A linear-elasticity-based mesh moving method with no cycle-to-cycle accumulated distortion

Patrícia Tonon¹ · Rodolfo André Kuche Sanches¹ · Kenji Takizawa² · Tayfun E. Tezduyar^{3,4}

Received: 9 September 2020 / Accepted: 26 October 2020 / Published online: 6 January 2021
© The Author(s) 2021

Abstract

Good mesh moving methods are always part of what makes moving-mesh methods good in computation of flow problems with moving boundaries and interfaces, including fluid–structure interaction. Moving-mesh methods, such as the space–time (ST) and arbitrary Lagrangian–Eulerian (ALE) methods, enable mesh-resolution control near solid surfaces and thus high-resolution representation of the boundary layers. Mesh moving based on linear elasticity and mesh-Jacobian-based stiffening (MJBS) has been in use with the ST and ALE methods since 1992. In the MJBS, the objective is to stiffen the smaller elements, which are typically placed near solid surfaces, more than the larger ones, and this is accomplished by altering the way we account for the Jacobian of the transformation from the element domain to the physical domain. In computing the mesh motion between time levels t_n and t_{n+1} with the linear-elasticity equations, the most common option is to compute the displacement from the configuration at t_n . While this option works well for most problems, because the method is path-dependent, it involves cycle-to-cycle accumulated mesh distortion. The back-cycle-based mesh moving (BCBMM) method, introduced recently with two versions, can remedy that. In the BCBMM, there is no cycle-to-cycle accumulated distortion. In this article, for the first time, we present mesh moving test computations with the BCBMM. We also introduce a version we call “half-cycle-based mesh moving” (HCBMM) method, and that is for computations where the boundary or interface motion in the second half of the cycle consists of just reversing the steps in the first half and we want the mesh to behave the same way. We present detailed 2D and 3D test computations with finite element meshes, using as the test case the mesh motion associated with wing pitching. The computations show that all versions of the BCBMM perform well, with no cycle-to-cycle accumulated distortion, and with the HCBMM, as the wing in the second half of the cycle just reverses its motion steps in the first half, the mesh behaves the same way.

Keywords Mesh moving method · Linear-elasticity equations · Mesh-Jacobian-based stiffening · Cycle-to-cycle accumulated distortion · Back-cycle-based mesh moving · Half-cycle-based mesh moving

1 Introduction

Mesh moving based on linear elasticity and mesh-Jacobian-based stiffening (MJBS) [1–3] has been in use with moving-

mesh methods since 1992 in computation of flow problems with moving boundaries and interfaces (MBIs), including fluid–structure interaction (FSI). In the MJBS, the objective is to stiffen the smaller elements, which are typically placed near solid surfaces, more than the larger ones, and this is accomplished by altering the way we account for the Jacobian of the transformation from the element domain

✉ Kenji Takizawa
Kenji.Takizawa@tafsm.org

Patrícia Tonon
patriciatonon@usp.br

Rodolfo André Kuche Sanches
rodolfo.sanches@usp.br

Tayfun E. Tezduyar
tezduyar@tafsm.org

¹ School of Engineering of São Carlos, Structural Engineering Department, University of São Paulo, Av. Trabalhador São-carlense, 400, São Carlos, São Paulo, Brazil

² Department of Modern Mechanical Engineering, Waseda University, 3-4-1 Ookubo, Shinjuku-ku, Tokyo 169-8555, Japan

³ Mechanical Engineering, Rice University – MS 321, 6100 Main Street, Houston, TX 77005, USA

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

to the physical domain. In computing the mesh motion between time levels t_n and t_{n+1} with the linear-elasticity equations, the most common option is to compute the displacement from the configuration at t_n . While this option works well for most problems, because the method is path-dependent, it involves cycle-to-cycle accumulated mesh distortion. Moving the mesh based on large-deformation mechanics equations, instead of the linear-elasticity equations, would be one good way of addressing the issue. This mesh moving approach has also been in use in FSI and MBI computations, such as those reported in [4–8]. In that direction, a low-distortion mesh moving method based on fiber-reinforced hyperelasticity and optimized zero-stress state (ZSS) was introduced very recently in [9]. As an alternative direction and also very recently, the back-cycle-based mesh moving (BCBMM) method was introduced in [10] as a linear-elasticity-based mesh moving method with no cycle-to-cycle accumulated distortion. In this article, for the first time, we present mesh moving test computations with the BCBMM. We also introduce a version we call “half-cycle-based mesh moving” (HCBMM) method, and that is for computations where the boundary or interface motion in the second half of the cycle consists of just reversing the steps in the first half and we want the mesh to behave the same way. To show how the BCBMM and HCBMM perform, we present detailed 2D and 3D test computations with finite element meshes, using as the test case the mesh motion associated with wing pitching.

The BCBMM and the method introduced in [9] offer alternative solutions to the same issue but have very similar overall objectives. Consequently, the context, history and directly related methods described in the introduction of [9] are also applicable here. Therefore, in much of the rest of this introduction, we include, for completeness, subsections from the introduction of [9].

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 represented, 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 [11], 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) [12] was introduced 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 [13] 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 [14–16], 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) [17] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [14] stabilizations, and therefore it is now called “ST-SUPS.” The ST Variational Multiscale (ST-VMS) method [8, 18, 19], 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 [20–23]. 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 [18, 19]), 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, [24]) and more commonly used. The ALE-VMS method [11,25–30] is the VMS version of the ALE. It succeeded the ST-SUPS and ALE-SUPS [31] 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 [32–34] and “sliding interfaces” [35,36]. 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 [31], wind turbine aerodynamics and FSI [37–49], more specifically, vertical-axis wind turbines (VAWTs) [46,47,50,51], floating wind turbines [52], wind turbines in atmospheric boundary layers [45–47,53–55], and fatigue damage in wind turbine blades [56], patient-specific cardiovascular fluid mechanics and FSI [25,57–62], biomedical-device FSI [63–70], ship hydrodynamics with free-surface flow and fluid–object interaction [71,72], hydrodynamics and FSI of a hydraulic arresting gear [73,74], hydrodynamics of tidal-stream turbines with free-surface flow [75], passive-morphing FSI in turbomachinery [76], bioinspired FSI for marine propulsion [77,78], bridge aerodynamics and fluid–object interaction [79–81], and mixed ALE-VMS/Immersogeometric computations [66–68,82–85] in the framework of the FSITICT, including heart valve computations. Recent advances in stabilized and multiscale methods may be found for stratified incompressible flows in [86], for divergence-conforming discretizations of incompressible flows in [87], and for compressible flows with emphasis on gas-turbine modeling in [88,89].

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 [90] for a comprehensive summary of the work prior to July 2018). The classes of problems include spacecraft parachute analysis for the landing-stage parachutes [4,11,91–93], cover-separation parachutes [94] and the drogue parachutes [95–97], wind turbine aerodynamics for horizontal-axis wind turbine (HAWT) rotors [11,37,98,99], full HAWTs [6,43,100,101] and VAWTs [46–49,102,103], flapping-wing aerodynamics for an actual locust [11,104–106], bioinspired MAVs [6,101,107,108] and wing-clapping [5,109], blood flow analysis of cerebral aneurysms [101,110], stent-blocked aneurysms [110–112], aortas [7,10,69,70,113–115], heart valves [5,6,10,69,70,114,116–120] and coronary arteries in

motion [121], spacecraft aerodynamics [94,122], thermo-fluid analysis of ground vehicles and their tires [8,54,55,117], thermo-fluid analysis of disk brakes [123], flow-driven string dynamics in turbomachinery [48,49,124–126], flow analysis of turbocharger turbines [127–131], flow around tires with road contact and deformation [117,132–135], fluid films [135,136], ram-air parachutes [54,55,137], and compressible-flow spacecraft parachute aerodynamics [138,139].

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 [102,123], ST Topology Change (ST-TC) method [5,116], ST Isogeometric Analysis (ST-IGA) [18,104,127], and their integration as the ST-SI-TC [132], ST-SI-IGA [127], and ST-SI-TC-IGA [118,119].

1.2.1 ST-SI

The ST-SI was introduced in [102], 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 [35,36]. 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 [102]. 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 [123] 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 VAWTs [46–49,102,103], thermo-fluid analysis of disk brakes [123], flow-driven string dynamics in turbomachinery [48,49,124–126], flow analysis of turbocharger turbines [127–131], flow around tires with road contact and deformation [117,132–135], fluid films [135,136], aerodynamic analysis of ram-air parachutes [54,55,137], and flow analysis of heart valves [69,70,114,118–120] and ventricle-valve-aorta sequences [10].

In the ST-SI version presented in [102] 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 [54,55,137]. The compressible-flow ST-SI methods were introduced in [138], 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 [138]. These, together with the compressible-flow ST SUPG method [140], 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 [138,139].

1.2.2 ST-TC

The ST-TC [5,116] 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 [5]. 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 [5,6,69,70,114,116–120], ventricle-valve-aorta flows [10], wing clapping [5,109], flow around a rotating tire with road contact and prescribed deformation [117,132–135], and fluid films [135,136].

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 [69,70,114,118–120], ventricle-valve-aorta flows [10], tire aerodynamics with road contact and deformation [117,132–135], and fluid films [135,136].

1.2.4 ST-IGA

The success with IGA basis functions in space [25,35,57,141] motivated the integration of the ST methods with isogeometric discretization, which we broadly call “ST-IGA.” The ST-IGA was introduced in [18]. Computations with the ST-VMS and ST-IGA were first reported in [18] 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 [18] 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 [18,19] and demonstrated in [104,105,107]. 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) [104,105,107], with the name coined in [100]. The STNMUM has a wide scope that includes spinning solid surfaces. With the spinning motion represented by quadratic NURBS in time, and with sufficient number of temporal patches for a full rotation, the circular paths are represented exactly. A “secondary mapping” [11,18,19,104] 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 [8,117,123–126,142]. 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 [11,104–106], bioinspired MAVs [6,101,107,108] and wing-clapping [5,109], separation aerodynamics of spacecraft [94], aerodynamics of horizontal-axis [6,43,100,101] and vertical-axis [46–49,102,103] wind turbines, thermo-fluid analysis of ground vehicles and their tires [8,54,117], thermo-fluid analysis of disk brakes [123], flow-driven string dynamics in turbomachinery [48,49,124–126], flow analysis of turbocharger turbines [127–131], and flow analysis of coronary arteries in motion [121].

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 [127–131], flow-driven string dynamics in turbomachinery [48,49,125,126], ram-air parachutes [54,55,137], spacecraft parachutes [139], aortas [69,70,114,115], heart valves [69,70,114,118–120], ventricle-valve-aorta sequences [10], coronary arteries in motion [121], tires with road contact and deformation [133–135], fluid films [135,136], and VAWTs [48,49,103]. The image-based arterial geometries used in patient-specific arterial FSI computations do not come from the ZSS of the artery. A number of methods [11,26,143–152] have been proposed for estimating the ZSS required in the computations. Using IGA basis functions in space is now a key part of some of the newest ZSS estimation methods [69,150–153] and related shell analysis [154]. The IGA has also been successfully applied to the structural analysis of wind turbine blades [155–159].

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 [127–131], flow-driven string dynamics in turbomachinery [48,49,125,126] and VAWTs [48,49,103] 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 [54,55,137] and spacecraft parachute compressible-flow analysis [139] 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 [69,70,114,118–120] and ventricle-valve-aorta flow analysis [10] 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 [133–135], 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 [135,136], the ST-SI-TC-IGA enables solution with a computational cost comparable to that of the Reynolds-equation model for the comparable solution quality [136]. With that, narrow gaps associated with the road roughness [135] 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 flow analysis of heart valves [69,70,114,118–120], ventricle-valve-aorta flow analysis [10], turbocharger-turbine flow analysis [127–131], spacecraft parachute compressible-flow analysis [139], flow analysis around tires with road contact and deformation [134,135], and flow analysis of a tsunami-shelter VAWT [49,103].

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 MJBS

In the mesh moving method introduced in 1992 [1–3], 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. When the MJBS was first introduced in [1–3], it consisted of simply dropping the 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” in [160]. It was also augmented in [160] 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 [1], and most of the time $\chi = 1$. To also clarify that the “Jacobian” in the name of the method is the mesh Jacobian, in [10] the method was renamed “Mesh-Jacobian-based stiffening.”

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 [1–3], 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 [2,3]). Earlier examples of element layers undergoing rigid-body motion, in combination with deforming structured meshes, were reported in 1990 [14]. 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 [161]. In the SEMMT [11,161–164], 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 [161]: 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 [11,163–165], 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 [11,163–165] and a 2D FSI model problem [163,164].

1.4.4 Path-dependence and cycle-to-cycle accumulated mesh distortion

As mentioned in [9,10], in computing the mesh motion between time levels t_n and t_{n+1} with the linear-elasticity equations, there are two closely related drawbacks associated with computing the displacement from the configuration at t_n . (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 and causes cycle-to-cycle accumulated mesh distortion. Because moving the mesh based on large-deformation mechanics equations, instead of the linear-elasticity equations, is an

alternative to using the BCBMM, we will include here, from [9], an introductory explanation of that alternative. This introductory material will include the components of the mesh moving method based on fiber-reinforced hyperelasticity and optimized ZSS.

1.4.5 Mesh moving based on large-deformation mechanics

Moving the mesh based on large-deformation mechanics equations was used in a number of FSI and other MBI computations with the ST-SUPS and ST-VMS (see, for example, [4–8]). Additional good features of this method are, as pointed out in [4,9], 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 [4], and neo-Hookean in [5–8]. As pointed out in [9], the MJBS can still be part of the mesh moving method, and it was in the computations reported in [7,8].

1.4.6 Element-based mesh relaxation (EBMR)

The EBMR was introduced in 2013 [4]. 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 [4,92,93,95,96,101]. When faced with a loss of mesh integrity, it was proposed in [4] 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 [4], 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 [4]. The EBMR was successfully used in FSI computation of clusters of spacecraft parachutes with modified geometric porosity (see [4]).

1.4.7 Locally-defined ZSS

Locally-defined ZSS originated as arterial ZSS estimation (see [69,148–153]). It was formulated first as the EBZSS in the finite element discretization context [148,149], then as the EBZSS in the isogeometric discretization context [150,151], and then as the integration point based ZSS (IPBZSS) in the isogeometric discretization context [152,153].

In the EBZSS, the ZSS is defined 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 the displacement is measured from that connected configuration. This way of formulating the structural mechanics problem was named “element-based total Lagrangian” (EBTL) method in [148]. The EBTL was used in the EBMR [4].

In the IPBZSS, the way the EBZSS is defined is extended 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. This way of formulating the structural mechanics problem was named “integration-point-based total Lagrangian” (IPBTL) method in [9].

1.4.8 Mesh relaxation and mesh moving based on fiber-reinforced hyperelasticity and optimized ZSS

Mesh relaxation and mesh moving methods based on fiber-reinforced hyperelasticity and optimized ZSS [9] were introduced targeting isogeometric discretization but are also applicable to finite element discretization. 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. The objective of the mesh relaxation is to improve the quality of the mesh after its initial creation and to have an equilibrium state with the optimized ZSS, boundary conditions and the constitutive law. The NURBS mesh used in the computational flow analysis of the tsunami-shelter VAWT [103] was created with the mesh relaxation method.

1.5 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 [11]). 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 [8,99,100,102,104,134,166–169]. Element lengths, including the direction-dependent element lengths, should have node-numbering invariance for all element types, including simplex elements. The direction-dependent element length expression introduced in [168] meets that requirement. It was proven in [170] that the local-length-scale expression introduced in [169] for isogeometric discretization is invariant with respect to element splitting. The expression was used in [89] in calculating the Courant number associated with the gas turbine computation. 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 [76,171–183].

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.

1.6 BCBMM

In the BCBMM, in computing the mesh motion between time levels t_n and t_{n+1} with the linear-elasticity equations, in any cycle, the mesh motion is computed from the configurations in the first cycle. As pointed out in [10], in later cycles, as needed, the configuration the mesh motion is computed from can be changed from the first cycle to a higher cycle. As also pointed out in [10], that would be a good way if there are significant differences between the FSI solutions at different cycles before the solution becomes cyclic or near-cyclic.

1.7 HCBMM

In the HCBMM, for computations where the boundary or interface motion in the second half of the cycle consists of just reversing the steps in the first half and we want the mesh to behave the same way, we modify how we compute the mesh motion in the second half of the first cycle. At any time level in the second half, we compute the mesh motion from the configuration at the corresponding time level in the first half. In subsequent cycles, the mesh motion is computed as in the BCBMM.

1.8 Outline of the remaining sections

The linear-elasticity-based mesh moving method and its versions are described in Sect. 2. The test computations are presented in Sect. 3, and the concluding remark are given in Sect. 4.

2 Mesh moving method

First, for completeness, we include, from [9], the description of the MJBS. Then we describe, partly also from [9], its options related to cycle-to-cycle mesh distortion.

2.1 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 [11], the formulation can be written as

$$\int_{\Omega_t} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\hat{\mathbf{y}}^h - \hat{\mathbf{y}}_t^h) d\Omega = 0. \quad (1)$$

Here Ω is the domain, with the subscript t denoting its time-dependence, $\boldsymbol{\varepsilon}$ is the strain tensor, $\mathbf{w}$ is the test function, the superscript “ h ” indicates that the function is coming from a finite-dimensional space, $\boldsymbol{\sigma}$ is the stress tensor, $\hat{\mathbf{y}}^h$ 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 $\hat{\mathbf{y}}_t^h$ is the displacement from $\mathbf{X}_{\text{REF}}^h$ to the time $\tilde{t}$ configuration $\mathbf{x}_t^h = \mathbf{X}_{\text{REF}}^h + \hat{\mathbf{y}}_t^h$ we compute the mesh motion from. The stress tensor is given as

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

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:

$$J_M = \det \left(\frac{\partial \mathbf{x}_t^h}{\partial \boldsymbol{\xi}} \right), \quad (3)$$

where $\boldsymbol{\xi}$ is the element coordinate. 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 [1–3], it consisted of simply dropping J_M from Eq. (1). That is equivalent to modifying Eq. (1) as

$$\int_{\Omega_i} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\hat{\mathbf{y}}^h - \hat{\mathbf{y}}_i^h) \left(\frac{J_M}{(J_M)_0} \right)^{-1} d\Omega = 0, \quad (4)$$

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 [160] 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:

$$\int_{\Omega_i} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\hat{\mathbf{y}}^h - \hat{\mathbf{y}}_i^h) \left(\frac{J_M}{(J_M)_0} \right)^{-\chi} d\Omega = 0. \quad (5)$$

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

Remark 1 A study was conducted in [3] to see the effect of varying ν on the mesh distortion. For positive values of ν , the best performance was seen in the range $0 \leq \nu \leq \frac{1}{3}$, with the variations over that range making very little difference.

Remark 2 Studies leading to the selection of $\nu = -0.2$ in the computations reported in [100] indicated that the performance for $\nu < 0$ was comparable to what was seen in the range $0 \leq \nu \leq \frac{1}{3}$, and sometimes better.

Remark 3 It was shown in [9,10] that in practical computations the element stiffening method proposed in [166], which is based on element volumes, is approximately equivalent to the MJBS method with $\chi = 1$.

Remark 4 It was noted in [9,10] 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. It was also noted in [9,10] that such an expression is naturally applicable to isogeometric discretization.

2.2 Options related to cycle-to-cycle mesh distortion

The options are defined by how we select $\tilde{t}$ when we are computing the mesh at t_{n+1} . The cycle period is represented by the symbol T .

2.2.1 “TZ”

In this option, $\tilde{t} = t_0$, which corresponds to the initial mesh. There is no cycle-to-cycle accumulated distortion, but, depending on the amount of boundary or interface motion, the distortion within a cycle might reach unacceptable levels.

2.2.2 “TN”

In this option, $\tilde{t} = t_n$. This is the most common option. As pointed out in [10], it gives reasonably good performance for most problems and 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. While this option works well for most problems, because the method is path-dependent, it involves cycle-to-cycle accumulated distortion.

2.2.3 BCBMM

Two versions of the BCBMM were proposed in [10]. Here we will call those versions “BC1” and “BC2.” In BC1,

$$\tilde{t} = t_n - (k - k_{\text{FORW}} - 1)T, \quad (6)$$

where $k = \text{floor}(t_n/T) + 1$ is the current cycle number, and the function $\text{floor}(x)$ returns the greatest integer less than or equal to x . We start with $k_{\text{FORW}} = 0$. With that, in any cycle, we compute the mesh motion based on the first cycle. In later cycles, as needed, we can set k_{FORW} to higher values to compute the mesh motion based on the recent past cycles. As pointed out in [10], that would be a good way if there are significant differences between the FSI solutions at different cycles before the solution becomes cyclic or near-cyclic. In BC2, for $k \geq 2$,

$$\tilde{t} = t_{n+1} - (k - k_{\text{FORW}} - 1)T. \quad (7)$$

2.2.4 HCBMM

If the boundary or interface motion in the second half of the cycle consists of just reversing the steps in the first half, we expect the mesh to behave the same way. For the first cycle, that would be $\mathbf{x}_t^h = \mathbf{x}_{T-t}^h$. With the HCBMM, we can meet that expectation. In the first cycle, we select $\tilde{t}$ as

$$\tilde{t} = \begin{cases} t_n & \text{if } t_n < \frac{T}{2} \\ t_{N-(n+1)} & \text{otherwise} \end{cases}, \quad (8)$$

where N is the number of time steps in T . In subsequent cycles, the mesh motion is computed as in the BCBMM.

3 Wing pitching

As the mesh moving computation test case, we use the mesh motion associated with wing pitching, with the prescribed angular displacement coming from [184]. Figure 1 shows the problem setup. The wing has an airfoil cross-section, with the chord length varying linearly in the spanwise direction, with a

Fig. 1 Wing pitching. Problem setup

Fig. 2 2D case. Mesh. The *green* color indicates the inner mesh. (Color figure online)

Fig. 3 2D case. TZ. $\|f_{AR}\|_2$. Entire mesh (*top*) and inner mesh (*bottom*)

shrinkage factor of 0.6 from the root to the tip. The spanwise length is the same as the root chord length. The pitching motion is a rigid-body rotation around the half-chord point. Working with nondimensional quantities, the angle of attack is given as

$$\theta = \frac{\theta_{\max} + \theta_{\min}}{2} - \frac{\theta_{\max} - \theta_{\min}}{2} \cos(2\pi f t), \quad (9)$$

where f is the pitching frequency, set as $f = 1$ in the test computations, and $\theta_{\max}$ and $\theta_{\min}$ are the maximum and minimum values of the angle of attack, set as $\theta_{\max} = 30^\circ$ and $\theta_{\min} = 10^\circ$. Only in the first cycle, however, as a special case, $\theta_{\max} = 26^\circ$.

Fig. 4 2D case. TN. $\epsilon_{L_2}^h(t, t - (k - 2)T)$ for $k \geq 2$. Entire mesh (top) and inner mesh (bottom)

On the root plane, the 3D computational domain is the same as the 2D computational domain. We use finite element meshes. The step-time size is 0.05. In the MJBS, we set $E = 1$, which has no significance, $\nu = 0.3$, and $\chi = 1$. In the BCBMM, $k_{\text{FORW}} = 0$.

3.1 Mesh quality measures

To measure the cycle-to-cycle mesh distortion, we use the L_2 norm of the error in the mesh position relative to the mesh position in a reference cycle:

$$\epsilon_{L_2}^h(t, t_{\text{ref}}) = \left(\frac{\int_{\Omega_{\text{ref}}} \|\mathbf{x}_t^h - \mathbf{x}_{t_{\text{ref}}}^h\|^2 d\Omega}{\int_{\Omega_{\text{ref}}} d\Omega} \right)^{\frac{1}{2}}, \quad (10)$$

Fig. 5 2D case. TN. $\|\mathbf{f}_{\text{AR}}\|_2$. Entire mesh (top) and inner mesh (bottom)

where t_{ref} is the corresponding time in the reference cycle.

The distortion is measured using the L_2 norm of the relative aspect ratio, represented by the symbol $\|\mathbf{f}_{\text{AR}}\|_2$ and calculated as

$$\|\mathbf{f}_{\text{AR}}\|_2 = \left(\frac{\sum_{e=1}^{n_{\text{el}}} (f_{\text{AR}}^e)^2}{n_{\text{el}}} \right)^{\frac{1}{2}}, \quad (11)$$

$$f_{\text{AR}}^e = \left| \ln \frac{AR^e}{AR_0^e} \right|. \quad (12)$$

Here e is the element counter, n_{el} is the number of the elements, and the subscript 0 indicates the initial mesh.

Fig. 6 2D case. BCBMM. $\|\mathbf{f}_{AR}\|_2$. Entire mesh (top) and inner mesh (bottom)

Fig. 7 2D case. HCBMM. $\|\mathbf{f}_{AR}\|_2$. Entire mesh (top) and inner mesh (bottom)

In 2D, AR^e is defined as

$$AR^e = \frac{(l_{\max}^e)^2}{A^e}, \quad (13)$$

where A^e and $l_{\max}^e$ are the area and maximum edge length of the element. In 3D, it is defined as

$$AR^e = \frac{(l_{\max}^e)^3}{V^e}, \quad (14)$$

where V^e is the volume of the element.

3.2 2D case

Figure 2 shows the mesh. It is made of 12,600 nodes and 6360 quadratic triangular elements. It has an inner mesh of four layers of refined elements.

3.2.1 TZ

When we inspect $\epsilon_{L_2}^h(t, t - (k - 2)T)$ for $k \geq 2$, we see that, as expected, it is zero. There is no cycle-to-cycle accumulated mesh distortion. Figure 3 shows $\|\mathbf{f}_{AR}\|_2$, and we can again see that the mesh motion is cyclic. The distortion within each

Fig. 8 3D case. Mesh. Root plane (*top*) and cut mesh (*bottom*). The green color indicates the inner mesh. (Color figure online)

cycle, however, is reaching higher levels than, as we will see in Sect. 3.2.2, what we get with TN in the first four cycles.

3.2.2 TN

Figure 4 shows $\epsilon_{L_2}^h(t, t - (k - 2)T)$ for $k \geq 2$. We can clearly see the cycle-to-cycle accumulated mesh distortion. Figure 5 shows $\|\mathbf{f}_{AR}\|_2$, and again we can see that the mesh motion is not cyclic. We can also see that, in the first four cycles, the distortion levels reached within each cycle is lower than what we get with TZ, with the difference being more pronounced for the inner mesh.

Fig. 9 3D case. TN. $\epsilon_{L_2}^h(t, t - (k - 2)T)$ for $k \geq 2$. Entire mesh (*top*) and inner mesh (*bottom*)

3.2.3 BCBMM

For both versions, BC1 and BC2, when we inspect $\epsilon_{L_2}^h(t, t - (k - 2)T)$ for $k \geq 2$, we see that, as expected, it is zero. There is no cycle-to-cycle accumulated mesh distortion. Figure 6 shows $\|\mathbf{f}_{AR}\|_2$ for the two versions, and we can again see that the mesh motion is cyclic. We can also see that the distortion levels reached within each cycle is lower than what we get with TZ or TN, significantly lower for the inner mesh.

3.2.4 HCBMM

The airfoil motion in the second half of the cycle consists of just reversing the steps in the first half. The mesh behaves the

Fig. 10 3D case. TN. Mesh displacement magnitude when $\theta = \theta_{\max}$ in the 7th cycle. Root plane (top) and tip plane (bottom)

same way as we can see in Fig. 7, which shows $\|\mathbf{f}_{\text{AR}}\|_2$ for both BC1 and BC2, with a barely detectable departure from the expected behavior for BC1.

3.3 3D case

Figure 8 shows the mesh. It is made of 261,198 nodes and 181,139 quadratic tetrahedral elements. As in 2D, it has an

Fig. 11 3D case. TN. $\|\mathbf{f}_{\text{AR}}\|_2$. Entire mesh (top) and inner mesh (bottom)

inner mesh of four layers of refined elements. The mesh on the root plane is almost the same as the 2D mesh.

3.3.1 TN

Figure 9 shows $\epsilon_{L_2}^h(t, t - (k - 2)T)$ for $k \geq 2$. The behavior, as expected, is similar to what we had in the 2D case. Figure 10 shows the displacement magnitude when $\theta = \theta_{\max}$ in the 7th cycle. Figure 11 shows $\|\mathbf{f}_{\text{AR}}\|_2$, and again the behavior is similar to what we had in the 2D case.

3.3.2 BCBMM

We only compute with BC2, since that is our preferred version. When we inspect $\epsilon_{L_2}^h(t, t - (k - 2)T)$ for $k \geq 2$, we

Fig. 12 3D case. BCBMM. BC2. $\|\mathbf{f}_{AR}\|_2$. Entire mesh (top) and inner mesh (bottom)

Fig. 13 3D case. HCBMM. BC2. $\|\mathbf{f}_{AR}\|_2$. Entire mesh (top) and inner mesh (bottom)

see that, as expected, it is zero. There is no cycle-to-cycle accumulated mesh distortion. Figure 12 shows $\|\mathbf{f}_{AR}\|_2$, and we can again see that the mesh motion is cyclic.

3.3.3 HCBMM

Again, we only compute with BC2. As in 2D, the wing motion in the second half of the cycle consists of just reversing the steps in the first half. The mesh behaves the same way as we can see in Fig. 13, which shows $\|\mathbf{f}_{AR}\|_2$.

4 Concluding remarks

Mesh moving based on linear elasticity and MJBS has been in wide use with moving-mesh methods in computation of flow problems with FSI and MBI. In the MJBS, the smaller elements, which are typically placed near solid surfaces, are protected more than the larger ones. In computing the mesh motion between time levels t_n and t_{n+1} with the linear-elasticity equations, computing the displacement from the configuration at t_n , which is a very common practice, leads to a path-dependent mesh motion and results in cycle-to-cycle accumulated mesh distortion. The BCBMM was introduced to remedy that and has no cycle-to-cycle accumulated dis-

tortion. In the BCBMM, in any cycle, the mesh motion is computed from the configurations in the first cycle. We have, for the first time, carried out mesh moving test computations with the BCBMM. We have also introduced the HCBMM, which is a special-case version of the BCBMM. The HCBMM is for computations where the boundary or interface motion in the second half of the cycle consists of just reversing the steps in the first half and we want the mesh to behave the same way. In the HCBMM, at any time level in the second half of the first cycle, we compute the mesh motion from the configuration at the corresponding time level in the first half. In subsequent cycles, the mesh motion is computed as in the BCBMM. We have presented detailed 2D and 3D test computations with finite element meshes, using as the test case the mesh motion associated with wing pitching. The computations show that all versions of the BCBMM perform well, with no cycle-to-cycle accumulated distortion, and with the HCBMM, as the wing in the second half of the cycle just reverses its motion steps in the first half, the mesh behaves the same way.

Acknowledgements This work was supported (third author) 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 work was also supported (fourth author) in part by ARO Grant W911NF-17-1-0046 and Top Global University Project of Waseda University. The first author was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES) - Finance Code 001.

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/>.

References

1. 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. In: New methods in transient analysis, PVP-Vol.246/AMD-Vol.143. ASME, New York, pp 7–24
2. 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>
3. 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)
4. 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>
5. 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>
6. 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>
7. 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>
8. 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>
9. Takizawa K, Tezduyar TE, Avsar R (2020) A low-distortion mesh moving method based on fiber-reinforced hyperelasticity and optimized zero-stress state. *Comput Mech* 65:1567–1591. <https://doi.org/10.1007/s00466-020-01835-z>
10. 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* 65:1343–1363. <https://doi.org/10.1007/s00466-020-01822-4>
11. Bazilevs Y, Takizawa K, Tezduyar TE (2013) Computational fluid–structure interaction: methods and applications. Wiley, February ISBN 978-0470978771
12. 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>
13. 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>
14. 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)
15. 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>
16. 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>
17. 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
18. 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>
19. 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>
20. 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
21. Hughes TJR, Oberai AA, Mazzei L (2001) Large eddy simulation of turbulent channel flows by the variational multiscale method. *Phys Fluids* 13:1784–1799

22. 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
23. 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
24. Hughes TJR, Liu WK, Zimmermann TK (1981) Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Comput Methods Appl Mech Eng* 29:329–349
25. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y (2008) Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Comput Mech* 43:3–37
26. 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>
27. 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 Methods Appl Sci* 22(suppl2):1230002. <https://doi.org/10.1142/S0218202512300025>
28. 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>
29. 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>
30. 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>
31. 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)
32. Bazilevs Y, Hughes TJR (2007) Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Comput Fluids* 36:12–26
33. 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
34. 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
35. Bazilevs Y, Hughes TJR (2008) NURBS-based isogeometric analysis for the computation of flows about rotating components. *Comput Mech* 43:143–150
36. Hsu M-C, Bazilevs Y (2012) Fluid-structure interaction modeling of wind turbines: simulating the full machine. *Comput Mech* 50:821–833
37. 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>
38. 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
39. Hsu M-C, Akkerman I, Bazilevs Y (2011) High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Comput Fluids* 49:93–100
40. 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
41. 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
42. 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
43. 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>
44. 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>
45. 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>
46. 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*. Springer, Berlin, pp 253–336, ISBN 978-3-319-96468-3. https://doi.org/10.1007/978-3-319-96469-0_7
47. 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>
48. Bazilevs Y, Takizawa K, Tezduyar TE, Hsu M-C, Otoguro Y, Mochizuki H, Wu MCH (2020) Wind turbine and turbomachinery computational analysis with the ALE and space-time variational multiscale methods and isogeometric discretization. *J Adv Eng Comput* 4:1–32. <https://doi.org/10.25073/jaec.202041.278>
49. Bazilevs Y, Takizawa K, Tezduyar TE, Hsu M-C, Otoguro Y, Mochizuki H, Wu MCH (2020) ALE and space-time variational multiscale isogeometric analysis of wind turbines and turbomachinery. In: Grama A, Sameh A (ed) *Parallel algorithms in computational science and engineering, modeling and simulation in science, engineering and technology*. Springer, Berlin, pp 195–233, ISBN 978-3-030-43735-0. https://doi.org/10.1007/978-3-030-43736-7_7
50. 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>
51. 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>
52. 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>
53. 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>
54. Takizawa K, Bazilevs Y, Tezduyar TE, Korobenko A (2020) Variational multiscale flow analysis in aerospace, energy and

- transportation technologies. *J Adv Eng Comput* 4:83–117. <https://doi.org/10.25073/jaec.202042.279>
55. Takizawa K, Bazilevs Y, Tezduyar TE, Korobenko A (2020) Variational multiscale flow analysis in aerospace, energy and transportation technologies. In: Grama A, Sameh A (eds) *Parallel algorithms in computational science and engineering, modeling and simulation in science, engineering and technology*. Springer, Berlin, pp 235–280, ISBN 978-3-030-43735-0. https://doi.org/10.1007/978-3-030-43735-0_8
 56. 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
 57. 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
 58. 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
 59. 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
 60. 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
 61. 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
 62. Hsu M-C, Bazilevs Y (2011) Blood vessel tissue prestress modeling for vascular fluid-structure interaction simulations. *Finite Elem Anal Des* 47:593–599
 63. 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>
 64. 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>
 65. 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>
 66. 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>
 67. 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>
 68. 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
 69. 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
 70. Hughes TJR, Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C (2020) Computational cardiovascular analysis with the variational multiscale methods and isogeometric discretization. In: Grama A, Sameh A (eds) *Parallel algorithms in computational science and engineering, modeling and simulation in science, engineering and technology*. Springer, Berlin, pp 151–193, ISBN 978-3-030-43735-0. https://doi.org/10.1007/978-3-030-43735-0_6
 71. 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
 72. 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
 73. 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>
 74. 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
 75. 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>
 76. 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 turbomachinery. *Math Models Methods Appl Sci* 29:967–994. <https://doi.org/10.1142/S0218202519410057>
 77. 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>
 78. 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>
 79. Helgedagsrud TA, Bazilevs Y, Mathisen KM, Oiseth OA (2019) Computational and experimental investigation of free vibration and flutter of bridge decks. *Comput Mech* 63:121–136. <https://doi.org/10.1007/s00466-018-1587-4>
 80. Helgedagsrud TA, Bazilevs Y, Korobenko A, Mathisen KM, Oiseth OA (2019) Using ALE-VMS to compute aerodynamic derivatives of bridge sections. *Comput Fluids* 179:820–832. <https://doi.org/10.1016/j.compfluid.2018.04.037>
 81. 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)
 82. Kamensky D, Evans JA, Hsu M-C, Bazilevs Y (2017) Projection-based stabilization of interface Lagrange multipliers in immersogeometric 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>
 83. 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>
 84. Wu MCH, Muchowski HM, Johnson EL, Rajanna MR, Hsu M-C (2019) Immersogeometric fluid-structure interaction modeling and simulation of transcatheter aortic valve replacement. *Comput Methods Appl Mech Eng* 357:112556
 85. Johnson EL, Wu MCH, Xu F, Wiese NM, Rajanna MR, Herrema AJ, Ganapathysubramanian B, Hughes TJR, Sacks MS, Hsu M-C (2020) Thinner biological tissues induce leaflet flutter in aortic heart valve replacements. *Proc Natl Acad Sci* 117:19007–19016

86. 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>
87. 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>
88. 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>
89. Bazilevs Y, Takizawa K, Wu MCH, Kuraishi T, Aysar R, Xu Z, Tezduyar TE (2020) Gas turbine computational flow and structure analysis with isogeometric discretization and a complex-geometry mesh generation method. *Comput Mech* (to appear). <https://doi.org/10.1007/s00466-020-01919-w>
90. 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>
91. 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>
92. 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>
93. 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>
94. 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>
95. 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>
96. 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>
97. 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>
98. 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>
99. 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>
100. 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>
101. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C, Øiseth 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>
102. 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>
103. Otoguro Y, Mochizuki H, Takizawa K, Tezduyar TE (2020) Space-time variational multiscale isogeometric analysis of a tsunami-shelter vertical-axis wind turbine. *Comput Mech* 66:1443–1460. <https://doi.org/10.1007/s00466-020-01910-5>
104. 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>
105. 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>
106. 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>
107. 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>
108. 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>
109. 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>
110. 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>
111. 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>
112. 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>
113. 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, ISBN 978-3-319-40825-5. https://doi.org/10.1007/978-3-319-40827-9_29
114. 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, ISBN 978-3-319-96468-3. https://doi.org/10.1007/978-3-319-96469-0_2
115. 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>
116. 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>
 117. 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, ISBN 978-3-319-40825-5. https://doi.org/10.1007/978-3-319-40827-9_13
 118. 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, ISBN 978-3-319-59547-4. https://doi.org/10.1007/978-3-319-59548-1_6
 119. 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>
 120. 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* 65:1167–1187. <https://doi.org/10.1007/s00466-019-01813-0>
 121. 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>
 122. 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>
 123. 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>
 124. 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>
 125. 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>
 126. 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>
 127. 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>
 128. 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>
 129. 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, ISBN 978-3-319-96468-3. https://doi.org/10.1007/978-3-319-96469-0_10
 130. 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>
 131. 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>
 132. 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>
 133. 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, ISBN 978-3-319-96468-3. https://doi.org/10.1007/978-3-319-96469-0_8
 134. 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>
 135. 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>
 136. 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>
 137. 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>
 138. 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>
 139. 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>
 140. 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)
 141. 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
 142. 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>

143. Tezduyar TE, Cragin T, Sathe S, Nanna B (2007) FSI computations in arterial fluid mechanics with estimated zero-pressure arterial geometry. In: Onate E, Garcia J, Bergan P, Kvamsdal T (eds) *Marine 2007*. CIMNE, Barcelona, Spain
144. Tezduyar TE, Sathe S, Schwaab M, Conklin BS (2008) Arterial fluid mechanics modeling with the stabilized space-time fluid-structure interaction technique. *Int J Numer Methods Fluids* 57:601–629. <https://doi.org/10.1002/fld.1633>
145. Takizawa K, Christopher J, Tezduyar TE, Sathe S (2010) Space-time finite element computation of arterial fluid-structure interactions with patient-specific data. *Int J Numer Methods Biomed Eng* 26:101–116. <https://doi.org/10.1002/cnm.1241>
146. Takizawa K, Moorman C, Wright S, Purdue J, McPhail T, Chen PR, Warren J, Tezduyar TE (2011) Patient-specific arterial fluid-structure interaction modeling of cerebral aneurysms. *Int J Numer Methods Fluids* 65:308–323. <https://doi.org/10.1002/fld.2360>
147. Tezduyar TE, Takizawa K, Brummer T, Chen PR (2011) Space-time fluid-structure interaction modeling of patient-specific cerebral aneurysms. *Int J Numer Methods Biomed Eng* 27:1665–1710. <https://doi.org/10.1002/cnm.1433>
148. 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>
149. 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>
150. 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, ISBN 978-3-319-59547-4. https://doi.org/10.1007/978-3-319-59548-1_7
151. 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>
152. 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>
153. 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>
154. 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>
155. Bazilevs Y, Hsu M-C, Kiendl J, Benson DJ (2012) A computational procedure for pre-bending of wind turbine blades. *Int J Numer Methods Eng* 89:323–336
156. Bazilevs Y, Deng X, Korobenko A, di Scalea FL, Todd MD, Taylor SG (2015) Isogeometric fatigue damage prediction in large-scale composite structures driven by dynamic sensor data. *J Appl Mech* 82:091008
157. Herrema AJ, Johnson EL, Proserpio D, Wu MCH, Kiendl J, Hsu M-C (2019) Penalty coupling of non-matching isogeometric Kirchhoff-Love shell patches with application to composite wind turbine blades. *Comput Methods Appl Mech Eng* 346:810–840
158. Herrema AJ, Kiendl J, Hsu M-C (2019) A framework for isogeometric-analysis-based optimization of wind turbine blade structures. *Wind Energy* 22:153–170
159. Johnson EL, Hsu M-C (2020) Isogeometric analysis of ice accretion on wind turbine blades. *Comput Mech* 66:311–322
160. 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>
161. 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
162. 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, Singapore, pp 221–239. https://doi.org/10.1142/9789812796837_0013
163. 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>
164. 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>
165. Stein K, Tezduyar T (2002) Advanced mesh update techniques for problems involving large displacements. In: *Proceedings of the fifth world congress on computational mechanics*. Paper-ID: 81489. <http://www.researchgate.net/publication/303737884/>, Vienna, Austria
166. 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>
167. 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>
168. 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>
169. Otoguro Y, Takizawa K, Tezduyar TE (2020) Element length calculation in B-spline meshes for complex geometries. *Comput Mech* 65:1085–1103. <https://doi.org/10.1007/s00466-019-01809-w>
170. Ueda Y, Otoguro Y, Takizawa K, Tezduyar TE (2020) Element-splitting-invariant local-length-scale calculation in B-spline meshes for complex geometries. *Math Models Methods Appl Sci* 30:2139–2174. <https://doi.org/10.1142/S0218202520500402>
171. 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>
172. 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>
173. 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>
174. 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>
175. 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>
176. 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>
 177. 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>
 178. 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>
 179. 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>
 180. 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>
 181. Castorrini A, Corsini A, Rispoli F, Venturini P, Takizawa K, Tezduyar TE (2016) Computational analysis of wind-turbineblade rain erosion. *Comput Fluids* 141:175–183. <https://doi.org/10.1016/j.compfluid.2016.08.013>
 182. 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>
 183. Castorrini A, Venturini P, Corsini A, Rispoli F, Takizawa K, Tezduyar TE (2020) Computational analysis of particle-laden-airflow erosion and experimental verification. *Comput Mech* 65:1549–1565. <https://doi.org/10.1007/s00466-020-01834-0>
 184. Mittal S, Tezduyar TE (1992) A finite element study of incompressible flows past oscillating cylinders and aerofoils. *Int J Numer Methods Fluids* 15:1073–1118. <https://doi.org/10.1002/flid.1650150911>

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