A low-distortion mesh moving method based on fiber-reinforced hyperelasticity and optimized zero-stress state

Kenji Takizawa¹ · Tayfun E. Tezduyar^{2,3} · Reha Avsar²

Received: 25 January 2020 / Accepted: 26 February 2020 / Published online: 11 March 2020
© The Author(s) 2020

Abstract

In computation of flow problems with moving boundaries and interfaces, including fluid–structure interaction, moving-mesh methods enable mesh-resolution control near the interface and consequently high-resolution representation of the boundary layers. Good moving-mesh methods require good mesh moving methods. We introduce a low-distortion mesh moving method based on fiber-reinforced hyperelasticity and optimized zero-stress state (ZSS). The method has been developed targeting isogeometric discretization but is also applicable to finite element discretization. With the large-deformation mechanics equations, we can expect to have a unique mesh associated with each step of the boundary or interface motion. With the fibers placed in multiple directions, we stiffen the element in those directions for the purpose of reducing the distortion during the mesh deformation. We optimize the ZSS by seeking orthogonality of the parametric directions, by mesh relaxation, and by making the ZSS time-dependent as needed. We present 2D and 3D test computations with isogeometric discretization. The computations show that the mesh moving method introduced performs well.

Keywords Mesh moving method · Low distortion · Fiber-reinforced hyperelasticity · Optimized zero-stress state · Mesh relaxation · Isogeometric discretization

1 Introduction

For computation of flow problems with moving boundaries and interfaces (MBI), including fluid–structure interaction (FSI), we introduce a low-distortion mesh moving method based on fiber-reinforced hyperelasticity and optimized zero-stress state (ZSS). With the large-deformation mechanics equations, we can expect to have a unique mesh associated with each step of the boundary motion. The method has been developed targeting isogeometric discretization but is also applicable to finite element discretization. To show how the

method introduced performs, we present 2D and 3D test computations with isogeometric discretization.

1.1 Moving-mesh methods

1.1.1 Key features

In moving-mesh (interface-tracking) methods, as the interface moves and the fluid mechanics domain changes its shape, the mesh moves to adjust to the shape change and to follow (i.e. “track”) the interface. Moving the fluid mechanics mesh to follow a fluid–solid interface enables us to control the mesh resolution near the interface, have high-resolution representation of the boundary layers, and obtain accurate solutions in such critical flow regions. These desirable features do not come easily or do not come at all with the nonmoving-mesh (interface-capturing) methods. In these methods, the interface geometry is somehow represented over a nonmoving fluid mechanics mesh, with more accuracy in some methods than in some others, but the key point is that the fluid mechanics mesh does not move to follow the interface. Because the mesh is not following the interface, independent of how accurately the interface geometry is rep-

✉ Kenji Takizawa
Kenji.Takizawa@tafsm.org

Tayfun E. Tezduyar
tezduyar@tafsm.org

¹ Department of Modern Mechanical Engineering, Waseda University, 3-4-1 Ookubo, Shinjuku-ku, Tokyo 169-8555, Japan

² Mechanical Engineering, Rice University – MS 321, 6100 Main Street, Houston, TX 77005, USA

³ Faculty of Science and Engineering, Waseda University, 3-4-1 Ookubo, Shinjuku-ku, Tokyo 169-8555, Japan

resented, the boundary layer resolution will be limited by the fluid mechanics mesh resolution where the interface is.

1.1.2 Mixed moving-mesh/nonmoving-mesh methods

As mentioned in [1], one of course recognizes that certain classes of interfaces (such as free-surface and two-fluid flows with splashing) might be too complex to handle with an interface-tracking technique and, therefore, for all practical purposes, require an interface-capturing technique. The Mixed Interface-Tracking/Interface-Capturing Technique (MITICT) [2] was introduced in 2000 for computation of MBI problems that involve both fluid–solid interfaces that can be accurately tracked with a moving-mesh method and fluid–fluid interfaces that are too complex or unsteady to be tracked. Such fluid–fluid interfaces are captured over the mesh tracking the fluid–solid interfaces. Thinking similarly about MBI problems with an actual contact or other topology change, the Fluid–Solid Interface-Tracking/Interface-Capturing Technique (FSITICT) was introduced in [3] as the FSI version of the MITICT. In the FSITICT, we track the interface we can with a moving mesh, and capture over that moving mesh the interfaces we cannot track, specifically the interfaces where we need to have an actual contact between the solid surfaces.

1.1.3 Space–time (ST) methods

The Deforming-Spatial-Domain/Stabilized ST (DSD/SST) method [4–6], introduced for computation of flows with MBI, including FSI, is a moving-mesh method, possessing the associated desirable features. The stabilization components of this original DSD/SST are the Streamline-Upwind/Petrov-Galerkin (SUPG) [7] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [4] stabilizations, and therefore it is now called “ST-SUPS.” The ST Variational Multiscale (ST-VMS) method [8–10], which subsumes its precursor ST-SUPS is the VMS version of the DSD/SST. The VMS components of the ST-VMS are from the residual-based VMS (RBVMS) method [11–14]. 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 [8,9]), are desirable also in computations without MBI.

1.1.4 ALE-SUPS and ALE-VMS methods and classes of problems computed

In the moving-mesh category, the DSD/SST is an alternative to the Arbitrary Lagrangian–Eulerian (ALE) method, which is older (see, for example, [15]) and more commonly used. The ALE-VMS method [1,16–21] is the VMS version of the

ALE. It succeeded the ST-SUPS and ALE-SUPS [22] and preceded the ST-VMS. To increase their scope and accuracy, the ALE-VMS and RBVMS are often supplemented with special methods, such as those for weakly-enforced Dirichlet boundary conditions [23–25] and “sliding interfaces” [26,27]. 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 [22], wind-turbine aerodynamics and FSI [28–38], more specifically, vertical-axis wind turbines [37–40], floating wind turbines [41], wind turbines in atmospheric boundary layers [36–38,42], and fatigue damage in wind-turbine blades [43], patient-specific cardiovascular fluid mechanics and FSI [16,44–49], biomedical-device FSI [50–55], ship hydrodynamics with free-surface flow and fluid–object interaction [56,57], hydrodynamics and FSI of a hydraulic arresting gear [58,59], hydrodynamics of tidal-stream turbines with free-surface flow [60], passive-morphing FSI in turbomachinery [61], bioinspired FSI for marine propulsion [62,63], bridge aerodynamics and fluid–object interaction [64–66], and mixed ALE-VMS/Immersogeometric computations [53–55,67,68] in the framework of the FSITICT. Recent advances in stabilized and multiscale methods may be found for stratified incompressible flows in [69], for divergence-conforming discretizations of incompressible flows in [70], and for compressible flows with emphasis on gas-turbine modeling in [71].

1.1.5 Classes of problems computed with the ST-SUPS and ST-VMS

The ST-SUPS and ST-VMS have also been applied to many classes of FSI, MBI and fluid mechanics problems (see [72] for a comprehensive summary). The classes of problems include spacecraft parachute analysis for the landing-stage parachutes [1,73–76], cover-separation parachutes [77] and the drogue parachutes [78–80], wind-turbine aerodynamics for horizontal-axis wind-turbine rotors [1,28,81,82], full horizontal-axis wind-turbines [34,83–85] and vertical-axis wind-turbines [37,38,86], flapping-wing aerodynamics for an actual locust [1,87–89], bioinspired MAVs [84,85,90,91] and wing-clapping [92,93], blood flow analysis of cerebral aneurysms [84,94], stent-blocked aneurysms [94–96], aortas [97–101], heart valves [85,92,99,101–106], ventricle-valve-aorta sequences [107], and coronary arteries in motion [108], spacecraft aerodynamics [77,109], thermo-fluid analysis of ground vehicles and their tires [10,103], thermo-fluid analysis of disk brakes [110], flow-driven string dynamics in turbomachinery [111–113], flow analysis of turbocharger turbines [114–118], flow around tires with road contact and deformation [103,119–122], fluid films [122,123], ram-air parachutes [124], and compressible-flow spacecraft parachute aerodynamics [125,126].

1.2 Special ST methods

A number of special ST methods have been developed in conjunction with the ST-SUPS and ST-VMS to increase their accuracy and scope. They are the ST Slip Interface (ST-SI) method [86,110], ST Topology Change (ST-TC) method [92,102], ST Isogeometric Analysis (ST-IGA) [8,87,114], and their integration as the ST-SI-TC [119], ST-SI-IGA [114], and ST-SI-TC-IGA [104,105].

1.2.1 ST-SI

The ST-SI was introduced in [86], in the context of incompressible-flow equations, to retain the desirable moving-mesh features of the ST-VMS and ST-SUPS in computations involving 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, while the mesh on the other side of the SI remains unaffected. This is accomplished by adding to the ST-VMS formulation interface terms similar to those in the version of the ALE-VMS for computations with sliding interfaces [26,27]. The interface terms account for the compatibility conditions for the velocity and stress at the SI, accurately connecting the two sides of the solution. An ST-SI version where the SI is between fluid and solid domains was also presented in [86]. The SI in that 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 [110] 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 [37,38,86], thermo-fluid analysis of disk brakes [110], flow-driven string dynamics in turbomachinery [111–113], flow analysis of turbocharger turbines [114–118], flow around tires with road contact and deformation [103,119–122], fluid films [122,123], aerodynamic analysis of ram-air parachutes [124], and flow analysis of heart valves [99,101,104–106] and ventricle-valve-aorta sequences [107].

In the ST-SI version presented in [86] 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 [124]. The compressible-flow ST-SI methods were introduced in [125], 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

[125]. These, together with the compressible-flow ST SUPG method [127], 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 [125,126].

1.2.2 ST-TC

The ST-TC [92,102] 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 [92]. 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 [85,92,99,101–106], ventricle-valve-aorta flows [107], wing clapping [92,93], flow around a rotating tire with road contact and prescribed deformation [103,119–122], and fluid films [122,123].

1.2.3 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 [99,101,104–106], ventricle-valve-aorta flows [107], tire aerodynamics with road contact and deformation [103,119–122], and fluid films [122,123].

1.2.4 ST-IGA

The success with IGA basis functions in space [16,26,44,128] motivated the integration of the ST methods with isogeometric discretization, which we broadly call “ST-IGA.” The ST-IGA was introduced in [8]. Computations with the ST-VMS and ST-IGA were first reported in [8] 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 deriving full benefit from using higher-order basis functions in space. This was demonstrated with the stability and accuracy analysis given in [8] 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 [8,9] and demonstrated in [87,88,90]. 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) [87,88,90], with the name coined in [83]. 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” [1,8,9,87] enables 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 [10,103,110–113,129]. The STNMUM and the ST-IGA with IGA basis functions in time have been used in many 3D computations. The classes of problems solved are flapping-wing aerodynamics for an actual locust [1,87–89], bioinspired MAVs [84,85,90,91] and wing-clapping [92,93], separation aerodynamics of spacecraft [77], aerodynamics of horizontal-axis [34,83–85] and vertical-axis [37,38,86] wind-turbines, thermo-fluid analysis of ground vehicles and their tires [10,103], thermo-fluid analysis of disk brakes [110], flow-driven string dynamics in turbomachinery [111–113], flow analysis of turbocharger turbines [114–118], and flow analysis of coronary arteries in motion [108].

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 [114–118], flow-driven string dynamics in turbomachinery [112,113],

ram-air parachutes [124], spacecraft parachutes [126], aortas [99–101], heart valves [99,101,104–106], ventricle-valve-aorta sequences [107], coronary arteries in motion [108], tires with road contact and deformation [120–122], and fluid films [122,123]. Using IGA basis functions in space is now a key part of some of the newest arterial ZSS estimation methods [101,130–135] and related shell analysis [136].

1.2.5 ST-SI-IGA and ST-SI-TC-IGA

The ST-SI-IGA is the integration of the ST-SI and ST-IGA, and the ST-SI-TC-IGA is the integration of the ST-SI, ST-TC and ST-IGA.

The turbocharger turbine flow [114–118] and flow-driven string dynamics in turbomachinery [112,113] were computed with the ST-SI-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 [124] and spacecraft parachute compressible-flow analysis [126] were conducted with the ST-SI-IGA, based on 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 flow analyses [99,101,104–106] and ventricle-valve-aorta flow analysis [107] were conducted with the ST-SI-TC-IGA. The method, 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 [120–122], 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.

In computational analysis of fluid films [122,123], the ST-SI-TC-IGA enables solution with a computational cost comparable to that of the Reynolds-equation model for the comparable solution quality [123]. With that, narrow gaps associated with the road roughness [122] can be accounted for in the flow analysis around tires.

An SI provides mesh generation flexibility in a general context by accurately connecting the two sides of the solution computed over nonmatching meshes. This type of mesh generation flexibility is especially valuable in complex-geometry flow computations with isogeometric discretization, removing the matching requirement between the NURBS patches without loss of accuracy. This feature was used in the flow analysis of heart valves [99,101,104,105], ventricle-valve-aorta sequences [107], turbocharger turbines [114–118], and spacecraft parachute compressible-flow analysis [126].

1.3 Moving-mesh methods are worth the effort to make them work

It is quite evident from Sects. 1.1.4, 1.1.5 and 1.2 that moving-mesh methods have been practical in more classes of complex FSI and MBI problems than commonly thought of, and, with the increased scope provided by the special methods, the ST methods can now do even more. In many classes of computations, the moving-mesh methods are worth the effort to make them work.

1.4 Mesh moving methods

Good moving-mesh methods require good mesh moving methods, and that has to be part of the effort to make the moving-mesh methods work.

1.4.1 Mesh moving and remeshing

In a more general context, moving-mesh methods require mesh update methods. Mesh update has two components: moving the mesh for as long as it is possible, which is the core component, and full or partial remeshing (i.e. generating a new set of elements, and sometimes also a new set of nodes) when the element distortion becomes too high. The key objectives of a mesh moving method should be to maintain the element quality near solid surfaces and to minimize

remeshing frequency. Provided that at the boundaries and interfaces the normal velocity of the mesh matches the normal velocity of the fluid, we can move the mesh any way we find most suitable for meeting those objectives.

1.4.2 Mesh moving based on linear elasticity and mesh-Jacobian-based stiffening

In the mesh moving method introduced in 1992 [137–139], the motion of the internal nodes is determined by solving the equations of linear elasticity. As the boundary condition, the normal velocity of the mesh at the boundaries and interfaces is specified to match the velocity of the fluid. The mesh deformation is dealt with selectively based on the sizes of the elements. Selective treatment based on element sizes is attained by altering the way we account for the Jacobian of the transformation from the element domain to the physical domain. The objective is to stiffen the smaller elements, which are typically placed near solid surfaces, more than the larger ones. When this method was first introduced in [137–139], it consisted of simply dropping Jacobian from the finite element formulation of the mesh moving (elasticity) equations. This results in the smaller elements being stiffened more than the larger ones. The method was given the name “Jacobian-based stiffening” (JBS) in [140]. It was also augmented in [140] to a more extensive kind by introducing a stiffening power χ that determines the degree by which the smaller elements are rendered stiffer than the larger ones. This approach, when $\chi = 1$, would be identical to the method first introduced in [137], and most of the time $\chi = 1$. To also clarify that the “Jacobian” in the name of the method is the mesh Jacobian, in [107] the method was renamed “Mesh-Jacobian-based stiffening” (MJBS).

1.4.3 Solid-Extension Mesh Moving Technique (SEMMT)

In dealing with fluid–solid interfaces, sometimes we need to place thin layers of elements near the solid surfaces to fully control the mesh resolution there and have more accurate representation of the boundary layers. In the mesh moving method introduced in [137–139], such layers of elements move “glued” to the solid objects, undergoing rigid-body motion. No equations are solved for the motion of the nodes in these layers, because these nodal displacements are not governed by the equations of elasticity. This results in some cost reduction. But, more importantly, we have full control of the mesh resolution in these layers. Examples of automatic mesh moving combined with thin layers of elements undergoing rigid-body motion with solid objects were reported as early as 1993 (see [138,139]). Earlier examples of element layers undergoing rigid-body motion, in combination with deforming structured meshes, were reported in 1990 [4]. In computation of flows with fluid–solid interfaces where

the solid is deforming, the motion of the fluid mesh near the interface cannot be represented by a rigid-body motion. Depending on the deformation mode of the solid, the mesh moving method described in Sect. 1.4.2 may need to be used. In such cases, the thin layers of elements near the solid surface become a challenge for the mesh moving method.

The SEMMT was proposed in 2001 [141]. In the SEMMT [1, 141–144], the thin layers of elements are treated almost like an extension of the solid elements. In solving the equations of elasticity governing the motion of the fluid nodes, higher stiffness is assigned to the thin layers of elements compared to the other fluid elements. Two ways of accomplishing this were proposed in [141]: solving the elasticity equations for the nodes connected by the thin layers of elements separate from the elasticity equations for the other nodes, or together. If they are solved separately, for the thin layers of elements, as boundary conditions at the interface with the other elements, traction-free conditions would be used. The separate solution option is referred to as “SEMMT – Multiple Domain (SEMMT-MD),” and the unified solution option as “SEMMT – Single Domain (SEMMT-SD).” In [1, 143–145], test computations were presented to demonstrate how the SEMMT works as part of the mesh update method used with the ST-SUPS method. Both SEMMT options described above were employed. The tests included mesh deformation tests [1, 143–145] and a 2D FSI model problem [143, 144].

1.4.4 Mesh moving based on large-deformation mechanics

In computing the mesh motion from time level t_n to t_{n+1} with the linear-elasticity equations, typically the configuration at t_n is where we compute the displacement from. That works for most problems, but, as mentioned in [107], there are two closely related drawbacks: i) the method is path-dependent, ii) once elements accumulate in some region, it is hard to undo that. The path-dependence leads to non-cyclic results in FSI computations that we expect to have cyclic or near-cyclic results. While the “back-cycle-based mesh moving” (BCBMM) method introduced in [107] can remedy that, moving the mesh based on large-deformation mechanics equations provides a more general solution. That was the mesh moving method used in a number of recent FSI and other MBI computations with the ST-SUPS and ST-VMS (see, for example, [10, 75, 85, 92, 97]). Additional good features of this method are, as pointed out in [75], having many constitutive models to choose from and being able to define the ZSS locally in arbitrary orientations. The constitutive model was St. Venant–Kirchhoff in [75], and neo-Hookean in [10, 85, 92, 97]. The MJBS can still be part of the mesh moving method, and it was in the computations reported in [10, 97].

1.4.5 Element-based mesh relaxation (EBMR)

The EBMR was introduced in 2013 [75]. It restores, without resorting to remeshing, the mesh integrity lost during the mesh motion. The loss of mesh integrity, though not frequent because of the advanced mesh moving methods used with the ST-SUPS and ST-VMS, can happen in FSI computations with a high level of complexity. A recent example of such complexity is FSI computation of clusters of spacecraft parachutes with modified geometric porosity [74–76, 78, 79, 84]. When faced with a loss of mesh integrity, it was proposed in [75] to use the EBMR, where the mesh is “relaxed” without altering the mesh at the fluid–structure interface and thus the mesh integrity is somewhat restored. As commented in [75], this is of course a less costly and less disruptive alternative to remeshing.

In the EBMR, the new mesh has the same number of nodes and elements as before, but some of the nodes are moved slightly to improve the quality of some of the elements. The motion is determined by using the large-deformation mechanics equations and an element-based ZSS (EBZSS). This is essentially a shape generated for each element. By design, the undeformed shape, made of “target elements,” is the shape we want to obtain from solving the solid mechanics equations. There are several options for constructing the target element shapes, and those options can be found in [75]. The EBMR was successfully used in FSI computation of clusters of spacecraft parachutes with modified geometric porosity (see [75]).

1.5 Locally-defined ZSS

Locally-defined ZSS originated as arterial ZSS estimation (see [101, 130–135]). It was formulated first as the EBZSS in the finite element discretization context [130, 131], then as the EBZSS in the isogeometric discretization context [132, 133], and then as the integration point based ZSS (IPBZSS) in the isogeometric discretization context [134, 135].

1.5.1 EBZSS

In the EBZSS, we define the ZSS with a set of positions for each element. Positions of nodes (control points) from different elements mapping to the same node in the mesh do not have to be the same. In the reference configuration, all elements are connected by nodes, and we measure the displacement from that connected configuration. This way of formulating the structural mechanics problem was named in [130] “element-based total Lagrangian (EBTL)” method. The EBTL was used in the EBMR [75]. We explain more in Sect. 2.1.

1.5.2 IPBZSS

In the IPBZSS, we extended the way we define EBZSS to integration-point counterpart of that. The ZSS is represented in terms of the metric tensor. Converting the EBZSS representation to IPBZSS representation is straightforward and the conversion will be exact. Converting the IPBZSS representation to EBZSS representation will, in general, not be exact because the IPBZSS has more parameters than the EBZSS. Here we name this way of formulating the structural mechanics problem “integration-point-based total Lagrangian (IPBTL)” method. We explain more in Sect. 2.2.

1.6 Stabilization parameters and their mesh sensitivity

The ST-SUPS, ALE-SUPS, RBVMS, ALE-VMS and ST-VMS all have some embedded stabilization parameters that play a significant role (see [1]). These parameters involve a measure of the local length scale (quite often called “element length”) and other parameters such as the element Reynolds and Courant numbers. The interface terms in the ST-SI also involve the local length scale, in the direction normal to the interface. There are many ways of defining the stabilization parameters. Some of the newer options for the stabilization parameters used with the SUPS and VMS can be found in [10, 82, 83, 86, 87, 121, 146–149]. Some of the earlier stabilization parameters used with the SUPS and VMS were also used in computations with other SUPG-like methods, such as the computations reported in [61, 150–161].

Because the stabilization parameters depend on the local length scale, we expect them to have increased mesh sensitivity near the solid surfaces as the mesh moves, because the elements there are typically very thin in the direction normal to the surface. This is another reason to have a good mesh moving method, with the thin layers of elements near solid surfaces shielded from distortion. The SEMMT (see Sect. 1.4.3) accomplishes that to a large extent.

1.7 Mesh moving based on fiber-reinforced hyperelasticity and optimized ZSS

In the mesh moving method we introduce in this article, with the fibers placed in multiple directions, we stiffen the element in those directions for the purpose of reducing the distortion during the mesh deformation. We optimize the ZSS by seeking orthogonality of the parametric directions, by mesh relaxation, and by making the ZSS time-dependent as needed.

1.8 Test computations

We first conduct 2D test computations with a flow domain containing a beam undergoing three different modes of

motion: translation, rotation and bending. Then we do 2D test computations with a rectangular domain undergoing compression and expansion. The 3D tests are with a box-shaped flow domain containing a square block undergoing torsion.

1.9 Outline of the remaining sections

In Sect. 2, we provide the large-deformation mechanics framework that we use in the mesh moving method we are introducing. We describe the fiber-reinforced hyperelasticity model in Sect. 3, and introduce the mesh moving method in Sect. 4. The test computations are presented in Sect. 5, and the concluding remarks in Sect. 6. The MJBS, which we also use in the test computations for comparison purposes, is described in Appendix A.

2 Large-deformation mechanics

We provide, mostly from [134], the large-deformation mechanics framework we will use in the mesh moving method we are introducing here. We note that there can be other ways to reach the same mesh moving functionality.

Let $\Omega_0 \subset \mathbb{R}^{n_{sd}}$ be the material domain in the ZSS, where n_{sd} is the number of space dimensions, and let Γ_0 be its boundary. Let $\Omega_t \subset \mathbb{R}^{n_{sd}}$, $t \in (0, T)$, be the material domain in the deformed state, and let Γ_t be its boundary. The structural mechanics equations based on the total Lagrangian formulation can be written as

$$\int_{\Omega_0} \mathbf{w} \cdot \rho_0 \frac{d^2 \mathbf{y}}{dt^2} d\Omega + \int_{\Omega_0} \delta \mathbf{E} : \mathbf{S} d\Omega - \int_{\Omega_0} \mathbf{w} \cdot \rho_0 \mathbf{f} d\Omega = \int_{(\Gamma_t)_h} \mathbf{w} \cdot \mathbf{h} d\Gamma. \quad (1)$$

Here, $\mathbf{y}$ is the displacement, $\mathbf{w}$ is the virtual displacement, $\delta \mathbf{E}$ is the variation of the Green–Lagrange strain tensor, $\mathbf{S}$ is the second Piola–Kirchhoff stress tensor, ρ_0 is the mass density in the ZSS, $\mathbf{f}$ is the body force per unit mass, and $\mathbf{h}$ is the external stress vector applied on the subset $(\Gamma_t)_h$ of the boundary Γ_t .

2.1 EBZSS

In the EBTL method the ZSS is defined with a set of positions $\mathbf{X}_0^e$ for each element e . Positions of nodes from different elements mapping to the same node in the mesh do not have to be the same. In the reference state, $\mathbf{X}_{REF} \in \Omega_{REF}$, all elements are connected by nodes, and we measure the displacement $\mathbf{y}$ from that connected state. The implementation of the method is simple. The deformation gradient tensor $\mathbf{F}$ is evaluated for each element:

$$\mathbf{F}^e \equiv \frac{\partial \mathbf{x}}{\partial \mathbf{X}_0^e}, \quad (2)$$

$$= \frac{\partial (\mathbf{X}_{\text{REF}} + \mathbf{y})}{\partial \mathbf{X}_0^e}, \quad (3)$$

where $\mathbf{x}$ is the position in the deformed configuration. The deformation gradient tensors for different elements are on different states, but the terms in Eq. (1), including the second term, do not depend on the orientation. Therefore the rest of the process is the same as it is in the total Lagrangian formulation.

2.2 IPBZSS

In the IPBTL method, we extend the way we see $\mathbf{X}_0^e$ to integration-point counterpart of $\mathbf{X}_0^e$. As we did with the EBZSS, we work with the reference domain. With the reference Jacobian

$$J_{\text{REF}} = \det \left(\frac{\partial \mathbf{X}_{\text{REF}}}{\partial \mathbf{X}_0} \right), \quad (4)$$

Eq. (1) can be rearranged as

$$\begin{aligned} \int_{\Omega_{\text{REF}}} \mathbf{w} \cdot \rho_0 \frac{d^2 \mathbf{y}}{dt^2} J_{\text{REF}}^{-1} d\Omega + \int_{\Omega_{\text{REF}}} \delta \mathbf{E} : \mathbf{S} J_{\text{REF}}^{-1} d\Omega \\ - \int_{\Omega_{\text{REF}}} \mathbf{w} \cdot \rho_0 \mathbf{f} J_{\text{REF}}^{-1} d\Omega = \int_{(\Gamma_t)_h} \mathbf{w} \cdot \mathbf{h} d\Gamma. \end{aligned} \quad (5)$$

In our implementation, we use the natural coordinates, with covariant basis vectors

$$\mathbf{g}_\alpha = \frac{\partial \mathbf{x}}{\partial \xi^\alpha}, \quad (6)$$

$$\mathbf{G}_\alpha = \frac{\partial \mathbf{X}}{\partial \xi^\alpha}, \quad (7)$$

where ξ^α is the parametric coordinate, and $\alpha = 1, \dots, n_{\text{pd}}$, with n_{pd} being the number of parametric dimensions. The contravariant basis vectors can be calculated with the metric tensor components as

$$\mathbf{g}^\alpha = g^{\alpha\beta} \mathbf{g}_\beta, \quad (8)$$

$$\mathbf{G}^\alpha = G^{\alpha\beta} \mathbf{G}_\beta, \quad (9)$$

where

$$g_{\alpha\beta} = \mathbf{g}_\alpha \cdot \mathbf{g}_\beta, \quad (10)$$

$$G_{\alpha\beta} = \mathbf{G}_\alpha \cdot \mathbf{G}_\beta, \quad (11)$$

$$[g^{\alpha\beta}] = [g_{\alpha\beta}]^{-1}, \quad (12)$$

$$[G^{\alpha\beta}] = [G_{\alpha\beta}]^{-1}. \quad (13)$$

With those vectors, we can express the deformation gradient tensor:

$$\mathbf{F} = \mathbf{g}_\beta \mathbf{G}^\beta, \quad (14)$$

and the Cauchy–Green deformation tensor:

$$\mathbf{C} = \mathbf{F}^T \cdot \mathbf{F} \quad (15)$$

$$= \mathbf{G}^\alpha \mathbf{g}_\alpha \cdot \mathbf{g}_\beta \mathbf{G}^\beta \quad (16)$$

$$= g_{\alpha\beta} \mathbf{G}^\alpha \mathbf{G}^\beta. \quad (17)$$

The Jacobian, $J = \det \mathbf{F}$, can be expressed as

$$J^2 = \det \mathbf{C}. \quad (18)$$

We can write $\det \mathbf{C}$ as

$$\det \mathbf{C} = \frac{\det [g_{\alpha\beta}]}{\det [G_{\alpha\beta}]}, \quad (19)$$

and from that we obtain

$$J = \left(\frac{\det [g_{\alpha\beta}]}{\det [G_{\alpha\beta}]} \right)^{\frac{1}{2}}. \quad (20)$$

We define the covariant basis vectors corresponding to $\mathbf{X}_{\text{REF}}$:

$$(\mathbf{G}_{\text{REF}})_\alpha = \frac{\partial \mathbf{X}_{\text{REF}}}{\partial \xi^\alpha}, \quad (21)$$

and the components of the metric tensor are

$$(G_{\text{REF}})_{\alpha\beta} = (\mathbf{G}_{\text{REF}})_\alpha \cdot (\mathbf{G}_{\text{REF}})_\beta. \quad (22)$$

In the process leading to Eq. (20), we replace $\mathbf{g}_\alpha$ and $\mathbf{g}_\beta$ with $(\mathbf{G}_{\text{REF}})_\alpha$ and $(\mathbf{G}_{\text{REF}})_\beta$ and obtain an alternative to the expression given by Eq. (4):

$$J_{\text{REF}} = \left(\frac{\det [(G_{\text{REF}})_{\alpha\beta}]}{\det [G_{\alpha\beta}]} \right)^{\frac{1}{2}}. \quad (23)$$

The Green–Lagrange strain tensor,

$$\mathbf{E} = \frac{1}{2} (\mathbf{C} - \mathbf{I}), \quad (24)$$

where $\mathbf{I}$ is the identity tensor, can be expressed with the contravariant basis vectors as

$$\mathbf{E} = \frac{1}{2} (g_{\alpha\beta} - G_{\alpha\beta}) \mathbf{G}^\alpha \mathbf{G}^\beta. \quad (25)$$

The second Piola–Kirchhoff tensor can be expressed with the covariant basis vectors as

$$\mathbf{S} = S^{\alpha\beta} \mathbf{G}_\alpha \mathbf{G}_\beta, \quad (26)$$

where $S^{\alpha\beta}$ can be expressed with the components of the metric tensors. Thus, the inner product $\delta \mathbf{E} : \mathbf{S}$, and all the other quantities in the weak form given by Eq. (5), can be evaluated without actually using the basis vectors $\mathbf{G}_\alpha$.

Based on that, we rearrange the formulation and obtain the “integration-point-based updated Lagrangian (IPBUL)” method:

$$\begin{aligned} \int_{\Omega_t} \mathbf{w} \cdot \rho \frac{d^2 \mathbf{x}}{dt^2} d\Omega + \int_{\Omega_t} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma} d\Omega \\ - \int_{\Omega_t} \mathbf{w} \cdot \rho \mathbf{f} d\Omega = \int_{(\Gamma_t)_h} \mathbf{w} \cdot \mathbf{h} d\Gamma, \end{aligned} \quad (27)$$

where $\rho = \rho_0 J^{-1}$,

$$\boldsymbol{\varepsilon}(\mathbf{w}) = \frac{1}{2} \left(\nabla \mathbf{w} + (\nabla \mathbf{w})^T \right), \quad (28)$$

and

$$\boldsymbol{\sigma} = J^{-1} \mathbf{F} \cdot \mathbf{S} \cdot \mathbf{F}^T \quad (29)$$

$$= J^{-1} S^{\alpha\beta} \mathbf{g}_\alpha \mathbf{g}_\beta. \quad (30)$$

We note that we obtained the second term in Eq. (27) from

$$\delta \mathbf{E} : \mathbf{S} = \frac{1}{2} \left(\frac{\partial \mathbf{w}}{\partial \xi^\alpha} \cdot \mathbf{g}_\beta + \mathbf{g}_\alpha \cdot \frac{\partial \mathbf{w}}{\partial \xi^\beta} \right) S^{\alpha\beta} \quad (31)$$

$$= \frac{1}{2} \left(\frac{\partial \mathbf{w}}{\partial \xi^\alpha} \cdot \mathbf{g}_\beta + \mathbf{g}_\alpha \cdot \frac{\partial \mathbf{w}}{\partial \xi^\beta} \right) S^{\alpha\beta} \quad (32)$$

$$= \frac{1}{2} \left(\mathbf{g}^\gamma \frac{\partial \mathbf{w}}{\partial \xi^\gamma} + \frac{\partial \mathbf{w}}{\partial \xi^\gamma} \mathbf{g}^\gamma \right) : \mathbf{g}_\alpha \mathbf{g}_\beta S^{\alpha\beta} \quad (33)$$

$$= \boldsymbol{\varepsilon}(\mathbf{w}) : \mathbf{g}_\alpha \mathbf{g}_\beta S^{\alpha\beta} \quad (34)$$

and $d\Omega_t = J^{-1} d\Omega_0$.

3 Fiber-reinforced hyperelasticity

3.1 Hyperelasticity model

Here we consider a compressible-material hyperelasticity model, where the volumetric part of the strain energy is represented separately. The general strain-energy-density function can be written as

$$\varphi(\mathbf{C}) = \varphi_{\text{iso}}(\bar{\mathbf{C}}) + \varphi_{\text{vol}}(J), \quad (35)$$

Fig. 1 The strain-energy-density dependence on the volume

where

$$\bar{\mathbf{C}} = J^{-\frac{2}{3}} \mathbf{C}, \quad (36)$$

with φ_{iso} and φ_{vol} representing the isochoric and volumetric parts.

The isochoric part can be expressed with the invariants:

$$\bar{I}_1 = \mathbf{I} : \bar{\mathbf{C}} + 3 - n_{\text{pd}}, \quad (37)$$

$$\bar{I}_2 = \frac{1}{2} \left((\mathbf{I} : \bar{\mathbf{C}})^2 - \mathbf{I} : (\bar{\mathbf{C}} \cdot \bar{\mathbf{C}}) \right) + 3 - n_{\text{pd}}, \quad (38)$$

making $\varphi_{\text{iso}} = \varphi_{\text{iso}}(\bar{I}_1, \bar{I}_2)$. In most models, $\varphi_{\text{iso}} = \varphi_{\text{iso}}(\bar{I}_1)$.

Remark 1 The last two terms on the right-hand-side of Eqs. (37) and (38) are to make the invariant calculations consistent with the plane-strain assumption in 2D computations.

For the volumetric part, we use the following expression:

$$\varphi_{\text{vol}} = \kappa_B \beta_B^{-2} (\beta_B \ln J + J^{-\beta_B} - 1), \quad (39)$$

where κ_B is the bulk modulus and β_B is a positive number based on [162]. However, we do not see anything wrong with using negative values for β_B , and actually we have been using $\beta_B = -2$ for nearly incompressible materials. We also note that the expression is defined for $\beta_B \rightarrow 0$ and gives

$$\varphi_{\text{vol}} = \kappa_B \frac{1}{2} (\ln J)^2. \quad (40)$$

With $\beta_B < 0$, the material is stiffer in expansion than in shrinkage, and with $\beta_B > 0$, it is stiffer in shrinkage than in expansion (see Fig. 1).

Fig. 2 A square domain. Expansion in the horizontal direction in a volume-preserving fashion. The thick blue line represents the ZSS and the thin red line represents the deformed state. (Color figure online)

Fig. 3 A square domain. Expansion in the horizontal direction in a volume-preserving fashion. An upright-orientation element in the ZS and deformed states and the parametric-coordinate directions

Fig. 4 A square domain. Expansion in the horizontal direction in a volume-preserving fashion. A rotated-orientation element in the ZS and deformed states and the parametric-coordinate directions

3.2 Need for fiber-reinforced hyperelasticity

Consider a square domain expanding horizontally in a volume-preserving fashion (see Fig. 2). How an element deforms, and consequently how the mesh lines change, will depend on how the element is oriented. In Fig. 3, the element has an upright orientation and therefore does not undergo distortion. The mesh lines retain their orthogonality. In Fig. 4, the element has a rotated orientation and therefore undergoes distortion. The mesh lines lose their orthogonality. Clearly, the constitutive model does not see the mesh lines, and keeping the mesh-line orthogonality is not an objective of the model.

3.3 Constitutive model for fiber-reinforced hyperelasticity

The constitutive model is obtained by adding an invariant that represents the fiber. We consider two options for that invariant, which are commonly seen in the literature. In the first option,

$$I_a = \mathbf{C} : \mathbf{a}\mathbf{a} \quad (41)$$

$$= \lambda_a^2, \quad (42)$$

where λ_a is the stretch in the fiber direction (see [163]). In the second option,

$$\bar{I}_a = \bar{\mathbf{C}} : \mathbf{a}\mathbf{a} \quad (43)$$

$$= \bar{\lambda}_a^2, \quad (44)$$

where $\bar{\lambda}_a$ is the stretch associated with $\bar{\mathbf{C}}$. With the index k being the fiber-direction counter, we write

$$I_{2k+2} = I_{a_k} \quad (45)$$

and

$$\bar{I}_{2k+2} = \bar{I}_{a_k}. \quad (46)$$

Remark 2 We note that there are other invariants [164], such as

$$\bar{I}_{2k+3} = (\mathbf{C} \cdot \mathbf{C}) : \mathbf{a}_k \mathbf{a}_k, \quad (47)$$

but we do not consider them here.

With n_{fib} fiber directions, we express the strain-energy-density function with

$$\varphi(\mathbf{C}) = \varphi_{\text{iso}}(\bar{I}_1, \bar{I}_2) + \sum_{k=1}^{n_{\text{fib}}} \varphi_{\text{ani}}(\bar{I}_1, \bar{I}_2, \bar{I}_{2k+2}) + \varphi_{\text{vol}}(J), \quad (48)$$

or with its slightly modified version:

$$\begin{aligned} \varphi_{\text{mod}}(\mathbf{C}) &= \varphi_{\text{iso}}(\bar{I}_1, \bar{I}_2) \\ &+ \sum_{k=1}^{n_{\text{fib}}} \varphi_{\text{ani}}(\bar{I}_1, \bar{I}_2, I_{2k+2}) + \varphi_{\text{vol}}(J). \end{aligned} \quad (49)$$

We use the neo-Hookean model for the isotropic part:

$$\varphi_{\text{iso}}(I_1, I_2) = \frac{\mu}{2} (I_1 - 3), \quad (50)$$

where μ is the shear modulus. The anisotropic part, from [165], is given as

$$\varphi_{\text{ani}}(I_1, I_2, I_a) = C_1 \left(\exp \left(C_2 (\kappa I_1 + (1 - 3\kappa)I_a - 1)^2 \right) - 1 \right), \quad (51)$$

where C_1 is a stress-like coefficient, and C_2 and $0 \leq \kappa < \frac{1}{3}$ are nondimensional parameters. When $\kappa = \frac{1}{3}$, the model becomes an isotropic Fung-type model.

4 Mesh-moving method

Here we explain the method and options for the mesh moving. We assume that the initial mesh we begin the mesh relaxation with has reasonable resolution and aspect ratio. Most of the time we create thin layers of elements near the solid surfaces and we want the mesh moving method to retain such features.

4.1 Mesh relaxation method

The objective of the mesh relaxation is to have a better quality mesh and to have an equilibrium state with the optimized ZSS, boundary conditions and the constitutive law. We let $\mathbf{x}_0$ denote our initial mesh. We evaluate the corresponding metric tensor at each integration point from the expressions given by Eqs. (6) and (10). When the parametric directions are orthogonal, the metric tensor does not have off-diagonal terms. As the simplest choice, we just remove the off-diagonal terms, and write the ZSS as

$$G_{\alpha\beta} = g_{\alpha\beta} \delta_{\alpha\beta} \quad (\text{no sum}). \quad (52)$$

We write the volume-preserving version of that as

$$G_{\alpha\beta} = \left(\frac{\det[g_{\gamma\delta}]}{\prod_{\kappa=1}^{n_{\text{pd}}} g_{\kappa\kappa}} \right)^{\frac{1}{n_{\text{pd}}}} g_{\alpha\beta} \delta_{\alpha\beta} \quad (\text{no sum}). \quad (53)$$

We note that $\sqrt{G_{\alpha\alpha}}/\sqrt{G_{\beta\beta}}$ represents the aspect ratio between the α and β directions.

Let $\mathbf{a} = \mathbf{a}^\alpha \mathbf{G}_\alpha$ denote the fiber direction in the ZSS. We recall that the metric tensor was used in Eqs. (1), (5) and (27) to represent the ZSS, without using the basis vectors $\mathbf{G}_\alpha$. Similarly, $\mathbf{a}^\alpha$ will be used in the constitutive model to represent the fiber direction, without using the basis vectors $\mathbf{G}_\alpha$. We will use $2^{n_{\text{pd}}-1}$ fiber directions, and we recall that the index k is the fiber-direction counter. We propose two options for defining $\mathbf{a}_k^\alpha$. In the first option,

$$\mathbf{a}_k^\alpha = \begin{cases} -1 & \text{if } k-1 = 2^{\alpha-1} \\ 1 & \text{otherwise} \end{cases}. \quad (54)$$

We omit the normalization $\|\mathbf{a}_k\| = 1$. In the second option,

$$\mathbf{a}_k^\alpha = \frac{1}{\sqrt{G_{\alpha\alpha}}} \begin{cases} -1 & \text{if } k-1 = 2^{\alpha-1} \\ 1 & \text{otherwise} \end{cases} \quad (\text{no sum}). \quad (55)$$

Beginning with $\mathbf{x} = \mathbf{x}_0$, metric tensor $G_{\alpha\beta}$, reinforcement fibers, and appropriate boundary conditions, including the prescribed boundary positions, we carry out a steady-state structural mechanics computation based on Eq. (27). The constitutive models and parameters can be defined individually for the elements or mesh regions. To avoid any negative impact of the ZSS modifications given by Eqs. (52) and (53), we put the modification into effect gradually:

$$G_{\alpha\beta} \leftarrow (1-s)g_{\alpha\beta} + sG_{\alpha\beta}. \quad (56)$$

The ramp-up parameter s is increased from 0 to 1 typically in 10 equal increments, with 3–5 nonlinear iterations at each increment to obtain the steady-state solution.

If the mesh quality is not satisfactory, we repeat the process with $\mathbf{x}_0 \leftarrow \mathbf{x}$ or adjust the metric tensor. The key point here is that the mesh $\mathbf{x}$ should be in equilibrium state with the ZSS before we start the mesh moving computation.

4.2 Mesh moving computation

With the starting mesh $\mathbf{x}$, metric tensor $G_{\alpha\beta}$, reinforcement fibers, and the prescribed boundary positions coming from the time-dependent boundary motion, we solve the structural mechanics equations at each time step to compute the mesh positions. Let $\mathbf{b}_n$ and $\mathbf{b}_{n+1}$ be the prescribed boundary positions at time levels n and $n+1$, and let $\mathbf{x}_n$ and $\mathbf{x}_{n+1}$ be the rest of the mesh positions at those time levels. As we move from time level n to $n+1$, sometimes the boundary motion might be too large relative to the thickness of the layers of elements close to the boundary. Simply setting the initial guess in the steady-state computation as

$$\mathbf{b}_{n+1}^0 = \mathbf{b}_{n+1}, \quad (57)$$

$$\mathbf{x}_{n+1}^0 = \mathbf{x}_n, \quad (58)$$

where the superscript is the iteration counter, would very likely lead to convergence difficulties. As an option, we have been setting the initial guess as

$$\mathbf{b}_{n+1}^0 = \mathbf{b}_{n+1}, \quad (59)$$

$$\mathbf{x}_{n+1}^0 = 2\mathbf{x}_n - \mathbf{x}_{n-1}, \quad (60)$$

which requires additional positions $\mathbf{x}_{n-1}$ as input, and even with that it may not work when there are sudden condition changes, such as reversing from an expansion to a compression. Here we propose an iteration sequence $i = 1, 2, \dots, L$,

Table 1 Material properties for the tests with the MJBS

	ν	E	χ
MJBS1	0.3	1	2
MJBS2	0.3	1	1

Table 2 Material properties for tests with the neo-Hookean model (“NH”) and fiber-reinforced hyperelasticity (“FR”)

	κ_B	β_B	μ	C_1	C_2
NH1	10^{-1}	4	2×10^2	0	–
NH2	10^{-3}	0	1	0	–
NH3	10^{-3}	4	2	0	–
FR1	10^{-1}	4	2×10^2	10^2	10^1
FR2	10^{-3}	0	1	0.5	1.0
FR3	10^{-3}	4	2	10^2	10^{-3}

where the prescribed boundary positions are applied gradually with a ramp-up factor $0 < \tilde{\alpha}^1 \leq \dots \leq \tilde{\alpha}^{L-1} \leq \tilde{\alpha}^L = 1$. We form the increment equations by evaluating the residuals and derivatives at

$$\tilde{\mathbf{b}}_{n+1}^i = \mathbf{b}_n + \tilde{\alpha}^i (\mathbf{b}_{n+1}^{i-1} - \mathbf{b}_n), \quad (61)$$

$$\tilde{\mathbf{x}}_{n+1}^i = \mathbf{x}_n + \tilde{\alpha}^i (\mathbf{x}_{n+1}^{i-1} - \mathbf{x}_n), \quad (62)$$

with $\mathbf{b}_{n+1}^0 = \mathbf{b}_{n+1}$ and $\mathbf{x}_{n+1}^0 = \mathbf{x}_n$. With the increment $\Delta \tilde{\mathbf{x}}_{n+1}^i$, we update $\mathbf{b}_{n+1}$ and $\mathbf{x}_{n+1}$ by extrapolation:

$$\mathbf{b}_{n+1}^i = \mathbf{b}_n + \frac{1}{\tilde{\alpha}^i} (\tilde{\mathbf{b}}_{n+1}^i - \mathbf{b}_n) \quad (63)$$

$$= \mathbf{b}_{n+1}, \quad (64)$$

$$\mathbf{x}_{n+1}^i = \mathbf{x}_n + \frac{1}{\tilde{\alpha}^i} (\tilde{\mathbf{x}}_{n+1}^i + \Delta \tilde{\mathbf{x}}_{n+1}^i - \mathbf{x}_n) \quad (65)$$

$$= \mathbf{x}_{n+1}^i + \frac{1}{\tilde{\alpha}^i} \Delta \tilde{\mathbf{x}}_{n+1}^i. \quad (66)$$

We note that for linear mesh moving equations, such as the linear-elasticity equations, the solution is independent of $\tilde{\alpha}^i$.

Here we also introduce the option of changing the ZSS at each time level. For example, we can calculate the domain volume as it changes in time, and scale with that the metric tensor globally or in selected mesh regions, excluding regions near the solid surfaces. That would give us a good mesh-resolution control through φ_{vol} . To avoid any negative impact on the solution, the ZSS change should be gradual.

Fig. 5 2D tests with a beam: translation, rotation and bending. Problem setup

5 Test computations

Here we compare the MJBS (see Appendix A), neo-Hookean model, and fiber-reinforced hyperelasticity. Both hyperelasticity models are based on Eqs. (48), (39), (50) and (51), with $\kappa = 0$. For both hyperelasticity models, the ZSS is based on Eq. (53). The fiber directions are based on Eq. (55). Table 1 shows the material properties for the MJBS, and Table 2 shows the material properties for the hyperelasticity models. We are working with nondimensional numbers. We note that for the hyperelasticity models, the Poisson’s ratio can be seen as

Fig. 6 2D tests with a beam: translation, rotation and bending. The initial mesh $\mathbf{x}_0$. The checkerboard pattern is for differentiating between the NURBS elements, and the colors are for differentiating between the NURBS patches. The lines are control lines, where the control points are at the intersections. (Color figure online)

Fig. 7 2D tests with a beam: translation, rotation and bending. Mesh after the relaxation. Neo-Hookean model (top) and fiber-reinforced hyperelasticity (bottom)

Table 3 2D tests with a beam. Overall mesh orthogonality for the initial mesh and meshes after relaxation with the neo-Hookean model (“NH”) and fiber-reinforced hyperelasticity (“FR”)

Method	L_2 norm	Max norm
$\mathbf{x}_0$	0.7°	14.7°
NH	0.3°	24.7°
HR	0.2°	15.0°

L_2 and max norms of the deviation of the mesh-line angles from $\pi/2$

$$\nu = \frac{3\kappa_B - \mu}{2(3\kappa_B + \mu)}. \quad (67)$$

5.1 2D tests with a beam: translation, rotation and bending

Figure 5 shows the problem setup. There are three deformation patterns: translation, rotation and bending. We use the material properties (MJBS1, NH1, FR1) for two layers of ele-

Fig. 8 2D tests with a beam: translation. MJBS (top), neo-Hookean model (middle) and fiber-reinforced hyperelasticity (bottom). The right frame is the zoomed view

Fig. 9 2D tests with a beam: rotation. MJBS (top), neo-Hookean model (middle) and fiber-reinforced hyperelasticity (bottom). The right frame is the zoomed view

ments on each side of the beam, and the material properties (MJBS2, NH2, FR2) for the rest of the mesh.

Figure 6 shows the initial quadratic NURBS mesh $\mathbf{x}_0$. The number of elements and control points are 3220 and 3626. For the numerical integration, 4×4 quadrature points are used. We do not use mesh relaxation with the MJBS. Figure 7 shows the mesh after the relaxation. Because the relaxations are based on different material models, the two meshes are slightly different. In both cases the mesh-line orthogonality is improved with the relaxation. Table 3 shows the overall mesh-orthogonality statistics in the whole domain, for the initial mesh and meshes after relaxation.

The translation, rotation and bending take place in 500 steps. The number of iterations per step (see Sect. 4.2) is 6, with $[\tilde{\alpha}^i] = [0.01, 0.04, 0.16, 1, 1, 1]$. We note that for the MJBS we have only one iteration per step. Figures 8, 9 and 10 show the mesh at the end of the translation, rotation and bending, respectively. For all the deformation patterns, the MJBS and fiber-reinforced hyperelasticity result in a better mesh than the neo-Hookean model does. Comparing the

Fig. 10 2D tests with a beam: bending. MJBS (top), neo-Hookean model (middle) and fiber-reinforced hyperelasticity (bottom). The right frame is the zoomed view

Table 4 2D tests with a beam. Overall mesh orthogonality at the maximum translation, with the MJBS, neo-Hookean model (“NH”) and fiber-reinforced hyperelasticity (“FR”)

Method	L_2 norm	Max norm
MJBS	9.6°	60.8°
NH	6.2°	84.2°
FR	4.2°	50.9°

L_2 and max norms of the deviation of the mesh-line angles from $\pi/2$

MJBS and fiber-reinforced hyperelasticity, we see that with the MJBS, distortion is more localized, close to the beam, while with the fiber-reinforced hyperelasticity, it spreads more to regions not close to the beam. Tables 4, 5 and 6 show, at the maximum translation, rotation and bending, the overall mesh-orthogonality statistics in the whole domain.

Table 5 2D tests with a beam. Overall mesh orthogonality at the maximum rotation, with the MJBS, neo-Hookean model (“NH”) and fiber-reinforced hyperelasticity (“FR”)

Method	L_2 norm	Max norm
MJBS	10.4°	76.1°
NH	4.7°	86.7°
FR	3.7°	52.8°

L_2 and max norms of the deviation of the mesh-line angles from $\pi/2$

Table 6 2D tests with a beam

Method	L_2 norm	Max norm
MJBS	5.3°	86.2°
NH	2.5°	86.3°
FR	1.8°	53.9°

Overall mesh orthogonality at the maximum bending, with the MJBS, neo-Hookean model (“NH”) and fiber-reinforced hyperelasticity (“FR”). L_2 and max norms of the deviation of the mesh-line angles from $\pi/2$

Fig. 11 2D tests with a rectangular domain undergoing compression and expansion. Domain in its initial (red), compressed (green) and expanded (gold) states. (Color figure online)

5.2 2D tests with a rectangular domain undergoing compression and expansion

Figure 11 shows the rectangular domain. The mesh first undergoes compression, then expansion. We use the material properties (MJBS2, NH3, FR3) for all the elements. Even though the domain is rectangular, we intentionally use a mesh with non-orthogonal mesh lines to see the effect of the relaxation.

Figure 12 shows the initial quadratic NURBS mesh $\mathbf{x}_0$. The number of elements and control points are 2600 and 2928. For the numerical integration, 4×4 quadrature points are used. We do not use mesh relaxation with the MJBS. Fig-

Fig. 12 2D tests with a rectangular domain undergoing compression and expansion. The initial mesh $\mathbf{x}_0$ **Fig. 13** 2D tests with a rectangular domain undergoing compression and expansion. Mesh after the relaxation. Neo-Hookean model (top) and fiber-reinforced hyperelasticity (bottom)**Table 7** 2D tests with a rectangular domain undergoing compression and expansion

Method	L_2 norm	Max norm
$\mathbf{x}_0$	8.8°	51.2°
NH	2.6°	43.5°
FR	2.3°	42.1°

Overall mesh orthogonality for the initial mesh and meshes after relaxation with the neo-Hookean model (“NH”) and fiber-reinforced hyperelasticity (“FR”). L_2 and max norms of the deviation of the mesh-line angles from $\pi/2$

ure 13 shows the mesh after the relaxation. Again, with both models, the mesh-line orthogonality is improved. At points where three patches meet, it is impossible to have orthogonality, the angle between the mesh lines becomes close to $2\pi/3$, and that is the best we can expect. Table 7 shows the overall mesh-orthogonality statistics in the whole domain, for the initial mesh and meshes after relaxation.

The compression takes place in 250 steps, and the expansion in 500 steps beyond that. The number of iterations per step is 6, with $[\tilde{\alpha}^i] = [0.01, 0.04, 0.16, 1, 1, 1]$. We note that for the MJBS we have only one iteration per step. Figures 14 and 15 show the mesh at the end of the compression

Fig. 14 2D tests with a rectangular domain undergoing compression. MJBS (top), neo-Hookean model (middle) and fiber-reinforced hyperelasticity (bottom). Frame in every second position is the zoomed view

and expansion, respectively. In all three cases the mesh quality is reasonably good. When we look closer, we see that the neo-Hookean model and fiber-reinforced hyperelasticity give smoothness across patch boundaries, which is because of targeting mesh-line orthogonality. Tables 8 and 9 show, at

Fig. 15 2D tests with a rectangular domain undergoing expansion. MJBS (top), neo-Hookean model (middle) and fiber-reinforced hyperelasticity (bottom)

the maximum compression and expansion, the overall mesh-orthogonality statistics in the whole domain.

5.3 3D tests with a square block undergoing torsion

Figure 16 shows the box-shaped domain containing the block and the block in its initial and twisted states. The block, with

Table 8 2D tests with a rectangular domain

Method	L_2 norm	Max norm
MJBS	15.0°	55.0°
NH	8.3°	44.8°
FR	8.6°	43.4°

Overall mesh orthogonality at the maximum compression, with the MJBS, neo-Hookean model (“NH”) and fiber-reinforced hyperelasticity (“FR”). L_2 and max norms of the deviation of the mesh-line angles from $\pi/2$

Table 9 2D tests with a rectangular domain

Method	L_2 norm	Max norm
MJBS	4.1°	53.0°
NH	1.9°	46.8°
FR	1.6°	45.3°

Overall mesh orthogonality at the maximum expansion, with the MJBS, neo-Hookean model (“NH”) and fiber-reinforced hyperelasticity (“FR”). L_2 and max norms of the deviation of the mesh-line angles from $\pi/2$

Fig. 16 3D tests with a square block undergoing torsion. Domain containing the block and the block in its initial (red) and twisted (green) states

one end fixed centrally on one of the box surfaces, extends halfway through the box. The opposite end of the block undergoes a 90° rotation about its own center. This motion defines the prescribed boundary positions in computing the rest of the mesh positions in the box. We use the material properties (MJBS1, NH1, FR1) for two layers of elements around the block, and the material properties (MJBS2, NH2, FR2) for the rest of the mesh.

Figure 17 shows the initial quadratic NURBS mesh $\mathbf{x}_0$. The number of elements and control points are 136,000 and 167,728. For the numerical integration, $4 \times 4 \times 4$ quadrature points are used. We do not use mesh relaxation with the MJBS. Figure 18 shows the mesh after the relaxation. Table 10 shows the overall mesh-orthogonality statistics in

Fig. 17 3D tests with a square block undergoing torsion. The initial mesh $\mathbf{x}_0$ **Fig. 18** 3D tests with a square block undergoing torsion. Mesh after the relaxation. Neo-Hookean model (top) and fiber-reinforced hyperelasticity (bottom)

Table 10 3D tests with a square block undergoing torsion

Method	L_2 norm	Max norm
$\mathbf{x}_0$	4.8°	43.1°
NH	0.4°	75.8°
FR	0.1°	35.6°

Overall mesh orthogonality for the initial mesh and meshes after relaxation with the neo-Hookean model (“NH”) and fiber-reinforced hyperelasticity (“FR”). L_2 and max norms of the deviation of the mesh-line angles from $\pi/2$

Fig. 19 3D tests with a square block undergoing torsion. The portion of the initial mesh $\mathbf{x}_0$ used in visualizing the deformed states**Table 11** 3D tests with a square block undergoing torsion

Method	L_2 norm	Max norm
MJBS	8.4°	78.1°
NH	1.2°	88.5°
FR	1.1°	64.2°

Overall mesh orthogonality at the maximum torsion, with the MJBS, neo-Hookean model (“NH”) and fiber-reinforced hyperelasticity (“FR”). L_2 and max norms of the deviation of the mesh-line angles from $\pi/2$

the whole domain, for the initial mesh and meshes after relaxation.

The torsion takes place in 500 steps. The number of iterations per step is 3, with $[\tilde{\alpha}^i] = [1, 1, 1]$. We note that for the MJBS we have only one iteration per step. Figure 19 shows the portion of the initial mesh $\mathbf{x}_0$ used in visualizing the deformed states. Because of the symmetry of the mesh, we will visualize only half of the NURBS patches in that portion. Figure 20 shows the NURBS patches at the end of the torsion. Table 11 shows, at the maximum torsion, the overall mesh-orthogonality statistics in the whole domain. Overall, the behavior tends to be the same as what we observed in the 2D cases.

Fig. 20 3D tests with a square block undergoing torsion. MJBS (top), neo-Hookean model (middle) and fiber-reinforced hyperelasticity (bottom). The right frame is the zoomed view

6 Concluding remarks

We have introduced a low-distortion mesh moving method based on fiber-reinforced hyperelasticity and optimized ZSS. A good mesh moving method is essential in computation of FSI and MBI problems with moving-mesh methods, which enable mesh-resolution control near the interfaces and consequently high-resolution representation of the boundary layers. The mesh moving method, developed targeting isogeometric discretization, is also applicable to finite element discretization. With the large-deformation mechanics equations, we can expect to have a unique mesh associated with each step of the boundary or interface motion. With the fibers placed in multiple directions, the element is stiffened in those directions for the purpose of reducing the distortion during the mesh deformation. The ZSS is optimized by seeking orthogonality of the parametric directions, by mesh relaxation, and by making the ZSS time-dependent as needed. We

conducted both 2D and 3D test computations with isogeometric discretization to demonstrate how the mesh moving method performs. The computations show that the method improves the quality of the initial mesh coming from the mesh generation, giving us a better starting mesh for the mesh motion, and reduces the element distortion experienced during the motion.

Acknowledgements This work was supported (Kenji Takizawa) in part by JST-CREST; Grant-in-Aid for Scientific Research (A) 18H04100 from Japan Society for the Promotion of Science; and Rice–Waseda research agreement. The mathematical model and computational method parts of the work were also supported in part by ARO Grant W911NF-17-1-0046 (Tayfun E. Tezduyar and Reha Avsar) and Top Global University Project of Waseda University (Tayfun E. Tezduyar).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix: MJBS

In the MJBS (see Sect. 1.4.2), the motion of the internal nodes is determined by solving the equations of linear elasticity. As the boundary condition, the normal velocity of the mesh at the boundaries and interfaces is specified to match the velocity of the fluid. With the notation of [1], the formulation can be written as

$$\int_{\Omega_i} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\hat{\mathbf{y}}^h - \hat{\mathbf{y}}_i^h) d\Omega = 0. \quad (68)$$

Here $\hat{\mathbf{y}}^h$ is the displacement from the reference configuration $\mathbf{X}_{\text{REF}}^h$ to the current configuration $\mathbf{x}^h = \mathbf{X}_{\text{REF}}^h + \hat{\mathbf{y}}^h$, and $\hat{\mathbf{y}}_i^h$ is the displacement from $\mathbf{X}_{\text{REF}}^h$ to the $\tilde{t}$ configuration $\mathbf{x}_i^h = \mathbf{X}_{\text{REF}}^h + \hat{\mathbf{y}}_i^h$ we compute the mesh motion from. The most common selection is $\tilde{t} = t_n$ when we are computing the mesh at t_{n+1} . The stress tensor is

$$\boldsymbol{\sigma}(\mathbf{y}) = \frac{E}{1+\nu} \left(\frac{\nu}{1-2\nu} \mathbf{I} + \boldsymbol{\varepsilon}(\mathbf{y}) \right), \quad (69)$$

where E and ν are the Young's modulus and Poisson's ratio.

The mesh deformation is dealt with selectively based on the sizes of the elements. Selective treatment based on element sizes is attained by altering the way we account for the

Jacobian of the transformation from the element domain to the physical domain:

$$J_M = \det \left(\frac{\partial \mathbf{x}_i^h}{\partial \boldsymbol{\xi}} \right), \quad (70)$$

where $\boldsymbol{\xi} = [\xi^\alpha]$, and the subscript “M” is to clarify that this is the mesh Jacobian. The objective is to stiffen the smaller elements, which are typically placed near solid surfaces, more than the larger ones. When this method was first introduced in [137–139], it consisted of simply dropping J_M from Eq. (68). That is equivalent to modifying Eq. (68) as

$$\int_{\Omega_i} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\hat{\mathbf{y}}^h - \hat{\mathbf{y}}_i^h) \left(\frac{J_M}{(J_M)_0} \right)^{-1} d\Omega = 0, \quad (71)$$

where $(J_M)_0$ was originally intended to be a global scaling value and is now seen as a free parameter. The modification results in the smaller elements being stiffened more than the larger ones, and the element stiffening factor is $\frac{(J_M)_0}{J_M}$. The method was augmented in [140] to a more extensive kind by introducing a stiffening power χ that determines the degree by which the smaller elements are rendered stiffer than the larger ones:

$$\int_{\Omega_i} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\hat{\mathbf{y}}^h - \hat{\mathbf{y}}_i^h) \left(\frac{J_M}{(J_M)_0} \right)^{-\chi} d\Omega = 0. \quad (72)$$

This approach, when $\chi = 1$, would be identical to the method first introduced in [137], and most of the time $\chi = 1$.

From earlier tests with different $\tilde{t}$ settings, we know that setting $\tilde{t} = t_n$ gives reasonably good performance for most problems. We believe that is partly because when we compute the mesh motion from a configuration close to the current configuration, J_M provides the method feedback on which elements to protect.

We note that in practical computations the element stiffening method proposed in [166] is approximately equivalent to the MJBS method with $\chi = 1$. To show that, we first approximate $\frac{(J_M)_0}{J_M}$ as

$$\frac{(J_M)_0}{J_M} \approx \frac{V_0}{V}, \quad (73)$$

where V is the element volume and V_0 is a global scaling value. The approximation becomes exact for simplex elements. In [166], the stiffening factor, represented by “ $1 + \tau_m$ ”, is given as

$$1 + \tau_m = 1 + \frac{1 - \frac{V_{\min}}{V_{\max}}}{\frac{V}{V_{\max}}}, \quad (74)$$

where $V_{\min}$ and $V_{\max}$ are the minimum and maximum element volumes. This can also be written as

$$1 + \tau_m = 1 + \frac{V_{\max} - V_{\min}}{V}. \quad (75)$$

In practical computations, $V_{\max} \gg V_{\min}$, and for most of the elements that matter,

$$\frac{V_{\max} - V_{\min}}{V} \gg 1. \quad (76)$$

Therefore

$$1 + \tau_m \approx \frac{V_{\max} - V_{\min}}{V}. \quad (77)$$

Because $V_{\max} - V_{\min}$ is a global scaling value just like $(J_M)_0$ is, the stiffening factor given by Eq. (77) is equivalent to the one given by Eq. (73).

Remark 3 We note that a stiffening-factor expression in terms of the mesh Jacobian gives us the option of evaluating the factor at the integration points, without the need for any precalculation. We also note that such an expression is naturally applicable to isogeometric discretization.

References

- Bazilevs Y, Takizawa K, Tezduyar TE (February 2013) Computational fluid–structure interaction: methods and applications. Wiley, New York. ISBN 978-0470978771
- Tezduyar TE (2001) Finite element methods for flow problems with moving boundaries and interfaces. Arch Comput Methods Eng 8:83–130. <https://doi.org/10.1007/BF02897870>
- Tezduyar TE, Takizawa K, Moorman C, Wright S, Christopher J (2010) Space-time finite element computation of complex fluid–structure interactions. Int J Numer Methods Fluids 64:1201–1218. <https://doi.org/10.1002/fld.2221>
- Tezduyar TE (1992) Stabilized finite element formulations for incompressible flow computations. Adv Appl Mech 28:1–44. [https://doi.org/10.1016/S0065-2156\(08\)70153-4](https://doi.org/10.1016/S0065-2156(08)70153-4)
- Tezduyar TE (2003) Computation of moving boundaries and interfaces and stabilization parameters. Int J Numer Methods Fluids 43:555–575. <https://doi.org/10.1002/fld.505>
- Tezduyar TE, Sathe S (2007) Modeling of fluid–structure interactions with the space-time finite elements: solution techniques. Int J Numer Methods Fluids 54:855–900. <https://doi.org/10.1002/fld.1430>
- Brooks AN, Hughes TJR (1982) Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng 32:199–259
- Takizawa K, Tezduyar TE (2011) Multiscale space-time fluid–structure interaction techniques. Comput Mech 48:247–267. <https://doi.org/10.1007/s00466-011-0571-z>
- Takizawa K, Tezduyar TE (2012) Space-time fluid–structure interaction methods. Math Models Methods Appl Sci 22(supp02):1230001. <https://doi.org/10.1142/S0218202512300013>
- Takizawa K, Tezduyar TE, Kuraishi T (2015) Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires. Math Models Methods Appl Sci 25:2227–2255. <https://doi.org/10.1142/S0218202515400072>
- Hughes TJR (1995) Multiscale phenomena: Green’s functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods. Comput Methods Appl Mech Eng 127:387–401
- Hughes TJR, Oberai AA, Mazzei L (2001) Large eddy simulation of turbulent channel flows by the variational multiscale method. Phys Fluids 13:1784–1799
- Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A, Scovazzi G (2007) Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. Comput Methods Appl Mech Eng 197:173–201
- Bazilevs Y, Akkerman I (2010) Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual-based variational multiscale method. J Comput Phys 229:3402–3414
- Hughes TJR, Liu WK, Zimmermann TK (1981) Lagrangian–Eulerian finite element formulation for incompressible viscous flows. Comput Methods Appl Mech Eng 29:329–349
- Bazilevs Y, Calo VM, Hughes TJR, Zhang Y (2008) Isogeometric fluid–structure interaction: theory, algorithms, and computations. Comput Mech 43:3–37
- Takizawa K, Bazilevs Y, Tezduyar TE (2012) Space-time and ALE-VMS techniques for patient-specific cardiovascular fluid–structure interaction modeling. Arch Comput Methods Eng 19:171–225. <https://doi.org/10.1007/s11831-012-9071-3>
- Bazilevs Y, Hsu M-C, Takizawa K, Tezduyar TE (2012) ALE-VMS and ST-VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction. Math Models Meth Appl Sci 22(supp02):1230002. <https://doi.org/10.1142/S0218202512300025>
- Bazilevs Y, Takizawa K, Tezduyar TE (2013) Challenges and directions in computational fluid–structure interaction. Math Models Methods Appl Sci 23:215–221. <https://doi.org/10.1142/S0218202513400010>
- Bazilevs Y, Takizawa K, Tezduyar TE (2015) New directions and challenging computations in fluid dynamics modeling with stabilized and multiscale methods. Math Models Methods Appl Sci 25:2217–2226. <https://doi.org/10.1142/S0218202515020029>
- Bazilevs Y, Takizawa K, Tezduyar TE (2019) Computational analysis methods for complex unsteady flow problems. Math Models Methods Appl Sci 29:825–838. <https://doi.org/10.1142/S0218202519020020>
- Kalro V, Tezduyar TE (2000) A parallel 3D computational method for fluid–structure interactions in parachute systems. Comput Methods Appl Mech Eng 190:321–332. [https://doi.org/10.1016/S0045-7825\(00\)00204-8](https://doi.org/10.1016/S0045-7825(00)00204-8)
- Bazilevs Y, Hughes TJR (2007) Weak imposition of Dirichlet boundary conditions in fluid mechanics. Comput Fluids 36:12–26
- Bazilevs Y, Michler C, Calo VM, Hughes TJR (2010) Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. Comput Methods Appl Mech Eng 199:780–790
- Hsu M-C, Akkerman I, Bazilevs Y (2012) Wind turbine aerodynamics using ALE-VMS: validation and role of weakly enforced boundary conditions. Comput Mech 50:499–511
- Bazilevs Y, Hughes TJR (2008) NURBS-based isogeometric analysis for the computation of flows about rotating components. Comput Mech 43:143–150
- Hsu M-C, Bazilevs Y (2012) Fluid–structure interaction modeling of wind turbines: simulating the full machine. Comput Mech 50:821–833

28. Bazilevs Y, Hsu M-C, Akkerman I, Wright S, Takizawa K, Henicke B, Spielman T, Tezduyar TE (2011) 3D simulation of wind turbine rotors at full scale. Part I: geometry modeling and aerodynamics. *Int J Numer Methods Fluids* 65:207–235. <https://doi.org/10.1002/fld.2400>
29. Bazilevs Y, Hsu M-C, Kiendl J, Wüchner R, Bletzinger K-U (2011) 3D simulation of wind turbine rotors at full scale. Part II: fluid–structure interaction modeling with composite blades. *Int J Numer Methods Fluids* 65:236–253
30. Hsu M-C, Akkerman I, Bazilevs Y (2011) High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Comput Fluids* 49:93–100
31. Bazilevs Y, Hsu M-C, Scott MA (2012) Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Comput Methods Appl Mech Eng* 249–252:28–41
32. Hsu M-C, Akkerman I, Bazilevs Y (2014) Finite element simulation of wind turbine aerodynamics: validation study using NREL Phase VI experiment. *Wind Energy* 17:461–481
33. Korobenko A, Hsu M-C, Akkerman I, Tippmann J, Bazilevs Y (2013) Structural mechanics modeling and FSI simulation of wind turbines. *Math Models Methods Appl Sci* 23:249–272
34. Bazilevs Y, Takizawa K, Tezduyar TE, Hsu M-C, Kostov N, McIntyre S (2014) Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods. *Arch Comput Methods Eng* 21:359–398. <https://doi.org/10.1007/s11831-014-9119-7>
35. Bazilevs Y, Korobenko A, Deng X, Yan J (2015) Novel structural modeling and mesh moving techniques for advanced FSI simulation of wind turbines. *Int J Numer Methods Eng* 102:766–783. <https://doi.org/10.1002/nme.4738>
36. Korobenko A, Yan J, Gohari SMI, Sarkar S, Bazilevs Y (2017) FSI simulation of two back-to-back wind turbines in atmospheric boundary layer flow. *Comput Fluids* 158:167–175. <https://doi.org/10.1016/j.compfluid.2017.05.010>
37. Korobenko A, Bazilevs Y, Takizawa K, Tezduyar TE (2018) Recent advances in ALE-VMS and ST-VMS computational aerodynamic and FSI analysis of wind turbines. In: Tezduyar TE (ed) *Frontiers in computational fluid–structure interaction and flow simulation: research from lead investigators under forty—2018, Modeling and Simulation in Science, Engineering and Technology*, pp 253–336. Springer, ISBN 978-3-319-96468-3. https://doi.org/10.1007/978-3-319-96469-0_7
38. Korobenko A, Bazilevs Y, Takizawa K, Tezduyar TE (2019) Computer modeling of wind turbines: 1. ALE-VMS and ST-VMS aerodynamic and FSI analysis. *Arch Comput Methods Eng* 26:1059–1099. <https://doi.org/10.1007/s11831-018-9292-1>
39. Korobenko A, Hsu M-C, Akkerman I, Bazilevs Y (2013) Aerodynamic simulation of vertical-axis wind turbines. *J Appl Mech* 81:021011. <https://doi.org/10.1115/1.4024415>
40. Bazilevs Y, Korobenko A, Deng X, Yan J, Kinzel M, Dabiri JO (2014) FSI modeling of vertical-axis wind turbines. *J Appl Mech* 81:081006. <https://doi.org/10.1115/1.4027466>
41. Yan J, Korobenko A, Deng X, Bazilevs Y (2016) Computational free-surface fluid–structure interaction with application to floating offshore wind turbines. *Comput Fluids* 141:155–174. <https://doi.org/10.1016/j.compfluid.2016.03.008>
42. Bazilevs Y, Korobenko A, Yan J, Pal A, Gohari SMI, Sarkar S (2015) ALE-VMS formulation for stratified turbulent incompressible flows with applications. *Math Models Methods Appl Sci* 25:2349–2375. <https://doi.org/10.1142/S0218202515400114>
43. Bazilevs Y, Korobenko A, Deng X, Yan J (2016) FSI modeling for fatigue-damage prediction in full-scale wind-turbine blades. *J Appl Mech* 83(6):061010
44. Bazilevs Y, Calo VM, Zhang Y, Hughes TJR (2006) Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Comput Mech* 38:310–322
45. Bazilevs Y, Gohean JR, Hughes TJR, Moser RD, Zhang Y (2009) Patient-specific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Comput Methods Appl Mech Eng* 198:3534–3550
46. Bazilevs Y, Hsu M-C, Benson D, Sankaran S, Marsden A (2009) Computational fluid–structure interaction: methods and application to a total cavopulmonary connection. *Comput Mech* 45:77–89
47. Bazilevs Y, Hsu M-C, Zhang Y, Wang W, Liang X, Kvamsdal T, Brekken R, Isaksen J (2010) A fully-coupled fluid–structure interaction simulation of cerebral aneurysms. *Comput Mech* 46:3–16
48. Bazilevs Y, Hsu M-C, Zhang Y, Wang W, Kvamsdal T, Hentschel S, Isaksen J (2010) Computational fluid–structure interaction: methods and application to cerebral aneurysms. *Biomech Model Mechanobiol* 9:481–498
49. Hsu M-C, Bazilevs Y (2011) Blood vessel tissue prestress modeling for vascular fluid–structure interaction simulations. *Finite Elem Anal Des* 47:593–599
50. Long CC, Marsden AL, Bazilevs Y (2013) Fluid–structure interaction simulation of pulsatile ventricular assist devices. *Comput Mech* 52:971–981. <https://doi.org/10.1007/s00466-013-0858-3>
51. Long CC, Esmaily-Moghadam M, Marsden AL, Bazilevs Y (2014) Computation of residence time in the simulation of pulsatile ventricular assist devices. *Comput Mech* 54:911–919. <https://doi.org/10.1007/s00466-013-0931-y>
52. Long CC, Marsden AL, Bazilevs Y (2014) Shape optimization of pulsatile ventricular assist devices using FSI to minimize thrombotic risk. *Comput Mech* 54:921–932. <https://doi.org/10.1007/s00466-013-0967-z>
53. Hsu M-C, Kamensky D, Bazilevs Y, Sacks MS, Hughes TJR (2014) Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. *Comput Mech* 54:1055–1071. <https://doi.org/10.1007/s00466-014-1059-4>
54. Hsu M-C, Kamensky D, Xu F, Kiendl J, Wang C, Wu MCH, Mineroff J, Reali A, Bazilevs Y, Sacks MS (2015) Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models. *Comput Mech* 55:1211–1225. <https://doi.org/10.1007/s00466-015-1166-x>
55. Kamensky D, Hsu M-C, Schillinger D, Evans JA, Aggarwal A, Bazilevs Y, Sacks MS, Hughes TJR (2015) An immersogeometric variational framework for fluid–structure interaction: application to bioprosthetic heart valves. *Comput Methods Appl Mech Eng* 284:1005–1053
56. Akkerman I, Bazilevs Y, Benson DJ, Farthing MW, Kees CE (2012) Free-surface flow and fluid-object interaction modeling with emphasis on ship hydrodynamics. *J Appl Mech* 79:010905
57. Akkerman I, Dunaway J, Kvandal J, Spinks J, Bazilevs Y (2012) Toward free-surface modeling of planing vessels: simulation of the Fridsma hull using ALE-VMS. *Comput Mech* 50:719–727
58. Wang C, Wu MCH, Xu F, Hsu M-C, Bazilevs Y (2017) Modeling of a hydraulic arresting gear using fluid–structure interaction and isogeometric analysis. *Comput Fluids* 142:3–14. <https://doi.org/10.1016/j.compfluid.2015.12.004>
59. Wu MCH, Kamensky D, Wang C, Herrema AJ, Xu F, Pigazzini MS, Verma A, Marsden AL, Bazilevs Y, Hsu M-C (2017) Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: application to a hydraulic arresting gear. *Comput Methods Appl Mech Eng* 316:668–693
60. Yan J, Deng X, Korobenko A, Bazilevs Y (2017) Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines. *Comput Fluids* 158:157–166. <https://doi.org/10.1016/j.compfluid.2016.06.016>
61. Castorriani A, Corsini A, Rispoli F, Takizawa K, Tezduyar TE (2019) A stabilized ALE method for computational fluid–structure interaction analysis of passive morphing in turboma-

- chinery. *Math Models Methods Appl Sci* 29:967–994. <https://doi.org/10.1142/S0218202519410057>
62. Augier B, Yan J, Korobenko A, Czarnowski J, Ketterman G, Bazilevs Y (2015) Experimental and numerical FSI study of compliant hydrofoils. *Comput Mech* 55:1079–1090. <https://doi.org/10.1007/s00466-014-1090-5>
 63. Yan J, Augier B, Korobenko A, Czarnowski J, Ketterman G, Bazilevs Y (2016) FSI modeling of a propulsion system based on compliant hydrofoils in a tandem configuration. *Comput Fluids* 141:201–211. <https://doi.org/10.1016/j.compfluid.2015.07.013>
 64. Helgedagsrud TA, Bazilevs Y, Mathisen KM, Oiseth OA. Computational and experimental investigation of free vibration and flutter of bridge decks. *Comput Mech*. <https://doi.org/10.1007/s00466-018-1587-4>
 65. Helgedagsrud TA, Bazilevs Y, Korobenko A, Mathisen KM, Oiseth OA. Using ALE-VMS to compute aerodynamic derivatives of bridge sections. *Comput Fluids*. <https://doi.org/10.1016/j.compfluid.2018.04.037>
 66. Helgedagsrud TA, Akkerman I, Bazilevs Y, Mathisen KM, Oiseth OA (2019) Isogeometric modeling and experimental investigation of moving-domain bridge aerodynamics. *ASCE J Eng Mech* 145:04019026. [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0001601](https://doi.org/10.1061/(ASCE)EM.1943-7889.0001601)
 67. Kamensky D, Evans JA, Hsu M-C, Bazilevs Y (2017) Projection-based stabilization of interface Lagrange multipliers in immersed-geometric fluid–thin structure interaction analysis, with application to heart valve modeling. *Comput Math Appl* 74:2068–2088. <https://doi.org/10.1016/j.camwa.2017.07.006>
 68. Yu Y, Kamensky D, Hsu M-C, Lu XY, Bazilevs Y, Hughes TJR (2018) Error estimates for projection-based dynamic augmented Lagrangian boundary condition enforcement, with application to fluid–structure interaction. *Math Models Methods Appl Sci* 28:2457–2509. <https://doi.org/10.1142/S0218202518500537>
 69. Yan J, Korobenko A, Tejada-Martinez AE, Golshan R, Bazilevs Y (2017) A new variational multiscale formulation for stratified incompressible turbulent flows. *Comput Fluids* 158:150–156. <https://doi.org/10.1016/j.compfluid.2016.12.004>
 70. van Opstal TM, Yan J, Coley C, Evans JA, Kvamsdal T, Bazilevs Y (2017) Isogeometric divergence-conforming variational multiscale formulation of incompressible turbulent flows. *Comput Methods Appl Mech Eng* 316:859–879. <https://doi.org/10.1016/j.cma.2016.10.015>
 71. Xu F, Moutsanidis G, Kamensky D, Hsu M-C, Murugan M, Ghoshal A, Bazilevs Y (2017) Compressible flows on moving domains: stabilized methods, weakly enforced essential boundary conditions, sliding interfaces, and application to gas-turbine modeling. *Comput Fluids* 158:201–220. <https://doi.org/10.1016/j.compfluid.2017.02.006>
 72. Tezduyar TE, Takizawa K (2019) Space-time computations in practical engineering applications: a summary of the 25-year history. *Comput Mech* 63:747–753. <https://doi.org/10.1007/s00466-018-1620-7>
 73. Takizawa K, Tezduyar TE (2012) Computational methods for parachute fluid–structure interactions. *Arch Comput Methods Eng* 19:125–169. <https://doi.org/10.1007/s11831-012-9070-4>
 74. Takizawa K, Fritze M, Montes D, Spielman T, Tezduyar TE (2012) Fluid–structure interaction modeling of ringsail parachutes with disreefing and modified geometric porosity. *Comput Mech* 50:835–854. <https://doi.org/10.1007/s00466-012-0761-3>
 75. Takizawa K, Tezduyar TE, Boben J, Kostov N, Boswell C, Buscher A (2013) Fluid–structure interaction modeling of clusters of spacecraft parachutes with modified geometric porosity. *Comput Mech* 52:1351–1364. <https://doi.org/10.1007/s00466-013-0880-5>
 76. Takizawa K, Tezduyar TE, Boswell C, Tsutsui Y, Montel K (2015) Special methods for aerodynamic-moment calculations from parachute FSI modeling. *Comput Mech* 55:1059–1069. <https://doi.org/10.1007/s00466-014-1074-5>
 77. Takizawa K, Montes D, Fritze M, McIntyre S, Boben J, Tezduyar TE (2013) Methods for FSI modeling of spacecraft parachute dynamics and cover separation. *Math Models Methods Appl Sci* 23:307–338. <https://doi.org/10.1142/S0218202513400058>
 78. Takizawa K, Tezduyar TE, Boswell C, Kolesar R, Montel K (2014) FSI modeling of the reefed stages and disreefing of the Orion spacecraft parachutes. *Comput Mech* 54:1203–1220. <https://doi.org/10.1007/s00466-014-1052-y>
 79. Takizawa K, Tezduyar TE, Kolesar R, Boswell C, Kanai T, Montel K (2014) Multiscale methods for gore curvature calculations from FSI modeling of spacecraft parachutes. *Comput Mech* 54:1461–1476. <https://doi.org/10.1007/s00466-014-1069-2>
 80. Takizawa K, Tezduyar TE, Kolesar R (2015) FSI modeling of the Orion spacecraft drogue parachutes. *Comput Mech* 55:1167–1179. <https://doi.org/10.1007/s00466-014-1108-z>
 81. Takizawa K, Henicke B, Tezduyar TE, Hsu M-C, Bazilevs Y (2011) Stabilized space-time computation of wind-turbine rotor aerodynamics. *Comput Mech* 48:333–344. <https://doi.org/10.1007/s00466-011-0589-2>
 82. Takizawa K, Henicke B, Montes D, Tezduyar TE, Hsu M-C, Bazilevs Y (2011) Numerical-performance studies for the stabilized space-time computation of wind-turbine rotor aerodynamics. *Comput Mech* 48:647–657. <https://doi.org/10.1007/s00466-011-0614-5>
 83. Takizawa K, Tezduyar TE, McIntyre S, Kostov N, Kolesar R, Habluetzel C (2014) Space-time VMS computation of wind-turbine rotor and tower aerodynamics. *Comput Mech* 53:1–15. <https://doi.org/10.1007/s00466-013-0888-x>
 84. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C, Oiseth O, Mathisen KM, Kostov N, McIntyre S (2014) Engineering analysis and design with ALE-VMS and space-time methods. *Arch Comput Methods Eng* 21:481–508. <https://doi.org/10.1007/s11831-014-9113-0>
 85. Takizawa K (2014) Computational engineering analysis with the new-generation space-time methods. *Comput Mech* 54:193–211. <https://doi.org/10.1007/s00466-014-0999-z>
 86. Takizawa K, Tezduyar TE, Mochizuki H, Hattori H, Mei S, Pan L, Montel K (2015) Space-time VMS method for flow computations with slip interfaces (ST-SI). *Math Models Methods Appl Sci* 25:2377–2406. <https://doi.org/10.1142/S0218202515400126>
 87. Takizawa K, Henicke B, Puntel A, Spielman T, Tezduyar TE (2012) Space-time computational techniques for the aerodynamics of flapping wings. *J Appl Mech* 79:010903. <https://doi.org/10.1115/1.4005073>
 88. Takizawa K, Henicke B, Puntel A, Kostov N, Tezduyar TE (2012) Space-time techniques for computational aerodynamics modeling of flapping wings of an actual locust. *Comput Mech* 50:743–760. <https://doi.org/10.1007/s00466-012-0759-x>
 89. Takizawa K, Henicke B, Puntel A, Kostov N, Tezduyar TE (2013) Computer modeling techniques for flapping-wing aerodynamics of a locust. *Comput Fluids* 85:125–134. <https://doi.org/10.1016/j.compfluid.2012.11.008>
 90. Takizawa K, Kostov N, Puntel A, Henicke B, Tezduyar TE (2012) Space-time computational analysis of bio-inspired flapping-wing aerodynamics of a micro aerial vehicle. *Comput Mech* 50:761–778. <https://doi.org/10.1007/s00466-012-0758-y>
 91. Takizawa K, Tezduyar TE, Kostov N (2014) Sequentially-coupled space-time FSI analysis of bio-inspired flapping-wing aerodynamics of an MAV. *Comput Mech* 54:213–233. <https://doi.org/10.1007/s00466-014-0980-x>
 92. Takizawa K, Tezduyar TE, Buscher A, Asada S (2014) Space-time interface-tracking with topology change (ST-TC). *Comput Mech* 54:955–971. <https://doi.org/10.1007/s00466-013-0935-7>

93. Takizawa K, Tezduyar TE, Buscher A (2015) Space-time computational analysis of MAV flapping-wing aerodynamics with wing clapping. *Comput Mech* 55:1131–1141. <https://doi.org/10.1007/s00466-014-1095-0>
94. Takizawa K, Bazilevs Y, Tezduyar TE, Long CC, Marsden AL, Schjodt K (2014) ST and ALE-VMS methods for patient-specific cardiovascular fluid mechanics modeling. *Math Models Methods Appl Sci* 24:2437–2486. <https://doi.org/10.1142/S0218202514500250>
95. Takizawa K, Schjodt K, Puntel A, Kostov N, Tezduyar TE (2012) Patient-specific computer modeling of blood flow in cerebral arteries with aneurysm and stent. *Comput Mech* 50:675–686. <https://doi.org/10.1007/s00466-012-0760-4>
96. Takizawa K, Schjodt K, Puntel A, Kostov N, Tezduyar TE (2013) Patient-specific computational analysis of the influence of a stent on the unsteady flow in cerebral aneurysms. *Comput Mech* 51:1061–1073. <https://doi.org/10.1007/s00466-012-0790-y>
97. Suito H, Takizawa K, Huynh VQH, Sze D, Ueda T (2014) FSI analysis of the blood flow and geometrical characteristics in the thoracic aorta. *Comput Mech* 54:1035–1045. <https://doi.org/10.1007/s00466-014-1017-1>
98. Suito H, Takizawa K, Huynh VQH, Sze D, Ueda T, Tezduyar TE (2016) A geometrical-characteristics study in patient-specific FSI analysis of blood flow in the thoracic aorta. In: Bazilevs Y, Takizawa K (eds) *Advances in computational fluid–structure interaction and flow simulation: new methods and challenging computations, modeling and simulation in science, engineering and technology*. Springer, Berlin, pp 379–386. https://doi.org/10.1007/978-3-319-40827-9_29 ISBN 978-3-319-40825-5
99. Takizawa K, Tezduyar TE, Uchikawa H, Terahara T, Sasaki T, Shiozaki K, Yoshida A, Komiya K, Inoue G (2018) Aorta flow analysis and heart valve flow and structure analysis. In: Tezduyar TE (ed) *Frontiers in computational fluid–structure interaction and flow simulation: research from lead investigators under forty—2018, Modeling and simulation in science, engineering and technology*. Springer, Berlin, pp 29–89. https://doi.org/10.1007/978-3-319-96469-0_2 ISBN 978-3-319-96468-3
100. Takizawa K, Tezduyar TE, Uchikawa H, Terahara T, Sasaki T, Yoshida A (2019) Mesh refinement influence and cardiac-cycle flow periodicity in aorta flow analysis with isogeometric discretization. *Comput Fluids* 179:790–798. <https://doi.org/10.1016/j.compfluid.2018.05.025>
101. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C (2019) Computational cardiovascular flow analysis with the variational multiscale methods. *J Adv Eng Comput* 3:366–405. <https://doi.org/10.25073/jaec.201932.245>
102. Takizawa K, Tezduyar TE, Buscher A, Asada S (2014) Space-time fluid mechanics computation of heart valve models. *Comput Mech* 54:973–986. <https://doi.org/10.1007/s00466-014-1046-9>
103. Takizawa K, Tezduyar TE (2016) New directions in space-time computational methods. In: Bazilevs Y, Takizawa K (eds) *Advances in computational fluid–structure interaction and flow simulation: new methods and challenging computations, modeling and simulation in science, engineering and technology*. Springer, Berlin, pp 159–178. https://doi.org/10.1007/978-3-319-40827-9_13 ISBN 978-3-319-40825-5
104. Takizawa K, Tezduyar TE, Terahara T, Sasaki T (2018) Heart valve flow computation with the space-time slip interface topology change (ST-SI-TC) method and isogeometric analysis (IGA). In: Wriggers P, Lenarz T (eds) *Biomedical technology: modeling, experiments and simulation, lecture notes in applied and computational mechanics*. Springer, Berlin, pp 77–99. https://doi.org/10.1007/978-3-319-59548-1_6 ISBN 978-3-319-59547-4
105. Takizawa K, Tezduyar TE, Terahara T, Sasaki T (2017) Heart valve flow computation with the integrated space-time VMS, slip interface, topology change and isogeometric discretization methods. *Comput Fluids* 158:176–188. <https://doi.org/10.1016/j.compfluid.2016.11.012>
106. Terahara T, Takizawa K, Tezduyar TE, Bazilevs Y, Hsu M-C (2020) Heart valve isogeometric sequentially-coupled FSI analysis with the space–time topology change method. *Comput Mech*. <https://doi.org/10.1007/s00466-019-01813-0>
107. Terahara T, Takizawa K, Tezduyar TE, Tsushima A, Shiozaki K (2020) Ventricle–valve–aorta flow analysis with the Space–Time isogeometric discretization and topology change. *Comput Mech*. <https://doi.org/10.1007/s00466-020-01822-4>
108. Yu Y, Zhang YJ, Takizawa K, Tezduyar TE, Sasaki T (2020) Anatomically realistic lumen motion representation in patient-specific space-time isogeometric flow analysis of coronary arteries with time-dependent medical-image data. *Comput Mech* 65:395–404. <https://doi.org/10.1007/s00466-019-01774-4>
109. Takizawa K, Montes D, McIntyre S, Tezduyar TE (2013) Space-time VMS methods for modeling of incompressible flows at high Reynolds numbers. *Math Models Methods Appl Sci* 23:223–248. <https://doi.org/10.1142/s0218202513400022>
110. Takizawa K, Tezduyar TE, Kuraishi T, Tabata S, Takagi H (2016) Computational thermo-fluid analysis of a disk brake. *Comput Mech* 57:965–977. <https://doi.org/10.1007/s00466-016-1272-4>
111. Takizawa K, Tezduyar TE, Hattori H (2017) Computational analysis of flow-driven string dynamics in turbomachinery. *Comput Fluids* 142:109–117. <https://doi.org/10.1016/j.compfluid.2016.02.019>
112. Komiya K, Kanai T, Otoguro Y, Kaneko M, Hirota K, Zhang Y, Takizawa K, Tezduyar TE, Nohmi M, Tsuneda T, Kawai M, Isono M (2019) Computational analysis of flow-driven string dynamics in a pump and residence time calculation. *IOP Conf Ser Earth Environ Sci* 240:062014. <https://doi.org/10.1088/1755-1315/240/6/062014>
113. Kanai T, Takizawa K, Tezduyar TE, Komiya K, Kaneko M, Hirota K, Nohmi M, Tsuneda T, Kawai M, Isono M (2019) Methods for computation of flow-driven string dynamics in a pump and residence time. *Math Models Methods Appl Sci* 29:839–870. <https://doi.org/10.1142/S021820251941001X>
114. Takizawa K, Tezduyar TE, Otoguro Y, Terahara T, Kuraishi T, Hattori H (2017) Turbocharger flow computations with the space-time isogeometric analysis (ST-IGA). *Comput Fluids* 142:15–20. <https://doi.org/10.1016/j.compfluid.2016.02.021>
115. Otoguro Y, Takizawa K, Tezduyar TE (2017) Space-time VMS computational flow analysis with isogeometric discretization and a general-purpose NURBS mesh generation method. *Comput Fluids* 158:189–200. <https://doi.org/10.1016/j.compfluid.2017.04.017>
116. Otoguro Y, Takizawa K, Tezduyar TE (2018) A general-purpose NURBS mesh generation method for complex geometries. In: Tezduyar TE (ed) *Frontiers in computational fluid–structure interaction and flow simulation: research from lead investigators under forty - 2018, Modeling and simulation in science, engineering and technology*. Springer, Berlin, pp 399–434. https://doi.org/10.1007/978-3-319-96469-0_10 ISBN 978-3-319-96468-3
117. Otoguro Y, Takizawa K, Tezduyar TE, Nagaoka K, Mei S (2019) Turbocharger turbine and exhaust manifold flow computation with the space-time variational multiscale method and isogeometric analysis. *Comput Fluids* 179:764–776. <https://doi.org/10.1016/j.compfluid.2018.05.019>
118. Otoguro Y, Takizawa K, Tezduyar TE, Nagaoka K, Avsar R, Zhang Y (2019) Space-time VMS flow analysis of a turbocharger turbine with isogeometric discretization: computations with time-dependent and steady-inflow representations of the intake/exhaust cycle. *Comput Mech* 64:1403–1419. <https://doi.org/10.1007/s00466-019-01722-2>

119. Takizawa K, Tezduyar TE, Asada S, Kuraishi T (2016) Space-time method for flow computations with slip interfaces and topology changes (ST-SI-TC). *Comput Fluids* 141:124–134. <https://doi.org/10.1016/j.compfluid.2016.05.006>
120. Kuraishi T, Takizawa K, Tezduyar TE (2018) Space-time computational analysis of tire aerodynamics with actual geometry, road contact and tire deformation. In: Tezduyar TE (ed) *Frontiers in computational fluid-structure interaction and flow simulation: research from lead investigators under forty - 2018*, Modeling and simulation in science, engineering and technology. Springer, Berlin, pp 337–376. https://doi.org/10.1007/978-3-319-96469-0_8 ISBN 978-3-319-96468-3
121. Kuraishi T, Takizawa K, Tezduyar TE (2019) Tire aerodynamics with actual tire geometry, road contact and tire deformation. *Comput Mech* 63:1165–1185. <https://doi.org/10.1007/s00466-018-1642-1>
122. Kuraishi T, Takizawa K, Tezduyar TE (2019) Space-time computational analysis of tire aerodynamics with actual geometry, road contact, tire deformation, road roughness and fluid film. *Comput Mech* 64:1699–1718. <https://doi.org/10.1007/s00466-019-01746-8>
123. Kuraishi T, Takizawa K, Tezduyar TE (2019) Space-time isogeometric flow analysis with built-in Reynolds-equation limit. *Math Models Methods Appl Sci* 29:871–904. <https://doi.org/10.1142/S0218202519410021>
124. Takizawa K, Tezduyar TE, Terahara T (2016) Ram-air parachute structural and fluid mechanics computations with the space-time isogeometric analysis (ST-IGA). *Comput Fluids* 141:191–200. <https://doi.org/10.1016/j.compfluid.2016.05.027>
125. Takizawa K, Tezduyar TE, Kanai T (2017) Porosity models and computational methods for compressible-flow aerodynamics of parachutes with geometric porosity. *Math Models Methods Appl Sci* 27:771–806. <https://doi.org/10.1142/S0218202517500166>
126. Kanai T, Takizawa K, Tezduyar TE, Tanaka T, Hartmann A (2019) Compressible-flow geometric-porosity modeling and spacecraft parachute computation with isogeometric discretization. *Comput Mech* 63:301–321. <https://doi.org/10.1007/s00466-018-1595-4>
127. Tezduyar TE, Aliabadi SK, Behr M, Mittal S (1994) Massively parallel finite element simulation of compressible and incompressible flows. *Comput Methods Appl Mech Eng* 119:157–177. [https://doi.org/10.1016/0045-7825\(94\)00082-4](https://doi.org/10.1016/0045-7825(94)00082-4)
128. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Comput Methods Appl Mech Eng* 194:4135–4195
129. Takizawa K, Tezduyar TE (2014) Space-time computation techniques with continuous representation in time (ST-C). *Comput Mech* 53:91–99. <https://doi.org/10.1007/s00466-013-0895-y>
130. Takizawa K, Takagi H, Tezduyar TE, Torii R (2014) Estimation of element-based zero-stress state for arterial FSI computations. *Comput Mech* 54:895–910. <https://doi.org/10.1007/s00466-013-0919-7>
131. Takizawa K, Torii R, Takagi H, Tezduyar TE, Xu XY (2014) Coronary arterial dynamics computation with medical-image-based time-dependent anatomical models and element-based zero-stress state estimates. *Comput Mech* 54:1047–1053. <https://doi.org/10.1007/s00466-014-1049-6>
132. Takizawa K, Tezduyar TE, Sasaki T (2018) Estimation of element-based zero-stress state in arterial FSI computations with isogeometric wall discretization. In: Wriggers P, Lenarz T (eds) *Biomedical technology: modeling, experiments and simulation*, lecture notes in applied and computational mechanics. Springer, Berlin, pp 101–122. https://doi.org/10.1007/978-3-319-59548-1_7 ISBN 978-3-319-59547-4
133. Takizawa K, Tezduyar TE, Sasaki T (2017) Aorta modeling with the element-based zero-stress state and isogeometric discretization. *Comput Mech* 59:265–280. <https://doi.org/10.1007/s00466-016-1344-5>
134. Sasaki T, Takizawa K, Tezduyar TE (2019) Aorta zero-stress state modeling with T-spline discretization. *Comput Mech* 63:1315–1331. <https://doi.org/10.1007/s00466-018-1651-0>
135. Sasaki T, Takizawa K, Tezduyar TE (2019) Medical-image-based aorta modeling with zero-stress-state estimation. *Comput Mech* 64:249–271. <https://doi.org/10.1007/s00466-019-01669-4>
136. Takizawa K, Tezduyar TE, Sasaki T (2019) Isogeometric hyperelastic shell analysis with out-of-plane deformation mapping. *Comput Mech* 63:681–700. <https://doi.org/10.1007/s00466-018-1616-3>
137. Tezduyar TE, Behr M, Mittal S, Johnson AA (1992) Computation of unsteady incompressible flows with the finite element methods: space-time formulations, iterative strategies and massively parallel implementations. *New Methods in Transient Analysis PVP-Vol.246/AMD-Vol.143*, ASME, New York, 7–24
138. Tezduyar T, Aliabadi S, Behr M, Johnson A, Mittal S (1993) Parallel finite-element computation of 3D flows. *Computer* 26(10):27–36. <https://doi.org/10.1109/2.237441>
139. Johnson AA, Tezduyar TE (1994) Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Comput Methods Appl Mech Eng* 119:73–94. [https://doi.org/10.1016/0045-7825\(94\)00077-8](https://doi.org/10.1016/0045-7825(94)00077-8)
140. Stein K, Tezduyar T, Benney R (2003) Mesh moving techniques for fluid-structure interactions with large displacements. *J Appl Mech* 70:58–63. <https://doi.org/10.1115/1.1530635>
141. Tezduyar T (2001) Finite element interface-tracking and interface-capturing techniques for flows with moving boundaries and interfaces. In: *Proceedings of the ASME symposium on fluid-physics and heat transfer for macro- and micro-scale gas-liquid and phase-change flows (CD-ROM)*, ASME paper IMECE2001/HTD-24206, ASME, New York, New York
142. Tezduyar TE (2003) Stabilized finite element formulations and interface-tracking and interface-capturing techniques for incompressible flows. In: Hafez MM (ed) *Numerical simulations of incompressible flows*. World Scientific, New Jersey, pp 221–239. https://doi.org/10.1142/9789812796837_0013
143. Stein K, Tezduyar TE, Benney R (2004) Automatic mesh update with the solid-extension mesh moving technique. *Comput Methods Appl Mech Eng* 193:2019–2032. <https://doi.org/10.1016/j.cma.2003.12.046>
144. Tezduyar TE, Sathe S, Keedy R, Stein K (2006) Space-time finite element techniques for computation of fluid-structure interactions. *Comput Methods Appl Mech Eng* 195:2002–2027. <https://doi.org/10.1016/j.cma.2004.09.014>
145. Stein K, Tezduyar T (2002) Advanced mesh update techniques for problems involving large displacements. In: *Proceedings of the fifth world congress on computational mechanics*, on-line publication: paper-ID: 81489. <http://www.researchgate.net/publication/303737884/>, Vienna, Austria
146. Hsu M-C, Bazilevs Y, Calo VM, Tezduyar TE, Hughes TJR (2010) Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Comput Methods Appl Mech Eng* 199:828–840. <https://doi.org/10.1016/j.cma.2009.06.019>
147. Takizawa K, Tezduyar TE, Otoguro Y (2018) Stabilization and discontinuity-capturing parameters for space-time flow computations with finite element and isogeometric discretizations. *Comput Mech* 62:1169–1186. <https://doi.org/10.1007/s00466-018-1557-x>
148. Takizawa K, Ueda Y, Tezduyar TE (2019) A node-numbering-invariant directional length scale for simplex elements. *Math Models Methods Appl Sci* 29:2719–2753. <https://doi.org/10.1142/S0218202519500581>

149. Otoguro Y, Takizawa K, Tezduyar TE (2020) Element length calculation in B-spline meshes for complex geometries. *Comput Mech*. <https://doi.org/10.1007/s00466-019-01809-w>
150. Corsini A, Menichini C, Rispoli F, Santoriello A, Tezduyar TE (2009) A multiscale finite element formulation with discontinuity capturing for turbulence models with dominant reactionlike terms. *J Appl Mech* 76:021211. <https://doi.org/10.1115/1.3062967>
151. Rispoli F, Saavedra R, Menichini F, Tezduyar TE (2009) Computation of inviscid supersonic flows around cylinders and spheres with the V-SGS stabilization and $YZ\beta$ shock-capturing. *J Appl Mech* 76:021209. <https://doi.org/10.1115/1.3057496>
152. Corsini A, Iossa C, Rispoli F, Tezduyar TE (2010) A DRD finite element formulation for computing turbulent reacting flows in gas turbine combustors. *Comput Mech* 46:159–167. <https://doi.org/10.1007/s00466-009-0441-0>
153. Corsini A, Rispoli F, Tezduyar TE (2011) Stabilized finite element computation of NOx emission in aero-engine combustors. *Int J Numer Methods Fluids* 65:254–270. <https://doi.org/10.1002/fld.2451>
154. Corsini A, Rispoli F, Tezduyar TE (2012) Computer modeling of wave-energy air turbines with the SUPG/PSPG formulation and discontinuity-capturing technique. *J Appl Mech* 79:010910. <https://doi.org/10.1115/1.4005060>
155. Corsini A, Rispoli F, Sheard AG, Tezduyar TE (2012) Computational analysis of noise reduction devices in axial fans with stabilized finite element formulations. *Comput Mech* 50:695–705. <https://doi.org/10.1007/s00466-012-0789-4>
156. Kler PA, Dalcin LD, Paz RR, Tezduyar TE (2013) SUPG and discontinuity-capturing methods for coupled fluid mechanics and electrochemical transport problems. *Comput Mech* 51:171–185. <https://doi.org/10.1007/s00466-012-0712-z>
157. Corsini A, Rispoli F, Sheard AG, Takizawa K, Tezduyar TE, Venturini P (2014) A variational multiscale method for particle-cloud tracking in turbomachinery flows. *Comput Mech* 54:1191–1202. <https://doi.org/10.1007/s00466-014-1050-0>
158. Rispoli F, Delibra G, Venturini P, Corsini A, Saavedra R, Tezduyar TE (2015) Particle tracking and particle-shock interaction in compressible-flow computations with the V-SGS stabilization and $YZ\beta$ shock-capturing. *Comput Mech* 55:1201–1209. <https://doi.org/10.1007/s00466-015-1160-3>
159. Cardillo L, Corsini A, Delibra G, Rispoli F, Tezduyar TE (2016) Flow analysis of a wave-energy air turbine with the SUPG/PSPG stabilization and discontinuity-capturing directional dissipation. *Comput Fluids* 141:184–190. <https://doi.org/10.1016/j.compfluid.2016.07.011>
160. Castorrini A, Corsini A, Rispoli F, Venturini P, Takizawa K, Tezduyar TE (2016) Computational analysis of wind-turbine blade rain erosion. *Comput Fluids* 141:175–183. <https://doi.org/10.1016/j.compfluid.2016.08.013>
161. Castorrini A, Corsini A, Rispoli F, Venturini P, Takizawa K, Tezduyar TE (2019) Computational analysis of performance deterioration of a wind turbine blade strip subjected to environmental erosion. *Comput Mech* 64:1133–1153. <https://doi.org/10.1007/s00466-019-01697-0>
162. Ogden RW, Rodney H (1972) Large deformation isotropic elasticity: on the correlation of theory and experiment for incompressible rubberlike solids. *Proc R Soc Lond A* 328:567–583
163. Nolan D, Gower A, Destrade M, Ogden R, McGarry J (2014) A robust anisotropic hyperelastic formulation for the modelling of soft tissue. *J Mech Behav Biomed Mater* 39:48–60. <https://doi.org/10.1016/j.jmbbm.2014.06.016>
164. Holzapfel G, Gasser T, Ogden R (2000) A new constitutive framework for arterial wall mechanics and a comparative study of material models. *J Elast* 61(1–3):1–48, cited By 1794, <https://doi.org/10.1023/A:1010835316564>
165. Gasser TC, Ogden RW, Holzapfel GA (2005) Hyperelastic modelling of arterial layers with distributed collagen fibre orientations. *J R Soc Interface* 3(6):15–35. <https://doi.org/10.1098/rsif.2005.0073>
166. Masud A, Hughes TJR (1997) A space-time Galerkin/least-squares finite element formulation of the Navier–Stokes equations for moving domain problems. *Comput Methods Appl Mech Eng* 146:91–126

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.