Computational Cardiovascular Flow Analysis with the Variational Multiscale Methods

Kenji Takizawa  
*Waseda University*

Yuri Bazilevs  
*Brown University*

Tayfun E. Tezduyar  
*Rice University*

Ming-Chen Hsu  
*Iowa State University*, jmchsu@iastate.edu

Follow this and additional works at: [https://lib.dr.iastate.edu/me_pubs](https://lib.dr.iastate.edu/me_pubs)  
Part of the Analytical, Diagnostic and Therapeutic Techniques and Equipment Commons, Biomechanical Engineering Commons, and the Cardiology Commons

The complete bibliographic information for this item can be found at [https://lib.dr.iastate.edu/me_pubs/366](https://lib.dr.iastate.edu/me_pubs/366). For information on how to cite this item, please visit [http://lib.dr.iastate.edu/howtocite.html](http://lib.dr.iastate.edu/howtocite.html).
Computational Cardiovascular Flow Analysis with the Variational Multiscale Methods

Abstract
Computational cardiovascular flow analysis can provide valuable information to medical doctors in a wide range of patient-specific cases, including cerebral aneurysms, aortas and heart valves. The computational challenges faced in this class of flow analyses also have a wide range. They include 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 the blood and the cardiovascular structure. Many of these challenges have been or are being addressed by the Space–Time Variational Multiscale (ST-VMS) method, Arbitrary Lagrangian–Eulerian VMS (ALE-VMS) method, and the VMS-based Immersogeometric Analysis (IMGA-VMS), which serve as the core computational methods, and the special methods used in combination with them. We provide an overview of the core and special methods and present examples of challenging computations carried out with these methods, including aorta and heart valve flow analyses.

Keywords
Cardiovascular flow analysis, Patient-specific blood flow analysis, Aorta, Heart valve, Space–Time VMS method, ALE-VMS method, Immersogeometric Analysis

Disciplines
Analytical, Diagnostic and Therapeutic Techniques and Equipment | Biomechanical Engineering | Cardiology | Mechanical Engineering

Comments

Creative Commons License
This work is licensed under a Creative Commons Attribution 4.0 License.
Computational Cardiovascular Flow Analysis with the Variational Multiscale Methods

Kenji TAKIZAWA\textsuperscript{1,*}, Yuri BAZILEVS\textsuperscript{2}, Tayfun E. TEZDUYAR\textsuperscript{3,1}, Ming-Chen HSU\textsuperscript{4}

\textsuperscript{1}Waseda University, Tokyo, Japan
\textsuperscript{2}Brown University, Providence, Rhode Island, USA
\textsuperscript{3}Rice University, Houston, Texas, USA
\textsuperscript{4}Iowa State University, Ames, Iowa, USA

*Corresponding Author: Kenji Takizawa (email: Kenji.Takizawa@tafsm.org)
(Received: 08-May-2019; accepted: 14-May-2019; published: 30-June-2019)
DOI: http://dx.doi.org/10.25073/jaec.201932.245

Abstract. Computational cardiovascular flow analysis can provide valuable information to medical doctors in a wide range of patient-specific cases, including cerebral aneurysms, aortas and heart valves. The computational challenges faced in this class of flow analyses also have a wide range. They include 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 the blood and the cardiovascular structure. Many of these challenges have been or are being addressed by the Space-Time Variational Multiscale (ST-VMS) method, Arbitrary Lagrangian-Eulerian VMS (ALE-VMS) method, and the VMS-based Immersogeometric Analysis (IMGA-VMS), which serve as the core computational methods, and the special methods used in combination with them. We provide an overview of the core and special methods and present examples of challenging computations carried out with these methods, including aorta and heart valve flow analyses.

Keywords

Cardiovascular flow analysis, Patient-specific blood flow analysis, Aorta, Heart valve, Space–Time VMS method, ALE-VMS method, Immersogeometric Analysis

1. Introduction

Many agree that computational cardiovascular flow analysis can provide surgeons and medical doctors valuable information in a wide range of patient-specific cases, including cerebral aneurysms (see, e.g., [1]), treatment of cerebral aneurysms (see, e.g., [2]), aortas (see, e.g., [3]) and heart valves (see, e.g., [4, 5]). The computational challenges faced in this class of flow analyses also have a wide range, many quite formidable. They include highly-unsteady flows and complex cardiovascular geometries. They also include moving boundaries and interfaces (MBI), 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 (FSI) between the blood and the cardiovascular structure. There is
also a computational challenge associated with using medical-image-based arterial geometries. These geometries do not come from the zero-stress state (ZSS) of the artery. A special method is needed to estimate the ZSS required in the FSI computation. The task becomes even more challenging for arteries with complex geometries, such as the aorta.

Many of these challenges have been or are being addressed by the Space–Time Variational Multiscale (ST-VMS) method [6], Arbitrary Lagrangian–Eulerian VMS (ALE-VMS) method [7], and the VMS-based Immersogeometric Analysis (IMGA-VMS) [4,8], which serve as the core computational methods, and the special methods used in combination with them. The special methods used in combination with the ST-VMS include the ST Slip Interface (ST-SI) method [9,10], ST Topology Change (ST-TC) [11,12] method, ST Isogeometric Analysis (ST-IGA) [6,13,14], integration of these methods, a general-purpose NURBS mesh generation method for complex geometries [15,16], and methods for estimating the ZSS of the artery [17–20]. The special methods used in combination with the ALE-VMS include weak enforcement of no-slip boundary conditions [21–23], “sliding interfaces” [24,25] (the acronym “SI” will also indicate that) and backflow stabilization [26].

We will provide an overview of the core and special methods and present examples of challenging computations carried out with these methods, including aorta and heart valve flow analyses.

2. Governing equations

2.1. Incompressible flow

Let $\Omega_t \subset \mathbb{R}^{n_{ad}}$ be the spatial domain with boundary $\Gamma_t$ at time $t \in (0,T)$, where $n_{ad}$ 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 $\Omega_t$ and $\forall t \in (0,T)$ as

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

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

where $\rho$, $\mathbf{u}$ and $\mathbf{f}$ are the density, velocity, and body force. The stress tensor $\mathbf{\sigma}(\mathbf{u},p) = -p\mathbf{I} + 2\mu\mathbf{E}(\mathbf{u})$, where $p$ is the pressure, $\mathbf{I}$ is the identity tensor, $\mu = \rho \nu$ is the viscosity, $\nu$ is the kinematic viscosity, and the strain rate $\mathbf{E} = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)/2$. The essential and natural boundary conditions for Eq. (1) are represented as $\mathbf{u} = \mathbf{g}$ on $(\Gamma_t)_e$ and $\mathbf{u} \cdot \mathbf{n} = \mathbf{h}$ on $(\Gamma_t)_h$, where $\mathbf{n}$ is the unit normal vector and $\mathbf{g}$ and $\mathbf{h}$ are given functions. A divergence-free velocity field $\mathbf{u}_0(x)$ 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, $\Omega_t \subset \mathbb{R}^{n_{ad}}$ and $\Gamma_t$ will represent the structure domain and its boundary. The structural mechanics equations are then written, on $\Omega_t$ and $\forall t \in (0,T)$, as

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

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

$$\mathbf{\sigma} = J^{-1} \mathbf{F} \mathbf{S} \mathbf{F}^T, \quad (4)$$

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 $\varphi$ as follows:

$$\mathbf{S} = \frac{\partial \varphi}{\partial \mathbf{E}}, \quad (5)$$
where $E$ is the Green–Lagrange strain tensor:

$$E = \frac{1}{2} (C - I),$$

and $C$ is the Cauchy–Green deformation tensor:

$$C \equiv F^T \cdot F.$$ (7)

From Eqs. (5) and (6),

$$S = 2 \frac{\partial \varphi}{\partial \dot{C}}.$$ (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 [1].

3. ST-VMS and ST-SUPS

The Deforming-Spatial-Domain/Stabilized ST (DSD/SST) method [27–29] was introduced for computation of flows with MBI, including 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) [30] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [27] 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 [31–34]. 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 ST framework (see [6, 35]), 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 [36] for a comprehensive summary). The classes of problems include spacecraft parachute analysis for the landing-stage parachutes [1, 37–40], coverseparation parachutes [41] and the drogue parachutes [42–44], wind-turbine aerodynamics for horizontal-axis wind-turbine rotors [1, 45–47], full horizontal-axis wind-turbines [48–51] and vertical-axis wind-turbines [9, 52, 53], flapping-wing aerodynamics for an actual locust [1, 13, 54, 55], bioinspired MAVs [49, 50, 56, 57] and wing-clapping [11, 58], blood flow analysis of cerebral aneurysms [49, 59], stent-treated aneurysms [2, 59, 60], aortas [3, 61–63] and heart valves [5, 11, 12, 50, 63–65], spacecraft aerodynamics [41, 66], thermo-fluid analysis of ground vehicles and their tires [64, 67], thermo-fluid analysis of disk brakes [10], flow-driven string dynamics in turbomachinery [68–70], flow analysis of turbocharger turbines [14–16, 71], flow around tires with road contact and deformation [64, 72–74], fluid films [75], ram-air parachutes [76], and compressible-flow spacecraft parachute aerodynamics [77, 78].

The ST computational methods have a relatively long track record in arterial FSI analysis, starting with computations reported in [79–82]. These were among the earliest arterial FSI computations, and the core method was the ST-SUPS. Many ST computations were also reported in the last 15 years. In the first 8 years of that period the ST computations were for FSI of abdominal aorta [83], carotid artery [83] and cerebral aneurysms [84–90]. In the last 7 years, the ST computations focused on even more challenging aspects of cardiovascular fluid mechanics and FSI, including comparative studies of cerebral aneurysms [49, 91], stent treatment of cerebral aneurysms [2, 59, 60, 92, 93], heart valve flow computation [5, 11, 12, 50, 63, 65], aorta flow analysis [3, 61–63], and coronary arterial dynamics [94].

For more on the ST-VMS and ST-SUPS, see [1]. In the flow analyses presented here, the ST framework provides higher-order accuracy in a
general context. 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 ST framework enables high-resolution computation near the moving heart valve leaflets.

4. ALE-VMS, RBVMS and ALE-SUPS

The ALE-VMS method [1, 7, 45, 95–99] is the VMS version of the ALE [100]. It succeeded the ST-SUPS [27] and ALE-SUPS [101] and preceded the ST-VMS. The VMS components are from the RBVMS [31–34]. The ALE-VMS originated from the RBVMS formulation of incompressible turbulent flows proposed in [33] 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 [7] 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 [21–23], “sliding interfaces” [24, 25] and backflow stabilization [26].

The ALE-SUPS, RBVMS and ALE-VMS have been applied to many classes of FSI, MBI and fluid mechanics problems. The classes of problems include ram-air parachute FSI [101], wind-turbine aerodynamics and FSI [45, 51–53, 102–108], more specifically, vertical-axis wind turbines [52, 53, 109, 110], floating wind turbines [111], wind turbines in atmospheric boundary layers [52, 53, 108, 112], and fatigue damage in wind-turbine blades [113], patient-specific cardiovascular fluid mechanics and FSI [7, 114–119], biomedical-device FSI [4, 8, 120–123], ship hydrodynamics with free-surface flow and fluid-object interaction [124, 125], hydrodynamics and FSI of a hydraulic arresting gear [126, 127], hydrodynamics of tidal-stream turbines with free-surface flow [128], passive-morphing FSI in turbomachinery [129], bioinspired FSI for marine propulsion [130, 131], bridge aerodynamics and fluid-object interaction [132–134], Recent advances in stabilized and multiscale methods may be found for stratified incompressible flows in [135], for divergence-conforming discretizations of incompressible flows in [136], and for compressible flows with emphasis on gas-turbine modeling in [137].

For more information on the ALE-VMS, RB-VMS and ALE-SUPS, see [1]. In the flow analyses presented here, the VMS feature of the 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.

5. IMGA-VMS

The IMGA was introduced in [8] as a geometrically flexible technique for solving FSI problems involving large, complex structural deformation 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 [138], was introduced in [8] for weakly enforcing constraints in time-dependent immersogeometric FSI problems. A mixed ALE-VMS/IMGA-VMS method was developed in [4] in the framework of the Fluid–Solid Interface-Tracking/Interface-Capturing Technique [139]; 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 [4]. The DAL-based ALE-IMGA-VMS was integrated with Computer-Aided Design (CAD) for heart-valve analysis in [123] with a thorough comparison between dynamic and FSI computations. An anisotropic constitutive modeling of BHV leaflets for immersogeometric FSI, based
on the Kirchhoff–Love shell formulation for general hyperelastic materials [140], is proposed in [141]. A divergence-conforming formulation of incompressible flow, which gives a pointwise divergence-free velocity field everywhere in the domain, is elegantly addressing the mass loss error across the valve interface in [142]. In addition, stable coupling strategies and suitable definition of Lagrange multipliers for the DAL numerical approach were proposed and analyzed in [143–145]. Finally, the FSI framework of ALE-IMGA-VMS was employed in patient-specific valve design in [146]. The DAL-based IMGA has also been combined with surrogate modeling in [127] for an efficient and effective use of FSI to optimize the design of a hydraulic arresting gear.

6. ST-SI

The ST-SI was introduced in [9], in the context of incompressible-flow equations, to retain the desirable moving-mesh features of the ST-VMS and ST-SUPS when we have spinning solid surfaces, such as a turbine rotor. The mesh covering the spinning surface spins with it, retaining the high-resolution representation of the boundary layers. The starting point in the development of the ST-SI was the version of the ALE-VMS for computations with sliding interfaces [24, 25]. Interface terms similar to those in the ALE-VMS version are added to the ST-VMS to account for the compatibility conditions for the velocity and stress at the SI. That accurately connects the two sides of the solution. An ST-SI version where the SI is between fluid and solid domains was also presented in [9]. The SI in this case is a “fluid–solid SI” rather than a standard “fluid–fluid SI” and enables weak enforcement of the Dirichlet boundary conditions for the fluid. The ST-SI introduced in [10] 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 [9, 52, 53], thermo-fluid analysis of disk brakes [10], flow-driven string dynamics in turbomachinery [68–70], flow analysis of turbocharger turbines [14–16, 71], flow around tires with road contact and deformation [64, 72–74], fluid films [73], aerodynamic analysis of ram-air parachutes [76], and flow analysis of heart valves [5, 63, 65].

In the ST-SI version presented in [9] the SI 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 SIs are dealt with and how the Dirichlet conditions are enforced weakly with fluid–solid SIs. 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 [76]. The compressible-flow ST-SI methods were introduced in [77], including the version where the SI is between a thin porous structure and the fluid on its two sides. Compressible-flow porosity models were also introduced in [77]. These, together with the compressible-flow ST SUPG method [147], extended the ST computational analysis range to compressible-flow aerodynamics of parachutes with fabric and geometric porosities. That enabled ST computational flow analysis of the Orion spacecraft drogue parachute in the compressible-flow regime [77, 78].

For more on the ST-SI, see [9, 10]. In the computations here, the ST-SI is used in combination with the ST-TC, and we will describe the ST-SI-TC in Section 9.

7. Stabilization parameters

The ST-SUPS, ALE-SUPS, RBVMS, ALE-VMS, ST-VMS, ALE-SI and ST-SI all have some embedded stabilization parameters that play a significant role (see [1, 9]). There are many ways of defining these stabilization parameters (for examples, see [9, 28, 29, 48, 67, 74, 148–164]). 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.
8. **ST-TC**

The ST-TC \cite{11,12} was introduced for moving-mesh computation of flow problems with TC, 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 TC, if the nearness is sufficiently near for the purpose of solving the problem. Many classes of problems can be solved that way with sufficient accuracy. For examples of such computations, see the references mentioned in \cite{11}. The ST-TC made moving-mesh computations possible even when there is an actual contact between solid surfaces or other TC. By collapsing elements as needed, without changing the connectivity of the “parent” mesh, the ST-TC can handle an actual TC while maintaining high-resolution boundary layer representation near solid surfaces. This enabled successful moving-mesh computation of heart valve flows \cite{5,11,12,50,63,65}, wing clapping \cite{58}, and flow around a rotating tire with road contact and prescribed deformation \cite{64,72,74}.

For more on the ST-TC, see \cite{11,12}. In the computational analyses here, the ST-TC enables moving-mesh computation even with the TC created by the actual contact between the valve leaflets. It deals with the contact while maintaining high-resolution flow representation near the leaflet.

9. **ST-SI-TC**

The ST-SI-TC is the integration of the ST-SI and ST-TC. A fluid–fluid SI requires elements on both sides of the SI. When part of an SI needs to coincide with a solid surface, which happens for example when the solid surfaces on two sides of an SI come into contact or when an SI reaches a solid surface, the elements between the coinciding SI part and the solid surface need to collapse with the ST-TC mechanism. The collapse switches the SI from fluid–fluid SI to fluid–solid SI. With that, an SI can be a mixture of fluid–fluid and fluid–solid SIs. 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 SI 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 \cite{5,63,65} and tire aerodynamics with road contact and deformation \cite{72-74}.

For more on the ST-SI-TC, see \cite{5,72}. 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 SIs. It enables contact location change and contact sliding between the leaflets.

10. **ST-IGA**

The ST-IGA is the integration of the ST framework with isogeometric discretization. It was introduced in \cite{6}. Computations with the ST-VMS and ST-IGA were first reported in \cite{6} 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 \cite{6} 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 \cite{6,35} and demonstrated in \cite{13,54,56}. 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) \cite{13,54,56}; the name was given in \cite{48}. 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” \cite{1,6,13,35} en-
ables also specifying a constant angular velocity for invariant speeds along the circular paths. The ST 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 [10,64,67–70,165]. The STN-MUM 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 [1,13,54,55], bioinspired MAVs [49,50,66,57] and wing-clapping [11,58], separation aerodynamics of space craft [41], aerodynamics of horizontal-axis [48–51] and vertical-axis [9,52,53] wind turbines, thermo-fluid analysis of ground vehicles and their tires [64,67], thermo-fluid analysis of disk brakes [10], flow-driven string dynamics in turbomachinery [68–70], and flow analysis of turbocharger turbines [14–16,71].

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 ST computational flow analysis of turbocharger turbines [14–16,71], flow-driven string dynamics in turbomachinery [69,70], ram-air parachutes [76], spacecraft parachutes [78], aortas [3,63], heart valves [5,63,65], tires with road contact and deformation [73,74], and fluid films [75]. Using IGA basis functions in space is now a key part of some of the newest ZSS estimation methods [18–20,166] and related shell analysis [167].

For more on the ST-IGA, see [1,14,54,76]. 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.

11. ST-SI-TC-IGA

The turbocharger turbine analysis [14–16,71] and flow-driven string dynamics in turbomachinery [69,70] 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 [76] and spacecraft parachute compressible-flow analysis [78] 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 [5,63,65] was based on the integration of the ST-SI, ST-TC and ST-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 SI 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 [74], 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-motion 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 [5, 63, 65], turbocharger turbines [14–16, 71], and spacecraft parachute compressible-flow analysis [78].

For more on the ST-SI-TC-IGA, see [5, 74]. 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.

12. 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 [15]. 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 recovery of the original surface is to the extent they are suitable for accurate and robust fluid mechanics computations. 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 [16]. A test computation for a turbocharger turbine and exhaust manifold was also presented in [16], with a more detailed computation in [71]. The mesh generation method was used also in the pump-flow analysis part of the flow-driven string-dynamics presented in [70] and in the aorta flow analysis presented in [3, 63]. 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 [15, 16]. In the computations presented here, the method used in the aorta flow analysis.

13. ZSS estimation methods

The attempt to find a ZSS for the artery in the FSI computation was first made in a 2007 conference paper [168], where the concept of estimated zero-pressure (EZP) arterial geometry was introduced. The method introduced in [168] for calculating an EZP geometry was also included in a 2008 journal paper on ST arterial FSI methods [84], as “a rudimentary technique” for addressing the issue. It was pointed out in [84, 168] that quite often the medical-image-based geometries were used as arterial geometries corresponding to zero blood pressure, and that it would be more realistic to use the medical-image-based geometry as the arterial geometry corresponding to the time-averaged value of the blood pressure. Given the arterial geometry at the time-
averaged pressure value, an estimated arterial geometry corresponding to zero blood pressure needed to be built. The special methods developed to address the issue include the newer EZP versions [1, 86, 89, 90, 95] and the prestress technique introduced in [118], which was refined in [119] and presented also in [1, 95].

A method for estimation of the element-based ZSS (EBZSS) was introduced in [17] in the context of finite element discretization of the arterial wall. The method has three main components. 1. An iteration technique, which starts with a calculated ZSS initial guess, is used for computing the EBZSS such that when a given pressure load is applied, the medical-image-based target shape is matched. 2. A technique for straight-tube segments is used for computing the EBZSS so that we match the given diameter and longitudinal stretch in the target configuration and the “opening angle.” 3. An element-based mapping between the artery and straight-tube is extracted from the mapping between the artery and straight-tube segments. This provides the mapping from the arterial configuration to the straight-tube configuration, and from the estimated EBZSS of the straight-tube configuration back to the arterial configuration, to be used as the ZSS initial guess for the iteration technique that matches the medical-image-based target shape. Test computations with the method were also presented in [17] for straight-tube configurations with single and three layers, and for a curved-tube configuration with single layer. The method was used also in [94] in coronary arterial dynamics computations with medical-image-based time-dependent anatomical models.

The version of the EBZSS estimation method with isogeometric wall discretization, using NURBS basis functions, was introduced in [18]. With isogeometric discretization, we can obtain the element-based mapping directly, instead of extracting it from the mapping between the artery and straight-tube segments. Because all we need for the element-based mapping, including the curvatures, can be obtained within an element. With NURBS basis functions, we may be able to achieve a similar level of accuracy as with the linear basis functions, but using larger-size and fewer elements, and the NURBS basis functions enable representation of more complex shapes within an element. The 2D test computations with straight-tube configurations presented in [18] showed how the EBZSS estimation method with NURBS discretization works. In [166], which is an expanded, journal version of [18], how the method can be used in a 3D computation where the target geometry is coming from medical image of a human aorta was also shown.

In the method introduced in [19], the estimate is based on T-spline discretization of the arterial wall and is in the form of integration-point-based ZSS (IPBZSS). The T-spline discretization enables dealing with complex arterial geometries, such as an aorta model with branches, while retaining the desirable features of isogeometric discretization. The IPBZSS is a convenient representation of the ZSS because with isogeometric discretization, especially with T-spline discretization, specifying conditions at integration points is more straightforward than imposing conditions on control points. The method has two main components. 1. An iteration technique, which starts with a calculated ZSS initial guess, is used for computing the IPBZSS such that when a given pressure load is applied, the medical-image-based target shape is matched. 2. A design procedure, which is based on the Kirchhoff–Love shell model of the artery, is used for calculating the ZSS initial guess.

In [20], the scope and robustness of the IPBZSS were increased method by introducing a new design procedure for the ZSS initial guess. The new design procedure has two features. a) An IPB shell-like coordinate system, which increases the scope of the design to general parameterization in the computational space. b) Analytical solution of the force equilibrium in the normal direction, based on the Kirchhoff–Love shell model, which places proper constraints on the design parameters. This increases the estimation accuracy, which in turn increases the robustness of the iterations and the convergence speed. To show how the new design procedure for the ZSS initial guess performs, the computations presented in [20] included 3D test computations with a straight tube and a Y-shaped tube. A 3D computation where the target geometry is coming from medical image
of a human aorta was also presented in [20], and the aorta model included the branches.

For more on the ZSS estimation methods, see [17, 19, 20, 166]. In the computations presented here, the newest method [20] is used in the ZSS estimate of the human aorta.

14. ST computation: aortic-valve flow analysis

This section is from [5].

14.1. Geometry

We have a typical aortic-valve model, such as the one in [58]. 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.

![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.](image)

14.2. Leaflet motion

We consider two cases of leaflet motion: symmetric and asymmetric. In the symmetric case, the leaflet motions have 3-fold rotational periodicity (i.e., 3-fold discrete rotational symmetry), and in the asymmetric case, they do not. We identify the individual leaflets as shown in Figure 2. The leaflet positions are defined by means of a pseudo-time parameter $\theta$, with the values 0 and 1 corresponding to the fully open and fully closed positions. The prescribed motion is given through $\theta$ as shown in Figure 3 for the symmetric case, and in Figure 4 for the asymmetric case.

![Fig. 2: Aortic-valve flow analysis. Leaflet identification. Leaflet 1 (red), 2 (green) and 3 (blue).](image)

![Fig. 3: Aortic-valve flow analysis. Symmetric leaflet motion. Pseudo-time parameter $\theta$ as a function of time for each of the three leaflets.](image)
and bottom circular planes in Figure 5, 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 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 6 shows a set of selected NURBS elements to illustrate how elements collapse. 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 7. The cycle period is 0.712 s. The no-slip condition on the arterial walls is enforced weakly.

![Fig. 4: Aortic-valve flow analysis. Asymmetric leaflet motion. Pseudo-time parameter $\theta$ as a function of time for each of the three leaflets.](image)

![Fig. 5: Aortic-valve flow analysis. Aortic valve and the five SIs.](image)

![Fig. 6: Aortic-valve flow analysis. A set of selected NURBS elements, from when the valve is fully open (top) to when it is fully closed (bottom). The corresponding $\theta$ values are 0.0, 0.42, 0.97, and 1.0. The right pictures are the zoomed views around the leaflet.](image)

### 14.4. Computational conditions

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
The number of GMRES iterations per nonlinear iteration is 300.

14.5. Results

1) Symmetric leaflet motion

Figures 8 and 9 show 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. The result shows how effectively the flow separation is captured near the leaflet edges and the solution is obtained as the surfaces come into contact.

Remark 1. We note that there is a thin vortex sheet near the aorta wall in the second and third row of pictures in Figure 9, which correspond to the fully closed positions of the valve. That is because full blocking of the flow requires a collapsed (i.e. zero-volume) element, which in turn requires that the control points of the element coalesce, and that is not happening for the elements with two edges on the leaflet edges and two edges on the aorta wall. In the current setting, the control points associated with the two edges on the aorta wall cannot coalesce.

We also report the wall shear stress (WSS) on the leaflet surfaces. The viewing angle is as we see the leaflets in Figure 10. Figures 11 and 12 show the magnitude of the WSS on the upper and lower surfaces of the leaflets. As can be seen in the figures, the ST-SI-TC-IGA method gives a smooth WSS distribution, and the upper and lower WSS distributions are unrelated. The WSS is high on the lower surface near the edge, which is due to the flow jet.

2) Asymmetric leaflet motion

Figures 13 and 14 show 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. In this case we have a biased flow jet due to the asymmetric leaflet closing. This can be seen from the third pair of pictures in Figure 13, and the first and second pair of pictures in Figure 14.

Figures 15 and 16 show the magnitude of the WSS on the upper and lower surfaces of the
Fig. 9: Aortic-valve flow analysis. Symmetric leaflet motion. 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 $t = 1.072$, $1.080$, and $1.252$ s.

Fig. 10: Aortic-valve flow analysis. Viewing angle for reporting the WSS. The leaflet identification is same as in Figure 2.

Fig. 11: Aortic-valve flow analysis. Symmetric leaflet motion. Magnitude of the WSS (Pa). Upper surface (left) and lower surface (right). The frames are for $t = 0.804$, $0.984$, and $1.028$ s.

15. ST computation: patient-specific aorta flow analysis

This section is from [3].

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 $1,050$ kg/m$^3$ and $4.2 \times 10^{-6}$ m$^2$/s.

15.1. Conditions

The computational domain and boundary conditions are shown in Figure 17. The diameters are given in Table 1. The inflow flow rate, plug flow,
Tab. 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.

<table>
<thead>
<tr>
<th>Diameter</th>
<th>Inlet</th>
<th>Outlet 1</th>
<th>Outlet 2</th>
<th>Outlet 3</th>
<th>Outlet 4</th>
<th>Outlet 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.6</td>
<td>5.81</td>
<td>3.90</td>
<td>4.41</td>
<td>6.43</td>
<td>19.9</td>
<td></td>
</tr>
</tbody>
</table>

is in Figure 18. The peak value of the average inflow velocity is 0.709 m/s. We estimate the outflows as distributed by Murray's law [169]:

\[ Q_o \propto D_o^3, \]

(9)

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

\[ D_o = 2\sqrt[3]{\frac{A_o}{\pi}}. \]

(10)

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

15.2. Mesh

We create a quadratic NURBS mesh from the T-spline surface, using the technique introduced in [15, 16]. Figure 19 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 20 shows the base mesh. Figure 21 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.

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>nc</th>
<th>ne</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Mesh</td>
<td>202,497</td>
<td>151,513</td>
</tr>
<tr>
<td>Refinement Mesh 1</td>
<td>266,437</td>
<td>205,733</td>
</tr>
<tr>
<td>Refinement Mesh 2</td>
<td>330,877</td>
<td>259,933</td>
</tr>
<tr>
<td>Refinement Mesh 3</td>
<td>394,317</td>
<td>314,173</td>
</tr>
<tr>
<td>Refinement Mesh 4</td>
<td>458,257</td>
<td>368,393</td>
</tr>
</tbody>
</table>

15.3. 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 [9].

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 22 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 ($h_v$) at the peak flow rate is shown for each mesh in Figure 23. 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

$$y^+ = \frac{u^* h}{\nu},$$  \hspace{1cm} (11)

![Fig. 16: Aortic-valve flow analysis. Asymmetric leaflet motion. Magnitude of the WSS (Pa). Upper surface (left) and lower surface (right). The frames are for $t = 1.072, 1.080, \text{ and } 1.252 \text{ s.}](image)

![Fig. 17: Patient-specific aorta flow analysis. Geometry and boundary conditions.](image)

![Fig. 18: Patient-specific aorta flow analysis. Volumetric flow rate at the inlet.](image)
where the friction velocity $u^*$ is based on the computed value of the WSS as follows:

$$u^* = \sqrt{\frac{\|h^k\|}{\rho}}.$$  \hspace{1cm} (12)

Figure 24 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 23 and 24, 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 25, and Figure 26 shows the spatially-averaged WSS magnitude in a cycle.

Figure 27 shows the oscillatory shear index (OSI), defined as

$$\text{OSI} = \frac{1}{2} \left( 1 - \frac{\|\int_0^T h^k \, dt\|}{\int_0^T \|h^k\| \, dt} \right).$$  \hspace{1cm} (13)

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 $\|u^k - u^l\|$, where the subscripts indicate Base Mesh and Refinement Mesh $k$.

**Remark 2.** 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 28 shows the velocity difference at 0.5 s.

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.

16. ZSS estimation: patient-specific aorta

This section is from [20], where more details can be found.

The largest diameter is about 30 mm. The inner-surface mesh is shown in Figure 29. We use Laplace’s equation to determine a smooth thickness distribution. We then generate the volume mesh as described in [20], which involves modification of the outer surface and consequently the thickness. The volume mesh is shown in Figure 30. The measured thickness is displayed on the inner-surface mesh in Figure 31. We generate the initial guess following the procedure described in [20].

![Fig. 21: 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.](image1)

![Fig. 22: 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).](image2)
Figure 23: Patient-specific aorta flow analysis. Mesh refinement study. WSS (dyn/cm²) at the peak flow rate.

Figure 24: 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.

Figure 32 shows the IPBZSS, using the EBZSS representation, from the ZSS initial guess and converged ZSS. The stretches are shown in Figures 33 and 34. Overall, the ZSS initial guess and converged ZSS are very similar. This indicates that reaching the ZSS design targets based on the analytical solution works well.

17. IMGA computation: patient-specific heart valve design and analysis

This section is from [146], 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 geom-
Fig. 25: Patient-specific aorta flow analysis. Mesh refinement study. TAWSS (dyn/cm^2).

Fig. 26: Patient-specific aorta flow analysis. Mesh refinement study. Spatially-averaged WSS during a cycle.

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 elastic-mesh moving problem [7, 171, 172] for the displacement from this regular domain to a deformed configuration that represents the artery and lumen. However, solving a linear elastic-static 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 35a.) At each cross section, we calculate its unit normal.
vector $n_k$ 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 35a.) 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 35b and 35c. 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 sub-

---

**Fig. 27:** Patient-specific aorta flow analysis. Mesh refinement study. OSI.

**Fig. 28:** Patient-specific aorta flow analysis. Mesh refinement study. Velocity difference $||u_{k}^h - u_{h}^k||$ (m/s) at 0.5 s, where the subscripts indicate Base Mesh and Refinement Mesh $k$.

**Fig. 29:** Patient-specific aorta. Inner-surface mesh, made of cubic and quartic T-splines. Red circles represent the control points. The parts with the quartic T-splines, obtained by order elevation [170], are around the two extraordinary points, each connected to six edges.
domains, shown in gray and red, respectively, in Figure 35d. Basis functions are made $C^0$-continuous at the 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

**Fig. 30:** Patient-specific aorta. Volume mesh, made of cubic and quartic T-splines. *Red* circles represent the control points.

**Fig. 31:** Patient-specific aorta. Measured thickness (mm).

**Fig. 32:** Patient-specific aorta. The IPBZSS, shown using the EBZSS representation. From the ZSS initial guess (left) and converged ZSS (right).
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 35f.

17.2. 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

![Fig. 33: Patient-specific aorta. Maximum principal stretch. From the ZSS initial guess (left) and converged ZSS (right).](image)

![Fig. 34: Patient-specific aorta. Minimum principal stretch. From the ZSS initial guess (left) and converged ZSS (right).](image)
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 36. 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 36. The attachment edges, free edges, and belly curves are then interpolated to obtain smooth bivariate B-spline representations of the leaflets.

Figure 37 shows the details of parameterizing the free-edge curve (red) and the belly-region curve (green). In Figure 37, \( p_1, p_2 \) and \( p_3 \) are the key points on the top of the commissure lines and \( p_4 \) is the key point on the sinus bottom, as labeled in Figure 36. Points \( p_1-p_3 \) define a triangle \( \Delta p_1-3 \), with \( p_c \) being its geometric center. The unit vector pointing from \( p_o \) to \( p_c \) is denoted by \( t_p \), and the unit normal vector of \( \Delta p_1-3 \) pointing downwards is \( n_p \). We first construct the free edge curve as a univariate quadratic B-spline curve determined by three control points, \( p_1, p_2, \) and \( p_2 \). The location of \( p_f \) is defined by \( p_f = p_c + x_1 t_p + x_2 n_p \). By changing \( x_1 \) and \( x_2 \) to control the location of \( p_f \), the curvature and the height of the free edge can be parametrically changed. We then take \( p_m \) as the midpoint of the free edge, the point \( p_o \), and the key point \( p_4 \) to construct a univariate quadratic B-spline curve (green). The point \( p_o \) is defined by \( p_o = p_o + x_3 n_p \), where \( p_o \) is the projection of \( p_m \) onto \( \Delta p_1-3 \) along the direction of \( n_p \). 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 [173].

### 17.3. 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 \(x_1, x_2, \text{ and } x_3\) 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 \(c_0 = 2.0 \times 10^6 \text{ dyn/cm}^2\), \(c_1 = 2.0 \times 10^5 \text{ dyn/cm}^2\), 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 [118]. Mass-proportional damping with constant \(C_{\text{damp}} = 10^4 \text{ Hz}\) is used to model the interaction of the artery with the surrounding tissue. The dynamic viscosity of the blood is set to \(\mu = 3 \times 10^{-2} \text{ g}/(\text{cm s})\).

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 \(-(p_0 + RQ)\mathbf{n}_f\) 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 \(p_0 = 80\ \text{mmHg}\) and \(R = 200\ \text{(dyn s)/cm}^5\). 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 \(\Delta t = 10^{-4}\ \text{s}\) is used in all simulations. To obtain the artery wall tissue prestress, we apply the highest left ventricular pressure during systole (127 mmHg at \(t = 0.25\ \text{s}\)) on the inlet and a resistance boundary condition (\(p_0 = 80\ \text{mmHg}\) and \(R = 200\ \text{(dyn s)/cm}^5\)) on the outlet for the calculation of \(\mathbf{h}_t\) in the prestress problem [119].

We perform FSI simulations of each of \([x_1, x_2, x_3] \in \{(0.05, 0.25, 0.45), (0.1, 0.3, 0.5), (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 [146]. 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 \(x_1 = 0.05\ \text{cm}, x_2 = 0.1\ \text{or 0.3 cm, and } x_3 = 0.5\ or 0.8\ \text{cm reliably yields a high EOA between 3.92 and 4.05 cm}^2\), 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\(^2\). Among these four cases, \(x^* = (x_1, x_2, x_3) = (0.05\ \text{cm}, 0.1\ \text{cm}, 0.8\ \text{cm})\),
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 38.

Figure 39 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 40, 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 41. 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.

18. Concluding remarks

We have provided an overview of how the computational challenges faced in cardiovascular flow analysis have been addressed with the ST-VMS, ALE-VMS, and the IMGA-VMS, which serve the core computational methods, and the special methods used in combination with them. By presenting examples of the challenging computations carried out, we demonstrated that these core and special methods now enable reliable and practical computational analysis in a wide range of patient-specific cases, including cerebral aneurysms, aortas and heart valves. The computational challenges overcome include unsteady flows, complex cardiovascular geometries, MBIs, such as the motion of the heart valve leaflets, contact between moving solid surfaces, such as the contact between the leaflets, and the FSI between the blood and the cardiovascular structure. They also include estimating the ZSS required in the FSI computation. With many of the challenges mostly out of the way, computational cardiovascular flow analysis is now in a better position to provide valuable information to surgeons and medical doctors.

ACKNOWLEDGEMENT

This work was supported (first 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
Fig. 39: Volume rendering visualization of the velocity field from our FSI simulation at several points during a cardiac cycle.

Fig. 40: 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. 41: 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.
of the work were also supported (third 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 second author was partially supported by NSF Grant 1854436, and the fourth author was partially supported by NIH/NHLBI Grants R01HL129077 and R01HL142504.

References


About Authors

Kenji TAKIZAWA received his PhD from Tokyo Institute of Technology in 2005, and he is currently a Professor in Department of Modern Mechanical Engineering at Waseda University. He has been conducting computational fluid mechanics research since 2000, teaching classes on that subject since 2010, and has been conducting computational FSI research since 2003. He has published over 90 Web-of-Science-indexed journal articles on computational fluid and structural mechanics and FSI. He is a Web of Science Highly Cited Researcher. He coauthored a textbook titled Computational Fluid–Structure Interaction: Methods and Applications, published by Wiley, with the Japanese translation published by Morikita Publishing Company. More information on Takizawa can be found at http://www.jp.tafsm.org/.

Yuri BAZILEVS received his PhD from University of Texas at Austin in 2006, and he is currently the E. Paul Sorensen Professor of Engineering at Brown University. He has been conducting computational fluid mechanics research since 2000, teaching classes on that subject since 2008, and has been conducting computational FSI research since 2003. He has published over 140 Web-of-Science-indexed journal articles on computational fluid and structural mechanics and FSI. He is a Web of Science Highly Cited Researcher. He coauthored a book on isogeometric analysis, a technique widely used in computational mechanics, and FSI. He coauthored a textbook titled Computational Fluid–Structure Interaction: Methods and Applications, published by Wiley, with the Japanese translation published by Morikita Publishing Company. He is an Associate Editor of Elsevier journal Computers and Fluids and is responsible for the manuscripts on computational fluid mechanics and FSI. More information on Bazilevs can be found at https://vivo.brown.edu/display/ybazilev.

Tayfun E. TEZDUYAR received his PhD from Caltech in 1982, and he is currently the James F. Barbour Professor of Mechanical Engineering at Rice University. He has been conducting computational fluid mechanics research since 1979, teaching classes on that subject since 1987, and has been conducting computational FSI research since 1991. He has published over 240 Web-of-Science-indexed journal articles on computational fluid and structural mechanics and FSI. He is a Web of Science Highly Cited Researcher. He coauthored a textbook titled Computational Fluid–Structure Interaction: Methods and Applications, published by Wiley, with the Japanese translation published by Morikita Publishing Company. He is an Editor of Springer journal Computational Mechanics and is responsible for the manuscripts on computational fluid mechanics and FSI. More information on Tezduyar can be found at http://www.tafsm.org/tezduyar/.

Ming-Chen HSU received his PhD from University of California, San Diego in 2012, and he is currently an Associate Professor in the Department of Mechanical Engineering at Iowa State University. He has been conducting computational fluid mechanics research since 2006, teaching classes on that subject since 2013, and has been conducting computational FSI research since 2008. He has published over 60 Web-of-Science-indexed journal articles on computational fluid and structural mechanics and FSI. He is a Web of Science Highly Cited Researcher. More information on Hsu can be found at https://web.me.iastate.edu/jmchsu/.

"This is an Open Access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium provided the original work is properly cited (CC BY 4.0)."