Element length calculation in B-spline meshes for complex geometries

Yuto Otoguro¹ · Kenji Takizawa¹ · Tayfun E. Tezduyar²,³

Received: 6 October 2019 / Accepted: 15 December 2019 / Published online: 3 January 2020
© The Author(s) 2020

Abstract

Variational multiscale methods, and their precursors, stabilized methods, have been playing a core-method role in semi-discrete and space–time (ST) flow computations for decades. These methods are sometimes supplemented with discontinuity-capturing (DC) methods. The stabilization and DC parameters embedded in most of these methods play a significant role. Various well-performing stabilization and DC parameters have been introduced in both the semi-discrete and ST contexts. The parameters almost always involve some element length expressions, most of the time in specific directions, such as the direction of the flow or solution gradient. Until recently, stabilization and DC parameters originally intended for finite element discretization were being used also for isogeometric discretization. Recently, element lengths and stabilization and DC parameters targeting isogeometric discretization were introduced for ST and semi-discrete computations, and these expressions are also applicable to finite element discretization. The key stages of deriving the direction-dependent element length expression were mapping the direction vector from the physical (ST or space-only) element to the parent element in the parametric space, accounting for the discretization spacing along each of the parametric coordinates, and mapping what has been obtained back to the physical element. Targeting B-spline meshes for complex geometries, we introduce here new element length expressions, which are outcome of a clear and convincing derivation and more suitable for element-level evaluation. The new expressions are based on a preferred parametric space and a transformation tensor that represents the relationship between the integration and preferred parametric spaces. The test computations we present for advection-dominated cases, including 2D computations with complex meshes, show that the proposed element length expressions result in good solution profiles.

Keywords

Space–time computational methods · Stabilization parameter · Discontinuity-capturing parameter · Element length · Advection–diffusion equation · Isogeometric discretization · Preferred parametric space

1 Introduction

Targeting complex geometries, in this article we introduce directional element length expressions for B-spline meshes used in flow computations with the stabilized and variational multiscale (VMS) methods, discontinuity-capturing (DC) methods, and other special methods that require a directional element length.

1.1 Stabilized and VMS methods

Stabilized and VMS methods have for decades been playing a core-method role in flow analysis with semi-discrete and space–time (ST) computational methods. The incompressible-flow Streamline-Upwind/Petrov-Galerkin (SUPG) [1,2] and compressible-flow SUPG [3–5] methods are two of the earliest and most widely used stabilized methods. The incompressible-flow Pressure-Stabilizing/Petrov-Galerkin (PSPG) method [6,7], with its Stokes-flow version introduced in [8], is also among the earliest and most widely used. These methods bring numerical stability in computation of flow problems at high Reynolds or Mach numbers and when using equal-order basis functions for velocity and pressure in incompressible flows. Because the methods are residual-
based, the stabilization is achieved without loss of accuracy. The residual-based VMS (RBVMS) [9–12], which is now also widely used, subsumes its precursor SUPG/PSPG.

1.2 DC methods

When the flow field has a discontinuity, stabilized and VMS methods are often supplemented with a DC term, which is also called “shock-capturing term” in the context of compressible flows. Supplemented the SUPG with a DC term started as early as 1986 [13,14]. The DC term played a significant role in the evolution of the compressible-flow SUPG, which was originally introduced in 1982 in the context of conservation variables. The 1982 method is now called “(SUPG)82.” At its inception, (SUPG)82 was not used with any DC term, and the need for additional stabilization at the shocks was obvious from the test computations. Then (SUPG)82 was recast in entropy variables, but also supplemented with a DC term [15]. This, as expected, resulted in better shock profiles. In a 1991 ASME paper [16], (SUPG)82 was supplemented with a very similar DC term. All the test computations presented in [16,17], which were among the most popular test computations at that time, showed that, with the DC term, (SUPG)82 was very comparable in accuracy to (SUPG)82 recast in entropy variables. The stabilized and DC methods introduced in [14] for the advection–diffusion–reaction equation included safeguards to prevent “compounding,” i.e., augmentation of the SUPG effect by the DC effect when the advection and discontinuity directions are not orthogonal.

1.3 Stabilized and VMS ST computational methods

Stabilized and VMS methods have been an important component of the ST computational methods, starting with the original Deforming-Spatial-Domain/Stabilized ST (DSD/SST) method [6,18,19] in 1990. The DSD/SST [6,20,21] was introduced for computation of flows with moving boundaries and interfaces (MBI), including fluid–structure interaction (FSI). In flow computations with MBI, 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. Because the stabilization components of the original DSD/SST are the SUPG and PSPG stabilizations, it is now also called “ST-SUPS.” The ST-VMS [22–24] is the VMS version of the DSD/SST. The VMS components of the ST-VMS are from the RBVMS. The ST-VMS, which subsumes its precursor ST-SUPS, 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 [22,23]), are desirable also in computations without MBI.

As a moving-mesh method, the DSD/SST is an alternative to the Arbitrary Lagrangian–Eulerian (ALE) method, which is older (see, for example, [25]) and more commonly used. The ALE-VMS method [26–32] is the VMS version of the ALE. It succeeded the ST-SUPS and ALE-SUPS [33] 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 [34–36] and “sliding interfaces” [37,38]. 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 [33], wind-turbine aerodynamics and FSI [39–49], more specifically, vertical-axis wind turbines [48–51], floating wind turbines [52], wind turbines in atmospheric boundary layers [47–49,53], and fatigue damage in wind-turbine blades [54], patient-specific cardiovascular fluid mechanics and FSI [26,55–60], biomedical-device FSI [61–66], ship hydrodynamics with free-surface flow and fluid–object interaction [67,68], hydrodynamics and FSI of a hydraulic arresting gear [69,70], hydrodynamics of tidal-stream turbines with free-surface flow [71], passive-morphing FSI in turbomachinery [72], bioinspired FSI for marine propulsion [73,74], bridge aerodynamics and fluid–object interaction [75–77], and mixed ALE-VMS/Immersogeometric computations [64–66,78,79] in the framework of the Fluid–Solid Interface-Tracking/Interface-Capturing Technique [80]. Recent advances in stabilized and multiscale methods may be found for stratified incompressible flows in [81], for divergence-conforming discretizations of incompressible flows in [82], and for compressible flows with emphasis on gas-turbine modeling in [83].

The ST-SUPS and ST-VMS have also been applied to many classes of FSI, MBI and fluid mechanics problems (see [84] for a comprehensive summary). The classes of problems include spacecraft parachute analysis for the landing-stage parachutes [29,85–88], cover-separation parachutes [89] and the drogue parachutes [90–92], wind-turbine aerodynamics for horizontal-axis wind-turbine rotors [29,39,93,94], full horizontal-axis wind-turbines [45,95–97] and vertical-axis wind-turbines [48,49,98], flapping-wing aerodynamics for an actual locust [29,99–101], bioinspired MAVs [96,97,102,103] and wing-clapping [104,105], blood flow analysis of cerebral aneurysms [96,106], stent-blocked aneurysms [106–108], aortas [109–113], heart valves [97,104,111,113–117] and coronary arteries in motion [118], spacecraft aerodynamics [89,119], thermo-fluid analysis of ground vehicles and their tires [24,115], thermo-fluid analysis of disk brakes [120], flow-driven string dynamics in turbomachinery [121–123], flow analysis of turbocharger turbines [124–128], flow around tires with road contact and deformation [115,129–
132], fluid films [132,133], ram-air parachutes [134], and compressible-flow spacecraft parachute aerodynamics [135, 136].

1.4 ST Slip Interface method

The ST Slip Interface (ST-SI) method was introduced in [98], 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 [37,38]. 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 [98]. 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 [120] 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 [48,49,98], thermo-fluid analysis of disk brakes [120], flow-driven string dynamics in turbomachinery [121–123], flow analysis of turbocharger turbines [124–128], flow around tires with road contact and deformation [115,129–132], fluid films [132,133], aerodynamic analysis of ram-air parachutes [134], and flow analysis of heart valves [111,113,116,117].

In the ST-SI version presented in [98] 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 [134]. The compressible-flow ST-SI methods were introduced in [135], 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 [135]. These, together with the compressible-flow ST SUPG method [137], 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 [135,136].

1.5 ST Isogeometric Analysis

The success with Isogeometric Analysis (IGA) basis functions in space [26,37,55,138] motivated the integration of the ST methods with isogeometric discretization, which we broadly call “ST-IGA.” The ST-IGA was introduced in [22]. Computations with the ST-VMS and ST-IGA were first reported in [22] 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 [22] 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 [22,23] and demonstrated in [99,100,102]. 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 (STNNUM) [99,100,102], with the name coined in [95]. The STNNUM 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” [22,23,29,99] 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 [24,115,120–123,139]. The STNNUM 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 [29,99–101], bioinspired MAVs [96,97,102,103] and wing-clapping [104,105], separation aerodynamics of spacecraft [89], aerodynamics of horizontal-axis [45,95–97] and vertical-axis [48,49,98] wind-turbines, thermo-fluid analysis of ground vehicles and their tires [24,115], thermo-fluid analysis of disk brakes [120], flow-driven string dynamics in turbomachinery [121–123], flow analysis of turbocharger turbines [124–128], and flow analysis of coronary arteries in motion [118].

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 [124–128], flow-
driven string dynamics in turbomachinery [122,123], ram-air parachutes [134], spacecraft parachutes [136], aortas [111–113], heart valves [111,113,116,117], coronary arteries in motion [118], tires with road contact and deformation [130–132], and fluid films [132,133]. Using IGA basis functions in space is now a key part of some of the newest arterial zero-stress-state estimation methods [113,140–145] and related shell analysis [146].

1.6 Stabilization parameters and element lengths

In all the semi-discrete and ST stabilized and VMS methods discussed in the earlier subsections, an embedded stabilization parameter, known as “τ,” plays a significant role (see [29]). This parameter involves a measure of the local length scale (also known as “element length”) and other parameters such as the element Reynolds and Courant numbers. The interface terms in the ST-SI also involve element length, in the direction normal to the interface. Various element lengths and τs were proposed, starting with those in [1,2] and [3–5], followed by the ones introduced in [13,14]. In many cases, the element length was seen as an advection length scale, in the flow-velocity direction. The set of τs introduced in [3–5] in conjunction with (SUPG)82 is now called “τ82.” The τ definition introduced in [14], which is for the advective limit and is now called “τSUGN1)” (and the corresponding element length is now called “hUGN”), automatically yields lower values for higher-order finite element basis functions (see [147,148]). Later, other τ definitions that are applicable to higher-order elements were proposed in [149] in the context of advective-diffusive systems. The τ used in [16] with (SUPG)82 was a slightly modified version of τ82. Subsequent minor modifications of τ82 took into account the interaction between the DC and (SUPG)82 terms in a fashion similar to how it was done in [14] for the advection–diffusion–reaction equation. Until 2004, all these slightly modified versions of τ82 were always used with the same DC parameter, which was introduced in the 1991 ASME paper [16] and is now called “δ91.” This DC parameter was derived from the one given in [15] for the entropy variables.

Calculating the τs based on the element-level matrices and vectors was introduced in [150] in the context of the advection–diffusion equation and the Navier–Stokes equations of incompressible flows. These definitions are expressed in terms of the ratios of the norms of the matrices or vectors. They automatically take into account the local length scales, advection field and the element Reynolds number. The definitions based on the element-level vectors were shown [150,151] to address the difficulties reported at small time-step sizes. A second element length scale, in the solution-gradient direction and called “hRGN,” was introduced in 2001 [20,152]. Recognizing this as a diffusion length scale, a new stabilization parameter for the diffusive limit, “τSUGN3,” was introduced in [20,153], to be used together with τSUGN1 and “τSUGN2,” the parameters for the advective and transient limits. For the stabilized ST methods, “τSUGN12,” representing both the advective and transient limits, was also introduced in [20].

New ways of calculating the τ and DC parameter to be used with (SUPG)82 were introduced in 2004 [153–155]. The new τs, now categorized under the label “τ01,” have a matrix structure for viscous flows and reduce to a scalar for inviscid flows. The new DC parameters were of two types: one defined in a style the Discontinuity-Capturing Directional Dissipation [20,155,156] parameter was defined, and one that is now called “YZβ” DC parameter. The YZβ DC parameter is residual-based, and it is simpler than δ91. It has options for smoother or sharper computed shocks. A number of 2D and 3D test computations with YZβ DC were reported in [157–159]. These computations showed that in addition to being simpler than δ91, the YZβ DC parameter was superior in accuracy. The computations reported in [157–159] were based on the compressible-flow ST SUPG.

Some new options for the stabilization parameters used with the SUPS and VMS were proposed in [21,24,94,95,99]. These include a fourth τ component, “τSUGN4” [24], which was introduced for the VMS, considering one of the two extra stabilization terms the VMS has compared to the SUPS. They also include stabilization parameters [24] for the thermal-transport part of the VMS for the coupled incompressible-flow and thermal-transport equations.

Some of the stabilization parameters described in this subsection were also used in computations with other SUPG-like methods, such as the computations reported in [72,160–171].

1.7 Directional element lengths for isogeometric discretization

The stabilization and DC parameters and element lengths discussed in the previous subsection were all originally intended for finite element discretization but quite often used also for isogeometric discretization. The element lengths and stabilization and DC parameters introduced in [172] target isogeometric discretization but are also applicable to finite element discretization. They were introduced in the context of the advection–diffusion equation and the Navier–Stokes equations of incompressible flows. The direction-dependent element length expression was outcome of a conceptually simple derivation. The key components of the derivation are mapping the direction vector from the physical ST element to the parent ST element, accounting for the discretization spacing along each of the parametric coordinates, and mapping what has been obtained in the parent element back to the physical element. The test computations presented in [172] for pure-advection cases, including those with discontinuous solution, showed that the element lengths and stabilization
parameters proposed result in good solution profiles. The test computations also showed that the “UGN” parameters give reasonably good solutions even with NURBS basis functions.

1.8 Preferred parametric space and directional element lengths for simplex elements

In general, we decide what parametric space to use based on reasons like numerical integration efficiency or implementation convenience. Obviously, choices based on such reasons should not influence the method in substance. We require the element lengths, including the directional element lengths, to have node-numbering invariance for all element types, including simplex elements. The directional element length expression introduced in [173] meets that requirement. This is accomplished by using in the element length calculations for simplex elements a preferred parametric space instead of the standard integration parametric space. The element length expressions based on the two parametric spaces were evaluated in [173] in the context of simplex elements. It was shown that when the element length expression is based on the integration parametric space, the variation with the node numbering could be by a factor as high as 1.9 for 3D elements and 2.2 for ST elements. It was also shown that the element length expression based on the integration parametric space could overestimate the element length by a factor as high as 2.8 for 3D elements and 3.2 for ST elements.

1.9 Directional element lengths for B-spline elements

Targeting B-spline meshes for complex geometries, we introduce here new directional element length expressions, which are outcome of a clear and convincing derivation and more suitable for element-level evaluation. The new expressions are based on a preferred parametric space and a transformation tensor that represents the relationship between the integration and preferred parametric spaces.

1.10 Outline of the remaining sections

We provide the advection–diffusion equation in Sect. 2, and the ST method for the advection–diffusion equation in Sect. 3. The element length definitions are given, from [172], in Sect. 4, and the transformation tensor for B-spline elements are presented in Sect. 5. The test computations are presented in Sect. 6, and the concluding remarks are given in Sect. 7. The stabilization and DC parameters are given in “Appendix A”.

2 Advection–diffusion equation

Let \( \Omega_t \subset \mathbb{R}^{nd} \) be the spatial domain with boundary \( \Gamma_t \) at time \( t \in (0, T) \), where \( n_{el} \) is the number of space dimensions. The subscript \( t \) indicates the time-dependence of the domain. The advection–diffusion equation is written on \( \Omega_t \) and \( \forall t \in (0, T) \) as

\[
\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi - \nabla \cdot (\nu \nabla \phi) = 0, \tag{1}
\]

where \( \phi \) represents the quantity being transported (e.g., temperature, concentration), \( \mathbf{u} \) is the flow velocity, and \( \nu \) is the diffusivity. The essential and natural boundary conditions associated with Eq. (1) are represented as \( \phi = g_\phi \) on \( (\Gamma_t \setminus \Gamma_p) \) and \( \mathbf{n} \cdot \nabla \phi = h_\phi \) on \( (\Gamma_t \setminus \Gamma_p) \), where \( g_\phi \) and \( h_\phi \) are given functions. A function \( \phi_0(\mathbf{x}) \) is specified as the initial condition.

3 ST SUPG method

In the ST methods [6,20–23,29], the formulation is written over a sequence of \( N \) ST slabs \( Q_n \), where \( Q_n \) is the slice of the ST domain between the time levels \( t_n \) and \( t_{n+1} \). The lateral boundary \( P_n \) will have complementary subsets where essential and natural boundary conditions are enforced, just like how it is with \( \Gamma_t \). At each time step, the integrations are performed over \( Q_n \). The functions are continuous within an ST slab, but discontinuous from one ST slab to another, and the superscripts “-” and “+” will indicate the values of the functions just below and just above the time level. The trial solution and test function spaces are defined over \( Q_n \), by using ST polynomials. Each \( Q_n \) is decomposed into elements \( Q_{n}^{el} \), where \( e = 1, 2, \ldots, (n_{el})_n \). The subscript \( n \) used with \( n_{el} \) is for the general case where the number of ST elements may change from one ST slab to another.

We assume that we have constructed some suitably-defined finite-dimensional trial solution and test function spaces \((S_{\phi}^{h})_n\) and \((V_{\phi}^{h})_n\). The ST SUPG formulation of Eq. (1) can be written as follows: given \((\phi^{h})_n^-,\) find \( \phi^h \in (S_{\phi}^{h})_n \), such that \( \forall w_{\phi}^{h} \in (V_{\phi}^{h})_n \):

\[
\int_{Q_n} w_{\phi}^{h} \left( \frac{\partial \phi^{h}}{\partial t} + \mathbf{u}^{h} \cdot \nabla \phi^{h} \right) dQ + \int_{Q_n} \nabla w_{\phi}^{h} \cdot \nu \nabla \phi^{h} dQ
- \int_{(P_n)_\phi} w_{\phi}^{h} h_{\phi}^{h} dP + \int_{\Omega_n} \left( (w_{\phi}^{h})^{+}_n - (\phi^{h})^{+}_n \right) d\Omega
+ \sum_{e=1}^{(n_{el})_n} \int_{Q_{n}^{el}} (r_{\text{SUPG}})_\phi \left( \frac{\partial w_{\phi}^{h}}{\partial t} + \mathbf{u}^{h} \cdot \nabla w_{\phi}^{h} \right) r_A \left( \phi^{h} \right) dQ
+ \sum_{e=1}^{(n_{el})_n} \int_{Q_{n}^{el}} \nabla w_{\phi}^{h} \cdot \mathbf{K}_{\text{DC}} \cdot \nabla \phi^{h} dQ = 0, \tag{2}
\]
where
\[ r_A(\phi^h) = \frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \phi^h - \nabla \cdot (\nu \nabla \phi^h), \tag{3} \]

and \((\tau_{\text{SUPG}})_0\) is the SUPG stabilization parameter, which will be given in “Appendix A.1”. The tensor \(K_{\text{DC}}\) is for DC, and that will be given in “Appendix A.2”. We start with \((\phi^h)_0 = \phi_0\) and apply the formulation sequentially to all ST slabs \(Q_0, Q_1, Q_2, \ldots, Q_{N-1}\).

4 Element length

Element length is a key component of the stabilization and DC parameters. For the purpose of element length calculation, a scaling tensor \(\mathbf{D}\) was introduced in [172] to take into account the polynomial order of the basis functions and other factors such as the dimensions of the element domain in the parametric space. Considering the non-diagonal nature of \(\mathbf{D}\) in general and the context in [173], the terminology was changed to “transformation” tensor in [173].

4.1 Element metric tensor in space

We first define the Jacobian tensor as
\[ \mathbf{Q} = \frac{\partial \mathbf{x}}{\partial \xi}. \tag{4} \]

Then, based on that and \(\mathbf{D}\), we define the tensor
\[ \hat{\mathbf{Q}} = \mathbf{Q} \cdot \mathbf{D}^{-1}, \tag{5} \]

where
\[ \mathbf{D} = \frac{\partial \hat{\xi}}{\partial \xi}. \tag{6} \]

with \(\hat{\xi}\) representing our preferred parametric space. For a unit vector \(\mathbf{r}\) representing a direction in the physical space, we define the corresponding direction vector in the preferred parametric space as
\[ \hat{\mathbf{r}} = \frac{\mathbf{r} \cdot \hat{\mathbf{Q}}^{-T}}{\| \mathbf{r} \cdot \hat{\mathbf{Q}}^{-T} \|}. \tag{7} \]

From that, the directional element length is defined as
\[ h_{\text{RQD}} = 2 \left\| \hat{\mathbf{r}} \cdot \hat{\mathbf{Q}}^T \right\| = 2 \left\| \mathbf{r} \cdot \hat{\mathbf{Q}}^{-T} \cdot \hat{\mathbf{Q}}^T \right\| \left\| \mathbf{r} \cdot \hat{\mathbf{Q}}^{-T} \right\|^{-1}. \tag{8} \]

For notational convenience, we introduce the metric tensor
\[ \mathbf{G} = \hat{\mathbf{Q}}^{-T} \cdot \hat{\mathbf{Q}}^{-1}, \tag{12} \]

and with that,
\[ h_{\text{RQD}} = 2 (\mathbf{r} : \mathbf{G})^{-\frac{1}{2}}. \tag{13} \]

Remark 1 What we get from this derivation with \(\mathbf{D} = \mathbf{I}\) has been used in many methods of calculating the stabilization parameters (see, for example, [29]). 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:
\[ h_{\text{MIN}} \equiv 2 \min_{\mathbf{r}} \left( (\mathbf{r} : \mathbf{G})^{-\frac{1}{2}} \right), \tag{14} \]
\[ h_{\text{MAX}} \equiv 2 \max_{\mathbf{r}} \left( (\mathbf{r} : \mathbf{G})^{-\frac{1}{2}} \right). \tag{15} \]

They are equivalent to
\[ h_{\text{MIN}} = 2 \left( \max_{\mathbf{r}} (\mathbf{r} : \mathbf{G}) \right)^{-\frac{1}{2}}, \tag{16} \]
\[ h_{\text{MIN}} = 2 \left( \lambda_{\text{max}} (\mathbf{G}) \right)^{-\frac{1}{2}}, \tag{17} \]

and
\[ h_{\text{MAX}} = 2 \left( \min_{\mathbf{r}} (\mathbf{r} : \mathbf{G}) \right)^{-\frac{1}{2}}, \tag{18} \]
\[ h_{\text{MAX}} = 2 \left( \lambda_{\text{min}} (\mathbf{G}) \right)^{-\frac{1}{2}}, \tag{19} \]

where \(\lambda_{\text{max}}\) and \(\lambda_{\text{min}}\) are the maximum and minimum eigenvalues of the argument tensor.

Remark 2 In the implementation, we take measures to keep the calculated element length between \(h_{\text{MIN}}\) and \(h_{\text{MAX}}\).

4.2 Element metric tensor in the ST framework

The ST Jacobian tensor is
\[ \mathbf{Q}^{\text{ST}} = \begin{bmatrix} \frac{\partial t}{\partial \xi} & \frac{\partial t}{\partial \eta} & \frac{\partial t}{\partial \zeta} \end{bmatrix} \mathbf{Q}. \tag{20} \]
where \( \theta \) is the parametric coordinate in time, and \( v \) is the mesh velocity:

\[
v = \frac{\partial x}{\partial t} |_{\xi}.
\]

(22)

The ST transformation tensor is given as

\[
D_{ST} = \begin{bmatrix} D_{\theta} & 0^T \\ 0 & D \end{bmatrix},
\]

(23)

and the transformation becomes

\[
\hat{Q}_{ST} = Q_{ST} \cdot (D_{ST})^{-1}.
\]

(24)

The ST metric tensor is defined as

\[
G_{ST} = (\hat{Q}_{ST})^{-T} \cdot (Q_{ST})^{-1}.
\]

(25)

## 5 Formulating D for B-spline elements

We explain the concept in 1D, and that can be applied to the multiple dimensions. The parametric space is set to \(-1 \leq \xi \leq 1\). We write the \( p \)th order B-spline basis functions as \( N^P_a \). A position is expressed as

\[
x^h(\xi) = \sum_{a=0}^{p} x_a N^P_a(\xi).
\]

(26)

From that, we obtain

\[
Q = \sum_{a=0}^{p} \frac{\partial N^P_a}{\partial \xi} x_a.
\]

(27)

By using the Bézier extraction operator, B-spline basis functions can be represented as

\[
N^P_a(\xi) = \sum_{b=0}^{p} C_{a,b} B^P_b(\xi),
\]

(28)

where \( B^P_b \) are the \( p \)th order Bernstein polynomials, and \( C_{a,b} \) are the coefficients constant within an element.

From Eq. (26), we write

\[
x^h(\xi) = \sum_{a=0}^{p} x_a N^P_a(\xi),
\]

(29)

\[
\Xi(\xi) = \sum_{b=0}^{p} \hat{\xi}_b B^P_b(\xi),
\]

(34)

where the Bézier control points are

\[
\hat{\xi}_b = \frac{\Delta \hat{\xi}}{p} b, \quad b = 0, \ldots, p.
\]

(35)

and \( \Delta \hat{\xi} \) is the Bézier-element length in the parametric space.

The corresponding control points are

\[
\xi_a = \frac{\Delta \hat{\xi}}{p} \sum_{b=0}^{p} b \left[ C^{-1} \right]_{b,a}.
\]

(36)

Figure 1 illustrates the relationship between the control points and Bézier control points.

Then the effective element length for \( a = 1, \ldots, p \) can be calculated as

\[
\Delta \xi_a = \hat{\xi}_a - \hat{\xi}_{a-1}
\]

(37)

\[
= \frac{\Delta \hat{\xi}}{p} \sum_{b=0}^{p} b \left( \left[ C^{-1} \right]_{b,a} - \left[ C^{-1} \right]_{b,a-1} \right).
\]

(38)

From that, we define the ratio between the Bézier-element length in the parametric space and the effective element length
length. We propose three versions of the ratio: “RQD-MAX,” “RQD-MIN” and “RQD-EL.”

5.1 RQD-MAX

In this version,

\[
D = \frac{\Delta \hat{\xi}}{\min_{a=1,...,p} \Delta \xi_a}
\]

\[
= p \left( \min_{a=1,...,p} \sum_{b=0}^{p} b \left( \begin{bmatrix} c^{-1} \end{bmatrix}_{ba} - \begin{bmatrix} c^{-1} \end{bmatrix}_{ba-1} \right) \right)^{-1}
\]

\[
= p \max_{a=1,...,p} \left( \sum_{b=0}^{p} b \left( \begin{bmatrix} c^{-1} \end{bmatrix}_{ba} - \begin{bmatrix} c^{-1} \end{bmatrix}_{ba-1} \right) \right)^{-1}
\]

5.2 RQD-MIN

In this version,

\[
D = \frac{\Delta \hat{\xi}}{\max_{a=1,...,p} \Delta \xi_a}
\]

\[
= p \left( \max_{a=1,...,p} \sum_{b=0}^{p} b \left( \begin{bmatrix} c^{-1} \end{bmatrix}_{ba} - \begin{bmatrix} c^{-1} \end{bmatrix}_{ba-1} \right) \right)^{-1}
\]

\[
= p \min_{a=1,...,p} \left( \sum_{b=0}^{p} b \left( \begin{bmatrix} c^{-1} \end{bmatrix}_{ba} - \begin{bmatrix} c^{-1} \end{bmatrix}_{ba-1} \right) \right)^{-1}
\]

5.3 RQD-EL

In this version,

\[
D = \Delta \hat{\xi} \left( \frac{1}{p} \sum_{a=1}^{p} \Delta \xi_a \right)^{-1}
\]

\[
= p^2 \left( \sum_{b=0}^{p} b \left( \begin{bmatrix} c^{-1} \end{bmatrix}_{b,p} - \begin{bmatrix} c^{-1} \end{bmatrix}_{b,0} \right) \right)^{-1}
\]

Remark 3 Although the 1D expressions we have here for RQD-MIN, RQD-MAX and RQD-EL give the same element-length values as those given by the expressions in [172], the expression forms are rather different. The forms we have here are outcome of a clear and convincing derivation. They are more suitable for element-level evaluation. They have straightforward extensions to multiple dimensions.

5.4 Extension to multiple dimensions

We split the Bézier extraction operator by direction and express the basis functions as

\[
N_a(\xi) = \prod_{i=1}^{n_{pd}} N_{d_i}^{p'}(\xi^i)
\]

\[
= \prod_{i=1}^{n_{pd}} \sum_{b=0}^{p'} C_{d_i,b} B_{b}^{p'}(\xi^i),
\]

where \( n_{pd} \) is the number of parametric dimensions, the superscript \( i \) indicates the direction, and \( d_i \) is the mapping table from the element control-point index to the corresponding local index in the \( i \)th direction. From the Bézier extraction operator \( C \) for direction \( i \), we get the coefficient \( D' \), and \( D_{ij} = D' \delta_{ij} \) (no sum), where \( \delta_{ij} \) is the Kronecker delta.

6 Test computations

6.1 Scaling study in 1D

Because the 1D expressions for RQD-MIN, RQD-MAX and RQD-EL give the same element length values as those given by the expressions in [172] (see Remark 3), for completeness, we include here the results from the 1D scaling study reported in [172]. The results included here are for RQD-MAX, RQD-EL, and RQD-1, which has \( D = 1 \).

The flow is from left to right, \( |u| = 1 \), and \( v = 0.0 \). We are working with nondimensional numbers. The boundary conditions are \( \phi = 1 \) at \( x = 0 \) and \( \phi = 0 \) at \( x = 1 \). We use 4 different meshes, all consisting of a single uniform B-spline
patch, with polynomial order 8, and with the number of elements ($n_{el}$) ranging from 1 to 4. The stabilization parameter is as given by Eq. (51) (see “Appendix A.1”), with $\tau_{SUGN2}$ excluded. The method includes the YZ$\beta$ DC, and we test both $\beta = 1$ and $\beta = 2$ (see “Appendix A.2”), with $\phi_{REF} = 1$, which is the expected value of $\phi_{MAX} - \phi_{MIN}$. The basis functions are linear in time, and the time-step size is 0.05. Figures 2 and 3 show the steady solutions.

We see that RQD-1 yields results that are overdamped. We get good solutions with RQD-EL and $\beta = 1$, but there are small oscillations when the problem is underresolved, even though the oscillations are not visible in the plots. Overall, we see RQD-MAX with $\beta = 1$ as giving the best solution. It gives better and better solutions with higher and higher number of elements, and there is no oscillation even when we have just one element.
6.2 Advection skew to the mesh

The problem setup, which originates from [1] and was also used in [14] to test the SUPG method supplemented with new DC methods, is shown in Figure 4.

The domain size is 1.0×1.0, and the test computations are for \( \theta = 63.4^\circ \) and \( \nu = 1.0 \times 10^{-6} \). We are working with nondimensional numbers. Even though we have a simple, square domain in the test computations, the mesh is complex because it is made of many NURBS patches, with different shapes and orientations (see Figure 5). The B-spline patches are with polynomial order 8, because that makes it easier to see the performance differences between the different scaling options. The stabilization parameter is as given by Eq. (51) (see “Appendix A.1”), with \( r_{\text{SUPG2}} \) excluded. The method includes the YZ\( \beta \) DC, and we test both \( \beta = 1 \) and \( \beta = 2 \), with \( \phi_{\text{REF}} = 1 \). The basis functions are linear in time, and the time-step size is 0.05. We compute with and compare RQD-MAX, RQD-EL, and RQD-I, which has \( D = I \).

Figures 6, 7 and 8 show the results for \( \beta = 1 \), and Figs. 9, 10 and 11 for \( \beta = 2 \). The boundary layer representations are
sharper with RQD-MAX, as also observed in the 1D tests. The internal layers lose their sharpness noticeably with RQD-I and RQD-EL. We see some oscillations across mesh lines with $\beta = 2$ and RQD-MAX, but they stay within ±0.8%. It may be possible to clean that up with a better choice for the $Y$ value in Eq. (60) (see “Appendix A.2”).

6.3 Zalesak’s solid-body rotation over a square domain

We compute the Zalesak’s solid-body rotation problem [174] with the same computational domain and mesh we used for the advection skew to the mesh. The stabilization parameter is as given by Eq. (49) (see “Appendix A.1”). The method includes the YZ$\beta$ DC, and we test both $\beta = 1$ and $\beta = 2$, with $\phi_{\text{REF}} = 1$. The basis functions are linear in time, and the
The number of time steps per rotation is 320 (see also [22] for the problem setup). We compute with and compare RQD-MAX and RQD-I.

Figure 12 shows the initial condition. Figures 13, 14, 15 and 16 show, for the four test combinations, the solution after a full rotation. For both $\beta = 1$ and $\beta = 2$, RQD-MAX performs better. As expected, $\beta = 2$ is better in this test problem because beyond the initial condition, there is no mechanism to sustain the discontinuity.

7 Concluding remarks

Stabilization parameters embedded in most of the stabilized and VMS methods almost always involve element lengths, most of the time in specific directions, such as the direction of the flow or solution gradient. These methods are sometimes supplemented with DC methods, and the DC parameters embedded in those also most of the time involve directional element lengths. Until recently, expressions for element lengths and stabilization and DC parameters originally intended for finite element discretization were being used also for isogeometric discretization. Recently, expressions targeting isogeometric discretization were introduced for ST and semi-discrete computations, and the expressions
Fig. 16  Zalesak’s solid-body rotation over a square domain, with $\beta = 2$. Scaling: RQD-MAX. Solution after a full rotation

are also applicable to finite element discretization. The key stages of deriving those directional element length expressions were mapping the direction vector from the physical (ST or space-only) element to the parent element in the parametric space, accounting for the discretization spacing along each of the parametric coordinates, and mapping what has been obtained back to the physical element. Going beyond that and targeting B-spline meshes for complex geometries, we introduced in this article new element length expressions, which are outcome of a clear and convincing derivation and more suitable for element-level evaluation. The new expressions are based on a preferred parametric space and a transformation tensor that represents the relationship between the integration and preferred parametric spaces. The test computations we presented for advection-dominated cases, including 2D computations with complex meshes, show that the new element length expressions result in good solution profiles.

Acknowledgements  This work was supported (first and third authors) in part by Grant-in-Aid for Challenging Exploratory Research 16K13779 from Japan Society for the Promotion of Science (JSPS); 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); Grant-in-Aid for Scientific Research (A) 18H04100 from JSPS; and Rice–Waseda research agreement. The computational method parts of the work were also supported (third author) in part by ARO Grant W911NF-17-1-0046 and Top Global University Project of Waseda University.

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

Appendix A: Stabilization and DC parameters

A.1 Stabilization parameter

The stabilization parameter $(\tau_{\text{SUPG}})_{\phi}$ is made of two components:

$$(\tau_{\text{SUPG}})_{\phi} = \left(\tau_{\text{SUPG}12}^{-2} + \tau_{\text{SUPG}3}^{-2}\right)^{-\frac{1}{2}}. \quad (49)$$

The first component is given as

$$\tau_{\text{SUPG}12}^{-2} = \left[\begin{array}{c} 1 \\
\mathbf{u} \end{array}\right] \left[\begin{array}{c} 1 \\
\mathbf{u} \end{array}\right] : \mathbf{G}^\text{ST}. \quad (50)$$

Alternatively, we can split $\tau_{\text{SUPG}12}$ into two and define $(\tau_{\text{SUPG}})_{\phi}$ to be made of three components:

$$(\tau_{\text{SUPG}})_{\phi} = \left(\tau_{\text{SUPG}1}^{-2} + \tau_{\text{SUPG}2}^{-2} + \tau_{\text{SUPG}3}^{-2}\right)^{-\frac{1}{2}}, \quad (51)$$

where

$$\tau_{\text{SUPG}1}^{-2} = (\mathbf{u} - \mathbf{v}) (\mathbf{u} - \mathbf{v}) : \mathbf{G}, \quad (52)$$

$$\tau_{\text{SUPG}2} = \frac{\Delta t}{2}, \quad (53)$$

and $\Delta t$ is the time-step size. The last component is defined as

$$\tau_{\text{SUPG}3}^{-1} = \mathbf{r} : \mathbf{G}, \quad (54)$$

where

$$\mathbf{r} = \frac{\nabla \phi}{\|\nabla \phi\|}, \quad (55)$$

and because $\nabla \phi$ could be zero, we modify the expression for $\tau_{\text{SUPG}3}$ as

$$\tau_{\text{SUPG}3}^{-1} = \nu \mathbf{r} : \mathbf{G} + \left(1 - r_{\text{reg}}^2\right) 4 h_{\text{MIN}}^{-2}. \quad (56)$$
Here

\[
\begin{align*}
\vec{r}_{\text{reg}} &= \frac{\nabla \phi}{\|\nabla \phi\| + \epsilon \|\nabla \phi\|_0}, \tag{57} \\
\vec{r}_{\text{reg}} &= \left\|\vec{r}_{\text{reg}}\right\| \leq 1, \tag{58}
\end{align*}
\]

(\|\nabla \phi\|_0) is a reference value of \|\nabla \phi\|, and \epsilon is a small constant.

### A.2 DC parameter

The YZβ DC [153–155] has the form

\[
\begin{align*}
K_{\text{DC}} &= K_{\text{DC}} \delta, \tag{59} \\
K_{\text{DC}} &= \left|Y^{-1} Z\right| (Y^{-1} \|\nabla \phi\|)^{\beta - 2} (h_{\text{DC}})^{\beta}, \tag{60}
\end{align*}
\]

where \(Y = \phi_{\text{REF}}\), which is a reference value,

\[
Z = r_A \left(\phi^h\right), \tag{61}
\]

and \(\beta = 1\) or 2. The element length \(h_{\text{DC}}\) is set to \(h_{\text{RQD}}\), or some other direction-dependent element length expression.

**Remark 4** The expression for \(K_{\text{DC}}\) has been slightly modified from its original form [153–155].

### References

1. Hughes TJR, Brooks AN (1979) A multi-dimensional upwind scheme with no crosswind diffusion. In: Hughes TJR (ed) Finite element methods for convection dominated flows, AMD, ASME, New York, vol 34, pp 19–35
2. 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
3. Tezduyar TE, Hughes TJR (1982) Development of time-accurate finite element techniques for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. NASA Technical Report NASA-CR-204772, NASA. http://www.researchgate.net/publication/24313718/
4. Tezduyar TE, Hughes TJR (1983) Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations. In Proceedings of AIAA 21st aerospace sciences meeting, AIAA paper 83-0125, Reno, Nevada. https://doi.org/10.2514/6.1983-125
5. Hughes TJR, Tezduyar TE (1984) Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. Comput Methods Appl Mech Eng 45:217–284. https://doi.org/10.1016/0045-7825(84)90157-9
6. 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
7. Tezduyar TE, Mittal S, Ray SE, Shih R (1992) Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. Comput Methods Appl Mech Eng 95:221–242. https://doi.org/10.1016/0045-7825(92)90141-6
8. Hughes TJR, Franca LP, Balestra M (1986) A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. Comput Methods Appl Mech Eng 59:85–99
9. 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
10. Hughes TJR, Oberai AA, Mazzei L (2001) Large eddy simulation of turbulent channel flows by the variational multiscale method. Phys Fluids 13:1784–1799
11. 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
12. 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
13. Hughes TJR, Mallet M, Mizukami A (1986) A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. Comput Methods Appl Mech Eng 54:341–355
14. Tezduyar TE, Park YJ (1986) Discontinuity capturing finite element formulations for nonlinear convection–diffusion–reaction equations. Comput Methods Appl Mech Eng 59:307–325. https://doi.org/10.1016/0045-7825(86)90003-4
15. Hughes TJR, Franca LP, Mallet M (1987) A new finite element formulation for computational fluid dynamics: VI. Convergence analysis of the generalized SUPG formulation for linear time-dependent multi-dimensional advective-diffusive systems. Comput Methods Appl Mech Eng 63:97–112
16. Le Beau GJ, Tezduyar TE (1991) Finite element computation of compressible flows with the SUPG formulation. In: Advances in finite element analysis in fluid dynamics, FED, ASME, New York, vol 123, pp 21–27
17. 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
18. Tezduyar TE, Behr M, Liu J (1992) A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: I. The concept and the preliminary numerical tests. Comput Methods Appl Mech Eng 94(3):339–351. https://doi.org/10.1016/0045-7825(92)90059-S
19. Tezduyar TE, Behr M, Mittal S, Liu J (1992) A new strategy for finite element computations involving moving boundaries and interfaces - the deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. Comput Methods Appl Mech Eng 94(3):353–371. https://doi.org/10.1016/0045-7825(92)90060-W
20. 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
21. 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
22. 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
23. Takizawa K, Tezduyar TE (2012) Space-time fluid–structure interaction methods. Math Models Methods
dynamics and cover separation. Math Models Methods Appl Sci 23:307–338. https://doi.org/10.1142/S0212206213400058
90. Takizawa K, Tezduyar TE, Boswell C, Kolesar R, Montel K (2014) FSI modeling of the reseeded stages and disreeling of the Orion spacecraft parachutes. Comput Mech 54:1203–1220. https://doi.org/10.1007/s00446-014-1052-y
91. 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/s00446-014-1069-2
92. 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/s00446-014-1108-z
93. 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/s00446-011-0589-2
94. 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 53:1–15. https://doi.org/10.1007/s00446-011-0388-x
95. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C, Ølseth 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
96. Takizawa K (2014) Computational engineering analysis with the new-generation space-time methods. Comput Mech 54:193–211. https://doi.org/10.1007/s00446-014-0999-z
97. Takizawa K, Tezduyar TE, McINTyre S, Kostov N, Kolesar R, Hablutezel C (2014) Space-time VMS computation of wind-turbine rotor and tower aerodynamics. Comput Mech 53:1–15. https://doi.org/10.1007/s00446-013-0888-x
98. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C, Ølseth 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
99. Takizawa K, Henicke B, Puntel A, Spielman T, Tezduyar TE (2012) Space-time computational techniques for the aerodynamics of flapping wings. J Appl Mech 79:011093. https://doi.org/10.1115/1.4005073
100. 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/s00446-012-0759-x
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, 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/s00446-012-0758-y
103. 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/s00446-014-0980-x
104. 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/s00446-013-0935-7
105. 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/s00446-014-1095-0
106. 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
107. 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
108. 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
109. 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
110. 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, New York, pp 379–386. https://doi.org/10.1007/978-3-319-40827-9_29. ISBN: 978-3-319-40825-5
111. Takizawa K, Tezduyar TE, Uchikawa H, Terahara T, Sasaki T, Shiozaki K, Yoshida A, Komiyama 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, New York, pp 29–89. https://doi.org/10.1007/978-3-319-96469-0_2. ISBN: 978-3-319-96468-3
112. 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
113. Takizawa K, Bazilevs Y, Tezduyar TE, Hsu M-C (2019) Computational cardiovascular flow analysis with the variational multiscale methods. J Adv Eng Comp 3:366–405. https://doi.org/10.25073/jaec.20193.245
114. 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/s00446-014-1046-9
115. 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, New York, pp 159–178. https://doi.org/10.1007/978-3-319-40827-9_13. ISBN: 978-3-319-40825-5
116. 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, New York, pp 77–99. https://doi.org/10.1007/978-3-319-59548-1_6. ISBN: 978-3-319-59547-4
117. 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
118. Yu Y, Zhang YJ, Takizawa K, Tezduyar TE, Sasaki T (October 2019) Anatomically realistic lumen motion representation in patient-specific space-time isogeometric flow analysis of coronary arteries with time-dependent medical-image data. Comput Mech. published online. https://doi.org/10.1007/s00446-019-01774-4

119. 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

120. Takizawa K, Tezduyar TE, Kuraishi T, Tabata S, Takagi H (2016) Computational thermo-fluid analysis of a disk brake. Compute Mech 57:965–977. https://doi.org/10.1007/s00446-016-1272-4

121. Takizawa K, Tezduyar TE, Hattori H (2017) Computational analysis of flow-driven string dynamics in turbomachinery. Compute Fluids 142:109–117. https://doi.org/10.1016/j.compfluid.2016.02.019

122. Komiya K, Kanai T, Otoro 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. In: IOP conference series earth and environmental science, vol 240. https://doi.org/10.1088/1755-1315/240/6/062014

123. 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/S0218202519400101X

124. Takizawa K, Tezduyar TE, Otoro 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

125. Otoro 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

126. Otoro 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, New York, pp 399–434. https://doi.org/10.1007/978-3-319-96469-0_10. ISBN: 978-3-319-96468-3

127. Otoro 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

128. Otoro 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/s00446-019-01722-2

129. Takizawa K, Tezduyar TE, Asada S, Kuraishi T (2016) Space-time method for flow computations with slip surfaces and topology changes (ST-SF-TC). Comput Fluids 141:124–134. https://doi.org/10.1016/j.compfluid.2016.05.006

130. 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, New York, pp 337–376. https://doi.org/10.1007/978-3-319-96469-0_8. ISBN:978-3-319-96468-3

131. 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/s00446-018-1642-1

132. 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/s00446-019-01746-8

133. 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/S0218202519400021

134. 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

135. 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

136. 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/s00446-018-1595-4

137. Tezduyar TE, Aliabadi SK, Behr M, Mittal S (1994) Massively parallel finite element simulation of compressible and incompressible flows. Comput Models Appl Mech Eng 119:157–177. https://doi.org/10.1016/0045-7825(94)00082-4

138. 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

139. 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/s00446-013-0895-y

140. 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/s00446-013-0919-7

141. 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/s00446-014-1049-6

142. 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, New York, pp 101–122. https://doi.org/10.1007/978-3-319-59548-1_7. ISBN: 978-3-319-59547-4

143. 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/s00446-016-1344-5

144. 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/s00446-018-1651-0

145. 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/s00446-019-01660-4

146. 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/s00446-018-1616-3
147. Akin JE, Tezduyar T, Ungr M, Mittal S (2003) Stabilization parameters and Smagorinsky turbulence model. J Appl Mech 70:2–9. https://doi.org/10.1115/1.1526569

148. Akin JE, Tezduyar TE (2004) Calculation of the advective limit of the SUPG stabilization parameter for linear and higher-order elements. Comput Methods Appl Mech Eng 193:1909–1922. https://doi.org/10.1016/j.cma.2003.12.050

149. Franca LP, Frey SL, Hughes T.J.R. (1992) Stabilized finite element methods: I. Application to the advective–diffusive model. Comput Methods Appl Mech Eng 95:253–276

150. Tezduyar TE, Osaka Y (2000) Finite element stabilization parameters computed from element matrices and vectors. Comput Methods Appl Mech Eng 190:411–430. https://doi.org/10.1016/S0045-7825(00)00211-5

151. Hsu M-C, Bazilevs Y, Calo VM, Tezduyar TE, Hughes TJR (2010) A multiscale finite element formulation with discontinuity-capturing for turbulence models with dominant reactionlike terms. Comput Methods Appl Mech Eng 199:828–840. https://doi.org/10.1016/j.cma.2009.06.019

152. Tezduyar TE (2001) Adaptive determination of the finite element stabilization parameters. In: Proceedings of the ECCOMAS computational fluid dynamics conference 2001. CD-ROM, Swansea, Wales

153. Tezduyar TE (2004) Finite element methods for fluid dynamics with moving boundaries and interfaces. In: Stein E, Bursted, Hughes T.J.R. (eds) Encyclopedia of computational mechanics, volume 3: fluids, Chapter 17, Wiley, New York. https://doi.org/10.1002/0470091355.ecm069. ISBN: 978-0-470-84699-5

154. Tezduyar TE (2007) Determination of the stabilization and shock-capturing parameters in SUPG formulation of compressible flows. In: Proceedings of the European congress on computational methods in applied sciences and engineering, ECCOMAS 2004. (CD-ROM), Jyvaskyla

155. 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

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

157. 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

158. 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

159. 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

160. 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

161. Rispoli F, Saavedra R, Menichini C, 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

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