Solver-Independent Aeroelastic Coupling For Large-Scale Multidisciplinary Design Optimization

Sebastiaan P. C. van Schie *, Han Zhao †, Jiayao Yan ‡, John T. Hwang §, David Kamensky ¶
University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093

I. Nomenclature

A	=	Bound vortex influence coefficient matrix
B	=	Wake vortex influence coefficient matrix
$\boldsymbol{n}$	=	Outward-pointing normal vector
$\boldsymbol{F}$	=	Force vector
p	=	Hydrodynamic pressure
$\boldsymbol{s}$	=	Bound vortex vector
$\boldsymbol{T}$	=	Traction vector
u	=	Wall-tangential velocity component
$\boldsymbol{v}$	=	Freestream velocity vector
y	=	Wall-normal distance
Γ	=	Vortex circulation strength matrix
∂	=	Boundary operator
$\boldsymbol{\epsilon}$	=	Viscous stress tensor
μ	=	Dynamic viscosity
ρ	=	Fluid density
$\boldsymbol{\sigma}$	=	Cauchy stress tensor
Ω	=	Domain

II. Introduction

Various new and innovative technologies, such as composite materials and electric propulsion, have emerged over the last couple of decades. As these technologies mature they are slowly being adopted by the aerospace industry. However, their adoption so far can be characterized as incremental improvement over the status quo, rather than opening up new design spaces. This is in part due to limitations in Multidisciplinary Design Optimization (MDO) approaches. Additionally, the need for new, high-efficiency aircraft designs has opened up interest in new air vehicle concepts. The absence of existing baseline designs makes it difficult to confidently design such new concepts without relying on expensive prototyping and physical testing.

Computational analysis capabilities have increased along with the rise of available computing power with Moore's law. Whereas nowadays large-scale computational simulations are used to inform engineers during the aircraft design process, their use in MDO algorithms has been limited due to their significant computational expense coupled with the large number of design evaluations that are necessary by traditional gradient-free optimization algorithms. In recent years gradient-based optimization has been on the rise within the MDO community, due to its superior performance for optimization problems with many design variables. Gradient-based optimization methods have been implemented in several open-source toolboxes. One example of such is NASA's OpenMDAO framework [1]. As pointed out by Martins and Hwang in [2], gradient-based optimization has found limited use outside of the MDO community since most

*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 member
¶Assistant Professor, Department of Mechanical and Aerospace Engineering

computational methods do not contain the capabilities that are necessary to efficiently and accurately compute derivatives.

The focus of this work is to establish an aeroelastic coupling approach within a large-scale multifidelity MDO framework where such derivatives are available. This MDO approach consists of linked stages with different fidelity requirements. Different solvers are used to fulfill these fidelity requirements, and as such the coupling approach has to be solver-independent. Our goal is to implement a unified aeroelastic coupling formulation that is non-intrusive and allows for derivative computations in both time-independent and dynamic simulations.

This abstract is structured as follows. First in section III various approaches for aeroelastic coupling in Fluid-Structure Interaction (FSI) problems are briefly introduced, after which the choice for a specific approach in this work is covered. Following this we take a continuum mechanics approach in section IV to discuss the mechanisms through which the solid and fluid subdomains affect one another. Section V introduces the fluid and solid models that will be used to show the effectiveness of the aeroelastic coupling method in the final paper. Lastly section VI covers the displacement and load transfer methods that will be used, after which the significance and expected results in the final paper are given in section VII.

III. Coupling in Fluid-Structure Interaction

Several approaches exist for modeling FSI problems. As mentioned by Hou et al. in [3] these approaches can be classified according to the degree to which the numerical fluid and solid models are intertwined. On one end of this classification spectrum are the monolithic approaches, which simultaneously solve the numerical models that are posed on the fluid and solid subdomains with a single solver. Whereas this leads to fully consistent numerical solutions on both subdomains it is also an intrusive approach, since the coupling between the subdomains is explicitly encoded in the numerical problem. It furthermore requires one to solve linear systems that contain both the fluid and solid subproblems, possibly plus parameters that couple both subproblems. The size of this combined system can lead to excessive computational costs. Another FSI solution procedure classification mentioned in [3] has to do with mesh conformity. This classification is concerned with whether both meshes on the fluid-solid interface share all of their vertices, edges and faces. Whereas using conforming meshes simplifies the transfer of loads and displacements between both subdomains, it restricts the discretization methods that can be used and requires both meshes to be constructed according to the most restrictive fidelity requirements of either domain.

A significant amount of work on aeroelastic coupling for FSI problems already exists, owing to the ubiquity of coupled fluid-structure models. We highlight here some works that are relevant within the current context of solid and fluid domains with non-matching geometries. Farhat et al. in [4] identify consistency and conservation as important properties of any projection that maps loads from the fluid to the solid domain. They define consistent methods as those methods that take the shape functions used in the fluid simulation to reconstruct a pointwise representation of the fluid loads. Conservative methods are those that conserve energy and momentum in the coupled simulation. As they point out consistency is easier to attain for fluid and solid domains with different discretization methods than exact conservation. Furthermore, consistency is found to be more important for accuracy than exact conservation.

These properties are also reflected in the work of Brown [5], which is in turn used by Kennedy and Martins in [6]. Kennedy and Martins couple an inviscid panel method to an FEA method to do aerostructural optimization and solve the coupled system with a monolithic/fully-coupled approach, wherein the fluid and solid solutions are computed simultaneously. For mapping aerodynamic pressure loads to the structural model they employ an expression for the virtual work of the pressure load. Posing this expression on the fluid boundary allows to insert the definition of the fluid's surface displacement in terms of the solid displacement and rotation. This gives a computation scheme that maps the pressure load from the fluid to the solid domain boundary in a consistent and conservative way. However, it remains to be seen whether this approach also leads to exact conservation in partitioned/loosely-coupled FSI schemes.

Rendall and Allen in [7] focus on using radial basis functions for load transfer from fluid to solid. This makes the load transfer map meshfree and independent from the discretization approaches of both subdomains. They point to a different notion of "consistency", applicable to both displacement and load transfer. When mapping displacements between both subdomains, points in both domains that are initially coincident must remain so during the coupled simulation. Any load transfer approach must reduce to the identity map in the limit of the fluid and solid meshes matching exactly, such that the load is transferred in a pointwise-exact way.

In this work a partitioned FSI approach is used. This is not without reason: Various fluid and solid solvers are used in the current multi-fidelity MDO framework, with different discretization approaches. With a partitioned FSI approach the fluid and solid solvers can be changed without requiring significant modifications to the MDO workflow and aerostructural coupling method.

Another FSI solution procedure classification mentioned in [3] has to do with mesh conformity. Within this work only non-conforming meshes are used, since the discretization and geometry modeling approaches used by the solid and fluid models are vastly different. Furthermore, we want to be able to change the mesh fineness of both models independently from one another. This non-conformity implies that projection operations are necessary for transferring quantities that are defined on one subdomain's mesh to the other subdomain's mesh.

IV. Physical Interaction Mechanisms

The two-way interaction between the solid and fluid subdomains in any FSI problem is local to the fluid-solid interface. These interactions manifest itself through traction vector $\boldsymbol{T}$. Consistent with the available literature on FSI we take a general continuum mechanics approach to define the interactions, based on textbooks such as Bonet & Wood [8] and Chung [9]. This is also touched upon in several of the works cited in section III. Denote the Cauchy stress tensor in some fluid parcel with $\boldsymbol{\sigma}^{(f)}$, and the outward-pointing normal vector of the fluid domain with $\boldsymbol{n}^{(f)}$. Conversely, denote the Cauchy stress tensor and outward-pointing normal vector of the solid domain with $\boldsymbol{\sigma}^{(s)}$ and $\boldsymbol{n}^{(s)}$ respectively. Newton's third law (action equals reaction) implies compatibility of the tractions $\boldsymbol{T}^{(f)}$ and $\boldsymbol{T}^{(s)}$ that act upon the fluid and solid at the fluid-solid interface:

$$\boldsymbol{T}^{(f)} = \boldsymbol{\sigma}^{(f)} \cdot \boldsymbol{n}^{(f)} = -\boldsymbol{\sigma}^{(s)} \cdot \boldsymbol{n}^{(s)} = \boldsymbol{T}^{(s)} \tag{1}$$

Where we note that the outward-pointing normal vectors of the fluid and solid subdomains are collinear and opposite, i.e. $\boldsymbol{n}^{(f)} = -\boldsymbol{n}^{(s)}$. The choice of a specific approach for using Eq. 1 depends on whether the subdomain meshes are conforming or non-conforming at their interface, and whether there is exact correspondence between the Degrees of Freedom (DoFs) of each subdomain on their interface. Only when this exact correspondence exists can Eq. 1 be used directly. With non-conforming meshes a projection is required in order to map the traction components from one subdomain to the other. This projection often entails relaxing pointwise traction compatibility, especially when different discretization approaches are used for both subdomains (and their DoFs thus have different interpretations). The details of these projections vary depending on the choice of fluid and solid model. We give a brief general overview of the implications of Eq. 1 for partitioned FSI models and how these impact the fluid and solid subdomains. Both subdomains are covered separately.

A. Fluid subdomain

Boundary conditions are the standard method of modeling the effects of solid boundaries on the fluid domain. The fluid's velocity components normal and tangential to the fluid-solid interfaces are set to predefined values. Whereas for standard solid boundaries these velocity components are both equal to zero, applications with nonzero wall-normal or nonzero wall-tangential velocities exist. One example of the former is active flow control, where boundary layer suction and blowing is used in an attempt to improve lift or drag performance [10]. Examples of the latter are external aerodynamics simulations in the automotive industry: Instead of modeling cars as moving over a static floor, the car is defined as being static whereas the floor is given a tangential velocity boundary condition. Doing this greatly reduces the computational complexity of such simulations, since modeling the car as being static removes mesh motions in Eulerian discretizations of the fluid flow.

The role of the geometry of the fluid-solid interface is not immediately obvious in Eq. 1. Solving the solid model entails computing the deformation of the solid domain under an applied load, and thus involves movement of the fluid-solid interface. Any consistent numerical solution of the solid and fluid subdomains must thus take into account the effects that this deformation imposes upon the fluid.

Lastly we go over the stresses that these conditions impose upon a fluid domain. Throughout literature the Cauchy stress tensor components $\boldsymbol{\sigma}_{ij}^{(f)}$ in fluid mechanics are usually written as:

$$\boldsymbol{\sigma}_{ij}^{(f)} = -p\delta_{ij} + \boldsymbol{\epsilon}_{ij}(\mu) \tag{2}$$

Where p is the local hydrodynamic pressure, δ_{ij} is the Kronecker delta and $\boldsymbol{\epsilon}$ is the viscous (shear) stress tensor. The dependence of $\boldsymbol{\epsilon}$ on the fluid's dynamic viscosity μ is explicitly shown. Inviscid flow models do not take the shearing effects of viscosity into account, which causes $\boldsymbol{\epsilon}$ to drop out of Eq. 2. Viscous flow models often take the viscous (shear) stress imposed by solid (wall) boundaries into account by using some type of wall model. The universality of near-wall (boundary layer) velocity profiles is well-known in the mechanics community, and is treated extensively in any textbook that covers viscosity or turbulence effects in fluid flows. Some examples are the books of Landau & Lifshitz [11] and White [12].

Let $u(y)$ denote the wall-tangential velocity at some distance y from a static one-dimensional wall. One feature of these near-wall velocity profiles is that $\lim_{y\to 0} u(y) = 0$, i.e. the flow velocity directly adjacent to any static wall is equal to zero. This is consistent with the zero wall-tangential velocity condition that is covered above. The wall shear stress ϵ that causes these velocity profiles is typically modeled as:

$$\epsilon = \mu \left.\frac{\partial u}{\partial y}\right|_{y=0} \tag{3}$$

In other words, the wall shear stress acting on a viscous fluid at any point along a fluid-solid interface is proportional to the velocity gradient normal to the interface.

B. Solid domain

Boundary conditions can also be involved in modeling the effects of the fluid-solid interface on the solid domain. The traction that acts on the solid domain's boundary can be taken from Eq. 1 as $\boldsymbol{T}^{(s)}$. With the fluid's Cauchy stress as given in Eq. 2 this means that:

$$\boldsymbol{T}^{(s)} = \underbrace{\left(p\delta_{ij} - \boldsymbol{\epsilon}_{ij}(\mu)\right)}_{=-\boldsymbol{\sigma}^{(f)}} \cdot \boldsymbol{n}^{(s)} = \boldsymbol{\sigma}^{(s)} \cdot \boldsymbol{n}^{(s)} \tag{4}$$

This traction is often taken into account in solid models in one of two ways. The first of these is as a traction boundary condition:

$$\boldsymbol{\sigma}^{(s)} \cdot \boldsymbol{n}^{(s)} = \boldsymbol{T}^{(s)} \text{ on } \partial\hat{\Omega}^{(s)} \tag{5}$$

Where $\partial\hat{\Omega}^{(s)}$ is the part of the solid domain boundary where $\boldsymbol{T}^{(s)}$ is known. Alternatively some discretization methods include $\boldsymbol{T}^{(s)}$ as external forcing term instead of as explicit boundary condition.

As was the case for the fluid domain, the formulation of $\boldsymbol{\sigma}^{(s)}$ is dependent on whether a viscous or inviscid fluid model is used. In the latter case the fluid's shear stress $\boldsymbol{\epsilon} = 0$, leaving hydrodynamic pressure p as the only source of traction.

C. Data streams

Based on the mechanisms identified above, the following data transfer streams between the fluid and solid simulations can be identified:

- The boundary displacements that are computed in the solid simulation need to be passed to the fluid solver.
- The Cauchy stress of the fluid at the fluid-solid interface needs to be passed to the solid solver.

At this point we remark that the fluid and solid simulations can in general have different, non-conforming, geometrical representations of the fluid-solid boundary. For example, suppose that the fluid model is discretized with a Finite Volume Method (FVM), whereas the solid model uses Finite Element Analysis (FEA). As such the possibly-curved fluid-solid boundary is represented by piecewise-linear segments in the simulation of the fluid domain, whereas the boundary geometry in the solid simulation depends on the degree of the FEA basis functions. This results in the outward-pointing normal vectors $\boldsymbol{n}^{(f)}$ and $\boldsymbol{n}^{(s)}$ not being collinear everywhere on the fluid-solid boundary.

V. Fluid & Solid Computational Models

The aerostructural coupling approach that is covered in this work is independent of the fluid and solid solvers that are used. To give a more concrete explanation of its implementation we apply it to couple a Vortex Lattice Method (VLM) for fluids and a solid shell formulation using Isogeometric Analysis on a representative aerospace structure. As such we cover some of the relevant aspects of these fluid and solid solvers in sections V.A and V.B before introducing the aeroelastic coupling approach in section VI.

A. Vortex Lattice Methods

We give a brief overview of the relevant aspects of Vortex Lattice Methods for incompressible flows. For a more comprehensive treatment of these methods the reader is referred to any standard textbook on aerodynamics, such as those written by Katz & Plotkin [13] and J.D. Anderson, Jr. [14, 15]. VLM is a numerical method to compute the aerodynamic performance quantities, such as lift and drag, for user-defined lifting surfaces. It is widely used as the aerodynamic submodel for aircraft large-scale MDO. VLM is based on potential flow theory, which assumes the fluid to be incompressible, irrotational, and inviscid. It models the lifting surfaces and their wakes as thin sheets of vortex panels (shown in Fig. 1).

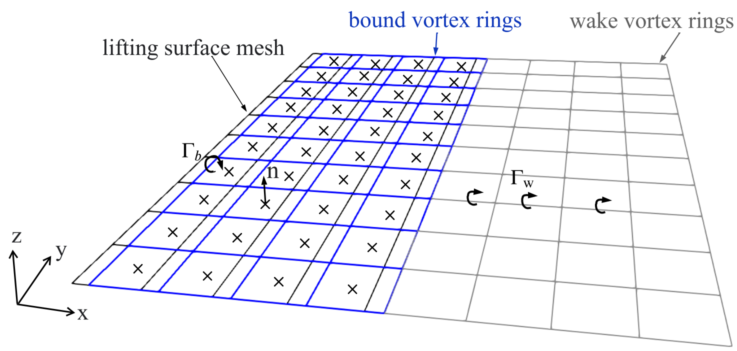

Fig. 1 Vortex ring visualization of the vortex lattice method, where $\boldsymbol{n}$ is the normal vector; $\boldsymbol{\Gamma}_\mathrm{b}$ and $\boldsymbol{\Gamma}_\mathrm{w}$ are the bound circulation strengths and the wake circulation strengths. The cross markers shows the locations of the collocation points.

The flow-tangency condition enforces zero-magnitude fluid velocities normal to the lifting surface, expressed as:

$$\boldsymbol{A}\boldsymbol{\Gamma}_\mathrm{b} + \boldsymbol{B}\boldsymbol{\Gamma}_\mathrm{w} + \boldsymbol{v} \cdot \boldsymbol{n} = 0 \tag{6}$$

Where $\boldsymbol{v}$ is the undisturbed flow velocity or kinematic velocity of the fluid; $\boldsymbol{n}$ is the normal vector of the bound vortex panels; $\boldsymbol{\Gamma}_\mathrm{b}$ and $\boldsymbol{\Gamma}_\mathrm{w}$ are the bound vortex circulation strengths and the wake vortex circulation strengths; and $\boldsymbol{A}$ and $\boldsymbol{B}$ are the aerodynamic influence coefficients. The forces acting on the vortex panels can be calculated using the Kutta–Joukowski theorem:

$$\boldsymbol{F} = \rho \boldsymbol{\Gamma}_\mathrm{b} \boldsymbol{v}_\mathrm{ind} \times \boldsymbol{s} \tag{7}$$

Where ρ is the density of the air; $\boldsymbol{v}_\mathrm{ind}$ is the induced velocities evaluated on the bound vortex collocation points computed with the Biot–Savart law; and $\boldsymbol{s}$ is the bound vector.

The VLM model is implemented in a large-scale MDO framework with automated derivative computation, and verified against the open-source tool OpenAerostruct. This tool was introduced by Jasa et al. in [16]. A viscous correction is implemented with the lift coefficient, Mach number, and Reynolds number as inputs.

B. Isogeometric Shell Analysis

Isogeometric Analysis (IGA) is the primary computational approach that will be used for the solid subproblem. The Kirchhoff-Love (KL) shell formulation covered by Kiendl et al. in [17] will be used to model thin-walled aerospace structures. This sets IGA apart from classical Finite Element Analysis (FEA); as the KL shell model requires a C^1-continuous basis classical FEA cannot use it, since it provides only C^0-continuity. Several open-source platforms and toolboxes are being leveraged to set up the computational model:

- FEniCS [18] is a computational platform that allows its users to pose weak forms of Partial Differential Equations (PDEs). FEA discretizations of these weak forms on user-defined function spaces are constructed automatically.
- tIGAr [19] is a toolbox that extends FEniCS' methods to carry out Isogeometric Analysis (IGA). It uses the process of Bézier extraction to map between the Finite Element function spaces that have been implemented in FEniCS and spline spaces used in IGA.
- ShNAPr [20] is a library that contains definitions of various shell models and functions (such as hyperelastic and St.Venant-Kirchhoff models) for shell IGA.
- PENGoLINS [21] leverages the aforementioned toolboxes to create analysis models of coupled non-matching shells. Each shell is parametrized separately, after which penalty-based constraint enforcement is used to couple the displacements and rotations of intersecting shells.

These platforms are all leveraged by PENGoLINS to carry out IGA simulations of coupled non-matching shells. This allows us to simulate the structural deformations of structural assemblies.

VI. Coupling VLM and IGA

We describe how the VLM and IGA will be coupled. Wherever appropriate we cover how the coupling is solver-independent and translates to other numerical methods. Since the intended application area of the coupling approach is aerospace structures, we use the wing geometry that is shown in Fig. 2 as example application in this work. In [21] IGA shell analysis was done on this geometry with the aforementioned PENGoLINS toolbox. Figure 3 shows the Von Mises stress computed on this geometry. The wing geometry consists of multiple wing skin patches that define the outer wing shape, plus internal surface patches that represent the wing ribs and spars. As mentioned in section V.B each surface patch is individually modeled as a Kirchhoff-Love shell.

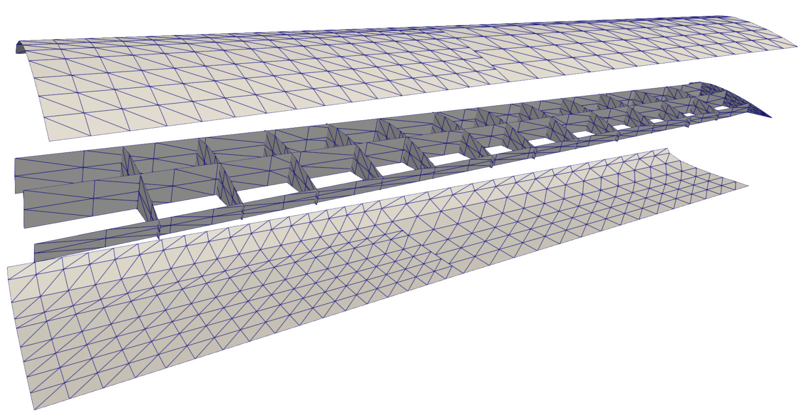

Fig. 2 Example multi-shell wing geometry, image reproduced from [21]

We cover the transfer of displacements and loads separately from one another. This because the nature of their coupling mechanisms is different. As mentioned in section III this work uses a partitioned FSI approach. Thus the fluid and solid subproblems are solved in an alternating, iterative way in order to arrive at consistent aerostructural solutions. Newton methods can be used to drive the coupled system to convergence, since the computational frameworks used by the solid and fluid simulation algorithms allow for the use of solution gradients.

A. Displacement Transfer

Whereas IGA methods use the exact geometry that is input from CAD (and can refine meshes on this geometry without introducing errors), VLM uses a different geometry representation. As shown in Fig. 1 the VLM wing mesh consists of a sheet of vortex panels. At every spanwise position these are defined to follow the camber line of the wing cross-section. This thus allows for a straightforward way of mapping the displacements computed with IGA to the VLM mesh in a way that is representative of the wing planform deformation. We compute a camber plane (i.e. the concept of a camber line, extended to three dimensions) reconstruction in the undeformed solid configuration and track its figures/Abstract_image.png

Fig. 3 Example Von Mises stress on multi-shell wing geometry, geometry displacement scaled with a factor of 50

movement under the displacement of the wing structure. This is straightforward since the camber plane is by definition exactly in the middle of the upper and lower wing surfaces. Thus the camber line displacement is just the average of the upper and lower wing skin displacements.

B. Load Transfer

By using its camber line to represent the wing geometry in the VLM we have a straightforward way of mapping the aerodynamic loads on the VLM geometry to the outer wing skins used in the solid model. This is the case because the volume defined by the solid wing skins contains the VLM geometry. Moreover, the upper and lower wing surfaces define the extremities of the VLM geometry on the wing's camber plane. We can use orthogonal projection to project the traction on the VLM geometry to the upper and lower wing surfaces. Such a projection leads to consistency in the total integrated aerodynamic loads in the solid and fluid simulations without depending on mesh fineness or discretization scheme. Once the aerodynamic loads are projected to the wing skins they can be interpolated onto the wing's surface mesh. With linear (C^0-continuous) functions the classical FEA and IGA function bases coincide, with the basis functions being nodally orthogonal in the mesh vertices. Hence interpolating the projected aerodynamic loads to this function space can be done independently in each vertex. This results in a straightforward and cheap interpolation step, that does not require solving any linear systems.

VII. Expected Results and Significance

The objective of this paper is to establish and implement a solver-independent aeroelastic coupling approach for large-scale multifidelity MDO problems. This coupling approach is to allow for fluid and solid solvers to be swapped out independently, depending on the fidelity requirements at different stages of the MDO process. It will also allow for computation of the derivatives that are leveraged by the MDO framework. The mapping steps of the aeroelastic coupling do not pose any restrictions on the mesh size of one solver (solid or fluid) relative to the other, and allow for the use of different geometrical representations in the fluid and solid models.

In the final paper several test cases will be present. The methods covered in section V will be coupled and tested on the multi-shell wing assembly model shown in Fig. 2 in a static setting. Furthermore, time-dependent tests will be used to determine the accuracy of the coupling approach in dynamic simulations, such as those where flutter occurs; this includes comparisons to established numerical or experimental results from literature. Additionally the approach developed in this work is to be used to couple a Vortex Particle Method implementation with a shell model implemented in classical FEA for several dynamic cases. These will be covered in a corresponding paper [22].

References

[1] 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.

[2] 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.

[3] Hou, G., Wang, J., and Layton, A., "Numerical Methods for Fluid-Structure Interaction — A Review," *Communications in Computational Physics*, Vol. 12, No. 2, 2012, p. 337–377.

[4] 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.

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

[6] 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.

[7] 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.

[8] Bonet, J., and Wood, R. D., *Nonlinear Continuum Mechanics for Finite Element Analysis*, 2nd ed., Cambridge University Press, 2008.

[9] Chung, T. J., *General Continuum Mechanics*, 2nd ed., Cambridge University Press, 2007.

[10] Fahland, G., Stroh, A., Frohnapfel, B., Atzori, M., Vinuesa, R., Schlatter, P., and Gatti, D., "Investigation of Blowing and Suction for Turbulent Flow Control on Airfoils," *AIAA Journal*, Vol. 59, No. 11, 2021, pp. 4422–4436.

[11] Landau, L. D., and Lifshitz, E. M., *Fluid Mechanics*, 2nd ed., Course of Theoretical Physics, Vol. 6, Butterworth-Heinemann, 1987.

[12] White, F. M., *Viscous Fluid Flow*, 3rd ed., McGraw-Hill Series in Mechanical Engineering, McGraw-Hill Education, 2006.

[13] Katz, J., and Plotkin, A., *Low-Speed Aerodynamics*, 2nd ed., Cambridge Aerospace Series, Cambridge University Press, 2001.

[14] Anderson, Jr., J. D., *Fundamentals of Aerodynamics*, 6th ed., McGraw-Hill Series in Aeronautical and Aerospace Engineering, McGraw-Hill Education, 2017.

[15] Anderson, Jr., J. D., *Introduction to Flight*, 9th ed., McGraw-Hill Series in Aeronautical and Aerospace Engineering, McGraw-Hill Education, 2022.

[16] 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.

[17] Kiendl, J., Bletzinger, K.-U., Linhard, J., and Wüchner, R., "Isogeometric shell analysis with Kirchhoff–Love elements," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 49, 2009, pp. 3902–3914.

[18] Logg, A., Mardal, K.-A., Wells, G. N., et al., *Automated Solution of Differential Equations by the Finite Element Method*, Springer, 2012.

[19] Kamensky, D., and Bazilevs, Y., "tIGAr: Automating isogeometric analysis with FEniCS," *Computer Methods in Applied Mechanics and Engineering*, Vol. 344, 2019, pp. 477–498.

[20] Kamensky, D., "Open-source immersogeometric analysis of fluid–structure interaction using FEniCS and tIGAr," *Computers & Mathematics with Applications*, Vol. 81, 2021, pp. 634–648. Development and Application of Open-source Software for Problems with Numerical PDEs.

[21] Zhao, H., Liu, X., Fletcher, A. H., Xiang, R., Hwang, J. T., and Kamensky, D., "An open-source framework for coupling non-matching isogeometric shells with application to aerospace structures," *Computers & Mathematics with Applications*, Vol. 111, 2022, pp. 109–123.

[22] 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," *SciTech*, 2023. Submitted for publication.