

Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: application to a hydraulic arresting gear

Michael C. H. Wu^a, David Kamensky^{b,c}, Chenglong Wang^a, Austin J. Herrema^a, Fei Xu^a, Marco S. Pigazzini^c, Aekaansh Verma^d, Alison L. Marsden^e, Yuri Bazilevs^c, Ming-Chen Hsu^{a,*}

^a*Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, USA*

^b*The Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712, USA*

^c*Department of Structural Engineering, University of California, San Diego, La Jolla, CA 92093, USA*

^d*Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA*

^e*Departments of Pediatrics and Bioengineering, Stanford University, Stanford, CA 94305, USA*

Abstract

This work describes a fluid–structure interaction (FSI) design optimization framework and applies it to improving the structural performance of a water brake used to stop aircraft landing on short runways. Inside the water brake, a dissipative torque is exerted on a rotor through interactions between rotor blades and a surrounding fluid. We seek to optimize blade shape over a parameterized design space, to prevent potentially-damaging stress concentrations without compromising performance. To avoid excessive numbers of costly simulations while exploring the design space, we use a surrogate management framework that combines derivative-free pattern search optimization with automated construction of a low-fidelity surrogate model, requiring only a handful of high-fidelity FSI simulations. We avoid the difficult problem of generating fluid and structure meshes at new points in the design space by using immersogeometric FSI analysis. The structure is analyzed isogeometrically: its design geometry also serves as a computational mesh. This geometry is then immersed in an unfitted fluid mesh that does not depend on the structure’s design parameters. We use this framework to make significant improvements to a baseline design found in the literature. Specifically, there is a 35% reduction of von Mises stress variance and a 25% reduction of maximum of stress, while the resisting torque and mass of the optimized blades remain uncompromised.

Keywords: Fluid–structure interaction; Immersogeometric analysis; Isogeometric analysis; Parametric design optimization; Surrogate management framework; Water brake

*Corresponding author

Email address: jmchsu@iastate.edu (Ming-Chen Hsu)

1. Introduction

A hydraulic arresting gear is a system to help dissipate the kinetic energy of landing airplanes, to prevent overrun accidents, in which aircraft do not stop before the end of the runway [1]. Overrun accidents are a major concern in naval aviation, where planes need to land on aircraft carrier flight decks, which are shorter than normal runways. While landing, a naval aircraft ejects a hook, which catches on a cable suspended above the flight deck. Tension in this cable is transferred into torque through a tape drum. This torque twists a rotor with thin paddles, called blades, protruding into a viscous fluid that is initially at rest with respect to a static casing that is fixed in the inertial reference frame of the aircraft carrier. The static casing, or stator, interacts with the fluid through vanes similar to the rotor blades. The interaction between the blades, vanes and fluid creates an arresting force on the rotor, which in turn slows the aircraft. The subsystem comprising the rotor, fluid, and stator is referred to as the *water twister* (see Figure 1). Due to the extreme loading conditions on the rotor blades and stator vanes, these components may suffer fatigue. The possibility of damage to the rotor blades and stator vanes is not speculation. The United States Government Accountability Office (GAO) reported in 2014 that a new arresting gear developed for the Ford-class aircraft carrier experienced several failures during the land-based testing, which led to redesign and modification of several subsystems, most notably the water twisters [2, page 15]. GAO estimated that this redesign contributed to 4.6 years of delays [2, Figure 2] in the development of the new arresting gear.

This case study illustrates how uncovering design flaws through physical testing can be quite costly; water twister design stands out among the most vexing technical problems encountered in the multi-billion dollar Ford-class carrier program [3]. Computer simulation could be used to avoid unnecessary testing of deficient designs and could let engineers explore larger design spaces in search of optimal solutions. Computer simulation of fluid–structure interaction (FSI) systems, such as water twisters, remains an active topic of research in the computational mechanics community [4], but some of the technologies under study have reached a sufficient level of maturity to be useful in real engineering efforts [5–12]. Publicly-available studies using computer simulation to predict arresting gear water twister performance go back nearly two decades. In the late 1990s, Chiu [13] used semi-empirical turbulence models in the commercial computational fluid dynamics (CFD) software ANSYS Fluent [14] to estimate the torque required to twist the rotor component at a fixed rate, but found large discrepancies of at least 30% between predictions from the computational model and an empirical model of torque that was calibrated through experimental testing. Wang et al. [15] recently revisited the problem, using an in-house code to perform both CFD and FSI simulations of a water twister. The torque values computed by [15] were in excellent agreement with the experimental results.

Simulations in [15] benefited from a number of advances in CFD and FSI technology. The

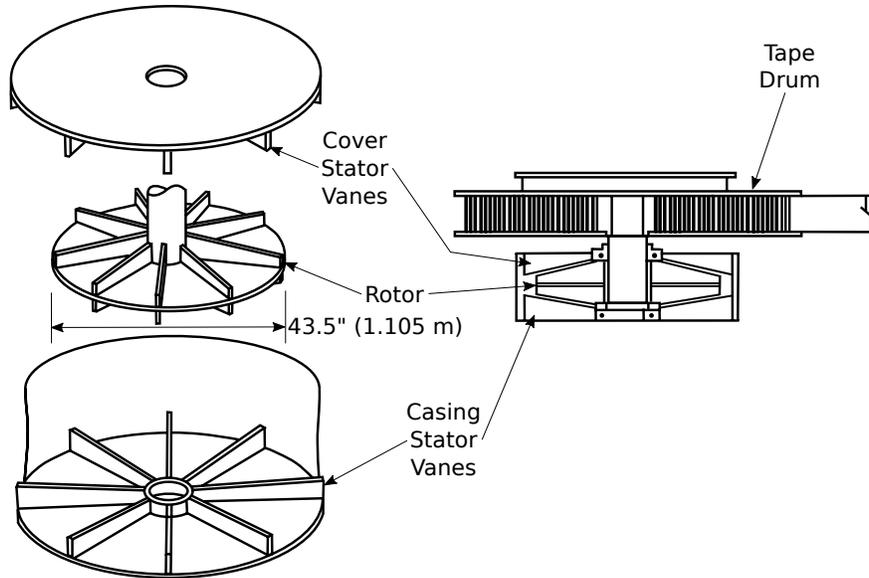


Figure 1: A schematic representation of Virginia Tech (VT) water twister from [13]. A similar design (Model 64) can be seen in [1].

CFD results were computed using the variational multiscale (VMS) method [16], which allows turbulence to be accounted for in a strongly-consistent manner [17–21]. The structure subproblem in FSI computations was approximated using isogeometric analysis (IGA) [22], which directly employs a spline-based CAD (computer-aided design) representation of structure geometry as a computational mesh. The FSI computations in [15] predicted rotor blade deflections and strains. The results showed a strain concentration of sufficient magnitude to potentially cause fatigue of the steel rotor blades. Distributing strain more uniformly would lower risk of machine failure. This work aims to optimize rotor blade design to reduce strain concentration and at the same time maintain rotor weight and torque, while considering the strong coupling effect between the fluid and structure.

One broad distinction among optimization techniques is between gradient-based and derivative-free methods. The choice of method for a particular problem depends on several factors. For example, the availability of gradient information, the level of noise in the function, the cost of function evaluations, and the complexity of implementation. Coupling complex FSI simulations to optimization algorithms involves several challenges. Each cost function evaluation requires a time-dependent, three-dimensional solution of the FSI equation system, which is computationally expensive to evaluate, and obtaining gradient information poses formidable challenges. The gradient information may be obtained directly using finite difference methods or numerically using adjoint solutions. Gradients obtained directly using finite differences are often prohibitively expensive for large problems and can be easily drowned out by numerical noise. Solving unsteady adjoint FSI problems necessitates the need to store large time history of primal solutions, which

can be practically challenging.

With these issues in mind, we consider the derivative-free optimization in this work and extend the isogeometric design optimization framework described by [23]. Herrema et al. [23] provided an example of how IGA can streamline the design-through-analysis pipeline, allowing it to be automated and placed in an optimization loop that searches for parameters defining an optimal design. In the case of the water twister application, however, we face new challenges. Solving FSI problems is much more computationally costly than solving the shell structure problems considered in [23], so we cannot feasibly execute the number of model evaluations required by the generalized pattern search used in [23]. To this end, the present work employs the surrogate management framework (SMF) [24–26] to accelerate parametric design optimization of systems that require evaluations of expensive cost functions with little or no gradient information. The SMF has previously been used in conjunction with CFD and FSI to optimize cardiovascular medical devices and surgical plans [27–29]. Further, rotor blade designs do not directly provide a conforming discretization of the volume occupied by fluid; Wang et al. [15] generated a boundary-fitted finite element mesh for the fluid subproblem and used a sliding-interface formulation [30] to accommodate the relative motion of the rotor and stator, but boundary-fitted mesh-generation requires some labor-intensive, geometry-specific work on the part of the analyst (see, e.g. [31]). In this work, we instead immerse each structure design in the same analysis mesh of the fluid domain, to be generated once, as a pre-processing step. We refer to this design-through-analysis workflow of immersing design geometries directly into unfitted analysis meshes as *immersogeometric analysis* [32].

The geometrical flexibility of immersogeometric FSI analysis means that the design optimization framework developed in this work can be extended immediately to other application areas. The technology described in this paper is directly applicable to constrained optimization of any FSI system involving thin structural components. The use of IGA allows parametrically generated thin structures to be directly employed in FSI analysis, whereas the use of immersogeometric analysis eliminates the repetitive generation of boundary-fitted fluid meshes. Further, in addition to performing optimization of expensive cost functions efficiently, the algorithm employed by the SMF is entirely automated. Thus, when using the presented framework to optimize a thin-shell design within a complex FSI system, a designer’s primary tasks would consist of the parametric definition and programmatic construction of the design, the identification of design variables, and the formulation of a cost function, all of which are fundamental design activities. The entire subsequent process of FSI analysis and design optimization is fully automated and requires no user interaction. Thus, the framework proposed in this work represents the potential not only for identifying optimal engineering designs in complex fluid–thin structure interaction systems, but also for significant cost and time savings during the design process.

The mathematical problem of fluid–thin structure interaction and our immersogeometric discretization of it are described in Sections 2 and 3. Section 4 explains how SMF can be used to accelerate parametric design optimization of systems with expensive cost function evaluations. We then apply these technologies to water twister optimization in Section 5 and find that the blade geometry of a water twister design used in previous studies can be improved to avoid stress concentrations without changing the overall weight or performance. In Section 6, we conclude with a summary of our findings and a discussion of future improvements to and applications of SMF-based design optimization of FSI systems.

2. Mathematical model for FSI

To optimize the performance of an FSI system, we first model it as a mathematical problem. We can estimate a typical flow speed by considering the radius (around half a meter) and rate of rotation (up to 800 rpm) of the water twister’s rotor. This characteristic speed is more than an order of magnitude lower than the sound speed in water, so we model the water contained in the arresting gear as an incompressible fluid. The blades of the rotor are about 10 mm thick, with in-plane dimensions about half a meter, so we model them using Kirchhoff–Love thin shell theory. Their strains remain small during normal operation, so we assume a St. Venant–Kirchhoff material model, i.e. the formal extension of linearized small-strain elasticity to the setting of large displacements and rotations. For the purposes of fluid–structure coupling, we model the thin blades geometrically as 2D surfaces and enforce kinematic and traction compatibility between the fluid and thin structures on these surfaces rather than on distinct fluid–structure interfaces corresponding to the different sides of each blade.

To make these modeling assumptions precise, we state them as a coupled partial differential equation (PDE) system. To naturally accommodate the distributional forcing associated with the thin immersed blades and vanes and to provide a clear template for the development of variational numerical methods, we state the problem in weak form. For a thin structure with time-dependent midsurface location Γ_t immersed in a fluid occupying the volume Ω , the weak FSI problem is: Find fluid velocity $\mathbf{u}_1 \in \mathcal{S}_u$, fluid pressure $p \in \mathcal{S}_p$, structure displacement $\mathbf{y} \in \mathcal{S}_y$ with velocity $\mathbf{u}_2 = \dot{\mathbf{y}}$, and fluid–structure traction $\boldsymbol{\lambda} \in \mathcal{S}_\ell$ such that, for all test functions $\mathbf{w}_1 \in \mathcal{V}_u$, $q \in \mathcal{S}_q$, $\mathbf{w}_2 \in \mathcal{V}_y$, and $\delta\boldsymbol{\lambda} \in \mathcal{V}_\ell$ we have, at a.e. time $t \in (0, T)$,

$$B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}) - F_1(\{\mathbf{w}_1, q\}) + \int_{\Gamma_t} \mathbf{w}_1 \cdot \boldsymbol{\lambda} \, d\Gamma + \int_{\Gamma_t} \mathbf{w}_1 \cdot \boldsymbol{\beta}(\mathbf{u}_1 - \mathbf{u}_2) \, d\Gamma = 0, \quad (1)$$

$$B_2(\mathbf{w}_2, \mathbf{y}) - F_2(\mathbf{w}_2) - \int_{\Gamma_t} \mathbf{w}_2 \cdot \boldsymbol{\lambda} \, d\Gamma - \int_{\Gamma_t} \mathbf{w}_2 \cdot \boldsymbol{\beta}(\mathbf{u}_1 - \mathbf{u}_2) \, d\Gamma = 0, \quad (2)$$

$$\int_{\Gamma_t} \delta\boldsymbol{\lambda} \cdot (\mathbf{u}_1 - \mathbf{u}_2) \, d\Gamma = 0, \quad (3)$$

subject to initial conditions

$$\mathbf{u}_1|_{t=0} = \mathbf{u}_1^0, \quad \mathbf{y}|_{t=0} = \mathbf{y}^0, \quad \text{and} \quad \mathbf{u}_2|_{t=0} = \mathbf{u}_2^0, \quad (4)$$

where $\mathcal{S}_{(\cdot)}$ and $\mathcal{V}_{(\cdot)}$ are trial solution and test function spaces and B_1 , B_2 , F_1 , and F_2 are functionals defining the fluid and structure subproblems, indexed 1 and 2 respectively. The coefficient β penalizes deviations from the fluid–structure kinematic compatibility condition on Γ_t . Obviously these penalty forces vanish for any solution satisfying the constraint equation (3) with respect to a sufficiently-rich test space \mathcal{V}_ℓ , but the penalty forces play an important role in numerical schemes based on this formulation. The augmented Lagrangian formulation of the fluid–structure kinematic constraint also incorporates the traction compatibility condition on the fluid–structure interface, as elaborated in [33].

Remark 1. This weak problem statement can be easily specialized to problems in which the motion of Γ_t is prescribed. In such cases, the trial solution space \mathcal{S}_y for the thin structure displacement field is set to the prescribed displacement of Γ_t and the corresponding test space is empty.

The fluid and structure subproblem functionals are defined in accordance with the modeling assumptions discussed at the beginning of the section. The fluid subproblem functionals are

$$B_1(\{\mathbf{w}, q\}, \{\mathbf{u}, p\}) = \int_{\Omega} \mathbf{w} \cdot \rho_1 \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) d\Omega + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}_1 d\Omega + \int_{\Omega} q \nabla \cdot \mathbf{u} d\Omega, \quad (5)$$

$$F_1(\{\mathbf{w}, q\}) = \int_{\Omega} \mathbf{w} \cdot \rho_1 \mathbf{f}_1 d\Omega + \int_{(\Gamma_h)_t} \mathbf{w} \cdot \mathbf{h}_1 d\Gamma, \quad (6)$$

where ρ_1 is the fluid mass density, μ is the dynamic viscosity, $\boldsymbol{\varepsilon}$ is the (Eulerian) symmetric gradient operator, $\boldsymbol{\sigma}_1 = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u})$ is the (Newtonian) Cauchy stress, \mathbf{f}_1 is a prescribed body force on the fluid subproblem, \mathbf{h}_1 is a prescribed traction on $(\Gamma_h)_t \subset \partial(\Omega_1)_t$.

The thin-structure subproblem functionals are

$$B_2(\mathbf{w}, \mathbf{y}) = \int_{\Gamma_t} \mathbf{w} \cdot \rho_2 h_{\text{th}} \left. \frac{\partial^2 \mathbf{y}}{\partial t} \right|_{\mathbf{X}} d\Gamma + \int_{\Gamma_0} \int_{-h_{\text{th}}/2}^{h_{\text{th}}/2} D_{\mathbf{w}} \mathbf{E} : \mathbf{S} d\xi^3 d\Gamma, \quad (7)$$

$$F_2(\mathbf{w}) = \int_{\Gamma_t} \mathbf{w} \cdot \rho_2 h_{\text{th}} \mathbf{f}_2 d\Gamma + \int_{\Gamma_t} \mathbf{w} \cdot \mathbf{h}^{\text{net}} d\Gamma, \quad (8)$$

where ρ_2 is the structure mass density, h_{th} is the shell thickness, ξ^3 is a through-thickness coordinate, \mathbf{f}_2 is a body force on the structure (averaged through its thickness), and $\partial(\cdot)/\partial t|_{\mathbf{X}}$ indicates time differentiation with respect to a fixed material point \mathbf{X} . The last term of F_2 sums the prescribed tractions on the two sides of Γ_t : $\mathbf{h}^{\text{net}} = \mathbf{h}(\xi^3 = -h_{\text{th}}/2) + \mathbf{h}(\xi^3 = +h_{\text{th}}/2)$. The elasticity term has been referred to the midsurface reference configuration Γ_0 . \mathbf{E} is the Green–Lagrange

strain tensor corresponding to midsurface displacement \mathbf{y} . The definition of \mathbf{E} in terms of the midsurface displacement \mathbf{y} relies on Kirchhoff–Love thin shell theory [34, Section 3.2]. $D_{\mathbf{w}}\mathbf{E}$ is the functional derivative of \mathbf{E} in the direction \mathbf{w} . \mathbf{S} is the second Piola–Kirchhoff stress tensor, which encapsulates the material constitutive behavior. To model metals subjected to small strains, we use an isotropic St. Venant–Kirchhoff model, in which $\mathbf{S} = \mathbb{C} : \mathbf{E}$, where \mathbb{C} is an isotropic elasticity tensor, determined by Young’s modulus E and Poisson ratio ν , as in standard linear elasticity.

In the context of the present application, the most questionable omission from this FSI model is cavitation. The model stated above does not provide any mechanism for the formation of voids in regions of extremely low fluid pressure. A previous study [13] speculated that cavitation might affect the performance of hydraulic arresting gears, but our previous work [15] on this application has shown that the incompressible Newtonian treatment of the fluid is sufficient to reproduce experimentally-calibrated torque–rotation rate curves with a much higher degree of accuracy than previous modeling efforts.

3. Immersogeometric discretization of the FSI model

Obtaining exact solutions to the mathematical problem stated in Section 2 is practically impossible for all but the most contrived instances. We therefore compute approximate solutions in finite-dimensional trial solutions spaces. As mentioned in Section 1, we aim to simplify the generation of FSI discretizations by immersing structure geometries into an unfitted fluid discretization. The principal disadvantage of this immersogeometric approach is that unfitted analysis meshes do not take advantage of *a priori* knowledge about the precise locations of sharp gradients or discontinuities in solution variables and their derivatives, such as those typically occurring at fluid–structure interfaces. The per-degree-of-freedom approximation power of immersogeometric discretizations is therefore severely limited relative to that of carefully designed meshes that are fitted to interfaces of reduced solution regularity and judiciously refined in anticipation of steep solution gradients. Obtaining immersogeometric solutions of comparable accuracy to solutions computed using well-designed boundary-fitted meshes is therefore more computationally costly [35]. However, in many situations, the accuracy achievable using unfitted meshes with practically-accessible resolutions may be sufficient for engineering purposes, so the advantage of convenience outweighs the disadvantage of reduced accuracy. In the present application, the converged boundary-fitted simulations of [15] and the availability of experimentally-calibrated torque–rotation rate curves allow us to directly assess the error introduced by unfitted discretization and compare its magnitude to the error introduced by modeling the hydraulic arresting gear’s physics as a mathematical problem. The remainder of this section details our immersogeometric discretization.

3.1. Variational multiscale fluid subproblem

To robustly discretize the fluid subproblem in space without needing to resolve small-scale turbulent flow features, we replace B_1 and F_1 with mesh-dependent functionals B_1^{VMS} and F_1^{VMS} , derived from variational multiscale (VMS) analysis. In short, VMS analysis substitutes an ansatz for fine scale velocities and pressures into the weak fluid subproblem [16–20]. This ansatz is consistent with the strong form of the Navier–Stokes equations, so the resulting discretization transitions seamlessly from turbulence modeling to direct numerical simulation as it is refined.

To define the mesh-dependent VMS functionals, we introduce a collection of disjoint fluid elements $\{\Omega^e\}$ such that $\Omega = \cup_e \overline{\Omega^e}$. Let \mathcal{V}_u^h and \mathcal{V}_p^h be discrete velocity and pressure spaces defined over $\{\Omega^e\}$. The semidiscrete VMS fluid subproblem is: Find $\mathbf{u}_1^h \in \mathcal{V}_u^h$ and $p^h \in \mathcal{V}_p^h$ such that, for all $\mathbf{w}_1^h \in \mathcal{V}_u^h$ and $q^h \in \mathcal{V}_p^h$,

$$B_1^{\text{VMS}}(\{\mathbf{w}_1^h, q^h\}, \{\mathbf{u}_1^h, p^h\}) - F_1^{\text{VMS}}(\{\mathbf{w}_1^h, q^h\}) + \int_{\Gamma_t} \mathbf{w}_1^h \cdot \boldsymbol{\lambda} \, d\Gamma + \int_{\Gamma_t} \mathbf{w}_1^h \cdot \beta(\mathbf{u}_1^h - \mathbf{u}_2) \, d\Gamma = 0, \quad (9)$$

where

$$\begin{aligned} B_1^{\text{VMS}}(\{\mathbf{w}, q\}, \{\mathbf{u}, p\}) &= B_1(\{\mathbf{w}, q\}, \{\mathbf{u}, p\}) - \sum_e \int_{\Omega^e} \left(\mathbf{u} \cdot \nabla \mathbf{w} + \frac{\nabla q}{\rho_1} \right) \cdot \mathbf{u}' \, d\Omega - \sum_e \int_{\Omega^e} p' \nabla \cdot \mathbf{w} \, d\Omega \\ &+ \sum_e \int_{\Omega^e} \mathbf{w} \cdot (\mathbf{u}' \cdot \nabla \mathbf{u}) \, d\Omega - \sum_e \int_{\Omega^e} \frac{\nabla \mathbf{w}}{\rho_1} : (\mathbf{u}' \otimes \mathbf{u}') \, d\Omega \\ &+ \sum_e \int_{\Omega^e} (\mathbf{u}' \cdot \nabla \mathbf{w}) \bar{\tau} \cdot (\mathbf{u}' \cdot \nabla \mathbf{u}) \, d\Omega, \end{aligned} \quad (10)$$

and

$$F_1^{\text{VMS}}(\{\mathbf{w}, q\}) = F_1(\{\mathbf{w}, q\}). \quad (11)$$

The forms B_1^{VMS} and F_1^{VMS} are the VMS semidiscrete counterparts of B_1 and F_1 . \mathbf{u}' is the fine scale velocity ansatz,

$$\mathbf{u}' = -\tau_M \left(\rho_1 \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f}_1 \right) - \nabla \cdot \boldsymbol{\sigma}_1 \right), \quad (12)$$

and p' is the fine scale pressure,

$$p' = -\rho_1 \tau_C \nabla \cdot \mathbf{u}. \quad (13)$$

These fine scale solution components are clearly proportional to the residuals of the strong momentum and continuity equations forming the incompressible Navier–Stokes system. The stabilization

parameters τ_M , τ_C , and $\bar{\tau}$ are defined as

$$\tau_M = \left(s \left(\frac{4}{\Delta t^2} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{G}(\mathbf{u} - \hat{\mathbf{u}}) + C_I \left(\frac{\mu}{\rho_1} \right)^2 \mathbf{G} : \mathbf{G} \right) \right)^{-1/2}, \quad (14)$$

$$\tau_C = (\tau_M \text{tr} \mathbf{G})^{-1}, \quad (15)$$

$$\bar{\tau} = (\mathbf{u}' \cdot \mathbf{G} \mathbf{u}')^{-1/2}, \quad (16)$$

where Δt is a timescale associated with the as-yet-unspecified temporal discretization, C_I is a dimensionless positive constant derived from element-wise inverse estimates [36, 37], and \mathbf{G} generalizes the element diameter “ h ” to physical elements mapped through $\mathbf{x}(\boldsymbol{\xi})$ from a parametric parent element:

$$G_{ij} = \sum_{k=1}^d \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j}. \quad (17)$$

The scaling factor s in the definition of τ_M is a dimensionless quantity that is allowed to vary in space. In most of Ω , $s = 1$, but, in an $O(h)$ neighborhood of Γ_t , $s = s^{\text{shell}} \geq 1$. This scaling was introduced by [32] to improve mass conservation near thin immersed structures.

The VMS formulation maintains discrete stability over broad classes of velocity and pressure finite element spaces. We are not restricted to special inf–sup stable combinations. For computations in this paper, we use an “equal order” discretization scheme: the same scalar discrete space is used to represent the pressure and each Cartesian component of velocity. In particular, we use piecewise linear functions defined over tetrahedral elements.

3.2. Isogeometric thin shell analysis

The Bubnov–Galerkin method of directly posing variational problems over finite dimensional subspaces is highly effective in structural mechanics. However, it has, until recently, rarely been applied to the Kirchhoff–Love thin shell problem in displacement form, as specified through the definitions of B_2 and F_2 in Eqs. (7) and (8). The reason for this is that B_2 involves $L^2(\Gamma_0)$ inner products of second derivatives of the displacement trial and test functions. This means that, for the formulation to be well defined, the spaces over which it is posed must be subsets of $H^2(\Gamma_0)$. Traditional finite element spaces do not meet this criterion. However, C^1 or smoother spline spaces used in isogeometric analysis do. The possibility of using isogeometric spline spaces to solve Kirchhoff–Love shell problems in displacement form was first exploited by [38]. In this paper, we apply the isogeometric discretization of the thin structure subproblem, which amounts to simply posing the thin shell weak problem as-is, over C^1 spline spaces.

3.3. Discretization of the fluid–structure kinematic constraint

As when using a Lagrange multiplier (pressure) to enforce incompressibility of the fluid velocity field, we must be careful when discretizing the Lagrangian of the fluid–structure kinematic constraint. We must either choose an inf–sup stable combination of spaces for $\boldsymbol{\lambda}$, \mathbf{u}_1 , and \mathbf{u}_2 , or adjust our numerical formulation to be stable for arbitrary choices of discrete spaces. Because Γ_t cuts through fluid elements in a different arbitrary way every time step, there is no obvious construction of stable discrete spaces, so we opt instead to develop a stable formulation.

3.3.1. Reformulation of the kinematic constraint

For reasons explained in [32], we formally eliminate the tangential component of the Lagrange multiplier $\boldsymbol{\lambda}$, leaving only the penalty method to enforce the no-slip condition,

$$\mathbf{u}_1 - (\mathbf{u}_1 \cdot \mathbf{n}_2) \mathbf{n}_2 = \mathbf{u}_2 - (\mathbf{u}_2 \cdot \mathbf{n}_2) \mathbf{n}_2, \quad (18)$$

on Γ_t . Following [39], we regularize the no-penetration condition, enforced by the scalar normal component λ of the interface traction $\boldsymbol{\lambda}$, as follows:

$$(\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2 \rightarrow (\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2 - \frac{r}{\beta} \lambda, \quad (19)$$

where $r \geq 0$ is a dimensionless constant, typically $\ll 1$. In the cited references, we elaborate on how these regularizations of FSI kinematics can be viewed as thin-structure degenerations of Nitsche’s method [40] and Barbosa–Hughes stabilization [41], respectively. The regularized problem that we proceed with may be written: Find $\mathbf{u}_1 \in \mathcal{S}_u$, $p \in \mathcal{S}_p$, $\mathbf{y} \in \mathcal{S}_d$, and $\lambda \in \mathcal{S}_\ell$ such that, for all test functions $\mathbf{w}_1 \in \mathcal{V}_u$, $q \in \mathcal{V}_p$, $\mathbf{w}_2 \in \mathcal{V}_d$, and $\delta\lambda \in \mathcal{V}_\ell$

$$\begin{aligned} & B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}) - F_1(\{\mathbf{w}_1, q\}) + B_2(\mathbf{w}_2, \mathbf{y}) - F_2(\mathbf{w}_2) \\ & + \int_{\Gamma_t} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \lambda \mathbf{n}_2 \, d\Gamma \\ & + \int_{\Gamma_t} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \tau_{\text{NOR}}^B ((\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2) \mathbf{n}_2 \, d\Gamma \\ & + \int_{\Gamma_t} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \tau_{\text{TAN}}^B ((\mathbf{u}_1 - \mathbf{u}_2) - ((\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2) \mathbf{n}_2) \, d\Gamma \\ & + \int_{\Gamma_t} \delta\lambda \left((\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2 - \frac{r\lambda}{\beta} \right) \, d\Gamma = 0, \end{aligned} \quad (20)$$

where we have split the penalty term into normal and tangential components. Inspired by applications of Nitsche’s method to viscous incompressible flows, we propose to scale the tangential

penalty like

$$\tau_{\text{TAN}}^B = C_{\text{TAN}} \frac{\mu}{h}, \quad (21)$$

where C_{TAN} is a dimensionless $\mathcal{O}(1)$ constant and h is a measure of the fluid element diameter, with units of length. This causes the no-slip portion of the boundary condition on Γ_t to disappear in the inviscid limit of $\mu \rightarrow 0$. The no-penetration constraint should remain in the inviscid limit, though, so we propose to scale it like

$$\tau_{\text{NOR}}^B = \max \left\{ C_{\text{NOR}}^{\text{inert}} \frac{\rho_1 h}{\Delta t}, C_{\text{NOR}}^{\text{visc}} \frac{\mu}{h} \right\}, \quad (22)$$

where $C_{\text{NOR}}^{\text{inert}}$ and $C_{\text{NOR}}^{\text{visc}}$ are dimensionless constants and Δt is a time scale associated with the temporal discretization. The low-viscosity branch of τ_{NOR}^B maintains the correct dimensions without being proportional to μ or invoking arbitrary global length or time scales, which may not be clearly defined or appropriate to local solution behavior. Tacit in the selection of a single global “ h ” associated with a fluid mesh are assumptions of quasi-uniformity and isotropy. These are not necessarily satisfied in practice. To define h is such a way that it remains robust for nonuniform and anisotropic meshes, we select it for each point \mathbf{x} in $\Gamma_t \cap \Omega_1$ using the formula (cf. [42, Eq. (13)])

$$h(\mathbf{x}) = (\mathbf{n}_2(\mathbf{x}) \cdot \mathbf{G}(\mathbf{x}) \mathbf{n}_2(\mathbf{x}))^{-1/2}, \quad (23)$$

where $\mathbf{n}_2(\mathbf{x})$ is the normal to Γ_t at point \mathbf{x} and the components of the tensor \mathbf{G} are defined as in Eq. (17). Heuristically, h is the local mesh size in the direction perpendicular to the immersed surface. That is the direction of mesh refinement that is relevant to resolving both the shearing associated with tangential forces on Γ_t and the pressure jumps associated with normal forces. If $\beta \sim \tau_{\text{NOR}}^B$, then the perturbation to the no-penetration residual introduced in Eq. (19) will vanish as $h \rightarrow 0$.

3.4. Discrete representation of immersed boundaries

A critical question of discretization is how to perform numerical quadrature of the integrals over Γ_t . Following [43], we define a Gaussian quadrature rule with respect to a parameterization of the reference configuration Γ_0 of the immersed boundary and weight it by the Jacobian determinant of the mapping from Γ_0 to Γ_t . For a quasi-uniform fluid mesh with elements diameters asymptotically bounded above and below by h , this quadrature scheme suggests that surface quadrature elements should be of diameter $\mathcal{O}(h)$, but we do not enforce this condition strictly. Studies from the finite cell literature [44–46] suggest that the impact of quadrature errors on $(d - 1)$ -dimensional boundaries is small relative to the effects of errors in volume quadrature. Techniques for evaluating functions defined on the tetrahedral fluid mesh at quadrature points on Γ_t are discussed in [47, Section 3.3].

The quadrature rule on Γ_t determines the discrete representation of the fluid–structure interface Lagrange multiplier, λ . We consider λ at time t to be a scalar function defined over Γ_0 , i.e. $\lambda(t) : \Gamma_0 \rightarrow \mathbb{R}$. $\lambda(t)$ is evaluated on Γ_t by composing it with the inverse motion of the structure midsurface. We represent λ in computations as a set of scalar values stored at the quadrature points of Γ_0 , as defined in Section 3.4. This can be viewed either as leaving \mathcal{S}_ℓ and \mathcal{V}_ℓ undiscretized and incurring a quadrature error, or as collocating the fluid–structure kinematic constraint at quadrature points.

3.5. Semi-implicit time integration

This section completes the FSI discretization by specifying a semi-implicit time-marching procedure: we integrate the fluid–structure penalty coupling implicitly, while updating the Lagrange multiplier for the no-penetration constraint explicitly. Summarizing [39, Section 3], we compute approximate solutions at a countable set of time levels, indexed by n and separated by time steps of size Δt . At time level n , the discrete fluid velocity is defined by a vector of coefficients \mathbf{U}^n , the fluid acceleration by $\dot{\mathbf{U}}^n$, the fluid pressure by \mathbf{P}^n , and the structure displacement, velocity, and acceleration by \mathbf{Y}^n , $\dot{\mathbf{Y}}^n$, and $\ddot{\mathbf{Y}}^n$, respectively. We refer to the multiplier at time level n as $\lambda^n : \Gamma_0 \rightarrow \mathbb{R}$, represented discretely as discussed in Section 3.4. Considering the solution variables at time level n known, we first construct a system of equations for all $(n + 1)$ -level unknowns, excluding λ^{n+1} , which we initially set equal to λ^n :

$$\text{Res}\left(\mathbf{U}^{n+\alpha_f}, \dot{\mathbf{U}}^{n+\alpha_m}, \mathbf{Y}^{n+\alpha_f}, \dot{\mathbf{Y}}^{n+\alpha_f}, \ddot{\mathbf{Y}}^{n+\alpha_m}, \mathbf{P}^{n+1}\right) = \mathbf{0}, \quad (24)$$

where $n + \alpha_m$ and $n + \alpha_f$ are intermediate time levels of the generalized- α time integration scheme [48, 49] that interpolate between levels n and $n + 1$. This integration scheme is defined more precisely by [50], in the context of FSI time integration and using the same notation. Following [50], we consider a subset of generalized- α methods, parameterized by a single scalar, $\rho_\infty \in [0, 1]$, which controls numerical damping. Following [20], all computations in this paper use $\rho_\infty = 0.5$. $\text{Res}(\dots)$ is the nonlinear residual corresponding to the discretization of Eq. (20) with $\delta\lambda \equiv 0$ and λ held fixed at λ^n . We solve the nonlinear penalty-coupled problem (24) using block iteration [32, Section 4.6], then compute the Lagrange multiplier at time level $n + 1$ explicitly:

$$\lambda^{n+1} = \frac{1}{1+r} \left(\lambda^n + \beta \left((\mathbf{u}_1^h)^{n+\alpha_f} - (\mathbf{u}_2^h)^{n+\alpha_f} \right) \cdot \mathbf{n}_2^{n+\alpha_f} \right), \quad (25)$$

In Eq. (25), $(\mathbf{u}_1^h)^{n+\alpha_f}$ is the fluid velocity defined by coefficients $\mathbf{U}^{n+\alpha_f}$, $(\mathbf{u}_2^h)^{n+\alpha_f}$ is the structure velocity defined by coefficients $\dot{\mathbf{Y}}^{n+\alpha_f}$, and $\mathbf{n}_2^{n+\alpha_f}$ is the normal to $\Gamma_{t+\alpha_f}$, as determined by the displacement coefficients $\mathbf{Y}^{n+\alpha_f}$. We studied the energetic stability of this algorithm using a linearized model problem in [32] and analyzed convergence for linear parabolic problems with immersed boundaries in [51], showing unconditional stability for the case of $\beta = \tau_{\text{NOR}}^B$.

4. Parametric design optimization with surrogate management framework

As mentioned in Section 1, this work expands on the isogeometric design optimization framework proposed by [23], which seeks to perform parametric optimization of an engineering design using IGA. However, this work differs in that it also incorporates relatively computationally expensive FSI analysis into each design-specific cost function evaluation. This increased computational expense necessitates the use of a more efficient optimization method; traditional pattern search algorithms generally require a relatively large number of function evaluations, as was the case in [23].

4.1. PDE-constrained optimization problem

As in [23], we consider the abstract problem of PDE-constrained optimization:

$$\begin{aligned}
 & \text{minimize} && \mathcal{J}(\mathbf{s}, \mathbf{x}) \\
 & \text{subject to} && \mathcal{B}(\mathbf{s}, \mathbf{x}) = \mathbf{0} , \\
 & && \mathbf{x} \in \mathbf{\Omega} , \\
 & && G_i(\mathbf{s}, \mathbf{x}) \leq \mathbf{0}, \quad i = 1, \dots, p , \\
 & && H_j(\mathbf{s}, \mathbf{x}) = \mathbf{0}, \quad j = 1, \dots, q ,
 \end{aligned} \tag{26}$$

where finding \mathbf{s} such that $\mathcal{B}(\mathbf{s}, \mathbf{x}) = \mathbf{0}$ is an instance of the FSI problem defined in Section 2, where the geometry of Γ_0 is determined by the design parameters \mathbf{x} and we denote all solution fields by \mathbf{s} for concise notation. $\mathbf{\Omega}$ is a Cartesian product of intervals delimiting the range of each design variable in the parametric design space while the functions $\{G_i\}_{i=1}^p$ and $\{H_i\}_{j=1}^q$ define inequality and equality constraints on the PDE solution and design variables.

This problem can be approached as-is, using so-called “full space” or “all-at-once” methods [52, 53], but, as reviewed in the cited references, these methods become prohibitively costly for problems involving unsteady PDE systems, such as the model of Section 2. A more popular methodology for such problems is to eliminate the PDE constraint, to obtain a so-called “reduced space” or “black box” method, which can then be approached with general purpose optimization algorithms. Using the assumption that the PDE system constraining the solution variables is well posed, we define an abstract solution operator \mathcal{S} that maps design parameters to PDE solutions, so that $\mathbf{s} = \mathcal{S}(\mathbf{x})$. Then we may re-write (26) as a generic constrained optimization problem:

$$\begin{aligned}
 & \text{minimize} && J(\mathbf{x}) \\
 & \text{subject to} && \mathbf{x} \in \mathbf{\Omega} , \\
 & && g_i(\mathbf{x}) \leq \mathbf{0}, \quad i = 1, \dots, p , \\
 & && h_j(\mathbf{x}) = \mathbf{0}, \quad j = 1, \dots, q ,
 \end{aligned} \tag{27}$$

where we have defined the new cost function $J(\mathbf{x}) = \mathcal{J}(\mathcal{S}(\mathbf{x}), \mathbf{x})$ and constraints $g_i(\mathbf{x}) = G_i(\mathcal{S}(\mathbf{x}), \mathbf{x})$ and $h_j(\mathbf{x}) = H_j(\mathcal{S}(\mathbf{x}), \mathbf{x})$ to formally eliminate the PDE and greatly reduce the dimensionality of the solution space. As a consequence, the evaluation of the cost function has become much more computationally expensive, requiring the solution of a PDE system, to evaluate $\mathcal{S}(\mathbf{x})$. In this work, we compute $\mathbf{s} = \mathcal{S}(\mathbf{x})$ approximately, using the immersogeometric FSI simulation technology of Section 3. We are therefore optimizing an approximation to J , but, for notational simplicity, we follow [52] in not distinguishing between the infinite-dimensional and discretized settings for \mathbf{s} , as there is little risk of confusion in the present case.

A variety of methods exist for ensuring that the constraints are satisfied. In this work, the exterior penalty method [54] is used and a general optimization problem, such as (27), can be approximated by unconstrained minimization of a pseudo objective function

$$J_{\text{pseudo}}(\mathbf{x}, r_1, r_2) = J(\mathbf{x}) + r_1 \sum_{i=1}^p \max(0, g_i(\mathbf{x}))^2 + r_2 \sum_{j=1}^q h_j(\mathbf{x})^2, \quad (28)$$

where $r_1 > 0$ and $r_2 > 0$ are penalty parameters. The inequality and equality penalty terms, $\max(0, g_i(\mathbf{x}))^2$ and $h_j(\mathbf{x})^2$, respectively, are zero when the constraints are satisfied. The penalty parameters must be properly specified in order to obtain a high quality result without violating the constraints. If penalty parameters are relatively high, the possibility of obtaining a final result that violates constraints is low, but the optimization problem is more difficult to solve. A typical approach to manage the trade-off between constraint satisfaction and computational cost is as follows. Initially, penalty parameters are set to zero, and then a series of optimizations are performed. If the optimization result violates any constraints, an increment is added to the associated penalty parameter. This procedure repeats until all constraints are satisfied (to within acceptable tolerances) by the optimum.

4.2. SMF algorithm

The surrogate management framework (SMF) was introduced by [25] and has been adapted for optimization of aeroacoustic [26, 55, 56] and cardiovascular [27–29, 57–59] problems. The SMF is a derivative-free method conceived with the intent of rigorously optimizing engineering design problems that require evaluations of expensive functions with little or no gradient information, for which the use of traditional optimization methods is impractical. The SMF is a mesh-based technique and features two fundamental strategies: the use of surrogate function as a predictive tool to enable global exploration of the design space and rapid identification of promising regions, and the employment of pattern-search methods for a local grid search to ensure convergence to at least a local minimum.

The SMF algorithm for an unconstrained minimization problem with a (pseudo) cost function

J consists of SEARCH steps and POLL steps. All points where the cost function J is evaluated in the SMF algorithm must lie on a mesh M , a variable-density mesh existing within the design space \mathbb{R}^n . In a SEARCH step, the Kriging method of interpolation is used to construct a surrogate function using all previously performed cost function evaluations. The initial set of design variables is generated via Latin hypercube sampling (LHS), which is shown to be an improvement over random sampling [60]. The search step locates minima of the Kriging surrogate function analytically, then identifies a set of trial points in the design space that are the closest mesh points to the minima of the surrogate. The cost function can be evaluated at the trial points using FSI simulations. If an improved cost function value is obtained at one of the trial points, the SEARCH is considered successful and another SEARCH step is performed. If no improvement is achieved by a SEARCH step, the SEARCH is considered unsuccessful and a POLL step is performed.

In a POLL step, a set of design points neighboring the current best design within M are selected and evaluated. Various methods for generating poll sets can be used; generalized pattern search (GPS) polling being a prevalent option. In this work, however, as was first demonstrated in CFD applications in [27, 56], poll sets are generated via mesh adaptive direct search (MADS) [61] which features an increasing number of possible search directions as the mesh M becomes more fine. In the context of MADS, a set of $n + 1$ POLL points are required to generate a positive basis, where n is the number of design variables. Additional details about MADS and an associated proof of convergence to local minima can be found in [61]. We note that in a POLL step, as soon as a POLL point that produces an improved cost function value is found, evaluating the remainder of the POLL points is no longer necessary; a SEARCH step is instead performed. This reduces the total number of (expensive) cost function evaluations.

Let M_0 denote the set of points in the initial mesh, M_k denote the mesh at iteration k , and \mathbf{x}_k be the current best point. The SMF algorithm may be summarized as: generate initial sampling points on the mesh M_0 and evaluate J at these points, then,

1. SEARCH

- (a) Construct the surrogate model using all previous evaluations of J .
- (b) Perform optimization on the surrogate to identify a finite set T_k of trial points at the minimizing locations.
- (c) Evaluate $J(\mathbf{x}_{\text{trial}})$ for all trial points $\mathbf{x}_{\text{trial}} \in T_k \subset M_k$.
- (d) If, for any trial point $\mathbf{x}_{\text{trial}}$ in T_k , $J(\mathbf{x}_{\text{trial}}) < J(\mathbf{x}_k)$, a lower cost function value has been found, and the SEARCH is *successful*. Increment k and go back to (a).
- (e) Else, if no trial point in T_k improves the cost function, SEARCH is *unsuccessful*. Increment k and go to POLL.

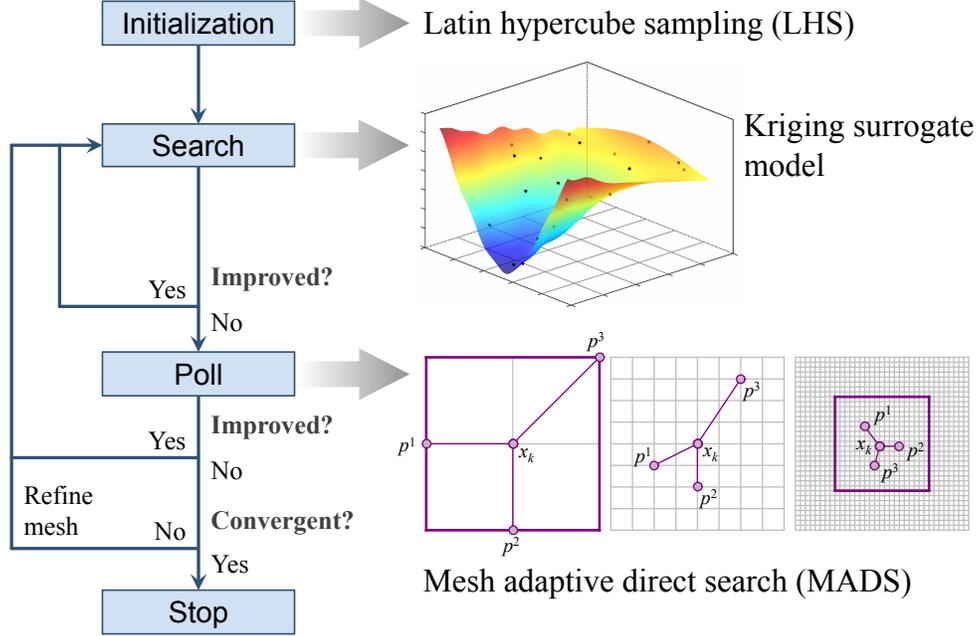


Figure 2: Surrogate management framework optimization algorithm, showing a set of initial data points generated using Latin hypercube sampling (LHS), a Kriging surrogate model constructed for the search step, and the mesh adaptive direct search (MADS) used for the poll step.

2. POLL

- (a) Choose a set of positive spanning directions, and form the poll set X_k as the set of mesh points adjacent to \mathbf{x}_k in these directions.
- (b) If $J(\mathbf{x}_{\text{poll}}) < J(\mathbf{x}_k)$ for any point $\mathbf{x}_{\text{poll}} \in X_k \subset M_k$, then a lower cost function has been found and the POLL is *successful*. Increment k and go to SEARCH.
- (c) Else, if no point in X_k improves the cost function, POLL is *unsuccessful*.
 - i. If convergence criteria are satisfied, a converged solution has been found. STOP.
 - ii. If convergence criteria are not satisfied, refine mesh. Increment k and go to SEARCH.

The algorithm is also represented as a flowchart shown in Figure 2.

4.3. Immersogeometric PDE solution operator

The total cost of optimization will be dominated by evaluations of the (discrete) PDE solution operator, \mathcal{S} . SMF allows us to reduce the number of such evaluations to a tractable level, but the choice of an operator $\mathcal{S}(\mathbf{x})$ mapping design variables \mathbf{x} to an approximate PDE solution \mathbf{s} remains critical for practical design optimization. As explained in Section 3, the greatest level of per-degree-of-freedom accuracy in \mathcal{S} would be obtained by using a boundary-fitted numerical method. However, in that case, each evaluation of $\mathcal{S}(\mathbf{x})$ involves not only the solution of the discretized PDE, but the generation of a boundary-fitted mesh for the fluid domain, as determined by the

design parameters \mathbf{x} . Ensuring that good quality meshes can be generated automatically for all \mathbf{x} in the design space Ω remains a challenging problem [62–66], often requiring manual intervention and thus reducing the overall efficiency of the optimization framework. However, by using the immersogeometric analysis methods reviewed in Section 3, we eliminate the difficulties associated with grid generation. The structure’s design geometry acts directly as the isogeometric mesh for the structure subproblem, while the fluid mesh into which this geometry is immersed is independent of the design parameters \mathbf{x} . For a moderate decrease in the efficiency of approximating PDE solutions, we can completely circumvent the additional computational cost and potential failures in the evaluation of $\mathcal{S}(\mathbf{x})$ due to mesh-generation procedures that may not accommodate every $\mathbf{x} \in \Omega$. We walk through the immersogeometric discretization of a parametrically-designed FSI system (a water twister) in Section 5.1.

5. Water twister optimization

This section combines surrogate modeling and immersogeometric analysis, through the FSI design optimization framework of Section 4, to efficiently and automatically optimize water twister blade shape. To permit direct comparison with earlier academic studies on water twister modeling [13, 15], we define our parametric design space in terms of modifications to a model first analyzed at Virginia Polytechnic Institute (Virginia Tech), which we call the VT model [13]. Schematics for this device are provided in Figure 1. In Section 5.1, we construct and discretize an instance of the FSI problem stated in Section 2, to model the VT water twister. To verify that the immersogeometric computations approximate the problem of Section 2 with sufficient accuracy for the problem at hand, Section 5.2 compares immersogeometric CFD and FSI simulations of the baseline VT design with the earlier boundary-fitted computations of [15]. In Section 5.3, we describe our parameterized space of modifications to the VT rotor blade and stator vane geometry. Sections 5.4 and 5.6 discuss how we search through this space to arrive at an optimized design.

5.1. Modeling and immersogeometric discretization of the VT water twister

Immersogeometric CFD and FSI simulations of the VT water twister shown in Figure 1 are performed. A geometric model that contains all important structural components described in [15] is shown in Figure 3. The key geometric parameters and dimensions of the rotor blade and stator vane are summarized in Figure 4 and Table 1. The casing, middle disk, and stator vanes are assumed to be rigid. The rotor blades are modeled as flexible shell structures, and linear tetrahedral elements are used to build the fluid-mechanics-domain mesh shown in Figure 5. The flexible blades, shown in Figure 6, are assumed to be clamped to the rotor hub and middle disk. According to [15], shell discretizations comprised of cubic NURBS (weights are set to one) and around 300 control points provide converged structural properties. Thus, the same discretization is used for

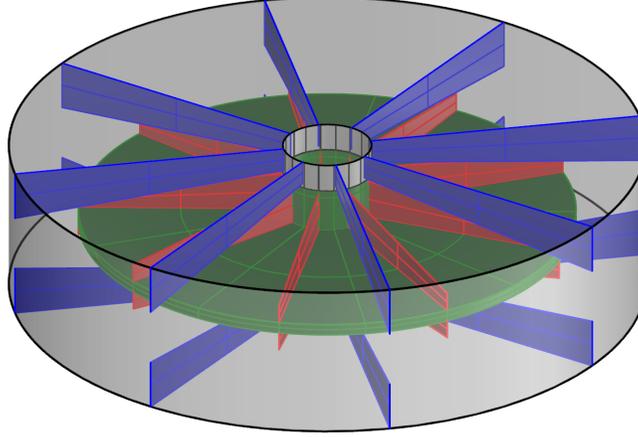


Figure 3: NURBS-based model of the VT water twister, including rotor, stator, and casing.

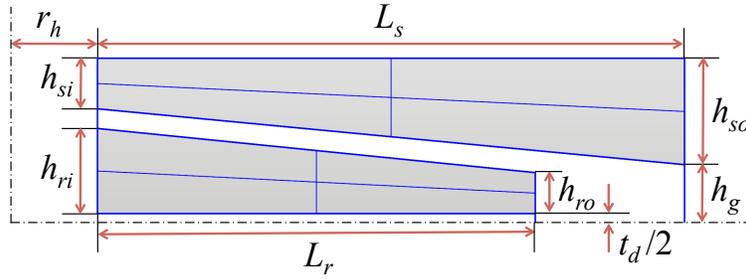


Figure 4: The key geometric parameters of the rotor blade and stator vane. The dimensions are given in Table 1.

Table 1: The dimensions of the rotor blade and stator vane. The thickness of the rotor blades is 0.348 inch (8.84×10^{-3} m).

Parameter	Symbol	Unit (in)	Unit (m)
Hub radius	r_h	3.880	0.0986
Stator length	L_s	23.96	0.6086
Stator inner height	h_{si}	2.066	0.0525
Stator outer height	h_{so}	4.350	0.1105
Rotor length	L_r	17.87	0.4539
Rotor inner height	h_{ri}	3.480	0.0884
Rotor outer height	h_{ro}	1.675	0.0425
Disk thickness	t_d	1.000	0.0254
Gap height (half)	h_g	2.499	0.0635

the shell structure in this study. Detailed procedures regarding blade shape generation are given in Section 5.3.

Remark 2. While most physical quantities in this paper are stated using SI units, as is conventional in the computational mechanics literature, some length scales are given in inches, to facilitate geometrical comparisons with the VT water twister of [13], in which the model dimension was reported in United States customary units.

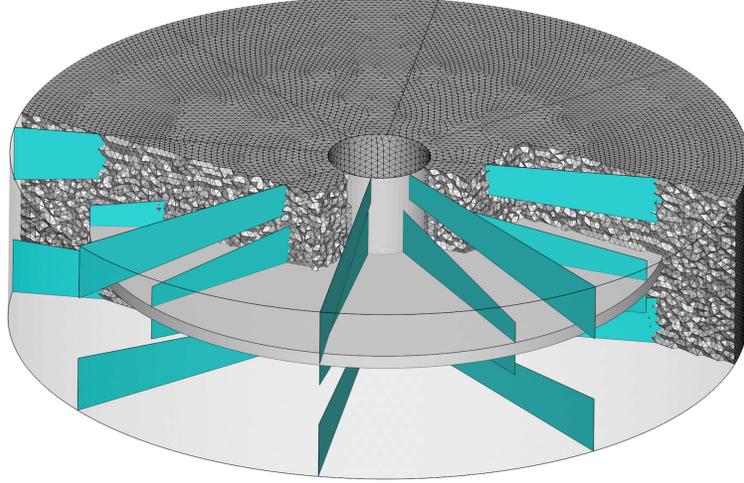


Figure 5: A cut through the tetrahedral fluid-mechanics-domain mesh used in immersogeometric CFD and FSI analysis of the VT water twister. A uniform mesh size of 0.5 inch is used.

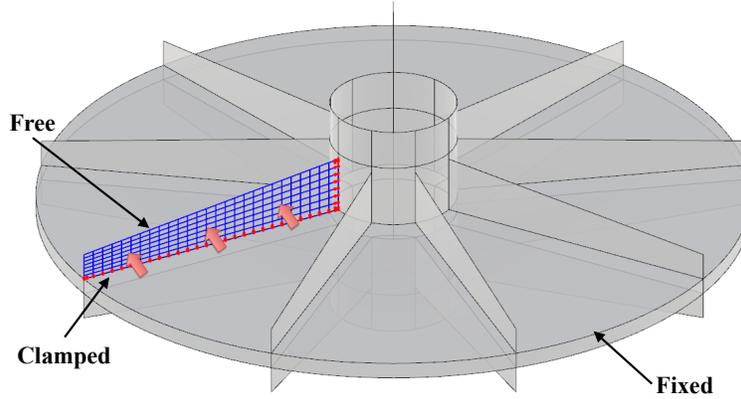


Figure 6: Rotor blade with clamped boundary conditions. Red points represent two rows of control points along the edge. These points are fixed for clamped boundary conditions. The red arrows indicate the pressure force applied on a single blade in the structural mechanics simulations, or fluid tractions acting on the blades in FSI simulations.

Dirichlet boundary conditions for the fluid velocity on the stator, middle disk, and inner cylinder are applied strongly at all boundaries of Ω_1 . The portions of $\partial\Omega_1$ that are colored gray in Figure 3 are subject to the boundary condition $\mathbf{u}_1 = \mathbf{0}$. The portions that are colored green are subject to

$$\mathbf{u}_1(\mathbf{x}) = \mathbf{u}_{\text{rot}}(\mathbf{x}) = \omega \mathbf{e}_\theta(\mathbf{x}), \quad (29)$$

where ω is the angular velocity of the prescribed rotation rate and $\mathbf{e}_\theta(\mathbf{x})$ is the (non-normalized) covariant basis vector corresponding to the angular coordinate θ of a standard cylindrical polar coordinate system centered on the rotor shaft. In the computational model, the Dirichlet boundary condition on $\partial\Omega_1$ is approximated by prescribing the velocity \mathbf{u}_1^h of boundary nodes of the fluid mesh to equal either \mathbf{u}_{rot} evaluated at the nodal location or $\mathbf{0}$, depending on what portion of the boundary the node belongs to. (This results in a linear interpolation of the discontinuous Dirichlet

boundary data along the curve where the rotating (green in Figure 3) and stationary (gray) portions of the boundary meet.) The clamped boundary condition on the attached edges of the rotor blades (shown in red in Figure 3) is applied strongly by fixing two rows of control points from the edges.

The steel rotor blades are modeled as a St. Venant–Kirchhoff material with Young’s modulus of 200 GPa, Poisson’s ratio of 0.27, and mass density of 8000 kg/m³. The thickness of the blades is 8.84×10^{-3} m. The quadrature rule that we use to evaluate integrals over Γ_t in both the structure subproblem and for the fluid–structure coupling (cf. Section 3.4) consists of an 8×8 Gaussian rule on each Bézier element. This is more quadrature points than is typically needed for accurate shell structure analysis using cubic NURBS trial and test spaces, but a higher density of quadrature points is needed for accurate integration of traces of fluid velocity on Γ_t , because the fluid mesh consists of smaller elements than the structure mesh.

The fluid density and viscosity used in this work are 9.982×10^2 kg/m³ and 1.003×10^{-3} kg/(m s), respectively. Different rotor rotational speeds are considered in the simulations, from 200 rpm to 800 rpm, with $\Delta t = 1.25 \times 10^{-4}$ s. In each case, a total of 12 revolutions are simulated. The first two revolutions, which are considered to be transient, are not included to compute time-average quantities. C_{TAN} and $C_{\text{NOR}}^{(i)}$ in Eqs. (21) and (22) are set to 10.0 and 1.0, respectively. In the CFD simulations, the setup is the same as for FSI simulations except that the rotor blades are considered to be rigid. The FSI computations are carried out on a Linux cluster. Each compute node is configured with two Intel Xeon E5-2680 2.7GHz 8-core processors and 32GB memory. The mesh is partitioned into 240 subdomains using METIS [67]. The parallel implementation of the methodology in this work is described in [68]. About 4.5 hours are needed to simulate one revolution.

5.2. Comparison with boundary-fitted simulations

The accuracy of immersogeometric CFD and FSI results is verified by comparing to results computed using the boundary-fitted method presented in [15]. Performing such comparisons before applying immersogeometric analysis in new application domains verifies that the PDE system in question is being approximated with sufficient accuracy to extract useful quantities of interest.¹ We support the validity of our modeling assumptions by comparing results from both boundary-fitted and immersogeometric simulations with an experimentally-calibrated analytical model for predicting torque at different rotation rates, provided by [13]. CFD simulations are computed using rotational speeds from 200 rpm to 800 rpm with mesh sizes of 1.0 and 0.5 inches used in boundary-fitted simulations and 0.5 inch used in immersogeometric simulations. An immersogeometric FSI simulation of the baseline design is computed at the highest rotational rate of 800 rpm, at which

¹Agreement with physical experiments alone is not sufficient to verify accuracy of numerical methods, as it entangles modeling and discretization error; see [69, Footnote 4].)

Table 2: Element and time-step sizes employed in the CFD simulations. The immersogeometric setup is also used in the FSI simulations.

	Boundary-fitted		Immersogeometric
Element size (in)	1.0	0.5	0.5
Time step size (s)	2.5×10^{-4}	1.25×10^{-4}	1.25×10^{-4}
Number of elements	314,462	2,008,047	2,621,079

Table 3: Rotor-torque values comparison.

	Torque (N m)			
	Boundary-fitted CFD		Immersogeometric CFD	Experimental data
	1.0 in	0.5 in		
200 rpm	8,780	8,186	9,354	8,400
400 rpm	35,971	33,951	36,729	33,600
600 rpm	81,657	77,363	80,353	75,600
800 rpm	145,132	139,569	141,884	134,400

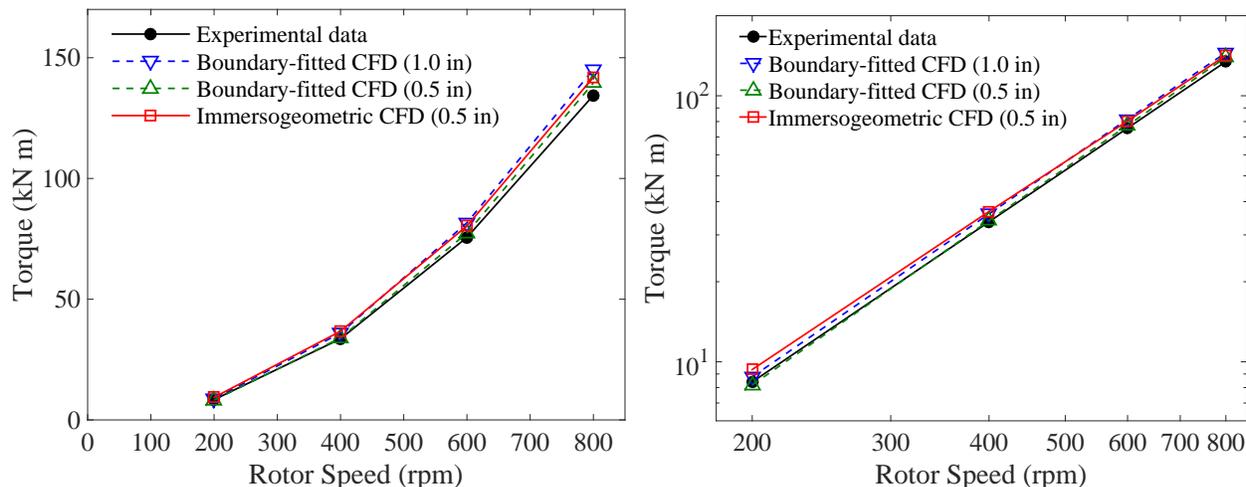


Figure 7: CFD simulation results of the VT water twister with rotational speeds from 200 rpm to 800 rpm using a boundary-fitted method and the immersogeometric method. The results are compared with data obtained using an experimentally-calibrated analytical model [13] and are plotted on a linear scale (left) and a logarithmic scale (right).

structural deformations are most pronounced. As described in Section 5.1, time-averaged hydrodynamic rotor torque is computed based on the time interval between the 3rd and 12th revolutions. The number of elements and corresponding time-step sizes are shown in Table 2. Accuracy of the CFD simulations is tested by comparing time-averaged torque. Rotor torque given by the various CFD cases is presented in Table 3 and Figure 7. The simulation results and the experimental data are in good agreement.

The accuracy of FSI simulations is studied by qualitatively comparing blade deflection² and

²We use the term “deflection” to indicate displacement from a rigid rotation of Γ_0 . This is more informative than

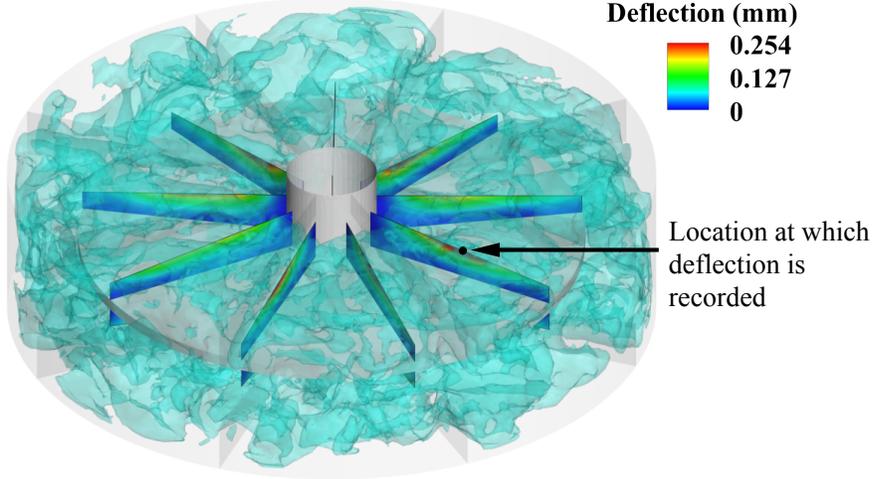


Figure 8: Immersogeometric FSI simulation results represented by rotor blade deflection contours superimposed with flow speed isosurfaces.

von Mises stress distribution, and quantitatively comparing the frequency of rotor blade vibrations in immersogeometric and boundary-fitted computations of the 800 rpm case. Figure 8 shows rotor blade deflection contours and flow speed isosurfaces. In Figure 8, one can identify the location on the blade that experiences the largest deflection (the red region). The deflection time history of this location is collected and the associated frequency profile computed via Fast Fourier Transform (FFT) is shown in Figure 9. Figure 10 illustrates both the distribution of von Mises stress and the scaled deflection of the rotor blades. Note that von Mises stress distribution of Figure 10a is calculated based on the FSI result in [15].

In Figure 9, frequencies with three most significant amplitudes from immersogeometric FSI simulation are in very good agreement with those from boundary-fitted FSI simulation. This implies the overall dynamical behavior of the record location is similar in both cases. Contours shown in Figure 10 indicate that von Mises stress distributions are qualitatively similar in both simulations.

5.3. Parametric design space

The blade shape shown in Figure 11 is modeled as single NURBS surface. Quadratic NURBS curves are employed to model the four edges of a blade. For a p th-degree NURBS curve with n_C control points, the nonuniform knot vector will be $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n_k}\}$, where $\xi_i \in \mathbb{R}$ is the i th knot, i is the knot index, $i = 1, 2, \dots, n_k$, and $n_k = n_C + p + 1$ is the number of knots in the knot vector. Thus, each of the four edge curves of the blade geometry are described by a knot vector

direct displacement from Γ_0 in the context of predicting material damage, since the rigid rotation does not contribute to strain in the material.

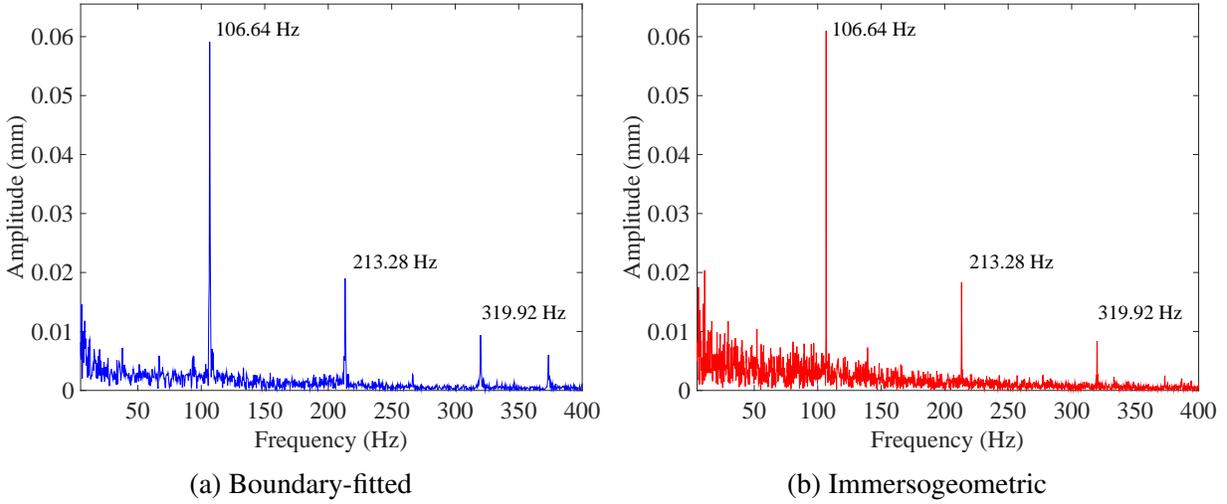


Figure 9: Frequencies of blade vibrations. The record location is indicated in Figure 8.

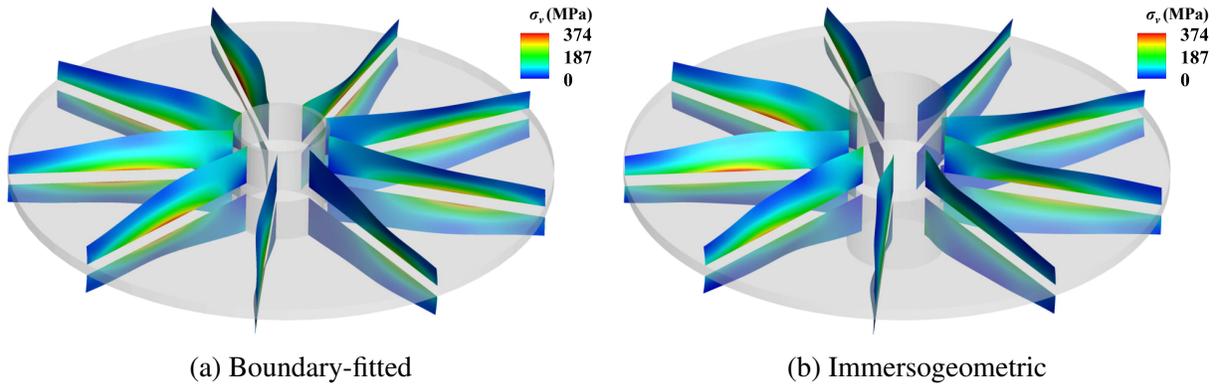


Figure 10: Contours of von Mises stress. The deflection is scaled by 100 times. The time frames in which maximum von Mises stress occurs are selected for each case. The boundary-fitted result is calculated using the FSI solutions from [15].

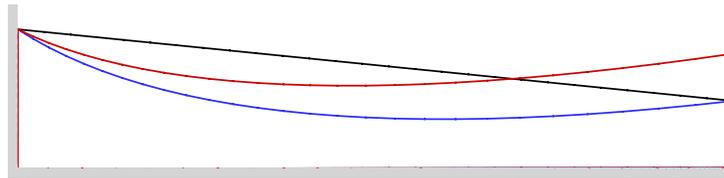


Figure 11: Possible rotor blade edge curve shapes. The gray areas on the left and the bottom of the blade represent the rotor hub and the middle disk, respectively. Black lines indicate the baseline shape. Blue lines indicate the curved upper edge using two fixed end points. Red lines depict the curved upper edge using a single fixed end point.

$\Xi = \{0, 0, 0, 1, 1, 1\}$ and three control points. The most straightforward way to alter the blade geometry is to change the position of control points. However, which control points are given the freedom to move should be guided both by *a priori* knowledge about the behavior of the baseline design in preliminary analyses and by the fundamental optimization goals.

Using the stress distribution in Figure 10, one can clearly identify a location of von Mises

stress concentration at the T-joint between the blades and the middle disk. This concentration could increase the risk of fatigue, the primary problem we wish to address. Figure 10 implies that the stress concentration is caused by bending of the center portion of the rotor blade, and the bending is caused by the local hydrodynamic force. Thus, we assume that, if we reduce the area of that section of the blade, we can reduce the local hydrodynamic force, in turn reducing the bending moment. This can be achieved by decreasing the height of the middle control point of the topmost rotor blade edge, that is represented by a quadratic NURBS curve. It yields a curve like the blue curve shown in Figure 11. However, this change may result in lower overall rotor torque since the total blade area is reduced. Because torque is proportional to radius, we seek to increase the blade area towards the tip of the blade, creating more torque-generation potential. This is achieved by allowing vertical movement of the control point on the tip of the topmost rotor edge curve, creating a curve like the red curve in Figure 11.

Because the shape of the rotor blade is variable and because we seek to maintain a constant gap between rotor blade and stator vane, we devise a strategy for modifying the shape of the stator according to the rotor shape manipulation. Firstly, we wish to mirror control point manipulations of the rotor blade with corresponding control point manipulations on the stator vane. We therefore construct the stator vane's lower edge curve using control points positioned directly above the rotor blade's control points. Any translation vector used on a rotor blade control point is used on the corresponding stator vane control point. Because the stator vane's lower edge curve and the rotor's upper edge curve are different lengths, to ensure that the gap between the rotor blade and stator vane remains constant throughout control point movement, we create a C^0 continuity, or kink, in the stator vane edge curve. This kink creates a control point corresponding to that of the rotor blade edge tip. The baseline geometry of the rotor blade and stator vane are shown in Figure 12a whereas geometries that were modified using this strategy are shown in Figure 12b.

Figure 12 also illustrates details of the design modification strategy that has been generally described thus far. In Figure 12, control points are classified as two types: red, square points, denoting fixed control points, and blue points of either circular or triangular shape, denoting moving control points. The red control points are fixed since they either are attached to the hub or disk or belong to an inactive region. For the moving, blue control points, the two different shapes indicate different moving algorithms. The dark blue circular control points of the rotor will move together in the same direction and magnitude as the closest corresponding point on the stator. A light blue triangular control point will move in the same direction as its neighboring dark blue circular control point but with half the magnitude. This approach produces a better knot span distribution than would be achieved by moving only the edge points. The center control points (A) are allowed to move both vertically and horizontally, corresponding to design variables x_1 and x_2 , respectively, whereas the control points on the outer edge of the rotor blade (B) are only allowed to move verti-

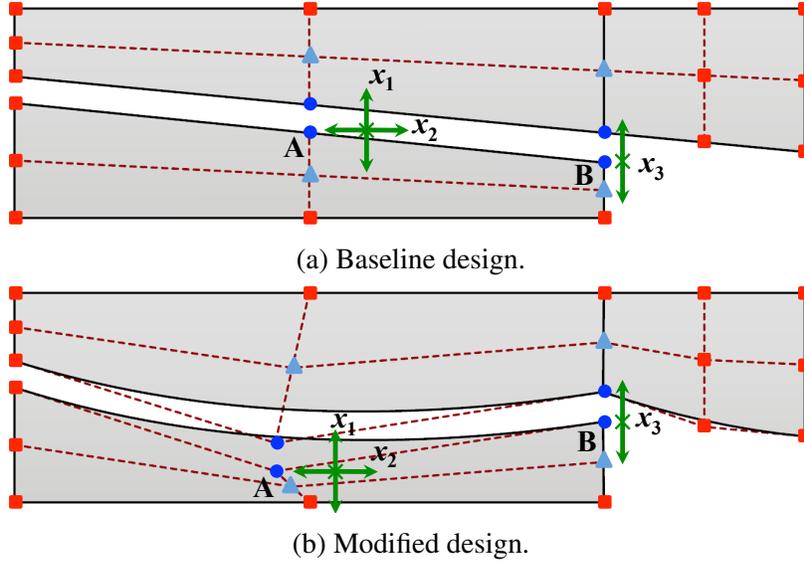


Figure 12: Design modification strategy used for rotor blades and stator vanes, depicted (a) before and (b) after design modification. Design variables x_1 and x_2 define the relative location of control point A; design variable x_3 defines the relative location of control point B. For the baseline design, $x_1 = x_2 = x_3 = 0$. If x_1 and x_3 are positive, control points A and B move up. If x_2 is positive, control point A moves right.

cally, corresponding to design variable x_3 , in order to maintain a fixed rotor blade radius. Once the new geometry is constructed, refinement procedures like p - and h -refinements are used to achieve the mesh density required for FSI simulation.

Blade thickness is also an important factor in structural performance. In this work, the thickness is assumed to be uniformly distributed. Non-uniform plate thickness represents increased manufacturing complexity. If thickness is considered an unconstrained design variable, thickness will dominate the optimization process, producing a design with very thick blades. In this work, the thickness is constrained by constraining the mass. To avoid changing too much of the dynamical properties of the rotor, we keep the blade mass the same as that of the baseline design during the optimization process. Given the same material density, the change of thickness is directly calculated from the change of blade area. For example, if the area of a given blade is two times of the baseline blade area, the thickness of the current blade will be 0.5 times of the baseline blade thickness.

In summary, the design modification strategy described in this section allows us to generate a variety of rotor blade designs given values for three design variables. Referring to Figure 12, these design variables are x_1 and x_2 , indicating vertical and horizontal movement of control point A, respectively, and x_3 , indicating vertical movement of control point B. These three values constitute the vector of design variables \mathbf{x} .

5.4. Optimization procedure

As mentioned in section 5.3, our primary objective is to optimize the distribution of von Mises stress along the T-joints between the rotor blades and the middle disk (i.e. the 1D curves along which the (closure of the) blade geometry $\bar{\Gamma}_t$ intersects the portion of $\partial\Omega_1$ associated with the spinning horizontal disk). We do this by minimizing the time-averaged spatial variance of the T-joint von Mises stress,

$$\text{var}(\sigma_v^{\text{T-j}}) = \frac{1}{T_2 - T_1} \int_{T_1}^{T_2} \frac{1}{N_{\text{bid}}} \sum_{i=1}^{N_{\text{bid}}} \left(\frac{1}{|(L_i)_t|} \int_{(L_i)_t} \left(\sigma_v^{\text{T-j}}(\mathbf{x}, t) - \left(\overline{\sigma_v^{\text{T-j}}} \right)_i(t) \right)^2 d\ell \right) dt, \quad (30)$$

where the time average is taken over the interval (T_1, T_2) , N_{bid} is the number of rotor blades, $d\ell$ is the 1D line integral measure (corresponding to the integration variable \mathbf{x}) associated with the T-joint configuration $(L_i)_t$ of the i^{th} blade at time t , $\sigma_v^{\text{T-j}}(\mathbf{x}, t)$ is the von Mises stress at a point \mathbf{x} at time t , and $\left(\overline{\sigma_v^{\text{T-j}}} \right)_i(t)$ is the i^{th} blade's spatial average of T-joint von Mises stress at time t ,

$$\left(\overline{\sigma_v^{\text{T-j}}} \right)_i(t) = \frac{1}{|(L_i)_t|} \int_{(L_i)_t} \sigma_v^{\text{T-j}}(\mathbf{x}, t) d\ell. \quad (31)$$

In addition to minimizing $\text{var}(\sigma_v^{\text{T-j}})$, the maximum von Mises stress along the T-joints and over the time interval (T_1, T_2) , denoted as $\max(\sigma_v^{\text{T-j}})$, and the time-averaged rotor torque, denoted as τ , are constrained, to prevent yielding of the material and deliver consistent device performance. Following the notation of Section 4.1, we define an instance of the problem (26) as follows:

$$\begin{aligned} & \text{minimize} && \mathcal{J}(\mathbf{s}, \mathbf{x}) = \text{var}(\sigma_v^{\text{T-j}}) \\ & \text{subject to} && -0.05 \text{ m} \leq x_1 \leq 0.05 \text{ m}, \\ & && -0.10 \text{ m} \leq x_2 \leq 0.10 \text{ m}, \\ & && -0.05 \text{ m} \leq x_3 \leq 0.05 \text{ m}, \\ & && G_1(\mathbf{s}, \mathbf{x}) = \max(\sigma_v^{\text{T-j}}) - \max((\sigma_v^{\text{T-j}})^{\text{base}}) \leq 0, \\ & && H_1(\mathbf{s}, \mathbf{x}) = \tau - \tau^{\text{base}} = 0, \end{aligned} \quad (32)$$

where $\max((\sigma_v^{\text{T-j}})^{\text{base}})$ is the maximum T-joint von Mises stress in the baseline design and τ^{base} is the time-averaged torque τ of the baseline design, $G_1(\mathbf{s}, \mathbf{x})$ defines the only inequality constraint ($p = 1$), and $H_1(\mathbf{s}, \mathbf{x})$ defines the only equality constraint ($q = 1$). We use an immersogeometric discrete approximation to the PDE solution operator \mathcal{S} to transform this problem into an instance of (27), then define an unconstrained minimization problem with a pseudo cost function of the form (28), to apply the unconstrained SMF algorithm summarized in Section 4.2. The penalty parameters r_1 and r_2 in (28) are determined using the results of trial optimizations using single-

blade structural mechanics simulation.

The functions $H_1(\mathbf{s}, \mathbf{x})$ and $G_1(\mathbf{s}, \mathbf{x})$ are computed by extracting quantities of interest from a discrete solution $\mathbf{s} = \mathcal{S}(\mathbf{x})$. The full solution \mathbf{s} consists of four complete revolutions of the rotor, but, to avoid spurious effects of the initialization procedure, the first two revolutions are ignored, so T_1 in Eq. (31) is the beginning of the third revolution and T_2 is the end of the fourth revolution. Truncation of the simulation in time introduces some uncertainty in time-averaged quantities of interest. It is overwhelmingly improbable that a finite-time simulation would exactly satisfy the equality constraint imposed on the average torque. To estimate an appropriate tolerance for the equality constraint imposed on the time-averaged torque, we simulate the baseline design for 12 revolutions, discard the first two, and compare the average torque from each two-revolution segment to the average torque over revolutions 3–12. We compute, for each $i \in \{3, 5, 7, 9, 11\}$, the relative deviation of the average torque over revolutions i – $(i + 1)$, denoted $\tau_{i,i+1}^{\text{base}}$, from the average torque over revolutions 3–12, denoted $\tau_{3,\dots,12}^{\text{base}}$:

$$\epsilon_{i,i+1} = \left| \frac{\tau_{i,i+1}^{\text{base}} - \tau_{3,\dots,12}^{\text{base}}}{\tau_{3,\dots,12}^{\text{base}}} \right|. \quad (33)$$

The equality constraint $H_1(\mathbf{s}, \mathbf{x}) = 0$ is considered to be satisfied if the torque τ extracted from the solution \mathbf{s} satisfies

$$\left| \frac{\tau - \tau_{3,\dots,12}^{\text{base}}}{\tau_{3,\dots,12}^{\text{base}}} \right| < \epsilon = \max_{i \in \{3,5,7,9,11\}} \epsilon_{i,i+1} = 1.6\%. \quad (34)$$

The rationale behind selecting the tolerance in this manner is that we cannot reasonably insist that a modified rotor’s two-cycle torque average be closer to the baseline design than a two-cycle average computed from the baseline design. Tighter tolerances could be obtained by averaging over more cycles, but this would incur large computational cost for little practical benefit. The relative tolerance of 1.6% is likely dwarfed by the expected operational variations in rotor torque due to differences in the mass and speed of incoming aircraft. The values of all $\{\epsilon_{i,i+1}\}$ are shown above their respective time intervals in Figure 13, which provides the full 12-revolution torque history of the baseline water twister design. Figure 13 also demonstrates the extreme transient behavior following impulsively-started rotation. In reality, the angular acceleration of the rotor occurs gradually, over some finite time scale, as modeled in [15] (see, e.g. [15, Figure 9]). However, to reduce the nontrivial computational costs associated with evaluations of \mathcal{S} in the optimization process, we apply the full rotation rate immediately, to reach a steady torque value at the maximum angular velocity as quickly as possible, then discard the unphysical transient occurring over the first two revolutions.

To explore the importance of including high-fidelity FSI modeling in optimization, we compare optimization using structural mechanics modeling with optimization using FSI modeling. For the

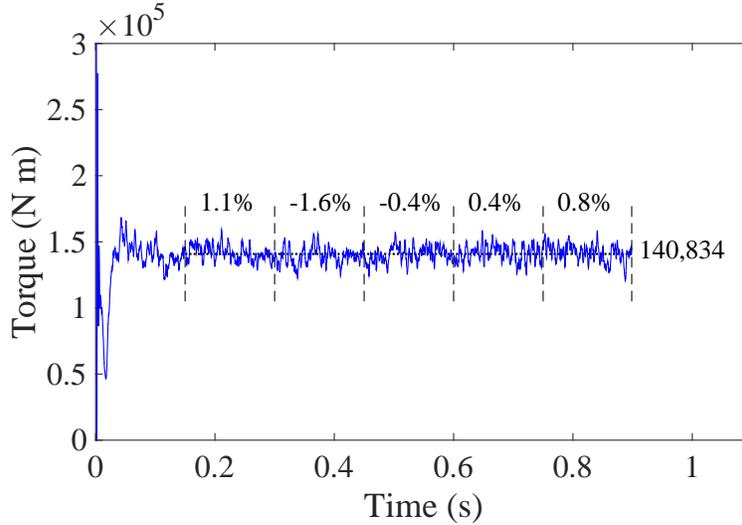


Figure 13: Torque of the baseline water twister design collected from an FSI simulation of 12 rotations. Intervals between the two dashed lines indicate sets of two revolutions. Dotted line shows time-average torque, 140,834 N m, of 10 revolutions. Relative difference of time-average torque of every 2 revolutions and time-average torque of 10 revolutions is presented as percentages.

structural mechanics simulations, we apply the same setup used in [15]. Because the potential for rotor damage is greatest at the highest rotational rate, we focus our optimization on the 800 rpm case. A uniform pressure of 9.58×10^5 Pa is applied to a model of a single rotor blade. Applying this pressure to all blades of the baseline design produces the 800 rpm torque predicted by the experimentally-calibrated analytical model. Observations such as torque and stress are collected from the final configuration. For the FSI simulations, we use immersogeometric FSI simulations with an 800 rpm rotor rotational speed and then collect time-averaged values of torque and stress, discussed above, to evaluate the pseudo objective function.

Because structural mechanics simulations are much faster than FSI simulations, we use a sequence of structural-mechanics-based trial optimizations to determine the smallest effective values of the penalty parameters r_1 and r_2 . Initially, r_1 and r_2 are set to zero. Then an optimization using structural mechanics modeling is performed, and an increment of 0.1 is added to r_1 if $g_1(\mathbf{x}) > 0$ in the final design produced by the optimization and an increment of 0.1 m^{-6} is added to r_2 if $|h_1(\mathbf{x})|$ exceeds the relative tolerance defined in Eq. (34). Optimizations and conditional penalty increases continue in this pattern until reaching a design that satisfies all constraints. The final r_1 and r_2 leading to a feasible optimum are 1 and 2.6 m^{-6} , respectively. These penalty values are then used from the outset in the FSI optimization to avoid repeating this incremental updating procedure with costly FSI model evaluations.

5.5. SMF statistics and progression

We now elaborate on the setup and performance of SMF, in the context of the present application. In both the structural mechanics and FSI optimizations, the initial mesh, M_0 , provided to the algorithm of Section 4.2 divides the n -dimensional design space, where $n = 3$ defined in Section 5.3, uniformly into $100 \times 100 \times 100$ cells. Twelve initial sampling points are selected on M_0 , using LHS; the cost function must be evaluated at all of these points. In each SEARCH step, only one simulation (of the design given by the mesh point that is closest to the minima predicted by the surrogate function) is necessary. In a POLL step, up to $n + 1$ simulations are required. Simulations corresponding to different elements of the poll set X_k (in the notation of Section 4.2) do not depend on one another and can be performed simultaneously, if sufficient parallel computing resources are available. If, for some point \mathbf{x} that is a trial point in a SEARCH step or a member of the poll set in a POLL step, $J(\mathbf{x})$ has already been evaluated, \mathbf{x} is outside the design space, or \mathbf{x} violates a design constraint, then \mathbf{x} is discarded and $J(\mathbf{x})$ is not (re-)evaluated.

Recall, from Section 4.2, that, as soon as a POLL point that produces an improved cost function value is found, evaluating the pseudo-cost function at the remaining POLL points is no longer strictly necessary. In general, the cost function evaluations for a POLL step may be conceptualized as existing in a queue, waiting for computing resources to become available. While there are available parallel computing resources, evaluations are dequeued and assigned resources. Once a completed function evaluation yields improvement, though, the evaluations remaining in the queue are ignored, the POLL step completes, and a new SEARCH step begins. In the computations of this work, computing resources are allocated such that all evaluations of each POLL step can be dequeued at once and executed fully in parallel. For the structural-mechanics-based optimization, each function evaluation is computed in serial, and requires approximately 20 seconds using a MacBook Pro with a 2.5 GHz Intel Core i7 and 16 GB memory. For the FSI optimization, each function evaluation is carried out using 240 processing cores of the Linux cluster described in Section 5.1 and requires about 18 hours of wall clock time.

Figures 14 and 15 show the progress of SMF in the structural mechanics and FSI optimization problems, respectively. The dashed line indicates the pseudo-cost function of the baseline design. The optimal initial design from the LHS is highlighted in green. Red triangles show when design space mesh refinement occurs. Groups of consecutive filled circles indicate POLLing function evaluations that can be queued for parallel execution, while empty circles indicate SEARCHing evaluations that must be done one at a time. Recall from the second paragraph above that redundant or constraint-violating function evaluations are not carried out, so there may be fewer than $n + 1 = 4$ function evaluations in some POLL steps. The overarching conclusion to be drawn from the data presented in Figures 14 and 15 is that SEARCHing for optimum values of surrogate functions is most effective in the early stages of optimization, while more frequent POLLing of the true cost function

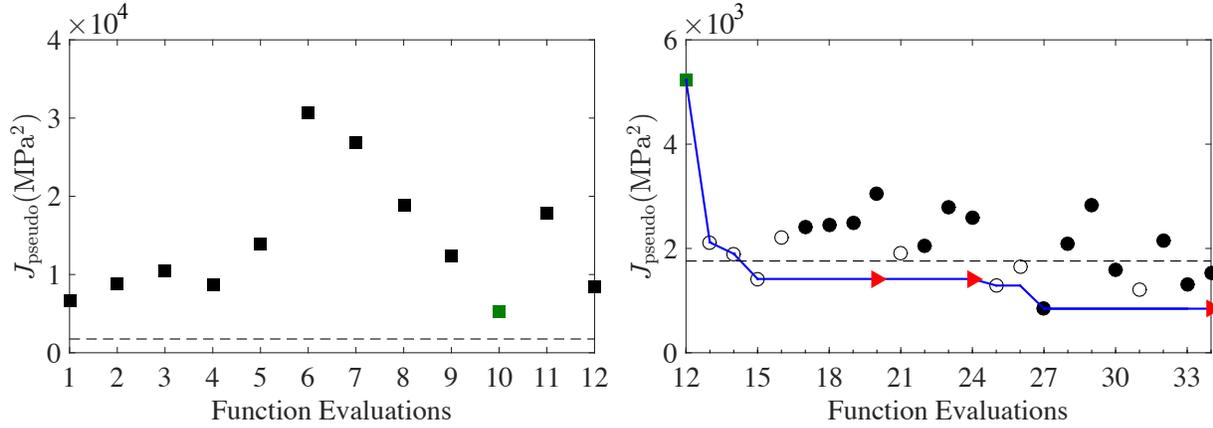


Figure 14: Function evaluations of SMF applied to the structural mechanics problem. Left: Results from the 12 initial designs. The 12 function evaluations can be queued for parallel execution. Right: The progress of SMF from the optimal initial design, highlighted in green. The dashed line indicates the pseudo-cost function of the baseline design. Red triangles show when design space mesh refinement occurs. Groups of consecutive filled circles indicate POLLING function evaluations that can be queued for parallel execution, while empty circles indicate SEARCHING evaluations that must be done one at a time. Note that the vertical axis ranges differ between the two plots.

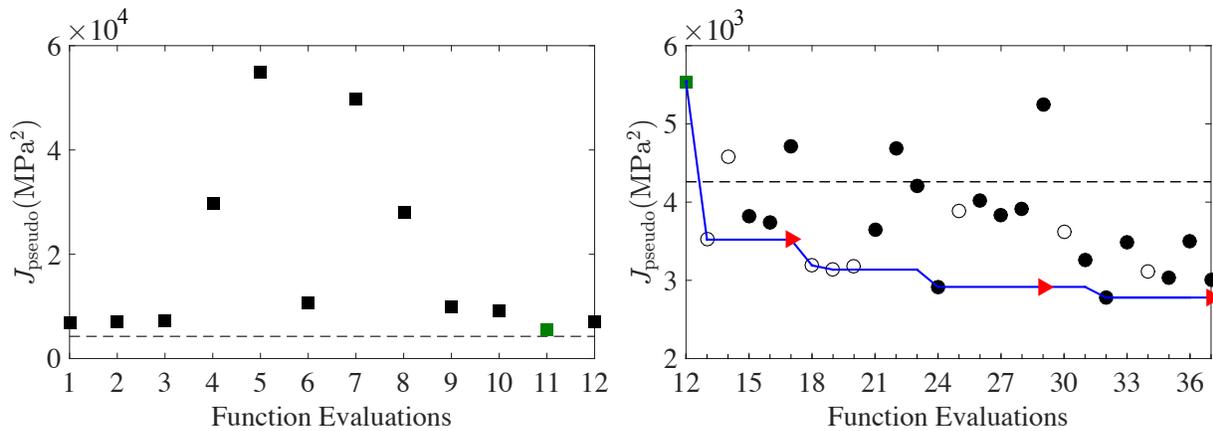


Figure 15: Function evaluations of SMF applied to the FSI problem, plotted according to the conventions described in the caption of Figure 14.

is needed to identify better designs later in the optimization process.

Convergence of the SMF is reached when a local minimizer on the mesh is found and the mesh has been refined to the desired accuracy [27]. In this work, the convergence criteria is set to three refinements of the parameter space mesh. Figures 14 and 15 show that convergence for both cases was reached in less than 37 function evaluations. This demonstrates the efficiency of using SMF for optimizing problems that are computationally expensive to evaluate. Further, recall that function evaluations associated with POLL steps can be executed in parallel so that, given sufficient computing resources, the wall clock time needed for optimization is significantly less than the total number of evaluations multiplied by the time required by a single function evaluation. Wall clock time savings due to parallelization will increase with the dimension of the design space.

For example, one can deduce from Figure 15 that, assuming that all function evaluations require the same length of time and that parallel computing is used whenever possible, the wall clock time required for the FSI optimization reduces to only 14 times that required for a single function evaluation.

5.6. Optimization results and discussion

Figure 16 shows the comparison between the baseline blade and optimized blade using structural-mechanics-based optimization. One can clearly see that, at the T-joint region, the distribution of stress is more uniform on the optimized blade. Additionally, the largest stress on the optimized blade is smaller than the largest stress on baseline blade. Table 4, which provides comparative values between the two designs, shows that the thickness of optimized blade is larger by 4.07%, which implies the area is 4.07% smaller. The difference of area does not significantly affect the torque, but the slight area difference and area redistribution lessens the stress-inducing effect of the fluid. This, in conjunction with increased thickness, reduces the $\text{var}(\sigma_v^{\text{T-j}})$ by about 52% and the $\text{max}(\sigma_v^{\text{T-j}})$ by about 27%, a significant improvement.

We can draw similar conclusions from the FSI-based optimization results, as shown in Figure 17 and Table 5. However, because the hydrodynamic forces are captured much more realistically in the FSI simulations, the location of stress concentration is different. Because the optimizer tends to decrease the area of the blade near the location of stress concentration, the optimized shape using FSI simulations is different from that using structural mechanics simulations. In this case, the thickness is increased by 3.95%, whereas the $\text{var}(\sigma_v^{\text{T-j}})$ is reduced by about 35% and the $\text{max}(\sigma_v^{\text{T-j}})$ by about 25%, also significant improvements. In the new design of blade, stress on upper edge connecting to hub is also significantly reduced. The difference between the design

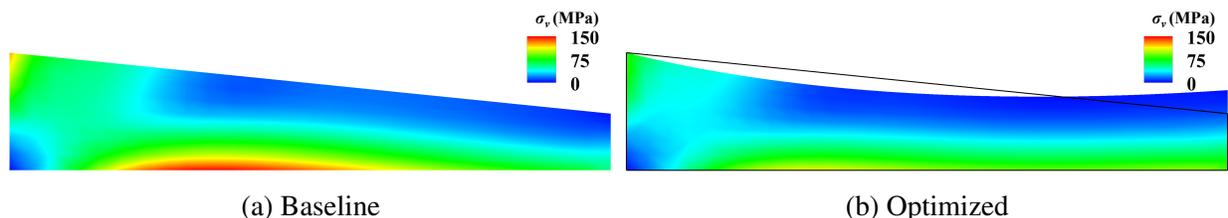


Figure 16: Contours of von Mises stress on baseline design and structural-mechanics-based optimization result.

Table 4: Structural-mechanics-based optimization results. Column of torque in this table represents torque of a single blade. The volume of the blades is constant.

	$\text{var}(\sigma_v^{\text{T-j}})$ (MPa ²)	$\text{max}(\sigma_v^{\text{T-j}})$ (MPa)	Torque (N m)	Thickness (mm)
Baseline	1.76×10^3	1.50×10^2	7,467	8.84
Optimized	8.45×10^2	1.10×10^2	7,475	9.20
Relative difference	-51.99%	-26.67%	-0.11%	+4.07%

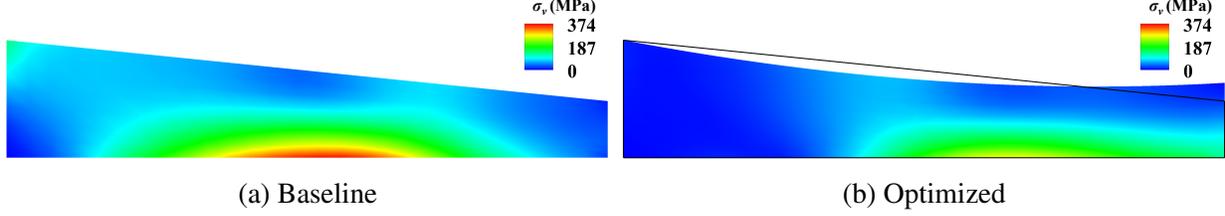


Figure 17: Contours of von Mises stress on baseline design and FSI-based optimization result. The time frames in which maximum von Mises stress occurs are selected for each case.

Table 5: FSI-based optimization results. Column of torque in this table represents torque of a whole rotor. The volume of the blades is constant. $\text{var}(\sigma_v^{\text{T-j}})$ and $\text{max}(\sigma_v^{\text{T-j}})$ obtained from the boundary-fitted FSI simulation of the baseline design are $4.16 \times 10^3 \text{ MPa}^2$ and $3.66 \times 10^2 \text{ MPa}$, respectively. These results are included here for completeness.

	$\text{var}(\sigma_v^{\text{T-j}})$ (MPa^2)	$\text{max}(\sigma_v^{\text{T-j}})$ (MPa)	Torque (N m)	Thickness (mm)
Baseline	4.26×10^3	3.74×10^2	140,834	8.84
Optimized	2.78×10^3	2.80×10^2	140,540	9.18
Relative difference	-34.74%	-25.13%	-0.21%	+3.85%

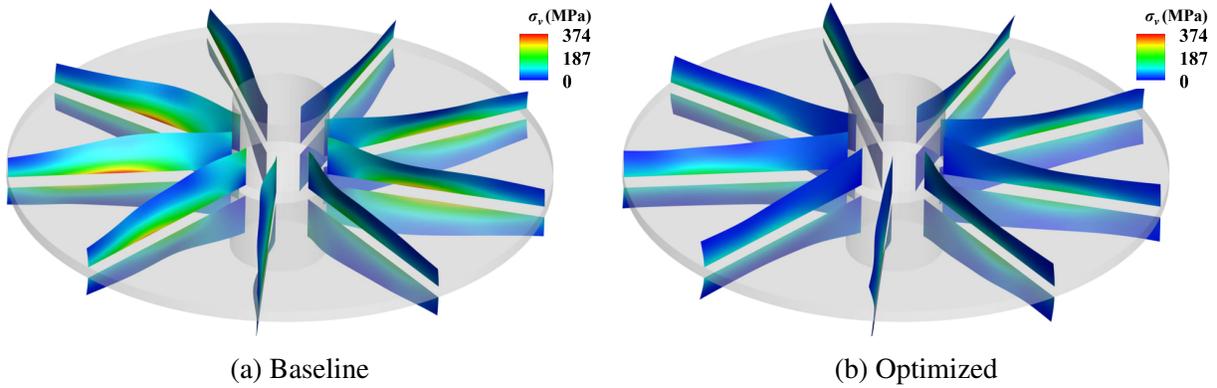


Figure 18: Contours of von Mises stress on baseline design and FSI-based optimization result. The deflection is scaled by 100 times. The time frames in which maximum von Mises stress occurs are selected for each case.

obtained from structural-mechanics-based optimization and the design obtained from FSI-based optimization highlights the value of using FSI-driven analyses in design.

In addition to comparing the stress contours of a single blade during FSI simulations, we can also compare von Mises stress distribution across all rotor blades as shown in Figure 18. These time frames are selected when $\text{max}(\sigma_v^{\text{T-j}})$ occurs. From the results, we learn that the alleviation of stress concentration is not only for a single blade presented in Figure 17, but also for all blades on the rotor. It is worth to emphasize that despite of the reduction of stress, the arresting torque and momentum of rotor are still unchanged, which implies that the functionality of retarding an aircraft is remaining the same as the baseline design.

Figure 19 shows the rotor blade deflection contours of baseline design and FSI-based optimization result superimposed with flow speed isosurfaces. The time frames are selected from which

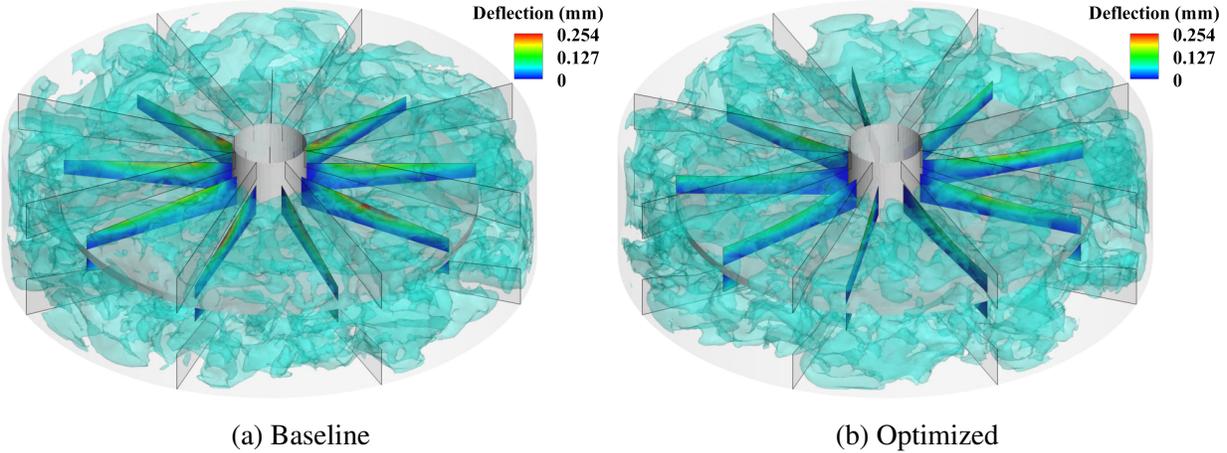


Figure 19: Rotor blade deflection contours of baseline design and FSI-based optimization result superimposed with flow speed isosurfaces. The time frames in which maximum deflection occurs are selected for each case. Note that shapes of stator vanes are different between baseline and optimized water twisters.

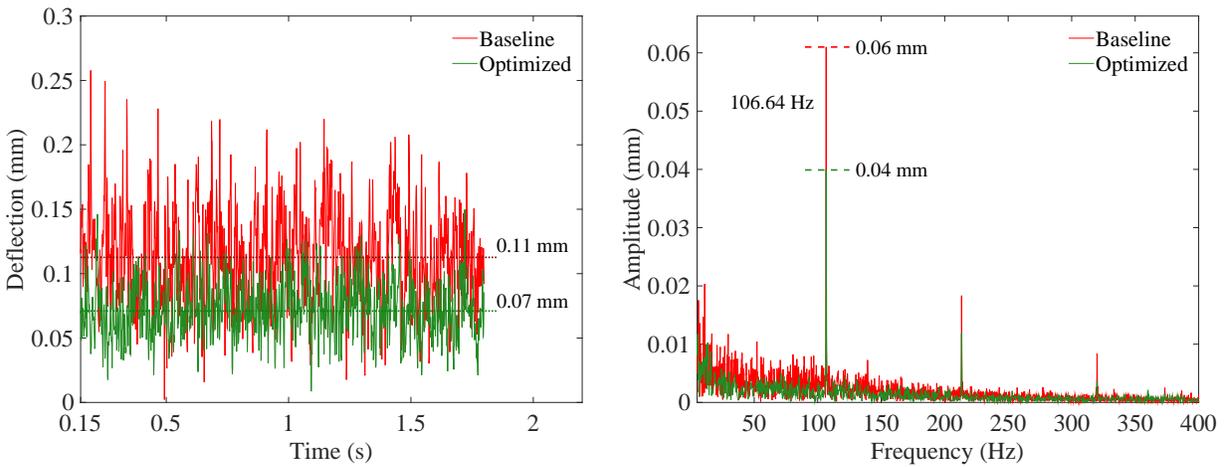


Figure 20: Comparison of blade deflection between baseline and optimized designs. Left: Blade deflection history. Dotted lines indicate time-average values of deflections. Right: Blade deflection frequencies. Dashed lines indicate amplitudes of frequency at 106.64 Hz.

any blade experiences the largest deflection throughout the entire time period. Note that these time frames coincide with those in which maximum von Mises stress occurs. The figure indicates that the overall blade deflections are reduced in the optimized case. Note that shapes of stator vanes are different between baseline and optimized water twisters. To further study the deflection, its frequency profile, computed via FFT, is shown in Figure 20. Despite the largest amplitude being smaller in the optimized case, the corresponding frequency remains 106.64 Hz, which is the same as that of the baseline blade vibration. This frequency is close to the frequency (106.67 Hz) at which the rotor blade passes the stator vane at 800 rpm. This implies that change of blade shape does not change the first mode frequency which is excited by the rotor–stator interaction.

6. Conclusions

This work proposes a computational framework that combines the advantages of immersogeometric FSI analysis and surrogate modeling to optimize the blade design for the water twister component of a hydraulic arresting gear. Immersogeometric FSI analysis invokes the isogeometric concept to use the structure design geometry directly as a computational mesh and avoids geometry-dependent mesh generation for the fluid subproblem by immersing the structure in an unfitted background mesh. This means that there is no design-dependent mesh generation required inside of the optimization loop. We compared immersogeometric simulations of a baseline water twister design with boundary-fitted computations based on thoroughly-verified numerical methods and found that the immersogeometric simulations were sufficiently accurate for computing quantities of interest relevant to our optimization procedure.

The proposed framework is employed to reduce stress concentrations in structural components of a water twister design. Immersogeometric FSI-based optimization using the SMF produces a design for which the variation of von Mises stress in critical structural components is reduced by 34.74% and the maximum von Mises stress is reduced by 25.13%. This improvement is made while maintaining an arresting torque within 0.21% of the original. This is achieved by reducing the area of the rotor blade in the region of stress concentration while increasing the area of the rotor blade in the outer portions, where torque production potential is the highest. Comparison to structural-mechanics-based optimization, which produces a slightly different design, reveals the value of using high-fidelity FSI analysis.

Future work may combine structural and FSI modeling in SMF to enhance the efficiency of optimization. This technology is directly applicable to a wide variety of FSI systems involving thin structural components. We believe that the results of this paper bode well for the prospects of optimizing thin blades and vanes of other turbomechanical devices using immersogeometric FSI analysis and SMF. We also anticipate applying this combination of technologies to cardiovascular problems, in which both immersogeometric FSI analysis [70, 71] and SMF [28] have already demonstrated value individually.

Acknowledgements

This work was supported by NAVAIR, Program Manager Mr. Nam Phan, and ARO Grant No. W911NF-14-1-0296, Program Managers Drs. Joseph Myers and Matthew Munson. This support is gratefully acknowledged. A.J. Herrema was partially supported by the NSF Grant No. DGE-1069283 which funds the activities of the Integrative Graduate Education and Research Traineeship (IGERT) in Wind Energy Science, Engineering and Policy (WESEP) at Iowa State University.

References

- [1] R. V. Parker. Arrestment considerations for the space shuttle. In *Proceedings of the Eighth Space Congress*, Cocoa Beach, Florida, 1971. Canaveral Council of Technical Societies.
- [2] United States Government Accountability Office. Ford-class aircraft carrier: Congress should consider revising cost cap legislation to include all construction costs. Report GAO-15-22, GAO, 2014.
- [3] United States Government Accountability Office. Defense acquisitions: assessment of selected weapon programs. Report GAO-15-342SP, GAO, 2015.
- [4] Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. Challenges and directions in computational fluid–structure interaction. *Mathematical Models and Methods in Applied Sciences*, 23:215–221, 2013.
- [5] K. Stein, R. Benney, V. Kalro, T. E. Tezduyar, J. Leonard, and M. Accorsi. Parachute fluid–structure interactions: 3-D Computation. *Computer Methods in Applied Mechanics and Engineering*, 190:373–386, 2000.
- [6] R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T. E. Tezduyar. Fluid–structure interaction modeling of aneurysmal conditions with high and normal blood pressures. *Computational Mechanics*, 38:482–490, 2006.
- [7] K. Takizawa, Y. Bazilevs, and T. E. Tezduyar. Space–time and ALE-VMS techniques for patient-specific cardiovascular fluid–structure interaction modeling. *Archives of Computational Methods in Engineering*, 19:171–225, 2012.
- [8] K. Takizawa and T. E. Tezduyar. Computational methods for parachute fluid–structure interactions. *Archives of Computational Methods in Engineering*, 19:125–169, 2012.
- [9] K. Takizawa, T. E. Tezduyar, and N. Kostov. Sequentially-coupled space–time FSI analysis of bio-inspired flapping-wing aerodynamics of an MAV. *Computational Mechanics*, 54:213–233, 2014.
- [10] Y. Bazilevs, K. Takizawa, T. E. Tezduyar, M.-C. Hsu, N. Kostov, and S. McIntyre. Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods. *Archives of Computational Methods in Engineering*, 21:359–398, 2014.
- [11] K. Takizawa, T. E. Tezduyar, C. Boswell, R. Kolesar, and K. Montel. FSI modeling of the reefed stages and disreefing of the Orion spacecraft parachutes. *Computational Mechanics*, 54:1203–1220, 2014.

- [12] Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. *Computational Fluid–Structure Interaction: Methods and Applications*. John Wiley & Sons, Chichester, 2013.
- [13] Y.-T. Chiu. *Computational Fluid Dynamics Simulations of Hydraulic Energy Absorber*. Master’s thesis, Virginia Polytechnic Institute and State University, 1999.
- [14] ANSYS Fluent. <http://www.ansys.com/Products/Fluids/ANSYS-Fluent>. Accessed 20 March 2016.
- [15] C. Wang, M. C. H. Wu, F. Xu, M.-C. Hsu, and Y. Bazilevs. Modeling of a hydraulic arresting gear using fluid-structure interaction and isogeometric analysis. *Computers & Fluids*, 2016. <http://dx.doi.org/10.1016/j.compfluid.2015.12.004>.
- [16] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J. B. Quincy. The variational multiscale method—A paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166:3–24, 1998.
- [17] T. J. R. Hughes, L. Mazzei, and K. E. Jansen. Large eddy simulation and the variational multiscale method. *Computing and Visualization in Science*, 3:47–59, 2000.
- [18] T. J. R. Hughes, A. A. Oberai, and L. Mazzei. Large eddy simulation of turbulent channel flows by the variational multiscale method. *Physics of Fluids*, 13:1784–1799, 2001.
- [19] T. J. R. Hughes and G. Sangalli. Variational multiscale analysis: the fine-scale Green’s function, projection, optimization, localization, and stabilized methods. *SIAM Journal of Numerical Analysis*, 45:539–557, 2007.
- [20] Y. Bazilevs, V. M. Calo, J. A. Cottrell, T. J. R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007.
- [21] Y. Bazilevs, M.-C. Hsu, K. Takizawa, and T. E. Tezduyar. ALE–VMS and ST–VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction. *Mathematical Models and Methods in Applied Sciences*, 22:1230002, 2012.
- [22] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
- [23] A. J. Herrema, N. M. Wiese, C. N. Darling, B. Ganapathysubramanian, A. Krishnamurthy, and M.-C. Hsu. A framework for parametric design optimization using isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 2016. Accepted.

- [24] D. B. Serafini. *A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions*. PhD thesis, Rice University, Houston, Texas, 1998.
- [25] A. J. Booker, J. E. Dennis Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17:1–13, 1999.
- [26] A. L. Marsden, M. Wang, J. E. Dennis Jr., and P. Moin. Optimal aeroacoustic shape design using the surrogate management framework. *Optimization and Engineering*, 5:235–262, 2004.
- [27] A. L. Marsden, J. A. Feinstein, and C. A. Taylor. A computational framework for derivative-free optimization of cardiovascular geometries. *Computer Methods in Applied Mechanics and Engineering*, 197:1890–1905, 2008.
- [28] A. L. Marsden. Optimization in cardiovascular modeling. *Annual Review of Fluid Mechanics*, 46:519–546, 2014.
- [29] C. C. Long, A. L. Marsden, and Y. Bazilevs. Shape optimization of pulsatile ventricular assist devices using FSI to minimize thrombotic risk. *Computational Mechanics*, 54:921–932, 2014.
- [30] M.-C. Hsu and Y. Bazilevs. Fluid–structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50:821–833, 2012.
- [31] M. W. Beall, J. Walsh, and M. S. Shephard. A comparison of techniques for geometry access related to mesh generation. *Engineering with Computers*, 20:210–221, 2004.
- [32] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.
- [33] Y. Bazilevs, M.-C. Hsu, and M. A. Scott. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249–252:28–41, 2012.
- [34] J. Kiendl. *Isogeometric Analysis and Shape Optimal Design of Shell Structures*. PhD thesis, Lehrstuhl für Statik, Technische Universität München, 2011.
- [35] T. E. Tezduyar. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering*, 8:83–130, 2001.

- [36] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, 2nd ed. Springer, Berlin, 2002.
- [37] J. A. Evans and T. J. R. Hughes. Explicit trace inequalities for isogeometric analysis and parametric hexahedral finite elements. *Computer Methods in Applied Mechanics and Engineering*, 123:259–290, 2013.
- [38] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.
- [39] D. Kamensky, J. A. Evans, and M.-C. Hsu. Stability and conservation properties of collocated constraints in immersogeometric fluid-thin structure interaction analysis. *Communications in Computational Physics*, 18:1147–1180, 2015.
- [40] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36:9–15, 1971.
- [41] H. J. C. Barbosa and T. J. R. Hughes. The finite element method with Lagrange multipliers on the boundary: circumventing the Babuška-Brezzi condition. *Computer Methods in Applied Mechanics and Engineering*, 85(1):109–128, 1991.
- [42] Y. Bazilevs, C. Michler, V. M. Calo, and T. J. R. Hughes. Weak Dirichlet boundary conditions for wall-bounded turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 196:4853–4862, 2007.
- [43] A. Düster, J. Parvizian, Z. Yang, and E. Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197(45–48):3768–3782, 2008.
- [44] A. Stavrev. *The Role of Higher-Order Geometry Approximation and Accurate Quadrature in NURBS Based Immersed Boundary Methods*. Master’s thesis, Technische Universität München, Munich, Germany, 2012.
- [45] M.-C. Hsu, C. Wang, F. Xu, A. J. Herrema, and A. Krishnamurthy. Direct immersogeometric fluid flow analysis using B-rep CAD models. *Computer Aided Geometric Design*, 43:143–158, 2016.

- [46] D. Schillinger, I. Harari, M.-C. Hsu, D. Kamensky, S.K.F Stoter, Y. Yu, and Y. Zhao. The non-symmetric Nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements. *Computer Methods in Applied Mechanics and Engineering*, 309:625–652, 2016.
- [47] F. Xu, D. Schillinger, D. Kamensky, V. Varduhn, C. Wang, and M.-C. Hsu. The tetrahedral finite cell method for fluids: Immersogeometric analysis of turbulent flow around complex geometries. *Computers & Fluids*, 2016. <http://dx.doi.org/10.1016/j.compfluid.2015.08.027>.
- [48] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- α method. *Journal of Applied Mechanics*, 60:371–75, 1993.
- [49] K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized- α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:305–319, 2000.
- [50] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid–structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43:3–37, 2008.
- [51] D. Kamensky, M.-C. Hsu, Y. Yu, J. A. Evans, M. S. Sacks, and T. J. R. Hughes. Immersogeometric cardiovascular fluid–structure interaction analysis using divergence-conforming B-splines. *Computer Methods in Applied Mechanics and Engineering*, 2016. <http://dx.doi.org/10.1016/j.cma.2016.07.028>.
- [52] R. Herzog and K. Kunisch. Algorithms for PDE-constrained optimization. *GAMM-Mitteilungen*, 33(2):163–176, 2010.
- [53] J. E. Hicken and J. J. Alonso. Comparison of reduced- and full-space algorithms for PDE-constrained optimization. In *Proceedings of AIAA 51st Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2013-1043, Grapevine, Texas, USA, 2013.
- [54] R. Fletcher. *Practical Methods of Optimization, 2nd ed.* John Wiley & Sons, Chichester, 1987.
- [55] A. L. Marsden, M. Wang, J. E. Dennis Jr., and P. Moin. Suppression of vortex-shedding noise via derivative-free shape optimization. *Physics of Fluids*, 16(10):L83–L86, 2004.
- [56] A. L. Marsden, M. Wang, J. E. Dennis Jr., and P. Moin. Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. *Journal of Fluid Mechanics*, 572: 13–36, 2007.

- [57] S. Sankaran, J. D. Humphrey, and A. L. Marsden. An efficient framework for optimization and parameter sensitivity analysis in arterial growth and remodeling computations. *Computer Methods in Applied Mechanics and Engineering*, 256:200–210, 2013.
- [58] W. Yang, J. A. Feinstein, S. C. Shadden, I. E. Vignon-Clementel, and A. L. Marsden. Optimization of a Y-graft design for improved hepatic flow distribution in the fontan circulation. *Journal of Biomechanical Engineering*, 135:011002, 2013.
- [59] T. J. Gundert, A. L. Marsden, W. Yang, and J. F. LaDisa Jr. Optimization of cardiovascular stent design using computational fluid dynamics. *Journal of Biomechanical Engineering*, 134:011002, 2012.
- [60] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42:55–61, 2000.
- [61] C. Audet and J. E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.
- [62] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. *Numerical Grid Generation: Foundations and Applications*. North-Holland, Amsterdam, 1985.
- [63] J. F. Thompson, B. K. Soni, and N. P. Weatherill. *Handbook of Grid Generation*. CRC Press, Boca Raton, FL, 1998.
- [64] P. J. Frey and P.-L. George. *Mesh Generation: Application to Finite Elements, 2nd ed.* ISTE Ltd, London, 2008.
- [65] S.-W. Cheng, T. K. Dey, and J. R. Shewchuk. *Delaunay Mesh Generation*. CRC Press, Boca Raton, FL, 2012.
- [66] Y. Zhang. *Geometric Modeling and Mesh Generation from Scanned Images*. CRC Press, Boca Raton, FL, 2016.
- [67] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal of Scientific Computing*, 20:359–392, 1998.
- [68] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Computers & Fluids*, 49:93–100, 2011.

- [69] I. Babuška and J. T. Oden. Verification and validation in computational engineering and science: basic concepts. *Computer Methods in Applied Mechanics and Engineering*, 193 (36–38):4057–4066, 2004.
- [70] M.-C. Hsu, D. Kamensky, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. *Computational Mechanics*, 54:1055–1071, 2014.
- [71] M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M. C. H. Wu, J. Mineroff, A. Reali, Y. Bazilevs, and M. S. Sacks. Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models. *Computational Mechanics*, 55:1211–1225, 2015.