Oden Institute REPORT 19-11

June 2019

Computational Cardiovascular Analysis with the Variational Multiscale Methods and Isogeometric Discretization

by

Thomas J.R. Hughes, Kenji Takizawa, Yuri Bazilevs, Tayfun E. Tezduyar, and Ming-Chen Hsu

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

Reference: Thomas J.R. Hughes, Kenji Takizawa, Yuri Bazilevs, Tayfun E. Tezduyar, and Ming-Chen Hsu, "Computational Cardiovascular Analysis with the Variational Multiscale Methods and Isogeometric Discretization," Oden Institute REPORT 19-11, Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, June 2019.

Computational Cardiovascular Analysis with the Variational Multiscale Methods and Isogeometric Discretization

Thomas J.R. Hughes, Kenji Takizawa, Yuri Bazilevs, Tayfun E. Tezduyar and Ming-Chen Hsu

Abstract Computational cardiovascular analysis can provide valuable information to cardiologists and cardiovascular surgeons on a patient-specific basis. There are many computational challenges that need to be faced in this class of flow analyses. They include highly unsteady flows, complex cardiovascular geometries, moving boundaries and interfaces, such as the motion of the heart valve leaflets, contact between moving solid surfaces, such as the contact between the leaflets, and the fluid–structure interaction between blood and cardiovascular structure. Many of these challenges have been or are being addressed by the Space–Time Variational Mul-

Thomas J.R. Hughes
Institute for Computational Engineering and Sciences
The University of Texas at Austin, 1 University Station, C0200
201 E. 24th Street, Austin, TX 78712, USA
e-mail: hughes@ices.utexas.edu

Kenji Takizawa
Department of Modern Mechanical Engineering, Waseda University
3-4-1 Ookubo, Shinjuku-ku, Tokyo 169-8555, Japan
e-mail: Kenji.Takizawa@tafsm.org

Yuri Bazilevs
School of Engineering, Brown University
184 Hope Street, Providence, RI 02912, USA
e-mail: yuri_bazilevs@brown.edu

Tayfun E. Tezduyar
Mechanical Engineering, Rice University – MS 321
6100 Main Street, Houston, TX 77005, USA
Faculty of Science and Engineering, Waseda University
3-4-1 Ookubo, Shinjuku-ku, Tokyo 169-8555, Japan
e-mail: tezduyar@tafsm.org

Ming-Chen Hsu
Department of Mechanical Engineering
Iowa State University
2025 Black Engineering, Ames, IA 50011, USA
e-mail: jmchsu@iastate.edu

tiscale (ST-VMS) method, the Arbitrary Lagrangian–Eulerian VMS (ALE-VMS) method, and VMS-based Immersogeometric Analysis (IMGA-VMS), which serve as the core computational methods, and other special methods used in combination with them. We provide an overview of these methods and present examples of challenging computations carried out with them, including aortic and heart valve flow analyses. We also point out that these methods are general computational fluid dynamics techniques and have broad applicability to a wide range of other areas of science and engineering.

1 Introduction

In this article we review general Computational Fluid Dynamics (CFD) methods that we have developed and used over an almost five-decade period on a variety of applications in science, engineering and medicine. However, our focal application area herein is Computational Medicine and in particular Computational Cardiovascular Analysis. This area has a long history, in fact the senior author (TJRH) did his PhD thesis in it in 1974, and there was even earlier work than this, but the area took on a new direction in the mid-1990s when the first patient-specific calculations were performed with models created from medical imaging data, such as MRI and CT. The archival journal paper that began this trend was [1]. Up to that time Computational Cardiovascular Analysis was focused on very simple two-dimensional geometries such as straight and circular channels, and thus had almost no clinical significance. After [1], the subject began dramatically transform to where it is today, in which detailed analyses of a wide variety of patient-specific configurations are routinely analyzed to diagnose disease, plan surgeries and interventions, such as stenting and bypass grafting, and to virtually evaluate medical devices, such as left ventricular assist devices (LVADs), implanted in individual patients. Our purpose here is not to describe the array of medical applications of Computational Cardiovascular Analysis (for these we would refer in particular to the works of Charles A. Taylor, Alison Marsden, and Alberto Figueroa, among others), but rather to describe the main technologies that support these applications. This started with the seminal work of the senior author [2] and the algorithm which has become known by the acronym SUPG, which was extracted from the name given by the authors, the "Streamline-Upwind Petrov-Galerkin" method. Reference [2] was the first archival journal publication of the basic ideas, but earlier, starting in 1979, there were several now obscure, conference proceedings papers that preceded it. We have to acknowledge that the name is not great. However, the ideas embodied therein were important and have had significant subsequent impact. The basic problem of Computational Fluid Dynamics (CFD) at the time was achieving a combination of good stability and high accuracy in one algorithm. Many investigators viewed stability and accuracy as competing attributes. Reference [2] proved otherwise computationally, and mathematical analyses justified what was observed subsequently, the first being [3]. The fundamental concept employed was "residual-based stabilization," which added weighted residuals of the numerical solution to basic Galerkin formulations. Residual-based methods are a priori consistent and thus capable of preserve the underlying accuracy of Galerkin methods, while at the same time appropriate weighting enhanced their stability. Numerous "Stabilized Methods," as they have been commonly referred to subsequently, were then developed over the years based on this paradigm. The success of Stabilized Methods, another somewhat unfortunate name in our opinion, cannot be over-estimated. The number of citations these works have garnered is staggering, e.g., [2] alone has received approximately 6,000 citations. Although the mathematical analysis of Stabilized Methods developed as a field in its own right shortly after the initial publications, the creation of new Stabilized Methods technologies, such as for example residual-based discontinuity capturing operators, was essentially based largely on intuition. The breakthrough concept that derived Stabilized Methods from the fundamental governing equations was the Variational Multiscale Method [4, 5, 6, 7]. This provided an approach to derive consistent Stabilized Methods directly from any system of linear or nonlinear equations in fluid dynamics, or any scientific discipline, and it has been perhaps the most powerful development tool in the arsenal of CFD technologies.

Stabilized Methods and the Variational Multiscale Method are fundamental to all our works in Computational Cardiovascular Analysis. Many other technologies have been developed that further extend these basic building blocks to specific classes of problems and phenomena. This article describes the use of these methods in Computational Cardiovascular Analysis, with a focus on two specific areas, namely, aortic flow phenomena [8]) and patient-specific and bioprosthetic heart-valve fluid-structure interaction [9, 10]. We wish to also emphasize that these applications are only a small sample of activity in this rapidly growing field. There are many formidable challenges posed by problems of these types, including highly unsteady flows, complex diseased geometries, moving boundaries and interfaces (e.g. motion of heart valve leaflets), contact between moving solid surfaces within a flow (e.g. contact between heart valve leaflets), and the fluid-structure interaction of blood flow with cardiovascular structures, such as arteries, heart valves, etc. Many of these challenges have been or are being addressed by the Space–Time Variational Multiscale (ST-VMS) method [11], Arbitrary Lagrangian–Eulerian VMS (ALE-VMS) method [12], and the VMS-based Immersogeometric Analysis (IMGA-VMS) [9], which serve as the core computational methods. The special methods used in combination with the ST-VMS include the Space–Time Slip Interface (ST-SI) method [13], Space–Time Topology Change (ST-TC) [14] method, Space–Time Isogeometric Analysis (ST-IGA) [15, 16], integration of these methods, and a general-purpose NURBS mesh generation method for complex geometries [17]. The special methods used in combination with ALE-VMS include weak enforcement of no-slip boundary conditions [18], "sliding interfaces" [19] (the acronym "SI" will also indicate that) and backflow stabilization [20].

Despite the focus of this article on problems of Computational Cardiovascular Analysis, the methods described herein are general CFD and fluid-structure interaction technologies that have wide applicability to diverse scientific and engineering applications, and therefore we also take the opportunity to draw attention to many such applications that the authors of this chapter have been actively involved with.

1.1 Space–Time Stabilized and VMS Methods

The Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) method [21] was introduced for computation of flows with moving boundaries and interfaces (MBI), including fluid–structure interaction (FSI). In MBI computations the DSD/SST functions as a moving-mesh method. Moving the fluid mechanics mesh to follow an interface enables mesh-resolution control near the interface and, consequently, high-resolution boundary-layer representation near fluid–solid interfaces. The stabilization components of the original DSD/SST are the Streamline-Upwind/Petrov-Galerkin (SUPG) [2] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [21] stabilizations, which are used widely. Because of the SUPG and PSPG components, the original DSD/SST is now called "ST-SUPS." The ST-VMS is the VMS version of the DSD/SST. The VMS components of the ST-VMS are from the residual-based VMS (RBVMS) method [4, 7]. The ST-VMS has two more stabilization terms beyond those in the ST-SUPS, and the additional terms give the method better turbulence modeling features. The ST-SUPS and ST-VMS, because of the higher-order accuracy of the Space–Time (ST) framework (see [11]), are desirable also in computations without MBI.

The ST-SUPS and ST-VMS have been applied to many classes of FSI, MBI and fluid mechanics problems (see [22] for a comprehensive summary). The classes of problems include spacecraft parachute analysis for the landing-stage parachutes [23], cover-separation parachutes [24] and drogue parachutes [25], wind-turbine aerodynamics for horizontal-axis wind-turbine rotors [26], full horizontal-axis wind-turbines [27] and vertical-axis wind-turbines [13], flapping-wing aerodynamics for an actual locust [28], bioinspired MAVs [29] and wing-clapping [30], blood flow analysis of cerebral aneurysms [31], stent-treated aneurysms [32], aortas [8] and heart valves [10], spacecraft aerodynamics [24], thermo-fluid analysis of ground vehicles and their tires [33], thermo-fluid analysis of disk brakes [34], flow-driven filament dynamics in turbomachinery [35], flow analysis of turbocharger turbines [36], flow around tires with road contact and deformation [37], fluid films [38], ram-air parachutes [39], and compressible-flow spacecraft parachute aerodynamics [40].

The space–time computational methods have a relatively long track record in arterial FSI analysis, starting with computations reported in [41, 42]. These were among the earliest arterial FSI computations, and the core method was the ST-SUPS. Many space–time computations were also reported in the last 15 years. In the first 8 years of that period the space–time computations were performed for FSI of the abdominal aorta [43], carotid artery [43] and cerebral aneurysms [44]. In the last 7 years, the space–time computations focused on even more challenging aspects of cardiovascular fluid mechanics and FSI, including comparative studies of cerebral aneurysms [31], stent treatment of cerebral aneurysms [45], heart valve flow computation [10], aortic flow analysis [8], and coronary arterial dynamics [46].

In the flow analyses presented here, the space–time framework provides higher-order accuracy. The VMS feature of the ST-VMS addresses the computational challenges associated with the multiscale nature of the unsteady flow. The moving-mesh feature of the space–time framework enables high-resolution computation near the moving heart valve leaflets.

1.2 ALE Stabilized and VMS Methods

The ALE-VMS method [12] is the VMS version of ALE [47]. It succeeded the ST-SUPS [21] and ALE-SUPS [48] and preceded the ST-VMS. The VMS components are from the RBVMS [4, 7]. The ALE-VMS originated from the RBVMS formulation of incompressible turbulent flows proposed in [7] for nonmoving meshes, and may be thought of as an extension of the RBVMS to moving meshes. As such, it was presented for the first time in [12] in the context of FSI. To increase their scope and accuracy, the ALE-VMS and RBVMS are often supplemented with special methods, such as those for weakly-enforced no-slip boundary conditions [18], “sliding interfaces” [19] and backflow stabilization [20]. The ALE-SUPS, RBVMS and ALE-VMS have been applied to many classes of FSI, MBI and fluid mechanics problems including ram-air parachute FSI [48], wind-turbine aerodynamics and FSI [49, 50], vertical-axis wind turbines [50], floating wind turbines [51], wind turbines in atmospheric boundary layers [50], fatigue damage in wind-turbine blades [52], patient-specific cardiovascular fluid mechanics and FSI [53, 54], biomedical-device FSI [55, 56], ship hydrodynamics with free-surface flow and fluid–object interaction [57], hydrodynamics and FSI of hydraulic arresting gear [58], hydrodynamics of tidal-stream turbines with free-surface flow [59], passive-morphing FSI in turbomachinery [60], bioinspired FSI for marine propulsion [61], and bridge aerodynamics and fluid–object interaction [62]. Recent advances in stabilized and multiscale methods may be found for stratified incompressible flows [63], divergence-conforming discretizations of incompressible flows [64], and compressible flows with emphasis on gas-turbine modeling [65].

In the flow analyses presented here, the VMS feature of ALE-VMS addresses the computational challenges associated with the multiscale nature of the unsteady flow. The moving-mesh feature of the ALE framework enables high-resolution computation near the moving wall of a thoracic aorta.

1.3 Slip Interface Space–Time Method

The Space–Time version of the Slip Interface (ST-SI) method was introduced in [13] in the context of incompressible-flow equations to retain the desirable moving-mesh features of the ST-VMS and ST-SUPS when there are spinning solid surfaces, such as for a turbine rotor. The mesh covering the spinning surface spins with it, retaining the high-resolution representation of boundary layers. The starting point in the development of ST-SI was the version of ALE-VMS for computations with sliding interfaces [19]. Interface terms similar to those in the ALE-VMS version are added to ST-VMS to account for the compatibility conditions for velocity and stress at the slip interface. That accurately connects the two sides of the solution. An ST-SI version where the slip interface is between fluid and solid domains was also presented in [13]. The slip interface in this case is a “fluid–solid” interface rather than a standard “fluid–fluid” interface, and enables weak enforcement of the Dirichlet boundary conditions for the fluid. The ST-SI introduced in [34] for the coupled incompressible-flow and thermal-transport equations retains the high-resolution representation of the thermo-fluid boundary layers near spinning solid surfaces. These ST-SI methods have been applied to aerodynamic analysis of vertical-axis wind turbines [13], thermo-fluid analysis of disk brakes [34], flow-driven filament dynamics in turbomachinery [35], flow analysis of turbocharger turbines [36], flow around tires with road contact and deformation [37], fluid films [38], aerodynamic analysis of ram-air parachutes [39], and flow analysis of heart valves [10].

In the ST-SI version presented in [13] the slip interface is between a thin porous structure and the fluid on its two sides. This enables dealing with the porosity in a fashion consistent with how the standard fluid–fluid slip interfaces are dealt with and how the Dirichlet conditions are enforced weakly with fluid–solid slip interfaces. This version also enables handling thin structures that have T-junctions. This method has been applied to incompressible-flow aerodynamic analysis of ram-air parachutes with fabric porosity [39]. The compressible-flow ST-SI methods were introduced in [40], including the version where the slip interface is between a thin porous structure and the fluid on both its sides. Compressible-flow porosity models were also introduced in [40]. These, together with the compressible-flow space–time SUPG method [66], extended the space–time computational analysis range to compressible-flow aerodynamics of parachutes with fabric and geometric porosities. That enabled space–time computational flow analysis of the Orion spacecraft drogue parachute in the compressible-flow regime [67].

1.4 Immersogeometric VMS Analysis

The Immersogeometric Analysis (IMGA) was introduced in [56] as a geometrically flexible technique for solving FSI problems involving large, complex structural deformations and change of fluid-domain topology (e.g., structural contact). The motivating application is the simulation of heart valve function over a complete cardiac cycle. The method directly analyzes a spline representation of a thin structure by immersing it into a non-body-fitted discretization of the background fluid domain, and focuses on accurately capturing the immersed design geometry within non-body-fitted analysis meshes. A new semi-implicit numerical method, which we now refer to as the Dynamic Augmented Lagrangian (DAL) approach [68], was introduced in [56] for weakly enforcing constraints in time-dependent immersogeometric FSI problems. A mixed ALE-VMS/IMGA-VMS (ALE-IMGA-VMS) method was developed in [9] in the framework of the Fluid–Solid Interface-Tracking/Interface-Capturing Technique [69]; a single computation combines a body-fitted, moving-mesh treatment of some fluid–structure interfaces, with a non-body-fitted treatment of others. This approach enables us to simulate the FSI of a bioprosthetic heart valve (BHV) in a deforming artery over the entire cardiac cycle under physiological conditions, and study the effect of arterial-wall elasticity on valve dynamics [9]. The DAL-based ALE-IMGA-VMS was integrated with Computer-Aided Design (CAD) for heart-valve analysis in [55] with a thorough comparison between pressure-driven only and full FSI computations. An anisotropic constitutive modeling of BHV leaflets for immersogeometric FSI, based on the Kirchhoff–Love shell formulation for general hyperelastic materials [70], is proposed in [71]. A divergence-conforming formulation of incompressible flow, which gives a pointwise divergence-free velocity field everywhere in the domain, completely eliminates mass loss error across the valve interface in [72]. Stable coupling strategies and suitable definition of Lagrange multipliers for the DAL numerical approach were proposed and analyzed in [73]. The FSI framework of ALE-IMGA-VMS was employed in patient-specific valve design in [74]. The DAL-based IMGA has also been combined with surrogate modeling in [58] for an efficient and effective use of FSI to optimize the design of a hydraulic arresting gear.

1.5 Stabilization Parameters

The methods discussed in this chapter all have some embedded stabilization parameters that play a significant role (see [75, 13]). There are many ways of defining these stabilization parameters (for examples, see [76, 77, 78, 79, 33, 80, 37]). The stabilization-parameter definitions used in the computations reported in this article can be found from the references cited in the sections where those computations are described.

1.6 Topology Change Space–Time Method

The Topology Change Space–Time method (ST-TC) [14] was introduced for moving-mesh computation of flow problems with topology change, such as contact between solid surfaces. Even before the ST-TC, the ST-SUPS and ST-VMS, when used with robust mesh update methods, have proven effective in flow computations where the solid surfaces are in near contact or create other near topology change. Many classes of problems can be solved that way with sufficient accuracy by approximating actual contact with a small gap between the solid surfaces. For examples of such computations, see the references mentioned in [14]. The ST-TC made moving-mesh computations possible even when there is an actual contact between solid surfaces or other topology change. By collapsing elements as needed, without changing the connectivity of the "parent" mesh, the ST-TC can handle an actual topology cgange while maintaining high-resolution boundary layer representation near solid surfaces. This enabled successful moving-mesh computation of heart valve flows [10], wing clapping [30], and flow around a rotating tire with road contact and prescribed deformation [37].

For more on the ST-TC, see [14]. In the computational analyses here, the ST-TC enables moving-mesh computation even with the topology change created by the actual contact between the valve leaflets. It deals with the contact while maintaining high-resolution flow representation near the leaflet.

1.7 Topology Change Slip Interface Space–Time Method

The Topology Change Slip Interface Space–Time Method (ST-SI-TC) is the integration of the ST-SI and ST-TC. A fluid–fluid slip interface requires elements on both sides of the interface. When part of a slip interface needs to coincide with a solid surface, which happens for example when the solid surfaces on two sides of the interface come into contact or when the inteface reaches a solid surface, the elements between the coinciding slip interface part and the solid surface need to collapse with the ST-TC mechanism. The collapse switches the slip interface from the fluid–fluid type to the fluid–solid type. With that, a slip interface can be a mixture of the fluid–fluid and fluid–solid types. With the ST-SI-TC, the elements collapse and are reborn independent of the nodes representing a solid surface. The ST-SI-TC enables high-resolution flow representation even when parts of the slip interface are coinciding with a solid surface. It also enables dealing with contact location change and contact sliding. This was applied to heart valve flow analysis [10] and tire aerodynamics with road contact and deformation [37].

For more on the ST-SI-TC, see [81]. In the computational analyses presented here, the ST-SI-TC enables high-resolution representation of the boundary layers even when the contact is between leaflets that are in mesh sectors connected by slip interfaces. It enables contact location change and contact sliding between the leaflets.

1.8 Space–Time IGA

The ST-IGA, introduced in [11], is the integration of the space–time framework with isogeometric discretization, motivated by the success of NURBS meshes in spatial discretization [82, 53, 12, 19]. Computations with the ST-VMS and ST-IGA were first reported in [11] in a 2D context, with IGA basis functions in space for flow past an airfoil, and in both space and time for the advection equation. Using higher-order basis functions in time enables getting full benefit out of using higher-order basis functions in space. This was demonstrated with the stability and accuracy analysis given in [11] for the advection equation.

The ST-IGA with IGA basis functions in time enables a more accurate representation of the motion of the solid surfaces and a mesh motion consistent with that. This was pointed out in [11] and demonstrated in [15]. It also enables more efficient temporal representation of the motion and deformation of the volume meshes, and more efficient remeshing. These motivated the development of the ST/NURBS Mesh Update Method (STNMUM) [15, 79]. The STNMUM has a wide scope that includes spinning solid surfaces. With the spinning motion represented by quadratic NURBS in time, and with sufficient number of temporal patches for a full rotation, the circular paths are represented exactly. A "secondary mapping" [11] enables also specifying a constant angular velocity for invariant speeds along the circular paths. The space–time framework and NURBS in time also enable, with the "ST-C" method, extracting a continuous representation from the computed data and, in large-scale computations, efficient data compression [83]. The STNMUM and the ST-IGA with IGA basis functions in time have been used in many 3D computations. The classes of problems solved are flapping-wing aerodynamics for an actual locust [28], bioinspired MAVs [29] and wing-clapping [30], separation aerodynamics of spacecraft [24], aerodynamics of horizontal-axis [31] and vertical-axis [13] wind turbines, thermo-fluid analysis of ground vehicles and their tires [33], thermo-fluid analysis of disk brakes [34], flow-driven string dynamics in turbomachinery [35], and flow analysis of turbocharger turbines [36].

The ST-IGA with IGA basis functions in space enables more accurate representation of the geometry and increased accuracy in the flow solution. It accomplishes that with fewer control points, and consequently with larger effective element sizes. That in turn enables using larger time-step sizes while keeping the Courant number at a desirable level for good accuracy. It has been used in space–time computational flow analysis of turbocharger turbines [36], flow-driven string dynamics in turbomachinery [35], ram-air parachutes [39], spacecraft parachutes [67], aortas [8], heart valves [10], tires with road contact and deformation [37], and fluid films [38]. Using IGA basis functions in space is now a key part of some of the newest Zero Stress State (ZSS) estimation methods [84] and related shell analysis [85].

For more on the ST-IGA, see [16]. In the computational flow analyses presented here, the ST-IGA enables more accurate representation of the cardiovascular geometries, increased accuracy in the flow solution, and using larger time-step sizes.

1.9 Space–Time IGA with Slip Interface and Topology Change

The turbocharger turbine analysis [36] and flow-driven string dynamics in turbomachinery [35] were based on the integration of the ST-SI and ST-IGA. The IGA basis functions were used in the spatial discretization of the fluid mechanics equations and also in the temporal representation of the rotor and spinning-mesh motion. That enabled accurate representation of the turbine geometry and rotor motion and increased accuracy in the flow solution. The IGA basis functions were used also in the spatial discretization of the string structural dynamics equations. That enabled increased accuracy in the structural dynamics solution, as well as smoothness in the string shape and fluid dynamics forces computed on the string.

The ram-air parachute analysis [39] and spacecraft parachute compressible-flow analysis [67] were based on the integration of the ST-IGA, the ST-SI version that weakly enforces the Dirichlet conditions, and the ST-SI version that accounts for the porosity of a thin structure. The ST-IGA with IGA basis functions in space enabled, with relatively few number of unknowns, accurate representation of the parafoil and parachute geometries and increased accuracy in the flow solution. The volume mesh needed to be generated both inside and outside the parafoil. Mesh generation inside was challenging near the trailing edge because of the narrowing space. The spacecraft parachute has a very complex geometry, including gores and gaps. Using IGA basis functions addressed those challenges and still kept the element density near the trailing edge of the parafoil and around the spacecraft parachute at a reasonable level.

The heart valve analysis [10] was based on the integration of the ST-SI, ST-TC and ST-IGA, which we refer to as ST-SI-TC-IGA. The ST-SI-TC-IGA, beyond enabling a more accurate representation of the geometry and increased accuracy in the flow solution, kept the element density in the narrow spaces near the contact areas at a reasonable level. When solid surfaces come into contact, the elements between the surface and the slip interface collapse. Before the elements collapse, the boundaries could be curved and rather complex, and the narrow spaces might have high-aspect-ratio elements. With NURBS elements, it was possible to deal with such adverse conditions rather effectively.

In computational analysis of flow around tires with road contact and deformation [37], the ST-SI-TC-IGA enables a more accurate representation of the geometry and motion of the tire surfaces, a mesh motion consistent with that, and increased accuracy in the flow solution. It also keeps the element density in the tire grooves and in the narrow spaces near the contact areas at a reasonable level. In addition, we benefit from the mesh generation flexibility provided by using SIs.

An SI provides mesh generation flexibility in a general context by accurately connecting the two sides of the solution computed over nonmatching meshes. This type of mesh generation flexibility is especially valuable in complex-geometry flow computations with isogeometric discretization, removing the matching requirement between the NURBS patches without loss of accuracy. This feature was used in the flow analysis of heart valves [10], turbocharger turbines [36], and spacecraft parachute compressible-flow analysis [67].

For more on the ST-SI-TC-IGA, see [10]. In the computations presented here, the ST-SI-TC-IGA is used in the heart valve flow analysis, for the reasons given and as described in an earlier paragraph of this section.

1.10 General-Purpose NURBS Mesh Generation Method

To make the ST-IGA use, and in a wider context the IGA use, even more practical in computational flow analysis with complex geometries, NURBS volume mesh generation needs to be easier and more automated. To that end, a general-purpose NURBS mesh generation method was introduced in [17]. The method is based on multi-block-structured mesh generation with existing techniques, projection of that mesh to a NURBS mesh made of patches that correspond to the blocks, and recovery of the original model surfaces. The method is expected to retain the refinement distribution and element quality of the multi-block-structured mesh that we start with. Because there are ample good techniques and software for generating multi-block-structured meshes, the method makes general-purpose mesh generation relatively easy.

Mesh-quality performance studies for 2D and 3D meshes, including those for complex models, were presented in [86]. A test computation for a turbocharger turbine and exhaust manifold was also presented in [86], with a more detailed computation in [36]. The mesh generation method was used also in the pump-flow analysis part of the flow-driven string dynamics presented in [35] and in the aorta flow analysis presented in [8]. The performance studies, test computations and actual computations demonstrated that the general-purpose NURBS mesh generation method makes the IGA use in fluid mechanics computations even more practical.

For more on the general-purpose NURBS mesh generation method, see [17, 86]. In the computations presented here, the method used in the aorta flow analysis.

1.11 Outline of the Remaining Sections

We provide the governing equations in Section 2. The ST-VMS and ST-SI are described in Section 3, and the ALE-VMS and IMGA-VMS in Section 4. In Section 5 we provide some brief comments on the parallel computations. In Sections 6 and 7, as examples of space–time computations, we present an aortic-valve flow analysis and a patient-specific aorta flow analysis. In Section 8, as an example of IMGA computations, we present a patient-specific heart valve design and analysis. The concluding remarks are given in Section 9.

2 Governing Equations

2.1 Incompressible Flow

Let $\Omega_t \subset \mathbb{R}^{n_{sd}}$ be the spatial domain with boundary Γ_t at time $t \in (0,T)$, where n_{sd} is the number of space dimensions. The subscript t indicates the time-dependence of the domain. The Navier–Stokes equations of incompressible flows are written on Ω_t and $\forall t \in (0,T)$ as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u} - \mathbf{f}\right) - \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} = \mathbf{0}, \tag{1}$$

$$\boldsymbol{\nabla}\cdot\mathbf{u} = 0, \tag{2}$$

where ρ, $\mathbf{u}$ and $\mathbf{f}$ are the density, velocity and body force. The stress tensor $\boldsymbol{\sigma}(\mathbf{u}, p) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u})$, where p is the pressure, $\mathbf{I}$ is the identity tensor, $\mu = \rho\nu$ is the viscosity, ν is the kinematic viscosity, and the strain rate $\boldsymbol{\varepsilon}(\mathbf{u}) = \left(\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^T\right)/2$. The essential and natural boundary conditions for Eq. (1) are represented as $\mathbf{u} = \mathbf{g}$ on $(\Gamma_t)_\mathrm{g}$ and $\mathbf{n}\cdot\boldsymbol{\sigma} = \mathbf{h}$ on $(\Gamma_t)_\mathrm{h}$, where $\mathbf{n}$ is the unit normal vector and $\mathbf{g}$ and $\mathbf{h}$ are given functions. A divergence-free velocity field $\mathbf{u}_0(\mathbf{x})$ is specified as the initial condition.

2.2 Structural Mechanics

In this article we will not provide any of our formulations requiring fluid and structure definitions simultaneously; we will instead give reference to earlier journal articles where the formulations were presented. Therefore, for notation simplicity, we will reuse many of the symbols used in the fluid mechanics equations to represent their counterparts in the structural mechanics equations. To begin with, $\Omega_t \subset \mathbb{R}^{n_\mathrm{sd}}$ and Γ_t will represent the structure domain and its boundary. The structural mechanics equations are then written, on Ω_t and $\forall t \in (0,T)$, as

$$\rho\left(\frac{\mathrm{d}^2\mathbf{y}}{\mathrm{d}t^2} - \mathbf{f}\right) - \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} = \mathbf{0}, \tag{3}$$

where $\mathbf{y}$ and $\boldsymbol{\sigma}$ are the displacement and Cauchy stress tensor. The essential and natural boundary conditions for Eq. (3) are represented as $\mathbf{y} = \mathbf{g}$ on $(\Gamma_t)_\mathrm{g}$ and $\mathbf{n}\cdot\boldsymbol{\sigma} = \mathbf{h}$ on $(\Gamma_t)_\mathrm{h}$. The Cauchy stress tensor can be obtained from

$$\boldsymbol{\sigma} = J^{-1}\mathbf{F}\mathbf{S}\mathbf{F}^T, \tag{4}$$

where $\mathbf{F}$ and J are the deformation gradient tensor and its determinant, and $\mathbf{S}$ is the second Piola–Kirchhoff stress tensor. It is obtained from the strain-energy density function φ as follows:

$$\mathbf{S} \equiv \frac{\partial\varphi}{\partial\mathbf{E}}, \tag{5}$$

where $\mathbf{E}$ is the Green–Lagrange strain tensor:

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{C} - \mathbf{I}\right), \tag{6}$$

and $\mathbf{C}$ is the Cauchy–Green deformation tensor:

$$\mathbf{C} \equiv \mathbf{F}^T \cdot \mathbf{F}. \tag{7}$$

From Eqs. (5) and (6),

$$\mathbf{S} = 2\frac{\partial \varphi}{\partial \mathbf{C}}. \tag{8}$$

2.3 Fluid–Structure Interface

In an FSI problem, at the fluid–structure interface, we will have the velocity and stress compatibility conditions between the fluid and structure parts. The details on those conditions can be found in Section 5.1 of [75].

3 ST-VMS and ST-SI

We include from [13, 81] the ST-VMS and ST-SI methods.

The ST-VMS is given as

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}Q + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&- \int_{(P_n)_\mathrm{h}} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}P + \int_{Q_n} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \mathrm{d}Q + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{\tau_\mathrm{SUPS}}{\rho} \left[\rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{w}^h \right) + \boldsymbol{\nabla} q^h \right] \cdot \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \nu_\mathrm{LSIC} \boldsymbol{\nabla} \cdot \mathbf{w}^h \rho r_\mathrm{C}(\mathbf{u}^h) \mathrm{d}Q \\
&- \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \tau_\mathrm{SUPS} \mathbf{w}^h \cdot \left(\mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \cdot \boldsymbol{\nabla}\mathbf{u}^h \right) \mathrm{d}Q \\
&- \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{\tau_\mathrm{SUPS}^2}{\rho} \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \cdot \left(\boldsymbol{\nabla}\mathbf{w}^h \right) \cdot \mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) \mathrm{d}Q = 0,
\end{aligned} \tag{9}$$

where

$$\mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) = \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{u}^h - \mathbf{f}^h \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\mathbf{u}^h, p^h), \tag{10}$$

$$r_{\mathrm{C}}(\mathbf{u}^h) = \boldsymbol{\nabla} \cdot \mathbf{u}^h \tag{11}$$

are the residuals of the momentum equation and incompressibility constraint. The test functions associated with the velocity and pressure are $\mathbf{w}$ and q. A superscript "h" indicates that the function is coming from a finite-dimensional space. The symbol Q_n represents the ST slice between time levels n and $n+1$, $(P_n)_{\mathrm{h}}$ is the part of the lateral boundary of that slice associated with the traction boundary condition $\mathbf{h}$, and Ω_n is the spatial domain at time level n. The superscript "e" is the ST element counter, and n_{el} is the number of ST elements. The functions are discontinuous in time at each time level, and the superscripts "$-$" and "$+$" indicate the values of the functions just below and just above the time level. See [76, 77, 79, 33, 13] for the definitions used here for the stabilization parameters τ_{SUPS} and ν_{LSIC}. For more ways of calculating the stabilization parameters in finite element computation of flow problems, see [78, 80, 37]).

Remark 1 *The ST-SUPS method can be obtained from the ST-VMS method by dropping the eighth and ninth integrations.*

In the ST-SI, labels "Side A" and "Side B" represent the two sides of the SI. We add boundary terms to Eq. (9). The boundary terms are first added separately for the two sides, using test functions $\mathbf{w}_{\mathrm{A}}^h$ and q_{A}^h and $\mathbf{w}_{\mathrm{B}}^h$ and q_{B}^h. Putting them together, the complete set of terms added becomes

$$\begin{aligned}
&-\int_{(P_n)_{\mathrm{SI}}} \left(q_{\mathrm{B}}^h \mathbf{n}_{\mathrm{B}} - q_{\mathrm{A}}^h \mathbf{n}_{\mathrm{A}}\right) \cdot \frac{1}{2}\left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h\right) \mathrm{d}P \\
&-\int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{B}}^h \cdot \frac{1}{2}\left(\left(\mathcal{F}_{\mathrm{B}}^h - \left|\mathcal{F}_{\mathrm{B}}^h\right|\right)\mathbf{u}_{\mathrm{B}}^h - \left(\mathcal{F}_{\mathrm{B}}^h - \left|\mathcal{F}_{\mathrm{B}}^h\right|\right)\mathbf{u}_{\mathrm{A}}^h\right) \mathrm{d}P \\
&-\int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{A}}^h \cdot \frac{1}{2}\left(\left(\mathcal{F}_{\mathrm{A}}^h - \left|\mathcal{F}_{\mathrm{A}}^h\right|\right)\mathbf{u}_{\mathrm{A}}^h - \left(\mathcal{F}_{\mathrm{A}}^h - \left|\mathcal{F}_{\mathrm{A}}^h\right|\right)\mathbf{u}_{\mathrm{B}}^h\right) \mathrm{d}P \\
&+\int_{(P_n)_{\mathrm{SI}}} \left(\mathbf{n}_{\mathrm{B}} \cdot \mathbf{w}_{\mathrm{B}}^h + \mathbf{n}_{\mathrm{A}} \cdot \mathbf{w}_{\mathrm{A}}^h\right) \frac{1}{2}\left(p_{\mathrm{B}}^h + p_{\mathrm{A}}^h\right) \mathrm{d}P \\
&-\int_{(P_n)_{\mathrm{SI}}} \left(\mathbf{w}_{\mathrm{B}}^h - \mathbf{w}_{\mathrm{A}}^h\right) \cdot \left(\hat{\mathbf{n}}_{\mathrm{B}} \cdot \mu\left(\boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{B}}^h) + \boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{A}}^h)\right)\right) \mathrm{d}P \\
&-\gamma_{\mathrm{ACI}} \int_{(P_n)_{\mathrm{SI}}} \hat{\mathbf{n}}_{\mathrm{B}} \cdot \mu\left(\boldsymbol{\varepsilon}\left(\mathbf{w}_{\mathrm{B}}^h\right) + \boldsymbol{\varepsilon}\left(\mathbf{w}_{\mathrm{A}}^h\right)\right) \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h\right) \mathrm{d}P \\
&+\int_{(P_n)_{\mathrm{SI}}} \frac{\mu C}{h} \left(\mathbf{w}_{\mathrm{B}}^h - \mathbf{w}_{\mathrm{A}}^h\right) \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h\right) \mathrm{d}P,
\end{aligned} \tag{12}$$

where

$$\mathcal{F}_{\mathrm{B}}^h = \mathbf{n}_{\mathrm{B}} \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{v}_{\mathrm{B}}^h\right), \tag{13}$$

$$\mathcal{F}_{\mathrm{A}}^h = \mathbf{n}_{\mathrm{A}} \cdot \left(\mathbf{u}_{\mathrm{A}}^h - \mathbf{v}_{\mathrm{A}}^h\right), \tag{14}$$

$$h = \frac{h_{\mathrm{B}} + h_{\mathrm{A}}}{2}, \tag{15}$$

$$h_{\mathrm{B}} = 2\left(\sum_{\alpha=1}^{n_{\mathrm{ent}}}\sum_{a=1}^{n_{\mathrm{ens}}}\left|\mathbf{n}_{\mathrm{B}}\cdot\boldsymbol{\nabla}N_a^{\alpha}\right|\right)^{-1} \text{ (for Side B)}, \tag{16}$$

$$h_{\mathrm{A}} = 2\left(\sum_{\alpha=1}^{n_{\mathrm{ent}}}\sum_{a=1}^{n_{\mathrm{ens}}}\left|\mathbf{n}_{\mathrm{A}}\cdot\boldsymbol{\nabla}N_a^{\alpha}\right|\right)^{-1} \text{ (for Side A)}, \tag{17}$$

$$\hat{\mathbf{n}}_{\mathrm{B}} = \frac{\mathbf{n}_{\mathrm{B}} - \mathbf{n}_{\mathrm{A}}}{\|\mathbf{n}_{\mathrm{B}} - \mathbf{n}_{\mathrm{A}}\|}. \tag{18}$$

Here, $(P_n)_{\mathrm{SI}}$ is the SI in the ST domain, $\mathbf{v}$ is the mesh velocity, n_{ens} and n_{ent} are the number of spatial and temporal element nodes, N_a^{α} is the basis function associated with spatial and temporal nodes a and α, $\gamma_{\mathrm{ACI}} = 1$, and C is a nondimensional constant. For our element length definition, we typically set $C = 1$.

A number of remarks were provided in [13] to explain the added terms and to comment on related interpretations. We refer the reader interested in those details to [13].

Remark 2 *A coefficient γ_{ACI} was added in [81] to the sixth integration so that we have the option of using $\gamma_{\mathrm{ACI}} = -1$. This option was added, in [40], also in the context of compressible flows. Using $\gamma_{\mathrm{ACI}} = 1$ in a discontinuous Galerkin method was introduced in the symmetric interior penalty Galerkin method [87], and using $\gamma_{\mathrm{ACI}} = -1$ was introduced in the nonsymmetric interior penalty Galerkin method [88]. Stabilized methods based on both $\gamma_{\mathrm{ACI}} = 1$ and -1 were reported in [18] in the context of the advection–diffusion equation. In the computations reported in this article, we set $\gamma_{\mathrm{ACI}} = 1$.*

4 ALE-VMS and ALE-IMGA-VMS

The ALE-VMS formulation is posed on a spatial domain Ω that is discretized into elements Ω^e. While $\{\Omega^e\}$, Ω, and its boundary Γ are time-dependent, when there is no risk of confusion, we drop the subscript t to simplify notation. The superscript h indicates association with discrete function spaces defined over Ω, which moves with the velocity $\hat{\mathbf{u}}^h$, which is the same as the mesh velocity $\mathbf{v}^h$ in Section 3. The semi-discrete formulation is given as

$$\begin{aligned}
&\int_{\Omega}\mathbf{w}^h\cdot\rho\left(\left.\frac{\partial\mathbf{u}^h}{\partial t}\right|_{\hat{\mathbf{x}}} + (\mathbf{u}^h-\hat{\mathbf{u}}^h)\cdot\nabla\mathbf{u}^h - \mathbf{f}^h\right)\mathrm{d}\Omega + \int_{\Omega}\boldsymbol{\varepsilon}(\mathbf{w}^h):\boldsymbol{\sigma}(\mathbf{u}^h,p^h)\,\mathrm{d}\Omega\\
&-\int_{\Gamma}\mathbf{w}^h\cdot\mathbf{h}^h\mathrm{d}\Gamma + \int_{\Omega}q^h\nabla\cdot\mathbf{u}^h\,\mathrm{d}\Omega\\
&-\beta\int_{\Gamma}\mathbf{w}^h\cdot\rho\left\{\left(\mathbf{u}^h-\hat{\mathbf{u}}^h\right)\cdot\mathbf{n}\right\}_{-}\mathbf{u}^h\mathrm{d}\Gamma\\
&+\sum_e\int_{\Omega^e}\tau_{\mathrm{SUPS}}\left((\mathbf{u}^h-\hat{\mathbf{u}}^h)\cdot\nabla\mathbf{w}^h + \frac{1}{\rho}\nabla q^h\right)\cdot\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h,p^h)\,\mathrm{d}\Omega
\end{aligned}$$

$$
\begin{aligned}
&+\sum_{e}\int_{\Omega^e} \nu_{\mathrm{LSIC}} \boldsymbol{\nabla}\cdot\mathbf{w}^h \rho r_{\mathrm{C}}(\mathbf{u}^h)\,\mathrm{d}\Omega \\
&-\sum_{e}\int_{\Omega^e} \tau_{\mathrm{SUPS}}\mathbf{w}^h\cdot\left(\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h,p^h)\cdot\boldsymbol{\nabla}\mathbf{u}^h\right)\mathrm{d}\Omega \\
&-\sum_{e}\int_{\Omega^e} \frac{\tau_{\mathrm{SUPS}}^2}{\rho}\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h,p^h)\cdot\left(\boldsymbol{\nabla}\mathbf{w}^h\right)\cdot\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h,p^h)\,\mathrm{d}\Omega \\
&+\sum_{e}\int_{\Omega^e} \left(\tau_{\mathrm{SUPS}}\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h,p^h)\cdot\nabla\mathbf{w}^h\right)\overline{\tau}\cdot\left(\tau_{\mathrm{SUPS}}\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h,p^h)\cdot\nabla\mathbf{u}^h\right)\mathrm{d}\Omega = 0\,, \qquad (19)
\end{aligned}
$$

where $\left.\frac{\partial(\cdot)}{\partial t}\right|_{\hat{\mathbf{x}}}$ is the time derivative taken with respect to the fixed reference coordinates $\hat{\mathbf{x}}$ of the spatial configuration, β (≥ 0) is associated with the backflow stabilization (see Remark 4), and $\{\cdot\}_-$ isolates the negative part of its argument. The additional stabilization parameter $\overline{\tau}$ is defined as

$$
\overline{\tau} = \left(\tau_{\mathrm{SUPS}}\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h,p^h)\cdot(\mathbf{G})\cdot\tau_{\mathrm{SUPS}}\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h,p^h)\right)^{-1/2}\,, \qquad (20)
$$

where $\mathbf{G}$ generalizes element size to physical elements mapped through $\mathbf{x}(\boldsymbol{\xi})$ from a parametric parent element: $G_{ij} = \xi_{k,i}\xi_{k,j}$.

The ALE-VMS formulation can be combined with the immersogeometric analysis (IMGA) [56], which we refer to as the ALE-IMGA-VMS method [9, 55, 74]. In the IMGA problem, the kinematic and traction compatibility conditions at the immersed fluid–structure interface are imposed weakly using the DAL. The details of this method can be found in [56, 68].

Remark 3 *To improve mass conservation of the ALE-IMGA-VMS technique near immersed boundaries, the following modification to* τ_{SUPS} *is introduced in [56]:*

$$
\tau_{\mathrm{SUPS}} = \left(s\left(\frac{4}{\Delta t^2}+(\mathbf{u}^h-\hat{\mathbf{u}}^h)\cdot\mathbf{G}(\mathbf{u}^h-\hat{\mathbf{u}}^h)+C_I\left(\frac{\mu}{\rho}\right)^2\mathbf{G}:\mathbf{G}\right)\right)^{-1/2}. \qquad (21)
$$

Almost everywhere in Ω *we set* $s = 1$*, which yields a traditional definition of* τ_{SUPS}*. However, in an* $O(h)$ *neighborhood of the immersed fluid–structure interface we set* $s \geq 1$*, which effectively reduces the size of* τ_{SUPS} *in that region. A theoretical motivation for this scaling is given in [72], and a numerical investigation of its effect is given in [73].*

Remark 4 *Unsteady flow computations may sometimes diverge due to significant inflow through the Neumann boundary* Γ_f^h*; this is known as backflow divergence and is frequently encountered in cardiovascular simulations. In order to preclude backflow divergence, a backflow stabilization method (the* β *term in Eq.* (19)*) originally proposed in [89] and further studied in [90] is employed in our ALE-VMS and ALE-IMGA-VMS formulations.*

Remark 5 *The $\overline{\tau}$ term of Eq.* (19) *is not derived from VMS analysis; it is an additional residual-based stabilization term that is included to provided extra stabilizing dissipation near steep solution gradients while maintaining consistency with the exact solution. It was introduced in [1] and bears resemblance to the DCDD [76] and YZβ [91, 92] discontinuity-capturing methods.*

5 Parallel Computations

Parallel computations with space–time methods go as far back as 1992 [93], with the 3D computations reported as early as 1993 [94]. All computations reported in this chapter were carried out on parallel computing platforms. The number of cores used in a typical computation ranges from 96 to 576. Because the computations were mostly for the purpose of testing a new computational method, parallel efficiency was not a high priority. Still the efficiencies we see are high enough to justify the use of the maximum number of cores available in the computer resources we have.

6 ST Computation: Aortic-Valve Flow Analysis

This section is from [10].

6.1 Geometry and Leafleat Motion

We have a typical aortic-valve model, such as the one in [30]. The model, shown in Figure 1, has three leaflets and one main outlet, corresponding to the beginning of the aorta. The leaflet motion is prescribed. They move in an asymmetric fashion. We identify the individual leaflets as shown in Figure 2. The leaflet positions are defined by means of a pseudo-time parameter θ, with the values 0 and 1 corresponding to the fully open and fully closed positions. The prescribed motion is given through θ as shown in Figure 3.

6.2 Mesh, Flow Conditions and Computational Conditions

We create the mesh with five SIs, with three of them connecting the mesh sectors containing the leaflets in the valve region of the aorta (see Figure 4). The other two SIs, which are the top and bottom circular planes in Figure 4, connect the meshes in the inlet and outlet regions to the valve region. They are for independent meshing in the inlet and outlet regions. The volume mesh is made of quadratic NURBS

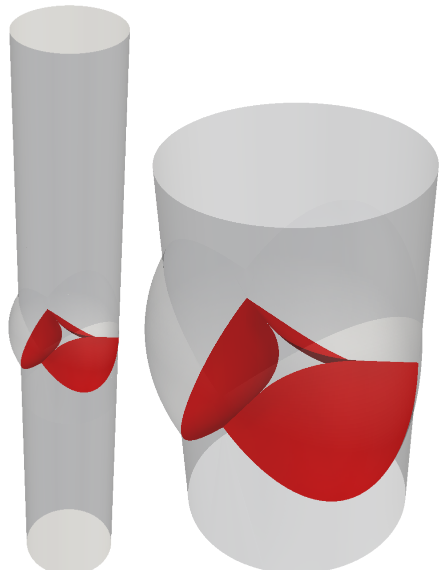

Fig. 1 Aortic-valve flow analysis. Model geometry. Aorta, leaflets, and sinuses. The left picture shows the entire computational domain, and the right picture is the zoomed view of the valve.

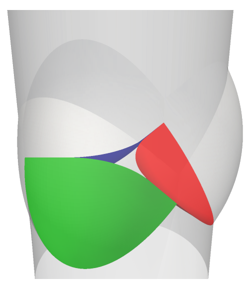

Fig. 2 Aortic-valve flow analysis. Leaflet identification. Leaflet 1 (red), 2 (green) and 3 (blue).

elements. The number of control points is 84,534, and the number of elements is 54,000. We prescribe the motion of the interior control points, and specify in each domain the master–slave mapping for all leaflet positions. Figure 5 shows a set of selected NURBS elements to illustrate how elements collapse.

The density and kinematic viscosity of the blood are 1,050 $\mathrm{kg/m^3}$ and $4.2{\times}10^{-6}$ $\mathrm{m^2/s}$. The boundary conditions are no-slip on the arterial walls and the leaflets, traction-free at the outflow boundary, and uniform velocity at the inflow boundary, with a temporal profile as shown in Figure 6. The cycle period is 0.712 s. The no-slip condition on the arterial walls is enforced weakly.

We use the ST-SUPS method. The time-step size is $4.00{\times}10^{-3}$ s. There are three nonlinear iterations at each time step. The number of GMRES iterations per nonlinear iteration is 300.

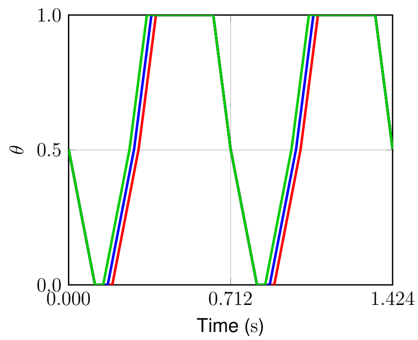

Fig. 3 Aortic-valve flow analysis. Leaflet motion. Pseudo-time parameter θ as a function of time for each of the three leaflets.

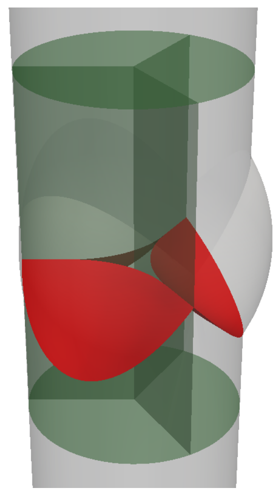

Fig. 4 Aortic-valve flow analysis. Aortic valve and the five SIs.

6.3 Results

Figure 7 shows the isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude. The viewing angle is as we see the leaflets in Figure 2. We have a biased flow jet due to the asymmetric leaflet closing. This can be seen from the third, fourth and fifth pair of pictures in Figure 7. We also report the the wall shear stress (WSS) on the leaflet surfaces. The viewing angle is as we see the leaflets in Figure 8. Figure 9 shows the magnitude of the WSS on the upper and lower surfaces of the leaflets.

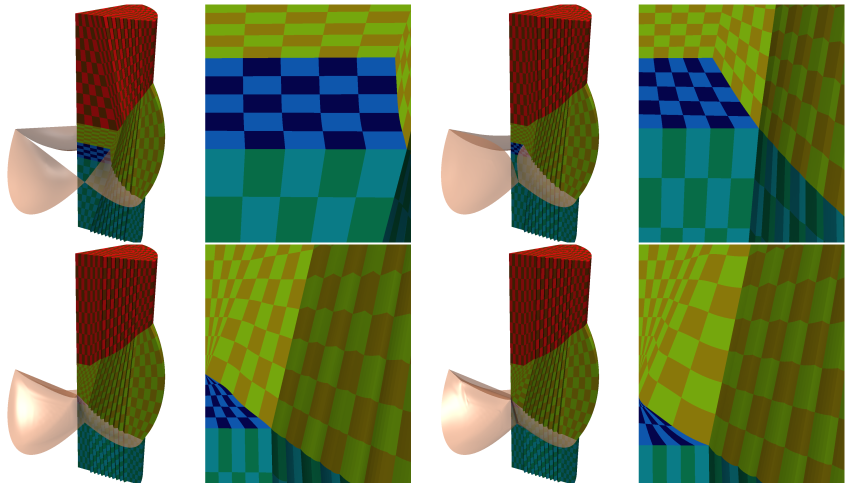

Fig. 5 Aortic-valve flow analysis. A set of selected NURBS elements, from when the valve is fully open (*top-left*) to when it is fully closed (*bottom-right*). The corresponding θ values are 0.0, 0.42, 0.97, and 1.0. The *right* pictures are the zoomed views around the leaflet.

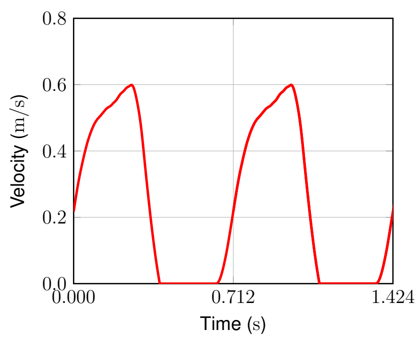

Fig. 6 Aortic-valve flow analysis. Inflow velocity (two cycles).

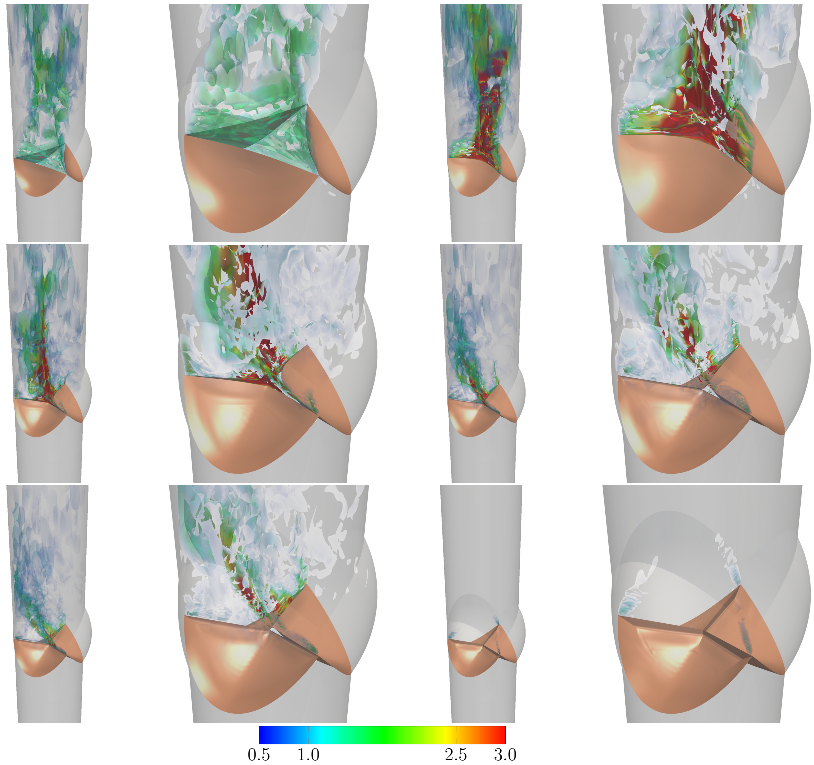

Fig. 7 Aortic-valve flow analysis. Isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude (m/s). The frames are for t = 0.804, 0.984, 1.028, 1.072, 1.080, and 1.252 s.

Fig. 8 Aortic-valve flow analysis. Viewing angle for reporting the WSS. The leaflet identification is same as in Figure 2.

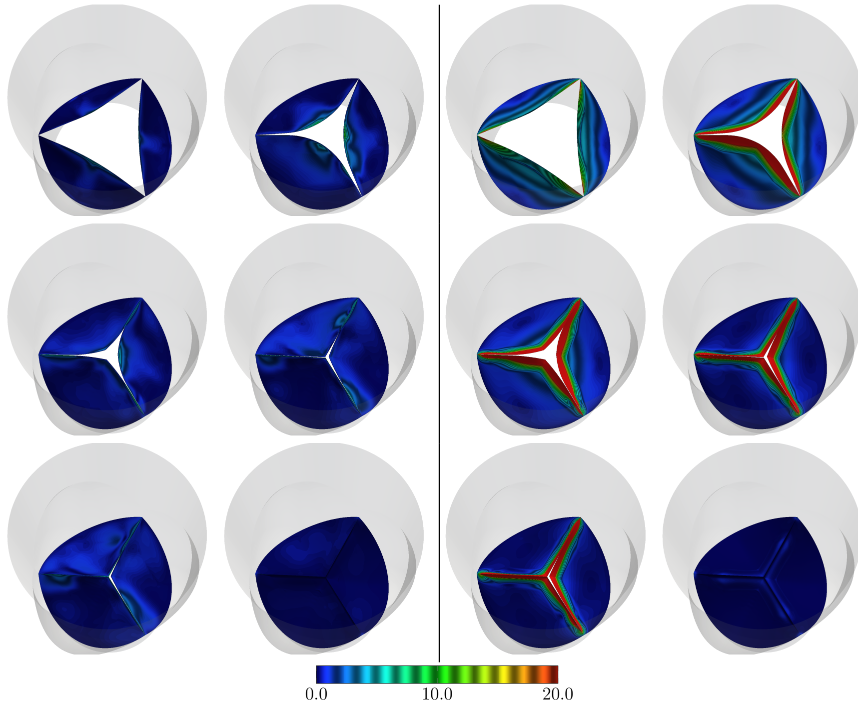

Fig. 9 Aortic-valve flow analysis. Magnitude of the WSS (Pa). Upper surface (*left*) and lower surface (*right*). The frames are for t = 0.804, 0.984, 1.028, 1.072, 1.080, and 1.252 s.

7 ST Computation: Patient-Specific Aorta Flow Analysis

This section is from [8].

We start with a geometry obtained from medical images and then use cubic T-splines to represent the surface. The density and kinematic viscosity of the blood are 1,050 $\mathrm{kg/m^3}$ and 4.2×10^{-6} $\mathrm{m^2/s}$.

7.1 Conditions

The computational domain and boundary conditions are shown in Figure 10. The

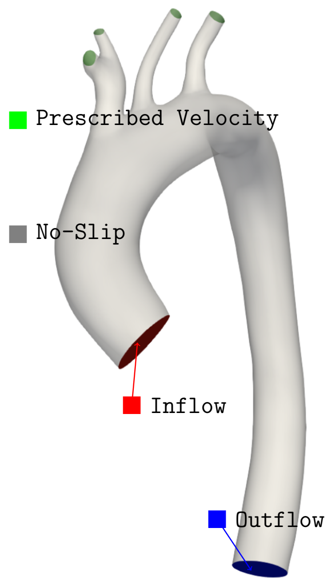

Fig. 10 Patient-specific aorta flow analysis. Geometry and boundary conditions.

diameters are given in Table 1. The inflow flow rate, plug flow, is in Figure 11. The

Table 1 Patient-specific aorta flow analysis. Diameter (mm) of the inlet and outlets. The outlets are listed in the order of closeness to the inlet.

	Inlet	Outlet 1	Outlet 2	Outlet 3	Outlet 4	Outlet 5
Diameter	25.6	5.81	3.90	4.41	6.43	19.9

peak value of the average inflow velocity is 0.709 m/s. We estimate the outflows as

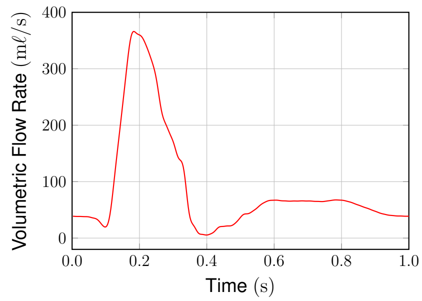

Fig. 11 Patient-specific aorta flow analysis. Volumetric flow rate at the inlet.

distributed by Murray's law [95]:

$$Q_o \propto D_o^3, \tag{22}$$

where Q_o is the volumetric outflow rate, and the outlet diameter D_o is defined based on the outlet area A_o:

$$D_o = 2\sqrt{\frac{A_o}{\pi}}. \tag{23}$$

We form a plug flow profile at the smaller outlets, and the main outlet is set to traction free.

7.2 Mesh

We create a quadratic NURBS mesh from the T-spline surface, using the technique introduced in [17, 86]. Figure 12 shows one of the NURBS patches and five of the patches together to illustrate the block-structured nature of the NURBS mesh. The function space has only C^0 continuity between the patches. Figure 13 shows the base mesh. Figure 14 shows the base and refined meshes at the inlet. The meshes are refined by knot insertion, therefore the geometry is unchanged, and the basis functions for the coarser meshes are subsets of the basis functions for the finer meshes. The refinement is in the normal direction, and at each refinement, the element thickness is halved in half of the most refined layers. For the base mesh, the element thickness in the normal direction is approximately 1 % of the local diameter. There is no refinement in the tangential directions. During the refinement, the original plug flow profiles of the base mesh are retained. Table 2 shows the number of elements and control points.

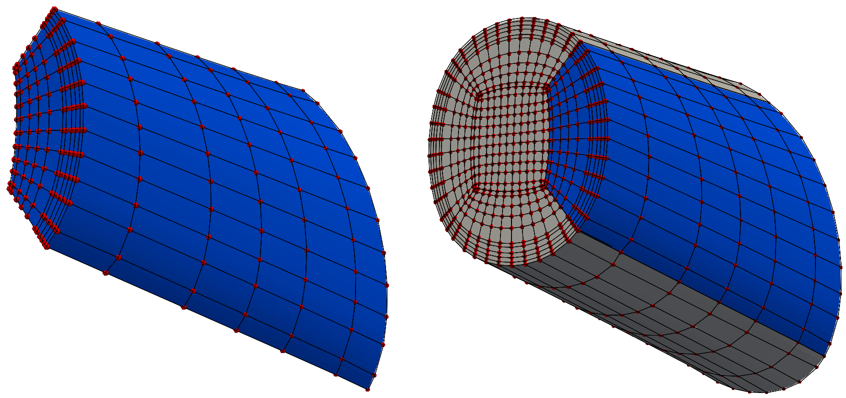

Fig. 12 Patient-specific aorta flow analysis. NURBS control mesh. One of the patches (*top*) and five of the patches together (*bottom*).

Table 2 Patient-specific aorta flow analysis. Number of control points (*nc*) and element (*ne*) for the quadratic NURBS meshes used in the computations.

	nc	*ne*
Base Mesh	202,497	151,513
Refinement Mesh 1	266,437	205,733
Refinement Mesh 2	330,377	259,953
Refinement Mesh 3	394,317	314,173
Refinement Mesh 4	458,257	368,393

7.3 Mesh Refinement Study

We compute with the 5 meshes in Table 2. The time-step sizes are $\Delta t = 0.0025$ s for Base Mesh and Refinement Mesh 1 and 2, and $\Delta t = 0.00125$ s for Refinement Mesh 3 and 4. The number of nonlinear iterations per time step is 3, and the number of GMRES iterations per nonlinear iteration is 800 for Base Mesh and Refinement Mesh 1, and 1,200, 1,400, and 1,600 for Refinement Mesh 2, 3, and 4, respectively. The ST-SUPS method is used and the stabilization parameters are those given by Eqs. (2.4)–(2.6), (2.8) and (2.10) in [13].

We first compute 9 cycles with Base Mesh, and the initial condition for the refined meshes is obtained by knot insertion. The solution reported here is for the 10th cycle. Figure 15 shows the solution computed with Refinement Mesh 4. At the peak flow rate a complex flow pattern is formed, and the vortex structure breaks down into smaller structures during the deceleration. The magnitude of the WSS ($\mathbf{h}_{\mathrm{v}}$) at the peak flow rate is shown for each mesh in Figure 16. Qualitatively, all results are in good agreement, and the convergence can be seen with refinement. To quantify the mesh refinement level, we calculate the y^+ value for the first-element thickness h as

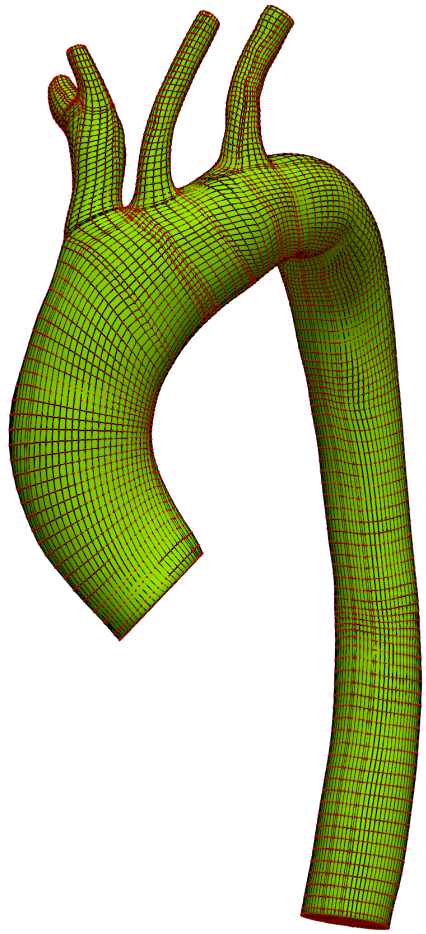

Fig. 13 Patient-specific aorta flow analysis. Base Mesh. Control mesh and surface (*green*). *Red* points are control points.

$$y^+ = \frac{u^* h}{\nu}, \tag{24}$$

where the friction velocity u^* is based on the computed value of the WSS as follows:

$$u^* = \sqrt{\frac{\left\|\mathbf{h}_{\mathrm{v}}^h\right\|}{\rho}}. \tag{25}$$

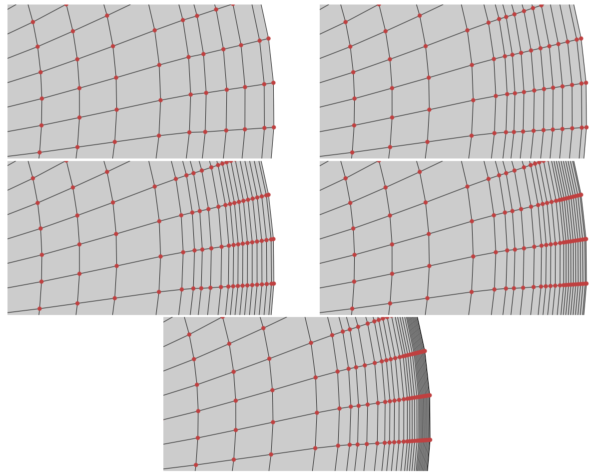

Fig. 14 Patient-specific aorta flow analysis. Control mesh at the inlet. Base Mesh, Refinement Mesh 1, Refinement Mesh 2, Refinement Mesh 3, and Refinement Mesh 4.

Figure 17 shows the spatial distribution of y^+ at the peak flow rate. It shows that for the meshes used here, y^+ range is from approximate maximum 10 to less than 1. Comparing Figures 16 and 17, we see that the WSS values computed over different meshes are in agreement where $y^+ \leq 1$.

The time-averaged WSS magnitude (TAWSS) is shown in Figure 18, and Figure 19 shows the spatially-averaged WSS magnitude in a cycle. Figure 20 shows the oscillatory shear index (OSI), defined as

$$\mathrm{OSI} = \frac{1}{2}\left(1 - \frac{\left\| \int_0^T \mathbf{h}_{\mathrm{v}}^h \mathrm{d}t \right\|}{\int_0^T \left\| \mathbf{h}_{\mathrm{v}}^h \right\| \mathrm{d}t}\right). \tag{26}$$

Overall for OSI, even Base Mesh is in a good agreement with others. However, if we compare details such as branches, we see some difference even where y^+ value is small. To see the flow differences, using the solution from Refinement Mesh 4 as the reference solution, we inspect the velocity difference $\left\|\mathbf{u}_k^h - \mathbf{u}_4^h\right\|$, where the subscripts indicate Base Mesh and Refinement Mesh k.

Remark 6 *To calculate the velocity difference, all meshes and corresponding solutions are refined by using the knot-insertion technique, and the control variables are obtained based on Refinement Mesh 4. The visualization is done after taking the difference between the control variables, interpolating the vector, and taking its magnitude.*

The spatial average of the difference is maximum at around 0.5 s. This indicates that the vortex breakdown, due to the small-scale flow behavior that needs to be dealt with, would not be easy to resolve. Figure 21 shows the velocity difference at 0.5 s.

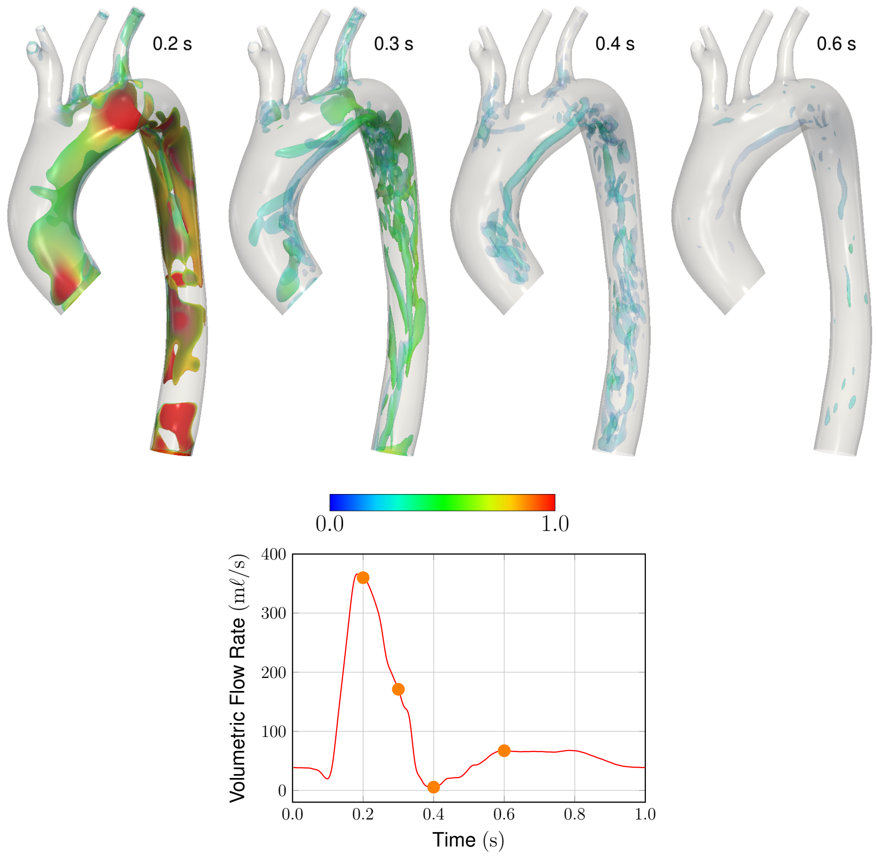

Fig. 15 Patient-specific aorta flow analysis. Mesh refinement study. Computed with Refinement Mesh 4. Isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude (m/s) (*top*). The time instants are shown with circles (*bottom*).

In summary, good accuracy in the WSS magnitude can be obtained with locally good representation, and the OSI requires a good flow representation overall, including the vortex breakdown.

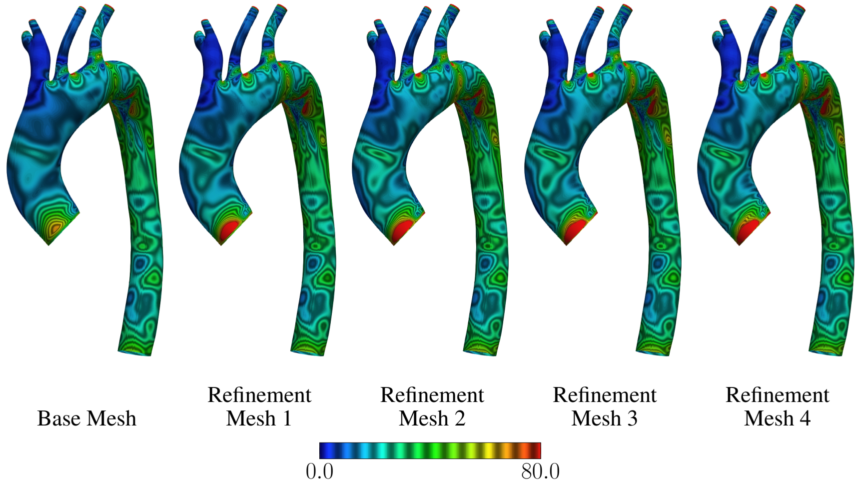

Fig. 16 Patient-specific aorta flow analysis. Mesh refinement study. WSS (dyn/cm^2) at the peak flow rate.

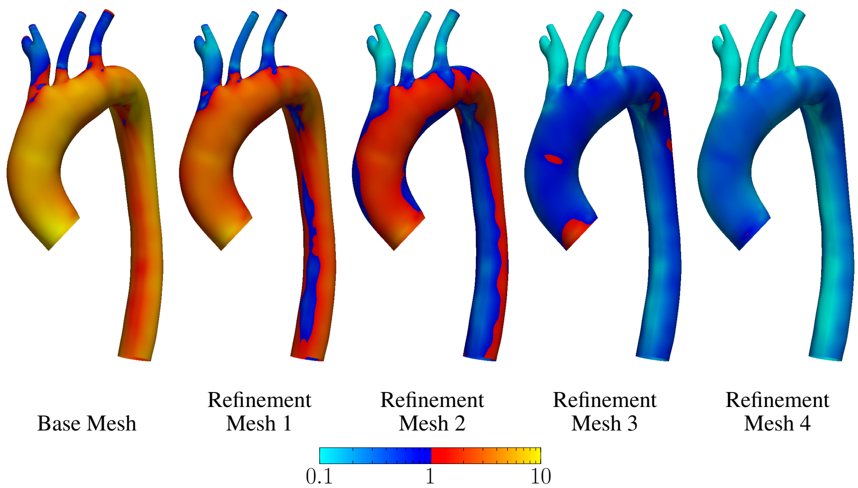

Fig. 17 Patient-specific aorta flow analysis. Mesh refinement study. y^+ value for the first-element thickness, based on the WSS computed at the peak flow rate.

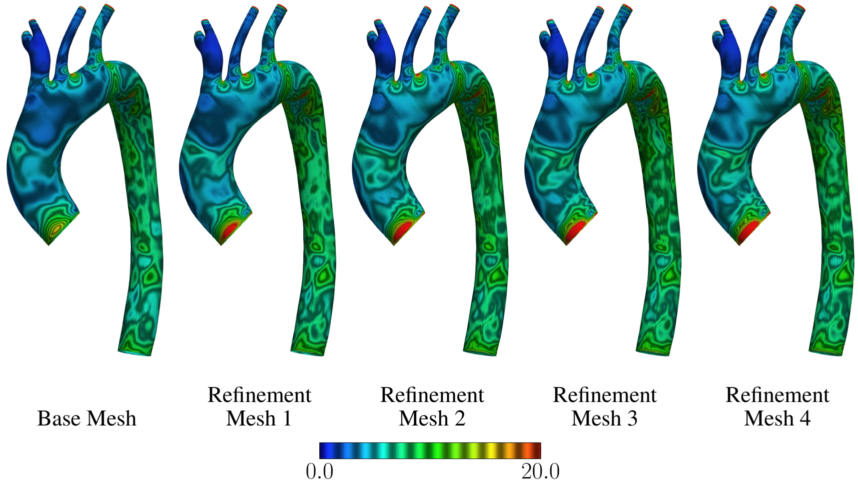

Fig. 18 Patient-specific aorta flow analysis. Mesh refinement study. TAWSS (dyn/cm^2).

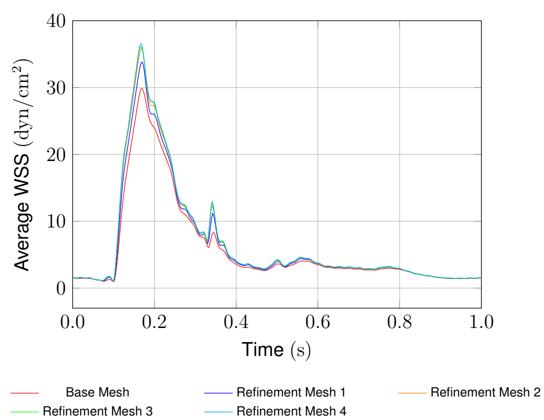

Fig. 19 Patient-specific aorta flow analysis. Mesh refinement study. Spatially-averaged WSS during a cycle.

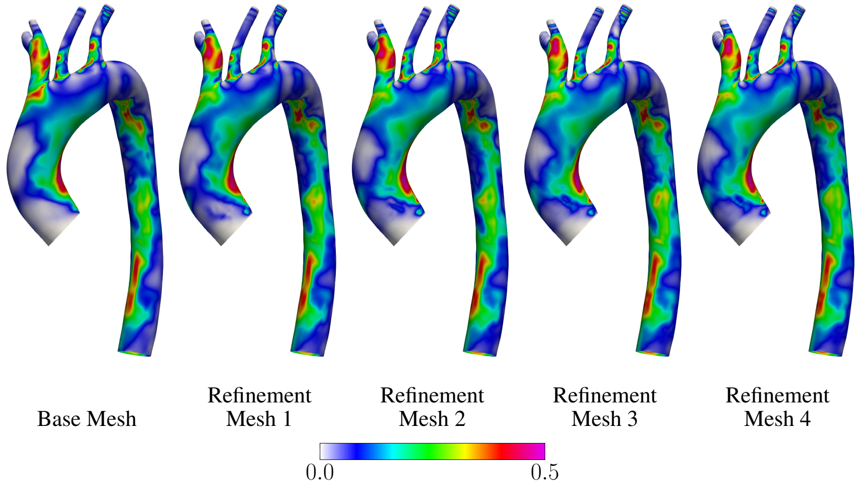

Fig. 20 Patient-specific aorta flow analysis. Mesh refinement study. OSI.

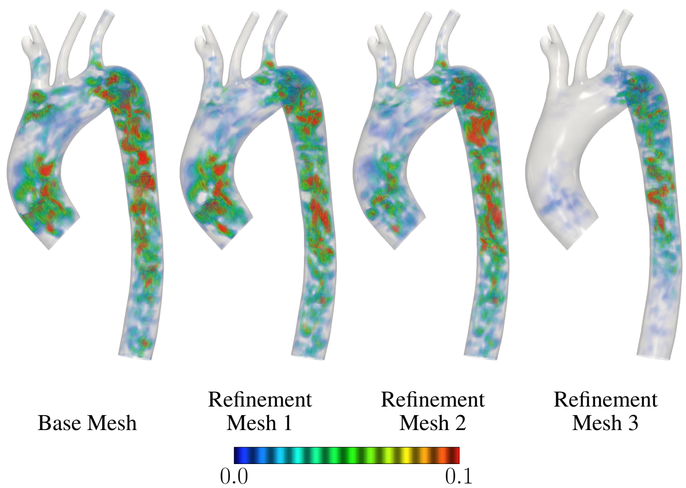

Fig. 21 Patient-specific aorta flow analysis. Mesh refinement study. Velocity difference $\left|\left|\mathbf{u}_k^h - \mathbf{u}_4^h\right|\right|$ (m/s) at 0.5 s, where the subscripts indicate Base Mesh and Refinement Mesh k.

8 IMGA Computation: Patient-Specific Heart Valve Design and Analysis

This section is from [74], where more details can be found.

Here we present a novel framework for designing personalized prosthetic heart valves using IMGA-VMS. We parameterize the leaflet geometry using several key design parameters. This allows for generating various perturbations of the leaflet design for the patient-specific aortic root reconstructed from the medical image data. Each design is analyzed using the IMGA-VMS FSI methodology, which allows us to efficiently simulate the coupling of the deforming aortic root, the parametrically designed prosthetic valves, and the surrounding blood flow under physiological conditions. A parametric study is carried out to investigate the influence of the geometry on heart valve performance, indicated by the effective orifice area (EOA) and the coaptation area (CA). Finally, the FSI simulation results of a design that reasonably well balances the EOA and CA are presented.

8.1 Trivariate NURBS Parameterization of the Ascending Aorta

To obtain a volumetric parameterization of the artery and lumen, we first construct a trivariate multi-patch NURBS in a regular shape, e.g. a tubular domain, then solve a linear elastostatic, mesh moving problem [94] for the displacement from this regular domain to a deformed configuration that represents the artery and lumen. However, solving a linear elastostatic problem to obtain the deformed interior mesh is only effective for relatively mild, translation-dominant deformations. For scenarios that involve large deformations, such as the deformation of a straight tubular domain into a curved shape of a patient-specific ascending aorta, the interior elements can become severely distorted. To avoid this, we first obtain a centerline along the axial direction of a patient-specific artery wall surface. Along this centerline, we define a number of cross sections corresponding to the control points of the NURBS artery wall surface in the axial direction. (These cross sections are shown as blue curves in Figure 22a.) At each cross section, we calculate its unit normal vector $\mathbf{n}_c$ and the effective radius r_c, which is determined such that the area of a circle calculated using this radius matches the area of the cross section. (A circle corresponding to one of the cross sections is shown in the red curve in Figure 22a.) Finally, using this information, we construct a tubular NURBS surface that has the same control-point and knot-vector topology as the target patient-specific artery wall surface, as shown in Figures 22b and 22c. Another tubular surface corresponding to the lumenal surface is also constructed, using the same cross sections but smaller effective radii coming from the lumenal NURBS surface.

These two tubular NURBS surfaces are used to construct a primitive trivariate multi-patch NURBS that includes the solid and fluid subdomains, shown in gray and red, respectively, in Figure 22d. Basis functions are made C^0-continuous at the

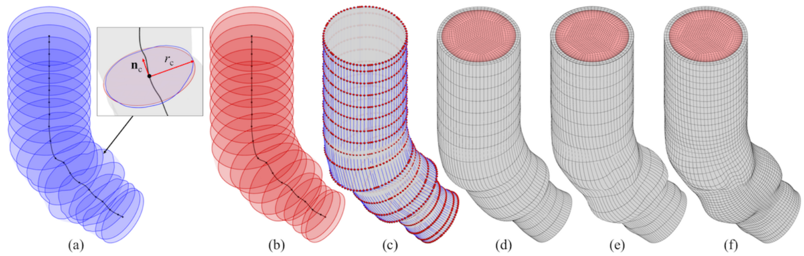

Fig. 22 The construction of the volumetric NURBS discretization of the blood and the artery wall domains. (a) Cross sections of the artery wall surface. (b) Circular cross sections. (c) NURBS tubular surface and corresponding control points. (d) Primitive volume mesh. (e) Deformed volume mesh. (f) *h*-refined volume mesh.

fluid–solid interface, so that velocity functions defined using the resulting spline space conform to standard fluid–structure kinematic constraints while retaining the ability to represent non-smooth behavior across the material interface. The resulting volumetric NURBS can then be morphed to match the patient-specific geometry with minimal rotation, so an elastostatic problem can provide an analysis-suitable parameterization. Displacements at the ends of the tube are constrained to remain within their respective cross sections. Finally, we refine the deformed trivariate NURBS for analysis purposes, by inserting knots at desired locations, such as around the sinuses and the flow boundary layers. The final volumetric NURBS discretization of the patient-specific ascending aorta is shown in Figure 22f.

8.2 Parametric BHV Design

To design effective prosthetic valves for specific patients, we focus specifically on the leaflet geometry and assume that non-leaflet components of stentless valves move with the aortic root and do not affect aortic deformation or flow. Starting from the NURBS surface of a patient-specific root, valve leaflets are parametrically designed as follows. We first pick nine “key points” located on the ends of commissure lines and the bottom of the sinuses. The positions of these points are indicated by blue spheres in Figure 23. These define how the leaflets attach to the sinuses. The key points solely depend on the geometry of the patient-specific aortic root and will remain unchanged for different valve designs. We then parameterize families of univariate B-splines defining the free edges and radial “belly curves” of the leaflets. These curves are shown in red and green in Figure 23. The attachment edges, free edges, and belly curves are then interpolated to obtain smooth bivariate B-spline representations of the leaflets.

Figure 24 shows the details of parameterizing the free-edge curve (red) and the belly-region curve (green). In Figure 24, $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ are the key points on the top of the commissure lines and $\mathbf{p}_4$ is the key point on the sinus bottom, as labeled in

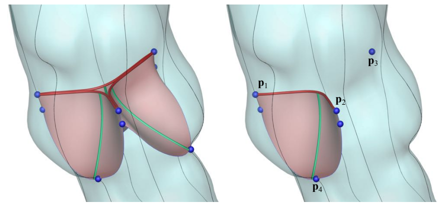

Fig. 23 The key geometric features used to parametrically control the valve designs. The blue key points define the attachment of the valve to the root. The red and green curves are parametrically controlled for valve design.

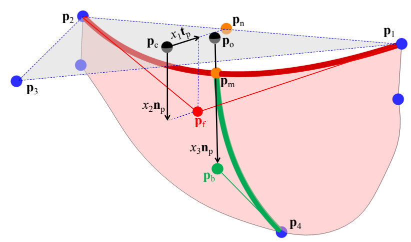

Fig. 24 The parametric control of the valve designs. The key points (blue spheres) are identical to those in the right plot of Figure 23. x_1, x_2, and x_3 control the location of $\mathbf{P}_\mathrm{f}$ and $\mathbf{P}_\mathrm{b}$ and thus control the curvature and height of the red free edge, and the curvature of the green belly curve.

Figure 23. Points $\mathbf{p}_1$-$\mathbf{p}_3$ define a triangle $\varDelta\mathbf{p}_{1-3}$, with $\mathbf{p}_\mathrm{c}$ being its geometric center. The unit vector pointing from $\mathbf{p}_\mathrm{c}$ to $\mathbf{p}_\mathrm{n}$ is denoted by $\mathbf{t}_\mathrm{p}$, and the unit normal vector of $\varDelta\mathbf{p}_{1-3}$ pointing downwards is $\mathbf{n}_\mathrm{p}$. We first construct the free edge curve as a univariate quadratic B-spline curve determined by three control points, $\mathbf{p}_1$, $\mathbf{p}_\mathrm{f}$, and $\mathbf{p}_2$. The location of $\mathbf{p}_\mathrm{f}$ is defined by $\mathbf{p}_\mathrm{f} = \mathbf{p}_\mathrm{c} + x_1\mathbf{t}_\mathrm{p} + x_2\mathbf{n}_\mathrm{p}$. By changing x_1 and x_2 to control the location of $\mathbf{p}_\mathrm{f}$, the curvature and the height of the free edge can be parametrically changed. We then take $\mathbf{p}_\mathrm{m}$ as the midpoint of the free edge, the point $\mathbf{p}_\mathrm{b}$, and the key point $\mathbf{p}_4$ to construct a univariate quadratic B-spline curve (green). The point $\mathbf{p}_\mathrm{b}$ is defined by $\mathbf{p}_\mathrm{b} = \mathbf{p}_\mathrm{o} + x_3\mathbf{n}_\mathrm{p}$, where $\mathbf{p}_\mathrm{o}$ is the projection of $\mathbf{p}_\mathrm{m}$ onto $\varDelta\mathbf{p}_{1-3}$ along the direction of $\mathbf{n}_\mathrm{p}$. Finally, the fixed attachment edges and the parametrically controlled free edge and belly curve are used to construct a cubic B-spline surface with desired parameterization. By choosing x_1, x_2 and x_3 as design variables, we can parametrically change the free edge and belly curve, and therefore change the valve design. This procedure is implemented in an interactive geometry modeling and parametric design platform [96].

8.3 Application to BHV Design

To determine an effective BHV design, we first need to identify quantitative measures of its performance. We focus on two quantities of clinical interest: to measure the systolic performance, we evaluate the EOA, which indicates how well the valve permits flow in the forward direction. For a quantitative evaluation of the diastolic performance, we measure the CA, which indicates how well the valve seals and prevents flow in the reverse direction. In this section, we study the impact of the design variables x_1, x_2, and x_3 on our two quantities of interest.

Constitutive parameters in the governing equations are held constant over the design space. Fluid, solid, and shell structure mass densities are set to 1.0 g/cm^3. The parameters of the Fung-type material model for the shell structure are $c_0 = 2.0\times10^6$ dyn/cm^2, $c_1 = 2.0\times10^5$ dyn/cm^2, and $c_2 = 100$. The thickness of the leaflet is set to 0.0386 cm. The bulk and shear modulii for the arterial wall are selected to give a Young's modulus of 10^7 dyn/cm^2 and Poisson's ratio of 0.45 in the small strain limit. The inlet and outlet cross sections are free to slide in their tangential planes and deform radially, but constrained not to move in the orthogonal directions [97]. Mass-proportional damping with constant $C_{\text{damp}} = 10^4$ Hz is used to model the interaction of the artery with the surrounding tissue. The dynamic viscosity of the blood is set to $\mu_{\text{f}} = 3\times10^{-2}$ g/(cm s).

We apply a physiologically-realistic left ventricular pressure time history as a traction boundary condition at the inflow. The applied pressure signal is periodic, with a period of 0.86 s for one cardiac cycle. The traction $-(p_0 + RQ)\mathbf{n}_{\text{f}}$ is applied at the outflow for the resistance boundary condition, where p_0 is a constant physiological pressure level, $R > 0$ is a resistance coefficient, and Q is the volumetric flow rate through the outflow. In the present computation, we set $p_0 = 80$ mmHg and $R = 200$ (dyn s)/cm^5. These values ensure a realistic transvalvular pressure difference of 80 mmHg across a closed valve when $Q = 0$, while permitting a flow rate within the normal physiological range and consistent with the flow rate estimated from the medical data (about 310 ml/s) during systole. A time step size of $\Delta t = 10^{-4}$ s is used in all simulations. To obtain the artery wall tissue prestress, we apply the highest left ventricular pressure during systole (127 mmHg at $t = 0.25$ s) on the inlet and a resistance boundary condition ($p_0 = 80$ mmHg and $R = 200$ (dyn s)/cm^5) on the outlet for the calculation of $\tilde{\mathbf{h}}_{\text{f}}$ in the prestress problem [54].

We perform FSI simulations of each of $(x_1, x_2, x_3) \in (\{0.05, 0.25, 0.45\}$ cm, $\{0.1, 0.3, 0.5\}$ cm, $\{0.5, 0.8, 1.1, 1.4\}$ cm), then calculate the EOA at peak systole and the maximum CA occurring during ventricular diastole. The simulation results and quantities of interest for each case are reported in [74]. An ideal valve would have both a large EOA and a large CA. However, these two quantities tend to compete with each other: valves that close easily can be more difficult to open and vice versa. In general, the results show that increasing x_1, which corresponds to decreasing the length of the free edge, decreases EOA and CA at the same time. Increasing x_2, which decreases the height of the free edge, may increase EOA slightly but reduces CA significantly. The reduction of CA due to Increasing x_2 reduces CA and causes many designs cannot seal completely. Increasing x_3, which increases the surface curvature in the leaflet belly region, improves CA but decreases EOA. Finally, the combination of $x_1 = 0.05$ cm, $x_2 = 0.1$ or 0.3 cm, and $x_3 = 0.5$ or 0.8 cm reliably yields a high EOA between 3.92 and 4.05 cm^2, near the upper end of the physiological range of 3.0–4.0 cm^2 in healthy adults, and a CA between 3.49 and 4.54 cm^2.

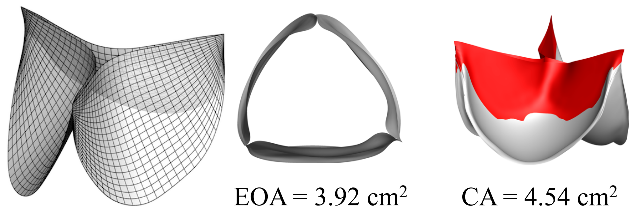

Fig. 25 The best-performing prosthetic valve design and its EOA and CA from the FSI simulation.

Among these four cases, $\mathbf{x}^* = (x_1, x_2, x_3) = (0.05\text{ cm}, 0.1\text{ cm}, 0.8\text{ cm})$, which has a CA of 4.54 cm^2 and EOA of 3.92 cm^2, strikes the best compromise between EOA and CA. The valve geometry of this best-performing design and its EOA and CA from the FSI simulation are shown in Figure 25.

Figure 26 shows several snapshots of the valve deformation and the details of the flow field at several points during the cardiac cycle. The color indicates the fluid velocity magnitude. The visualizations clearly show the instantaneous valve response to the left ventricular pressure. The valve opens with the rising left ventricular pressure in early systole (0.0–0.20 s), and then stays fully open near peak systole (0.25–0.27 s), allowing sufficient blood flow to enter the ascending aorta. A quick valve closure is then observed in early diastole (0.32–0.38 s). This quick closure of the valve minimizes the reverse flow into the left ventricle, as the left ventricular pressure drops rapidly in this period. After that, the valve properly seals, and the flow reaches a near-hydrostatic state (0.65 s). These features observed during the cardiac cycle characterize a well functioning valve within the objectives considered in this study: a large EOA during systole and a proper CA during diastole. In Figure 27, the models are superposed in the configurations corresponding to the fully-open and fully-closed phases for better visualization of the leaflet–wall coupling results. The deformation of the attachment edges can be clearly seen. The expansion and contraction of the arterial wall, as well as its sliding motion between systole and diastole can also be observed. The maximum in-plane principal Green–Lagrange strain (MIPE) evaluated on the aortic side of the leaflet is shown in Figure 28. The figure shows that during opening the strain is concentrated in the belly region of the leaflet, while during closing the highest strain happens near the valve commissure.

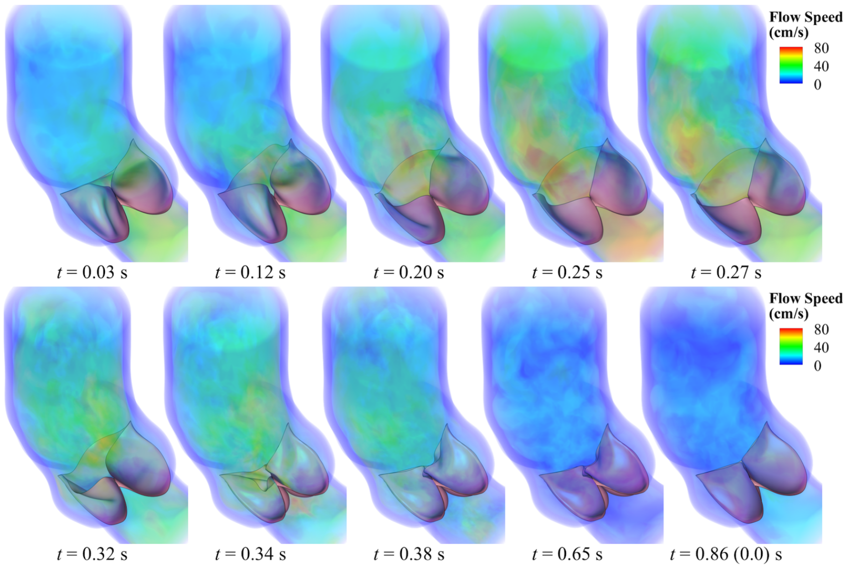

Fig. 26 Volume rendering visualization of the velocity field from our FSI simulation at several points during a cardiac cycle.

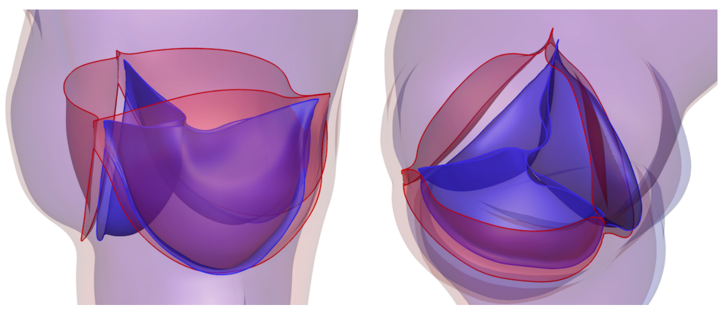

Fig. 27 Relative displacement between fully-open (red) and fully-closed (blue) configurations, showing the effect of leaflet–wall coupling. The deformation of the attachment edges can be clearly seen. The expansion and contraction of the arterial wall as well as its sliding motion between systole (red) and diastole (blue) can also be observed.

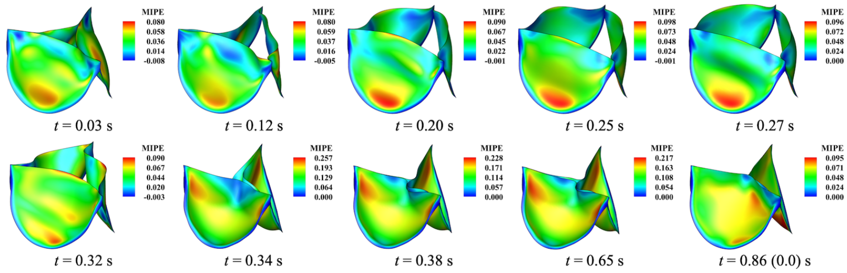

Fig. 28 Deformed valve configuration, colored by the maximum in-plane principal Green–Lagrange strain (MIPE) evaluated on the aortic side of the leaflet. Note the different scale for each time instant.

9 Concluding Remarks

In this chapter we have reviewed various technologies that have been developed by us and our colleagues and used to solve general classes of problems in Computational Cardiovascular Analysis, with focus herein on aortic flows and patient-specific and bioprosthetic heart-valve FSI. Our work on these problems, and in other more general areas of engineering, science and medicine, is based on Stabilized Methods and the Variational Multiscale Method (VMS), which have enjoyed enormous attention in the research literature and are used widely in industry and national laboratories. Stabilized Methods and the Variational Multiscale Method are at the center of development of core technologies such as Space–Time VMS, Arbitrary Lagrangian–Eulerian VMS, and Immersogeometric VMS, which we emphasized herein. They are in turn enhanced by many other special technologies that are used to deal with specific features of the applications, many of which we also described.

Computational Cardiovascular Analysis is now used routinely in medical device design, diagnosis of cardiovascular disease, surgical planning, virtual stent placement, and numerous other areas. It is only part of the more general field of Computational Medicine, which is rapidly growing. Just as the capacity of the underlying computational methods described in this article depend on the growing power of computers, Computational Medicine depends upon the increasing fidelity of medical imaging technologies and devices. Like computers, these are also advancing rapidly, which portends a bright future for the further development of Computational Medicine and its enormous potential impact on health and the human condition.

Acknowledgement

This work was supported (second author) in part by JST-CREST; Grant-in-Aid for Scientific Research (S) 26220002 from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT); Grant-in-Aid for Scientific Research (A) 18H04100 from Japan Society for the Promotion of Science; and Rice–Waseda research agreement. The mathematical model and computational method parts of the work were also supported (fourth author) in part by ARO Grant W911NF-17-1-0046, ARO DURIP Grant W911NF-18-1-0234, and Top Global University Project of Waseda University. The third author was partially supported by NSF Grant 1854436, and the fifth author was partially supported by NIH/NHLBI Grants R01HL129077 and R01HL142504.

References

1. C.A. Taylor, T.J.R. Hughes, and C.K. Zarins, “Finite element modeling of blood flow in arteries”, *Computer Methods in Applied Mechanics and Engineering*, **158** (1998) 155–196.

2. 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** (1982) 199–259.
3. C. Johnson, U. Navert, and J. Pitkäranta, “Finite element methods for linear hyperbolic problems”, *Computer Methods in Applied Mechanics and Engineering*, **45** (1984) 285–312.
4. T.J.R. Hughes, “Multiscale phenomena: Green’s functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods”, *Computer Methods in Applied Mechanics and Engineering*, **127** (1995) 387–401.
5. T.J.R. Hughes, G. Feijóo., L. Mazzei, and J.B. Quincy, “The variational multiscale method–A paradigm for computational mechanics”, *Computer Methods in Applied Mechanics and Engineering*, **166** (1998) 3–24.
6. T.J.R. Hughes and G. Sangalli, “Variational multiscale analysis: the fine-scale Green’s function, projection, optimization, localization, and stabilized methods”, *SIAM Journal of Numerical Analysis*, **45** (2007) 539–557.
7. Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi, “Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows”, *Computer Methods in Applied Mechanics and Engineering*, **197** (2007) 173–201.
8. K. Takizawa, T.E. Tezduyar, H. Uchikawa, T. Terahara, T. Sasaki, and A. Yoshida, “Mesh refinement influence and cardiac-cycle flow periodicity in aorta flow analysis with isogeometric discretization”, *Computers & Fluids*, **179** (2019) 790–798, doi:10.1016/j.compfluid.2018.05.025.
9. M.-C. Hsu, D. Kamensky, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes, “Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation”, *Computational Mechanics*, **54** (2014) 1055–1071, doi:10.1007/s00466-014-1059-4.
10. K. Takizawa, T.E. Tezduyar, T. Terahara, and T. Sasaki, “Heart valve flow computation with the integrated Space–Time VMS, Slip Interface, Topology Change and Isogeometric Discretization methods”, *Computers & Fluids*, **158** (2017) 176–188, doi:10.1016/j.compfluid.2016.11.012.
11. K. Takizawa and T.E. Tezduyar, “Multiscale space–time fluid–structure interaction techniques”, *Computational Mechanics*, **48** (2011) 247–267, doi:10.1007/s00466-011-0571-z.
12. Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang, “Isogeometric fluid–structure interaction: theory, algorithms, and computations”, *Computational Mechanics*, **43** (2008) 3–37.
13. K. Takizawa, T.E. Tezduyar, H. Mochizuki, H. Hattori, S. Mei, L. Pan, and K. Montel, “Space–time VMS method for flow computations with slip interfaces (ST-SI)”, *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2377–2406, doi:10.1142/S0218202515400126.
14. K. Takizawa, T.E. Tezduyar, A. Buscher, and S. Asada, “Space–time interface-tracking with topology change (ST-TC)”, *Computational Mechanics*, **54** (2014) 955–971, doi:10.1007/s00466-013-0935-7.
15. K. Takizawa, B. Henicke, A. Puntel, T. Spielman, and T.E. Tezduyar, “Space–time computational techniques for the aerodynamics of flapping wings”, *Journal of Applied Mechanics*, **79** (2012) 010903, doi:10.1115/1.4005073.
16. K. Takizawa, T.E. Tezduyar, Y. Otoguro, T. Terahara, T. Kuraishi, and H. Hattori, “Turbocharger flow computations with the Space–Time Isogeometric Analysis (ST-IGA)”, *Computers & Fluids*, **142** (2017) 15–20, doi:10.1016/j.compfluid.2016.02.021.
17. Y. Otoguro, K. Takizawa, and T.E. Tezduyar, “Space–time VMS computational flow analysis with isogeometric discretization and a general-purpose NURBS mesh generation method”, *Computers & Fluids*, **158** (2017) 189–200, doi:10.1016/j.compfluid.2017.04.017.
18. Y. Bazilevs and T.J.R. Hughes, “Weak imposition of Dirichlet boundary conditions in fluid mechanics”, *Computers and Fluids*, **36** (2007) 12–26.
19. Y. Bazilevs and T.J.R. Hughes, “NURBS-based isogeometric analysis for the computation of flows about rotating components”, *Computational Mechanics*, **43** (2008) 143–150.
20. M.E. Moghadam, Y. Bazilevs, T.-Y. Hsia, I.E. Vignon-Clementel, A.L. Marsden, and M. 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** (2011) 277–291, doi:10.1007/s00466-011-0599-0.

21. T.E. Tezduyar, “Stabilized finite element formulations for incompressible flow computations”, *Advances in Applied Mechanics*, **28** (1992) 1–44, doi:10.1016/S0065-2156(08)70153-4.
22. T.E. Tezduyar and K. Takizawa, “Space–time computations in practical engineering applications: A summary of the 25-year history”, *Computational Mechanics*, **63** (2019) 747–753, doi:10.1007/s00466-018-1620-7.
23. K. Takizawa and T.E. Tezduyar, “Computational methods for parachute fluid–structure interactions”, *Archives of Computational Methods in Engineering*, **19** (2012) 125–169, doi:10.1007/s11831-012-9070-4.
24. K. Takizawa, D. Montes, M. Fritze, S. McIntyre, J. Boben, and T.E. Tezduyar, “Methods for FSI modeling of spacecraft parachute dynamics and cover separation”, *Mathematical Models and Methods in Applied Sciences*, **23** (2013) 307–338, doi:10.1142/S0218202513400058.
25. K. Takizawa, T.E. Tezduyar, and R. Kolesar, “FSI modeling of the Orion spacecraft drogue parachutes”, *Computational Mechanics*, **55** (2015) 1167–1179, doi:10.1007/s00466-014-1108-z.
26. K. Takizawa, B. Henicke, D. Montes, T.E. Tezduyar, M.-C. Hsu, and Y. Bazilevs, “Numerical-performance studies for the stabilized space–time computation of wind-turbine rotor aerodynamics”, *Computational Mechanics*, **48** (2011) 647–657, doi:10.1007/s00466-011-0614-5.
27. K. Takizawa, “Computational engineering analysis with the new-generation space–time methods”, *Computational Mechanics*, **54** (2014) 193–211, doi:10.1007/s00466-014-0999-z.
28. K. Takizawa, B. Henicke, A. Puntel, N. Kostov, and T.E. Tezduyar, “Space–time techniques for computational aerodynamics modeling of flapping wings of an actual locust”, *Computational Mechanics*, **50** (2012) 743–760, doi:10.1007/s00466-012-0759-x.
29. K. Takizawa, T.E. Tezduyar, and N. Kostov, “Sequentially-coupled space–time FSI analysis of bio-inspired flapping-wing aerodynamics of an MAV”, *Computational Mechanics*, **54** (2014) 213–233, doi:10.1007/s00466-014-0980-x.
30. K. Takizawa, T.E. Tezduyar, and A. Buscher, “Space–time computational analysis of MAV flapping-wing aerodynamics with wing clapping”, *Computational Mechanics*, **55** (2015) 1131–1141, doi:10.1007/s00466-014-1095-0.
31. K. Takizawa, Y. Bazilevs, T.E. Tezduyar, M.-C. Hsu, O. Øiseth, K.M. Mathisen, N. Kostov, and S. McIntyre, “Engineering analysis and design with ALE-VMS and space–time methods”, *Archives of Computational Methods in Engineering*, **21** (2014) 481–508, doi:10.1007/s11831-014-9113-0.
32. K. Takizawa, K. Schjodt, A. Puntel, N. Kostov, and T.E. Tezduyar, “Patient-specific computational analysis of the influence of a stent on the unsteady flow in cerebral aneurysms”, *Computational Mechanics*, **51** (2013) 1061–1073, doi:10.1007/s00466-012-0790-y.
33. K. Takizawa, T.E. Tezduyar, and T. Kuraishi, “Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires”, *Mathematical Models and Methods in Applied Sciences*, **25** (2015) 2227–2255, doi:10.1142/S0218202515400072.
34. K. Takizawa, T.E. Tezduyar, T. Kuraishi, S. Tabata, and H. Takagi, “Computational thermo-fluid analysis of a disk brake”, *Computational Mechanics*, **57** (2016) 965–977, doi:10.1007/s00466-016-1272-4.
35. T. Kanai, K. Takizawa, T.E. Tezduyar, K. Komiya, M. Kaneko, K. Hirota, M. Nohmi, T. Tsuneda, M. Kawai, and M. Isono, “Methods for computation of flow-driven string dynamics in a pump and residence time”, *Mathematical Models and Methods in Applied Sciences*, **29** (2019) 839–870, doi:10.1142/S021820251941001X.
36. Y. Otoguro, K. Takizawa, T.E. Tezduyar, K. Nagaoka, and S. Mei, “Turbocharger turbine and exhaust manifold flow computation with the Space–Time Variational Multiscale Method and Isogeometric Analysis”, *Computers & Fluids*, **179** (2019) 764–776, doi:10.1016/j.compfluid.2018.05.019.
37. T. Kuraishi, K. Takizawa, and T.E. Tezduyar, “Tire aerodynamics with actual tire geometry, road contact and tire deformation”, *Computational Mechanics*, **63** (2019) 1165–1185, doi:10.1007/s00466-018-1642-1.
38. T. Kuraishi, K. Takizawa, and T.E. Tezduyar, “Space–Time Isogeometric flow analysis with built-in Reynolds-equation limit”, *Mathematical Models and Methods in Applied Sciences*, **29** (2019) 871–904, doi:10.1142/S0218202519410021.

39. K. Takizawa, T.E. Tezduyar, and T. Terahara, “Ram-air parachute structural and fluid mechanics computations with the space–time isogeometric analysis (ST-IGA)”, *Computers & Fluids*, **141** (2016) 191–200, doi:10.1016/j.compfluid.2016.05.027.
40. K. Takizawa, T.E. Tezduyar, and T. Kanai, “Porosity models and computational methods for compressible-flow aerodynamics of parachutes with geometric porosity”, *Mathematical Models and Methods in Applied Sciences*, **27** (2017) 771–806, doi:10.1142/S0218202517500166.
41. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, “Computation of cardiovascular fluid–structure interactions with the DSD/SST method”, in *Proceedings of the 6th World Congress on Computational Mechanics (CD-ROM)*, Beijing, China, (2004).
42. R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar, “Computer modeling of cardiovascular fluid–structure interactions with the Deforming-Spatial-Domain/Stabilized Space–Time formulation”, *Computer Methods in Applied Mechanics and Engineering*, **195** (2006) 1885–1895, doi:10.1016/j.cma.2005.05.050.
43. T.E. Tezduyar, S. Sathe, T. Cragin, B. Nanna, B.S. Conklin, J. Pausewang, and M. Schwaab, “Modeling of fluid–structure interactions with the space–time finite elements: Arterial fluid mechanics”, *International Journal for Numerical Methods in Fluids*, **54** (2007) 901–922, doi:10.1002/fld.1443.
44. T.E. Tezduyar, K. Takizawa, T. Brummer, and P.R. Chen, “Space–time fluid–structure interaction modeling of patient-specific cerebral aneurysms”, *International Journal for Numerical Methods in Biomedical Engineering*, **27** (2011) 1665–1710, doi:10.1002/cnm.1433.
45. K. Takizawa, Y. Bazilevs, T.E. Tezduyar, C.C. Long, A.L. Marsden, and K. Schjodt, “ST and ALE-VMS methods for patient-specific cardiovascular fluid mechanics modeling”, *Mathematical Models and Methods in Applied Sciences*, **24** (2014) 2437–2486, doi:10.1142/S0218202514500250.
46. K. Takizawa, R. Torii, H. Takagi, T.E. Tezduyar, and X.Y. Xu, “Coronary arterial dynamics computation with medical-image-based time-dependent anatomical models and element-based zero-stress state estimates”, *Computational Mechanics*, **54** (2014) 1047–1053, doi:10.1007/s00466-014-1049-6.
47. 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** (1981) 329–349.
48. V. Kalro and T.E. Tezduyar, “A parallel 3D computational method for fluid–structure interactions in parachute systems”, *Computer Methods in Applied Mechanics and Engineering*, **190** (2000) 321–332, doi:10.1016/S0045-7825(00)00204-8.
49. Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T.E. Tezduyar, “3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics”, *International Journal for Numerical Methods in Fluids*, **65** (2011) 207–235, doi:10.1002/fld.2400.
50. A. Korobenko, Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, “Computer modeling of wind turbines: 1. ALE-VMS and ST-VMS aerodynamic and FSI analysis”, *Archives of Computational Methods in Engineering*, to appear, DOI: 10.1007/s11831-018-9292-1, September 2018, doi:10.1007/s11831-018-9292-1.
51. J. Yan, A. Korobenko, X. Deng, and Y. Bazilevs, “Computational free-surface fluid–structure interaction with application to floating offshore wind turbines”, *Computers and Fluids*, **141** (2016) 155–174, doi:10.1016/j.compfluid.2016.03.008.
52. Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan, “FSI modeling for fatigue-damage prediction in full-scale wind-turbine blades”, *Journal of Applied Mechanics*, **83** (6) (2016) 061010.
53. Y. Bazilevs, V.M. Calo, Y. Zhang, and T.J.R. Hughes, “Isogeometric fluid–structure interaction analysis with applications to arterial blood flow”, *Computational Mechanics*, **38** (2006) 310–322.
54. M.-C. Hsu and Y. Bazilevs, “Blood vessel tissue prestress modeling for vascular fluid–structure interaction simulations”, *Finite Elements in Analysis and Design*, **47** (2011) 593–599.

55. M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M.C.H. Wu, J. Mineroff, A. Reali, Y. Bazilevs, and M.S. Sacks, "Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models", *Computational Mechanics*, **55** (2015) 1211–1225, doi:10.1007/s00466-015-1166-x.
56. D. Kamensky, M.-C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes, "An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves", *Computer Methods in Applied Mechanics and Engineering*, **284** (2015) 1005–1053.
57. I. Akkerman, J. Dunaway, J. Kvandal, J. Spinks, and Y. Bazilevs, "Toward free-surface modeling of planing vessels: simulation of the Fridsma hull using ALE-VMS", *Computational Mechanics*, **50** (2012) 719–727.
58. M.C.H. Wu, D. Kamensky, C. Wang, A.J. Herrema, F. Xu, M.S. Pigazzini, A. Verma, A.L. Marsden, Y. Bazilevs, and M.-C. Hsu, "Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: Application to a hydraulic arresting gear", *Computer Methods in Applied Mechanics and Engineering*, **316** (2017) 668–693.
59. J. Yan, X. Deng, A. Korobenko, and Y. Bazilevs, "Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines", *Computers and Fluids*, **158** (2017) 157–166, doi:10.1016/j.compfluid.2016.06.016.
60. A. Castorrini, A. Corsini, F. Rispoli, K. Takizawa, and T.E. Tezduyar, "A stabilized ALE method for computational fluid–structure interaction analysis of passive morphing in turbomachinery", *Mathematical Models and Methods in Applied Sciences*, **29** (2019) 967–994, doi:10.1142/S0218202519410057.
61. J. Yan, B. Augier, A. Korobenko, J. Czarnowski, G. Ketterman, and Y. Bazilevs, "FSI modeling of a propulsion system based on compliant hydrofoils in a tandem configuration", *Computers and Fluids*, **141** (2016) 201–211, doi:10.1016/j.compfluid.2015.07.013.
62. T.A. Helgedagsrud, I. Akkerman, Y. Bazilevs, K.M. Mathisen, and O.A. Oiseth, "Isogeometric modeling and experimental investigation of moving-domain bridge aerodynamics", *ASCE Journal of Engineering Mechanics*, Accepted for publication.
63. J. Yan, A. Korobenko, A.E. Tejada-Martinez, R. Golshan, and Y. Bazilevs, "A new variational multiscale formulation for stratified incompressible turbulent flows", *Computers & Fluids*, **158** (2017) 150–156, doi:10.1016/j.compfluid.2016.12.004.
64. T.M. van Opstal, J. Yan, C. Coley, J.A. Evans, T. Kvamsdal, and Y. Bazilevs, "Isogeometric divergence-conforming variational multiscale formulation of incompressible turbulent flows", *Computer Methods in Applied Mechanics and Engineering*, **316** (2017) 859–879, doi:10.1016/j.cma.2016.10.015.
65. F. Xu, G. Moutsanidis, D. Kamensky, M.-C. Hsu, M. Murugan, A. Ghoshal, and Y. Bazilevs, "Compressible flows on moving domains: Stabilized methods, weakly enforced essential boundary conditions, sliding interfaces, and application to gas-turbine modeling", *Computers & Fluids*, **158** (2017) 201–220, doi:10.1016/j.compfluid.2017.02.006.
66. T.E. Tezduyar, S.K. Aliabadi, M. Behr, and S. Mittal, "Massively parallel finite element simulation of compressible and incompressible flows", *Computer Methods in Applied Mechanics and Engineering*, **119** (1994) 157–177, doi:10.1016/0045-7825(94)00082-4.
67. T. Kanai, K. Takizawa, T.E. Tezduyar, T. Tanaka, and A. Hartmann, "Compressible-flow geometric-porosity modeling and spacecraft parachute computation with isogeometric discretization", *Computational Mechanics*, **63** (2019) 301–321, doi:10.1007/s00466-018-1595-4.
68. M.-C. Hsu and D. Kamensky, "Immersogeometric Analysis of Bioprosthetic Heart Valves, Using the Dynamic Augmented Lagrangian Method", in T.E. Tezduyar, editor, *Frontiers in Computational Fluid–Structure Interaction and Flow Simulation*, 167–212, Springer International Publishing, Cham, 2018.
69. T.E. Tezduyar, K. Takizawa, C. Moorman, S. Wright, and J. Christopher, "Space–time finite element computation of complex fluid–structure interactions", *International Journal for Numerical Methods in Fluids*, **64** (2010) 1201–1218, doi:10.1002/fld.2221.

70. J. Kiendl, M.-C. Hsu, M.C.H. Wu, and A. Reali, “Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials”, *Computer Methods in Applied Mechanics and Engineering*, **291** (2015) 280–303.
71. M.C.H. Wu, R. Zakerzadeh, D. Kamensky, J. Kiendl, M.S. Sacks, and M.-C. Hsu, “An anisotropic constitutive model for immersogeometric fluid–structure interaction analysis of bioprosthetic heart valves”, *Journal of Biomechanics*, **74** (2018) 23–31.
72. D. Kamensky, M.-C. Hsu, Y. Yu, J.A. Evans, M.S. Sacks, and T.J.R. Hughes, “Immersogeometric cardiovascular fluid–structure interaction analysis with divergence-conforming B-splines”, *Computer Methods in Applied Mechanics and Engineering*, **314** (2017) 408–472.
73. D. Kamensky, J.A. Evans, and M.-C. Hsu, “Stability and Conservation Properties of Collocated Constraints in Immersogeometric Fluid–Thin Structure Interaction Analysis”, *Communications in Computational Physics*, **18** (04) (2015) 1147–1180.
74. F. Xu, S. Morganti, R. Zakerzadeh, D. Kamensky, F. Auricchio, A. Reali, T.J.R. Hughes, M.S. Sacks, and M.-C. Hsu, “A framework for designing patient-specific bioprosthetic heart valves using immersogeometric fluid–structure interaction analysis”, *International Journal for Numerical Methods in Biomedical Engineering*, **34** (4) (2018) e2938.
75. Y. Bazilevs, K. Takizawa, and T.E. Tezduyar, *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, February 2013, ISBN 978-0470978771.
76. T.E. Tezduyar, “Computation of moving boundaries and interfaces and stabilization parameters”, *International Journal for Numerical Methods in Fluids*, **43** (2003) 555–575, doi:10.1002/fld.505.
77. T.E. Tezduyar and S. Sathe, “Modeling of fluid–structure interactions with the space–time finite elements: Solution techniques”, *International Journal for Numerical Methods in Fluids*, **54** (2007) 855–900, doi:10.1002/fld.1430.
78. 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** (2010) 828–840, doi:10.1016/j.cma.2009.06.019.
79. K. Takizawa, T.E. Tezduyar, S. McIntyre, N. Kostov, R. Kolesar, and C. Habluetzel, “Space–time VMS computation of wind-turbine rotor and tower aerodynamics”, *Computational Mechanics*, **53** (2014) 1–15, doi:10.1007/s00466-013-0888-x.
80. K. Takizawa, T.E. Tezduyar, and Y. Otoguro, “Stabilization and discontinuity-capturing parameters for space–time flow computations with finite element and isogeometric discretizations”, *Computational Mechanics*, **62** (2018) 1169–1186, doi:10.1007/s00466-018-1557-x.
81. K. Takizawa, T.E. Tezduyar, S. Asada, and T. Kuraishi, “Space–time method for flow computations with slip interfaces and topology changes (ST-SI-TC)”, *Computers & Fluids*, **141** (2016) 124–134, doi:10.1016/j.compfluid.2016.05.006.
82. 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** (2005) 4135–4195.
83. K. Takizawa and T.E. Tezduyar, “Space–time computation techniques with continuous representation in time (ST-C)”, *Computational Mechanics*, **53** (2014) 91–99, doi:10.1007/s00466-013-0895-y.
84. T. Sasaki, K. Takizawa, and T.E. Tezduyar, “Medical-image-based aorta modeling with zero-stress-state estimation”, *Computational Mechanics*, published online, DOI: 10.1007/s00466-019-01669-4, February 2019, doi:10.1007/s00466-019-01669-4.
85. K. Takizawa, T.E. Tezduyar, and T. Sasaki, “Isogeometric hyperelastic shell analysis with out-of-plane deformation mapping”, *Computational Mechanics*, **63** (2019) 681–700, doi:10.1007/s00466-018-1616-3.
86. Y. Otoguro, K. Takizawa, and T.E. Tezduyar, “A general-purpose NURBS mesh generation method for complex geometries”, in T.E. Tezduyar, editor, *Frontiers in Computational Fluid–Structure Interaction and Flow Simulation: Research from Lead Investigators under Forty – 2018*, Modeling and Simulation in Science, Engineering and Technology, 399–434, Springer, 2018, ISBN 978-3-319-96468-3, doi:10.1007/978-3-319-96469-0_10.

87. M.F. Wheeler, “An elliptic collocation-finite element method with interior penalties”, *SIAM Journal on Numerical Analysis*, **15** (1978) 152–161.
88. P. Houston, C. Schwab, and E. Suli, “Discontinuous hp-finite element methods for advection-diffusion reaction problems”, *SIAM Journal on Numerical Analysis*, **39** (2002) 2133–2163.
89. 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** (2009) 3534–3550.
90. M. Esmaily-Moghadam, Y. Bazilevs, T.-Y. Hsia, I.E. Vignon-Clementel, and A.L. Marsden, “A comparison of outlet boundary treatments for prevention of backflow divergence with relevance to blood flow simulations”, *Computational Mechanics*, **48** (2011) 277–291.
91. T.E. Tezduyar and M. Senga, “Stabilization and shock-capturing parameters in SUPG formulation of compressible flows”, *Computer Methods in Applied Mechanics and Engineering*, **195** (2006) 1621–1632, doi:10.1016/j.cma.2005.05.032.
92. Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, “YZβ discontinuity-capturing for advection-dominated processes with application to arterial drug delivery”, *International Journal for Numerical Methods in Fluids*, **54** (2007) 593–608, doi:10.1002/fld.1484.
93. T.E. Tezduyar, M. Behr, S. Mittal, and A.A. Johnson, “Computation of unsteady incompressible flows with the finite element methods: Space–time formulations, iterative strategies and massively parallel implementations”, in *New Methods in Transient Analysis*, PVP-Vol.246/AMD-Vol.143, ASME, New York, (1992) 7–24.
94. T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal, “Parallel finite-element computation of 3D flows”, *Computer*, **26** (10) (1993) 27–36, doi:10.1109/2.237441.
95. C.D. Murray, “The physiological principle of minimum work: I. the vascular system and the cost of blood volume”, *Proceedings of the National Academy of Sciences of the United States of America*, **12** (1926) 207–214.
96. M.-C. Hsu, C. Wang, A.J. Herrema, D. Schillinger, A. Ghoshal, and Y. Bazilevs, “An interactive geometry modeling and parametric design platform for isogeometric analysis”, *Computers and Mathematics with Applications*, **70** (2015) 1481–1500.
97. 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** (2010) 481–498.