Computational Mechanics (2019) 64:1403–1419
https://doi.org/10.1007/s00466-019-01722-2

ORIGINAL PAPER

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

Yuto Otoguro[1] · Kenji Takizawa[1] · Tayfun E. Tezduyar[2,3] · Kenichiro Nagaoka[1] · Reha Avsar[2] · Yutong Zhang[2]

Received: 6 March 2019 / Accepted: 9 May 2019 / Published online: 22 May 2019

Abstract

Many of the computational challenges encountered in turbocharger-turbine flow analysis have been addressed by an integrated set of space–time (ST) computational methods. The core computational method is the ST variational multiscale (ST-VMS) method. The ST framework provides higher-order accuracy in general, and the VMS feature of the ST-VMS addresses the computational challenges associated with the multiscale nature of the unsteady flow. The moving-mesh feature of the ST framework enables high-resolution computation near the rotor surface. The ST slip interface (ST-SI) method enables moving-mesh computation of the spinning rotor. The mesh covering the rotor spins with it, and the SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the solution. The ST Isogeometric Analysis enables more accurate representation of the turbine geometry and increased accuracy in the flow solution. The ST/NURBS Mesh Update Method enables exact representation of the mesh rotation. A general-purpose NURBS mesh generation method makes it easier to deal with the complex geometries involved. An SI also provides mesh generation flexibility in a general context by accurately connecting the two sides of the solution computed over nonmatching meshes, and that is enabling the use of nonmatching NURBS meshes in the computations. The computational analysis needs to cover a full intake/exhaust cycle, which is much longer than the turbine rotation cycle because of high rotation speeds, and the long duration required is an additional computational challenge. As one way of addressing that challenge, we propose here to calculate the turbine efficiency for the intake/exhaust cycle by interpolation from the efficiencies associated with a set of steady-inflow computations at different flow rates. The efficiencies obtained from the computations with time-dependent and steady-inflow representations of the intake/exhaust cycle compare well. This demonstrates that predicting the turbine performance from a set of steady-inflow computations is a good way of addressing the challenge associated with the multiple time scales.

Keywords Turbocharger · Turbine · Space–time variational multiscale method (ST-VMS) · ST slip interface method (ST-SI) · ST isogeometric analysis (ST-IGA) · Steady-inflow representation

1 Introduction

The challenges faced in computational flow analysis of turbocharger turbines include unsteady flow through a complex geometry, the need for high-resolution flow representation near the rotor surface, high Reynolds numbers, and multiscale flow behavior. The flow unsteadiness comes from the intake/exhaust cycle and the flow in the turbine. An additional challenge is associated with the multiple time scales involved. The computational analysis needs to cover a full intake/exhaust cycle, which is much longer than the turbine rotation cycle because of high rotation speeds. Therefore computations with time-dependent representation of the

✉ Kenji Takizawa
Kenji.Takizawa@tafsm.org

Tayfun E. Tezduyar
tezduyar@tafsm.org

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

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

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

intake/exhaust cycle require long-durations in the turbine time scale.

The parts of these challenges faced in turbine computations in a more general context have been addressed also by other researchers, with approaches ranging from using a single blade with spatially-periodic boundary conditions (see, e.g., [1–8]) to "sliding interfaces" (see, e.g., [9–12]). The precursors of the work presented in this article were reported in [13–16]. The challenge associated with the multiple time scales was addressed in [16] by going through with the long-duration computation. Here we propose to address that challenge in a different way, by calculating the turbine efficiency from computations with steady-inflow representation of the intake/exhaust cycle.

The core computational method is the space–time variational multiscale (ST-VMS) method [17–19], and the other key methods are the ST Slip Interface (ST-SI) method [20,21], ST Isogeometric Analysis (ST-IGA) [13,17,22], ST/NURBS Mesh Update Method (STNMUM) [22–25], and a general-purpose NURBS mesh generation method for complex geometries [14,15].

1.1 ST-VMS and ST-SUPS

The deforming-spatial-domain/stabilized ST (DSD/SST) method [26–28] was introduced for computation of flows with moving boundaries and interfaces (MBI), including fluid–structure interactions (FSI). In MBI computations the DSD/SST functions as a moving-mesh method. Moving the fluid mechanics mesh to follow an interface enables mesh-resolution control near the interface and, consequently, high-resolution boundary-layer representation near fluid–solid interfaces. The stabilization components of the original DSD/SST are the Streamline-Upwind/Petrov–Galerkin (SUPG) [29] and Pressure-Stabilizing/Petrov–Galerkin (PSPG) [26] stabilizations, which are used widely. Because of the SUPG and PSPG components, the original DSD/SST is now called "ST-SUPS." The ST-VMS is the VMS version of the DSD/SST. The VMS components of the ST-VMS are from the residual-based VMS (RBVMS) method [30–33]. 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 [17,18]), are desirable also in computations without MBI.

The DSD/SST is an alternative to the Arbitrary Lagrangian–Eulerian (ALE) method, which is an older and more commonly used moving-mesh method. The ALE-VMS method [34–40] is the VMS version of the ALE. It succeeded the ST-SUPS [26] and ALE-SUPS [41] 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 no-slip boundary conditions [42–44], "sliding interfaces" [45,46] and backflow stabilization [47]. They have been applied to many classes of FSI, MBI and fluid mechanics problems. The classes of problems include wind-turbine aerodynamics and FSI [9–12,48–54], more specifically, vertical-axis wind turbines [11,12,55,56], floating wind turbines [57], wind turbines in atmospheric boundary layers [11,12,54,58], and fatigue damage in wind-turbine blades [59], patient-specific cardiovascular fluid mechanics and FSI [34,60–65], biomedical-device FSI [66–71], ship hydrodynamics with free-surface flow and fluid–object interaction [72,73], hydrodynamics and FSI of a hydraulic arresting gear [74,75], hydrodynamics of tidal-stream turbines with free-surface flow [76], bioinspired FSI for marine propulsion [77,78], bridge aerodynamics and fluid–object interaction [79–81], and mixed ALE-VMS/Immersogeometric computations [69–71,82,83] in the framework of the Fluid–Solid Interface-Tracking/Interface-Capturing Technique [84]. Recent advances in stabilized and multiscale methods may be found for stratified incompressible flows in [85], for divergence-conforming discretizations of incompressible flows in [86], and for compressible flows with emphasis on gas-turbine modeling in [87].

The ST-SUPS and ST-VMS have also been applied to many classes of FSI, MBI and fluid mechanics problems (see [88] for a comprehensive summary). The classes of problems include spacecraft parachute analysis for the landing-stage parachutes [37,89–92], cover-separation parachutes [93] and the drogue parachutes [94–96], wind-turbine aerodynamics for horizontal-axis wind-turbine rotors [9,37,97,98], full horizontal-axis wind-turbines [10,25,99,100] and vertical-axis wind-turbines [11,12,20], flapping-wing aerodynamics for an actual locust [22,23,37,101], bioinspired MAVs [24,99,100,102] and wing-clapping [103,104], blood flow analysis of cerebral aneurysms [99,105], stent-blocked aneurysms [105–107], aortas [108–111] and heart valves [100,103,110,112–115], spacecraft aerodynamics [93,116], thermo-fluid analysis of ground vehicles and their tires [19,113], thermo-fluid analysis of disk brakes [21], flow-driven string dynamics in turbomachinery [117–119], flow analysis of turbocharger turbines [13–16], flow around tires with road contact and deformation [113,120–122], lubrication fluid dynamics [123], ram-air parachutes [124], and compressible-flow spacecraft parachute aerodynamics [125,126].

In the flow analysis presented here, the ST framework provides higher-order accuracy in a general context. The VMS feature of the ST-VMS addresses the computational challenges associated with the multiscale nature of the unsteady flow. The moving-mesh feature of the ST framework enables high-resolution computation near the rotor surface.

1.2 ST-SI

The ST-SI was introduced in [20], in the context of incompressible-flow equations, to retain the desirable moving-mesh features of the ST-VMS and ST-SUPS when we have spinning solid surfaces, such as a turbine rotor. The mesh covering the spinning surface spins with it, retaining the high-resolution representation of the boundary layers. The starting point in the development of the ST-SI was the version of the ALE-VMS for computations with sliding interfaces [45,46]. Interface terms similar to those in the ALE-VMS version are added to the ST-VMS to account for the compatibility conditions for the velocity and stress at the SI. That accurately connects the two sides of the solution. An ST-SI version where the SI is between fluid and solid domains was also presented in [20]. The SI in this case is a "fluid–solid SI" rather than a standard "fluid–fluid SI" and enables weak enforcement of the Dirichlet boundary conditions for the fluid. The ST-SI introduced in [21] for the coupled incompressible-flow and thermal-transport equations retains the high-resolution representation of the thermo-fluid boundary layers near spinning solid surfaces. These ST-SI methods have been applied to aerodynamic analysis of vertical-axis wind turbines [11,12,20], thermo-fluid analysis of disk brakes [21], flow-driven string dynamics in turbomachinery [117–119], flow analysis of turbocharger turbines [13–16], flow around tires with road contact and deformation [113,120–122], lubrication fluid dynamics [123], aerodynamic analysis of ram-air parachutes [124], and flow analysis of heart valves [110,114,115].

In the ST-SI version presented in [20] the SI is between a thin porous structure and the fluid on its two sides. This enables dealing with the porosity in a fashion consistent with how the standard fluid–fluid SIs are dealt with and how the Dirichlet conditions are enforced weakly with fluid–solid SIs. This version also enables handling thin structures that have T-junctions. This method has been applied to incompressible-flow aerodynamic analysis of ram-air parachutes with fabric porosity [124]. The compressible-flow ST-SI methods were introduced in [125], including the version where the SI is between a thin porous structure and the fluid on its two sides. Compressible-flow porosity models were also introduced in [125]. These, together with the compressible-flow ST SUPG method [127], extended the ST computational analysis range to compressible-flow aerodynamics of parachutes with fabric and geometric porosities. That enabled ST computational flow analysis of the Orion spacecraft drogue parachute in the compressible-flow regime [125,126].

In the computations here, the ST-SI enables spinning the mesh covering the solid surface together with the surface. This retains the high-resolution representation of the flow near the surface, and the SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the solution.

1.3 ST-IGA and STNMUM

The ST-IGA is the integration of the ST framework with isogeometric discretization. It was introduced in [17]. Computations with the ST-VMS and ST-IGA were first reported in [17] in a 2D context, with IGA basis functions in space for flow past an airfoil, and in both space and time for the advection equation. Using higher-order basis functions in time enables getting full benefit out of using higher-order basis functions in space. This was demonstrated with the stability and accuracy analysis given in [17] 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 [17,18] and demonstrated in [22–24]. 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 STNMUM [22–24]; the name was given in [25]. 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" [17,18,22,37] 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 [19,21,113,117–119,128]. 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 [22,23,37,101], bioinspired MAVs [24,99,100,102] and wing-clapping [103,104], separation aerodynamics of spacecraft [93], aerodynamics of horizontal-axis [10,25,99,100] and vertical-axis [11,12,20] wind-turbines, thermo-fluid analysis of ground vehicles and their tires [19,113], thermo-fluid analysis of disk brakes [21], flow-driven string dynamics in turbomachinery [117–119], and flow analysis of turbocharger turbines [13–16].

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 [13–16], flow-driven string dynamics in turbomachinery [118,119], ram-air parachutes

[124], spacecraft parachutes [126], aortas [110,111], heart valves [110,114,115], tires with road contact and deformation [121,122], and lubrication fluid dynamics [123]. The image-based arterial geometries used in patient-specific arterial FSI computations do not come from the zero-stress state (ZSS) of the artery. A number of methods [35,37,129–138] 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 [136–139] and related shell analysis [140].

In the flow analysis presented here, the ST-IGA enables more accurate representation of the turbine geometry, increased accuracy in the flow solution, and using larger time-step sizes. The STNMUM enables exact representation of the mesh rotation.

1.4 General-purpose NURBS mesh generation method

To make the ST-IGA use, and in a wider context the IGA use, even more practical in computational flow analysis with complex geometries, NURBS volume mesh generation needs to be easier and more automated. To that end, a general-purpose NURBS mesh generation method was introduced in [14]. The method is based on multi-block-structured mesh generation with existing techniques, projection of that mesh to a NURBS mesh made of patches that correspond to the blocks, and recovery of the original model surfaces. The recovery of the original surfaces is to the extent they are suitable for accurate and robust fluid mechanics computations. The method is expected to retain the refinement distribution and element quality of the multi-block-structured mesh that we start with. Because there are ample good techniques and software for generating multi-block-structured meshes, the method makes general-purpose mesh generation relatively easy. Mesh-quality performance studies for 2D and 3D meshes, including those for complex models, were presented in [15]. A test computation for a turbocharger turbine and exhaust manifold was also presented in [15], with a more detailed computation in [16]. The mesh generation method was used also in the pump-flow analysis part of the flow-driven string dynamics presented in [119]. The performance studies and test computations demonstrated that the general-purpose NURBS mesh generation method makes the IGA use in fluid mechanics computations even more practical.

The general-purpose NURBS mesh generation method is used also in the turbocharger-turbine flow analysis presented here.

1.5 ST-SI-IGA

An SI provides mesh generation flexibility in a general context by accurately connecting the two sides of the solution computed over nonmatching meshes. This type of mesh generation flexibility is especially valuable in complex-geometry flow computations with isogeometric discretization, removing the matching requirement between the NURBS patches without loss of accuracy. This feature was used in the flow analysis of heart valves [110,114,115], turbocharger turbines [13–16], spacecraft parachute compressible-flow analysis [126], and flow around tires with road contact and deformation [122]. It is used also in the turbocharger-turbine flow analysis presented here.

1.6 Steady-inflow representation of the intake/exhaust cycle

In calculating the turbine efficiency from the computations with the steady-inflow representation of the intake/exhaust cycle, we first perform a set of steady-inflow computations at different flow rates. We then obtain the turbine efficiencies associated with those computations and tabulate them as a function of the tip-speed ratio. Next we calculate the tip-speed ratio for the moving-averaged inflow velocity coming from the time-dependent representation of the intake/exhaust cycle. The turbine efficiency corresponding to that time-dependent tip-speed ratio is obtained by interpolation from the tabulated values of the steady-inflow turbine efficiencies.

1.7 Computations presented

For comparison purposes, we include from [16] some of the results reported for the computation with the time-dependent representation of the intake/exhaust cycle. The steady-inflow computations are carried out at 12 different flow rates.

1.8 Outline of the remaining sections

The governing equations are given in Sect. 2. For completeness, in Sect. 3 we describe the ST-VMS. The computations are presented in Sect. 4, and the concluding remarks in Sect. 5. In the Appendix, we describe the ST-SI and give the stabilization parameters used in the ST-VMS and ST-SI.

2 Governing equations

Let $\Omega_t \subset \mathbb{R}^{n_{\mathrm{sd}}}$ be the spatial domain with boundary Γ_t at time $t \in (0, T)$, where n_{sd} is the number of space dimensions. The subscript t indicates the time-dependence of the domain.

The Navier–Stokes equations of incompressible flows are written on Ω_t and $\forall t \in (0, T)$ as

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

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

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

3 ST-VMS

For completeness, we include, mostly from [17], the ST-VMS:

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}Q \\
&\quad + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&\quad - \int_{(P_n)_{\mathrm{h}}} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}P \\
&\quad + \int_{Q_n} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h \mathrm{d}Q + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) \mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \frac{\tau_{\mathrm{SUPS}}}{\rho} \left[\rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{w}^h \right) + \boldsymbol{\nabla} q^h \right] \\
&\qquad\qquad\qquad \cdot \mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&\quad + \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \nu_{\mathrm{LSIC}} \boldsymbol{\nabla} \cdot \mathbf{w}^h \rho r_{\mathrm{C}}(\mathbf{u}^h) \mathrm{d}Q \\
&\quad - \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \tau_{\mathrm{SUPS}} \mathbf{w}^h \cdot \left(\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) \cdot \boldsymbol{\nabla}\mathbf{u}^h \right) \mathrm{d}Q \\
&\quad - \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \frac{\tau_{\mathrm{SUPS}}^2}{\rho} \mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) \cdot \left(\boldsymbol{\nabla}\mathbf{w}^h \right) \cdot \mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) \mathrm{d}Q \\
&= 0,
\end{aligned} \tag{3}$$

where

$$\mathbf{r}_{\mathrm{M}}(\mathbf{u}^h, p^h) = \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{u}^h - \mathbf{f}^h \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\mathbf{u}^h, p^h), \tag{4}$$

$$r_{\mathrm{C}}(\mathbf{u}^h) = \boldsymbol{\nabla} \cdot \mathbf{u}^h \tag{5}$$

are the residuals of the momentum equation and incompressibility constraint. The test functions associated with the velocity and pressure are $\mathbf{w}$ and q. A superscript "h" indicates that the function is coming from a finite-dimensional space. The symbol Q_n represents the ST slice between time levels n and $n+1$, $(P_n)_{\mathrm{h}}$ is the part of the lateral boundary of that slice associated with the stress boundary condition $\mathbf{h}$, and Ω_n is the spatial domain at time level n. The superscript "e" is the ST element counter, and n_{el} is the number of ST elements. The functions are discontinuous in time at each time level, and the superscripts "$-$" and "$+$" indicate the values of the functions just below and just above the time level.

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

We also use in the computations the ST-SI, which we include in Appendix A. The stabilization parameters, embedded in the ST-VMS and ST-SI, are included in Appendix B.

4 Computations

4.1 Problem setup

The computation with the time-dependent representation of the full intake/exhaust cycle was reported in [16]. Here we carry out the computations with the steady-inflow representation of the intake/exhaust cycle, at 12 different values of the tip-speed ratio: $r\omega/U_{\mathrm{IN}}$, where $r\omega$ and U_{IN} are the tip speed and inflow velocity. We will compare the efficiencies extracted from the computations with these two representations.

Figure 1 shows the turbocharger-turbine model used in the computations with the steady-inflow representation. The model consists of the volute, stator, turbine and the exhaust pipe. The computations with the time-dependent representation had an exhaust gas purifier, instead of an exhaust pipe, and it also had an exhaust manifold. When we compare the efficiencies from the computations with the two representations, we use the same flow region.

The rotor diameter is 30 mm and the rotor speed is 30,000 rpm, which translates to a turbine rotation period of $T_{\mathrm{TR}} = 2.0\times10^{-3}$ s. The gas density and kinematic viscosity are 0.9 kg/m^3 and 2.8×10^{-5} m^2/s. The tip-speed ratios are 0.333, 0.455, 0.625, 0.714, 0.833, 1.0, 1.25, 1.667, 2.5, 3.333, 4.0 and 5.0. In setting those tip-speed ratios, we use a constant tip speed $r\omega = 47.1$ m/s, and vary the inflow speed accordingly.

We note that for the operational conditions we have here, incompressible-flow modeling is reasonable. Besides, comparing the laboratory test data from scaled-up models to data from actual-scale models, it was observed that the Mach number made little difference in the turbine efficiency

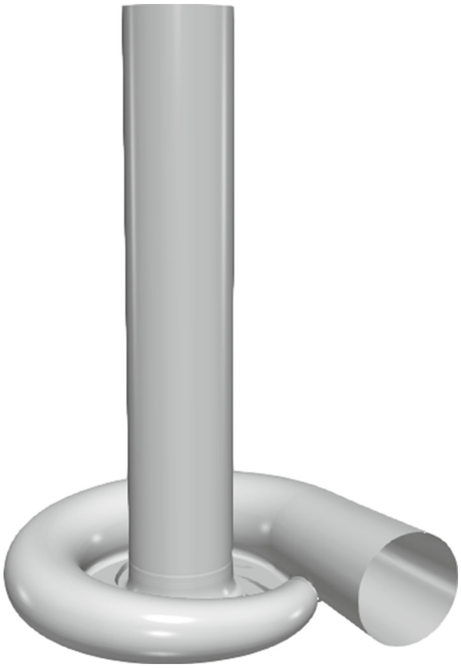

Fig. 1 Turbocharger-turbine model. The volute, stator, turbine and the exhaust pipe

4.2 Efficiency definition

In reporting the turbine efficiency, we first define two quantities, one instantaneous and the other interval-based:

$$K(t) = \int_{\Gamma_{\mathrm{INF}}} (-\mathbf{n}\cdot\mathbf{u}) \left(\frac{1}{2}\rho\|\mathbf{u}\|^2 + p\right) \mathrm{d}\Gamma + \int_{\Gamma_{\mathrm{OUTF}}} (-\mathbf{n}\cdot\mathbf{u})\, p \mathrm{d}\Gamma, \tag{6}$$

$$K_{\mathrm{IB}}(t_1, t_2) = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} K(t)\mathrm{d}t, \tag{7}$$

where Γ_{INF} and Γ_{OUTF} are the inflow and outflow boundaries used in the efficiency calculation. Similarly,

$$P_{\mathrm{IB}}(t_3, t_4) = \frac{1}{t_4 - t_3}\int_{t_3}^{t_4} P(t)\mathrm{d}t, \tag{8}$$

where $P(t)$ is the instantaneous power extracted from the turbine. Then, a general form of the interval-based efficiency is written as

$$\eta_{\mathrm{IB}}(t_1, t_2, t_3, t_4) = \frac{P_{\mathrm{IB}}(t_3, t_4)}{K_{\mathrm{IB}}(t_1, t_2)}. \tag{9}$$

We note that we selected the locations of Γ_{INF} and Γ_{OUTF} with the objective of having the smallest possible domain in the efficiency calculation so that we minimize the effect of the manifold and gas purifier.

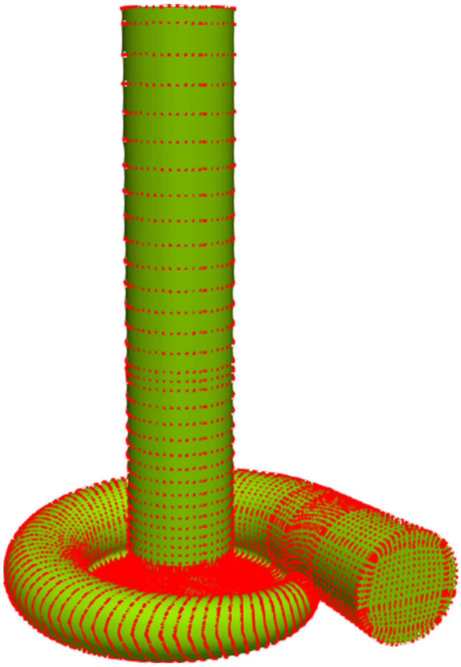

Fig. 2 Quadratic NURBS control mesh for the turbocharger-turbine model

Table 1 Number of control points (*nc*) and elements (*ne*) in different parts of the mesh

Part	*nc*	*ne*
Volute and stator	116, 945	73, 037
Turbine	71, 445	40, 380
Exhaust pipe	28, 776	21, 600
Total	217, 166	135, 017

4.3 Mesh

Figure 2 shows the quadratic NURBS control mesh for the turbine model. The mesh was generated with the general-purpose NURBS mesh generation method [14,15] described in Sect. 1.4. Table 1 shows the number of control points and elements in different parts of the mesh. Figure 3 shows the four SIs of the mesh. Two of the SIs have an actual slip. The other two are just for mesh generation purpose and connect nonmatching meshes. They are used for independent meshing in the volute region of the computational domain. The STNMUM enables exact representation of the mesh rotation.

4.4 Computational conditions

In temporal representation of the mesh rotation, we use quadratic NURBS basis functions. There are 90 time steps per rotation, which is equivalent to a time-step size of 2.22×10^{-5} s. The number of nonlinear iterations per time step is 4, with 500, 500, 600 and 800 GMRES iterations

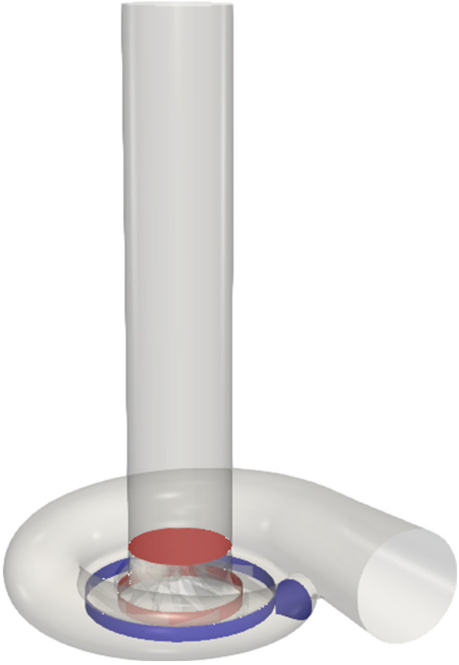

Fig. 3 Four SIs of the mesh. The red SIs have an actual slip, and the blue SIs are just for mesh generation purpose and connect nonmatching meshes. (Color figure online)

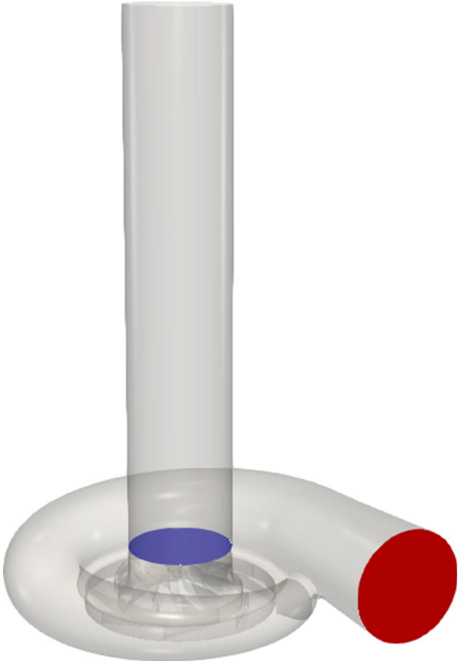

Fig. 4 Boundaries used in calculating the turbine efficiency: Γ_{INF} (red) and Γ_{OUTF} (blue). (Color figure online)

for the first, second, third and fourth nonlinear iterations, respectively. The first two nonlinear iterations are based on the ST-SUPS, and the last two the ST-VMS. We use the stabilization parameters given by Eqs. (32)–(36) and (38) and the scaling (see Appendix B.2) $D_\theta = 1$ and $\mathbf{D} = \mathbf{I}$. In the ST-SI, we use $C = 8$ (see "Appendix A").

4.5 Computations with the steady-inflow representation

The boundaries Γ_{INF} and Γ_{OUTF} are shown in Fig. 4. Figure 5 shows, at the 12 tip-speed ratios, the turbine efficiency averaged over one rotation, based on Eq. (9), where $t_3 = t_1$, $t_4 = t_2$, and $t_2 - t_1 = T_{\mathrm{TR}}$.

4.6 Computation with the time-dependent representation

For completeness, we include from [16] some of the results reported for the computation with the time-dependent representation of the intake/exhaust cycle. The symbol T will denote the duration of the intake/exhaust cycle, and $T = 6.0\times10^{-2}$ s. Figure 6 shows the velocity magnitude for the full cycle. Figure 7 shows the turbine efficiency $\eta_{\mathrm{IB}}(0,\ T,\ t,\ t)$.

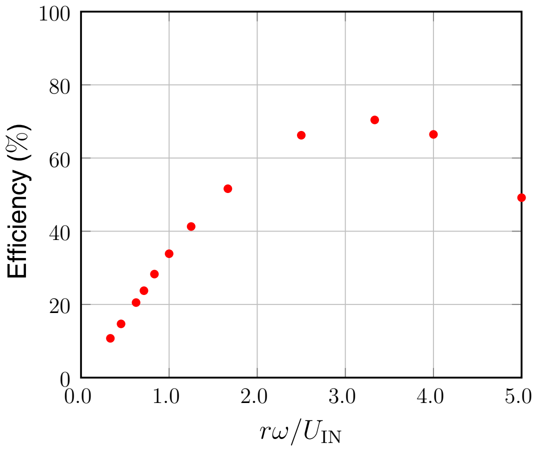

Fig. 5 Turbine efficiency from the computations with the steady-inflow representation

4.7 Discussion

Figures 8, 9 and 10 show, for different periods in the intake/exhaust cycle, turbine efficiencies from the computations with the time-dependent and steady-inflow representations. In calculating the turbine efficiency from the computations with the steady-inflow representation, we first calculate the tip-speed ratio for the moving-averaged inflow velocity coming from the time-dependent representation of the intake/exhaust cycle. The moving average is based on the expression

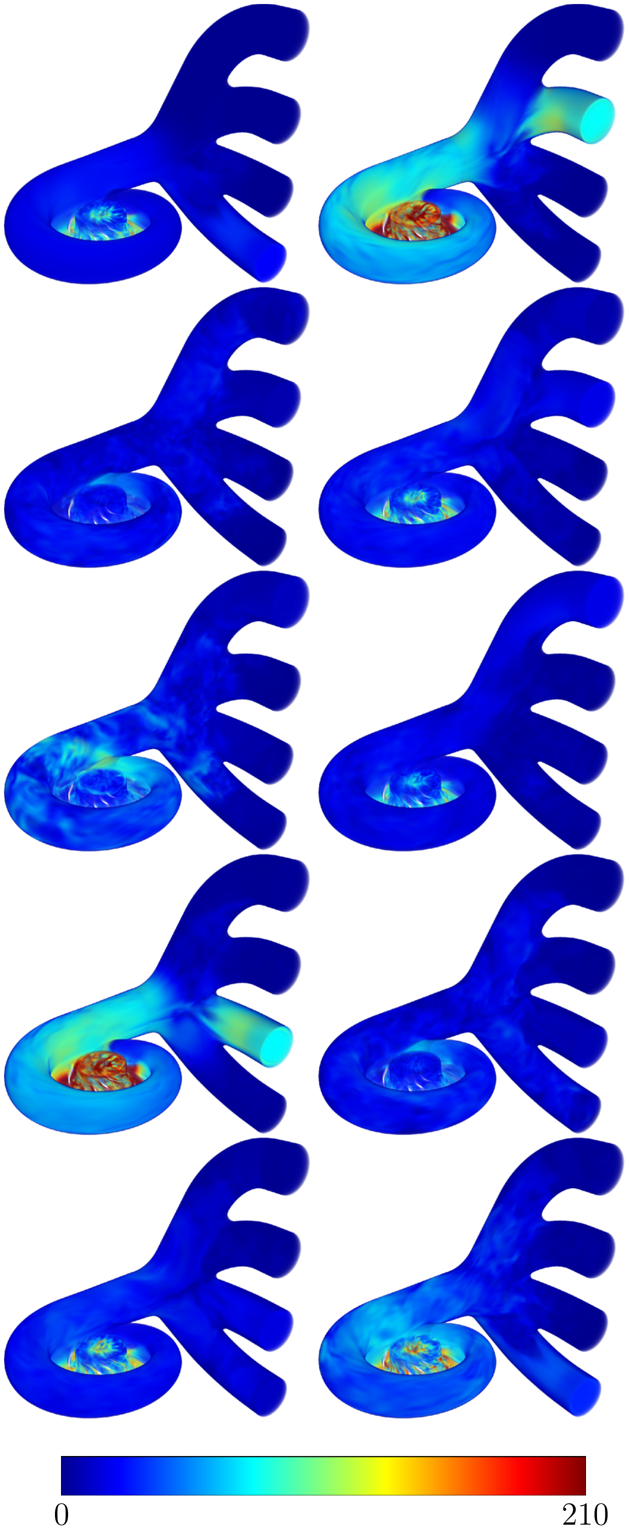

Fig. 6 Velocity magnitude (m/s) from the computation with the time-dependent representation [16], at $t/T = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ and 1.0 (left–right and top–bottom)

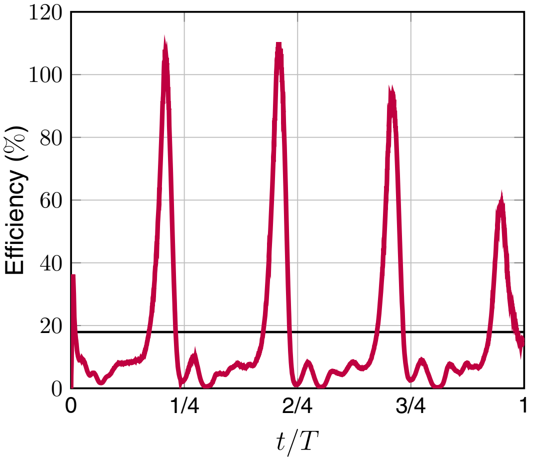

Fig. 7 Turbine efficiency $\eta_{\mathrm{IB}}(0,\ T,\ t,\ t)$ from the computation with the time-dependent representation [16]. The black line shows the average efficiency $\eta_{\mathrm{IB}}(0,\ T,\ 0,\ T)$

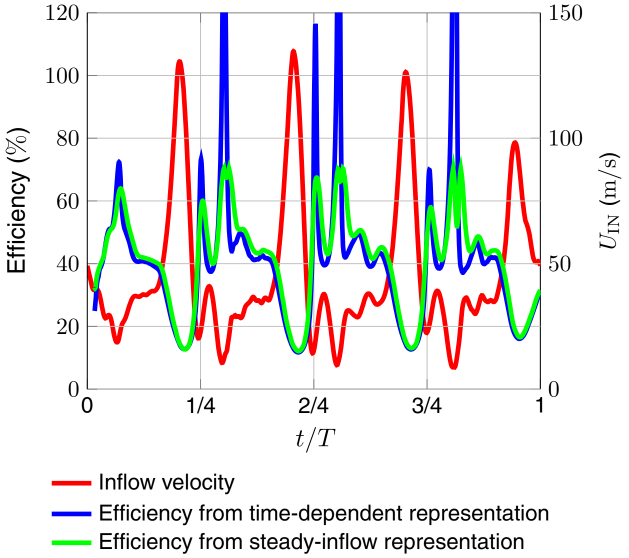

Fig. 8 Turbine efficiencies from the time-dependent and steady-inflow representations of the intake/exhaust cycle. Efficiency $\eta_{\mathrm{IB}}(t - T_{\mathrm{TR}}/2,\ t,\ t - T_{\mathrm{TR}}/2,\ t)$ and efficiency based on $(U_{\mathrm{IN}})_{\mathrm{IB}}\ (t - T_{\mathrm{TR}}/2,\ t)$

$$(U_{\mathrm{IN}})_{\mathrm{IB}}\ (t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} U_{\mathrm{IN}}(t)\mathrm{d}t. \tag{10}$$

The turbine efficiency corresponding to the time-dependent tip-speed ratio is obtained by interpolation from the steady-inflow turbine efficiencies displayed in Fig. 5. We note from Figs. 8, 9 and 10 that if the averaging period is too short or too long, the efficiencies from the steady-inflow and time-dependent representations do not match well. However, if we choose the right averaging period, which is in this case the turbine rotation period, the efficiencies match very well. We also note that the effect of the manifold

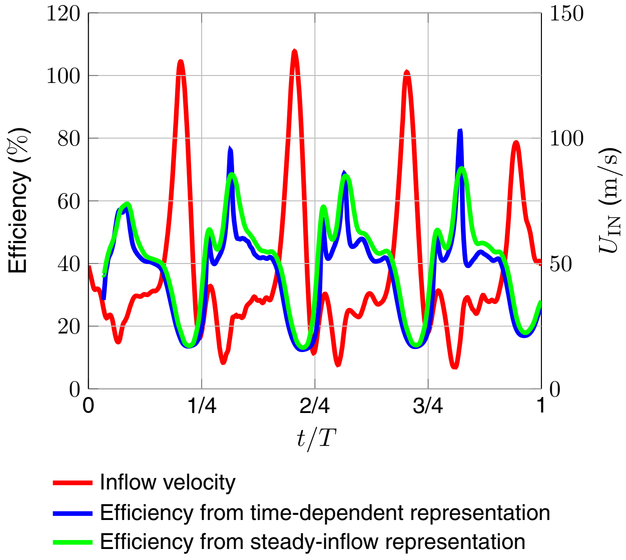

Fig. 9 Turbine efficiencies from the time-dependent and steady-inflow representations of the intake/exhaust cycle. Efficiency $\eta_{\mathrm{IB}}(t - T_{\mathrm{TR}},\ t,\ t - T_{\mathrm{TR}},\ t)$ and efficiency based on $(U_{\mathrm{IN}})_{\mathrm{IB}}\ (t - T_{\mathrm{TR}},\ t)$

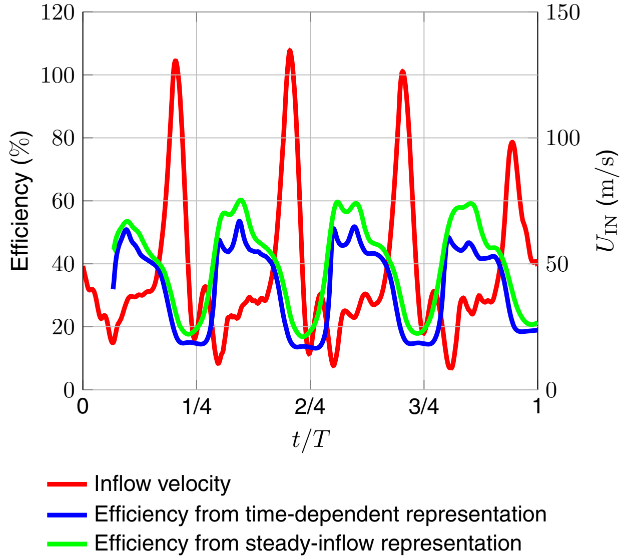

Fig. 10 Turbine efficiencies from the time-dependent and steady-inflow representations of the intake/exhaust cycle. Efficiency $\eta_{\mathrm{IB}}(t - 2T_{\mathrm{TR}},\ t,\ t - 2T_{\mathrm{TR}},\ t)$ and efficiency based on $(U_{\mathrm{IN}})_{\mathrm{IB}}\ (t - 2T_{\mathrm{TR}},\ t)$

and gas purifier is not significant, provided of course that we report the efficiency based on the same flow region, shown in Fig. 4. Overall, the turbine efficiency can be predicted from a set of computations with the steady-inflow representation of the intake/exhaust cycle, with little difference compared to the efficiency from the time-dependent representation.

5 Concluding remarks

One of the main computational challenges encountered in turbocharger-turbine flow analysis is the multiple time scales involved. The computational analysis needs to cover a full intake/exhaust cycle, which is much longer than the turbine rotation cycle because of high rotation speeds. Therefore computations with time-dependent representation of the intake/exhaust cycle require long-durations in the turbine time scale. In this article we focused on that challenge.

Many of the other computational challenges were addressed earlier by an integrated set of ST computational methods, and we used those methods in the computations here. The core computational method is the ST-VMS. The ST framework provides higher-order accuracy in general, the VMS feature of the ST-VMS addresses the computational challenges associated with the multiscale nature of the unsteady flow, and the moving-mesh feature of the ST framework enables high-resolution computation near the rotor surface. The ST-SI enables moving-mesh computation of the spinning rotor. The mesh covering the rotor spins with it, and the SI between the spinning mesh and the rest of the mesh accurately connects the two sides of the solution. The ST-IGA enables more accurate representation of the turbine geometry and increased accuracy in the flow solution. The STNMUM enables exact representation of the mesh rotation. A general-purpose NURBS mesh generation method makes it easier to deal with the complex geometries involved. An SI also provides mesh generation flexibility in a general context by accurately connecting the two sides of the solution computed over nonmatching meshes, and that is enabling the use of nonmatching NURBS meshes in the computations.

The challenge associated with the multiple time scales was addressed earlier by going through with the long-duration computation. As another way of addressing that challenge, we proposed here to calculate the turbine efficiency from computations with steady-inflow representation of the intake/exhaust cycle. For that, we first performed a set of steady-inflow computations at different flow rates. We then obtained the turbine efficiencies associated with those computations and tabulated them as a function of the tip-speed ratio. Next we calculated the tip-speed ratio for the moving-averaged inflow velocity coming from the time-dependent representation of the intake/exhaust cycle. We obtained the turbine efficiency corresponding to that time-dependent tip-speed ratio by interpolation from the tabulated values of the steady-inflow turbine efficiencies. The efficiencies obtained from the computations with time-dependent and steady-inflow representations of the intake/exhaust cycle compare well. This demonstrates that predicting the turbine performance from a set of steady-inflow computations is a good way of addressing the challenge associated with the multiple time scales.

Acknowledgements This work was supported in part by Grant-in-Aid for Challenging Exploratory Research 16K13779 from Japan Society for the Promotion of Science; Grant-in-Aid for Scientific Research (S) 26220002 from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT); Council for Science, Technology and Innovation (CSTI), Cross-Ministerial Strategic Innovation Promotion Program (SIP), "Innovative Combustion Technology" (Funding agency: JST); and Rice–Waseda research agreement. This work was also supported in part by Grant-in-Aid for Early-Career Scientists 19K20287 (first author) and ARO Grant W911NF-17-1-0046 and Top Global University Project of Waseda University (third author). The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper. The HPC resources provided by Nagoya University High Performance Computing Research Project for Joint Computational Science also contributed to obtaining the results reported in the paper.

Appendix A: ST-SI

In describing the ST-SI (see [20]), labels "Side A" and "Side B" will represent the two sides of the SI. The ST-SI version of the formulation given by Eq. (3) includes added boundary terms corresponding to the SI. The boundary terms for the two sides are first added separately, using the test functions $\mathbf{w}_{\mathrm{A}}^h$ and q_{A}^h and $\mathbf{w}_{\mathrm{B}}^h$ and q_{B}^h. Then, putting together the terms added to each side, the complete set of terms added becomes

$$
\begin{aligned}
&-\int_{(P_n)_{\mathrm{SI}}} \left(q_{\mathrm{B}}^h \mathbf{n}_{\mathrm{B}} - q_{\mathrm{A}}^h \mathbf{n}_{\mathrm{A}}\right) \cdot \frac{1}{2}\left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h\right) \mathrm{d}P \\
&\quad -\int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{B}}^h \cdot \frac{1}{2}\left(\left(\mathcal{F}_{\mathrm{B}}^h - \left|\mathcal{F}_{\mathrm{B}}^h\right|\right)\mathbf{u}_{\mathrm{B}}^h \right. \\
&\qquad\qquad \left. - \left(\mathcal{F}_{\mathrm{B}}^h - \left|\mathcal{F}_{\mathrm{B}}^h\right|\right)\mathbf{u}_{\mathrm{A}}^h\right) \mathrm{d}P \\
&\quad -\int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{A}}^h \cdot \frac{1}{2}\left(\left(\mathcal{F}_{\mathrm{A}}^h - \left|\mathcal{F}_{\mathrm{A}}^h\right|\right)\mathbf{u}_{\mathrm{A}}^h \right. \\
&\qquad\qquad \left. - \left(\mathcal{F}_{\mathrm{A}}^h - \left|\mathcal{F}_{\mathrm{A}}^h\right|\right)\mathbf{u}_{\mathrm{B}}^h\right) \mathrm{d}P \\
&\quad +\int_{(P_n)_{\mathrm{SI}}} \left(\mathbf{n}_{\mathrm{B}} \cdot \mathbf{w}_{\mathrm{B}}^h + \mathbf{n}_{\mathrm{A}} \cdot \mathbf{w}_{\mathrm{A}}^h\right) \frac{1}{2}\left(p_{\mathrm{B}}^h + p_{\mathrm{A}}^h\right) \mathrm{d}P \\
&\quad -\int_{(P_n)_{\mathrm{SI}}} \left(\mathbf{w}_{\mathrm{B}}^h - \mathbf{w}_{\mathrm{A}}^h\right) \cdot \left(\hat{\mathbf{n}}_{\mathrm{B}} \cdot \mu\left(\boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{B}}^h) + \boldsymbol{\varepsilon}(\mathbf{u}_{\mathrm{A}}^h)\right)\right) \mathrm{d}P \\
&\quad -\gamma \int_{(P_n)_{\mathrm{SI}}} \hat{\mathbf{n}}_{\mathrm{B}} \cdot \mu\left(\boldsymbol{\varepsilon}\left(\mathbf{w}_{\mathrm{B}}^h\right) + \boldsymbol{\varepsilon}\left(\mathbf{w}_{\mathrm{A}}^h\right)\right) \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h\right) \mathrm{d}P \\
&\quad +\int_{(P_n)_{\mathrm{SI}}} \frac{\mu C}{h}\left(\mathbf{w}_{\mathrm{B}}^h - \mathbf{w}_{\mathrm{A}}^h\right) \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{u}_{\mathrm{A}}^h\right) \mathrm{d}P,
\end{aligned}
\tag{11}
$$

where

$$
\mathcal{F}_{\mathrm{B}}^h = \mathbf{n}_{\mathrm{B}} \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{v}_{\mathrm{B}}^h\right), \tag{12}
$$

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

and h is the element length at the SI (see Appendix B.4). Here, $(P_n)_{\mathrm{SI}}$ is the SI in the ST domain, $\mathbf{v}$ is the mesh velocity, $\gamma = 1$, and C is a nondimensional constant. For explanation of the added SI terms, see [20,120].

On solid surfaces where we prefer weak enforcement of the Dirichlet conditions [20,42] for the fluid, we use the ST-SI version where the SI is between the fluid and solid domains. Using the terminology of Sect. 1.2, that is a fluid–solid SI. This version is obtained (see [20]) by starting with the terms added to Side B and replacing the Side A velocity with the velocity $\mathbf{g}^h$ coming from the solid domain. Then the SI terms added to Eq. (3) to represent the weakly-enforced Dirichlet conditions become

$$
\begin{aligned}
&-\int_{(P_n)_{\mathrm{SI}}} q_{\mathrm{B}}^h \mathbf{n}_{\mathrm{B}} \cdot \mathbf{u}_{\mathrm{B}}^h \mathrm{d}P - \int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{B}}^h \cdot \mathcal{F}_{\mathrm{B}}^h \mathbf{u}_{\mathrm{B}}^h \mathrm{d}P \\
&\quad +\int_{(P_n)_{\mathrm{SI}}} q_{\mathrm{B}}^h \mathbf{n}_{\mathrm{B}} \cdot \mathbf{g}^h \mathrm{d}P \\
&\quad +\int_{(P_n)_{\mathrm{SI}}} \rho \mathbf{w}_{\mathrm{B}}^h \cdot \frac{1}{2}\left(\left(\mathcal{F}_{\mathrm{B}}^h + \left|\mathcal{F}_{\mathrm{B}}^h\right|\right)\mathbf{u}_{\mathrm{B}}^h \right. \\
&\qquad\qquad \left. + \left(\mathcal{F}_{\mathrm{B}}^h - \left|\mathcal{F}_{\mathrm{B}}^h\right|\right)\mathbf{g}^h\right) \mathrm{d}P \\
&\quad -\int_{(P_n)_{\mathrm{SI}}} \mathbf{w}_{\mathrm{B}}^h \cdot \left(\mathbf{n}_{\mathrm{B}} \cdot \boldsymbol{\sigma}_{\mathrm{B}}^h\right) \mathrm{d}P \\
&\quad -\gamma \int_{(P_n)_{\mathrm{SI}}} \mathbf{n}_{\mathrm{B}} \cdot 2\mu\boldsymbol{\varepsilon}\left(\mathbf{w}_{\mathrm{B}}^h\right) \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{g}^h\right) \mathrm{d}P \\
&\quad +\int_{(P_n)_{\mathrm{SI}}} \frac{\mu C}{h_{\mathrm{B}}} \mathbf{w}_{\mathrm{B}}^h \cdot \left(\mathbf{u}_{\mathrm{B}}^h - \mathbf{g}^h\right) \mathrm{d}P.
\end{aligned}
\tag{14}
$$

Appendix B: Stabilization parameters

We include from [141] the element metric tensor in space and in the ST framework. These are used in Appendices B.3 and B.4 in calculation of the stabilization parameters and element lengths.

Appendix B.1: Element metric tensor in space

Components of the Jacobian matrix $\mathbf{Q}$ are written as

$$
Q_{ij} = \frac{\partial x_i}{\partial \xi_j}, \tag{15}
$$

where ξ_j is the parametric coordinate in jth direction. We first scale it with a matrix $\mathbf{D}$ to take into account the polynomial order or other factors such as the dimensions of the element domain in the parametric space:

$$\hat{\mathbf{Q}} = \mathbf{Q}\mathbf{D}^{-1}. \tag{16}$$

Remark 2 In the case of finite elements with Lagrange polynomials of order p_i in ith direction, with a parametric space of $-1 \leq \xi_i \leq 1$, the scaling matrix can be chosen as

$$\mathbf{D} = \begin{bmatrix} p_1 & 0 & 0 \\ 0 & p_2 & 0 \\ 0 & 0 & p_3 \end{bmatrix}. \tag{17}$$

With $\hat{\mathbf{Q}}$ and the direction $\mathbf{r}$ to which we want to associate the length, we define the element length (see [141]) as

$$h_{\mathrm{RQD}} = 2\left(\mathbf{rr} : \mathbf{G}\right)^{-\frac{1}{2}}, \tag{18}$$

where

$$\mathbf{G} = \hat{\mathbf{Q}}^{-T}\hat{\mathbf{Q}}^{-1}. \tag{19}$$

Remark 3 From this derivation, what we get with $\mathbf{D} = \mathbf{I}$ has been used in many methods for calculating the stabilization parameters (see, for example, [37]). In those methods, a scaling factor taking the polynomial order into account is applied to the element length, and here we do the scaling in the parametric space, for each of the parametric directions.

Sweeping over all the directions represented by $\mathbf{r}$, we obtain the minimum and maximum element lengths:

$$h_{\mathrm{MIN}} \equiv 2 \min_{\mathbf{r}} \left(\left(\mathbf{rr} : \mathbf{G}\right)^{-\frac{1}{2}} \right), \tag{20}$$

$$h_{\mathrm{MAX}} \equiv 2 \max_{\mathbf{r}} \left(\left(\mathbf{rr} : \mathbf{G}\right)^{-\frac{1}{2}} \right). \tag{21}$$

They are equivalent to

$$h_{\mathrm{MIN}} = 2 \left(\max_{\mathbf{r}} \left(\mathbf{rr} : \mathbf{G}\right) \right)^{-\frac{1}{2}}, \tag{22}$$

$$= 2 \left(\lambda_{\max}\left(\mathbf{G}\right)\right)^{-\frac{1}{2}}, \tag{23}$$

and

$$h_{\mathrm{MAX}} = 2 \left(\min_{\mathbf{r}} \left(\mathbf{rr} : \mathbf{G}\right) \right)^{-\frac{1}{2}}, \tag{24}$$

$$= 2 \left(\lambda_{\min}\left(\mathbf{G}\right)\right)^{-\frac{1}{2}}, \tag{25}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues of the argument matrix.

Remark 4 In the implementation, we take measures to keep the calculated element length between h_{MIN} and h_{MAX}.

Appendix B.2: Element metric tensor in the ST framework

The ST Jacobian matrix is

$$\mathbf{Q}^{\mathrm{ST}} = \begin{bmatrix} \frac{\partial t}{\partial \theta} & \frac{\partial t}{\partial \boldsymbol{\xi}} \\ \frac{\partial \mathbf{x}}{\partial \theta} & \mathbf{Q} \end{bmatrix} \tag{26}$$

$$= \begin{bmatrix} \frac{\partial t}{\partial \theta} & \frac{\partial t}{\partial \boldsymbol{\xi}} \\ \mathbf{v}\frac{\partial t}{\partial \theta} & \mathbf{Q} \end{bmatrix}, \tag{27}$$

where θ is the parametric coordinate in time, and the mesh velocity $\mathbf{v}$ is

$$\mathbf{v} = \left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\boldsymbol{\xi}}. \tag{28}$$

The ST scaling matrix is given as

$$\mathbf{D}^{\mathrm{ST}} = \begin{bmatrix} D_\theta & \mathbf{0}^T \\ \mathbf{0} & \mathbf{D} \end{bmatrix}, \tag{29}$$

and the scaling becomes

$$\hat{\mathbf{Q}}^{\mathrm{ST}} = \mathbf{Q}^{\mathrm{ST}} \left(\mathbf{D}^{\mathrm{ST}}\right)^{-1}. \tag{30}$$

The ST metric tensor is defined as

$$\mathbf{G}^{\mathrm{ST}} = \left(\hat{\mathbf{Q}}^{\mathrm{ST}}\right)^{-T} \left(\hat{\mathbf{Q}}^{\mathrm{ST}}\right)^{-1}. \tag{31}$$

Appendix B.3: ST-VMS

There are various ways of defining the stabilization parameters τ_{SUPS} and ν_{LSIC}. Here, τ_{SUPS} is mostly from [141]:

$$\tau_{\mathrm{SUPS}} = \left(\tau_{\mathrm{SUGN12}}^{-2} + \tau_{\mathrm{SUGN3}}^{-2} + \tau_{\mathrm{SUGN4}}^{-2}\right)^{-\frac{1}{2}}. \tag{32}$$

The first component is given as

$$\tau_{\mathrm{SUGN12}}^{-2} = \begin{bmatrix} 1 \\ \mathbf{u} \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{u} \end{bmatrix} : \mathbf{G}^{\mathrm{ST}}. \tag{33}$$

The second component is defined as

$$\tau_{\mathrm{SUGN3}}^{-1} = \nu \mathbf{rr} : \mathbf{G}, \tag{34}$$

where $\mathbf{r}$ is the solution direction:

$$\mathbf{r} = \frac{\boldsymbol{\nabla} \left\|\mathbf{u}\right\|}{\left\|\boldsymbol{\nabla} \left\|\mathbf{u}\right\|\right\|}. \tag{35}$$

The third component, originating from [19], is defined as

$$\tau_{\mathrm{SUGN4}} = \left\| \nabla \mathbf{u}^h \right\|_F^{-1} \tag{36}$$

or

$$\tau_{\mathrm{SUGN4}} = \left\| \boldsymbol{\varepsilon}(\mathbf{u}^h) \right\|_F^{-1} . \tag{37}$$

Here $\|\cdot\|_F$ represents the Frobenius norm. The stabilization parameter ν_{LSIC} is from [25]:

$$\nu_{\mathrm{LSIC}} = \frac{h_{\mathrm{LSIC}}^2}{\tau_{\mathrm{SUPS}}}, \tag{38}$$

where h_{LSIC} is set equal to the minimum element length h_{MIN} (see 1). For more ways of calculating the stabilization parameters in flow computations, see [1,2,19,20,25,27,28,142–161].

Appendix B.4: ST-SI

The element length used in the ST-SI is given as

$$h = \left(\frac{h_{\mathrm{B}}^{-1} + h_{\mathrm{A}}^{-1}}{2} \right)^{-1}, \tag{39}$$

$$h_{\mathrm{B}} = 2 \left(\mathbf{n}_{\mathrm{B}} \mathbf{n}_{\mathrm{B}} : \mathbf{G} \right)^{-\frac{1}{2}} \quad \text{(for Side B)}, \tag{40}$$

$$h_{\mathrm{A}} = 2 \left(\mathbf{n}_{\mathrm{A}} \mathbf{n}_{\mathrm{A}} : \mathbf{G} \right)^{-\frac{1}{2}} \quad \text{(for Side A)}, \tag{41}$$

$$\hat{\mathbf{n}}_{\mathrm{B}} = \frac{\mathbf{n}_{\mathrm{B}} - \mathbf{n}_{\mathrm{A}}}{\|\mathbf{n}_{\mathrm{B}} - \mathbf{n}_{\mathrm{A}}\|}. \tag{42}$$

These were introduced in [16].

References

1. 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
2. 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
3. Cardillo L, Corsini A, Delibra G, Rispoli F, Tezduyar TE (2016) Flow analysis of a wave-energy air turbine with the SUPG/PSPG method and DCDD 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, pp 39–53. ISBN 978-3-319-40825-5. https://doi.org/10.1007/978-3-319-40827-9_4
4. 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
5. Castorrini A, Corsini A, Rispoli F, Venturini P, Takizawa K, Tezduyar TE (2016) SUPG/PSPG computational analysis of rain erosion in wind-turbine blades. 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, pp 77–96. ISBN 978-3-319-40825-5. https://doi.org/10.1007/978-3-319-40827-9_7
6. Castorrini A, Corsini A, Rispoli F, Venturini P, Takizawa K, Tezduyar TE (2016) Computational analysis of wind-turbine blade rain erosion. Comput Fluids 141:175–183. https://doi.org/10.1016/j.compfluid.2016.08.013
7. Castorrini 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. https://doi.org/10.1142/S0218202519410057
8. 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. https://doi.org/10.1007/s00466-019-01697-0
9. 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
10. 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
11. 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, pp 253–336. ISBN 978-3-319-96468-3. https://doi.org/10.1007/978-3-319-96469-0_7
12. Korobenko A, Bazilevs Y, Takizawa K, Tezduyar TE (2018) Computer modeling of wind turbines: 1. ALE-VMS and ST-VMS aerodynamic and FSI analysis. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-018-9292-1
13. 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
14. 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
15. 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, pp 399–434. ISBN 978-3-319-96468-3. https://doi.org/10.1007/978-3-319-96469-0_10
16. 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

17. 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
18. 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
19. 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
20. 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
21. 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
22. 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
23. 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
24. 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
25. 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
26. 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
27. 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
28. 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
29. 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
30. 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
31. Hughes TJR, Oberai AA, Mazzei L (2001) Large eddy simulation of turbulent channel flows by the variational multiscale method. Phys Fluids 13:1784–1799
32. 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
33. 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
34. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y (2008) Isogeometric fluid–structure interaction: theory, algorithms, and computations. Comput Mech 43:3–37
35. 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
36. 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(supp02):1230002. https://doi.org/10.1142/S0218202512300025
37. Bazilevs Y, Takizawa K, Tezduyar TE (2013) Computational fluid–structure interaction: methods and applications, Wiley. ISBN 978-0470978771
38. 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
39. 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
40. Bazilevs Y, Takizawa K, Tezduyar TE (2019) Computational analysis methods for complex unsteady flow problems. Math Models Methods Appl Sci. https://doi.org/10.1142/S0218202519020020
41. 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
42. Bazilevs Y, Hughes TJR (2007) Weak imposition of Dirichlet boundary conditions in fluid mechanics. Comput Fluids 36:12–26
43. 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
44. 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
45. Bazilevs Y, Hughes TJR (2008) NURBS-based isogeometric analysis for the computation of flows about rotating components. Comput Mech 43:143–150
46. Hsu M-C, Bazilevs Y (2012) Fluid–structure interaction modeling of wind turbines: simulating the full machine. Comput Mech 50:821–833
47. Moghadam ME, Bazilevs Y, Hsia T-Y, Vignon-Clementel IE, Marsden AL, M. of Congenital Hearts Alliance (MOCHA) (2011) A comparison of outlet boundary treatments for prevention of backflow divergence with relevance to blood flow simulations. Comput Mech 48:277–291. https://doi.org/10.1007/s00466-011-0599-0
48. 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
49. Hsu M-C, Akkerman I, Bazilevs Y (2011) High-performance computing of wind turbine aerodynamics using isogeometric analysis. Comput Fluids 49:93–100
50. 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
51. 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
52. 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

53. 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
54. 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
55. 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
56. 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
57. 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
58. 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
59. 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
60. 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
61. 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
62. 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
63. 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
64. 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
65. Hsu M-C, Bazilevs Y (2011) Blood vessel tissue prestress modeling for vascular fluid-structure interaction simulations. Finite Elem Anal Des 47:593–599
66. 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
67. 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
68. 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
69. 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
70. 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
71. 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
72. 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
73. 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
74. 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
75. 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
76. 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
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 (2018) Computational and experimental investigation of free vibration and flutter of bridge decks. Comput Mech. https://doi.org/10.1007/s00466-018-1587-4
80. Helgedagsrud TA, Bazilevs Y, Korobenko A, Mathisen KM, Oiseth OA (2018) Using ALE-VMS to compute aerodynamic derivatives of bridge sections. Comput Fluids. https://doi.org/10.1016/j.compfluid.2018.04.037
81. Helgedagsrud TA, Akkerman I, Bazilevs Y, Mathisen KM, Oiseth OA Isogeometric modeling and experimental investigation of moving-domain bridge aerodynamics. ASCE Journal of Engineering Mechanics, Accepted for publication
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. 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
85. 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
86. van Opstal TM, Yan J, Coley C, Evans JA, Kvamsdal T, Bazilevs Y (2017) Isogeometric divergence-conforming variational mul-

tiscale formulation of incompressible turbulent flows. Comput Methods Appl Mech Eng 316:859–879. https://doi.org/10.1016/j.cma.2016.10.015

87. 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
88. 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
89. 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
90. 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
91. 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
92. 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
93. 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
94. 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
95. 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
96. 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
97. 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
98. 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
99. 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
100. 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
101. 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
102. 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
103. 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
104. 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
105. 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
106. 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
107. 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
108. 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
109. 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, pp 379–386. ISBN 978-3-319-40825-5. https://doi.org/10.1007/978-3-319-40827-9_29
110. 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, pp 29–89. ISBN 978-3-319-96468-3. https://doi.org/10.1007/978-3-319-96469-0_2
111. 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
112. 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
113. 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, pp 159–178. ISBN 978-3-319-40825-5. https://doi.org/10.1007/978-3-319-40827-9_13
114. 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, pp 77–99. ISBN 978-3-319-59547-4. https://doi.org/10.1007/978-3-319-59548-1_6

115. 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
116. 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
117. 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
118. 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 Seri Earth Environ Sci 240:062014. https://doi.org/10.1088/1755-1315/240/6/062014
119. 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. https://doi.org/10.1142/S021820251941001X
120. 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
121. 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, pp 337–376. ISBN 978-3-319-96468-3. https://doi.org/10.1007/978-3-319-96469-0_8
122. 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
123. Kuraishi T, Takizawa K, Tezduyar TE (2019) Space-Time isogeometric flow analysis with built-in Reynolds-equation limit. Math Models Methods Appl Sci. https://doi.org/10.1142/S0218202519410021
124. Takizawa K, Tezduyar TE, Terahara T (2016) Ram-air parachute structural and fluid mechanics computations with the space-time isogeometric analysis (ST-IGA). Comput Fluids 141:191–200. https://doi.org/10.1016/j.compfluid.2016.05.027
125. Takizawa K, Tezduyar TE, Kanai T (2017) Porosity models and computational methods for compressible-flow aerodynamics of parachutes with geometric porosity. Math Models Methods Appl Sci 27:771–806. https://doi.org/10.1142/S0218202517500166
126. Kanai T, Takizawa K, Tezduyar TE, Tanaka T, Hartmann A (2019) Compressible-flow geometric-porosity modeling and spacecraft parachute computation with isogeometric discretization. Comput Mech 63:301–321. https://doi.org/10.1007/s00466-018-1595-4
127. Tezduyar TE, Aliabadi SK, Behr M, Mittal S (1994) Massively parallel finite element simulation of compressible and incompressible flows. Comput Methods Appl Mech Eng 119:157–177. https://doi.org/10.1016/0045-7825(94)00082-4
128. 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
129. 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, CIMNE, Barcelona, Spain
130. 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 Biomedl Eng 57:601–629. https://doi.org/10.1002/fld.1633
131. 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 Biomedl Eng 26:101–116. https://doi.org/10.1002/cnm.1241
132. 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 Biomedl Eng 65:308–323. https://doi.org/10.1002/fld.2360
133. Tezduyar TE, Takizawa K, Brummer T, Chen PR (2011) Space-time fluid-structure interaction modeling of patient-specific cerebral aneurysms. Int J Numer Methods Biomedl Eng 27:1665–1710. https://doi.org/10.1002/cnm.1433
134. 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
135. 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
136. 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, pp 101–122. ISBN 978-3-319-59547-4. https://doi.org/10.1007/978-3-319-59548-1_7
137. 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
138. 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
139. Sasaki T, Takizawa K, Tezduyar TE (2019) Medical-image-based aorta modeling with zero-stress-state estimation. Comput Mech.https://doi.org/10.1007/s00466-019-01669-4
140. 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
141. 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
142. Tezduyar TE, Ganjoo DK (1986) Petrov–Galerkin formulations with weighting functions dependent upon spatial and temporal discretization: applications to transient convection-diffusion problems. Comput Methods Appl Mech Eng 59:49–71. https://doi.org/10.1016/0045-7825(86)90023-X
143. Le Beau GJ, Ray SE, Aliabadi SK, Tezduyar TE (1993) SUPG finite element computation of compressible flows with the entropy and conservation variables formulations. Comput Methods Appl Mech Eng 104:397–422. https://doi.org/10.1016/0045-7825(93)90033-T

144. Tezduyar TE (2007) Finite elements in fluids: stabilized formulations and moving boundaries and interfaces. Comput Fluids 36:191–206. https://doi.org/10.1016/j.compfluid.2005.02.011

145. Tezduyar TE, Senga M (2006) Stabilization and shock-capturing parameters in SUPG formulation of compressible flows. Comput Methods Appl Mech Eng 195:1621–1632. https://doi.org/10.1016/j.cma.2005.05.032

146. Tezduyar TE, Senga M (2007) SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing. Comput Fluids 36:147–159. https://doi.org/10.1016/j.compfluid.2005.07.009

147. Tezduyar TE, Senga M, Vicker D (2006) Computation of inviscid supersonic flows around cylinders and spheres with the SUPG formulation and YZβ shock-capturing. Comput Mech 38:469–481. https://doi.org/10.1007/s00466-005-0025-6

148. Tezduyar TE, Sathe S (2006) Enhanced-discretization selective stabilization procedure (EDSSP). Comput Mech 38:456–468. https://doi.org/10.1007/s00466-006-0056-7

149. Corsini A, Rispoli F, Santoriello A, Tezduyar TE (2006) Improved discontinuity-capturing finite element techniques for reaction effects in turbulence computation. Comput Mech 38:356–364. https://doi.org/10.1007/s00466-006-0045-x

150. Rispoli F, Corsini A, Tezduyar TE (2007) Finite element computation of turbulent flows with the discontinuity-capturing directional dissipation (DCDD). Comput Fluids 36:121–126. https://doi.org/10.1016/j.compfluid.2005.07.004

151. Tezduyar TE, Ramakrishnan S, Sathe S (2008) Stabilized formulations for incompressible flows with thermal coupling. Int J Numer Methods Fluids 57:1189–1209. https://doi.org/10.1002/fld.1743

152. Rispoli F, Saavedra R, Corsini A, Tezduyar TE (2007) Computation of inviscid compressible flows with the V-SGS stabilization and YZβ shock-capturing. Int J Numer Methods Fluids 54:695–706. https://doi.org/10.1002/fld.1447

153. Bazilevs Y, Calo VM, Tezduyar TE, Hughes TJR (2007) YZβ discontinuity-capturing for advection-dominated processes with application to arterial drug delivery. Int J Numer Methods Fluids 54:593–608. https://doi.org/10.1002/fld.1484

154. 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

155. Rispoli F, Saavedra R, Menichini F, Tezduyar TE (2009) Computation of inviscid supersonic flows around cylinders and spheres with the V-SGS stabilization and YZβ shock-capturing. J Appl Mech 76:021209. https://doi.org/10.1115/1.3057496

156. 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

157. 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

158. 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

159. 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

160. 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

161. 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 YZβ shock-capturing. Comput Mech 55:1201–1209. https://doi.org/10.1007/s00466-015-1160-3

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