ICES REPORT 14-12

May 2014

A variational immersed boundary framework for fluid–structure interaction: Isogeometric implementation and application to bioprosthetic heart valves

by

David Kamensky, Ming-Chen Hsu■, Dominik Schillinger, John A. Evans, Ankush Aggarwal, Yuri Bazilevs, Michael S. Sacks, Thomas J. R. Hughes

The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: David Kamensky, Ming-Chen Hsu■, Dominik Schillinger, John A. Evans, Ankush Aggarwal, Yuri Bazilevs, Michael S. Sacks, Thomas J. R. Hughes, "A variational immersed boundary framework for fluid–structure interaction: Isogeometric implementation and application to bioprosthetic heart valves," ICES REPORT 14-12, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, May 2014.

A variational immersed boundary framework for fluid–structure interaction: Isogeometric implementation and application to bioprosthetic heart valves

David Kamensky[a], Ming-Chen Hsu[b,*], Dominik Schillinger[c], John A. Evans[d], Ankush Aggarwal[a], Yuri Bazilevs[e], Michael S. Sacks[a], Thomas J. R. Hughes[a]

[a]*Center for Cardiovascular Simulation, Institute for Computational Engineering and Sciences, The University of Texas at Austin, 201 East 24th St, Stop C0200, Austin, TX 78712, USA*
[b]*Department of Mechanical Engineering, Iowa State University, 2025 Black Engineering, Ames, IA 50011, USA*
[c]*Department of Civil Engineering, University of Minnesota, 500 Pillsbury Drive S.E., Minneapolis, MN 55455, USA*
[d]*Department of Aerospace Engineering Sciences, University of Colorado at Boulder, 429 UCB, Boulder, CO 80309, USA*
[e]*Department of Structural Engineering, University of California, San Diego, 9500 Gilman Drive, Mail Code 0085, La Jolla, CA 92093, USA*

Abstract

In this paper, we develop a geometrically flexible technique for computational fluid–structure interaction (FSI). The motivating application is the simulation of tri-leaflet bioprosthetic heart valve function over the complete cardiac cycle. Due to the complex motion of the heart valve leaflets, the fluid domain undergoes large deformations, including changes of topology. The proposed method directly analyzes a NURBS surface representation of the structure by immersing it into a non-boundary-fitted discretization of the surrounding fluid domain.

The framework starts with an augmented Lagrangian formulation for FSI that enforces kinematic constraints with a combination of Lagrange multipliers and penalty forces. For immersed volumetric objects, we formally eliminate the multiplier field by substituting a fluid–structure interface traction, arriving at Nitsche's method for enforcing Dirichlet boundary conditions on object surfaces. For immersed thin shell structures modeled geometrically as surfaces, the tractions from opposite sides cancel due to the continuity of the background fluid solution space, leaving a penalty method. We find this penalty method sufficient to accurately compute quantities of interest for some problem types, but application to a bioprosthetic heart valve, where there is a large pressure jump across the leaflets, reveals shortcomings of the penalty approach. To counteract steep pressure gradients through the structure without the conditioning problems that accompany strong penalty forces, we resurrect the Lagrange multiplier field. Further, since the fluid discretization is not tailored to the structure geometry, there is a significant error in the approximation of pressure

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

discontinuities across the shell. This error becomes especially troublesome in residual-based stabilized methods for incompressible flow, leading to problematic compressibility at practical levels of refinement. We modify existing stabilized methods to improve performance.

To evaluate the accuracy of the proposed methods, we test them on benchmark problems and compare the results with those of established boundary-fitted techniques. Finally, we simulate the coupling of the bioprosthetic heart valve and the surrounding blood flow under physiological conditions, demonstrating the effectiveness of the proposed techniques in practical computations.

Keywords: Fluid–structure interaction, Bioprosthetic heart valve, Variational immersed boundary method, Isogeometric analysis, B-spline and NURBS, Nitsche's method, Weakly enforced boundary conditions, Penalty-based contact

---

Contents

1. Introduction

Heart valves are passive structures that open and close in response to hemodynamic forces, ensuring proper unidirectional blood flow through the heart. At least 280,000 diseased heart valves are surgically replaced annually [1, 2]. By far the most popular surgical replacements are the bioprosthetic heart valves (BHV), which are fabricated from biologically derived materials, with the design goal of mechanical similarity to native valves. Like native valves, BHVs are composed of thin flexible leaflets that are pushed open by blood flow in one direction and closed by flow in the other direction. BHVs have more natural hemodynamics than the older "mechanical" prostheses designs, which are comprised of rigid leaflets and require life-long anticoagulation therapy [2]. However, the durability of a typical BHV remains limited to about 10–15 years, with failure resulting from structural deterioration, mediated by fatigue and tissue mineralization [1–3]. While

much effort has gone into developing methods to mitigate mineralization, methods to extend durability remain largely unexplored. A critical part of such efforts to improve the design of BHVs is understanding the stresses acting on leaflets over the complete cardiac cycle.

Some previous computational studies on heart valve mechanics have used (quasi-)static [4] and dynamic [5] structural analysis, with assumed pressure loads on the leaflets. This produces deformation and stress distributions that can be used to understand the mechanical behavior of BHVs. However, the assumed pressure load only crudely approximates the interaction between blood and valvular structures; the results of a purely structural analysis are almost certainly inaccurate. It is therefore important to develop a computational framework that is able to simulate the dynamics of heart valves interacting with hemodynamics—a method for computational FSI—which considers the complete mechanical environment of the valve and applies more accurate tractions to the leaflets during the cardiac cycle.

Many FSI methods employ boundary-fitted approaches, where the fluid problem is solved on a mesh that deforms around a Lagrangian structure mesh, matching it at the shared interface. The fluid problem on the deforming domain is said to be posed in an arbitrary Lagrangian–Eulerain (ALE) coordinate system [6–8]. In the FSI literature, the term ALE is sometimes reserved for numerical methods using finite elements in space and finite differences in time, distinguishing them from methods that use space–time finite elements, such as the deforming-spatial-domain/stabilized space–time (DSD/SST) technique [9, 10]. Boundary-fitted FSI has been applied to challenging classes of real-world problems, including cardiovascular [11–15], parachute [16–19], and wind turbine [20–22] applications. The history, state-of-the-art, and practical applications of ALE and DSD/SST methods for FSI are covered thoroughly by Bazilevs et al. [23]. Boundary-fitted methods have the advantage of satisfying kinematic constraints by construction but, for scenarios that involve large translational and/or rotational structural motions, the boundary-fitted fluid mesh can become severely distorted, which harms both the conditioning of the discrete problem and the accuracy of its solution.

Applying boundary-fitted methods to complex engineered systems may therefore require specialized solution strategies to maintain fluid mesh quality. One approach is remeshing, in which all or part of the fluid domain is automatically re-discretized in space when mesh distortion becomes too extreme [24–27]. However, repeating this mesh regeneration process throughout the computation can be time-consuming and projection of solutions between meshes introduces additional errors. Mesh management is complicated further if the structure moves into and out of contact with itself, changing the topology of the fluid domain. For some applications, it may be sufficient to use specialized contact algorithms that modify the problem to enforce a small minimum separation between surfaces that would otherwise come into contact [28]. In our application to a heart valve, however, the ability of the structure to close and block flow is an essential aspect of the prob-

lem. Recent work [29] has extended boundary-fitted DSD/SST methods to include true changes of topology, but has so far only been applied to problems in two spatial dimensions, where the boundary motion is known beforehand and prescribed. While the rigid motions of hinged mechanical prosthetic heart valves have been successfully studied with boundary-fitted methods [30, 31], it is our opinion that maintaining mesh quality would become prohibitively difficult in a boundary-fitted simulation of a native or bioprosthetic heart valve, where flexible leaflets deform and contact each other in complex patterns that cannot be parameterized by a small set of variables.

For these reasons, non-boundary-fitted approaches have become a popular alternative for computational FSI [32–38]. The first non-boundary-fitted approach to become widely known for computational fluid dynamics (CFD) was Peskin's immersed boundary method [39, 40]. In non-boundary-fitted methods, a separate structural discretization is arbitrarily superimposed onto (or *immersed* into) a background fluid mesh. Such methods are particularly attractive for applications with complex moving boundaries, because they alleviate the difficulties of deforming the fluid mesh. Non-boundary-fitted methods can also handle change of fluid domain topology (e.g. structural contact) without special treatment in the fluid sub-problem. Contact algorithms [41–44] developed in structural dynamics can be adopted directly for the structure sub-problem. However, the non-boundary-fitted approach suffers from reduced accuracy of the solution near the fluid–structure interface. Dirichlet boundary conditions cannot be imposed strongly on the discrete solution space, because this space cannot interpolate functions given on an arbitrary immersed boundary. To apply interface conditions, one must devise a suitable method for weak enforcement.

The association between non-boundary-fitted methods and cardiovascular applications goes back to Peskin's original work [45] in 1972 and has been amplified by many publications in the intervening decades. Borazjani [46] compiled a current and thorough literature review and computed one of the most sophisticated and realistic heart valve analyses to date, using the curvilinear immersed boundary (CURVIB) method [47, 48]. Our work follows most directly from the fictitious domain method devised by Baaijens [49] and applied to heart valves by de Hart [50]. Baaijens and de Hart used Lagrange multipliers to enforce kinematic constraints between finite element discretizations of the fluid and thin immersed structures.

Prior simulations of heart valve FSI have suffered from a number of shortcomings. de Hart's implementation of the fictitious domain method does not contain any contact model and, while the author notes that the FSI kinematics alone should prevent the structure from self-intersecting, he found that, in practical discretizations, the weak constraint enforcement afforded by Lagrange multipliers still allowed significant penetrations. Further, de Hart's computations relied on symmetry assumptions that do not hold in the relevant flow regime [46]. Borazjani included contact in a computation of a full valve, but neither author satisfactorily computed the closed state of the valve, in which the leaflets must oppose a steep pressure gradient to enforce nearly hydrostatic flow.

In this work, we derive several related variational formulations from an augmented Lagrangian framework for FSI proposed by Bazilevs et al. [51]. The variational equations are the sum of fluid and structure sub-problems, with additional terms to enforce the kinematic constraint of velocity continuity at the fluid–structure interface. One additional term enforces the constraint through a Lagrange multiplier defined on the interface, while another term augments this constraint enforcement with a penalty to increase convexity of the formulation about the subset of the solution space satisfying the kinematic constraint.

For immersed volumetric objects, we follow the idea given in Bazilevs et al. [51] to formally eliminate the multiplier field, arriving at a method for weak enforcement of Dirichlet boundary conditions. This method of weak enforcement may be viewed as an extension of Nitsche's method [52]. We implement this with an adaptive quadrature rule, to accurately integrate over the fluid domain. As an added benefit, imposing the Dirichlet boundary conditions weakly in fluid dynamics allows the flow to slip on the solid surface when the wall-normal mesh size is relatively large. This effect mimics the thin boundary layer that would otherwise need to be resolved with spatial refinement, allowing more accurate solutions on coarse meshes [53–57]. In a non-boundary-fitted method, the fluid mesh is arbitrarily cut by the structural boundary, producing a boundary layer discretization of inferior quality compared to the body-fitted case. Therefore, the weakly enforced Dirichlet boundary conditions are *crucial* to obtaining more accurate fluid solutions when the non-boundary-fitted approach is used.

To model the valve leaflets we utilize immersed shell structures. We study various interpretations of the augmented Lagrangian framework applied to vanishingly-thin structures immersed in non-boundary-fitted fluid discretizations. We find that our extension of Nitsche's method reduces to a penalty method. This penalty method is sufficient to accurately compute quantities of interest for some problem types, but applications (such as the BHV) with large pressure jumps across the thin shell reveal shortcomings of the penalty approach. To counteract steep pressure gradients through the structure without the conditioning problems that accompany strong penalty forces, we introduce the additional unknowns to approximate the multiplier field. Further, since the fluid discretization is not tailored to the structure geometry, there is an inherent error in the approximation of pressure discontinuities across the shell. Our fluid formulation uses residual-based stabilization derived from a variational multiscale (VMS) analysis [58, 59]. This stabilization interacts with the large pressure error near the shell, leading to problematic compressibility at practical levels of refinement. To counteract this artificial compression, we weaken stabilization near the immersed shell structure.

Our discretizations of the fluid and structure sub-problems use isogeometric analysis (IGA) [60]; we use non-uniform rational B-spline (NURBS) basis functions to represent both geometry and solutions. The cited reference motivates IGA primarily as a means of simplifying mesh

generation by directly analyzing spline-based engineering designs. Our use of IGA in the present work is motivated instead by the desirable mathematical properties of the spline functions used in design. NURBS function spaces can have higher continuity than the approximation spaces found in traditional finite element analysis. For the fluid sub-problem, this continuity provides special benefits in turbulent flow simulation [61, 62] and, for the structure sub-problem, it eliminates the need for extra rotational degrees of freedom for thin shells [63] and better represents sliding contact between smooth surfaces [64].

The paper is organized as follows. In Section 2, we introduce the augmented Lagrangian framework for FSI and relate it to Nitsche's method. In Section 3, we employ an adaptive quadrature technique to implement Nitsche's method for immersed boundaries, testing it on the benchmark problem of 2D flow over a cylinder. Section 4 addresses the difficulties of enforcing constraints when the structure becomes infinitesimally thin. We discuss the computational methods implied by various interpretations of the augmented Lagrangian in this limit and present results for the benchmark problems of 2D flow over an elastic beam and an idealization of a closed heart valve. In Section 5, we combine our FSI technology with a penalty-based dynamic contact algorithm for shell structures, allowing us to compute a realistic FSI simulation of a bioprosthetic heart valve. Section 6 draws conclusions and provides a graphical representation (Figure 33) of the interrelations between ideas, methods, and computations presented throughout the paper. The reader may find this conceptual map helpful while navigating the body of the paper.

2. Augmented Lagrangian framework for FSI

Our starting point is the augmented Lagrangian framework for FSI introduced by Bazilevs et al. [51]. We consider $(\Omega_1)_t$ and $(\Omega_2)_t$ to be regions (subsets of $\mathbb{R}^d, d \in \{2, 3\}$) occupied by an incompressible fluid and an elastic solid, respectively, at time t, with $(\Gamma_1)_t$ and $(\Gamma_2)_t$ to be their corresponding boundaries. These regions meet at a shared interface, $(\Gamma_\mathrm{I})_t$. Let $\mathbf{u}_1$ and p denote the fluid velocity and pressure, respectively, and $\mathbf{u}_2$ denote the velocity of the structure. We impose the kinematic constraint that $\mathbf{u}_1 = \mathbf{u}_2$ on $(\Gamma_\mathrm{I})_t$ through the addition of the following augmented Lagrangian terms:

$$\int_{(\Gamma_\mathrm{I})_t} \boldsymbol{\lambda} \cdot (\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma + \frac{1}{2} \int_{(\Gamma_\mathrm{I})_t} \beta |\mathbf{u}_1 - \mathbf{u}_2|^2\, d\Gamma\,, \tag{1}$$

where $\boldsymbol{\lambda}$ is a Lagrange multiplier and $\beta \geq 0$ is a penalty parameter to increase convexity around the feasible region defined by the constraint. The variational problem is: Find $\mathbf{u}_1 \in \mathcal{S}_u$, $p \in \mathcal{S}_p$,

$\mathbf{u}_2 \in \mathcal{S}_d$, and $\boldsymbol{\lambda} \in \mathcal{S}_\ell$ 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 $\delta\boldsymbol{\lambda} \in \mathcal{V}_\ell$

$$B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}; \hat{\mathbf{u}}) - F_1(\{\mathbf{w}_1, q\}) + \int_{(\Gamma_1)_t} \mathbf{w}_1 \cdot \boldsymbol{\lambda}\, d\Gamma + \int_{(\Gamma_1)_t} \mathbf{w}_1 \cdot \beta(\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma \;=\; 0\,, \tag{2}$$

$$B_2(\mathbf{w}_2, \mathbf{u}_2) - F_2(\mathbf{w}_2) - \int_{(\Gamma_1)_t} \mathbf{w}_2 \cdot \boldsymbol{\lambda}\, d\Gamma - \int_{(\Gamma_1)_t} \mathbf{w}_2 \cdot \beta(\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma \;=\; 0\,, \tag{3}$$

$$\int_{(\Gamma_1)_t} \delta\boldsymbol{\lambda} \cdot (\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma \;=\; 0\,, \tag{4}$$

where $\mathcal{S}_u$, $\mathcal{S}_p$, $\mathcal{S}_d$, and $\mathcal{S}_\ell$ are the function spaces for the fluid velocity, fluid pressure, structural velocity, and Lagrange multiplier solutions, respectively, and $\mathcal{V}_u$, $\mathcal{V}_p$, $\mathcal{V}_d$, and $\mathcal{V}_\ell$ are the corresponding weighting function spaces. B_1, B_2, F_1, and F_2 are the semi-linear forms and linear functionals corresponding to the fluid and structural mechanics problems, respectively, and are given by

$$B_1(\{\mathbf{w}, q\}, \{\mathbf{u}, p\}; \hat{\mathbf{u}}) = \int_{(\Omega_1)_t} \mathbf{w} \cdot \rho_1 \left(\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\hat{\mathbf{x}}} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \boldsymbol{\nabla}\mathbf{u} \right) d\Omega + \int_{(\Omega_1)_t} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}_1\, d\Omega + \int_{(\Omega_1)_t} q \boldsymbol{\nabla} \cdot \mathbf{u}\, d\Omega\,, \tag{5}$$

$$F_1(\{\mathbf{w}, q\}) = \int_{(\Omega_1)_t} \mathbf{w} \cdot \rho_1 \mathbf{f}_1\, d\Omega + \int_{(\Gamma_{1\mathrm{h}})_t} \mathbf{w} \cdot \mathbf{h}_1\, d\Omega\,, \tag{6}$$

$$B_2(\mathbf{w}, \mathbf{u}) = \int_{(\Omega_2)_t} \mathbf{w} \cdot \rho_2 \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\mathbf{X}} d\Omega + \int_{(\Omega_2)_t} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}_2\, d\Omega\,, \tag{7}$$

$$F_2(\mathbf{w}) = \int_{(\Omega_2)_t} \mathbf{w} \cdot \rho_2 \mathbf{f}_2\, d\Omega + \int_{(\Gamma_{2\mathrm{h}})_t} \mathbf{w} \cdot \mathbf{h}_2\, d\Omega\,, \tag{8}$$

where ρ_1 and ρ_2 are the fluid and structural densities, respectively, $\hat{\mathbf{u}}$ is the velocity of the fluid domain $(\Omega_1)_t$, $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ are the fluid and structural Cauchy stresses, respectively, $\boldsymbol{\varepsilon}(\cdot)$ is the symmetric gradient operator given by $\boldsymbol{\varepsilon}(\mathbf{w}) = \frac{1}{2}(\boldsymbol{\nabla}\mathbf{w} + \boldsymbol{\nabla}\mathbf{w}^T)$, $\mathbf{f}_1$ and $\mathbf{f}_2$ are the applied body forces and $\mathbf{h}_1$ and $\mathbf{h}_2$ are the applied surface tractions on the fluid and structure, respectively, $(\Gamma_{1\mathrm{h}})_t$ and $(\Gamma_{2\mathrm{h}})_t$ are the boundaries where the surface tractions are specified, $\left.\frac{\partial(\cdot)}{\partial t}\right|_{\hat{\mathbf{x}}}$ is the time derivative taken with respect to the fixed spatial coordinate $\hat{\mathbf{x}}$ in the referential domain (which does not follow the motion of the fluid itself), and $\left.\frac{\partial(\cdot)}{\partial t}\right|_{\mathbf{X}}$ is the time derivative holding the material coordinates $\mathbf{X}$ fixed. The gradient $\boldsymbol{\nabla}$ is taken with respect to the spatial coordinate $\mathbf{x}$ of the current configuration. We assume that the fluid is Newtonian with dynamic viscosity μ, and Cauchy stress $\boldsymbol{\sigma}_1 = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}_1)$.

Bazilevs et al. [51] demonstrate how the multiplier, $\boldsymbol{\lambda}$, may be formally eliminated by substituting an expression for the fluid–structure interface traction in terms of the other unknowns. This leads to the following variational formulation for the coupled problem: find $\mathbf{u}_1 \in \mathcal{V}_u$, $p \in \mathcal{V}_p$, and

$\mathbf{u}_2 \in \mathcal{V}_d$ such that for all $\mathbf{w}_1 \in \mathcal{V}_u$, $q \in \mathcal{V}_p$, and $\mathbf{w}_2 \in \mathcal{V}_d$

$$
\begin{aligned}
&B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}; \hat{\mathbf{u}}) - F_1(\{\mathbf{w}_1, q\}) + B_2(\mathbf{w}_2, \mathbf{u}_2) - F_2(\mathbf{w}_2) \\
&- \int_{(\Gamma_I)_t} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \boldsymbol{\sigma}_1(\mathbf{u}, p)\, \mathbf{n}_1 \, d\Gamma \\
&- \int_{(\Gamma_I)_t} \delta\boldsymbol{\sigma}_1(\mathbf{w}_1, q)\, \mathbf{n}_1 \cdot (\mathbf{u}_1 - \mathbf{u}_2) \, d\Gamma \\
&+ \int_{(\Gamma_I)_t} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \beta(\mathbf{u}_1 - \mathbf{u}_2) \, d\Gamma \;=\; 0 \,.
\end{aligned} \tag{9}
$$

Further manipulations arrive at a formulation for weak imposition of Dirichlet boundary conditions on the fluid problem,

$$
\begin{aligned}
B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}; \hat{\mathbf{u}}) - F_1(\{\mathbf{w}_1, q\}) &- \int_{(\Gamma_I)_t} \mathbf{w}_1 \cdot \boldsymbol{\sigma}_1 \mathbf{n}_1 \, d\Gamma \\
&- \int_{(\Gamma_I)_t} \delta\boldsymbol{\sigma}_1 \mathbf{n}_1 \cdot (\mathbf{u}_1 - \mathbf{u}_2) \, d\Gamma \\
&+ \int_{(\Gamma_I)_t} \mathbf{w}_1 \cdot \beta(\mathbf{u}_1 - \mathbf{u}_2) \, d\Gamma \;=\; 0 \,,
\end{aligned} \tag{10}
$$

and a traction boundary condition for the structure problem that is a combination of the fluid Cauchy stress and a penalty force:

$$
B_2(\mathbf{w}_2, \mathbf{u}_2) - F_2(\mathbf{w}_2) + \int_{(\Gamma_I)_t} \mathbf{w}_2 \cdot (\boldsymbol{\sigma}_1 \mathbf{n}_1 + \beta(\mathbf{u}_2 - \mathbf{u}_1)) \, d\Gamma \;=\; 0 \,. \tag{11}
$$

This approach to weak imposition of Dirichlet boundary conditions in fluid mechanics was first proposed by Bazilevs and Hughes [53] and further refined in Bazilevs et al. [54, 55]. It may be interpreted as an extension of Nitsche's method [65], which is a consistent and stabilized method for imposing constraints on the boundaries by augmenting the governing equations with additional constraint equations. While Nitsche's method may be motivated independently of the augmented Lagrangian formulation, we find that some cases require us to revisit Eq. (2) and account for the multipliers directly. The solution techniques for the fluid sub-problem (10) are discussed in Section 3, which follows.

3. Nitsche's method for immersed boundaries

In a non-boundary-fitted method, the elements of the fluid discretization may extend into the interior of an immersed object. Imposing Dirichlet boundary conditions is no longer straightforward given that the basis functions are non-interpolating at the object boundaries. In order to

enforce essential boundary conditions, one can either modify the basis functions so they vanish at the interface [66] or augment the governing equations with additional constraint equations. In this work we choose the latter approach. We formally eliminate the Lagrange multiplier from Eq. (2), as mentioned in Section 2 and detailed in Bazilevs et al. [51], to yield the fluid sub-problem (10), which corresponds to an application of Nitsche's method to the boundary condition on the fluid–structure interface. This weak imposition of the Dirichlet boundary conditions is the starting point of our variational immersed boundary approach.

3.1. Semi-discrete fluid formulation with weak boundary conditions

Consider a collection of disjoint elements $\{\Omega^e\}$, $\cup_e \Omega^e \subset \mathbb{R}^d$, with closures covering the fluid domain: $\Omega_1 \subset \cup_e \overline{\Omega^e}$. Note that Ω^e is not necessarily a subset of Ω_1. $\{\Omega^e\}$, Ω_1, and Γ_{I} remain time-dependent, but we drop the subscript t for notational convenience. The mesh defined by $\{\Omega^e\}$ deforms with a velocity field $\hat{\mathbf{u}}^h$ and the boundary Γ_{I} moves with velocity $\mathbf{u}_2$. We consider discrete velocity and pressure spaces $\mathcal{V}_u^h$ and $\mathcal{V}_p^h$ of functions supported on these elements and pose the semi-discrete problem of finding $\mathbf{u}_1^h \in \mathcal{V}_u^h$ and $p^h \in \mathcal{V}_p^h$ such that for all $\mathbf{w}_1^h \in \mathcal{V}_u^h$ and $q^h \in \mathcal{V}_p^h$

$$
\begin{aligned}
& B_1^{\mathrm{VMS}}\left(\{\mathbf{w}_1^h, q^h\}, \{\mathbf{u}_1^h, p^h\}; \hat{\mathbf{u}}^h\right) - F_1^{\mathrm{VMS}}\left(\{\mathbf{w}_1^h, q^h\}\right) \\
& - \int_{\Gamma_{\mathrm{I}}} \mathbf{w}_1^h \cdot \left(-p^h \mathbf{n}_1 + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}_1^h)\mathbf{n}_1\right) \, d\Gamma \\
& - \int_{\Gamma_{\mathrm{I}}} \left(2\mu\boldsymbol{\varepsilon}(\mathbf{w}_1^h)\mathbf{n}_1 + q^h \mathbf{n}_1\right) \cdot \left(\mathbf{u}_1^h - \mathbf{u}_2\right) \, d\Gamma \\
& - \int_{(\Gamma_{\mathrm{I}})^-} \mathbf{w}_1^h \cdot \rho_1 \left(\left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h\right) \cdot \mathbf{n}_1\right)\left(\mathbf{u}_1^h - \mathbf{u}_2\right) \, d\Gamma \\
& + \int_{\Gamma_{\mathrm{I}}} \tau_{\mathrm{TAN}}^B \left(\mathbf{w}_1^h - \left(\mathbf{w}_1^h \cdot \mathbf{n}_1\right)\mathbf{n}_1\right) \cdot \left(\left(\mathbf{u}_1^h - \mathbf{u}_2\right) - \left(\left(\mathbf{u}_1^h - \mathbf{u}_2\right) \cdot \mathbf{n}_1\right)\mathbf{n}_1\right) \, d\Gamma \\
& + \int_{\Gamma_{\mathrm{I}}} \tau_{\mathrm{NOR}}^B \left(\mathbf{w}_1^h \cdot \mathbf{n}_1\right)\left(\left(\mathbf{u}_1^h - \mathbf{u}_2\right) \cdot \mathbf{n}_1\right) \, d\Gamma \; = \; 0 \,, \qquad (12)
\end{aligned}
$$

where $(\Gamma_{\mathrm{I}})^-$ is the "inflow" part of Γ_{I}, on which $(\mathbf{u}_1^h - \hat{\mathbf{u}}^h) \cdot \mathbf{n}_1 < 0$. Note that Γ_{I} may cut through element interiors. The constants τ_{TAN}^B and τ_{NOR}^B correspond to a splitting of the penalty, β, into the tangential and normal directions, respectively. The forms B_1^{VMS} and F_1^{VMS} are the VMS discretiza-

tions of B_1 and F_1, respectively, given by

$$
\begin{aligned}
B_1^{\mathrm{VMS}}(\{\mathbf{w},q\},\{\mathbf{u},p\};\hat{\mathbf{u}}) = & \int_{(\Omega_1)_t} \mathbf{w}\cdot\rho_1\left(\left.\frac{\partial\mathbf{u}}{\partial t}\right|_{\hat{\mathbf{x}}} + (\mathbf{u}-\hat{\mathbf{u}})\cdot\boldsymbol{\nabla}\mathbf{u}\right)\,d\Omega \\
& + \int_{(\Omega_1)_t} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}_1\,d\Omega + \int_{(\Omega_1)_t} q\boldsymbol{\nabla}\cdot\mathbf{u}\,d\Omega \\
& + \sum_e \int_{\Omega^e\cap\Omega_1} \left((\mathbf{u}-\hat{\mathbf{u}})\cdot\boldsymbol{\nabla}\mathbf{w} + \frac{\boldsymbol{\nabla}q}{\rho_1}\right)\cdot\mathbf{u}'\,d\Omega \\
& + \sum_e \int_{\Omega^e\cap\Omega_1} \boldsymbol{\nabla}\cdot\mathbf{w}\rho_1\tau_{\mathrm{C}}\boldsymbol{\nabla}\cdot\mathbf{u}\,d\Omega \\
& - \sum_e \int_{\Omega^e\cap\Omega_1} \mathbf{w}\cdot(\mathbf{u}'\cdot\boldsymbol{\nabla}\mathbf{u})\,d\Omega \\
& - \sum_e \int_{\Omega^e\cap\Omega_1} \frac{\boldsymbol{\nabla}\mathbf{w}}{\rho_1} : (\mathbf{u}'\otimes\mathbf{u}')\,d\Omega \\
& + \sum_e \int_{\Omega^e\cap\Omega_1} (\mathbf{u}'\cdot\boldsymbol{\nabla}\mathbf{w})\,\overline{\tau}\cdot(\mathbf{u}'\cdot\boldsymbol{\nabla}\mathbf{u})\,d\Omega\,,
\end{aligned}
\tag{13}
$$

and

$$
F_1^{\mathrm{VMS}}(\{\mathbf{w},q\}) = F_1(\{\mathbf{w},q\})\,, \tag{14}
$$

where

$$
\mathbf{u}' = \tau_{\mathrm{M}}\left(\rho_1\left(\left.\frac{\partial\mathbf{u}}{\partial t}\right|_{\hat{\mathbf{x}}} + (\mathbf{u}-\hat{\mathbf{u}})\cdot\boldsymbol{\nabla}\mathbf{u} - \mathbf{f}\right) - \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}_1\right)\,. \tag{15}
$$

Equations (13)–(15) correspond to the ALE–VMS formulation of the Navier–Stokes equations of incompressible flows [67]. The additional terms may be interpreted both as stabilization and as a turbulence model [62, 68–73]. The specific form of VMS stabilization that we use was presented and applied to FSI problems by Bazilevs et al. [74]. The stabilization parameters are

$$
\tau_{\mathrm{M}} = \left(\frac{C_t}{\Delta t^2} + (\mathbf{u}-\hat{\mathbf{u}})\cdot\mathbf{G}(\mathbf{u}-\hat{\mathbf{u}}) + C_I\left(\frac{\mu}{\rho_1}\right)^2\mathbf{G}:\mathbf{G}\right)^{-1/2}\,, \tag{16}
$$

$$
\tau_{\mathrm{C}} = (\tau_{\mathrm{M}}\mathrm{tr}\mathbf{G})^{-1}\,, \tag{17}
$$

$$
\overline{\tau} = (\mathbf{u}'\cdot\mathbf{G}\mathbf{u}')^{-1/2}\,, \tag{18}
$$

where C_I is a positive constant derived from an appropriate element-wise inverse estimate [75–78], $\mathbf{G}$ generalizes the notion of element size to physical elements mapped from a parametric parent

element by $\mathbf{x}(\xi)$:

$$G_{ij} = \sum_{k=1}^{d} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j} \tag{19}$$

and the parameter C_t is typically equal to 4 [62, 70]. Note that we have modified the usual formulation, so that integrals are taken only over intersections of elements with Ω_1. Accurate evaluation of such integrals for general immersed geometries is the primary practical challenge associated with this formulation. Our approach to computing these integrals is discussed in detail in Section 3.2.

Remark 1. The fluid mesh motion given by $\hat{\mathbf{u}}^h$ may at first appear superfluous in the context of non-boundary-fitted methods. However, a single computation might gainfully combine a boundary-fitted, deforming-mesh treatment of some structures with a non-boundary-fitted treatment of others. Several such computations are performed by Wick [79]. An example relevant to our application would be immersion of non-boundary-fitted heart valve leaflets into a boundary-fitted discretization of the interior of a flexible artery. In computations with a fixed background fluid mesh, one can simply set $\hat{\mathbf{u}}^h = 0$ in the above formulations. A similar computation is performed with prescribed arterial movement by de Hart [50].

The terms from the second to the last line of Eq. (12) are responsible for the weak enforcement of kinematic and traction constraints at the immersed boundaries. It was shown in earlier work [53–57] that imposing the Dirichlet boundary conditions weakly in fluid dynamics allows the flow to slip on the solid surface when the wall-normal mesh size is relatively large. This effect mimics the thin boundary layer that would otherwise need to be resolved with spatial refinement, allowing more accurate solutions on coarse meshes. In the immersed boundary method, the fluid mesh is arbitrarily cut by the structural boundary, leaving a boundary layer discretization of inferior quality compared to the body-fitted case. Therefore, in addition to imposing the constraints easily in the context of non-boundary-fitted approach, we may obtain more accurate fluid solutions as an added benefit of using the weak boundary condition formulation (12).

Remark 2. Equation (12) includes an "inflow" stabilization term that is not associated with Nitsche's approach. This term is added to better satisfy the inflow boundary condition and to enhance the stability of the formulation, without affecting consistency or adjoint consistency. See Bazilevs et al. [53] for details. To ensure balanced interface tractions between the fluid and structure, we append the corresponding reaction force term

$$\int_{(\Gamma_1)^-} \mathbf{w}_2 \cdot \rho_1 \left(\left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h \right) \cdot \mathbf{n}_1 \right) \left(\mathbf{u}_1^h - \mathbf{u}_2 \right) \, d\Gamma \tag{20}$$

to the left-hand side of structure sub-problem, Eq. (11).

In Eq. (12), the parameters τ^B_{TAN} and τ^B_{NOR} must be sufficiently large to stabilize the formulation, but not so large as to degenerate Nitsche's method into a pure penalty method, which entails the disadvantages of losing variational consistency and having an ill-conditioned stiffness matrix. Based on previous studies of weakly-enforced Dirichlet boundary conditions in fluid mechanics [53–55], we expect these parameters to scale as

$$\tau^B_{(\cdot)} = \frac{C^B_I \mu}{h} \tag{21}$$

where h is a measure of the element size at the boundary and C^B_I is a dimensionless constant. However, in the case of an immersed boundary, neither the appropriate definition of h nor the principle for deriving C^B_I is straightforward. In subsequent sections, we investigate different penalty values through numerical experiments.

Remark 3. A more sophisticated approach to determine the values of the stabilization parameters for Nitsche's method is to solve local eigenvalue problems. See Hughes and Harari [80], Embar et al. [81], and Ruess et al. [82, 83] for more details.

3.1.1. Backflow stabilization

Unsteady CFD computations may sometimes diverge due to flow reversal on outflow boundaries. This is known as backflow divergence and is frequently encountered in cardiovascular simulations. In some problems studied in this paper, we encounter this backflow divergence and an outflow stabilization method originally proposed in Bazilevs et al. [84] is applied to compensate for it. The backflow stabilization method was further studied and found to be the least intrusive and computationally expensive of all the techniques examined in Esmaily-Moghadam et al. [85]. The method adds the following term to the left-hand side of Eq. (12):

$$\sum_{a=1}^{n_{\text{out}}} \left(-\gamma \int_{\Gamma_1^a} \mathbf{w}_1^h \cdot \rho_1 \left\{ \left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h \right) \cdot \mathbf{n}_1 \right\}_- \mathbf{u}_1^h \, d\Gamma \right) \tag{22}$$

where $\{\Gamma_1^a\}_{a=1}^{n_{\text{out}}}$ are the outflow portions of the fluid domain boundary, γ is a dimensionless non-negative scalar controlling the strength of the stabilization, and

$$\left\{ \left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h \right) \cdot \mathbf{n}_1 \right\}_- = \frac{1}{2} \left(\left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h \right) \cdot \mathbf{n}_1 - \left| \left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h \right) \cdot \mathbf{n}_1 \right| \right) \tag{23}$$

is the component of velocity pointing opposite the outward-facing normal of the fluid domain.

3.2. The finite cell method and adaptive quadrature

A similar idea of using Nitsche's method for immersed boundary FSI has been studied by Benk et al. [86], who assume that the immersed boundary is a triangulated surface and use methods from

computational geometry to decompose the exterior parts of cut fluid elements into polyhedrons with known quadrature rules. We apply instead an adaptive quadrature rule from the finite cell method [87–89] that relies only on a test to determine whether or not an arbitrary point lies inside of an immersed object. This relaxes Benk et al.'s assumption that the immersed boundaries are triangulated.

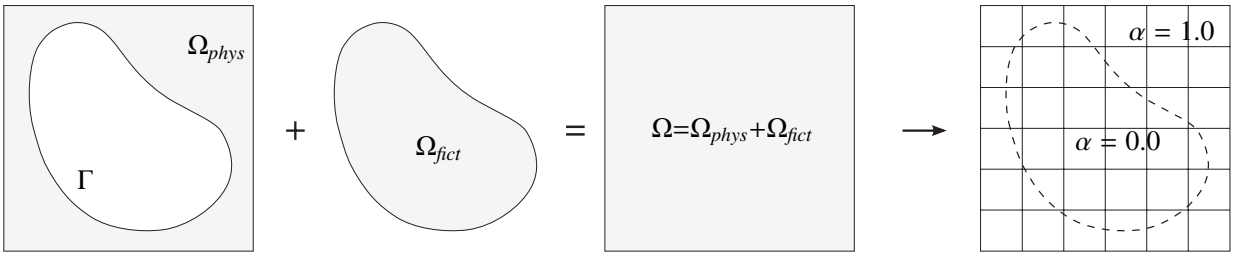


Figure 1: The physical domain of interest Ω_{phys} is extended by the fictitious domain Ω_{fict} into an embedding domain Ω to allow easy meshing of complex geometries. The influence of Ω_{fict} is penalized by α.

The finite cell method, introduced by Parvizian et al. [90] and illustrated in Figure 1, is a technique for solving partial differential equations posed on complex geometries by extending the computational domain to a more tractable shape, such as a rectangular prism bounding the original domain. The finite cell method discretizes this extended domain into elements and penalizes the effects of the fictitious extension by modifying the problem's coefficients to have extreme values outside the domain of interest. This introduces discontinuities in coefficients along the boundary of the original domain. Because the extended domain is discretized without respect to the original geometry, these discontinuities may occur within elements. The standard Gaussian quadrature rules typically applied to finite elements [91] assume that a polynomial can accurately approximate the integrand, but this assumption is not true if the integrand is discontinuous. Düster et al. [87] describe a method of automatically generating more accurate quadrature rules for finite cell computations by dividing cut elements into sub-cells and applying standard quadrature rules within the sub-cells. We apply the same method to the integrals over fluid portions of cut elements in Eq. (13). For completeness, we restate this adaptive quadrature technique, specializing it to the context of immersed boundary FSI. For a summary of recent developments in the finite cell method, we refer the interested reader to Schillinger and Ruess [89].

The quadrature scheme assumes that elements have d-dimensional rectangular parameterizations. The parameter space for each element may be partitioned into 2^d equal sub-cells. Each sub-cell may be likewise divided, as may its children, and so on, yielding a hierarchical 2^d-tree. A sub-cell at any level of this tree has an associated Gaussian quadrature rule. We may construct a quadrature rule for the entire element by summing quadrature rules from disjoint sub-cells covering the element. Not all sub-cells used for this rule need to be from the same level of the tree. Ideally, we would use sub-cells from more refined levels of the hierarchy near the immersed boundary

while using larger cells away from the boundary, to reduce the computational cost due to integration. Such an adaptive quadrature rule may be generated by applying the following recursive algorithm, with input $0 \leq l \in \mathbb{Z}$, to a sub-cell covering the entire element:

1. Propose a set of Gaussian quadrature points and weights associated with the current sub-cell.
2. Count the numbers N_{in} and N_{out} of the corners of the sub-cell falling inside and outside of the immersed structure.
3. If $N_{\text{in}} = 0$, $N_{\text{out}} = 0$, or $l = 0$, add the proposed quadrature points falling in the fluid domain to the quadrature rule.
4. Otherwise, if $N_{\text{in}} > 0$, $N_{\text{out}} > 0$, and $l > 0$, discard the proposed points, divide the sub-cell into 2^d children, and apply this algorithm to each child, with input $l - 1$.

Figure 2 illustrates the terminal sub-cells and the adaptive quadrature points that result from applying this algorithm to a 2D circular boundary, with $l = 3$ levels of recursion. The adaptive quadrature points outside the cylinder belong to the fluid domain and are used in the numerical integration. The quadrature points inside the cylinder belong to the fictitious domain extension and are discarded.

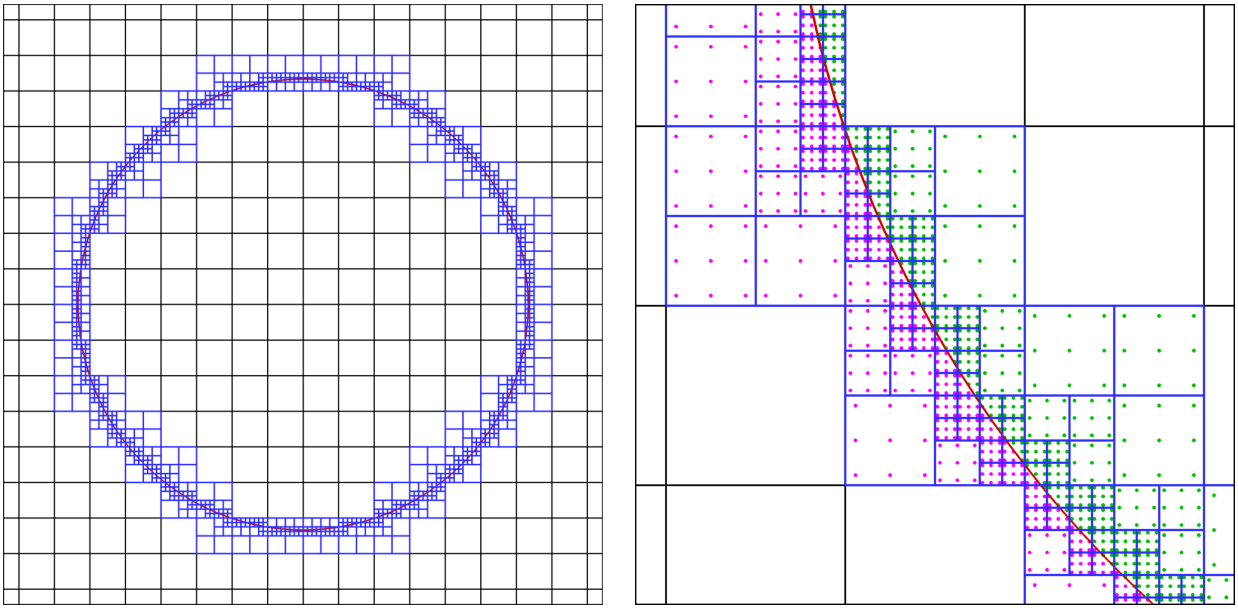


Figure 2: The sub-cells (blue lines) used to generate an adaptive quadrature rule for a circular boundary, with $l = 3$ levels of recursion. The adaptive quadrature points outside the cylinder (marked in pink) belong to the physical domain of interest and are used in the numerical integration. The quadrature points inside the cylinder (marked in green) belong to the fictitious domain extension and are discarded.

Remark 4. In the above algorithm, the geometry of the immersed structure is abstracted behind an inside/outside test that maps spatial positions to truth values. The efficient implementation of this mapping for general geometries is outside the scope of this paper, as we only consider

benchmark problems for which it is trivial. A more general implementation could cast rays from a point and count intersections with the closed immersed surface geometry. The operation of ray-surface intersection has been thoroughly optimized within the computer graphics community and was applied to real-time rendering of NURBS surfaces as early as the 1980s [92].

3.2.1. Surface integrals

The surface integrals of Eq. (12) also require special treatment. We employ a variant of the approach used by Düster et al. [87] to integrate immersed boundary traction in finite cell solutions of solid mechanics problems. We define a Gaussian quadrature rule with respect to a parameterization of the immersed boundary. This parameterization need not be informed by the fluid discretization, but we recommend ensuring that the physical space density of surface quadrature points is reasonably high with respect to the fluid element size. The relevant integrals involve traces of functions defined on the fluid domain. To evaluate these traces, we must be able to locate the quadrature points of the surface in the parameter space of the background mesh. The physical location, $\mathbf{x}_g \in \mathbb{R}^d$, of an integration point can be obtained by evaluating the surface parameterization. Finding the point $\xi_g \in \mathbb{R}^d$ that the fluid mesh parameterization maps to $\mathbf{x}_g$ requires solving a system of d equations to invert the mapping from the fluid mesh parameter space to physical space. If the fluid is represented on a rectangular grid, this inversion is trivial. For more general fluid discretizations, one may apply Newton iteration within parametric elements. It is usually not necessary to attempt this iteration in every fluid element for each quadrature point. The searching process can be streamlined by using element bounding boxes and assuming that each surface quadrature point will most likely remain in the same background element or move to a neighboring element between time steps in an unsteady calculation with moving boundaries.

3.3. Time integration of the fluid sub-problem

We complete the discretization of the fluid sub-problem by applying a time integration scheme to Eq. (12). Our scheme falls within the family of generalized-α integrators, introduced by Chung and Hulbert [93]. The generalized-α framework was first used for the unsteady Navier-Stokes problem by Jansen et al. [94]. The particular integration scheme that we use in the current work is detailed and applied to FSI problems in Bazilevs et al. [74]. The subset of generalized-α methods used in Bazilevs et al. [74] is parameterized by a single number, ρ_∞, where $0 \leq \rho_\infty \leq 1$. In agreement with Bazilevs et al. [62], we find no significant differences between admissible choices of ρ_∞, and use $\rho_\infty = 0.5$ as a default value. The generalized-α time integration is an implicit scheme and requires solution of a nonlinear algebraic problem at each time step. For situations in which only the fluid sub-problem is nontrivial (such as the CFD benchmark problem studied in Section 3.4), we directly apply Newton iteration (with an approximate tangent) to converge the residual of this algebraic problem. For coupled FSI, we apply the same time integration scheme, but use

more complicated solution strategies for the resulting nonlinear problem. We defer presenting the details of these solution strategies until Section 4.5.

3.4. Flow around an immersed cylinder

In this section, we apply our Nitsche-based variational implementation of the immersed boundary method with adaptive quadrature to the classic benchmark problem of 2D flow past a circular cylinder. The problem setup and computational domain are shown in Figure 3. The data given in the diagram is non-dimensional. We use a unit density and define the viscosity in terms of the Reynolds number, $\mu = Re^{-1}$. We strongly enforce the inflow and slip boundary conditions stated in Figure 3. For the no-slip, no-penetration condition $\mathbf{u}_1 = \mathbf{0}$ on the surface of the circular cylinder, we compare the results of weak enforcement, using our variational immersed boundary method, with results of strong enforcement, using a body-fitted mesh.

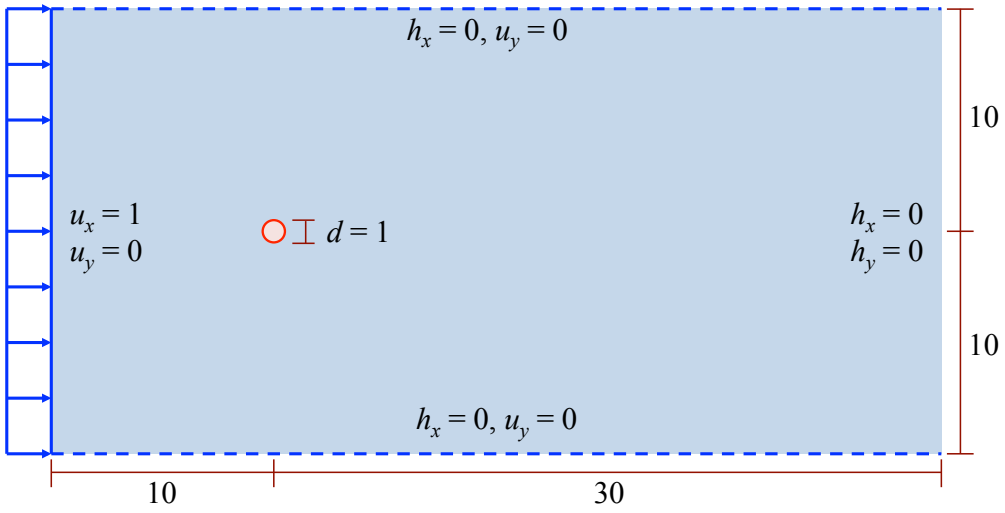


Figure 3: The domain and boundary conditions for the benchmark problem of 2D flow past a circular cylinder.

We expect that, for low Reynolds numbers, this problem will reach a stable steady state and, for moderate Reynolds numbers, it will develop a time-periodic solution. These expectations are characterized more precisely alongside our computed results in Section 6.

3.4.1. Immersed discretizations

We test the Nitsche-based variational immersed boundary method on two discretizations of the fluid domain. Both meshes use quadratic B-spline elements. The first mesh, abbreviated herein as "M1", contains 12240 elements, with refinement focused around the cylinder as shown in Figure 4. The element size near the cylinder is 0.079. The second mesh, M2, is a uniform h-refinement of M1. The inside/outside test required to adaptively generate quadrature rules for the exterior portions of cut cells is, in this case, a trivial distance check from the cylinder's center. The parametric

surface used to obtain a quadrature rule for surface integrals over Γ_I, the surface of the cylinder, is a quadratic NURBS circle divided into 256 knot spans in the circumferential direction, with 3-point Gaussian quadrature rules defined on each span. The circumference of this circle is π, giving elements of arc length $\pi/256 \approx 0.012$, which is significantly smaller than the element size in either M1 or M2. For the non-boundary-fitted computations, we consistently use a time step of $\Delta t = 0.1$ when steady solutions are anticipated and $\Delta t = 0.05$ when we expect periodicity. This ensures that there will be at least 100 time steps per period in all periodic solutions. The computations are initialized by linearly increasing the inflow velocity from zero to one over some time interval. Details of the initialization procedure should not affect the steady or time-periodic solutions that the system approaches.

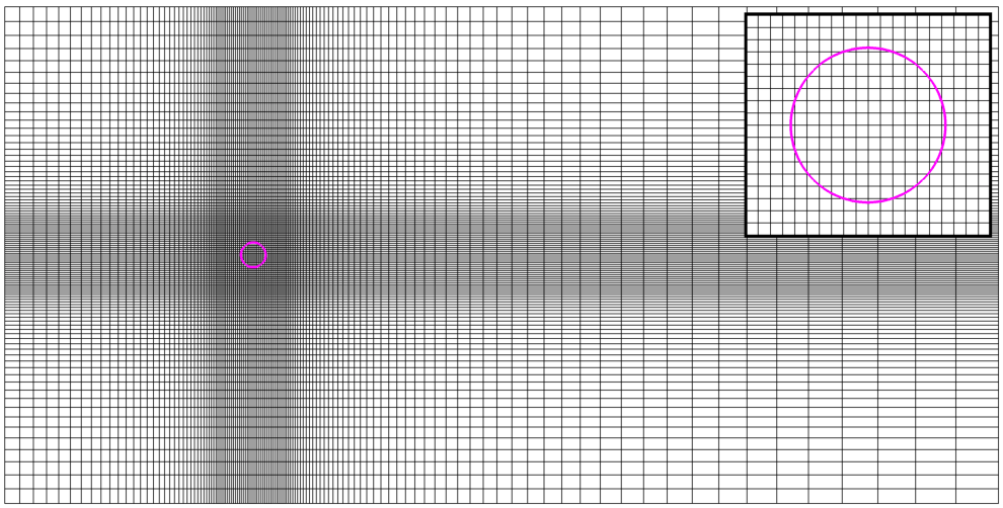


Figure 4: The immersed boundary mesh M1.

Remark 5. We partition the fluid domain into sub-domains, for efficient parallel computation on distributed-memory supercomputers. M1 is decomposed into 12 sub-domains, and M2 into 48. A detailed technical explanation and scalability study of our parallelization strategy may be found in Hsu et al. [95]. In the present computations, we have reduced continuity of the approximation space to C^0 at boundaries between sub-domains. While this is not technically necessary, it minimizes communication bandwidth while maintaining the benefits of higher continuity throughout most of the domain. We find that the impact on quantities of interest is negligible, especially at low Reynolds numbers.

3.4.2. Body-fitted reference mesh

The problem at hand has been studied extensively by the CFD community (see, e.g. [96–103]), but, to control for any discrepancies introduced by differences in fluid formulations or turbulence models, we apply the same VMS formulation (13) to a body-fitted discretization of the problem,

with a strongly-enforced boundary condition on the surface of the cylinder. The body-fitted mesh is shown in Figure 5. It contains 11376 quadratic NURBS elements and the wall-normal element size near the cylinder is 0.0173. By using NURBS elements, we can exactly represent the circular boundary geometry, completely eliminating geometry error. Time steps for the body-fitted computations are selected to ensure that there are roughly 200 time steps per period in periodic solutions.

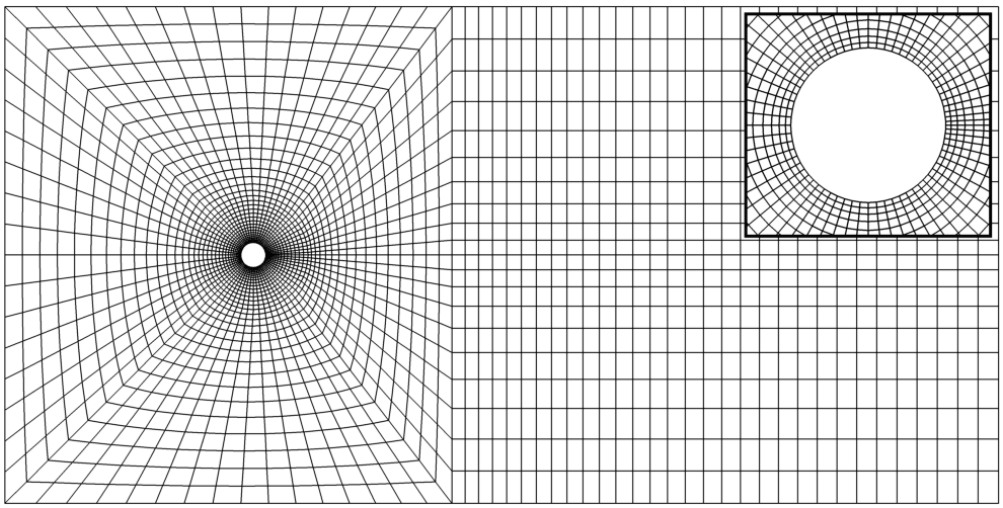


Figure 5: The body-fitted reference mesh.

3.4.3. Comparison of results

We consider four quantities of interest for this problem, although some are relevant only in certain flow regimes. We always measure the drag coefficient, C_D, defined as $2F_D/\rho U^2 d$, where F_D is the drag force or horizontal component of traction integrated over the cylinder surface, ρ is the fluid density, U is the inflow velocity, and d is the diameter of the cylinder. At low Reynolds numbers cases that reach steady solutions, we consider the bubble recirculation length, L_W. L_W measures how far the stationary vortices occurring at low Reynolds numbers extend downstream of the cylinder. It is defined precisely in Lima E Silva et al. [99]. At higher Reynolds numbers, where flow symmetry breaks, leading to periodic solutions, we consider the lift coefficient, C_L, and the Strouhal number, St. C_L is defined as $2F_L/\rho U^2 d$, where F_L is the lift force or vertical component of traction integrated over the cylinder surface. St is given as fd/U, where f is the frequency of vortex shedding. The vortex shedding only occurs if the Reynolds number is sufficiently high. We identify the frequency of vortex shedding with the frequency of oscillation in C_L. In periodic solutions, the reported value of C_L is the amplitude of its oscillation and the reported value of C_D is its time average.

The evaluations of C_L and C_D rely on computing the traction at the fluid–structure interface. A naive evaluation of traction from the fluid Cauchy stress, $-\boldsymbol{\sigma}_1\mathbf{n}_1$, will converge poorly to the true

traction, so we prefer to use variationally-consistent, conservative definitions of traction [56, 104]. In the case of Nitsche's method, the appropriate discrete traction on surfaces with weakly enforced Dirichlet boundary conditions includes the penalty terms, matching the traction boundary condition of the FSI structural sub-problem (11):

$$\begin{aligned}\mathbf{t}^h = &- \boldsymbol{\sigma}_1^h \mathbf{n}_1 - \rho_1 \left\{ \left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h \right) \cdot \mathbf{n}_1 \right\}_- \left(\mathbf{u}_1^h - \mathbf{u}_2 \right) \\ &+ \tau_{\mathrm{TAN}}^B \left(\left(\mathbf{u}_1^h - \mathbf{u}_2 \right) - \left(\left(\mathbf{u}_1^h - \mathbf{u}_2 \right) \cdot \mathbf{n}_1 \right) \mathbf{n}_1 \right) + \tau_{\mathrm{NOR}}^B \left(\left(\mathbf{u}_1^h - \mathbf{u}_2 \right) \cdot \mathbf{n}_1 \right) \mathbf{n}_1 \,, \end{aligned} \tag{24}$$

where $\{ \cdot \}_-$ denotes the negative part of the bracketed quantity, that is, $\{\mathcal{A}\}_- = \mathcal{A}$ if $\mathcal{A} < 0$ and $\{\mathcal{A}\}_- = 0$ if $\mathcal{A} \geq 0$. In this case, Γ_{I} is stationary, so $\mathbf{u}_2 = \mathbf{0}$. On a surface with a strongly-enforced Dirichlet condition, as seen in the body-fitted computation, the conservative traction must satisfy

$$\int_{(\Gamma_{\mathrm{I}})_t} \mathbf{w}_1^h \cdot \mathbf{t}^h \, d\Gamma \;=\; B_1^{\mathrm{VMS}}(\{\mathbf{w}_1^h, q^h\}, \{\mathbf{u}_1^h, p^h\}; \hat{\mathbf{u}}^h) - F_1^{\mathrm{VMS}}(\{\mathbf{w}_1^h, q^h\}) \tag{25}$$

for all $\mathbf{w}_1^h$ in an expanded discrete velocity test space that does not strongly enforce the Dirichlet condition. To obtain the i^{th} component of the integral of this conservative traction over the boundary $(\Gamma_{\mathrm{I}})_t$, we would evaluate the right-hand side of Eq. (25) with $\mathbf{w}_1^h = \mathbf{e}_i$ on $(\Gamma_{\mathrm{I}})_t$ and $q^h = 0$. The desired $\mathbf{w}_1^h$ is straightforward to construct from shape functions that satisfy the partition of unity property.

First, using three levels of adaptive quadrature, we investigate the effects of different penalty values. We consider only the case in which $\tau_{\mathrm{NOR}}^B = \tau_{\mathrm{TAN}}^B = \tau^B$. In the current non-dimensional setting, we state these penalty values without units. However, they have the physical interpretation of traction per unit difference in speed (between fluid and structure), and the corresponding dimensions of pressure per speed. Further, we would generally expect these values to increase with mesh refinement, so the numbers given here should not be blindly transplanted into other computations without first applying dimensional analysis and considering the relative level of refinement.

Tables 1 and 2 collect the results of applying $\tau^B = 10^2$ and $\tau^B = 10^3$ at various Reynolds numbers for meshes M1 and M2. For comparison, we also give ranges of typical values for these quantities from the CFD literature, specifically [96–103], in Table 3. Figure 6 displays several snapshots of velocity and pressure fields computed using the variational immersed boundary method with $\tau^B = 10^2$ and $l = 3$ on M1.

From this study, we find that the penalties of the order 10^1 are not consistently stable, while penalties of the order 10^4 and higher become costly to compute with, due to their effect on the conditioning of the problem. This suggests that, while we do not provide a formula for τ^B, it may be chosen from within a wide range of computable values while still providing accurate results. As long as the penalty is chosen such that the computation converges with a reasonable amount

M1						
	$Re = 40$			$Re = 80$		
	C_D	L_W		C_D	C_L	St
$\tau^B = 10^2$	1.611	2.26		1.413	±0.252	0.159
$\tau^B = 10^3$	1.611	2.26		1.411	±0.250	0.159
Body-fitted	1.612	2.27		1.415	±0.254	0.159
	$Re = 100$			$Re = 200$		
	C_D	C_L	St	C_D	C_L	St
$\tau^B = 10^2$	1.384	±0.338	0.170	1.369	±0.694	0.200
$\tau^B = 10^3$	1.381	±0.336	0.171	1.369	±0.696	0.200
Body-fitted	1.386	±0.341	0.170	1.378	±0.706	0.200

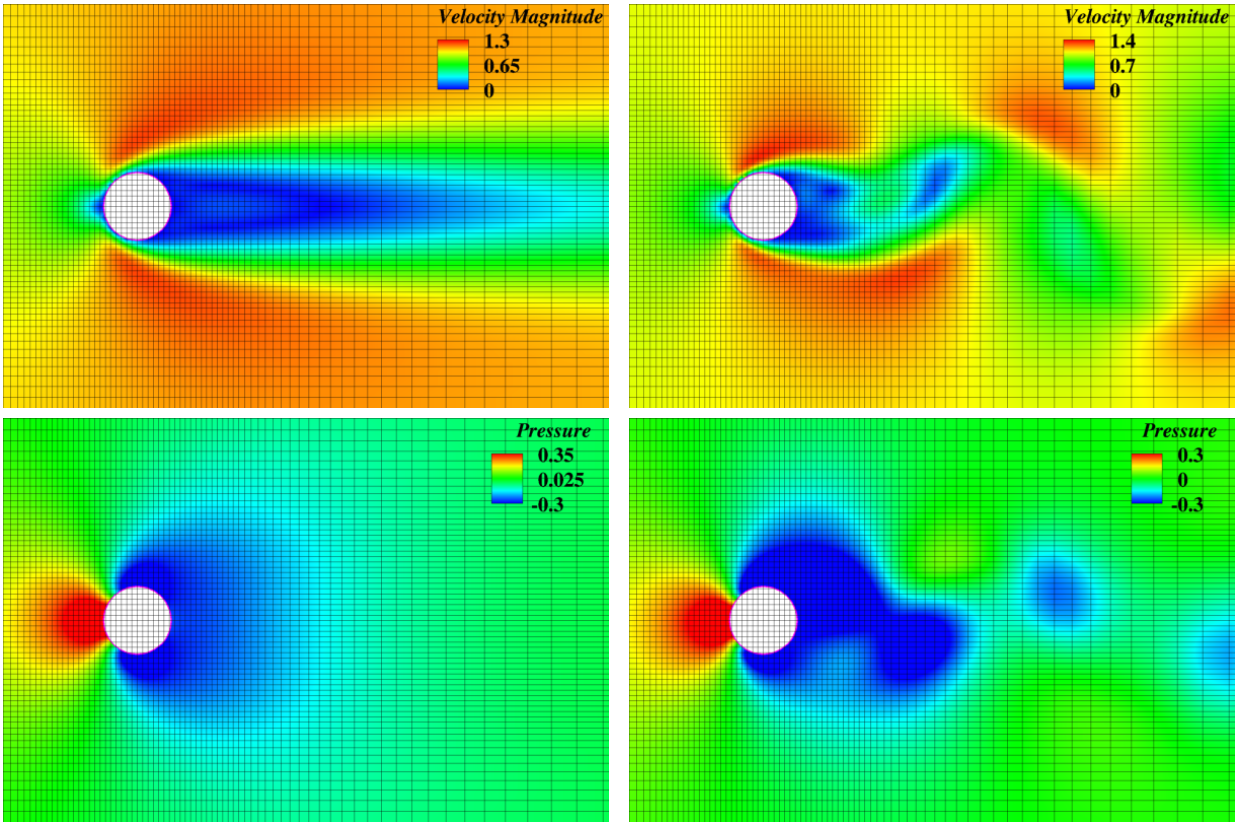
Table 1: Comparison of quantities of interest with various penalty values and $l = 3$ levels of adaptive quadrature on mesh M1.

M2						
	$Re = 40$			$Re = 80$		
	C_D	L_W		C_D	C_L	St
$\tau^B = 10^2$	1.612	2.27		1.415	±0.254	0.159
$\tau^B = 10^3$	1.612	2.27		1.415	±0.253	0.158
Body-fitted	1.612	2.27		1.415	±0.254	0.159
	$Re = 100$			$Re = 200$		
	C_D	C_L	St	C_D	C_L	St
$\tau^B = 10^2$	1.386	±0.341	0.170	1.378	±0.706	0.200
$\tau^B = 10^3$	1.386	±0.341	0.170	1.378	±0.705	0.200
Body-fitted	1.386	±0.341	0.170	1.378	±0.706	0.200

Table 2: Comparison of quantities of interest with various penalty values and $l = 3$ levels of adaptive quadrature on mesh M2.

	$Re = 40$			$Re = 80$		
	C_D	L_W		C_D	C_L	St
Literature	1.52–1.63	2.24–2.32		1.34–1.44	± 0.26	0.15–0.16
	$Re = 100$			$Re = 200$		
	C_D	C_L	St	C_D	C_L	St
Literature	1.33–1.43	± 0.30–0.34	0.16–0.17	1.31–1.45	±0.64–0.71	0.19–0.20

Table 3: Ranges of typical values of quantities of interest from the CFD literature [96–103]



(a) $Re = 40$ (b) $Re = 100$

Figure 6: Visualizations of velocity and pressure fields about a cylinder immersed in M1, showing both steady ($Re = 40$) and time-periodic ($Re = 100$) solutions. Results are obtained using $\tau^B = 10^2$ and $l = 3$.

of work, our Nitsche-based variational immersed boundary method achieves good agreement (at the quantity of interest level) with a much more refined boundary-fitted computation. In some cases on M1 we see slightly worse results with the higher value of τ^B. This is consistent with the idea that approaching strong enforcement of Dirichlet boundary conditions on a mesh that is too coarse to resolve the boundary layer will result in lower quality solutions. Some violation of the no-slip boundary condition can in fact be desirable on a coarse mesh, as it imitates the presence of a boundary layer [53–57].

M2						
	$Re = 40$			$Re = 80$		
	C_D	L_W		C_D	C_L	St
$l = 0$	1.620	2.31		1.473	±0.247	0.158
$l = 3$	1.612	2.27		1.415	±0.254	0.159
Body-fitted	1.612	2.27		1.415	±0.254	0.159
	$Re = 100$			$Re = 200$		
	C_D	C_L	St	C_D	C_L	St
$l = 0$	1.477	±0.335	0.169	1.613	±0.631	0.199
$l = 3$	1.386	±0.341	0.170	1.378	±0.706	0.200
Body-fitted	1.386	±0.341	0.170	1.378	±0.706	0.200

Table 4: Comparison of quantities of interest with penalty $\tau^B = 10^2$ and different levels, l, of adaptive quadrature. The $l = 3$ results are repeated from Table 2 for the reader's convenience.

Finding that the penalty $\tau^B = 10^2$ applied to discretization M2 produces quantities of interest that largely agree with our body-fitted reference and results from the literature, we proceed to consider the effect of adaptive quadrature with this value of the penalty parameter. These results are collected in Table 4 and again compared with our reference computation. The degradation of results in the absence of adaptive quadrature demonstrates the effects of error introduced by under-integrating discontinuous functions. This degradation becomes more severe with increased Reynolds number, suggesting that adaptive quadrature would be especially crucial in computations involving turbulent flows. The agreement of the non-boundary-fitted results with those computed on a refined, body-fitted reference mesh shows that the non-boundary-fitted methodology is accurate, even when the boundary layer is composed of larger, haphazardly-cut elements.

Remark 6. The results in Table 2 demonstrate interesting correlations: removal of adaptive quadrature consistently increases drag and decreases lift. This suggests that inadequate integration may tend to overestimate viscous forces and underestimate pressure forces, but we do not investigate that question further in this paper. Table 2 also shows that the errors due to the lack of adaptive quadrature grow with Reynolds number. Thus adaptive quadrature is critical when the proposed technique is taken to the high Reynolds number regime (and for example, turbulence).

4. Immersed shell structures

The preceding examples involve flow around bulky objects. We would also like to study flow around extremely thin immersed structures, such as heart valve leaflets. The method developed in Section 3 could be applied if the thin structures were fully modeled as 3D solids and immersed into a sufficiently refined fluid mesh. However, we would prefer a computationally more efficient approach that models the solid as a two-dimensional manifold shell structure. Such a technique would necessarily decouple the fluid resolution from the structure thickness.

This presents a conceptual difficulty. The exact solution for the pressure around a shell structure may be discontinuous at the structure. Since, for practical reasons discussed in Section 1, we are committed to using a non-boundary-fitted method, the fluid discretization cannot be informed by the structure's position. This means that our fluid approximation space cannot be selected in such a way that the pressure basis functions are themselves discontinuous at the immersed boundary. This implies an inherent approximation error in the pressure field. This error will converge slowly for polynomial bases [105]. Nonetheless, we believe that solutions of sufficient accuracy for engineering purposes can be obtained in this fashion and we focus on developing a robust method for obtaining these solutions.

4.1. Reduction of Nitsche's method to the penalty method

Consider integrating the boundary terms of Eq. (12) over both sides of a thin immersed shell structure. If the velocity and pressure approximation spaces are continuous through the vanishing thickness of the shell (and the velocity approximation space is continuously differentiable), then the dependence of the consistency and adjoint consistency terms on the normal vector will cause contributions from opposing sides to cancel one another. The only remaining terms will be the penalty and the inflow stabilization. In the case of an immersed shell structure, we may view the inflow term as a velocity-dependent penalty. The Nitsche-type formulation given in Eq. (12) therefore reduces to the following penalty method

$$
\begin{aligned}
&B_1^{\mathrm{VMS}}\left(\{\mathbf{w}_1^h, q^h\}, \{\mathbf{u}_1^h, p^h\}; \hat{\mathbf{u}}^h\right) - F_1^{\mathrm{VMS}}\left(\{\mathbf{w}_1^h, q^h\}\right) \\
&\quad - \int_{(\Gamma_1)^-} \mathbf{w}_1^h \cdot \rho_1 \left(\left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h\right) \cdot \mathbf{n}_1\right)\left(\mathbf{u}_1^h - \mathbf{u}_2\right)\, d\Gamma \\
&\quad + \int_{\Gamma_1} \tau_{\mathrm{TAN}}^B \left(\mathbf{w}_1^h - \left(\mathbf{w}_1^h \cdot \mathbf{n}_1\right)\mathbf{n}_1\right) \cdot \left(\left(\mathbf{u}_1^h - \mathbf{u}_2\right) - \left(\left(\mathbf{u}_1^h - \mathbf{u}_2\right) \cdot \mathbf{n}_1\right)\mathbf{n}_1\right)\, d\Gamma \\
&\quad + \int_{\Gamma_1} \tau_{\mathrm{NOR}}^B \left(\mathbf{w}_1^h \cdot \mathbf{n}_1\right)\left(\left(\mathbf{u}_1^h - \mathbf{u}_2\right) \cdot \mathbf{n}_1\right)\, d\Gamma \;=\; 0
\end{aligned}
\tag{26}
$$

when the approximation spaces $\mathcal{V}_u^h$ and $\mathcal{V}_p^h$ are sufficiently regular around the shell. In Section 4.6, we will demonstrate that this method is sufficient to compute quantities of interest in the benchmark

problem of flow over an elastic beam.

Remark 7. The inflow term in Eq. (26) is often much smaller in magnitude than the $\tau^B_{(\cdot)}$ terms, when practical values of the penalty parameters are selected. For the problems considered in this paper, the inflow term may be omitted entirely, with little to no effect. We have, however, included it, for consistency with our formulation for volumetric objects. It is possible that this term may play a significant role in other FSI problems.

To determine the velocity and pressure about an immersed valve in its closed state, a method must be capable of developing nearly hydrostatic solutions in the presence of large pressure gradients. Penalty forces will only exist if there are nonzero violations of kinematic constraints. A pure penalty method rules out the desired hydrostatic solutions: every term that could resist the pressure gradient to satisfy balance of linear momentum depends on velocity. Increasing β may diminish leakage through a structure, but it is a well-known disadvantage of penalty methods that extreme values of penalty parameters will adversely affect the numerical solvability of the resulting problem. This motivates us to return to Eq. (2) and develop a method that does not formally eliminate the multiplier field.

4.2. Reintroducing the multipliers

Since the introduction of constraints tends to make discrete problems more difficult to solve, we will only reintroduce a scalar multiplier field to strengthen enforcement of the no-penetration part of the FSI kinematic constraint, rather than the vector-valued multiplier field of Eq. (2). The viscous, tangential component of the constraint will continue to be enforced by only the penalty τ^B_{TAN}. This may be thought of as a formal elimination of just the tangential component of the multiplier field, which also retains the ability to allow the flow to slip at the boundary, which tends to produce more accurate fluid solutions as discussed in Section 3.1. For clarity, we redefine the FSI boundary terms on the mid-surface of the shell structure, Γ_t, rather than considering the full boundary, Γ_{I}. This means that constants in the current formulation may differ from those of Eq. (2) by factors of two. We arrive, then, at the formulation

$$B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}; \hat{\mathbf{u}}) - F_1(\{\mathbf{w}_1, q\}) + \int_{\Gamma_t} \mathbf{w}_1 \cdot (\lambda_n \mathbf{n}_2)\, d\Gamma + \int_{\Gamma_t} \mathbf{w}_1 \cdot \beta(\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma = 0, \tag{27}$$

$$B_2(\mathbf{w}_2, \mathbf{u}_2) - F_2(\mathbf{w}_2) - \int_{\Gamma_t} \mathbf{w}_2 \cdot (\lambda_n \mathbf{n}_2)\, d\Gamma - \int_{\Gamma_t} \mathbf{w}_2 \cdot \beta(\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma = 0, \tag{28}$$

$$\int_{\Gamma_t} \delta\lambda_n \mathbf{n}_2 \cdot (\mathbf{u}_1 - \mathbf{u}_2)\, d\Gamma = 0, \tag{29}$$

where λ_n is the new scalar multiplier field and, to emphasize the relation to Eq. (2), the penalty force has not been split into normal and tangential components. The consistency and adjoint consistency

terms associated with eliminating the tangential component of the multiplier have been omitted under the assumption that they will vanish after integrating over both sides of the thin shell, as discussed in Section 4.1.

4.2.1. Implementation of the Lagrange multipliers

We wish to implement the constraint between the fluid and structure solutions in a way that is minimally disruptive to the two sub-problems, allowing existing methods for computational fluid and solid mechanics to be applied to each. A monolithic solution for the velocities and multipliers would limit our ability to quickly interchange fluid or structure formulations and, as a mixed formulation, would require either special choices of approximation spaces [106] or additional stabilization terms [107] to satisfy the Babuška-Brezzi stability conditions. Appropriate approximation spaces or stabilization terms are not obvious for the current case. This section discusses two alternative solution strategies for implementing the Lagrange multipliers.

The unconstrained problem that follows from considering $\boldsymbol{\lambda}$ to be fixed is similar to that following from the penalty method. The multiplier simply enters each sub-problem as a prescribed boundary traction. We consider, then, an iterative strategy that updates $\boldsymbol{\lambda}$ between solutions of such unconstrained problems.

Our starting point is the iterative method independently introduced by Hestenes [108] and Powell [109] in 1969. This method attempts to minimize an augmented Lagrangian of the form

$$L(x, \lambda) = f(x) + \lambda g(x) + \beta \|g(x)\|^2 \tag{30}$$

where x is the primal variable, λ is the multiplier, $\beta > 0$ is a penalty parameter, and $f(x)$ is an objective function that we seek to minimize, subject to the constraint $g(x) = 0$. The method consists of starting with $\lambda = 0$ and repeating the steps

1. Solve $x \leftarrow \arg\min L(\cdot, \lambda)$, where λ is treated as a fixed parameter.
2. Update the multiplier by $\lambda \leftarrow \lambda + \beta g(x)$,

until $\|g(x)\| < \epsilon$. We may attempt to apply this strategy to our problem by representing the field λ_n by samples at quadrature points of Γ_t and repeating the following steps

1. Solve for approximate fluid and structure velocities $\mathbf{u}_1^h$ and $\mathbf{u}_2^h$, treating λ_n as fixed data. We discuss specific solution strategies for this unconstrained (but still coupled) problem in Section 4.5.
2. Update the multiplier field by $\lambda_n \leftarrow \lambda_n + \tau^B_{\mathrm{NOR}}(\mathbf{u}_1^h - \mathbf{u}_2^h) \cdot \mathbf{n}_2$, where λ_n and $\mathbf{u}_i^h$ are evaluated at the quadrature points of Γ_t,

until $\left(\int_{\Gamma_t} |(\mathbf{u}_1^h - \mathbf{u}_2^h) \cdot \mathbf{n}_2|^2 \, d\Gamma \right)^{1/2} < \epsilon$.

However, if the approximation spaces are not selected in a stable way, there may not be a solution to the discrete problem and the iteration may never converge to arbitrary ϵ. We observe that, in some cases, specifically those discussed in Section 4.3, the iteration does appear to converge linearly. However, for more general fluid and structure geometries, the procedure does not appear to converge. It may be possible, and practically effective, to formulate a variety of *ad hoc* termination criteria, but, for problems in which the iterative procedure will not converge, we consider only the case of applying a single iteration within each time step and using the updated λ_n as the initial guess for the (severely truncated) iteration within the next time step. In this case, the multiplier becomes an accumulation of penalty tractions from previous time steps. This is equivalent to replacing the multiplier and normal penalty terms

$$\int_{\Gamma_t} (\mathbf{w}_1 - \mathbf{w}_2) \cdot (\lambda_n \mathbf{n}_2)\, d\Gamma + \int_{\Gamma_t} ((\mathbf{w}_1 - \mathbf{w}_2) \cdot \mathbf{n}_2)\, \tau^B_{\mathrm{NOR}}\, ((\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2)\; d\Gamma \tag{31}$$

by a penalization of (a backward Euler evaluation of) the time integral of pointwise normal velocity differences on the immersed surface Γ_t

$$\int_{\Gamma_t} \left\{ \frac{\tau^B_{\mathrm{NOR}}}{\Delta t} (\mathbf{w}_1(\mathbf{x},t) - \mathbf{w}_2(\mathbf{x},t)) \cdot \mathbf{n}_2(\mathbf{x},t) \right. \\ \left. \int_0^t \left(\mathbf{u}_1(\varphi_\tau(\varphi_t^{-1}(\mathbf{x})),\tau) - \mathbf{u}_2(\varphi_\tau(\varphi_t^{-1}(\mathbf{x})),\tau) \right) \cdot \mathbf{n}_2(\varphi_\tau(\varphi_t^{-1}(\mathbf{x})),\tau)\, d\tau \right\}\; d\Gamma\,, \tag{32}$$

where $\varphi_\tau(\mathbf{X})$ gives the spatial position at time τ of material point $\mathbf{X} \in \Gamma_0$ and the measure $d\Gamma$ corresponds to the integration variable $\mathbf{x} \in \Gamma_t$. That the time integral in Eq. (32) is evaluated using the backward Euler method is demonstrated in the following exposition. First define (at fixed $\mathbf{X}$)

$$I(t) = \frac{\tau^B_{\mathrm{NOR}}}{\Delta t} \int_0^t (\mathbf{u}_1(\tau) - \mathbf{u}_2(\tau)) \cdot \mathbf{n}_2(\tau)\, d\tau\,. \tag{33}$$

The time rate-of-change of the integral I will be its integrand

$$\dot{I} = \frac{\tau^B_{\mathrm{NOR}}}{\Delta t} (\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{n}_2\,. \tag{34}$$

We approximate I at time $t^{n+1+\alpha_f}$ by

$$I^{n+1+\alpha_f} = I^{n+\alpha_f} + \Delta t \dot{I}^{n+1+\alpha_f} \tag{35}$$

where $I^{n+\alpha_f}$ is an accumulation of previous single-iteration approximations to λ_n and $\Delta t \dot{I}^{n+1+\alpha_f}$

is the current time step's penalty forcing, which is the penalty τ^B_{NOR} times the α-level[1] velocity difference between the structure and fluid. Eq. 35 is precisely the backward Euler algorithm for computing I. Thus the term of Eq. (32) is accounted for in a fully implicit manner within the discrete solution process, using a manifestly stable time integrator. An order of accuracy is lost relative to the generalized-α scheme, but, in our application, other considerations have driven the time step down to small enough values for this distinction to have few practical implications; we are primarily concerned with stability.

Integrating a constraint residual in time is not a new concept for approximation of a Lagrange multiplier. The differential equation given in Eq. (34) resembles the method of artificial compressibility, devised by Chorin [110] in 1967 and widely used since to simulate incompressible flows (see, e.g., Brooks and Hughes [68]). In this method, the approximated Lagrange multiplier p representing the pressure evolves through time in an analogous way to I:

$$\partial_t p = -\frac{1}{\delta}\boldsymbol{\nabla}\cdot\mathbf{u}\,, \tag{36}$$

where the constraint is $\boldsymbol{\nabla}\cdot\mathbf{u} = 0$ (instead of $(\mathbf{u}_1 - \mathbf{u}_2)\cdot\mathbf{n}_2 = 0$), $1/\delta$ is the penalty parameter, and the difference in sign is due to the arbitrary choice of sign with which λ enters the augmented Lagrangian formulation (1). A physical interpretation of this, similar to Chorin's original formulation of Eq. (36) in terms of a fictitious density variable, is that we are penalizing a displacement penetration of the fluid through the leaflet, using the penalty $\tau^B_{\text{NOR}}/\Delta t$. This interpretation makes clear how penalizing the time integral of velocity prevents the steady creep of flow through a barrier.

4.3. Managing pressure approximation error with stabilization

Due to the poor approximation properties of a pressure space that does not allow discontinuities on the surface of the shell, we expect the pressure to converge slowly. We show that reasonable levels of refinement can circumvent this difficulty for certain types of problems in Section 4.6. However, in problems with large pressure jumps, unphysical compression incurred by the poorly-approximated pressure will ruin even the qualitative character of solutions. In Section 4.3.1, we use a model problem to show that this effect becomes practically important in the analysis of heart valves. Then, in Section 4.3.2, we introduce and test a proposed solution.

4.3.1. A demonstration of the effect of pressure approximation error

We now consider a simplified model of a closed valve, with fluid properties and boundary conditions similar to those found in cardiovascular applications. We show that we cannot develop hydrostatic solutions with a reasonable spatial discretization and practical time step.

[1] See Bazilevs et al. [74] for a discussion of generalized-α time integration using this notation.

Consider an axis-aligned 2 cm $\times$ 2 cm $\times$ 2 cm cube, filled with an incompressible Newtonian fluid of density $\rho_1 = 1.0$ g cm^{-3} and viscosity $\mu = 3.0 \times 10^{-2}$ g cm^{-1}s^{-1}. The vertical faces have a no-slip boundary condition, the bottom has a zero-traction outflow boundary condition, and the top has a pressure traction of 120 mmHg. The length scale, fluid properties, and pressure difference produce conditions comparable to those surrounding a closed aortic valve in diastole. Now consider immersing a rigid, impermeable horizontal plate into this cube, blocking its entire cross section at a distance of 1.1 cm from the bottom. The exact solution for this problem should be hydrostatic, with a discontinuous pressure at the location of the plate. However, in an immersed-boundary finite element discretization, the continuity of the pressure approximation functions through the plate means that the irregularity of the exact solution cannot be exactly reproduced in a computation.

Remark 8. The plate's height of 1.1 cm is deliberately selected so that the plate will never coincide with an element boundary for any uniform division of the cube into 2^n elements in the z-direction. This may be seen by considering the fact that 0.1_{10} is a repeating fraction in binary. Even if a discontinuous pressure basis is used, the discontinuities will not be located on the structure.

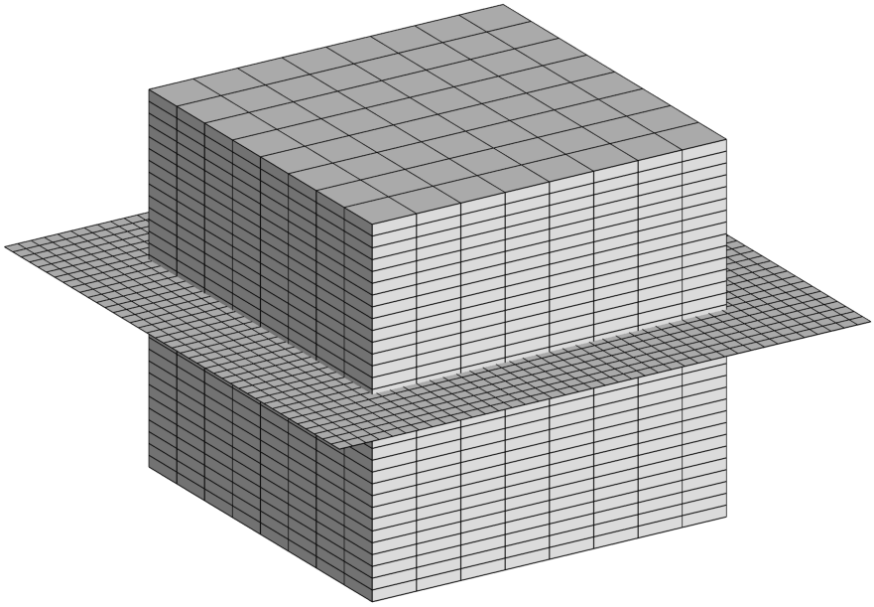


Figure 7: The computational mesh used for the closed-valve model problem.

We now compute a solution to this problem, starting from homogeneous initial conditions for the velocity and using Lagrange multipliers to enforce the no-penetration condition on the shell. For the mesh we use a trivariate C^1-continuous quadratic B-spline patch, uniformly refined into $8 \times 8 \times 32$ elements. The quadrature rule for surface integrals over the immersed plate is a sum of Gaussian quadrature rules on 40×40 quadrilaterals, evenly dividing a 3 cm $\times$ 3 cm square surface,

cutting through the channel as shown in Figure 7. Surface quadrature points falling outside of the channel do not contribute to integrals. We find that, if large flow velocities develop with the given boundary conditions, backflow divergence may occur, and we apply the outflow stabilization discussed in Section 3.1.1 to both traction boundaries, with $\gamma = 0.5$.

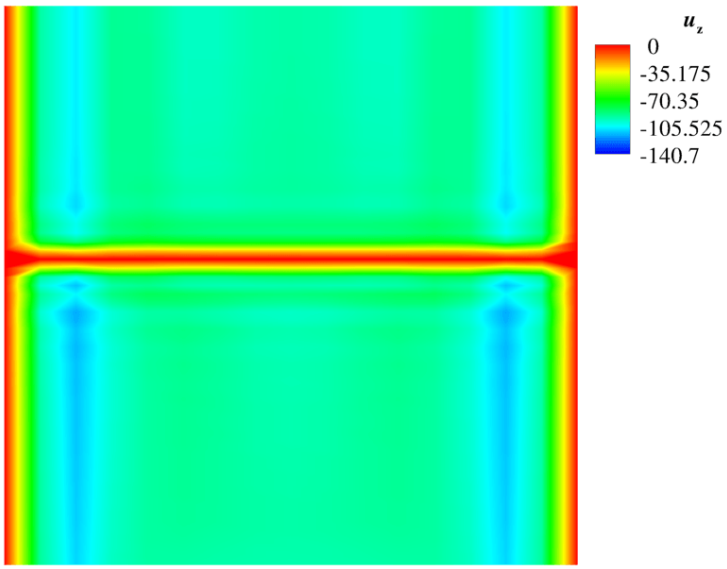


Figure 8: The z-component of velocity, in cm s^{-1}, for a highly unphysical steady-state flow solution through a blocked channel, as computed with $\Delta t = 10^{-4}$ s and no modifications to fluid stabilization terms. The fluid spuriously compresses to meet the velocity constraint imposed by the barrier while maintaining a large downward flow through the channel.

We consider the time step $\Delta t = 10^{-4}$ s practical for computing dynamic FSI at the time scale of a cardiac cycle. Computing with this time step and using the iterative multiplier approximation of Section 4.2.1, we see a highly unphysical behavior. Figure 8 shows the vertical velocity component on a slice of the resulting solution, after the volumetric flow rate through the top of the cube reached a steady value ($t > 0.01$ s). While the Lagrange multipliers enforce the constraint very effectively[2], there is still a significant flow through the top face of the cube. The steady-state volumetric flow rate is 355.2 mL s^{-1}, which is unacceptable for simulation of a valve structure that exists primarily to block flow. This would be a typical flow rate through an *open* aortic valve, during systole [111]. The flow rate varies between cross-sections of the channel, which obviously violates the incompressibility condition.[3] The compression caused by local pressure approximation error pollutes the entire velocity solution.

[2]As discussed in Section 4.2.1, we do not always expect the constraint to fully converge, since we have not selected a stable discretization, but it seems that, for this simple problem, the iterative approach converges. This is not, in general, expected or found in calculations with different immersed geometries.

[3]The VMS formulation discretely satisfies global mass conservation for any reasonable test space (which may be seen by setting $q = 1$ and $\mathbf{w} = \mathbf{0}$ in Eq. (13)). However, we have no guarantee of local mass conservation.

We now refine the time step by two orders of magnitude to $\Delta t = 10^{-6}$ s. The steady-state flow rate shrinks by roughly the same factor as the time step, leveling off at 3.877 mL s^{-1} after $t = 1.2 \times 10^{-4}$ s. Refining further in time to $\Delta t = 10^{-8}$ s, the flow rate reduces again to 3.877×10^{-2} mL s^{-1}. It seems that, with a sufficiently small time step, we are able to approach the desired hydrostatic solution. We believe that this steep time step requirement traces back to poor approximation of the pressure near the immersed shell structure. Reducing the time step causes the momentum stabilization constant τ_{M}, which multiplies the momentum residual in Eq. (13), to be smaller near the shell. It simultaneously causes the pressure stabilization constant τ_{C} to be larger, penalizing velocity divergence.

4.3.2. A proposed solution

A natural way to replicate these effects without reducing the time step is to locally shrink τ_{M} in elements near the immersed structure. To accomplish this, we modify the definition of τ_{M} in Eq. (16) to be

$$\tau_{\mathrm{M}} = \left(s \left(\frac{C_t}{\Delta t^2} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{G} (\mathbf{u} - \hat{\mathbf{u}}) + C_I \left(\frac{\mu}{\rho_1} \right)^2 \mathbf{G} : \mathbf{G} \right) \right)^{-1/2} , \tag{37}$$

which affects all quantities defined in terms of τ_{M}, such as $\mathbf{u}'$ and τ_{C}. The new factor $s > 1$ is dimensionless and allowed to vary in space. For most of the domain, $s = 1$, but near the shell, we may make it larger, with the effect of reducing τ_{M}. To smooth the transition between larger and smaller values of s, we define it as a nodal variable, using the pressure approximation space. For nodes corresponding to pressure basis functions with supports intersecting the shell (i.e. containing quadrature points for the integration rule on Γ_t), this nodal variable is set to $s^{\mathrm{shell}} \geq 1$. For all other nodes, it is set to the usual value of 1. If the pressure shape functions form a partition of unity, then s will be uniformly equal to s^{shell} on elements intersecting the shell.

Remark 9. From stability and convergence analysis of analogous stabilized methods for the steady Stokes and Oseen problems, we see that, for stability and asymptotic convergence, τ_{M} is subject only to upper bounds. It is typically chosen to saturate these bounds, to reduce constants in the error estimate [72]. However, for flow conditions and approximation spaces of interest, it seems that we may improve the qualitative character of solutions at coarse discretizations by choosing smaller values of τ_{M} (by using $s^{\mathrm{shell}} > 1$).

We now test this preliminary solution by applying it to the model problem of the previous section. We investigate the effect of s^{shell} at the practical time step of 10^{-4} s. Because the time step for our application is small relative to the spatial discretization of the fluid, τ_{M} scales roughly like Δt, so we hypothesize that scaling $s^{1/2}$ by a factor c will have comparable effects to scaling

Δt by c^{-1}. To mirror the time step changes tested above, we consider $s^{\text{shell}} = 10^4$ and $s^{\text{shell}} = 10^8$, which, as stabilization terms are concerned, imitate time step reductions by factors of 10^{-2} and 10^{-4} respectively. Table 5 summarizes these results, comparing the effects of changing Δt and s^{shell}. The imitated reduction of time step through s appears to have a comparable effect (at the order-of-magnitude level) to actual reduction in time step.

Δt	Volumetric flow rate		s^{shell}
10^{-4} s	355.2 mL s^{-1}		1
10^{-6} s	3.877 mL s^{-1}	4.037 mL s^{-1}	10^4
10^{-8} s	3.877×10^{-2} mL s^{-1}	4.048×10^{-2} mL s^{-1}	10^8

Table 5: Comparison of the effects of time step and s^{shell} on volumetric flow rate through a blocked tube. The left column has $s^{\text{shell}} = 1$ for all entries and the right column has $\Delta t = 10^{-4}$ s.

Remark 10. An undesirable consequence of increasing s^{shell} is that the resulting penalty nature of continuity enforcement near the shell harms the conditioning of the discrete problem. Due to the simplistic nature of the blocked tube model problem, conditioning is not a significant issue, but applying the modified stabilization terms to more complex calculations, such as those presented in Section 5.4, increases the cost of sufficient iterative solution in the linear problem at each Newton step. The development of a suitable preconditioner may avert this difficultly, but is beyond the scope of the current work.

4.4. Treatment of shell structure mechanics

In this section, we give concrete form to the structure sub-problem (11). We assume that the structure is a thin shell, represented mathematically by its mid-surface. Further, we assume this surface to be piecewise C^1-continuous and apply the Kirchhoff–Love shell formulation and isogeometric discretization studied by Kiendl et al. [63, 112, 113].

4.4.1. Basic kinematics of a Kirchhoff–Love shell

The current and reference configurations of the shell mid-surface are given by the parametric mappings $\mathbf{x}(\xi_1, \xi_2)$ and $\mathbf{X}(\xi_1, \xi_2)$. Assuming the range $\{1, 2\}$ for Greek letter indices, we define bases

$$\mathbf{g}_\alpha = \frac{\partial \mathbf{x}}{\partial \xi_\alpha} , \tag{38}$$

$$\mathbf{g}_3 = \frac{\mathbf{g}_1 \times \mathbf{g}_2}{\|\mathbf{g}_1 \times \mathbf{g}_2\|} , \tag{39}$$

and

$$\mathbf{G}_\alpha = \frac{\partial \mathbf{X}}{\partial \xi_\alpha} \,, \tag{40}$$

$$\mathbf{G}_3 = \frac{\mathbf{G}_1 \times \mathbf{G}_2}{\|\mathbf{G}_1 \times \mathbf{G}_2\|} \,, \tag{41}$$

in the current and reference configurations, which yield metric tensors

$$g_{\alpha\beta} = \mathbf{g}_\alpha \cdot \mathbf{g}_\beta \,, \tag{42}$$

$$G_{\alpha\beta} = \mathbf{G}_\alpha \cdot \mathbf{G}_\beta \,, \tag{43}$$

and curvature coefficients

$$b_{\alpha\beta} = -\mathbf{g}_\alpha \cdot \frac{\partial \mathbf{g}_3}{\partial \xi_\beta} = \frac{\partial \mathbf{g}_\alpha}{\partial \xi_\beta} \cdot \mathbf{g}_3 \,, \tag{44}$$

$$B_{\alpha\beta} = -\mathbf{G}_\alpha \cdot \frac{\partial \mathbf{G}_3}{\partial \xi_\beta} = \frac{\partial \mathbf{G}_\alpha}{\partial \xi_\beta} \cdot \mathbf{G}_3 \,. \tag{45}$$

Using kinematic assumptions and mathematical manipulations given in Kiendl [113], we split the in-plane Green-Lagrange strain $E_{\alpha\beta}$ into membrane and curvature contributions

$$E_{\alpha\beta} = \varepsilon_{\alpha\beta} + \xi_3 \kappa_{\alpha\beta} \,, \tag{46}$$

where

$$\varepsilon_{\alpha\beta} = \frac{1}{2}(g_{\alpha\beta} - G_{\alpha\beta}) \,, \tag{47}$$

$$\kappa_{\alpha\beta} = B_{\alpha\beta} - b_{\alpha\beta} \,, \tag{48}$$

are the membrane strain and curvature tensors, respectively, at the shell mid-surface, $\xi_3 \in [-h_{\text{th}}/2, h_{\text{th}}/2]$ is the through-thickness coordinate and h_{th} is the (variable) shell thickness.

4.4.2. St. Venant–Kirchhoff material model

For the purposes of this paper, we assume a St. Venant–Kirchhoff material, in which the second Piola-Kirchhoff stress, $\mathbf{S}$, is computed from a constant elasticity tensor, $\mathbb{C}$, applied to $\mathbf{E}$. We are well aware of the shortcomings of this material model under states of high compression, but these are precluded in the current situation. The in-plane stresses due to extension and bending are

integrated through the shell thickness to obtain

$$n^{\alpha\beta} = C^{\alpha\beta\gamma\delta}\varepsilon_{\gamma\delta}h_{\text{th}} \,, \tag{49}$$

$$m^{\alpha\beta} = C^{\alpha\beta\gamma\delta}\kappa_{\gamma\delta}\frac{h_{\text{th}}^3}{12} \,. \tag{50}$$

Using the above notation, we specialize the generic structural sub-problem by defining

$$B_2(\mathbf{w},\mathbf{u}) = \int_{\Gamma_t} \mathbf{w}\cdot\rho_2 h_{\text{th}}\left.\frac{\partial\mathbf{u}}{\partial t}\right|_{\mathbf{X}} d\Gamma + \int_{\Gamma_0} (\mathbf{n}:\delta\boldsymbol{\varepsilon} + \mathbf{m}:\delta\boldsymbol{\kappa})\, d\Gamma \,, \tag{51}$$

$$F_2(\mathbf{w}) = \int_{\Gamma_t} \mathbf{w}\cdot\rho_2 h_{\text{th}}\mathbf{f}\, d\Gamma + \int_{\Gamma_t} \mathbf{w}\cdot\mathbf{h}^{\text{net}}\, d\Gamma \,, \tag{52}$$

where $\mathbf{h}^{\text{net}} = \mathbf{h}(\xi_3 = -h_{th}/2) + \mathbf{h}(\xi_3 = +h_{th}/2)$ sums traction contributions from the two sides of the shell. For isotropic materials, the material tensor may be derived from a Young's modulus, E, and Poisson ratio, ν.

4.4.3. Isogeometric shell discretization

We discretize shell structures isogeometrically, using C^1-continuous quadratic B-spline patches to represent both the reference configuration and the approximate displacement solution. The details of this discretization are given in Kiendl et al. [63, 113]. A noteworthy aspect of this discretization is the fact that it requires no rotational degrees of freedom; our C^1-continuous approximation space (for a single patch) is in H^2, so we may directly apply Galerkin's method to the forms defined in Eqs. (51) and (52). It should be noted that for complex structures, the continuity of the geometrical mapping is often reduced to the C^0 level (e.g geometries comprised of multiple patches). The problem of reduced continuity across patch boundaries can be resolved by applying linear constraint equations for simple geometries [63], through the bending strip method [112], or through the blended shell formulation [114].

4.5. Time integration and fluid–structure coupling

We apply the same implicit generalized-α scheme that we use for the fluid sub-problem in Section 3.3 to coupled FSI problems. Given our variational formulation for the coupled problem, it would be possible, in principle, to simultaneously solve for the fluid, structure, and multiplier solutions at each time step, in a monolithic fashion. However, as discussed in Section 4.2.1, we use an iterative scheme for updating the Lagrange multiplier unknowns, wherein an unconstrained problem with a constant multiplier field is solved one or more times within each time step. For the unconstrained problem, we opt to derive separate tangent matrices for the fluid and structure problems, considering the solution variables of each (along with the multipliers) to be constant in the other's linearization. This is not equivalent to a full linearization of the problem, as it discards

some information regarding the coupling of the sub-problems. However, the ability to isolate the structural tangent makes the method more modular, easing the implementation of new material models.

For time steps of the penalty method and the unconstrained problems at each step of the iteration described in Section 4.2.1, we apply what is called, in the terminology of Tezduyar and Sathe [115], a block-iterative approach. This approach alternates between solving for increments of the fluid and structure solutions. Schematically, consider $R_f(u_f, u_s)$ to be the nonlinear residual for the fully-discrete fluid sub-problem at a particular time step, which depends on the discrete fluid and structure solutions, u_f and u_s. Likewise, $R_s(u_f, u_s)$ is the residual for the discrete structure sub-problem. Then the block-iterative procedure to find a root of (R_f, R_s) is to start with guesses for u_f and u_s, then repeat the steps

1. Assemble $R_f(u_f, u_s)$ and a (typically approximate) tangent matrix, $A_f \approx \partial R_f / \partial u_f$.
2. Solve the linear system $A_f \Delta u_f = -R_f$ for the fluid solution increment.
3. Update the fluid solution: $u_f \leftarrow u_f + \Delta u_f$.
4. Assemble $R_s(u_f, u_s)$ and $A_s \approx \partial R_s / \partial u_s$.
5. Solve $A_s \Delta u_s = -R_s$ for the structure solution increment.
6. Update the structure solution: $u_s \leftarrow u_s + \Delta u_s$.

until R_f and R_s are sufficiently converged. Note that this resembles Newton iteration with an inexact tangent, wherein off-diagonal blocks of the tangent matrix for the combined system,

$$\begin{pmatrix} A_f & (\partial R_f / \partial u_s) \\ (\partial R_s / \partial u_f) & A_s \end{pmatrix} \begin{pmatrix} \Delta u_f \\ \Delta u_s \end{pmatrix} = - \begin{pmatrix} R_f \\ R_s \end{pmatrix}, \tag{53}$$

are neglected. However, the update of the fluid solution in step 3 distinguishes block iteration from an inexact tangent method. To ensure predictable running times and avoid stagnation in pathological configurations, we typically select the resolution of the nonlinear algebraic solution by choosing a fixed number of iterations rather than a percentage by which the residual must be reduced. This may be interpreted as a predictor–multi-corrector scheme based on Newton's method [74]. While it is possible that error from isolated, poorly-solved time steps can pollute the future of a solution, we find that, within reasonable limits, quantities of engineering interest are typically more sensitive to spatial and temporal discretizations than nonlinear solution tolerance.

4.6. Flow over an elastic beam

In this section, we test our variational immersed boundary framework for shells on the 2D FSI benchmark problem of flow over an elastic beam. The problem is first proposed by Wall [116] to solve the coupled fluid–structure dynamics of an elastic beam that is fixed at one end to an

immobile block and surrounded by flowing fluid. The problem geometry and boundary conditions are shown in Figure 9. A uniform inflow velocity of 51.3 cm/s is prescribed at the inlet. Lateral boundaries are assigned zero normal velocity and zero tangential traction. A zero traction boundary condition is applied at the outflow. The densities of the beam and the fluid are $\rho_2 = 0.1$ g cm^{-3} and $\rho_1 = 1.18 \times 10^{-3}$ g cm^{-3}, respectively. The fluid viscosity is $\mu = 1.82 \times 10^{-4}$ g cm^{-1}s^{-1}. Taking the box width as a length scale, the Reynolds number is $Re = 332.6$. The beam is composed of an isotropic St. Venant–Kirchhoff material, with Young's modulus $E = 2.5 \times 10^6$ g cm^{-1}s^{-2} and Poisson ratio $\nu = 0.35$. Due to the small ratio of the beam's thickness to its length, we model it using the Kirchhoff–Love shell formulation defined by Eqs. (51) and (52), in contrast to other authors that model the beam using solid elements [74, 116–118]. The problem setup is completely symmetrical, but the inherent instability of unsteady fluid flow amplifies small numerical errors, causing a transition to unsteady but periodic motion, in which the beam oscillates up and down with some frequency and amplitude. We take the frequency and amplitude of this oscillation as quantities of interest, providing a basis for quantitative comparison with results from other studies.

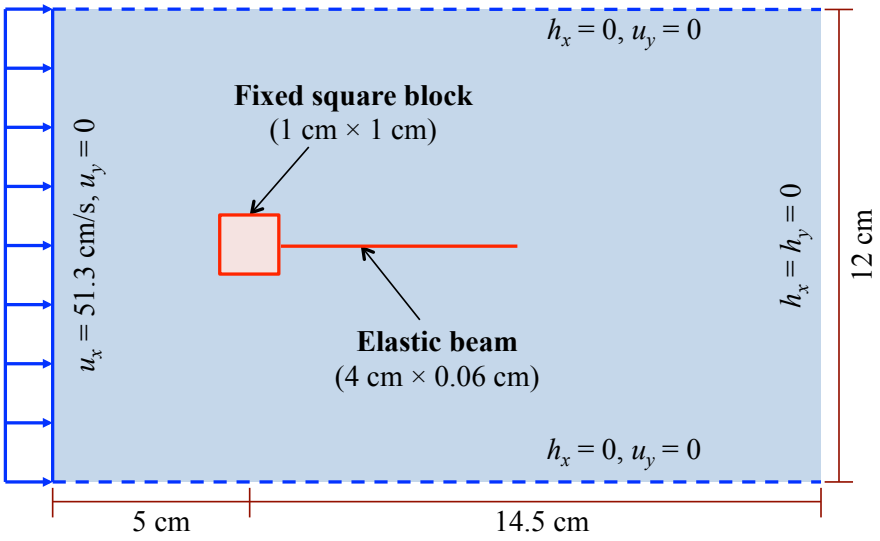


Figure 9: Problem setup for flow over an elastic beam. Not to scale.

We use Nitsche's method for immersed boundaries with adaptive quadrature to apply boundary conditions to the perimeter of the box, and the immersed shell formulation to compute interactions between the beam and fluid. Note that the adaptive quadrature is only applied to the fluid elements that are cut by the box, where the functions are discontinuous across the object interface. The adaptive quadrature is not necessary for the fluid elements intersecting the shell, because the functions are still continuous and the integrand is smooth.

To control for potential differences from other studies, due to either the use of shell theory or the VMS fluid formulation, we also compute a body-fitted reference case, using the same VMS

and Kirchhoff–Love shell formulations as the immersed-boundary computation.

4.6.1. Immersed discretizations

To study the properties of our FSI framework under refinement, we construct three meshes, M1, M2, and M3, of the fluid domain into which the box and attached elastic beam are immersed. M1 contains 7144 quadratic B-spline elements, with refinement focused near the box and beam, as shown in Figure 10. The beam mesh consists of 32 quadratic B-spline shell elements. Another surface consisting of 36 elements is used to generate the surface quadrature rule for the box. M2 is a uniform h-refinement of M1, and M3 is a uniform refinement of M2. The box surface and beam discretizations used in conjunction with M2 and M3 are likewise refined.

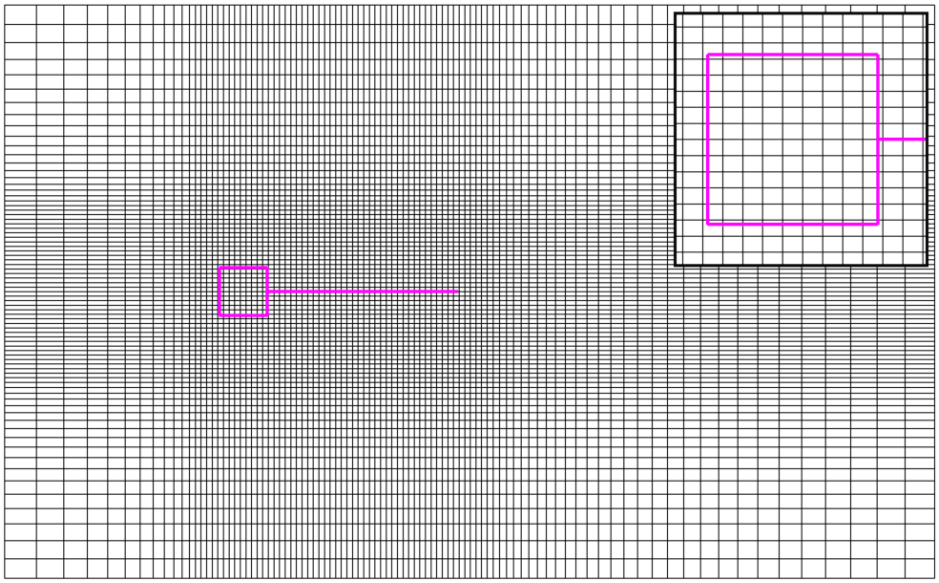


Figure 10: Mesh M1 for flow over an elastic beam, with the locations of the immersed box and beam highlighted.

For all three meshes, we use $l = 3$ levels of adaptive quadrature to resolve the boundary of the immersed box. The pressure differences between sides of the beam are relatively small and a pure penalty method appears to enforce the no-penetration constraint sufficiently. We therefore use only the penalty method to couple the beam and fluid, with $\tau^B_{(\cdot)} = 10^3$ g cm^{-2}s^{-1} in all cases.

Remark 11. In our computations, the structural surface elements also define the surface quadrature rule used to compute integrals over Γ_{I} in Eq. (12), as discussed in Section 3.2.1. However, the surface discretizations and quadrature rules used for shell mechanics and FSI integrals over Γ_{I} do not, in general, need to coincide. One may wish to use different discretizations if the physical aspects of the fluid and structural problems are vastly different.

4.6.2. Body-fitted reference discretizations

As mentioned earlier, the use of shell elements in this benchmark problem is atypical, with most authors directly modeling the beam as a solid structure. We would like to compare the results of our variational immersed boundary method with those of other investigations, but we must be able to distinguish whether differences are in fact due the immersed boundary FSI technique, or simply the use of shell theory. We compute a reference case, then, with a deforming fluid mesh fitted directly to the box and a shell representation of the beam. In this case, we strongly enforce the kinematic constraint at the fluid–structure interface.

We use two body-fitted meshes, one a refinement of the other, to ensure that the reference solution is properly converged. The first mesh, BM1, shown in Figure 11, contains 6800 quadratic B-spline elements. The second mesh, BM2, is a uniform h-refinement of BM1.

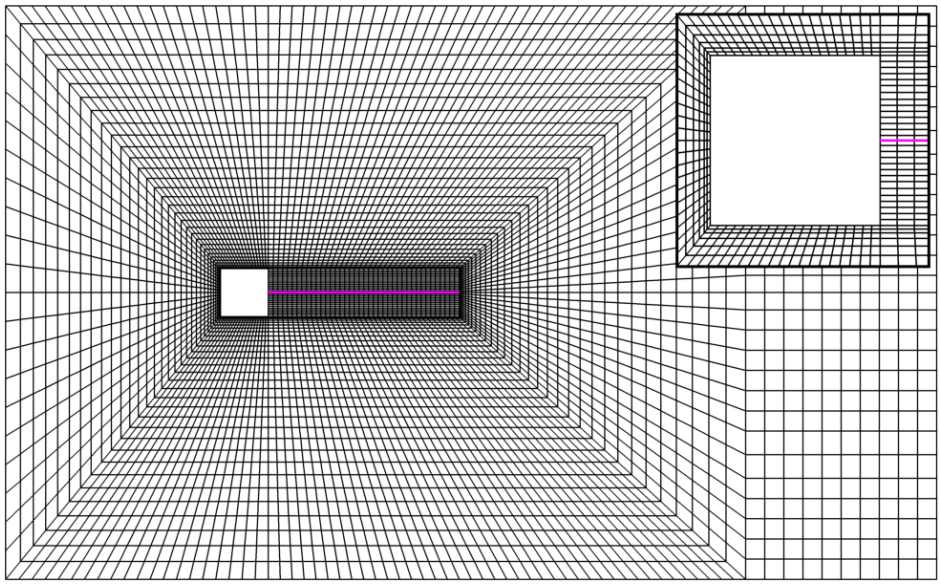
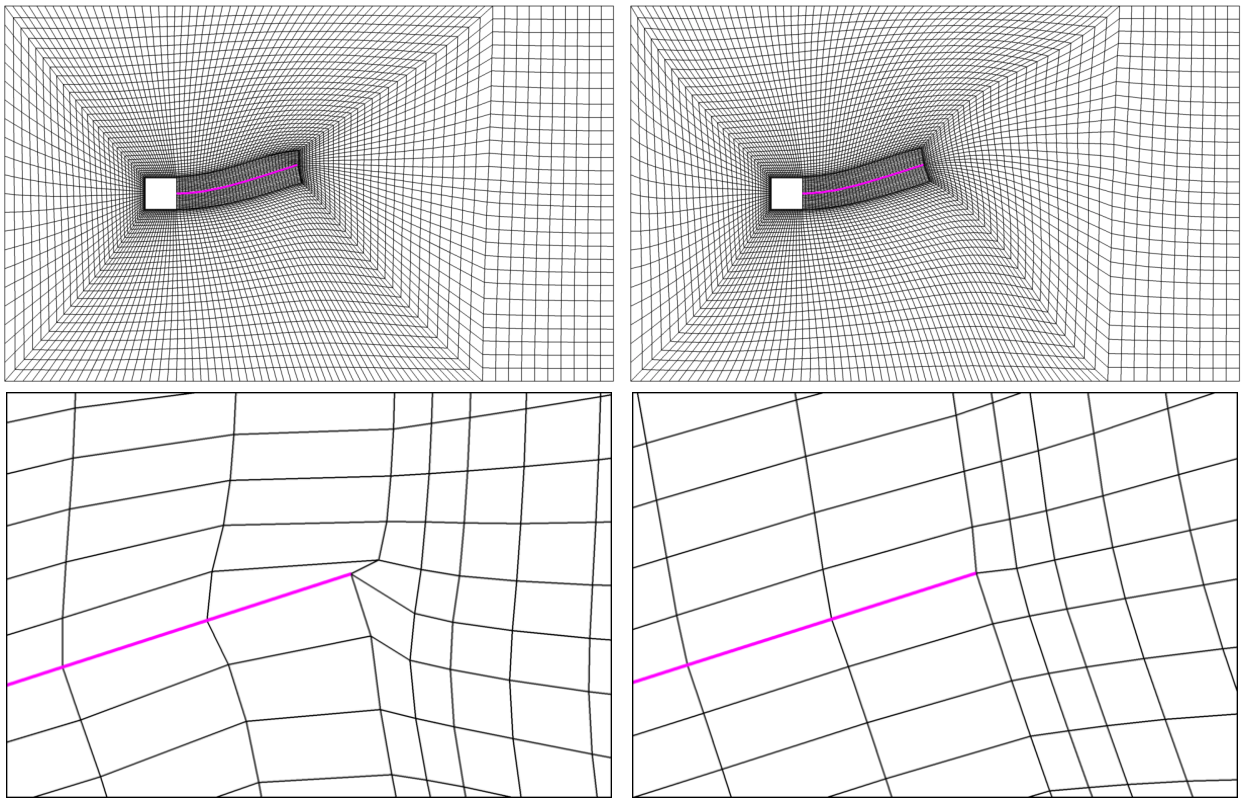


Figure 11: Mesh BM1 for the body-fitted reference computation of flow over an elastic beam.

The matching interface between the beam and surrounding fluid mesh allows the no-slip condition on the surface of the beam to be enforced strongly. The mesh deforms according to the solution of a fictitious elastostatic problem that takes the location of the beam as a displacement boundary condition. The velocity of this deformation enters into the fluid formulation (13) as $\hat{\mathbf{u}}^h$. This velocity is derived from displacements of the mesh in consecutive time steps. These displacements are solutions to a fictitious linear elastostatic problem with boundary conditions imposed by the structural displacement. Mesh quality is preserved throughout this deformation by stiffening the fictitious material in response to compression: the material tensor is modified such that the mesh Young's modulus, E_{mesh}, scales inversely with the square of the Jacobian determinant, J_ξ, of the mesh's parametric mapping in the previous time step. A more detailed discussion of

this technology can be found in [23, 74, 119–121]. The importance of this Jacobian stiffening is demonstrated by the mesh deformations shown in Figure 12.



(a) $E_{\text{mesh}} \propto J_{\xi}^{-1}$ (b) $E_{\text{mesh}} \propto J_{\xi}^{-2}$

Figure 12: Sample mesh deformations, illustrating the effects of Jacobian stiffening on the solution of the fictitious linear elastic problem. The beam is highlighted in magenta and the mesh solution near the tip is magnified.

4.6.3. Comparison of results

To compare results of different computations we record the y-displacement of the beam's tip as a function of time. Specifically, we focus on the frequency at which this displacement oscillates after reaching a steady periodic motion.

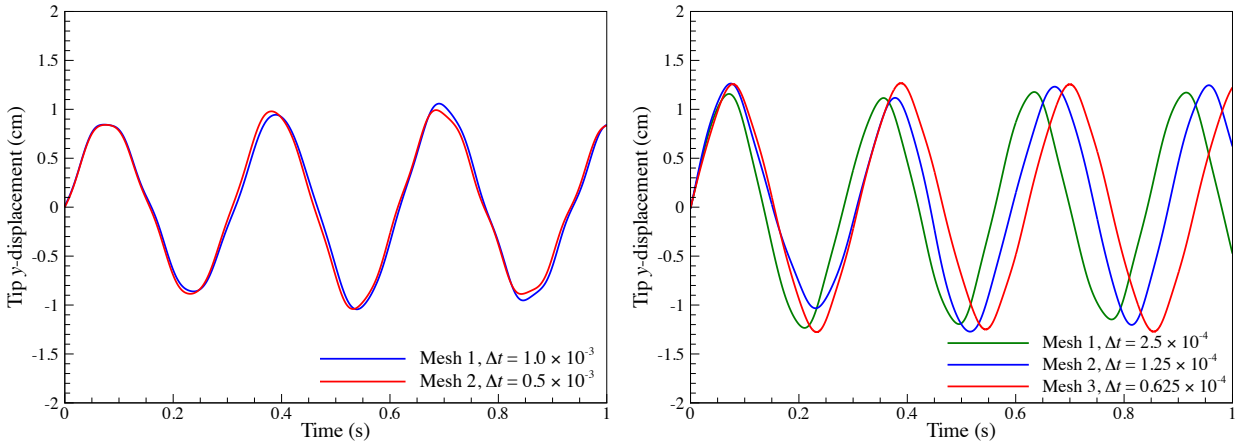
For this problem, the pressure gradients driving the beam are relatively mild and we directly apply the VMS stabilization of Eq. (16), with none of the modifications discussed in Section 4.3. Table 6 collects the frequencies of oscillation from computations at various time steps for the body-fitted reference while Table 7 shows frequencies for the immersed-boundary computations. The amplitudes and waveforms of the beam tip y-displacement are similar for all cases; Figures 13(a) and (b) show these waveforms for the body-fitted and immersed-boundary computations, respectively. We see that the two methods converge to periodic solutions of the same frequency. The convergence of the variational immersed boundary method, however, is considerably slower with respect to time step.

	BM1	BM2
$\Delta t = 1.0 \times 10^{-3}$	3.2 Hz	3.2 Hz
$\Delta t = 0.5 \times 10^{-3}$	3.2 Hz	3.2 Hz

Table 6: Frequencies of oscillation for the y-displacement of the beam tip, computed using body-fitted reference meshes.

	M1	M2	M3
$\Delta t = 2.5 \times 10^{-4}$	3.6 Hz	3.5 Hz	3.6 Hz
$\Delta t = 1.25 \times 10^{-4}$	3.5 Hz	3.4 Hz	3.4 Hz
$\Delta t = 0.625 \times 10^{-4}$	–	3.4 Hz	3.2 Hz

Table 7: Frequencies of oscillation for the y-displacement of the beam tip, computed using immersed boundary FSI.



(a) (b)

Figure 13: Tip y-displacements over time for computations on (a) body-fitted meshes BM1 and BM2, using $\Delta t = 1.0 \times 10^{-3}$ s and $\Delta t = 0.5 \times 10^{-3}$ s, respectively, and (b) on immersed meshes M1, M2, and M3, using $\Delta t = 2.5 \times 10^{-4}$ s, $\Delta t = 1.25 \times 10^{-4}$ s, and $\Delta t = 0.625 \times 10^{-4}$ s, respectively.

Figure 14 shows a snapshot of the fluid pressure and beam deformation on M1, when the beam tip is at its largest displacement. This illustrates the mechanism of low pressure vortices periodically shedding from the box and driving the oscillatory motion of the beam. The pressure solution is represented on a stationary background mesh, using continuous basis functions. In the exact solution, we expect pressure to be discontinuous along the beam. The snapshot illustrates how the computed solution approximates these discontinuities with sharp gradients.

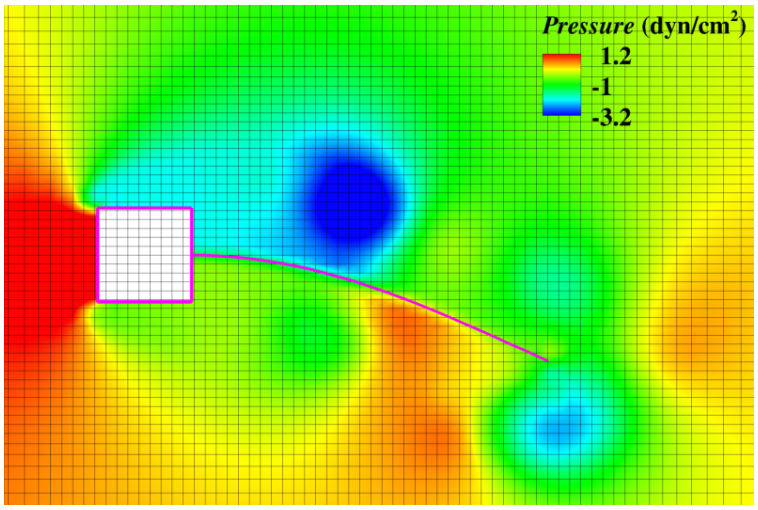


Figure 14: A snapshot of the pressure field about the immersed beam, computed on M1. The elements of the fluid mesh remain in the rectilinear configuration shown as the beam moves. The pressure discontinuity at the beam is approximated by a steep gradient.

5. Application to a bioprosthetic heart valve

In this section, we use the thin shell immersed boundary FSI method developed in Section 4 to simulate an aortic bioprosthetic heart valve (BHV) and the surrounding blood flow during a cardiac cycle. The aortic valve regulates flow between the left ventricle of the heart and the ascending aorta. Figure 15 provides a schematic depiction of its position in relation the surrounding anatomy. As mentioned in Section 4.4, the weak form of Kirchhoff–Love shell theory requires the shell geometry to be C^1-continuous. We first describe our strategy of mapping a given valve leaflet geometry to a quadratic B-spline patch. We then address the issue of contact between leaflets. A benefit of using an immersed boundary FSI method is that the contact formulation can be added to the structure sub-problem without needing to consider the fluid. We develop a penalty-based dynamic contact algorithm in Section 5.2 and test it in Section 5.3 to show that this method is sufficient for our purposes. Finally, we proceed to compute FSI for the BHV in Section 5.4.

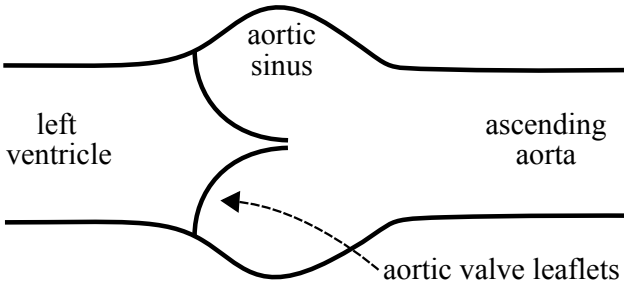


Figure 15: A schematic drawing illustrating the position of the aortic valve relative to the left ventricle of the heart and the ascending aorta.

5.1. Valve model

We model the geometry of the prosthetic valve using three quadratic B-spline patches—one for each leaflet. The spline surface for a single leaflet is based on a 23-mm BHV design[4] by Edwards Lifesciences, supplied in the form of a quadrilateral mesh. The spline surface, parameterized as a square in the knot space with $(u, v) \in [0, 1] \times [0, 1]$, is specialized for aortic valve leaflets by degenerating the two edges of spline space ($u = 0$ and $u = 1$) to the two commissure points, as illustrated in Figure 16. We fit the physical space of the B-spline patch to the quadrilateral mesh surface in two stages. To avoid oscillations at the edges, we first fit a piecewise C^1-continuous spline curve to the edges, with C^0 points (repeated knots) at the commissure points. We then fit the interior physical space of the B-spline patch to the interior of the leaflet surface, holding the boundary control points fixed.

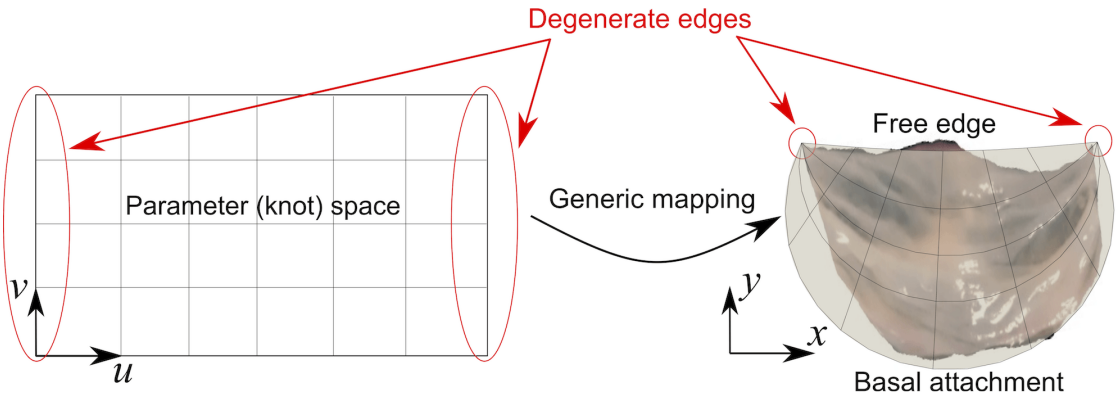


Figure 16: Generic mapping for an aortic valve leaflet using a B-spline patch, where two edges in the parametric space are degenerated to commissure points.

The fitting of both the edges and the interior is performed by minimizing the ℓ^2-norm of the Euclidean distances between the vertices of the given quadrilateral mesh, $\{x_i\}$, and their projections

[4]This type of pericardial BHV is fabricated from bovine pericardium sheets that are chemically fixed after being die-cut and mounted onto a metal frame to form the leaflets. As a result, the given geometries are without internal stress and can be used directly as stress-free configurations.

onto the spline curve or surface, $\{x_i^p\}$. The control points of the fitted spline, $\{C_j\}$, are therefore the solution of

$$\min \sum_i \left| x_i - x_i^p \right|^2 \equiv \min \sum_i \left| x_i - \sum_j N_j(\xi_i^p) C_j \right|^2 \tag{54}$$

where N_j is the basis function associated with the j-th control point and ξ_i^p (in $\mathbb{R}$ for curves and $\mathbb{R}^2$ for surfaces) is the parametric location corresponding to the projected point x_i^p. The parameters of projected points are determined from the condition that the difference between an input point, x_i, and its projection, x_i^p, should be normal to the curve or surface being fit. Thus ξ_i^p is the solution of the nonlinear system $(x_i - x_i^p) \cdot \frac{\partial x_i^p}{\partial \xi_i^p} = 0$, which can be found by Newton's iteration. To solve the minimization problem of Eq. (54), we start with an initial guess of $\{C_j\}$, then repeat the steps

1. Compute projected points, $\{x_i^p\}$, and their parameters, $\{\xi_i^p\}$, with the control points, $\{C_j\}$, fixed.
2. Solve the linear least-squares problem for $\{C_j\}$ that is implied by holding $\{\xi_i\}$ fixed in Eq. (54),

until a norm of the change in control point values from one iteration to the next is smaller than some tolerance. The control mesh and the physical images of knot spans of the resulting BHV mesh, prior to any refinement (knot insertions) for analysis purposes, are shown in Figure 17. The refined mesh, which is comprised of 1404 quadratic B-spline elements, is shown in Figure 18.

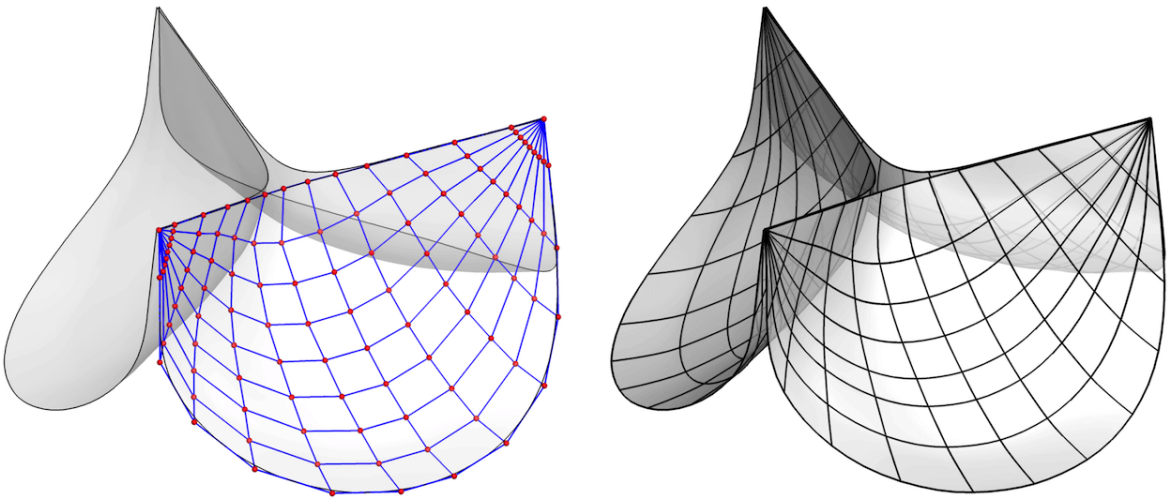


Figure 17: Left: Control mesh. Right: The physical images of knot spans in the BHV mesh, prior to analysis refinements.

Remark 12. This method of fitting a B-spline patch to the leaflet can be used for patient-specific valve geometries from in-vivo imaging. The degeneration of two edges to the commissure points

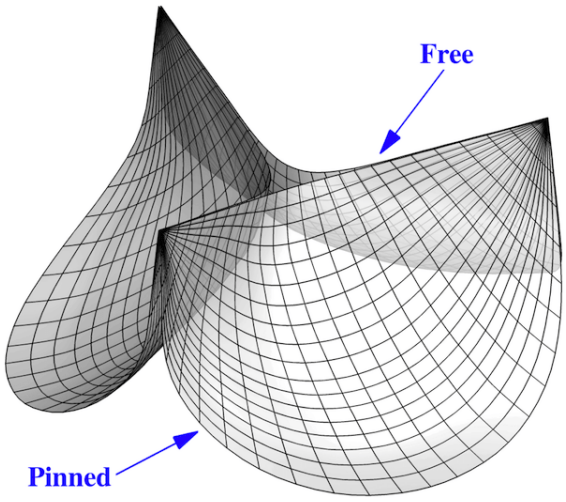


Figure 18: Refined B-spline mesh for analysis purposes. It is comprised of 1404 quadratic elements. The pinned boundary condition is applied to the leaflet attachment edge.

provides a physical connection that can be used to map the collagen architecture either in a patient-specific way or in an average sense. More details on mapping the collagen architecture and calculating its average using this method can be found in Aggarwal et al. [122].

Remark 13. The use of small, degenerated elements is not intuitively appealing and indeed appears to inhibit convergence of the nonlinear structure sub-problem (11) to machine precision, with our relatively straightforward implementations of Galerkin's method and Newton's iteration. In practice, however, we can reduce the residual sufficiently to obtain meaningful simulation results.

5.2. Contact algorithm

Contact between leaflets is an essential feature of a functioning heart valve. We find that it occurs during both the opening and closing phases. While the kinematic constraint of continuous velocity through the fluid and structure should technically obviate any special treatment of structural contact, weak enforcement of the fluid–structure kinematic constraint allows some structural interpenetration and we find that additional enforcement of structural non-penetration improves the quality of solutions. In this section, we describe the penalty method that we use to model contact and address its physical plausibility. The penalty method has been widely used to handle contact problems [42–44, 64, 123] because of its conceptual simplicity and because it is straightforward to implement.

To handle the contact between leaflets using the penalty method, we wish to penalize the penetration of the leaflets. Because the leaflets are modeled as shell structures, it is not immediately clear how penetration should be defined; a shell has no interior in which to detect penetrating geometry. However, an aortic valve leaflet, operating under normal anatomical conditions, will contact other leaflets on only one side, motivating the following definition of penetration.

Consider leaflets S_1 and S_2 to be smooth parametric surfaces in $\mathbb{R}^3$. For $\mathbf{x}_1 \in S_1$, with surface normal $\mathbf{n}_1$ determining the side on which contact will occur, we say that $\mathbf{x}_1$ contacts leaflet S_2 if the following conditions are met:

1. There exists a point $\mathbf{x}_2 \in S_2$ with normal $\mathbf{n}_2$ such that $(\mathbf{x}_1 - \mathbf{x}_2)$ is perpendicular to S_2. We call $\mathbf{x}_2$ the closest point on S_2 to $\mathbf{x}_1$, but, without additional assumptions on S_2, the defining conditions guarantee neither that $\mathbf{x}_2$ is unique nor that it minimizes $|\mathbf{x}_1 - \mathbf{x}_2|$. In practice, we determine $\mathbf{x}_2$ by iteratively solving the nonlinear problem of finding $\xi = (\xi_1, \xi_2)$ in the parameter space for S_2 such that

$$\begin{cases} (\mathbf{x}_1 - \mathbf{x}_2(\xi)) \cdot \dfrac{\partial \mathbf{x}_2(\xi)}{\partial \xi_1} &= 0 \\ \\ (\mathbf{x}_1 - \mathbf{x}_2(\xi)) \cdot \dfrac{\partial \mathbf{x}_2(\xi)}{\partial \xi_2} &= 0 \end{cases} . \tag{55}$$

2. $|\mathbf{x}_1 - \mathbf{x}_2| < c$, were $c > 0$ is a parameter chosen to avoid false positive contact of distant geometry. We assume that penalties will be strong enough to prevent penetrations larger than c.
3. $\mathbf{n}_1 \cdot \mathbf{n}_2 < \alpha$, for some $-1 \leq \alpha \leq 0$. Choosing $\alpha < 0$ allows a hinge-like boundary between S_1 and S_2 that can open through angles larger than 270° without immediately incurring a contact penalty.

For a contacting point $\mathbf{x}_1$, its signed penetration is defined as $d = (\mathbf{x}_2 - \mathbf{x}_1) \cdot \mathbf{n}_2$. We consider $\mathbf{x}_1$ to penetrate S_2 if $d > -h$, where $c > h \geq 0$ indicates a minimum desired distance between the contacting sides of S_1 and S_2. This notation is illustrated for a pair of contacting points in Figure 19.

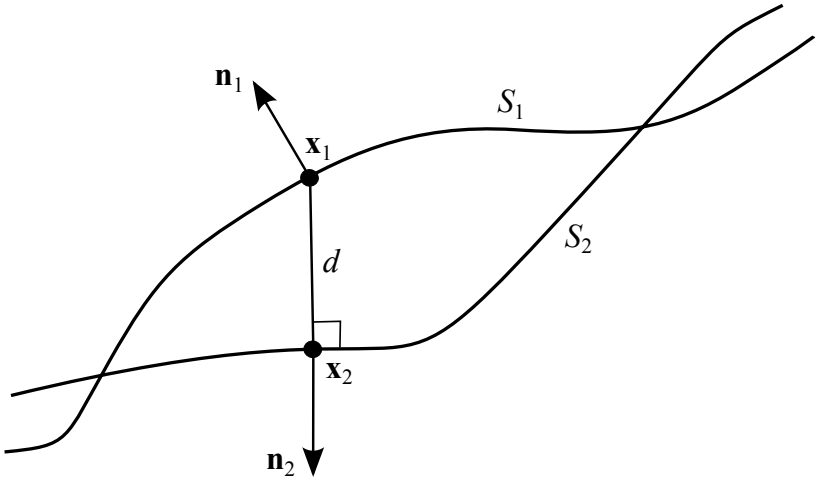


Figure 19: Illustration of contact notation.

Non-penetration is enforced weakly, by penalizing $d > -h$. To motivate our contact algorithm, consider adding the following term

$$\int_{S_1} \left(\left((\mathbf{w}_2)^2 - (\mathbf{w}_2)^1 \right) \cdot \mathbf{n}_2 \right) (kd^+) \; d\Gamma \tag{56}$$

to the left-hand side of Eq. (11). This term tests a penetration residual against a difference of weighting functions, $(\mathbf{w}_2)^1$ and $(\mathbf{w}_2)^2$, where $(\mathbf{w}_2)^i$ is the structure weighting function on surface i. The term is not a rigorous formulation because the change-of-variables to integrate $(\mathbf{w}_2)^2$ over S_1 is not precisely defined and the definition of d is ambiguous. With some regularity assumptions on S_1 and S_2, and c sufficiently small, we could treat the leaflets as manifolds and use the tubular neighborhood theorem of differential geometry to assert the existence of a well-behaved mapping between contacting regions, but we do not have a constructive estimate for the bound on c, and prefer to disambiguate our formulation in an *ad hoc* manner, by simply detailing our discrete implementation below.

We test for penetration and apply penalty forces at a discrete set of contact points, $\{\mathbf{x}_1^1, \ldots, \mathbf{x}_1^n\} \subset S_1$. For the subset $\{\mathbf{x}_1^{j_k}\}$ contacting $\{\mathbf{x}_2^{j_k}\} \subset S_2$, we apply opposing forces on S_2, conserving linear momentum. To conserve angular momentum, the contact forces between $\mathbf{x}_1$ and $\mathbf{x}_2$ are along their separation $\mathbf{x}_1 - \mathbf{x}_2$, which is, by construction, parallel to $\mathbf{n}_2$. The force on $\mathbf{x}_1$ is $\mathbf{f}_1 = -w(P_k(d))\mathbf{n}_2$ and the force on $\mathbf{x}_2$ is $\mathbf{f}_2 = -\mathbf{f}_1$, where w is a weight associated to $\mathbf{x}_1$ and $P_k(d)$ penalizes penetration. For our computations, we use the penalty function

$$P_k(d) = \begin{cases} \frac{k}{2h}(d+h)^2 & , \quad d \in (-h, 0) \\ kh/2 + kd & , \quad d \geq 0 \\ 0 & , \quad \text{otherwise} \end{cases} , \tag{57}$$

where k decides the strength of the position penalty. The behavior of P_k on the interval $-h < d < 0$, illustrated in Figure 20, ensures that the penalty activates smoothly as contact begins, helping us to resolve the nonlinearity through Newton's iteration. Motivated by Eq. (56), we choose $\{\mathbf{x}_1^j\}$ to be Gaussian integration points on elements of S_1 and weight forces using the corresponding integration rule. In general, we expect the contact parameters to scale like

$$k = c_1 E / \Delta x \tag{58}$$

$$h = c_2 \Delta x \tag{59}$$

where Δx is a measure of the structural element size. In this paper, however, we focus on a single application and use values determined effective through numerical experiments.

The above method does not preserve geometrical symmetries. To see this, consider contacting

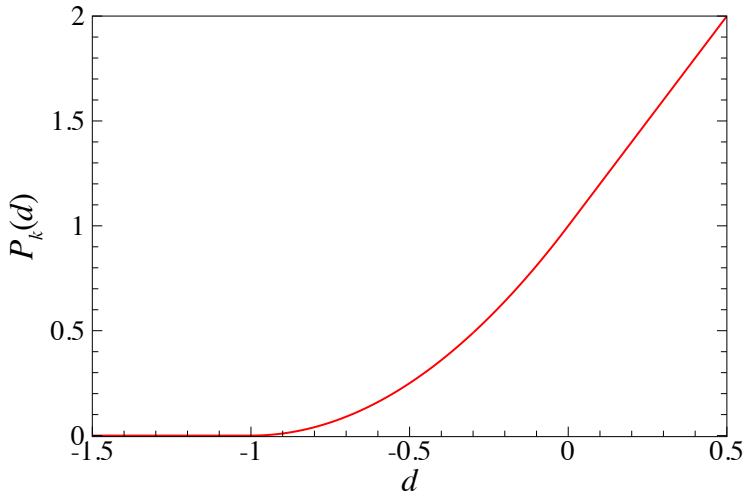


Figure 20: The function $P_k(d)$ for $k = 2$ and $h = 1$.

planes at an angle; the directions of contact forces depend on the choice of S_1 and S_2, as shown in Figure 21. To ensure that results are independent of this arbitrary distinction, we compute forces with both choices and sum the results. To prevent the introduction of contradictory constraints by this double application of our algorithm, we continuously re-evaluate the contact points $\{\mathbf{x}_2^{j_k}\}$ over time and throughout the nonlinear iteration[5] at each time step.

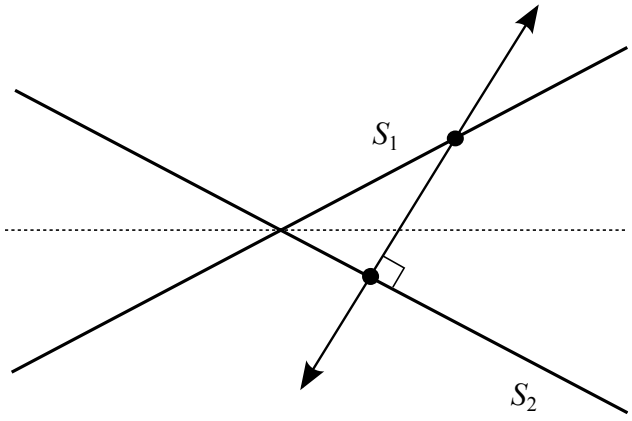


Figure 21: Symmetrical geometry results in asymmetrical contact forces.

5.3. Dynamic simulation of a heart valve, with prescribed pressure loading

To test the suitability of our contact algorithm for the simulation of an aortic valve, we apply a physiological transvalvular pressure load in a dynamic simulation of a BHV. This eliminates the

[5]Our linearization does not account for nonlinearity arising from the dependence of the parameters of the closest point on the displacement solution, but the resulting inexact tangent appears practically effective in spite of this omission.

complexity associated with FSI while exercising the contact method at appropriate velocities and pressures. Further, we can expect to produce symmetrical results in this simplified setting, while the same cannot be assumed of FSI calculations [46]. Our testing loosely emulated the dynamic simulation by Kim et al. [5], but, due to differences in geometry and material parameters, we do not expect to precisely reproduce the results of the cited study.

5.3.1. Description of the problem

This simulation uses the valve geometry discussed in Section 5.1 and an isotropic St. Venant–Kirchhoff material with $E = 10^7$ dyn/cm^2 and $\nu = 0.45$. The thickness of the leaflets is 0.0386 cm. We use the contact algorithm discussed in Section 5.2, setting the parameters to $k = 10^8$ dyn/cm^3, $h = 0.005$ cm, $\alpha = 0.7$, and $c = 0.1$ cm. The time-step size used in the dynamic simulation is 0.0001 s and the pinned boundary condition is applied to the leaflet attachment edge as shown in Figure 18.

In accordance with the expected contact pattern and the convention established in Section 5.2, the surface normal, $\mathbf{n}$, points from the aortic to the ventricular side of each leaflet. We model the transvalvular pressure (i.e. pressure difference between left ventricle and aorta) with the traction $-P(t)\mathbf{n}$, where $P(t)$ is the pressure difference at time t, taken from the profile used by Kim et al. [5] and reproduced in Figure 22. The duration of a single cardiac cycle is 0.76 s.

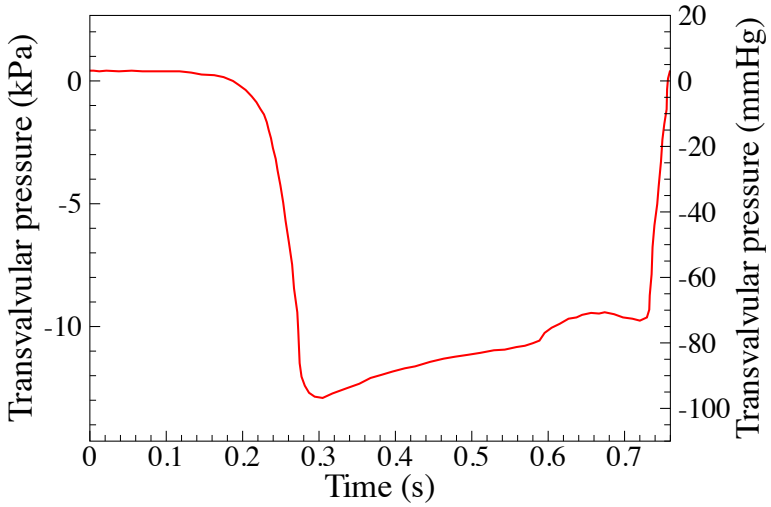


Figure 22: Transvalvular pressure applied to the leaflets as a function of time. The duration of a single cardiac cycle is 0.76 s.

As in the computations of Kim et al. [5], we use damping to model the viscous and inertial resistance of the surrounding fluid. We apply a traction of $-C\mathbf{v}$, where $\mathbf{v}$ is the leaflet velocity and $C = 80$ (dyn s)/cm^3. This value of C is selected to ensure that the valve opens at a physiologically reasonable time scale when the given pressure is applied.

5.3.2. Results and discussion

The deformation and strain distribution of the leaflets at several points in the cardiac cycle (after reaching a periodic solution) is illustrated in Figure 23. The opening begins in a manner that is qualitatively similar to the results computed by Kim et al. [5], but the fully-open state differs, in that the belly regions of the leaflets do not snap through to become concave toward the ventricular side. We find that this snap-through behavior (with our choice of constitutive model) is sensitive to the level of damping and slight variations in the leaflet geometry. The purpose of the present computation, however, is largely to test the robustness of the contact algorithm in the impacting and closed states, so we do not dwell on the details of the fully-open configuration. The pressurized diastolic state exhibits much greater sagging of the belly region; this is presumably because our simplified material neglects the stiffening of true tissue under strain. The important conclusion for our contact algorithm is that the results do not show noticeable penetrations under physiological pressure levels and there are no spurious asymmetries. Note that in our computation, no symmetry planes are assumed between the leaflets. The symmetric pattern is obtained as a result of the symmetric implementation of the contact algorithm described at the end of Section 5.2. We may therefore proceed to FSI simulation with the same contact parameters and conclusively attribute any asymmetries in the FSI results to the effects of the fluid.

5.4. FSI simulation

In this section, we immerse the BHV model of Section 5.1 into a pressure-driven incompressible flow through a rigid channel. The fluid properties are the same as those used in the blocked channel model problem of Figure 7: $\rho_1 = 1.0$ g cm^{-3} and $\mu = 3.0 \times 10^{-2}$ g cm^{-1}s^{-1}. These parameters model the physical properties of human blood. The valve leaflets have material properties $E = 10^7$ dyn/cm^2 and $\nu = 0.45$.

5.4.1. Parameters of the numerical scheme

As mentioned in Section 4.3, we find that time-step sizes significantly smaller than 10^{-4} s are impractical for computing valve FSI through multiple entire cycles. In this study, we compare the results of using $\Delta t = 1.0 \times 10^{-4}$ s and $\Delta t = 0.5 \times 10^{-4}$ s. To compute reasonable solutions at practical time steps, we need to modify s^{shell}. Taking Table 5 as a guide for the effects of s^{shell} on volumetric flow through a closed valve, we choose $s^{\text{shell}} = 10^6$. We find that results are relatively insensitive to the tangential FSI penalty, τ^B_{TAN}, but conditioning and nonlinear convergence improve with lower values. For the heart valve, we use a value of $\tau^B_{\text{TAN}} = 2.0 \times 10^2$ g cm^{-2}s^{-1}. The no-penetration boundary condition is more critical to the valve's behavior, and, in the computations that follow, we use the higher value of $\tau^B_{\text{NOR}} = 2.0 \times 10^3$ g cm^{-2}s^{-1}. This penalty alone is not large enough to resist flow, but larger penalties prevent the block-iterative fluid–structure coupling from converging. We must therefore include some approximation of the Lagrange multiplier

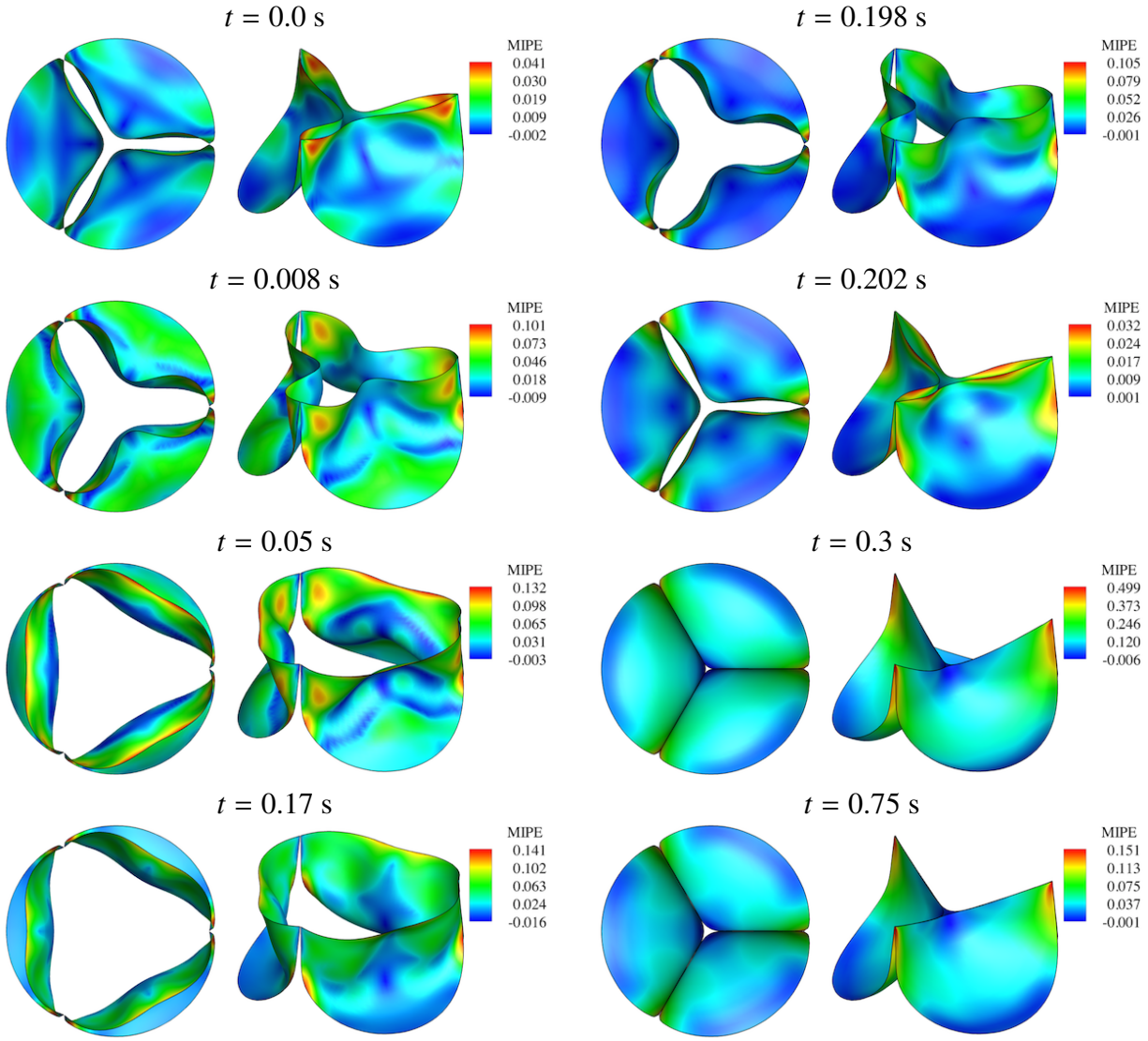


Figure 23: Deformations of the valve from a cycle of the dynamic computation, colored by maximum in-plane principal Green-Lagrange strain (MIPE, the largest eigenvalue of $\mathbf{E}$), evaluated on the aortic side of the leaflet. Note the different scale for each time. Time is synchronized with Figure 22. The initial condition at $t = 0$ s comes from the preceding cycle and is not the stress-free configuration.

field. With the complex time-dependent geometry of the immersed leaflets, the iterative approximation of Lagrange multipliers discussed in Section 4.2.1 does not converge. We therefore opt to use the single-iteration approximation of multipliers. Section 4.2.1 discusses this approximation and compares it to the method of artificial compressibility for incompressible flows and also to a penalization of the displacement difference between the fluid and structure.

5.4.2. Channel geometry

The channel geometry, shown in Figure 24, is a circular tube of diameter 2.3 cm and length 16 cm, with a three-lobed dilation near the valve to model the aortic sinus. It is comprised of quadratic NURBS patches, allowing us to exactly represent the circular portions. We use a multi-patch design to avoid including a singularity at the center of the cylindrical sections. Cross-sections of this multi-patch design are shown in Figure 25. The mesh contains a total of 57600 quadratic NURBS elements. Refinement is focused near the valve and sinus, as shown in Figure 24. The mesh is also clustered towards the wall to better capture boundary-layer phenomena. The modeling of the sinus, magnified in Figure 26, does not include the flexible wall in the human aorta, but the experiments of Bellhouse and Bellhouse [124] determined that the presence of such a channel dilation near the valve plays an important role in the valve's dynamics.

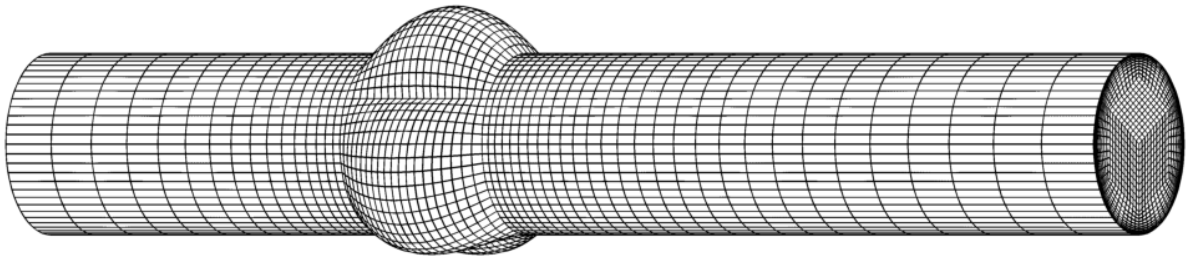


Figure 24: A view of the fluid domain into which the valve is immersed.

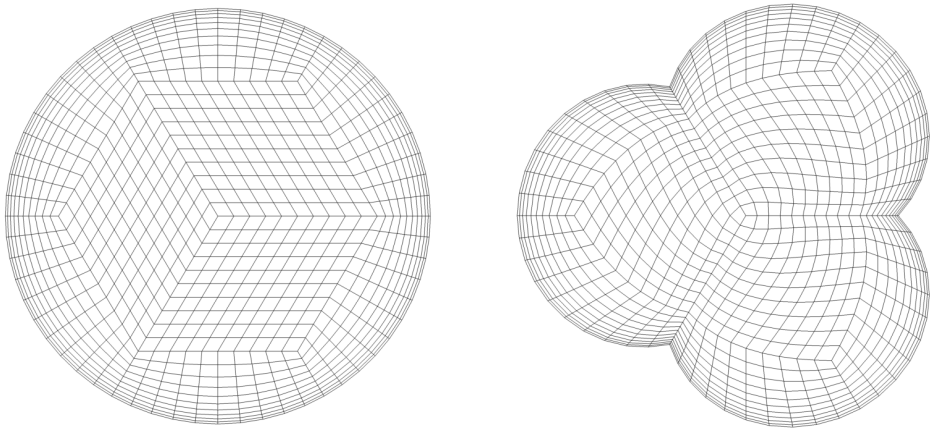


Figure 25: Cross-sections of the fluid mesh, taken from the cylindrical portion and from the sinus.

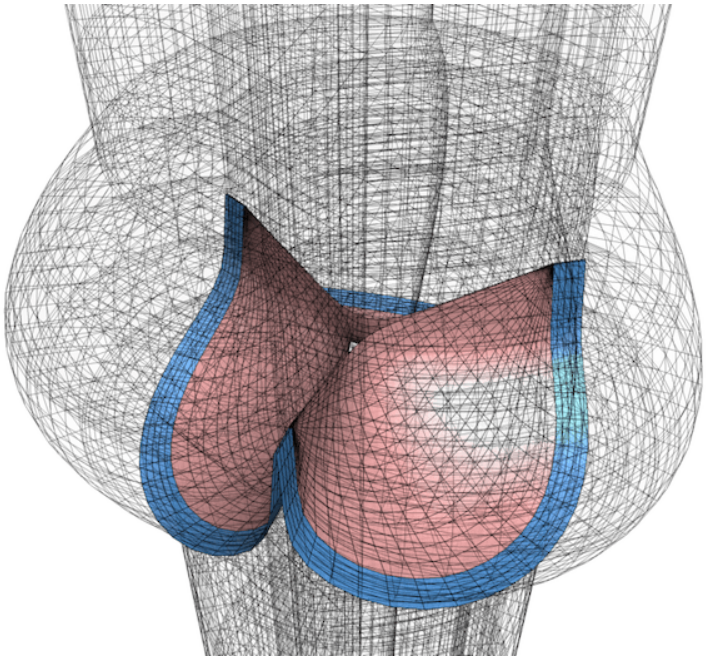


Figure 26: The sinus, magnified and shown in relation to the valve leaflets (pink) and rigid stent (blue).

Remark 14. As in the elastic beam computations of Section 4.6, We use comparable spatial resolutions for the fluid and structure meshes. The shell structure elements are used to define the surface quadrature rule for fluid–structure interface integrals in Eq. (12).

5.4.3. Boundary and initial conditions

The nominal outflow boundary is 11 cm downstream of the valve, located at the right end of the channel, based on the orientation of Figure 24. The nominal inflow is located 5 cm upstream at the left end of the channel. The designations of inflow and outflow are based on the prevailing flow direction during systole, where the valve is open and the majority of flow occurs. In general, fluid may move in both directions and there is typically some regurgitation during diastole. An idealized left ventricular pressure profile, shown in Figure 27, is applied as a traction boundary condition at the inflow. The duration of a single cardiac cycle used in the FSI computation is 0.86 s. The traction $-(p_0 + RQ)\mathbf{n}_1$ is applied at the outflow, where p_0 is a constant physiological pressure level, Q is the volumetric flow rate through the outflow (with the convention that $Q > 0$ indicates flow leaving the domain), $R > 0$ is a resistance constant, and $\mathbf{n}_1$ is the outward facing normal of the fluid domain. This resistance boundary condition and its implementation are discussed in Bazilevs et al. [84]. In the present computation, we use $p_0 = 80$ mmHg and $R = 70$ (dyn s)/cm^5. These values ensure a realistic transvalvular pressure difference of 80 mmHg in the diastolic steady state (where Q is nearly zero) while permitting a reasonable flow rate during systole. Such boundary conditions are sufficient to demonstrate the robustness of our thin shell FSI and contact methodologies under the

range of relevant flow regimes, but the resistance outflow boundary condition is relatively crude, neglecting several important physical phenomena. Section 5.4.4 points out how this simplified outflow boundary condition affects our solution. For a discussion of more realistic cardiovascular outflow boundary conditions, see Vignon-Clementel et al. [125]. At both inflow and outflow, we apply the backflow stabilization discussed in Section 3.1.1, with $\gamma = 0.5$. On the walls of the channel, we strongly enforce the Dirichlet condition $\mathbf{u}_1 = \mathbf{0}$.

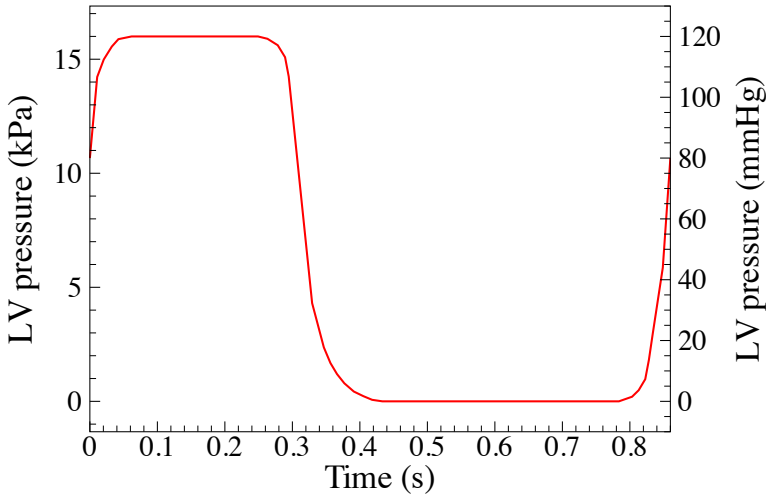


Figure 27: The idealized left ventricular (LV) pressure profile applied to the nominal inflow of the fluid domain. The duration of a single cardiac cycle is 0.86 s.

The left ventricular pressure profile of Figure 27 deliberately coincides with p_0 at $t = 0$. In this way, we may begin from an initial condition of $\mathbf{u}_1 = \mathbf{0}$, $\mathbf{u}_2 = \mathbf{0}$, and $\lambda_n = 0$: a stationary, stress-free state. While the fluid–structure interface multiplier, λ_n, is independent of previous history in the continuous formulation (27), our use of the previous time step's fluid–structure traction as an initial (and, in the single-iteration scheme, only) guess for the multiplier introduces a history dependence, so the initial value of λ_n becomes significant.

To properly seal the gap between the pinned edge of the valve and the channel wall, we extend the pinned edges of the valve leaflets with a rigid stent, as shown in Figure 26. It is important to note that our variational immersed boundary method does not require this stent to exactly match the channel wall; it extends outside of the fluid domain, much like the rigid plate in the model problem of Section 4.3.

5.4.4. Results and discussion

We now discuss the results of computing with the setup described above. We compute for several cycles from the homogeneous initial condition, until reaching a time-periodic solution. We first consider the volumetric flow through the channel and how its features follow from our

boundary conditions. Next, we examine finer features of the fluid solution field. Finally, we compare the deformations of the valve leaflets to the results of the pressure-driven structural dynamics computation of Section 5.3.

Figure 28 shows the volumetric flow rate through the top of the tube throughout the cardiac cycle. Magnitudes of computed flow rate during systole and diastole are comparable to typical aortic flow rates, but we discuss below several unusual features of the computed profile. The most striking feature of the computed flow profile is the oscillation during diastole. This is a reverberation of the fluid hammer impact on the closing valve. This is a physical phenomenon, not a computational artifact, and is the source of the S_2 heart sound, marking the beginning of diastole [126, 127]. A similar decaying flow rate oscillation has been observed *in vitro* with flow loop experiments [128–130]. Further, the frequency of the computed oscillation (about 40 Hz) is within the range of observed aortic heart sound frequencies in patients with recently-implanted bioprosthetic aortic valves [131]. However, the magnitude of our computed oscillation is larger and it decays more slowly.

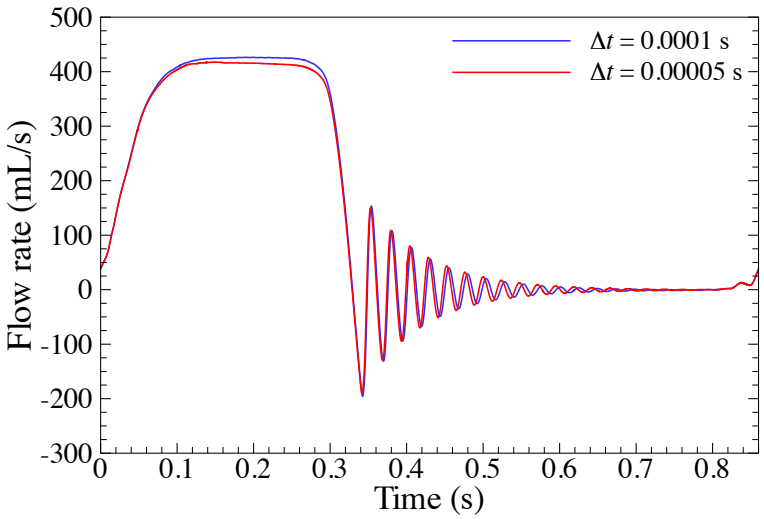


Figure 28: Computed volumetric flow rate through the top of the fluid domain, during a full cardiac cycle of 0.86 s, with $\Delta t = 1.0 \times 10^{-4}$ s and $\Delta t = 0.5 \times 10^{-4}$ s.

One may suspect that this prominent oscillation is the result of insufficient fluid–structure coupling, but, if this was the source of the oscillation, we would expect a significant difference between the computations with $\Delta t = 1.0 \times 10^{-4}$ s and $\Delta t = 0.5 \times 10^{-4}$ s, due to the twofold stiffening of the "displacement" penalty coefficient, $\tau^B_{\mathrm{NOR}}/\Delta t$, at the smaller time step. With the simplified fluid boundary conditions that we have applied, the oscillation in flow rate is most plausibly a consequence of the physical model, not the numerical method. In the analysis of closed hydraulic systems (such as the cardiovascular system), it is common to consider analogous electrical circuits [132]. The "circuit" that we are modeling is shown in Figure 29. The closed elastic valve acts

as a hydraulic accumulator, which is analogous to an electrical capacitor[6]. The inertia of the fluid acts like an inductor. These components, in series with the resistance of viscous forces and the boundary condition, are driven by a pressure difference, which fills the role of a time-varying voltage source in the electrical analogy. The exponentially decaying current oscillation observed in Figure 28 is qualitatively similar to the transient response of the corresponding RLC circuit to a sudden change in voltage. A more sophisticated model might include inductance and capacitance in the boundary conditions, to represent the inertia of blood outside of the computational domain and the Windkessel effect from large elastic arteries. The amplitude of the oscillation may also be exaggerated by our leaflet material model, which approximates the stiffness of a bioprosthetic valve about zero strain. The recruitment of collagen fibers in biological soft tissue leads to an exponential stiffening with strain that we have not attempted to model in this work, so the storage of a given amount of energy requires greater strain with our simplified valve.

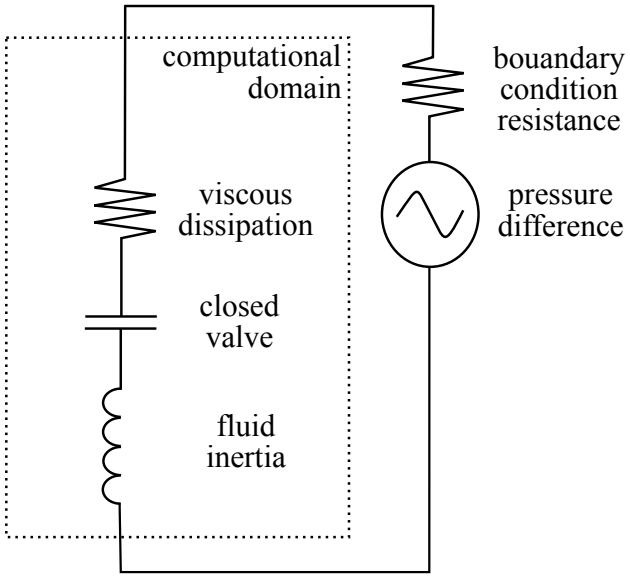


Figure 29: The electrical circuit that is analogous to the valve model of this paper (when the valve is closed).

Another physiologically unrealistic feature of the computed flow profile is the relatively flat flow rate during systole. Typically, the aortic flow rate reaches a rounded peak. This discrepancy may again be attributed to the simplified boundary conditions. Because the left ventricular pressure in our idealized pressure profile is constant for most of systole and the external flow loop is modeled only by a resistance and pressure difference, we expect the velocity of flow to asymptotically approach a terminal value at which the resistance of viscous forces and the boundary condition

[6]When current reverses and the valve opens, it will behave more like a resistor. Its overall behavior is not like that of any standard electrical circuit component; the typical analogy between valves and diodes would omit the capacitance in the closed state.

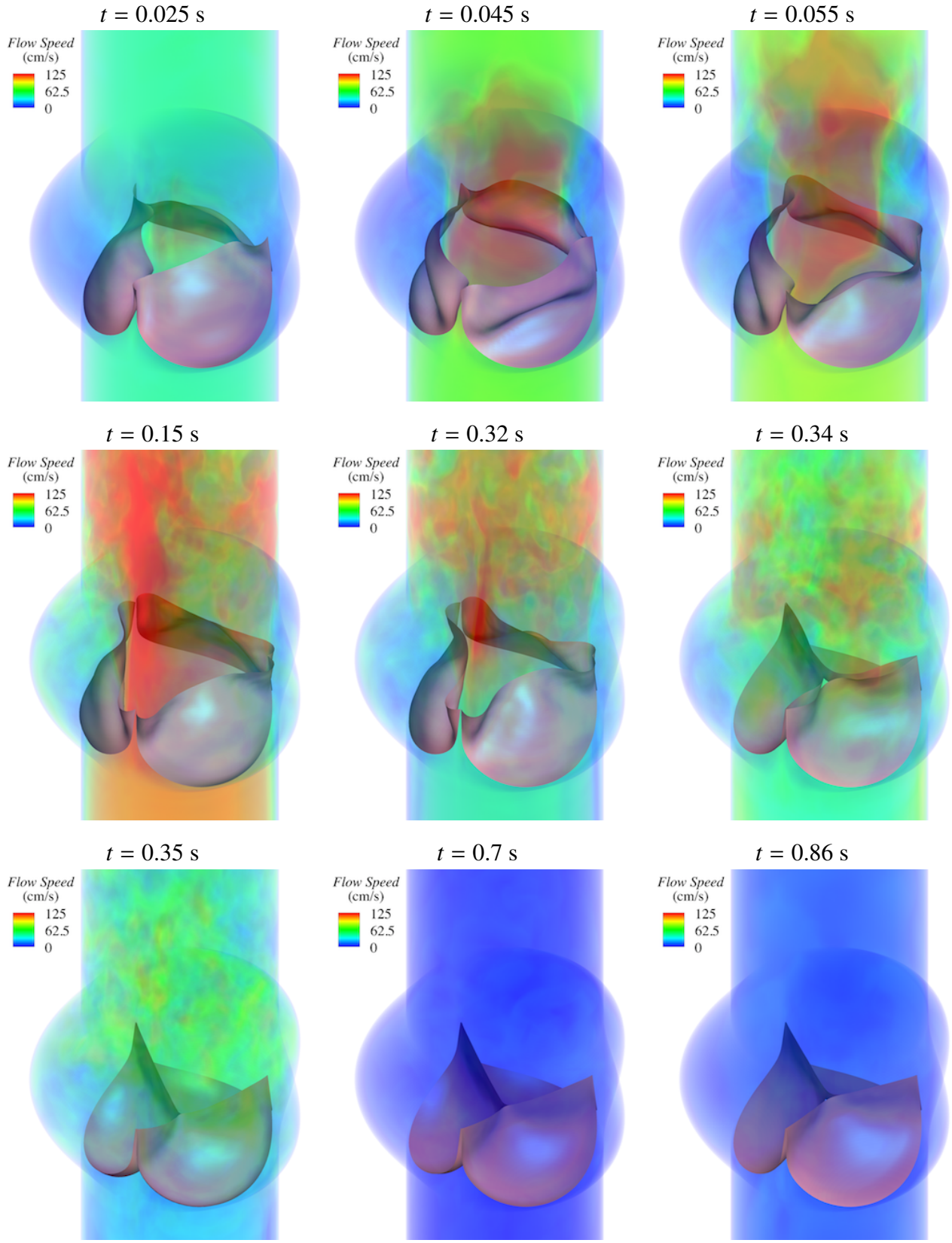


Figure 30: Volume-renderings of the fluid velocity field at several points during a cardiac cycle. The time t is synchronized with Figure 27 for the current cycle.

exactly balance the difference between the left ventricular pressure and p_0. This is in contrast to the physiological setting, in which flow contributes to a stored pressure as large arteries temporarily expand to accommodate the systolic output of the left ventricle. In the electrical analogy, these arteries act like a reservoir capacitor, smoothing the cardiac output.

The small rise in flow rate at the end of the cycle may seem counter-intuitive, given that the left ventricular pressure is still less than p_0. However, this flow corresponds to the valve returning to its stress-free configuration as the transvalvular pressure goes back to zero. In the electrical analogy, this corresponds to the current released by the capacitor (valve) discharging as the external voltage (pressure) difference is removed.

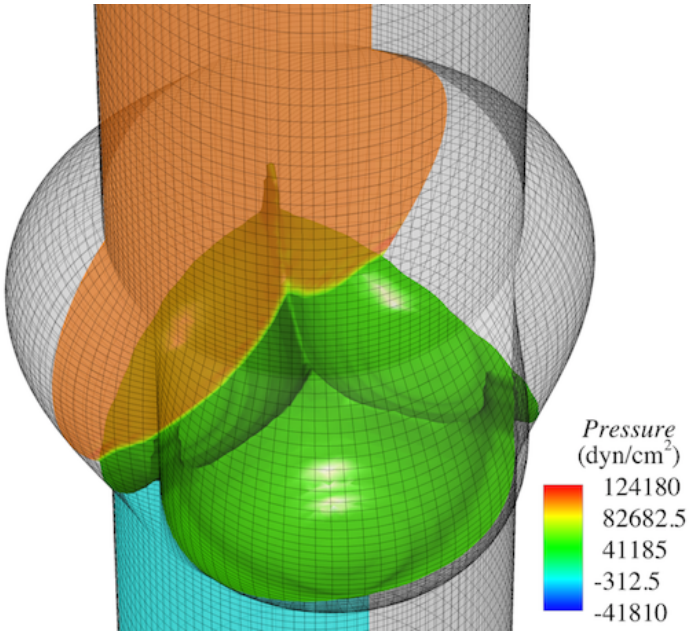


Figure 31: Pressure at time $t = 0.7$ s, shown on a slice and with an iso-surface at $p = 40$ mmHg.

We now look at the details of the fluid solution fields. In Figure 30, we show several snapshots of the fluid velocity field computed with the smaller time step of $\Delta t = 0.5 \times 10^{-4}$ s. As the valve opens, we see a transition to turbulent flow. This turbulence is exaggerated, in comparison to the physiological case [133], by the flow rate plateau at peak ejection. The valve begins to close under forward flow, as shown by the snapshot at $t = 0.32$ s. The snapshot at $t = 0.35$ s illustrates the fluid hammer effect that initially excites the oscillation evident in the flow rate. After 0.7 s, the S_2 heart sound is decayed and the solution becomes effectively hydrostatic. The fluid solution at $t = 0.7$ s is, however, not trivial. In Figure 31, we show a slice and iso-surface of the corresponding pressure field. The pressure below the valve is nearly zero, as prescribed by the left ventricular profile, and the pressure above the valve is around 106000 dyn/cm^2 (80 mmHg), which is the value chosen for p_0 in the outflow boundary condition. The iso-surface is at $p = 40$ mmHg, halfway between the pressures above and below the valve. It clearly displays the shape of the closed tri-leaflet valve and

rigid stent. A careful examination of this figure reveals small pressure oscillations near the valve, visible in both the slice and iso-surface. This is possibly a result of the weakened fluid stabilization near the structure.

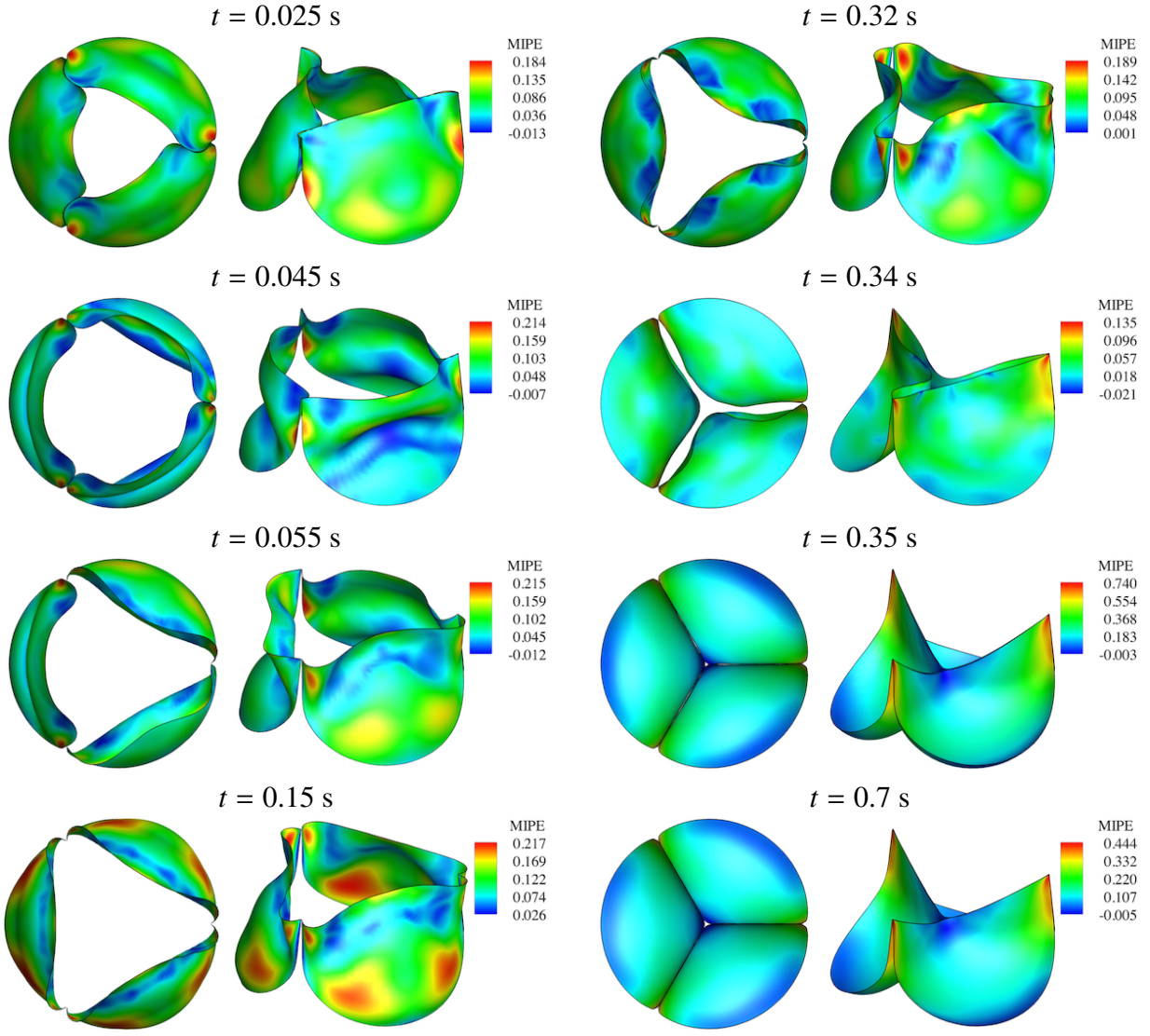


Figure 32: Deformations of the valve from the FSI computation, colored by maximum in-plane principal Green-Lagrange strain (MIPE, the largest eigenvalue of $\mathbf{E}$), evaluated on the aortic side of the leaflet. Note the different scale for each time.

The loading produced by the fluid differs significantly from the uniform pressure load prescribed in the computation of Section 5.3. Figure 32 shows the deformations and strain fields of the leaflets at several points during the cardiac cycle. The deformations during systole are markedly different from those computed using only structural dynamics. Specifically, the leaflets remain partially in contact while opening in the FSI simulation, whereas they immediately separate when a pressure load is applied in the structural dynamics computation. The strain field at time $t = 0.35$

s is also interesting in that the strain near the commissure points is significantly higher than it is at $t = 0.7$ s. This is due to the effect of the fluid hammer striking the valve as it initially closes. This phenomenon is completely neglected by both quasi-static and pressure-driven dynamic computations, as neither accounts for the inertia of the fluid. The FSI solution does not preserve the geometrical symmetry of the initial data. This loss of symmetry is typical of turbulent flow and was observed as well in the heart valve FSI computations of Borazjani [46]. This result underscores the importance of computing FSI for the entire valve, without symmetry assumptions.

6. Conclusions

This paper develops several variations of immersed boundary FSI within a variational framework based on the augmented Lagrangian Eq. (1). Prior work has connected this framework to an extension of Nitsche's method for fluid mechanics [51]. We apply this formulation to the CFD benchmark of 2D flow over a cylinder in Section 3.4. When applied to immersed boundary FSI for thin shell structures, modeled geometrically as surfaces, Nitsche's method reduces to the penalty method. The penalty method is effective for some problems, where pressure gradients are not too large, such as the benchmark problem considered in Section 4.6. However, we find that for applications such as heart valves, where large pressure gradients develop across thin structures, the penalty method has undesirable properties. We attempt to correct its deficiencies by retaining the Lagrange multiplier as a solution variable. We consider an iterative approximation of the multiplier, based on the work of Hestenes [108] and Powell [109]. For computations in which this method does not converge, we reduce it to the degenerate case of a single iteration in each time step. In that limit, it becomes analogous to Chorin's method of artificial compressibility [110], where the multiplier field solves an auxiliary differential equation in time.

We find that the approximation error that comes from representing a pressure discontinuity with continuous basis functions leads to poor local mass conservation near the discontinuity. This allows large velocity errors to develop in the rest of the domain. In Section 4.3, we introduce a preliminary work-around that modifies stabilization terms near the immersed structure. This appears to limit local compression without rendering the formulation unstable.

Figure 33 summarizes the interrelationships between the various FSI technologies developed in this work. Combined with a penalty-based contact algorithm for shell structures, these technologies allow us to simulate the dynamics of a BHV immersed and coupled in a cyclic, pressure-driven flow, with physiologically realistic pressure differences. We note that as is typically the case in FSI, different problem features demand different computational strategies.

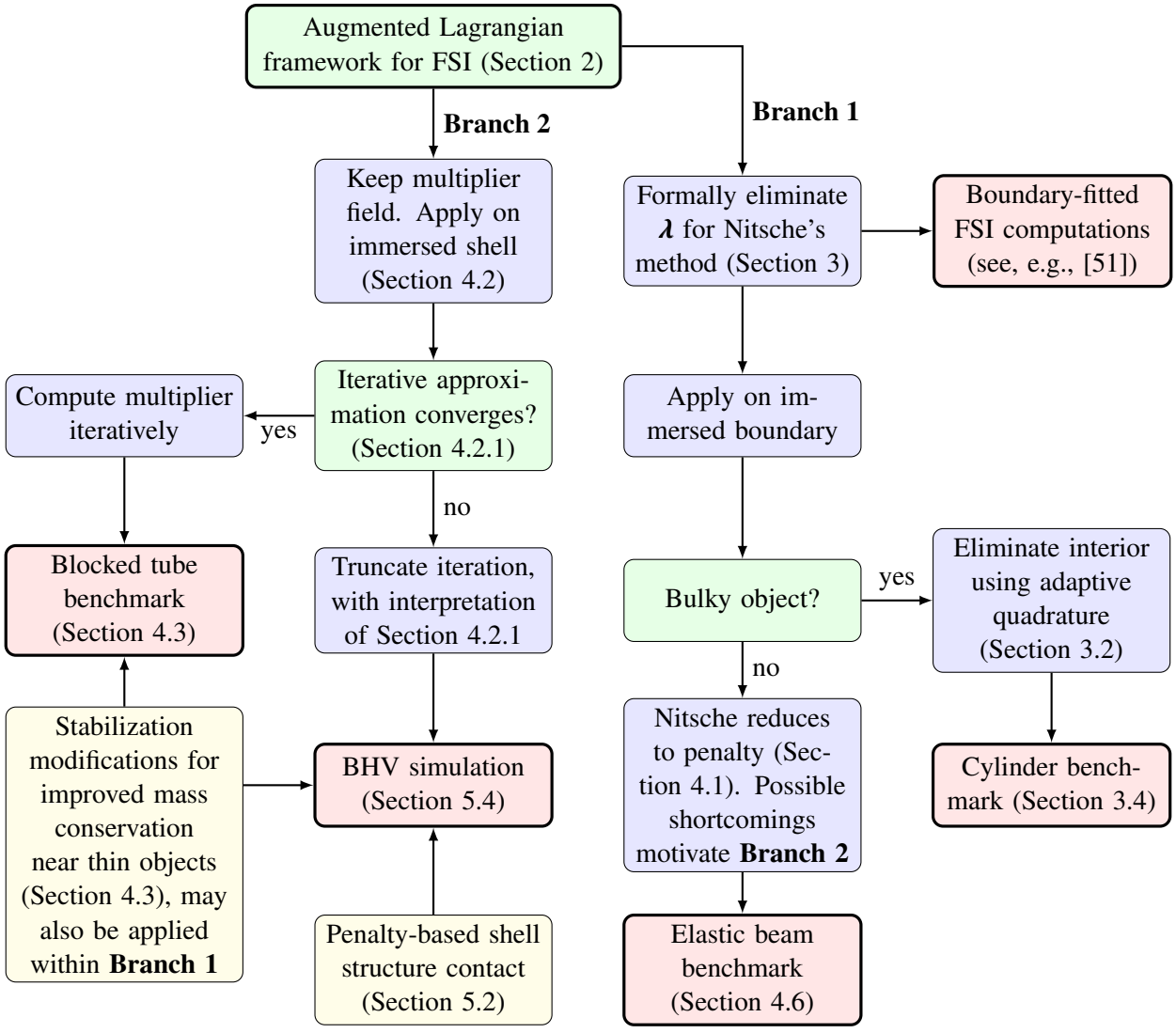


Figure 33: A graphical map of the interrelated ideas, methods, and results of this paper. Arrows indicate conceptual flow from ideas to numerical methods to specific computations. Branch-points in this flow are highlighted in green and computations are highlighted in red.

6.1. Limitations and further work

The current work motivates a number of refinements and extensions that we allude to throughout the body of the paper and summarize below.

- The FSI methods of this paper rely on penalty parameters. We have suggested guidelines, such as Eq. (21), for scaling these penalties with the approximation spaces and physical parameters, but we have not introduced explicit formulas. The appropriate definition of mesh size, "h", is not clear for the case of immersed boundaries. We have, in the computations of this paper, simply used constant penalties deemed effective through numerical experiments.

- We would like to develop a stable formulation to solve for the fluid–structure interface multiplier. As noted above, there is no obvious way to develop an inf-sup stable approximation space for the multiplier field, but we may be able to work within the framework of stabilized methods [107].

- The suppression of momentum stabilization near immersed shell structures that we develop in Section 4.3 is practically effective but aesthetically unappealing and not thoroughly studied. A theoretical study of the underlying approximation issue may reveal a more elegant solution. Alternatively, because the methods from this paper for enforcing the fluid–structure kinematic constraint are largely independent of the specific formulation for the fluid subproblem, they may easily be combined with variational fluid solvers that do not directly invoke the pressure gradient. For example, the use of a divergence-conforming approximation space for the fluid velocity could be modified to include concentrated boundary forces, while completely eliminating the problem of poor pressure approximation. The Lagrange multiplier (pressure) would no longer be needed to enforce a constraint that is built directly into the solution space. The emerging technology of divergence-conforming B-splines has been successfully applied to unsteady Navier–Stokes and would allow us to combine the advantages of isogeoemetric discretization with pointwise mass conservation [134].

- We discuss the lack of physical realism in our heart valve model at length in Section 5.4.4. To experimentally validate our method for valve simulation, we will need to introduce a more realistic material model for the valve leaflets and more sophisticated boundary conditions for the fluid domain.

- We also plan to use the hierarchical B-spline or NURBS refinement [135]. By hierarchically refining near the structure, one would be able to better resolve the pressure jump and the boundary layer. This could lead to improved results.

7. Acknowledgements

Funding for this work was supported by NIH/NHLBI grants R01 HL108330 and HL119297, and FDA contract HHSF223201111595P. T.J.R. Hughes was supported by grants from the Office of Naval Research (N00014-08-1-0992), the National Science Foundation (CMMI-01101007), and SINTEF (UTA10-000374) with the University of Texas at Austin. D. Kamensky was partially supported by the CSEM Graduate Fellowship. D. Schillinger was partially supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under grants SCHI 1249/1-1 and SCHI 1249/1-2. We thank the Texas Advanced Computing Center (TACC) at the University of Texas at Austin for providing HPC resources that have contributed to the research results reported in this paper. We would also like to thank Dr. Laura De Lorenzis at Technische Universität Braunschweig for helpful discussions on the contact problem and related algorithms.

References

[1] F. J. Schoen and R. J. Levy. Calcification of tissue heart valve substitutes: progress toward understanding and prevention. *Ann. Thorac. Surg.*, 79(3):1072–1080, 2005.

[2] P. Pibarot and J. G. Dumesnil. Prosthetic heart valves: selection of the optimal prosthesis and long-term management. *Circulation*, 119(7):1034–1048, 2009.

[3] R. F. Siddiqui, J. R. Abraham, and J. Butany. Bioprosthetic heart valves: modes of failure. *Histopathology*, 55:135–144, 2009.

[4] W. Sun, A. Abad, and M. S. Sacks. Simulated bioprosthetic heart valve deformation under quasi-static loading. *J Biomech Eng*, 127(6):905–914, 2005.

[5] H. Kim, J. Lu, M. S. Sacks, and K. B. Chandran. Dynamic simulation of bioprosthetic heart valves using a stress resultant shell model. *Annals of Biomedical Engineering*, 36(2):262–275, 2008.

[6] T. J. R. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29:329–349, 1981.

[7] J. Donea, S. Giuliani, and J. P. Halleux. An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1-3):689–723, 1982.

[8] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodriguez-Ferran. Arbitrary Lagrangian–Eulerian methods. In *Encyclopedia of Computational Mechanics*, Volume 3: Fluids, chapter 14. John Wiley & Sons, 2004.

[9] T. E. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: I. The concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, 94(3):339–351, 1992.

[10] T. E. Tezduyar, M. Behr, S. Mittal, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Computer Methods in Applied Mechanics and Engineering*, 94(3):353–371, 1992.

[11] A. Figueroa, I. E. Vignon-Clementel, K. E. Jansen, T. J. R. Hughes, and C. A. Taylor. A coupled momentum method for modeling blood flow in three-dimensional deformable arteries. *Computer Methods in Applied Mechanics and Engineering*, 195:5685–5706, 2006.

[12] R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T. E. Tezduyar. Fluid–structure interaction modeling of blood flow and cerebral aneurysm: Significance of artery and aneurysm shapes. *Computer Methods in Applied Mechanics and Engineering*, 198:3613–3621, 2009.

[13] Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel, and J. Isaksen. Computational fluid–structure interaction: Methods and application to cerebral aneurysms. *Biomechanics and Modeling in Mechanobiology*, 9:481–498, 2010.

[14] 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.

[15] C. C. Long, M.-C. Hsu, Y. Bazilevs, J. A. Feinstein, and A. L. Marsden. Fluid–structure interaction simulations of the Fontan procedure using variable wall properties. *International Journal for Numerical Methods in Biomedical Engineering*, 28:512–527, 2012.

[16] K. Takizawa, C. Moorman, S. Wright, T. Spielman, and T. E. Tezduyar. Fluid–structure interaction modeling and performance analysis of the Orion spacecraft parachutes. *International Journal for Numerical Methods in Fluids*, 65:271–285, 2011.

[17] K. Takizawa, S. Wright, C. Moorman, and T. E. Tezduyar. Fluid–structure interaction modeling of parachute clusters. *International Journal for Numerical Methods in Fluids*, 65:286–307, 2011.

[18] K. Takizawa, T. Spielman, and T. E. Tezduyar. Space–time FSI modeling and dynamical analysis of spacecraft parachutes and parachute clusters. *Computational Mechanics*, 48:345–364, 2011.

[19] K. Takizawa and T. E. Tezduyar. Computational methods for parachute fluid–structure interactions. *Archives of Computational Methods in Engineering*, 19:125–169, 2012.

[20] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65:236–253, 2011.

[21] M.-C. Hsu and Y. Bazilevs. Fluid–structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50:821–833, 2012.

[22] A. Korobenko, M.-C. Hsu, I. Akkerman, and Y. Bazilevs. Aerodynamic simulation of vertical-axis wind turbines. *Journal of Applied Mechanics*, 81:021011, 2014.

[23] Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, Chichester, 2013.

[24] A. A. Johnson and T. E. Tezduyar. Parallel computation of incompressible flows with complex geometries. *International Journal for Numerical Methods in Fluids*, 24:1321–1340, 1997.

[25] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal. Massively parallel finite element computation of 3D flows – mesh update strategies in computation of moving boundaries and interfaces. In A. Ecer, J. Hauser, P. Leca, and J. Periaux, editors, *Parallel Computational Fluid Dynamics – New Trends and Advances*, pages 21–30. Elsevier, 1995.

[26] A. A. Johnson and T. E. Tezduyar. 3D simulation of fluid-particle interactions with the number of particles reaching 100. *Computer Methods in Applied Mechanics and Engineering*, 145:301–321, 1997.

[27] A. A. Johnson and T. E. Tezduyar. Advanced mesh generation and update methods for 3D flow simulations. *Computational Mechanics*, 23:130–143, 1999.

[28] S. Sathe and T. E. Tezduyar. Modeling of fluid–structure interactions with the space–time finite elements: Contact problems. *Computational Mechanics*, 43:51–60, 2008.

[29] K. Takizawa, T. E. Tezduyar, A. Buscher, and S. Asada. Space–time interface-tracking with topology change (ST-TC). *Computational Mechanics*, 2013. doi:10.1007/s00466-013-0935-7.

[30] R. Cheng, Y. G. Lai, and K. B. Chandran. Three-dimensional fluid-structure interaction simulation of bileaflet mechanical heart valve flow dynamics. *Annals of Biomedical Engineering*, 32(11):1471–1483, 2004.

[31] Y. G. Lai, K. B. Chandran, and J. Lemmon. A numerical simulation of mechanical heart valve closure fluid dynamics. *Journal of Biomechanics*, 35(7):881–892, 2002.

[32] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193:2051–2067, 2004.

[33] Z. Yu. A DLM/FD method for fluid/flexible-body interactions. *Journal of Computational Physics*, 207:1–27, 2005.

[34] A. Gerstenberger and W. A. Wall. An eXtended Finite Element Method/Lagrange multiplier based approach for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 197:1699–1714, 2008.

[35] C. Hesch, A. J. Gil, A. Arranz Carreño, and J. Bonet. On continuum immersed strategies for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 247-248:51–64, 2012.

[36] T. Wick. Fully eulerian fluid–structure interaction for time-dependent problems. *Computer Methods in Applied Mechanics and Engineering*, 255:14–26, 2013.

[37] T Rüberg and F Cirak. A fixed-grid b-spline finite element technique for fluid–structure interaction. *International Journal for Numerical Methods in Fluids*, 2014. doi:10.1002/fld.3864.

[38] F. Sotiropoulos and X. Yang. Immersed boundary methods for simulating fluid–structure interaction. *Progress in Aerospace Sciences*, 65:1–21, 2014.

[39] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.

[40] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.

[41] J. O. Hallquist, G. L. Goudreau, and D. J. Benson. Sliding interfaces with contact-impact in large-scale Lagrangian computations. *Computer Methods in Applied Mechanics and Engineering*, 51:107–137, 1985.

[42] P. Wriggers. Finite element algorithms for contact problems. *Archives of Computational Methods in Engineering*, 2:1–49, 1995.

[43] P. Wriggers. *Computational Contact Mechanics, 2nd ed.* Springer-Verlag, Berlin Heidelberg, 2006.

[44] T. A. Laursen. *Computational Contact and Impact Mechanics: Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis*. Springer-Verlag, Berlin Heidelberg, 2003.

[45] C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.

[46] I. Borazjani. Fluid–structure interaction, immersed boundary-finite element method simulations of bio-prosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 257(0):103–116, 2013.

[47] L. Ge and F. Sotiropoulos. A numerical method for solving the 3D unsteady incompressible Navier–Stokes equations in curvilinear domains with complex immersed boundaries. *Journal of Computational Physics*, 225(2):1782–1809, 2007.

[48] I. Borazjani, L. Ge, and F. Sotiropoulos. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies. *Journal of Computational Physics*, 227(16):7587–7620, 2008.

[49] F. P. T. Baaijens. A fictitious domain/mortar element method for fluid–structure interaction. *International Journal for Numerical Methods in Fluids*, 35(7):743–761, 2001.

[50] J. de Hart. *Fluid–Structure Interaction in the Aortic Heart Valve: a three-dimensional computational analysis*. Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven, Netherlands, 2002.

[51] 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.

[52] J. Juntunen and R. Stenberg. Nitsche's method for general boundary conditions. *Mathematics of Computation*, 78:1353–1374, 2009.

[53] Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers and Fluids*, 36:12–26, 2007.

[54] 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.

[55] Y. Bazilevs, C. Michler, V. M. Calo, and T. J. R. Hughes. Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. *Computer Methods in Applied Mechanics and Engineering*, 199:780–790, 2010.

[56] Y. Bazilevs and I. Akkerman. Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual–based variational multiscale method. *Journal of Computational Physics*, 229:3402–3414, 2010.

[57] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. Wind turbine aerodynamics using ALE–VMS: Validation and the role of weakly enforced boundary conditions. *Computational Mechanics*, 50:499–511, 2012.

[58] 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.

[59] 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.

[60] 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.

[61] I. Akkerman, Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and S. Hulshoff. The role of continuity in residual-based variational multiscale modeling of turbulence. *Computational Mechanics*, 41:371–378, 2008.

[62] Y. Bazilevs, V. M. Calo, J. A. Cottrel, 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.

[63] 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.

[64] L. De Lorenzis, İ. Temizer, P. Wriggers, and G. Zavarise. A large deformation frictional contact formulation using NURBS-based isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 87:1278–1300, 2011.

[65] J. Nitsche. Uber ein variationsprinzip zur losung von Dirichlet-problemen bei verwendung von teilraumen, die keinen randbedingungen unterworfen sind. *Abh. Math. Univ. Hamburg*, 36:9–15, 1971.

[66] K. Höllig. *Finite Element Methods with B-Splines*. SIAM, Philadelphia, Pennsylvania, USA, 2003.

[67] 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.

[68] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[69] T. E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics*, 28:1–44, 1992.

[70] T. E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190:411–430, 2000.

[71] T. J. R. Hughes, L. Mazzei, A. A. Oberai, and A. Wray. The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence. *Physics of Fluids*, 13:505–512, 2001.

[72] T. J. R. Hughes, G. Scovazzi, and L. P. Franca. Multiscale and stabilized methods. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, Volume 3: Fluids, chapter 2. John Wiley & Sons, 2004.

[73] M.-C. Hsu, Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199:828–840, 2010.

[74] 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.

[75] C. Johnson. *Numerical solution of partial differential equations by the finite element method*. Cambridge University Press, Sweden, 1987.

[76] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods, 2nd ed.* Springer, 2002.

[77] A. Ern and J. L. Guermond. *Theory and Practice of Finite Elements*. Springer, 2004.

[78] 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.

[79] T. Wick. Flapping and contact FSI computations with the fluid–solid interface-tracking/interface-capturing technique and mesh adaptivity. *Computational Mechanics*, 53(1):29–43, 2014.

[80] Isaac Harari and Thomas J. R. Hughes. What are C and h?: Inequalities for the analysis and design of finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 97:157–192, 1992.

[81] A. Embar, J. Dolbow, and I. Harari. Imposing Dirichlet boundary conditions with Nitsche's method and spline-based finite elements. *International Journal for Numerical Methods in Engineering*, 83(7):877–898, 2010.

[82] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank. Weakly enforced essential boundary conditions for nurbs-embedded and trimmed nurbs geometries on the basis of the finite cell method. *International Journal for Numerical Methods in Engineering*, 95:811–846, 2013.

[83] M. Ruess, D. Schillinger, A. I. Özcan, and E. Rank. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 269:46–731, 2014.

[84] Y. Bazilevs, J. R. Gohean, T. J. R. Hughes, R. D. Moser, and Y. Zhang. Patient-specific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Computer Methods in Applied Mechanics and Engineering*, 198:3534–3550, 2009.

[85] M. Esmaily-Moghadam, Y. Bazilevs, T.-Y. Hsia, I. E. Vignon-Clementel, A. L. Marsden, and Modeling of Congenital Hearts Alliance (MOCHA). A comparison of outlet boundary

treatments for prevention of backflow divergence with relevance to blood flow simulations. *Computational Mechanics*, 48:277–291, 2011.

[86] J. Benk, M. Ulbrich, and M. Mehl. The Nitsche method of the Navier–Stokes equations for immersed and moving boundaries. In *Proceedings of the Seventh International Conference on Computational Fluid Dynamics*, ICCFD7. International Conference on Computational Fluid Dynamics, 2012.

[87] 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.

[88] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, and E. Rank. Small and large deformation analysis with the p- and B-spline versions of the Finite Cell Method. *Computational Mechanics*, 50(4):445–478, 2012.

[89] D. Schillinger and M. Ruess. The Finite Cell Method: A review in the context of higher-order structural analysis of CAD and image-based geometric models. *Archives of Computational Methods in Engineering*, accepted for publication, 2014.

[90] J. Parvizian, A. Düster, and E. Rank. Finite cell method. *Computational Mechanics*, 41(1):121–133, 2007.

[91] T. J. R. Hughes. *The finite element method: Linear static and dynamic finite element analysis*. Prentice Hall, Englewood Cliffs, NJ, 1987.

[92] P. R. Stay. The definition and ray-tracing of B-spline objects in a combinatorial solid geometric modeling system. In *Fourth USENIX Computer Graphics Workshop*. USENIX Association, 1987.

[93] 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.

[94] 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.

[95] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Computers & Fluids*, 49:93–100, 2011.

[96] M. Behr, D. Hastreiter, S. Mittal, and T. E. Tezduyar. Incompressible flow past a circular cylinder: Dependence of the computed flow field on the location of the lateral boundaries. *Computer Methods in Applied Mechanics and Engineering*, 123:309–316, 1995.

[97] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, 156(2):209–240, 1999.

[98] S. Mittal and A. Raghuvanshi. Control of vortex shedding behind circular cylinder for flows at low reynolds numbers. *International Journal for Numerical Methods in Fluids*, 35(4):421–447, 2001.

[99] A. L. F. Lima E Silva, A. Silveira-Neto, and J. J. R. Damasceno. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. *Journal of Computational Physics*, 189(2):351–370, 2003.

[100] D. Russell and Z. J. Wang. A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow. *Journal of Computational Physics*, 191:177–205, 2003.

[101] M.-H. Wu, C.-Y. Wen, R.-H. Yen, M.-C. Weng, and A.-B. Wang. Experimental and numerical study of the separation angle for flow around a circular cylinder at low Reynolds number. *Journal of Fluid Mechanics*, 515:233–260, 2004.

[102] P. H. Chiu, R. K. Lin, and T. W. H. Sheu. A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier–Stokes equations in time-varying complex geometries. *Journal of Computational Physics*, 229(12):4476–4500, 2010.

[103] T. Rüberg and F. Cirak. Subdivision-stabilised immersed B-spline finite elements for moving boundary flows. *Computer Methods in Applied Mechanics and Engineering*, 209–212(0):266–283, 2012.

[104] E. H. van Brummelen, K. G. van der Zee, V. V. Garg, and S. Prudhomme. Flux evaluation in primal and dual boundary-coupled problems. *Journal of Applied Mechanics*, 79:010904, 2011.

[105] L. N. Trefethen. Gibbs phenomenon. In *Approximation Theory and Approximation Practice*, chapter 9. SIAM, Philadelphia, Pennsylvania, USA, 2012.

[106] F. Hartmann. The discrete Babuška-Brezzi condition. *Ingenieur-Archiv*, 56(3):221–228, 1986.

[107] 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.

[108] M. R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[109] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, New York, 1969.

[110] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 135(2):118–125, 1967.

[111] C. H. Yap, N. Saikrishnan, G. Tamilselvan, and A. P. Yoganathan. Experimental technique of measuring dynamic fluid shear stress on the aortic surface of the aortic valve leaflet. *Journal of Biomechanical Engineering*, 133(6):061007, 2011.

[112] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199:2403–2416, 2010.

[113] J. Kiendl. *Isogeometric Analysis and Shape Optimal Design of Shell Structures*. PhD thesis, Lehrstuhl für Statik, Technische Universität München, 2011.

[114] D. J. Benson, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. A large deformation, rotation-free, isogeometric shell. *Computer Methods in Applied Mechanics and Engineering*, 200:1367–1378, 2011.

[115] T. E. Tezduyar and S. Sathe. Modelling of fluid–structure interactions with the space–time finite elements: Solution techniques. *International Journal for Numerical Methods in Fluids*, 54(6–8):855–900, 2007.

[116] W. Wall. *Fluid–Structure Interaction with Stabilized Finite Elements*. PhD thesis, University of Stuttgart, 1999.

[117] W. G. Dettmer and D. Perić. A fully implicit computational strategy for strongly coupled fluidsolid interaction. *Archives of Computational Methods in Engineering*, 14:205–247, 2007.

[118] C. Kassiotis, A. Ibrahimbegovic, R. Niekamp, and H. G. Matthies. Nonlinear fluid–structure interaction problem. Part I: implicit partitioned algorithm, nonlinear stability proof and validation examples. *Computational Mechanics*, 47:305–323, 2011.

[119] A. A. Johnson and T. E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 119:73–94, 1994.

[120] K. Stein, T. Tezduyar, and R. Benney. Mesh moving techniques for fluid–structure interactions with large displacements. *Journal of Applied Mechanics*, 70:58–63, 2003.

[121] K. Stein, T. E. Tezduyar, and R. Benney. Automatic mesh update with the solid-extension mesh moving technique. *Computer Methods in Applied Mechanics and Engineering*, 193:2019–2032, 2004.

[122] A. Aggarwal, G. Ferrari, E. Joyce, M. J. Daniels, R. Sainger, J. H. Gorman III, R. Gorman, and M. S. Sacks. Architectural trends in the human normal and bicuspid aortic valve leaflet and its relevance to valve disease. *Annals of Biomedical Engineering*, 42(5), 2014.

[123] R. Dimitri, L. De Lorenzis, M. A. Scott, P. Wriggers, R. L. Taylor, and G. Zavarise. Isogeometric large deformation frictionless contact using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 269:394–414, 2014.

[124] B. J. Bellhouse and F. H. Bellhouse. Mechanism of closure of the aortic valve. *Nature*, 217(5123):86–87, 1968.

[125] I. E. Vignon-Clementel, C. A. Figueroa, K. E. Jansen, and C. A. Taylor. Outflow boundary conditions for three-dimensional finite element modeling of blood flow and pressure in arteries. *Computer Methods in Applied Mechanics and Engineering*, 195:3776–3796, 2006.

[126] J. M. Felner. The second heart sound. In *Clinical Methods: The History, Physical, and Laboratory, 3rd edition*, chapter 23. Butterworths, Boston, USA, 1990.

[127] H. N. Sabbah and P. D. Stein. Relation of the second sound to diastolic vibration of the closed aortic valve. *American Journal of Physiology - Heart and Circulatory Physiology*, 234(6):H696–H700, 1978.

[128] H. Sugimoto and M. S. Sacks. Effects of leaflet stiffness on in vitro dynamic bioprosthetic heart valve leaflet shape. *Cardiovascular Engineering and Technology*, 4(1):2–15, 2013.

[129] H. L. Leo, H. Simon, J. Carberry, S. C. Lee, and A. P. Yoganathan. A comparison of flow field structures of two tri-leaflet polymeric heart valves. *Annals of Biomedical Engineering*, 33(4):429–443, 2005.

[130] H. L. Leo, L. P. Dasi, J. Carberry, H. A. Simon, and A. P. Yoganathan. Fluid dynamic assessment of three polymeric heart valves using particle image velocimetry. *Annals of Biomedical Engineering*, 34(6):936–952, 2006.

[131] P. D. Stein, H. N. Sabbah, J. B. Lakier, and S. Goldstein. Frequency spectrum of the aortic component of the second heart sound in patients with normal valves, aortic stenosis and aortic porcine xenografts: Potential for detection of porcine xenograft degeneration. *The American Journal of Cardiology*, 46(1):48–52, 1980.

[132] N. Westerhof, J.-W. Lankhaar, and B. E. Westerhof. The arterial Windkessel. *Medical & Biological Engineering & Computing*, 47:131–141, 2009.

[133] P. D. Stein and H. N. Sabbah. Turbulent blood flow in the ascending aorta of humans with normal and diseased aortic valves. *Circ. Res.*, 39(1):58–65, 1976.

[134] J. A. Evans and T. J. R. Hughes. Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations. *Journal of Computational Physics*, 241:141–167, 2013.

[135] D. Schillinger, L. Dedè, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, and T. J. R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249–252:116–150, 2012.