Hwang, J. T., and Martins, J. R. R. A., "Open-source coupled aerostructural optimization using Python," *Structural and Multidisciplinary Optimization*, Vol. 57, No. 4, 2018, pp. 1815–1827.

[5] Su, W., and Cesnik, C. E. S., "Nonlinear Aeroelasticity of a Very Flexible Blended-Wing-Body Aircraft," *Journal of Aircraft*, Vol. 47, No. 5, 2010, pp. 1539–1553.

[6] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations," *AIAA Journal*, Vol. 52, No. 5, 2014, pp. 935–951.

[7] Kiviaho, J. F., Jacobson, K., Smith, M. J., and Kennedy, G., "A Robust and Flexible Coupling Framework for Aeroelastic Analysis and Optimization," *AIAA Aviation 2017 Forum*, 2017.

[8] Slotnick, J. P., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D. J., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," Contractor Report NASA/CR-2014-218178, National Aeronautics and Space Administration, March 2014.

[9] Van Schie, S. P. C., Zhao, H., Yan, J., Xiang, R., Hwang, J. T., and Kamensky, D., "Solver-Independent Aeroelastic Coupling For Large-Scale Multidisciplinary Design Optimization," *AIAA SciTech 2023 Forum*, 2023.

[10] Anderson, R. M., Ning, A., Xiang, R., van Schie, S. P. C., Sperry, M., Sarojini, D., Kamensky, D., and Hwang, J. T., "Aerostructural Predictions Combining FEniCS and a Viscous Vortex Particle Method," *AIAA SciTech 2023 Forum*, 2023.

[11] Warner, M. A. P., Fletcher, A. H., van Schie, S. P. C., and Hwang, J. T., "A Method for Modular MDO Model Assembly via a Solver Independent Field Representation," *AIAA SciTech 2024 Forum*, (under review).

[12] Sarojini, D., Ruh, M. L., Joshy, A. J., Yan, J., Ivanov, A., Scotzniovsky, L., Fletcher, A. H., Orndorff, N., Sperry, M. Z., Gandarillas, V., Asher, I., Chambers, J., Gill, H., Lee, S., Cheng, Z., Rodriguez, G., Zhao, S., Mi, C., Nascenzi, T., and Hwang, J. T., "Large-Scale Multidisciplinary Design Optimization of an eVTOL Aircraft using Comprehensive Analysis," *AIAA SciTech 2023 Forum*, 2023.

[13] Maman, N., and Farhat, C., "Matching fluid and structure meshes for aeroelastic computations: A parallel approach," *Computers & Structures*, Vol. 54, No. 4, 1995, pp. 779–785.

[14] 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, 1998, pp. 95–114.

[15] Smith, M. J., Cesnik, C. E. S., and Hodges, D. H., "Evaluation of Some Data Transfer Algorithms for Noncontiguous Meshes," *Journal of Aerospace Engineering*, Vol. 13, No. 2, 2000, pp. 52–58.

[16] Beckert, A., and Wendland, H., "Multivariate interpolation for fluid-structure-interaction problems using radial basis functions," *Aerospace Science and Technology*, Vol. 5, No. 2, 2001, pp. 125–134.

[17] Rendall, T. C. S., 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.

[18] De Boer, A., Van Zuijlen, A., 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. Domain Decomposition Methods: recent advances and new challenges in engineering.

[19] De Boer, A., Van Zuijlen, A., and Bijl, H., "Comparison of conservative and consistent approaches for the coupling of non-matching meshes," *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, No. 49, 2008, pp. 4284–4297.

[20] Gerstenberger, A., and Wall, W. A., "An eXtended Finite Element Method/Lagrange multiplier based approach for fluid–structure interaction," *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, No. 19, 2008, pp. 1699–1714. Computational Methods in Fluid–Structure Interaction.

[21] Brown, S., "Displacement extrapolations for CFD+CSM aeroelastic analysis," *38th Structures, Structural Dynamics, and Materials Conference*, 1997. AIAA Paper 97-1090.

[22] Kennedy, G., and Martins, J. R. R. A., "Parallel Solution Methods for Aerostructural Analysis and Design Optimization," *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, 2010. AIAA Paper 2010-9308.

[23] Kenway, G. K. W., and Martins, J. R. R. A., "Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration," *Journal of Aircraft*, Vol. 51, No. 1, 2014, pp. 144–160.

[24] 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.

[25] Sarojini, D., Ruh, M. L., Yan, J., Scotzniovsky, L., Orndorff, N. C., Xiang, R., Zhao, H., Krokowski, J. J., Warner, M., van Schie, S. P. C., Cronk, A., Guibert, A. T. R., Chambers, J. T., Wolfe, L., Doring, R., Despins, R., Cibin, J., Anderson, R., Ning, A., Gill, H., Lee, S., Cheng, Z., Cao, Z., Mi, C., Meng, Y. S., Silva, C., Chen, J. S., Kim, H. A., and Hwang, J. T., "Review of Computational Models for Large-Scale MDAO of Urban Air Mobility Concepts," *AIAA SciTech 2024 Forum*, 2024.

[26] Katz, J., and Plotkin, A., *Low-Speed Aerodynamics*, 2$^{\text{nd}}$ ed., Cambridge Aerospace Series, Cambridge University Press, 2001.

[27] Xiang, R., van Schie, S. P. C., Scotzniovsky, L., Yan, J., Kamensky, D., and Hwang, J. T., "Automating adjoint sensitivity analysis for multidisciplinary models involving partial differential equations," *Structural and Multidisciplinary Optimization*, in review.

[28] Scroggs, M. W., Dokken, J. S., Richardson, C. N., and Wells, G. N., "Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes," *ACM Transactions on Mathematical Software*, 2022. To appear.

[29] Avin, O., Raveh, D. E., Drachinsky, A., Ben-Shmuel, Y., and Tur, M., "Experimental Aeroelastic Benchmark of a Very Flexible Wing," *AIAA Journal*, Vol. 60, No. 3, 2022, pp. 1745–1768.

[30] Hilger, J., and Ritter, M. R., "Nonlinear Aeroelastic Simulations and Stability Analysis of the Pazy Wing Aeroelastic Benchmark," *Aerospace*, Vol. 8, No. 10, 2021.

[31] Cartieri, A., "Experimental investigations on the Common Research Model at ONERA-S2MA," *AIAA Scitech 2020 Forum*, 2020.