**Computational Cardiovascular Medicine with Isogeometric Analysis**

Kenji TAKIZAWA¹*, Yuri BAZILEVS², Tayfun E. TEZDUYAR³, Ming-Chen HSU⁴, Takuya TERAHARA¹

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

*Corresponding Author: Kenji TAKIZAWA (Email: Kenji.Takizawa@tafsm.org)

(Received: 12-May-2022; accepted: 7-Jul-2022; published: 30-Sep-2022)

DOI: http://dx.doi.org/10.55579/jaec.202263.381

**Abstract.** Isogeometric analysis (IGA) brought superior accuracy to computations in both fluid and solid mechanics. The increased accuracy has been in representing both the problem geometry and the variables computed. Beyond using IGA basis functions in space, with IGA basis functions in time in a space–time (ST) context, we can have increased accuracy also in representing the motion of solid surfaces. Around the core methods such as the residual-based variational multiscale (VMS), ST-VMS and arbitrary Lagrangian–Eulerian VMS methods, with complex-geometry IGA mesh generation methods and immersogeometric analysis, and with special methods targeting specific classes of computations, the IGA has been very effective in computational cardiovascular medicine. We provide an overview of these IGA-based computational-cardiovascular-medicine methods and present examples of the computations performed.

**Keywords**

Computational cardiovascular medicine, Isogeometric analysis, Variational multiscale methods, Space–time method, ALE method, Complex-geometry IGA mesh generation methods, Immersogeometric analysis.

1. Introduction

The challenges involved in computational cardiovascular medicine (see, for example, [1–21]) include complex geometries, fluid–structure interaction (FSI) between the blood and the cardiovascular tissues, and topology changes in the computational domain, such as the contact between the heart valve leaflets. Because validation by comparing to experimental or test data is exceedingly difficult in many cases, striving for the best accuracy in all aspects of the computation is even more important. For example, as early as in 2004, with some of the earliest patient-specific arterial FSI computations [22, 23], which were performed with the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) method [24–26], it was shown that taking the FSI into account was essential for accurate prediction of the wall shear stress (WSS). The patient-specific arterial geometries used in [22, 23] were for the middle cerebral artery and internal carotid artery, and assuming rigid arteries was overpredicting the
WSS by large margins. The WSS overprediction trend was also reported in [27], with patient-specific arterial FSI computation for 10 different cases of cerebral aneurysm, with reasonably good boundary layer resolution near the arterial walls. As another example, the WSS significance of having a good boundary layer resolution near the arterial wall in finite element FSI computations was pointed out [28] and quantified [29] as early as in 2008.

Isogeometric analysis (IGA) [30–33] brought superior accuracy to computations in both fluid and solid mechanics. The increased accuracy has been in representing both the problem geometry and the variables computed. Beyond using IGA basis functions in space, with IGA basis functions in time in an ST context [34–37], we can have increased accuracy also in representing the motion of solid surfaces. The IGA has been used around core methods such as the residual-based variational multiscale (RBVMS) [38–41], arbitrary Lagrangian–Eulerian VMS (ALE-VMS) [1, 4, 32, 42–45], DSD/SST [24–26], which gained the alternate name “ST-SUPS” in [4], and ST-VMS [34, 35, 46]. It is known that IGA mesh generation for complex geometries is significantly more challenging than finite element mesh generation. This challenge has mostly been overcome or circumvented with the Complex-Geometry IGA Mesh Generation (CGIMG)1 method [47, 48], NURBS Surface-to-Volume Guided Mesh Generation (NSVGMG) method [49], and immersogeometric analysis (IMGA) [11, 50]. With the core methods, with the methods that overcome or circumvent the complex-geometry IGA mesh generation challenge, and with special methods targeting specific classes of computations, the IGA has been very effective in computational cardiovascular medicine. We provide an overview of these IGA-based computational-cardiovascular-medicine methods and present examples of the computations performed. The overview of the methods is in Sections 2–12, and the examples are in Sections 13–15. The concluding remarks are given in Section 16.

2. Moving-mesh and nonmoving-mesh methods

Flows with moving boundaries and interfaces (MBI) [4, 51–53] is a wider class of flow problems that includes FSI problems. It also includes flows with moving mechanical components, free-surface and two fluid-flows, and fluid–particle interactions. For a given moving boundary or interface, if the method is relying on a mesh that moves to follow that boundary or interface, it is a moving-mesh method, if not, it is a nonmoving-mesh method. Moving the mesh to follow a fluid–solid interface gives us mesh resolution control near the moving solid surfaces and higher accuracy in representing the boundary layers. The moving-mesh methods have also been referred to as “interface-tracking methods” [4, 51–54], meaning that the mesh is moving for the purpose of “tracking” the interface, rather than just “capturing” it within the mesh resolution where the interface is in the nonmoving mesh. More on the moving-mesh and nonmoving-mesh (i.e. interface-tracking and interface-capturing) methods and their mixtures, the Mixed Interface-Tracking/Interface-Capturing Technique [55] and Fluid–Solid Interface-Tracking/Interface-Capturing Technique (FSITICT) [56], can be found in [4, 51–53].

3. ST-SUPS, ALE-SUPS, RBVMS, ALE-VMS, ST-VMS, and the classes of problems computed

The ST-SUPS, introduced in 1990 with the name DSD/SST, is the oldest one of the

---

1The method name and abbreviation are being coined here.

2We note that, in a flow computation in general, a method might be relying on a mesh that moves to follow a moving boundary or interface but does not move to follow a second moving boundary or interface. Then the method is a moving-mesh method for the first moving boundary or interface and a nonmoving-mesh method for the second.
moving-mesh methods we include in this article. The alternate name “ST-SUPS” brought clarity to the stabilization components, which are the SUPG and PSPG, with widespread awareness that these two acronyms imply stabilization methods Streamline-Upwind/Petrov-Galerkin [57] and Pressure-Stabilizing/Petrov-Galerkin [24]. The ram-air parachute FSI analysis in 1999 [58] was one of the earliest computations with the ALE-SUPS moving-mesh method. The root moving-mesh method ALE is of course much older, with the finite element version dating back to 1981 [59]. The ALE-VMS and ST-VMS are the VMS versions of the ALE and DSD/SST. In both, the stabilization components are from the RBVMS.

The ST-SUPS, ALE-SUPS, ALE-VMS, and ST-VMS, like all moving-mesh methods, need to be complemented with mesh update methods in FSI and MBI computations. The mesh update most of the time consists of moving the mesh to accommodate the motion of the boundaries and interfaces and to control the mesh resolution near solid surfaces that are moving, and remeshing if the element distortion exceeds an acceptable level. We expect two things from a good mesh moving method: to reduce the need for remeshing and to give high priority to maintaining element quality near solid surfaces where accurate representation of the boundary layers matters. Since the inception of the ST-SUPS in 1990, a large number of special- and general-purpose mesh moving methods have been developed for computations with the ST-SUPS and ST-VMS. Some of them have also been used in computations with the ALE-SUPS and ALE-VMS. A recent article [60] on mesh moving methods provides an overview. The general-purpose methods include, as the first one, the linear-elasticity mesh moving method with mesh-Jacobian-based stiffening [61, 62] introduced in 1992, and, as the most recent ones, element-based mesh relaxation method [63], mesh relaxation and mesh moving based on fiber-reinforced hyperelasticity [53], and back-cycle-based mesh moving method [20, 64].

With some of the most diverse and challenging classes of flow problems computed over the time period since their inception, the ST-SUPS, ALE-SUPS, RBVMS, ALE-VMS, and ST-VMS built a track record for being a powerful set of methods with wide scope. The classes of problems computed include those itemized below.

ALE-SUPS, RBVMS, and ALE-VMS:

- wind turbines [65–86],
- cardiovascular medicine [5, 8, 9, 11, 12, 17, 18, 31, 32, 87–92],
- mixed ALE-VMS/IMGA computations [9, 11, 12, 93–101] in the framework of the FSITICT,
- turbomachinery [102–108],
- two-phase flows [109–115],
- bridges [116–120],
- free-surface flows [121–125],
- IMGA FSI and flow analysis [50, 126–129],
- marine applications [130–132],
- stratified flows [133, 134],
- aircraft applications [135, 136],
- parachutes [58],
- hypersonic flows [137],
- additive manufacturing [138].

ST-SUPS and ST-VMS:

- classes of problems summarized in [139] (all computed during the 25-year period 1993–2018),
- cardiovascular medicine [2, 3, 6, 7, 10, 13–21, 140–146],
- wind turbines [4, 65, 72, 79–82, 141, 142, 147–154],
- parachutes [4, 63, 83, 84, 155–164],
- ground vehicles and tires [10, 46, 49, 83, 84, 165–170],
- flapping-wing aerodynamics [4, 36, 37, 60, 140–142, 171–174],
• turbomachinery [47, 48, 81, 82, 175–180],
• fluid films [168, 170, 181],
• spacecraft [157, 182],
• Taylor–Couette flow [62, 183],
• disk brakes [184],
• U-ducts [185].

4. Slip (sliding) interface methods

The sliding interface method was introduced in [33, 186] in the context of the ALE-VMS. The ST version of that is the ST Slip Interface (ST-SI) method, introduced in [150]. In this article, the acronym “SI” will imply both “sliding” and “slip.” Both the ALE-SI and ST-SI were introduced in the context of incompressible flows. The objective was to retain the favorable moving-mesh features of the ST-SUPS, ALE-SUPS, ALE-VMS, and ST-VMS in flow computations that have a rotating solid surface, such as a turbine rotor. The mesh around the rotating solid and inside the SI, which would typically have higher refinement near the solid surface, rotates with it and thus preserves the high-resolution boundary layer representation. The mesh outside the SI is not affected by the rotation of the solid. Connecting the two sides of the solution accurately is accomplished by adding to the ST-SUPS, ALE-SUPS, ALE-VMS, and ST-VMS in flow computations that have a rotating solid surface, such as a turbine rotor. The integrations, inherently residual-based, account for the velocity and stress compatibility at the SI. A number of versions of the ST-SI were introduced to serve purposes that are different than the original one but just as important.

In addition to the ST-SI version with a standard “fluid–fluid SI,” a version with “fluid–solid SI” was introduced in [150]. The SI between the fluid and solid domains helps enforce the fluid mechanics Dirichlet boundary conditions weakly. The version for coupled incompressible-flow and thermal-transport equations was introduced in [184]. With that, thermo-fluid boundary layers near rotating solid surfaces can also have high-resolution representation. The version introduced in [150] has the SI between a thin porous structure and the fluid on its two sides. With that, how the porosity is dealt with is consistent with how the fluid–fluid and fluid–solid SIs are dealt with. The ST-SI versions introduced in [163] are the compressible-flow counterparts of the three versions we discussed so far. They work with the compressible-flow porosity models introduced in [163] and compressible-flow ST SUPG method [187].

The classes of problems computed with the ST-SI include turbomachinery [47, 48, 81, 82, 175–180], cardiovascular medicine [13–15, 17–21, 146], ground vehicles and tires [10, 49, 165–170], wind turbines [79–82, 150, 151], parachutes [83, 84, 162–164], fluid films [168, 170, 181], disk brakes [184], Taylor–Couette flow [183], and U-ducts [185].

5. ST Topology Change (ST-TC) method

In FSI and MBI problems, contact between moving solid surfaces can be of two types: “near contact” and actual contact. In a near contact, there is still a narrow gap between the solid surfaces, and therefore there is no topology change in the fluid mechanics domain. In such cases, the nearness is close enough for obtaining physically meaningful results from the flow computation, basically good enough for solving the problem. With a robust mesh moving method, no element needs to collapse and good boundary layer resolutions can be retained. Several classes of flow problems were computed with the ST-SUPS and ST-VMS under near-contact conditions with sufficient accuracy. Examples of such computations can be found in the references mentioned in [140].

In some classes of flow problems, it is essential to represent the contact as an actual contact. For example, in heart valve flow analysis, for obvious reasons, the contact between the valve leaflets needs to be represented as an actual contact, without leaving a narrow gap. As another example, in wing clapping aerodynamics of insects, the contact between the upper and lower wings needs to be an actual contact. The ST-
TC method [7, 140] was introduced to make ST moving-mesh computations possible even in flow problems that involve an actual contact. With that, we can both represent an actual contact and retain the good boundary layer resolution. Elements collapse as needed, which is doable in the ST context. The connectivity of the “parent” mesh, however, does not change during the process of an element collapse or rebirth, and therefore the computational efficiency does not degrade.

The ST-TC has been used without giving up neither the actual contact representation nor the high-resolution boundary layer representation in heart valve flow analysis [7, 10, 13–15, 17–19, 140, 142], ventricle-valve-aorta flow analysis [20, 21, 146], tire aerodynamics with road contact and deformation [10, 49, 165–170], fluid films [168, 170, 181], and wing clapping aerodynamics of insects [140, 174].

6. ST-SI-TC

Effective usage of the ST-TC requires its integration with the ST-SI, and the ST-SI-TC [14, 165] was introduced as that integration. We briefly summarize the need for and benefits from the ST-SI-TC. The ST-SI needs elements on both sides of a fluid–fluid SI. We need to take measures to meet that requirement in flow computations where part of the SI coincides with a solid surface. We face that, for example, when the SI is between two solid surfaces coming into contact, and in a more general context, when it merges with a fluid–solid interface. In such cases, the elements between the solid surface and the part of the SI coinciding with it collapse in the ST-TC contact mechanism. With that, the part of the SI coinciding with the solid surface switches from fluid–fluid SI to fluid–solid SI. That could create an SI that is a mixture of fluid–fluid and fluid–solid SIs. The ST-SI-TC makes it possible to have high-resolution boundary layer representation near the fluid–solid interfaces even when part of the SI coincides with a solid surface. With the ST-SI-TC, we can manage contact location change and contact sliding in an effective fashion.

The ST-SI-TC has been used in heart valve flow analysis [13–15, 17–19], ventricle-valve-aorta flow analysis [20, 21, 146], tire aerodynamics with road contact and deformation [10, 49, 165–170], and fluid films [168, 170, 181].

7. ST-IGA

The ST-IGA is a broadly-defined term for the integration of the ST methods with isogeometric discretization. The method was introduced in [34]. The test computations reported in [34] were with the ST-VMS and ST-IGA and in 2D. The computations were for flow past an airfoil, with IGA basis functions in space, and for the advection equation, with IGA basis functions in both space and time. The test computations for the advection equation showed that using higher-order basis functions in time increases the accuracy return from using higher-order basis functions in space.

In early ST-IGA computations, the focus was on IGA basis function in time [34–37, 171], to have i) increased accuracy in representing the motion of solid surfaces, ii) a mesh motion consistent with that, iii) increased efficiency in representing the motion of the volume meshes, and iv) increased efficiency in remeshing. The ST/NURBS Mesh Update Method (STNMUM) [36, 37, 149, 171] was introduced possessing these desirable features. The STNMUM is suitable for, among a wider class of problems, rotating solid surfaces, and that is good context for explaining its good performance. The representation of the circular trajectory in the STNMUM is exact with the use of quadratic NURBS basis functions in time and an adequate patch count. In addition, it is possible to specify a constant angular velocity for speeds invariant along the circular trajectory. That requires a secondary mapping, given in [4, 34–36]. The “ST-C” method [188] is another positive outcome of combining the ST context with the IGA basis functions in time. It is a way of extracting a continuous representation in time from the computed data. It is also an efficient way of data...
compression [10, 46, 81–84, 152–154, 175, 178, 179, 184, 188].

The ST-IGA with IGA basis functions in time has been used in flapping-wing aerodynamics [4, 36, 37, 60, 140–142, 171–174], wind turbines [72, 79–82, 141, 142, 149–151], turbomachinery [47, 48, 81, 82, 175–180], and spacecraft cover separation aerodynamics [157].

Because ST-IGA with IGA basis functions in space brings increased accuracy with fewer control points and therefore larger effective element sizes, larger time-step sizes can be used while keeping the Courant number at a desirable level for good accuracy. The ST-IGA with IGA basis functions in space has been used in cardiovascular medicine [13–21, 146] turbomachinery [47, 48, 81, 82, 176–180], ground vehicles and tires [49, 166–170], wind turbines [81, 82, 151–154], parachutes [83, 84, 162, 164], fluid films [168, 170, 181], Taylor–Couette flow [183], and U-ducks [183].

The IGA basis functions in space play a key role also in the newest zero-stress-state (ZSS) estimation methods [17, 189–192] and related hyperelastic shell analysis [193]. The ZSS estimation is needed in patient-specific arterial FSI computations, because the image-based arterial geometries used in the computations do not correspond to the ZSS of the artery. The IGA basis functions in space have been used numerous advanced computational technologies in structural analysis and design, such as those reported in [194–203], including those for wind turbine blades and heart valves.

8. ST-SI-IGA and ST-SI-TC-IGA

The ST-SI-IGA [176] and ST-SI-TC-IGA [13, 14, 167] are essentially the IGA expansions of the ST-SI and ST-SI-TC discussed in Sections 4 and 6. We get ST-SI-IGA and ST-SI-TC-IGA by building an integrated combination of the ST-SI and ST-SI-TC with the ST-IGA.

The ST-SI-IGA retains the favorable moving-mesh features of the ST-SUPS, ALE-SUPS, ALE-VMS, and ST-VMS in IGA-based flow computations that have a rotating solid surface, such as a turbine rotor. The ST-SI-IGA mechanism and desirable features include those that are basically the same as what we described in Section 4. for the ST-SI. Beyond that, the ST-SI-IGA addresses the mesh generation challenge in IGA discretization. This is accomplished with an SI that does not have a slip between the two sides. The SI just connects the parts of the solution obtained over two IGA mesh zones with nonmatching meshes at the SI between the zones. Because we are no longer constrained by a matching requirement, in computation of flow problems with complex geometries, the IGA discretization becomes more practical. In IGA-based computations with a thin porous structure embedded in the flow field, the ST-SI-IGA mechanism is essentially the same as what we described in Section 4. for the ST-SI. In some cases, the rotating solid surface has grooves or creates narrow spaces or the thin porous structure has gaps and slits. In computation of such flow problems, the ST-SI-IGA makes it possible to keep the element density, and consequently the computational cost, at an acceptable level. That makes computations even with such geometric complexities practical.

The ST-SI-TC-IGA, in flow computations with contact between moving solid surfaces, makes it possible to keep the element density in the narrow spaces close to the contact region at an acceptable level. While the solid surfaces come into contact, prior to the elements between a solid surface and SI collapsing, we may have curved and complex boundaries and narrow spaces. These would need high-aspect-ratio elements. The ST-SI-TC-IGA makes it possible to compute under such adverse conditions with an acceptable level of computational cost. With the enhancements introduced in [181], the ST-SI-TC-IGA acquired a built-in Reynolds-equation limit. With that, when the solid surfaces coming into contact have fluid films between them, we do not need to use separately a Reynolds-equation model in those regions. The ST-SI-TC-IGA can handle that with comparable solution quality and computational cost and also work in the other parts of the flow domain where the Reynolds-equation model would not work.
The ST-SI-IGA and ST-SI-TC-IGA have been used in cardiovascular medicine [13–15, 17–21, 146], turbomachinery [47, 48, 81, 82, 176–180], ground vehicles and tires [49, 166–170], parachutes [83, 84, 162, 164], wind turbines [81, 82, 151], and fluid films [168, 170, 181].

9. Complex-geometry IGA mesh generation

While the IGA offers superior accuracy, IGA mesh generation for complex geometries is significantly more challenging than finite element mesh generation. Widely available mesh generation software packages for finite element and finite difference methods encourage the usage of these methods. To make IGA-based flow computations more applicable to problems with complex geometries, and consequently more practical in computational analysis of real-world problems, the IGA mesh generation will have to be less challenging and more encouraging. The Complex-Geometry IGA Mesh Generation (CGIMG) [47, 48] and NURBS Surface-to-Volume Guided Mesh Generation (NSVGMG) [49] methods were introduced to that end. We will provide a brief overview of the CGIMG here, and for the NSVGMG, we refer the interested reader to [49].

The CGIMG consists of three steps. In the first step, a block-structured mesh is generated using existing techniques for such meshes. In the second step, that mesh is projected to a NURBS mesh that is built from patches corresponding to the blocks of the block-structured mesh. In the third step, the original model surfaces are recovered, to the extent the nature of the recovered surfaces does not impede the robustness or accuracy of the flow computations. The CGIMG should normally preserve the element quality and refinement distribution of the block-structured mesh. Mesh generation and mesh quality tests were included in [47, 48]. The tests showed that the CGIMG is a practical IGA mesh generation method with good performance.

The CGIMG has been used in cardiovascular medicine [15–18, 20, 21, 48, 146], turbomachinery [47, 48, 81, 82, 107, 177, 179, 180], wind turbines [82, 151], and parachutes [48].

10. IMGA

The IMGA was first introduced in [12] as an immersed, geometrically flexible approach for solving computational FSI problems involving large, complex structural deformation and change of fluid domain topology (e.g., structural contact). 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 a non-body-fitted analysis mesh. A new semi-implicit dynamic augmented Lagrangian (DAL) approach [204] was introduced in [12] for weakly enforcing the constraints at the fluid–structure interface in time-dependent immersedogeometric FSI problems. In [12], the method was first applied to the FSI simulation of bioprosthetic heart valves (BHV).

A mixed ALE-VMS/IMGA methodology was developed in [9] in the framework of the FSITICT [56], where a single computation combines a body-fitted, deforming-mesh treatment of some fluid–structure interfaces with a non-body-fitted treatment of others. This approach enabled us to simulate the FSI of a heart valve in a deforming artery over the entire cardiac cycle under physiological conditions and to study the effect of arterial-wall elasticity on the valve dynamics [9]. The DAL-based ALE-VMS/IMGA approach was integrated with CAD for heart valve analysis in [11] with a comparison between dynamic and FSI computations to demonstrate the importance of including FSI in heart-valve simulations. An anisotropic constitutive modeling of the BHV leaflets, based on the isogeometric Kirchhoff–Love shell formulation for general hyperelastic materials [196], was proposed in [98] and employed in the IMGA of heart-valve FSI. The framework was extended to include the coupling between the heart valve leaflets and the arterial wall in [96] to study patient-specific valve designs that do not employ stents. The results were compared with phase-contrast MRI data, in which a qualitative similarity of the flow pat-
terns in the ascending aorta was found. More recently, the framework was applied to study leaflet flutter and its potentially damaging impact on the cardiac system due to the use of thinner, more flexible biological tissues in BHVs [100, 205]. A model of a moving left ventricle was added to the heart-valve FSI framework in [101] to study the changes in the left ventricular hemodynamics following the BHV replacement of both aortic and mitral valves. In order to simulate the next generation of BHVs employed in a minimally-invasive transcatheter aortic valve replacement (TAVR) procedure, the IGA-based Bernoulli beam formulation was added to the ALE-VMS/IMGA framework in [99]. This enabled the modeling of a complex TAVR device, which includes a frame, skirt, and three leaflets, and simulation of the interaction between the blood flow, arterial wall, and TAVR in order to study the anchoring ability of the device.

Other noteworthy work on IMGA includes a divergence-conforming formulation of fluid mechanics that delivers a divergence-free velocity field everywhere in the domain and addresses the mass loss error across the valve interface [95]. In addition, clever stable coupling strategies and the appropriate definition of the Lagrange multipliers in the DAL method were proposed and analyzed in [93, 94, 97]. The DAL-based IMGA was also combined with surrogate modeling in [136] for the effective use of FSI to optimize the design of a hydraulic arresting gear. Parts of the IMGA methodology were recently implemented in the FEniCS-based tIGAr software [206] as a new open-source library ConDAlFISH in [207]. A rigid-wall, idealized, 3D heart valve FSI example was included as part of that implementation.

11. Stabilization parameters and element lengths targeting IGA discretization

The ST-SUPS, ALE-SUPS, RBVMS, ALE-VMS, ST-VMS, like most stabilized methods, have “stabilization parameters” [4]. These parameters, which play an important role, are called “\( \tau_{SUPS} \),” “\( \tau_{SUPG} \),” and “\( \nu_{LSIC} \)” [208]. Typically a single parameter, called “\( \tau_{SUPS} \)” [4, 34], is used instead of separate \( \tau_{SUPG} \) and \( \tau_{SUPP} \). A local length scale, quite often called “element length,” appears in the expressions for the stabilization parameters. The element length also appears in the integrations over the SIs, with the length measured in the interface-normal direction. The expressions for the element lengths and stabilization parameters used with the ST-SUPS, ALE-SUPS, RBVMS, ALE-VMS, ST-VMS go way back to 1979–1982 [57, 209–212]. Many different expressions were introduced after that (see for example, [25, 26, 36, 46, 51, 148, 149, 213–217]), all intended, until 2018, for finite element discretization. For the first few decades, the element length was an advection length scale, measured in the flow direction, and then a second element length, which is a diffusion length scale, was added. Some of the expressions were in the ST context [25, 216], some were specific to the VMS stabilization [46], some were in the context of the coupled incompressible-flow and thermal-transport equations [46], and some made sure that element lengths, including the direction-dependent ones, had node-numbering invariance also for simplex elements [217]. All these element lengths and stabilization parameters intended for finite element discretization have also been used in computations with IGA discretization.

Element lengths and stabilization parameters targeting IGA discretization were introduced in [218]. They can also be used in computations with finite element discretization. There were three key steps in conceptually simple derivation of the direction-dependent element length expression. i) Map the direction vector from the physical ST element to the parent ST element. ii) Account for the discretization spacing along each of the parametric coordinates. iii) Map what has been obtained in the parent element back to the physical element. The latest stabilization parameters designed for the ST-VMS are those given in [167], and they are mostly from [218]. The direction-dependent element length expressions introduced in [219], which target complex-geometry B-spline meshes, are
based on a preferred parametric space instead of the standard integration parametric space. They involve a transformation tensor that represents the relationship between the two parametric spaces. These new expressions yield local length scales that are invariant with respect to element splitting (see [220] for the proof). Meeting this invariance requirement in the element length definition is essential because otherwise the solution is influenced by the element splitting, which, of course, we do not want.

The local-length-scale expressions introduced in [218,219] have been used in ground vehicles and tires [49,167–170], wind turbines [151–154], cardiovascular medicine [19–21,146], fluid films [168,181], turbomachinery [177,180], parachutes [164], Taylor–Couette flow [183], and U-ducts [185]. They have also been used in calculating the Courant number based on the NURBS mesh local length scale in the flow direction [107]. That was in connection with an IGA-based gas turbine flow computation.

12. Constrained-Flow-Profile Traction

The Constrained-Flow-Profile (CFP) Traction [20] is an example of the special methods targeting specific classes of computations. The CFP Traction gives us flow stability at an inflow boundary where we specify not the flow velocity, as it is typically done, but the traction. The method was introduced to meet a need in computational cardiovascular medicine but can also be used in other classes of computations where we need to specify the traction at an inflow boundary.

Figure 1 shows a 2D quadratic NURBS mesh for a channel flow, which we are using as a context to describe the CFP Traction method. We are specifying the traction at both the inflow and outflow boundaries. A large, single element placed at the inflow is the special-purpose element that serves as the core of the CFP Traction method. It has nine basis functions in 2D, and would have 27 basis functions in 3D.

An SI connect the solutions obtained over the large element and the rest of the mesh. The special-purpose element has only one unspecified control-point velocity component at the inflow boundary, and that is in the normal direction. With that configuration, a constrained flow profile is produced at the inflow boundary. It is of course a quadratic velocity profile because of the NURBS basis functions used. The combination of the quadratic velocity profile at the inflow boundary and the solution obtained for the unspecified velocity component results in the flow rate associated with the specified inflow and outflow traction values.

13. Computational example: channel flow with CFP Traction

This is a summary of the 2D test computations presented in [20]. The objective in the computations was to evaluate the performance of the CFP Traction in inflow stabilization. For this type of a test problem, computing in 2D is sufficient. We compare what we get to the analytical solution and to the solution obtained by using the outflow stabilization of [221] also at the inflow, which we will identify with the abbreviation “OS.” The parameter “$\beta$” embedded in the formulation of the OS is selected from the range $\beta \geq 1$ [4]. We perform three test computations. In all three, at the outflow, we use the OS with $\beta = 0.5$. Two of the test computations...
are with the OS also at the inflow, and we try \( \beta = 0.5 \) and \( \beta = 1.0 \).

\[
\begin{align*}
\frac{x_2}{L} & = p - p_0, \\
\frac{L}{2} & = p_{\text{in}} - p_{\text{out}}.
\end{align*}
\]

Fig. 2: Channel flow with CFP Traction. Problem setup. The parameters appearing in the figure are set as \( L = 1.365 \times 10^{-2} \) m, \( H = 1.3 \times 10^{-2} \) m, \( p_{\text{in}} = 1.313 \) Pa, and \( p_{\text{out}} = 0 \).

Figure 2 shows the problem setup. The density and viscosity are set as \( \rho = 1,050 \) kg/m\(^3\) and \( \mu = 4.2 \times 10^{-3} \) Pa s. The channel walls have no-slip conditions. The traction conditions at the inflow and outflow are based on \( p_{\text{in}} \) and \( p_{\text{out}} \) given in Figure 2. Quadratic NURBS meshes are used in the computations. Figure 3 shows the meshes used with the OS and CFP at the inflow. The two meshes are identical everywhere in the domain other than the part covered by the special-purpose element in the CFP mesh. The number of control points and elements are 9,472 and 8,820 for the OS mesh, and 9,033 and 8,401 for the CFP mesh. The computations were performed with the ST-VMS. The time-step size was \( 8.6 \times 10^{-3} \) s.

![Fig. 3: Channel flow with CFP Traction. Meshes used with the OS and CFP at the inflow. The thin lines are the element boundaries. The two meshes are identical everywhere in the domain other than the part covered by the special-purpose element in the CFP mesh.](image)

The maximum value in the parabolic velocity profile of the analytical solution is \( 4.84 \times 10^{-2} \) m/s. The Reynolds number, defined based on that, is 157. Figure 4 shows the velocity from all three test computations. Figure 5 shows, also from all three computations, the pressure profile obtained by averaging over the channel cross-section. It is clear that the CFP Traction performs very well.

![Fig. 4: Channel flow with CFP Traction. Velocity magnitude (m/s) from computations with OS (\( \beta = 0.5 \)), OS (\( \beta = 1.0 \)), and CFP.](image)

![Fig. 5: Channel flow with CFP Traction. Pressure profile obtained by averaging over the channel cross-section. The analytical solution is the red line.](image)

14. Computational example: flow analysis of a bioprosthetic heart valve

This flow computation of a bioprosthetic heart valve (BHV) is from [19]. It is an ST-SI-TC-IGA computation where the BHV and arterial-surface motion come from the FSI solution obtained with a mixed ALE-VMS/IMGA computation [11] in the framework of the FSITICT. In that mixed framework, the arterial surface is tracked (i.e. followed by the mesh) with the ALE-VMS, and the BHV surfaces are captured with the IMGA. The cardiac cycle is \( T = 0.86 \) s. The BHV model is shown in Figure 6. It has three leaflets and a metal frame. In the ALE-
VMS/IMGA computation, the BHV was represented by cubic T-splines, and the arterial surface by quadratic NURBS. Even when the valve should be closed with no gaps between the model surfaces, there were some small gaps. However, due to the nature of the IMGA computations with zero-thickness structures and limited mesh refinement, those gaps were seen as closed.

The objective in the ST-SI-TC-IGA computation was to see the model surfaces accurately, with the gaps closed, and have a high-resolution representation of the flow near those surfaces. An elaborate method was introduced in [19] to close the gaps for the ST-SI-TC-IGA computation and to have a better representation of the flow patterns near the free edges of the valve leaflets. In the ST-SI-TC-IGA computation, both the valve and arterial surfaces were represented by quadratic NURBS. Figure 7 shows the valve NURBS surfaces. Figure 8 shows the artery quadratic NURBS surfaces.

A template mesh with three SIs and three parts (“Part 1,” “Part 2,” and “Part 3”) was created in [19]. Figure 9 shows the SIs and the parts. The mesh has 429,780 control points and and 289,452 elements. Part 1 faces the SIs. It contains the elements that collapse and are reborn as the leaflets move. Its motion...
is created with a special-purpose mesh moving method that takes into account the contact. Part 2 does not change during the leaflet motion. Part 3 is the rest of the mesh, between Part 1, Part 2, and the artery surface. It changes during the computation. Its motion is determined with the nonlinear-elasticity mesh moving method [46, 140, 142, 143] based on the neo-Hookean constitutive model and the mesh-Jacobian-based stiffening [4, 61, 62, 222, 223]. The motion is driven by the motion of Part 1, the valve motion, and the motion of the artery surface. Figure 10 shows the mesh motion.

The valve and the artery surface have no-slip boundary conditions, the outflow boundary has uniform velocity. The flow rate at the inflow is a modified version of the one in the ALE-VMS/IMGA computation. The modification makes sure that during the closed-valve part of the cardiac cycle, the inflow-boundary flow rate and the closed-space volume-change rate match. The inflow velocity corresponding to the modified flow rate, which was the velocity used in the ST-SI-TC-IGA computation, is shown in Figure 11. The computation was performed with the ST-SUPS, and the time-step size was $5.00 \times 10^{-3}$ s. Figures 12 and 13 show the flow patterns during the second cardiac cycle, and Figure 14 shows the corresponding valve wall shear stress (WSS).
Flow analysis of a BHV. Cross-section of the mesh at $t = 0.175, 0.275, 0.315, 0.495, 0.585, 0.590, 0.595, 0.605, 0.635, 0.785$ s. The colors are for differentiating between the NURBS patches. The checkerboard pattern is for differentiating between the elements.

Fig. 11: Flow analysis of a BHV. Inflow velocity corresponding to the modified flow rate, which was the velocity used in the ST-SI-TC-IGA computation.

15. Computational example: FSI analysis of transcatheter aortic valve replacement

This FSI computation of a transcatheter heart valve (THV) is from [99]. Transcatheter aortic valve replacement (TAVR) is a minimally invasive alternative to open-heart valve replacement that has been increasingly used for treating various valvular diseases. The prosthetic aortic valve is deployed using a catheter and is anchored to the aortic annulus, crushing the diseased valve and assuming its function. A successful TAVR procedure depends on the proper anchoring of the THV in the aortic root of the patient. Computational FSI of TAVR offers an effective approach to understand the interaction between the THV and cardiac system and to improve THV designs for pre-operative planning. This is a mixed ALE-VMS/IMGA computation in the framework of the FSITICT, and the IGA is used in discretizing also the structural mechanics part of the FSI problem. To achieve physiological realism, the dynamics of the THV is coupled to the deforming arterial wall and the enclosed blood flow domain. The leaflets are
Fig. 12: Flow analysis of a BHV. Isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude (m/s), at $t = 1.035, 1.135, 1.175, 1.355, 1.445$ s.

Fig. 13: Flow analysis of a BHV. Isosurfaces corresponding to a positive value of the second invariant of the velocity gradient tensor, colored by the velocity magnitude (m/s), at $t = 1.450, 1.455, 1.465, 1.495, 1.645$ s.
modeled using a transversely isotropic material that represents the mechanics of the extracellular matrix with the embedded network of collagen fibers.

A comprehensive TAVR system, the 26 mm CoreValve, was simulated in three stages in order to obtain the radial outward and friction forces between the aortic wall and the THV frame. The first stage is the crimping of THV. This is necessary since THVs are designed to have a diameter larger than that of the aortic root. During the second stage, the deployment procedure, the interaction between the THV and arterial wall is studied as the wall expands and contracts. In the third stage, the FSI simulation is employed to calculate the friction and radial forces. Their ratio is an important factor for determining the THV’s anchoring ability.

Several cardiac cycles are computed until a time-periodic solution is achieved. The TAVR FSI results are shown in Figure 15. Snapshots of the detailed fluid solution fields, strain distribution, and the top view of of the valve during the cardiac cycle were plotted to illustrate the complexity of this dynamic, multiphysics system. During systole, starting at $t = 0.0$ s, the ventricular pressure is larger than the aortic pressure, which causes the valve to open and the strain to start increasing. The leaflets open and contact the frame at $t = 0.06$ s, and the valve fully opens at $t = 0.25$ s. The valve starts to close and reaches the fully closed configuration around $t = 0.38$ s. The fully loaded configuration at $t = 0.52$ s, where the maximum strain occurs, is also shown in the figure. Generally, the highest level of strain occurs during diastole when the leaflets are fully loaded. The friction force magnitude distribution is shown in Figure 16 for the fully open and fully closed configurations. In the fully closed configuration, the magnitude of the friction force is significantly larger compared to the fully opened configuration, especially around the annulus. As the valve closes and the leaflets provide a blockage for the blood flow, the magnitude of the friction force increases to counter the effect of the additional force on the THV. The coefficient of static friction is shown in Figure 17, which is defined by the ratio of the friction force to the radial force. The results indicate that the minimum of the friction coefficient, 0.22, is

![Flow analysis of a BHV. Magnitude of the WSS (Pa) at $t = 1.035, 1.135, 1.175, 1.355, 1.445, 1.450, 1.455, 1.465, 1.495, 1.645$ s. One-third of the valve is transparent.](image)
necessary to anchor the TAVR without migration.

16. Concluding remarks

We have provided an overview of the IGA-based computational-cardiovascular-medicine methods built around the core methods ST-SUPS, ALE-SUPS, RBVMS, ALE-VMS, and ST-VMS.
Friction force (N)
X
Y
Z
0.100
0.075
0.050
0.025
0.000

Fig. 16: The friction force magnitude distribution at the fully-opened configuration (t = 0.25 s) and the fully-closed configuration (t = 0.52 s).

Friction force magnitude distribution at the fully-opened configuration (t = 0.25 s) and the fully-closed configuration (t = 0.52 s).

Fig. 17: Ratio of friction force magnitude to radial force magnitude over a cardiac cycle.

The superior accuracy the IGA discretization brings in both fluid and solid mechanics is in representing both the problem geometry and the variables computed. While IGA mesh generation for complex geometries is significantly more challenging than finite element mesh generation, the accuracy attained makes the effort involved in addressing that challenge worth it. We provided an overview of how the challenge has mostly been overcome with the complex-geometry mesh generation methods CGIMG and NSVGMG or circumvented with the immersed-boundary method IMGA. Quite often, specific classes of problems, such as cardiovascular medicine, require special methods targeting specific classes of computations, and we provided, as an example, an overview of one of such methods. The computational examples we presented for complex-geometry cardiovascular medicine clearly show that the advanced methods introduced have significantly increased the scope of the IGA-based computational analysis in cardiovascular medicine.

Acknowledgment

This work was supported in part by JST-CREST (first author); Grant-in-Aid for Scientific Research (A) 18H04100 from Japan Society for the Promotion of Science (first author); Rice–Waseda research agreement (first author); and Grant-in-Aid for Research Activity Start-up 16K13779 and Grant-in-Aid for Early-Career Scientists 22K17903 from Japan Society for the Promotion of Science (fifth author). The mathematical model and computational method parts of the work were also supported in part by ONR Grant N00014-21-1-2670 (second author); ARO Grant W911NF-17-1-0046 (third author) and Top Global University Project of Waseda University (third author). The second and fourth author acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper.

References


[119] Helgedagsrud, T.A., Bazilevs, Y., Mathisen, K.M., Yan, J., & Oiseth,


[149] Takizawa, K., Tezduyar, T.E., McIntyre, S., Kostov, N., Kolesar, R., & Habluetzel,


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 110 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 served an Associate Editor of ASME Journal of Applied Mechanics and was responsible for the manuscripts on computational fluid mechanics and FSI. 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 2005. He has published over 170 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 Editor of Springer journal Computational Mechanics 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 TEZDUYAR received his PhD from Caltech in 1982, and he is currently the James F. Barbour Professor of Mechanical Engineering at Rice University and Professor in Faculty of Science and Engineering at Waseda 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 260 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 Elsevier journal Computers & Fluids 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 80 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/.

Takuya TERAHARA received his PhD from Waseda University in 2020, and he is currently an Assistant Professor in the Department of Modern Mechanical Engineering at Waseda. He has been conducting computational fluid mechanics research since 2015. He has published 9 Web-of-Science-indexed journal articles on computational fluid and structural mechanics and FSI. More information on Terahara can be found at https://www.jp.tafsm.org/members/tterahara.